Lec 23: DoS and Firewalls

CSED415: Computer Security

Spring 2025

Seulbae Kim

POSTECH

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY

Administrivia

- Lab 05 has been released
  - Final lab assignment
  - Due the end of May 23 (Fri)
- Group project
  - Final presentation: May 27 and 29
  - All teams should submit their final reports, source code, and presentation slides by the end of May 26

Recap

POSTECH

- Previously discussed: Compromising confidentiality and integrity
  - Prerequisites:
    - Skilled attacker
    - Program with vulnerabilities
    - Flawed security policy
    - Entry points and exfiltration channels

How about availability? → Today's topic

Denial of Service (DoS)

Availability and Denial of Service

- Availability
  - Ensuring that a network service is accessible for legitimate users
- Denial of Service (DoS)
  - Attacks designed to disrupt availability, making services inaccessible to legitimate users

Availability and Denial of Service

- Motivations behind DoS attacks
  - Competitive advantage between rival services
  - Financial gain through ransom demands
  - Warfare tactics
  - Personal amusement or revenge

DoS in real life

LILY HAY NEWMAN SECURITY MAR 1, 2018 11:01 AM

GitHub Survived the Biggest DDoS Attack Ever Recorded

On Wednesday, a 1.3Tbps DDoS attack pummeled GitHub for 15-20 minutes. Here's how it stayed online.

ON WEDNESDAY, AT about 12:15 pm EST, 1.35 terabits per second of traffic hit the developer platform GitHub all at once. It was the most powerful distributed denial of service attack recorded to date—and it used an increasingly popular DDoS method, no botnet required.

GitHub briefly struggled with intermittent outages as a digital system assessed the situation. Within 10 minutes it had automatically called for help from its DDoS mitigation service, Akamai Prolexic. Prolexic took over as an intermediary, routing all the traffic coming into and out of GitHub, and sent the data through its scrubbing centers to weed out and block malicious packets. After eight minutes, attackers relented and the assault dropped off.

TRENDING

DoS in real life

POSTECH

Eight-Hour DDoS Attack Struck AWS Customers

Google Cloud Platform suffered issues around the same time as Amazon Web Services but claims they were not caused by DDoS.

Dark Reading Staff, Dark Reading
October 24, 2019

1 Min Read

A significant distributed denial-of-service (DDoS) attack lasting approximately eight hours affected Amazon Web Services yesterday, knocking its S3 service and other services offline between 10:30 a.m. and 6:30 p.m. PDT.

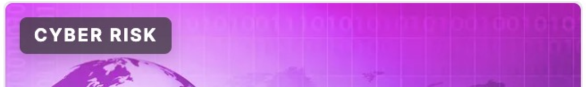
The attack struck AWS's Router 53 DNS Web service, which led to outages for other services that require public DNS resolution: Elastic Load Balancing, Relational Database Service, and Elastic Compute Cloud. AWS alerted customers while the attack was ongoing to inform them of "intermittent errors with resolution of some AWS DNS names." Starting at 5:16 p.m., a small number of specific DNS names experienced a higher error rate. The issues have been resolved.

Editor's Choice

Blinken: Digital Solidarity Is 'North Star' for US Policy

by Karen Spiegelman, Features Editor

MAY 7, 2024 3 MIN READ



DoS in real life

POSTECH

Daryna Antoniuk

January 19th, 2024

News Briefs

Nation-state

Swiss websites hit by DDoS attacks during World Economic Forum in Davos

Swiss websites were hit by a wave of distributed denial-of-service (DDoS) attacks this week, likely orchestrated by pro-Russian hackers.

According to the Swiss National Cybersecurity Centre (NCSC), the attacks temporarily disrupted access to several websites run by the Federal Administration — the government's executive branch.

“The cyberattack was promptly detected and the Federal Administration's specialists took the necessary action to restore access to the websites as quickly as possible,” said NCSC’s statement.

DDoS attacks are aimed at making websites unavailable but do not result in any data being lost or compromised.

A Russian politically-motivated hacker group known as NoName claimed responsibility for the attacks on its Telegram channel.

DoS attack strategies

POSTECH

- Exploiting software flaws
  - Security vulnerabilities can lead to service disruption
    - e.g., exploit a buffer overflow and execute `system("shutdown now");`
- Resource exhaustion
  - Every computing system has limited resources
  - Attacker consumes all resources so legitimate users cannot use them
  - Identifying and targeting system <u>bottlenecks</u> is sufficient to cause disruption

DoS attack strategies

- Bottlenecks
  - The component of the system with the least resources
    - Different components have different resource limits
  - The attacker only needs to exhaust the bottleneck
    - e.g., the component with the least CPU time allocated, the component with the smallest amount of memory allocated, …

DoS targets

- Application-level DoS
  - Attacks on specific applications running on a host
- Network-level DoS
  - Attacks targeting network protocols to impair host connectivity

Application-level DoS

Application-level DoS

- Target application-specific resources
- Attackers exploit resource-handling asymmetry:
  - Small amount of attack input
  - Large amount of resource consumed to handle the input

Resource consumption

- Idea: Force the server to consume all its resources
  - Potential payloads
    - Exhaust filesystem space:

```
int fd = open("/tmp/junk", O_CREAT | O_RDWR, 0664);
char buf[4096];
while (1) {write(fd, buf, 4096);}
```

    - Exhaust main memory:

```
while (1) {malloc(10000000000);}
```

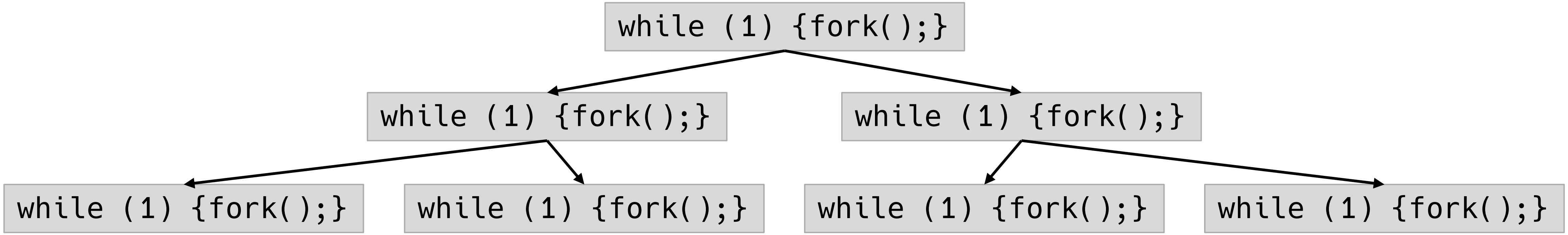
    - Exhaust process table and CPU cycles (fork bomb):

```
while (1) {fork();}
```

Resource consumption

- Idea: Force the server to consume all its resources
  - Fork bomb explained

```
$ man fork
fork() creates a new process by duplicating the calling process. The new process is referred
to as the child process. The calling process is referred to as the parent process.
The child process and the parent process run in separate memory spaces. The child process
is an exact duplicate of the parent process.
```



