11/9/26

COMP 435: *Computer Security Concepts*

Lecture 30: Intro to Network Security



The University of North Carolina at Chapel Hill

https://tinyurl.com/comp435-fa26

Plan for Class Today

- 10:10-10:25 – Finish Basic SW Security Best Practices
- 10:25 -11:00 – Intro to Network Security

Guiding Question ☺

How do we secure our communication and data over a network?

Logistics Update



- Written project checkpoint due November 16
  - Worth 15% of the project grade.
- Lab 4 is out
  - Due 11/18



Software Security: Best Practices

Guidelines for Writing Secure Code



- Check bounds
- Validate and sanitize Input
- Use safe utilities
- Grant least privilege
- Sandbox
- Use type safe languages

Grant Least Privilege



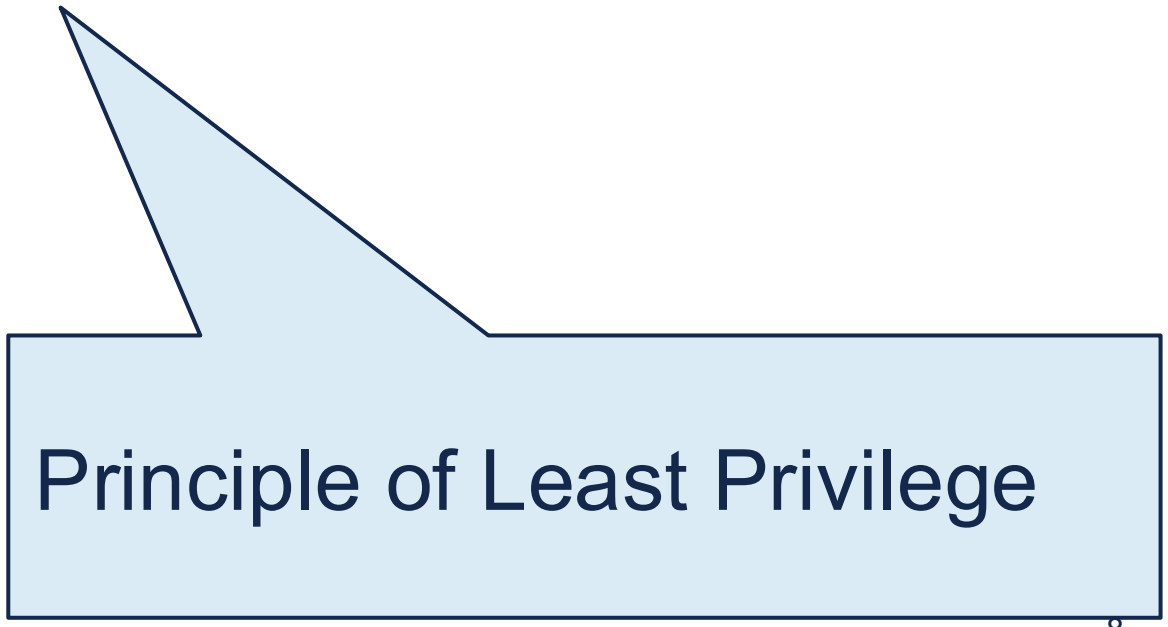
Def’n: a subject should be granted only the privileges necessary to complete its work

E.g., Your browser does not need admin privileges

Grant Least Privilege

Def'n: a subject should be granted only the privileges necessary to complete its work

E.g., Your browser does not need admin privileges



Grant Least Privilege



- Give code only the access required to complete the work

E.g., a calculator does not need access to the filesystem

Grant Least Privilege



- Build modular code

E.g., separate HTML rendering from filesystem access in your browser

Principle of Least Privilege



Limits the damage that occurs when security is compromised

Sandboxing



Def’n: allow code access to only a portion of the filesystem

E.g., The browser can access only files in `/tmp`

Type Safety



Def'n: Memory is only ever accessed according to its type

E.g.,

- Memory allocated as an int is only ever used as an int
- Indexing into an array can only ever access the memory allocated as part of the array

Memory Safety



Def'n: A program is memory safe if it never accesses undefined memory

- buffer overflows
- null pointer dereference
- use-after-free
- use of uninitialized memory
- illegal free

violate memory safety

Memory Safety



Def'n: A program is memory safe if it never accesses undefined memory

- buffer overflows
- null pointer dereference
- use-after-free
- use of uninitialized memory
- illegal free

violate memory safety

Type safe languages produce memory safe programs



Heartbleed



logo: https://en.wikipedia.org/wiki/Heartbleed

Terminology

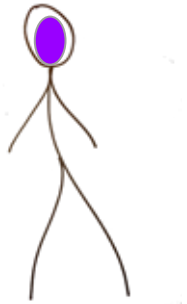


- Buffer overflow
- Buffer overrun
- Buffer overflow attack
- Stack smashing



reading past the end of an allocated array

TLS Heartbeat Protocol



TLS (Transport Layer Security): a cryptographic protocol that secures communication over a network



Heartbeat request: 3 bytes, “cat”

Heartbeat reply: “cat”

liveness

https://xkcd.com/1354/

Heartbleed Attack





Heartbeat request: 64 kbytes, “cat”

Heartbeat reply: “cat0xcafep2ssw0rd!”

https://xkcd.com/1354/

History of Heartbleed



- Bug introduced to code repository Dec 2011
- Request For Comment (RFC) published Feb 2012
- Included and enabled by default in OpenSSL March 2012
- First discovered March 2014

500,000 servers

→ 17% of web servers

OpenSSL is the library implementing TLS

Lec 28 WSQ2