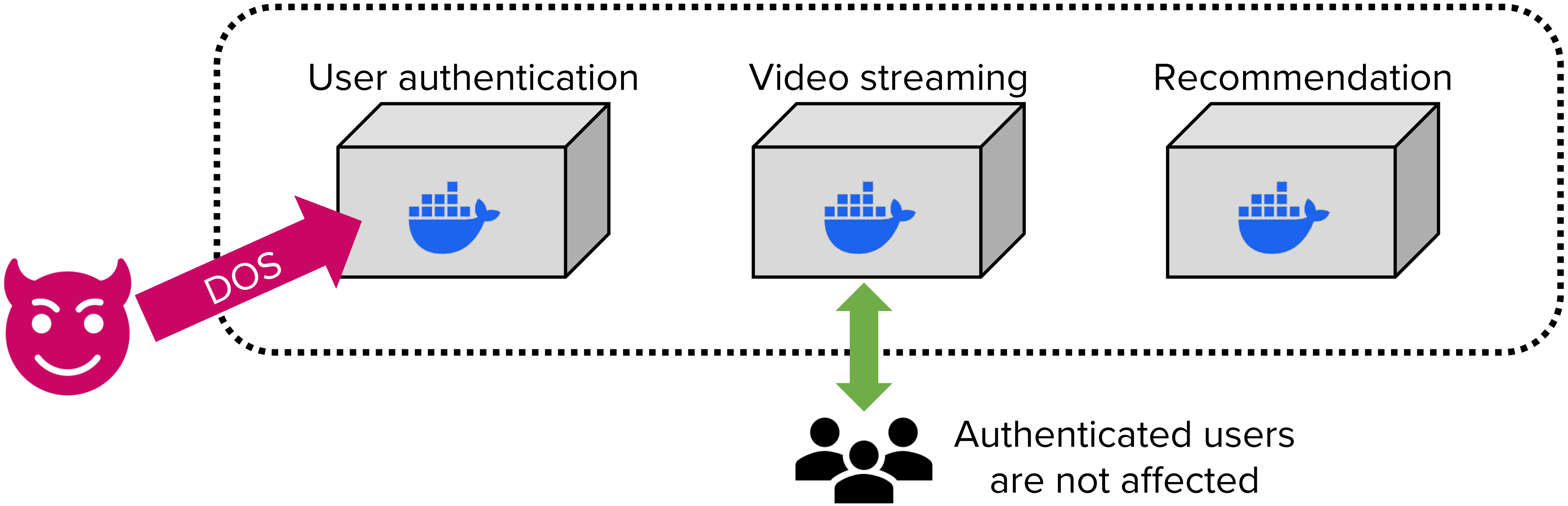
Exhausts OS process table, CPU cycles, and memory

Algorithmic complexity attacks

- Supplying inputs that trigger worst-case scenarios for algorithms and data structures
  - e.g., consider an application that runs quicksort on user data
    - Average time complexity: $O(n\ log(n))$
    - Worst-case: $O(n^2)$ // when?

Defenses against application-level DoS

- Isolation
  - Ensure one user's actions do not affect other users' experience
  - e.g., modularize / containerize services (e.g., Docker)



Defenses against application-level DoS

- Quota allocation
  - Ensure that users can only access a certain amount of resources
    - e.g., `rlimit`: Kernel-level resource limits defined per process

```
struct rlimit lim;
lim.rlim_cur = 512;
lim.rlim_max = 1024;
setrlimit(resource, &lim);
```

→ Sets the process's resource soft limit (initial config) to 512 and hard limit (upper bound) to 1024, where resource can be one of:

- `RLIMIT_AS`: Max size of this process's virtual memory
- `RLIMIT_CPU`: Amount of CPU time this process can consume
- `RLIMIT_FSIZE`: Max size of files that this process may create
- `RLIMIT_NOFILE`: Max # of file descriptors that can be opened by this process
- `RLIMIT_NPROC`: Max # of processes that can be forked by this process

…

Defenses against application-level DoS

- Require proof-of-work
  - Force users to spend some of their resources to make requests
  - Idea: Make DoS expensive for attackers
    - e.g., CAPTCHA
- Overprovisioning
  - Allocate excessive amount of resource
    - A costly solution
  - Applicable if the importance of availability far outweighs the money
    - e.g., CDN (content delivery network) ensures availability by geographically distribute cache servers

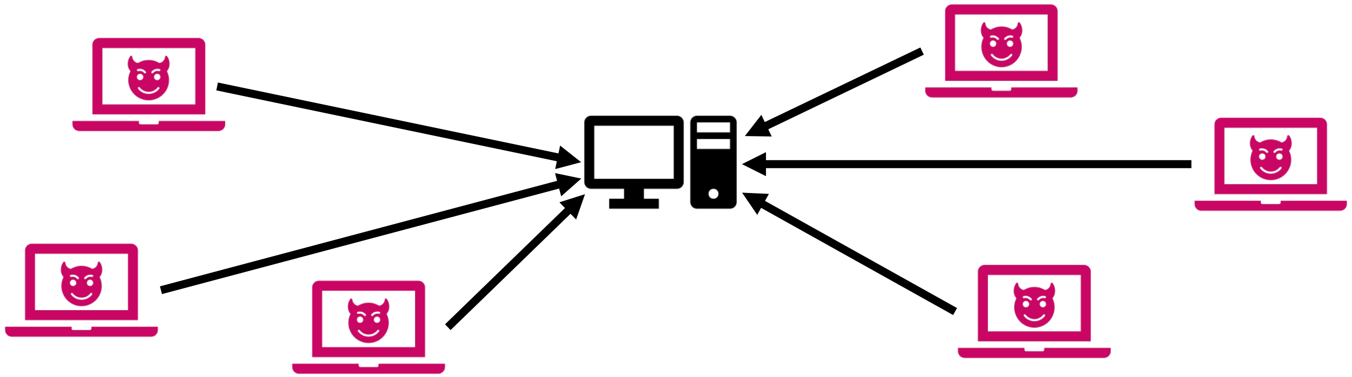
Network-level DoS

Network-Level DoS

- Attack network protocols to affect victim's internet access
  - Overwhelm victim's bandwidth
    - Bandwidth: The amount of data that can be uploaded and downloaded through a channel within a given time
    - Attacker can send <u>large</u> packets
  - Overwhelm victim's packet processing capability
    - Example: The server can process 50 packets per second. Attacker sends the server 500 packets per second.
    - Attacker can send many <u>small</u> packets

Distributed Denial-of-Service (DDoS)

- DDoS: Using multiple systems to overwhelm target system
  - Attacker can consume a huge amount of victim's bandwidth by controlling many bots (i.e., a botnet)
  - Packet filters to distinguish DDoS traffic from normal traffic can be evaded because bots send packets from different locations
    - Blocking several IP addresses is not effective

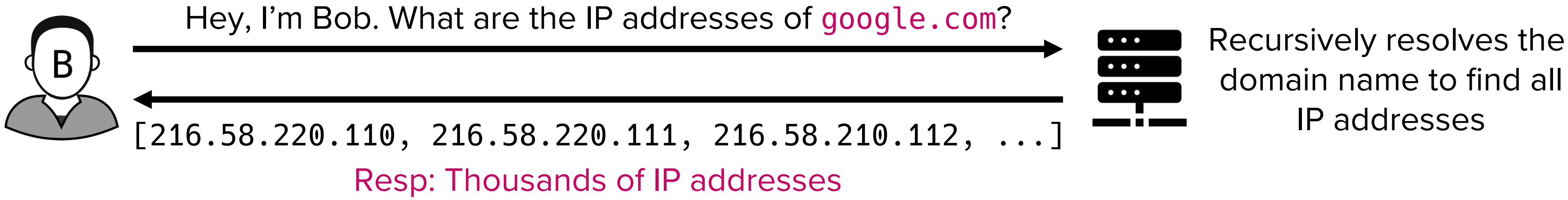


Amplification DoS

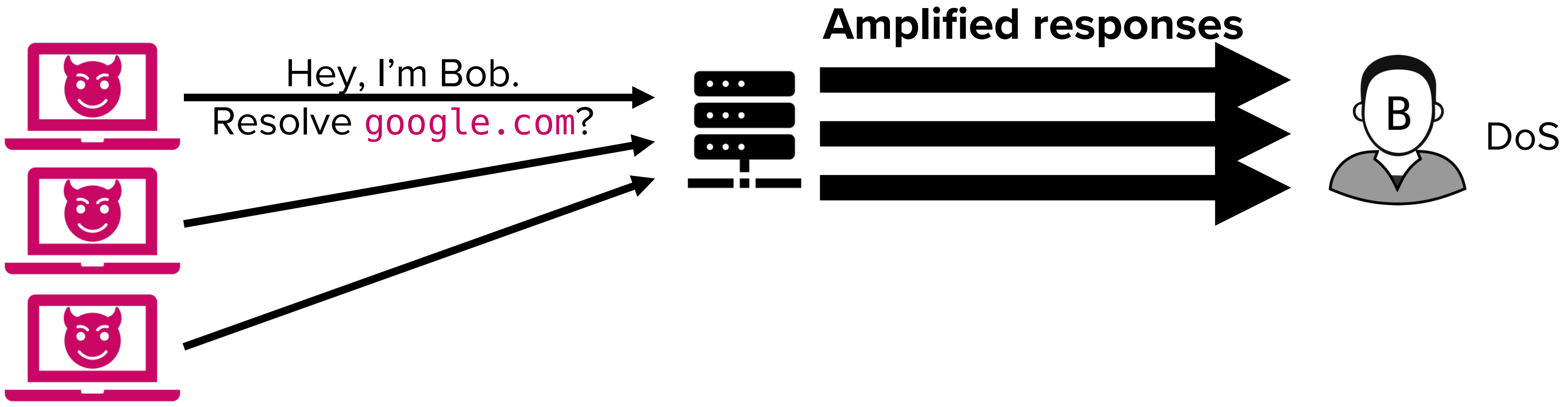
- Amplification DoS: Using an amplifier to overwhelm the target more effectively
  - Some services produce a significantly larger response compared to the size of the request
  - Attackers can send a small request with spoofed source IP address (e.g., disguising the sender as the victim) to exploit such services, causing a large volume of data to be sent to the victim

Amplification DoS

- Example: Domain Name Server (DNS) amplification
  - DNS lookup



  - Attack



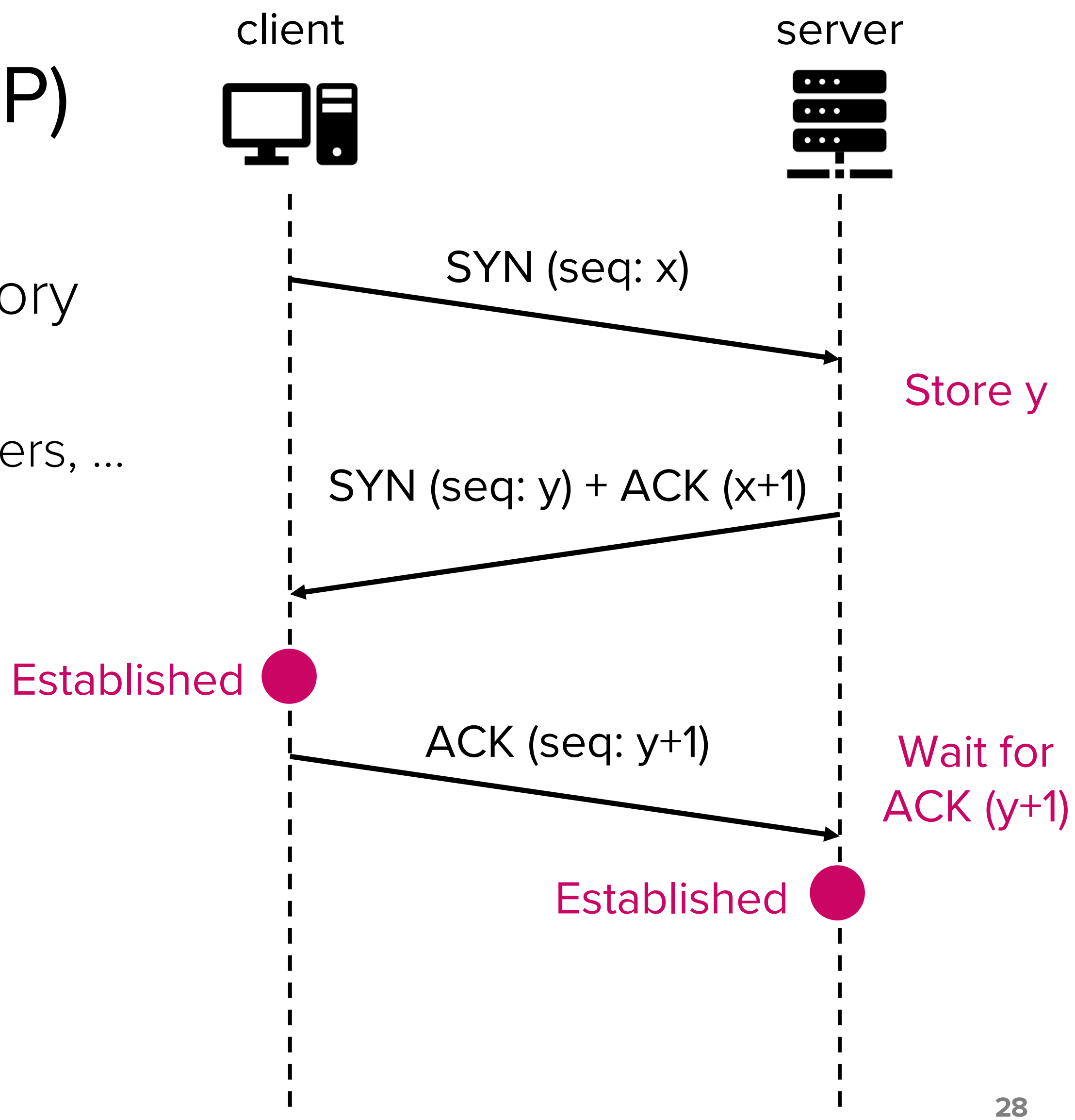
Defenses against network-level DoS

- Packet filtering:
  - Drop packets based on source IP or suspicious patterns
    - e.g., Discard packet if source IP is known attacker's IP
  - Limitations:
    - Attack packets come from multiple IP addresses (DDoS)
    - Attackers can spoof source IP address to make attack packets look like they are coming from multiple IP addresses

Stateful DoS

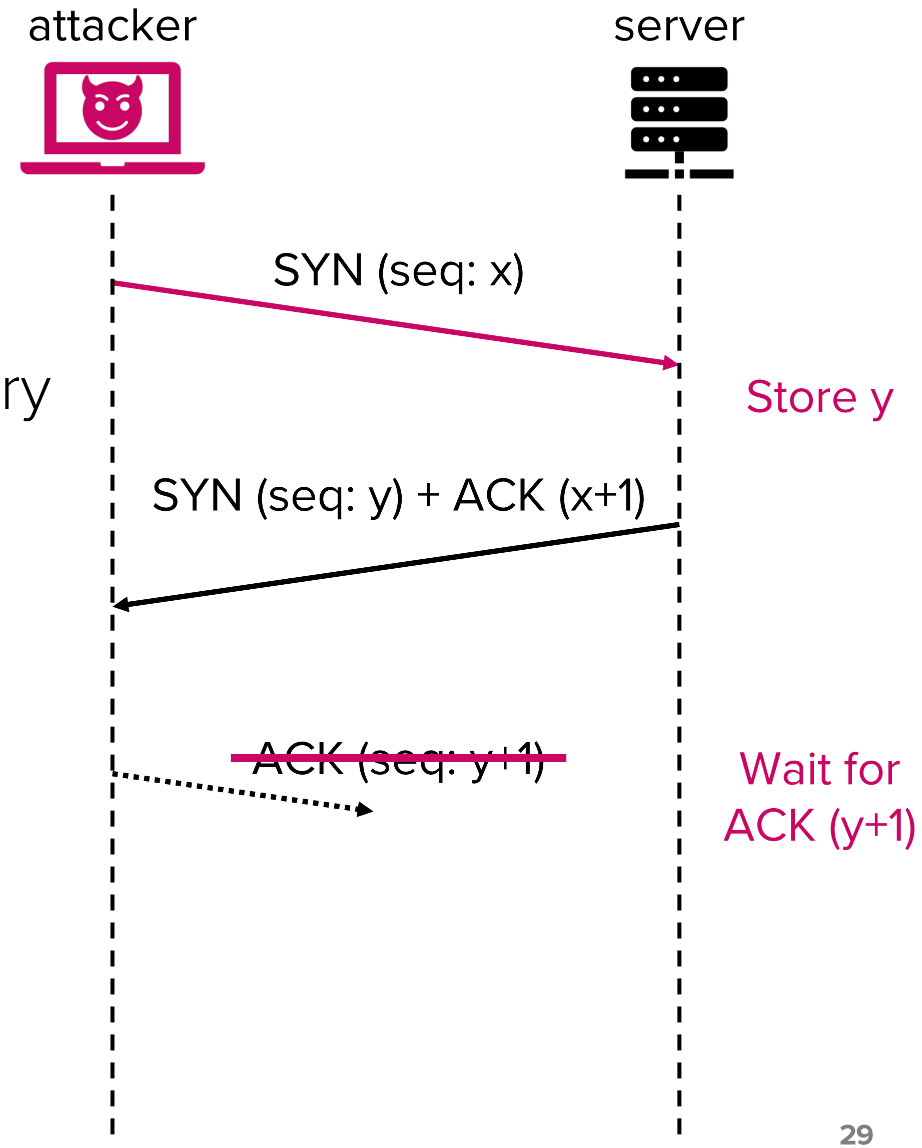
SYN flooding

- Transmission Control Protocol (TCP)
  - Stateful protocol (recall Lecture 13)
    - The server need to allocate some memory for each connection established
      - Store states: sequence numbers, ack numbers, …
    - The server expects an ACK packet



SYN flooding

- SYN flooding attack
  - Attacker establishes many connections with the server
    - Force the server to consume excessive memory
  - Attacker's asymmetric advantage:
    - Sending the initial SYN packet is sufficient to make the server waste its resources
    - In contrast, attacker does not waste much of his/her own resources

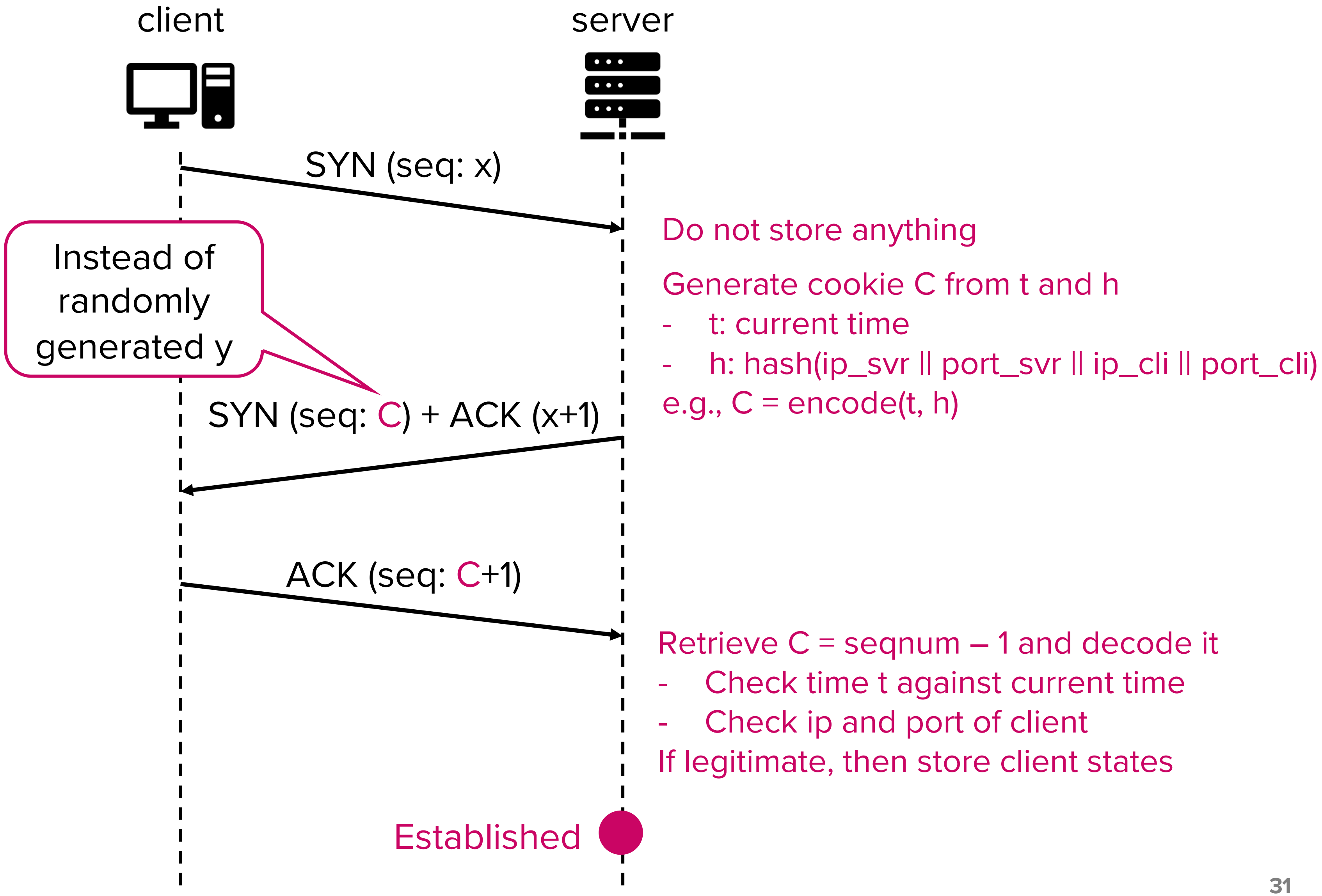


Defenses against SYN flooding

- Overprovisioning: Ensure the server has a plenty of memory
  - Can be expensive
- Filtering: Ensure that only legitimate connections can initiate handshake
  - Can be effective if packets are not spoofed (e.g., by only allowing known source)
  - Same drawbacks as network-level DoS prevention
- How about not storing any state during handshake?
  - SYN cookies (next slide)

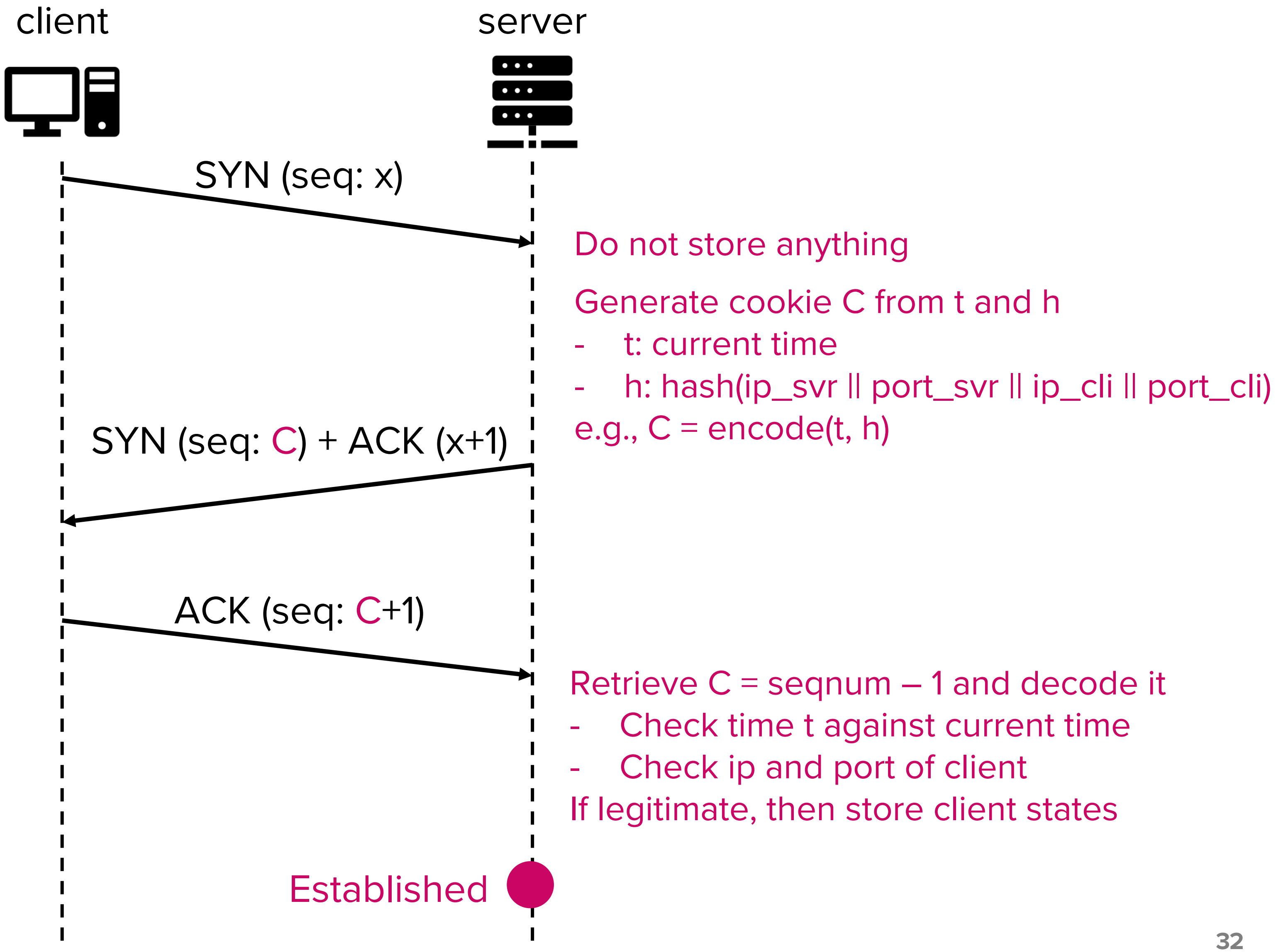
SYN cookies

- Workflow



SYN cookies

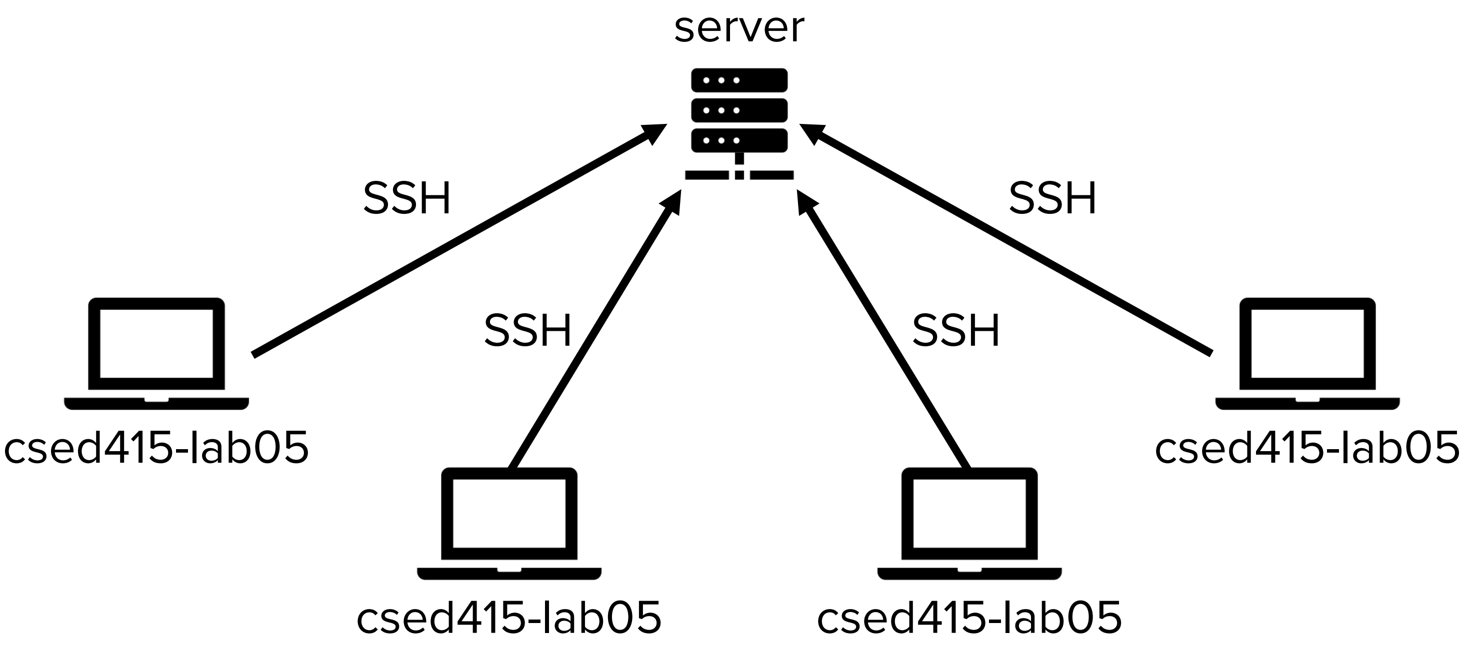
- Observation
  - Attacker no longer has asymmetric advantage
    - Must send SYN, receive and parse resp, and then send ACK



DoS Prevention Case Study: CSED415

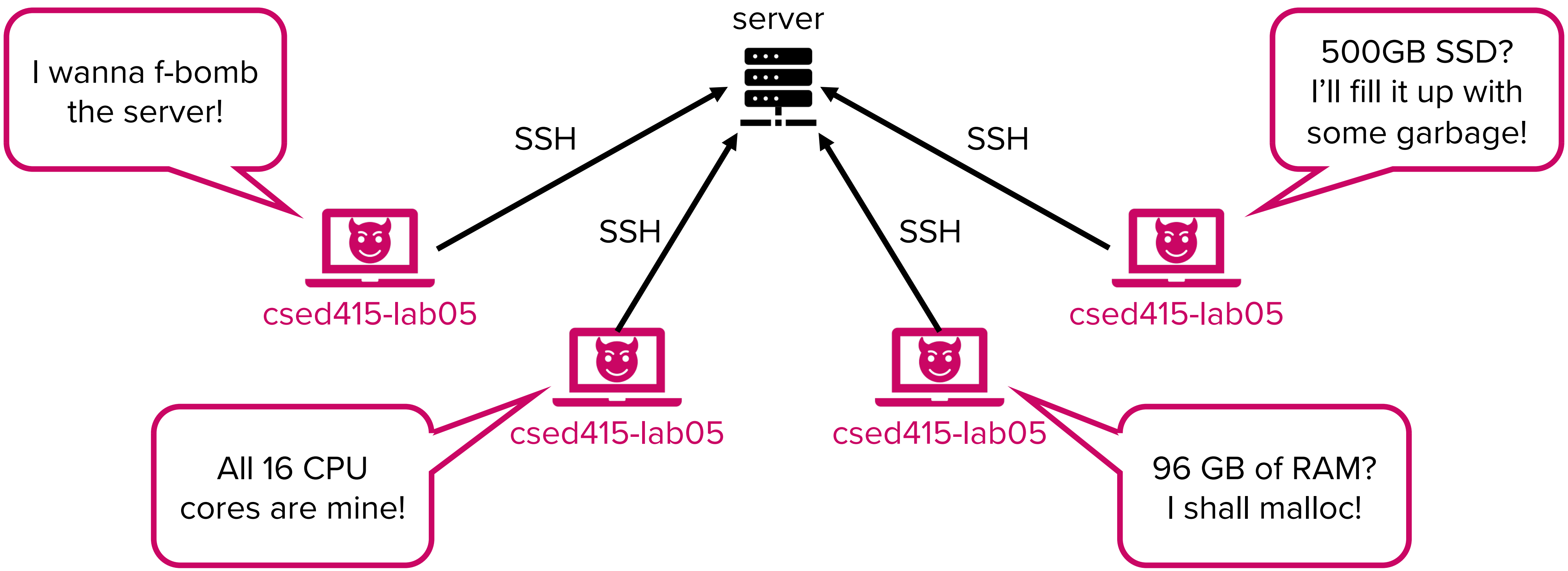
CSED415 lab server

- Single-server, single-user



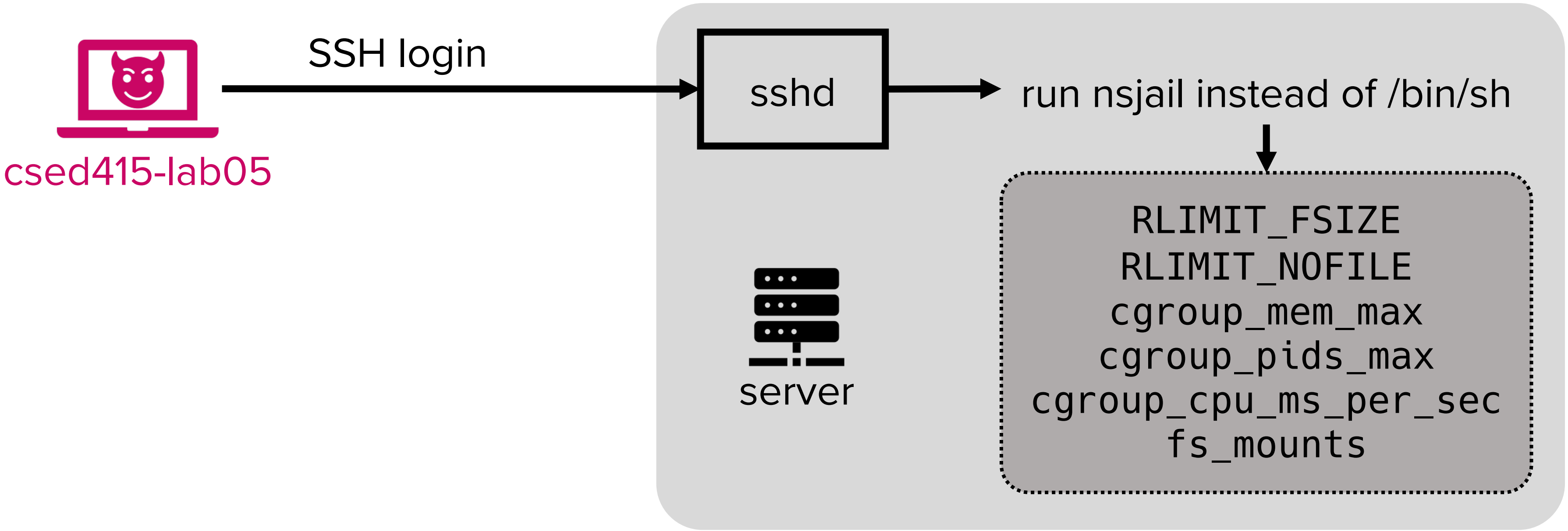
CSED415 lab server

- Single-server, single-user, malicious students are behind!



CSED415 lab server

- Our mitigation strategy:
  - `rlimit` and control groups (`cgroup`) inside a sandbox (`nsjail`)



CSED415 lab server

- Try:
  - `/* fbomb.c */ while (1) { fork(); }`
  - `/* mbomb.c */ while (1) { malloc(0x10000000); }`
  - `$ fallocate -l 100G file`
  - `$ sha1sum /dev/zero | sha1sum /dev/zero | sha1sum /dev/zero | sha1sum /dev/zero | sha1sum /dev/zero | sha1sum /dev/zero`

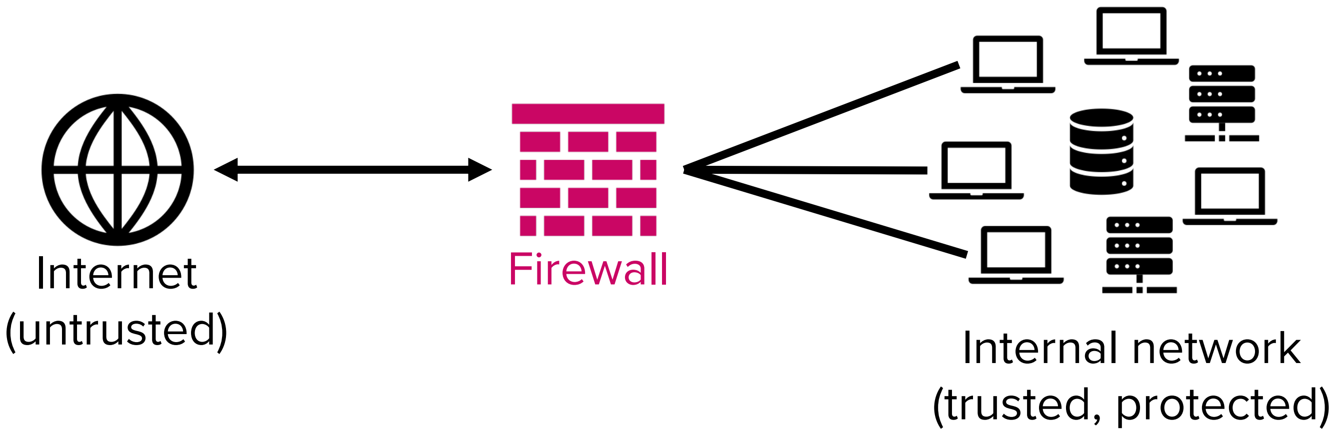
Firewalls

Motivation: Scalable defenses

- Goal: Protect an entire network rather than individual machines against external attacks
  - e.g., a company network with many servers and desktops
- Attack surface is very large
  - More network services == more risk
    - ssh, ftp, http, printer, …
  - More network-connected machines == more risk
    - Different environments (OS, software, …) where some may be vulnerable
    - Different user behaviors (phishing, social engineering, …)

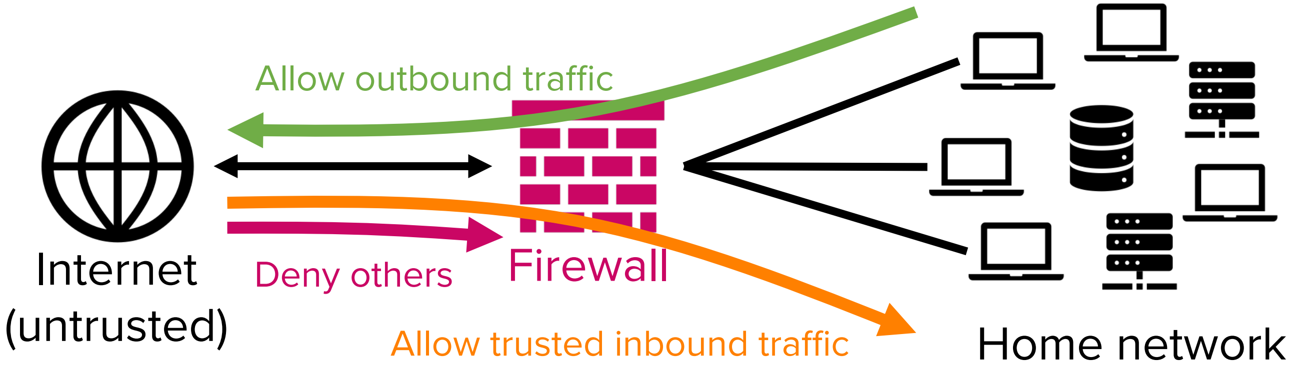
Firewalls and security policies

- Firewall idea: Interpose a single point of access in and out of the network with a security monitor
  - Any traffic must pass through the firewall
- Security policies control network access
  - What traffic is allowed to enter the network (inbound policy)
  - What traffic is allowed to exit the network (outbound policy)



Firewalls and security policies

- Security policies for a standard home network:
  - Outbound: Allow all
    - Users inside the home network can connect to any external service
  - Inbound: Allow some
    - Allow inbound traffic in response to an outbound connection
    - Allow inbound traffic to certain trusted services (e.g., SSH)
    - Deny all other inbound traffic



Firewalls and security policies

- Possible default security policies
  - Default-allow: Allow all traffic, but deny blacklisted traffic
    - When problems arise, add them to the blacklist
  - Default-deny: Deny all traffic, but allow whitelisted traffic
    - When need be, e.g., when users complain, add them to the whitelist
- Which policy is better?
  - Default-allow is more flexible, but flaws are catastrophic
  - Default-deny is more conservative, but less prone to flaws
  - Considering the “fail-safe defaults” principle, default-deny is generally deemed more secure

Two types of firewalls

- Stateless firewall
- Stateful firewall

Stateless firewall

- Implemented as stateless packet filters
- Inspect each incoming and outgoing network packet
  - All decision is made using only the information in the packet itself
    - Source IP, destination IP, source port, destination port, and protocol
  - Do not consider communication history
- Example:
  - No inbound connection to low ports except
    - 20, 21 for FTP / 23 for Telnet / 25 for SMTP / 80 for HTTP

Stateless firewall

- Stateless packet filtering examples
  - Allow inbound from some ports
    - Allow port 20, 21 for FTP (if running file server)
    - Allow port 22 for SSH (if running secure shell server)
    - Allow 23 for Telnet
    - Allow 25 for SMTP (if running mail server)
    - Allow 80 and 443 for HTTP, HTTPS (if running web server)
  - Allow inbound from src IP 8.8.8.8
    - Google's DNS
  - Deny all other inbound IP and ports

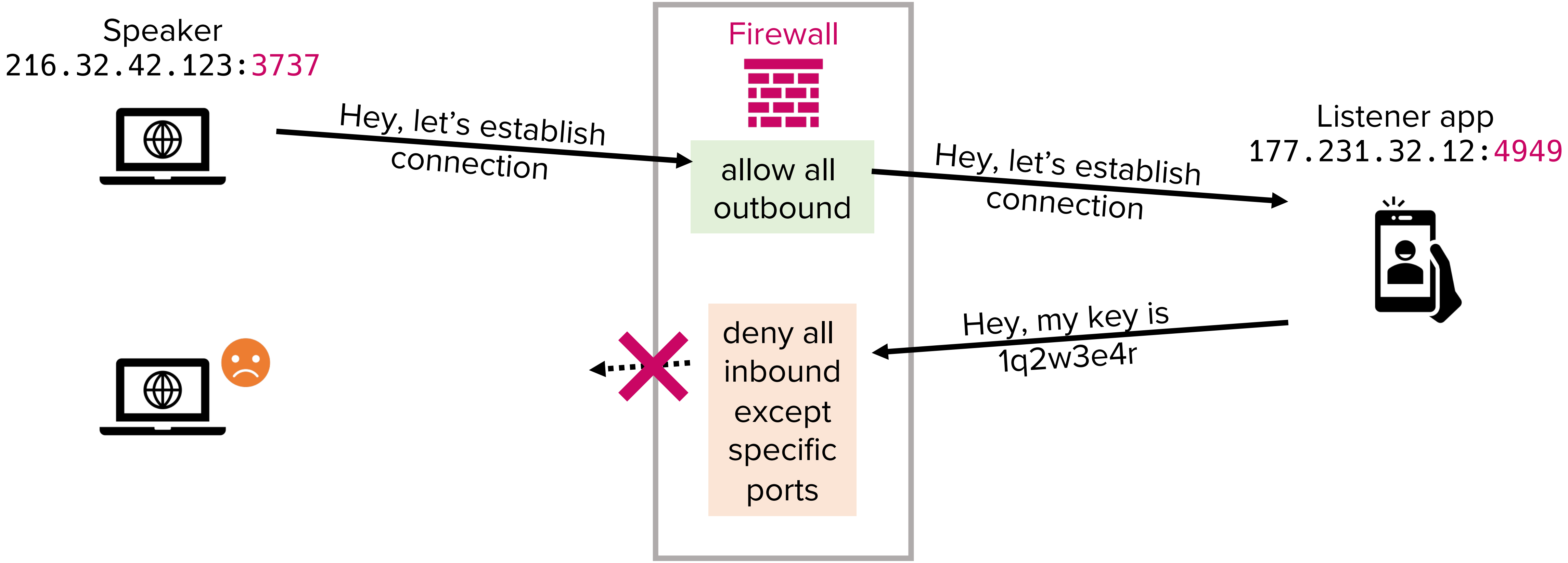
Stateless firewall

- Applying simple packet filter to the home network example
  - Allow all outbound ☑ Allow all
  - Allow inbound traffic to certain trusted services (e.g., SSH) ☑ Using portnum
  - Allow inbound traffic in response to an outbound connection
  - Deny all other inbound traffic ☑ Deny all

Issue: How do we know if inbound traffic is in response to an outbound connection without keeping track of states?

Stateless firewall

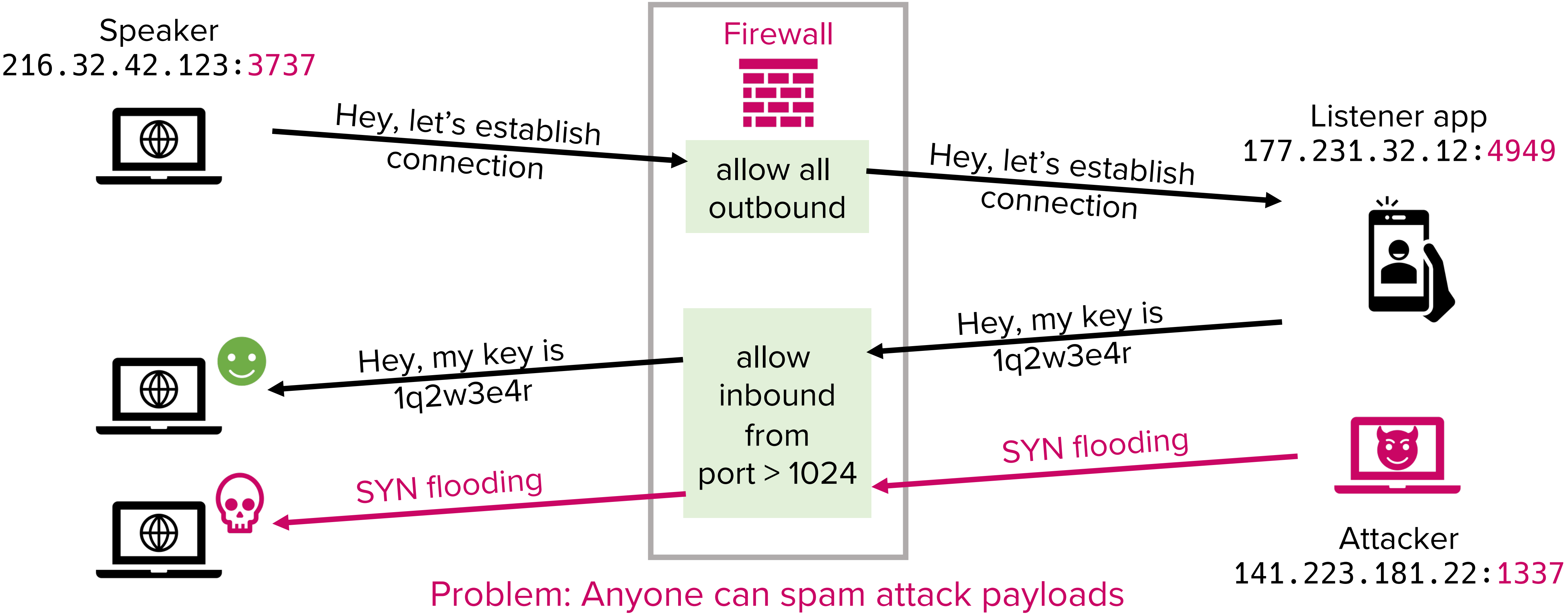
- Example: Temporary P2P connection with strict policy



Need to add rules, e.g., allow inbound from ports > 1024

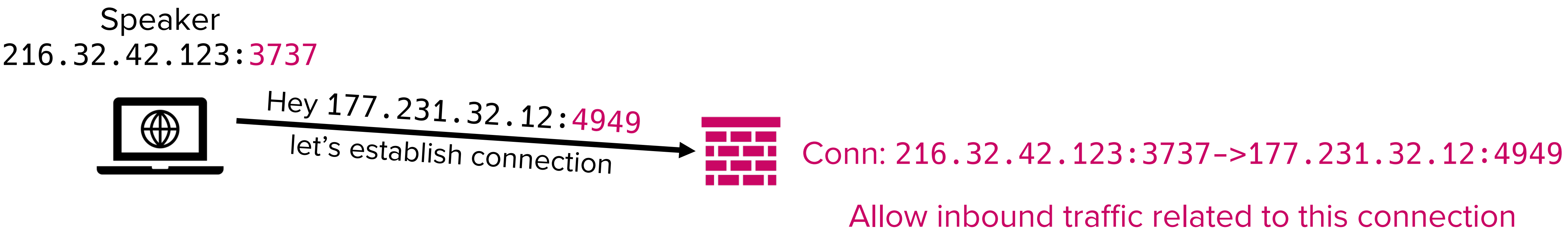
Stateless firewall

- Example: Temporary P2P connection with relieved policy



Stateful firewall

- Solution: Store and use full context (state) of connections
  - Create a directory of outbound connections
  - Allow packets from the destination of the recorded outbound connections
  - Use connection information along with strict stateless rules



Stateless vs Stateful firewalls

- Stateless
  - Fast and scalable to large volumes of traffic
  - Cheaper than stateful firewalls
  - Less secure
- Stateful
  - More secure and versatile
    - Don’t need to allow a range of ports
    - Dynamically configured
  - Slower and more expensive than stateless firewalls

Firewall pros and cons

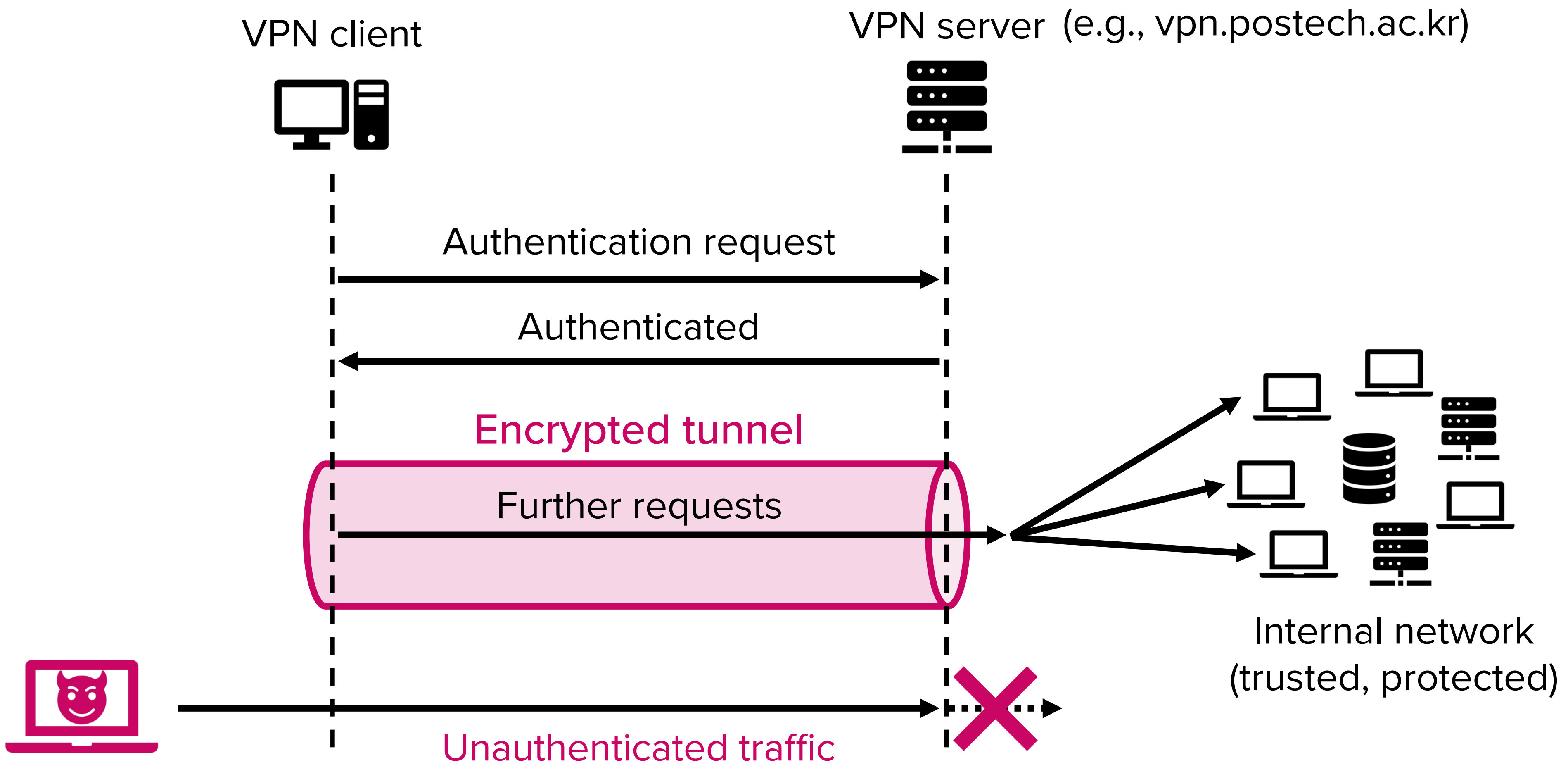
- Pros
  - Centralized management of security policies (single point of control)
  - Transparent operation to end users
  - Mitigates security vulnerabilities
- Cons
  - Reduced network connectivity
  - Vulnerable to insider attacks
    - Employees can be bribed or threatened
    - Untrusted devices are brought into the network (e.g., employee laptops)

Alternative for firewalls

- Virtual private network (VPN)
  - A set of protocols that allows direct access to an internal network from external entities
  - VPN server authenticates user (VPN client)
  - Provide encrypted VPN tunnel
    - Only authenticated clients can send traffic via the tunnel
    - VPN server routes the traffic to individual services

VPN for perimeter security

- VPN workflow



Summary

- Availability: Making sure users are able to use a service
- DoS: Attack availability of services
  - Application-level DoS
  - Network-level DoS
  - Stateful DoS
  - Overprovisioning can mitigate DoS but it is very expensive
- Firewalls: Perimeter security to defend a network
  - Utilize stateless and/or stateful packet filters

Questions?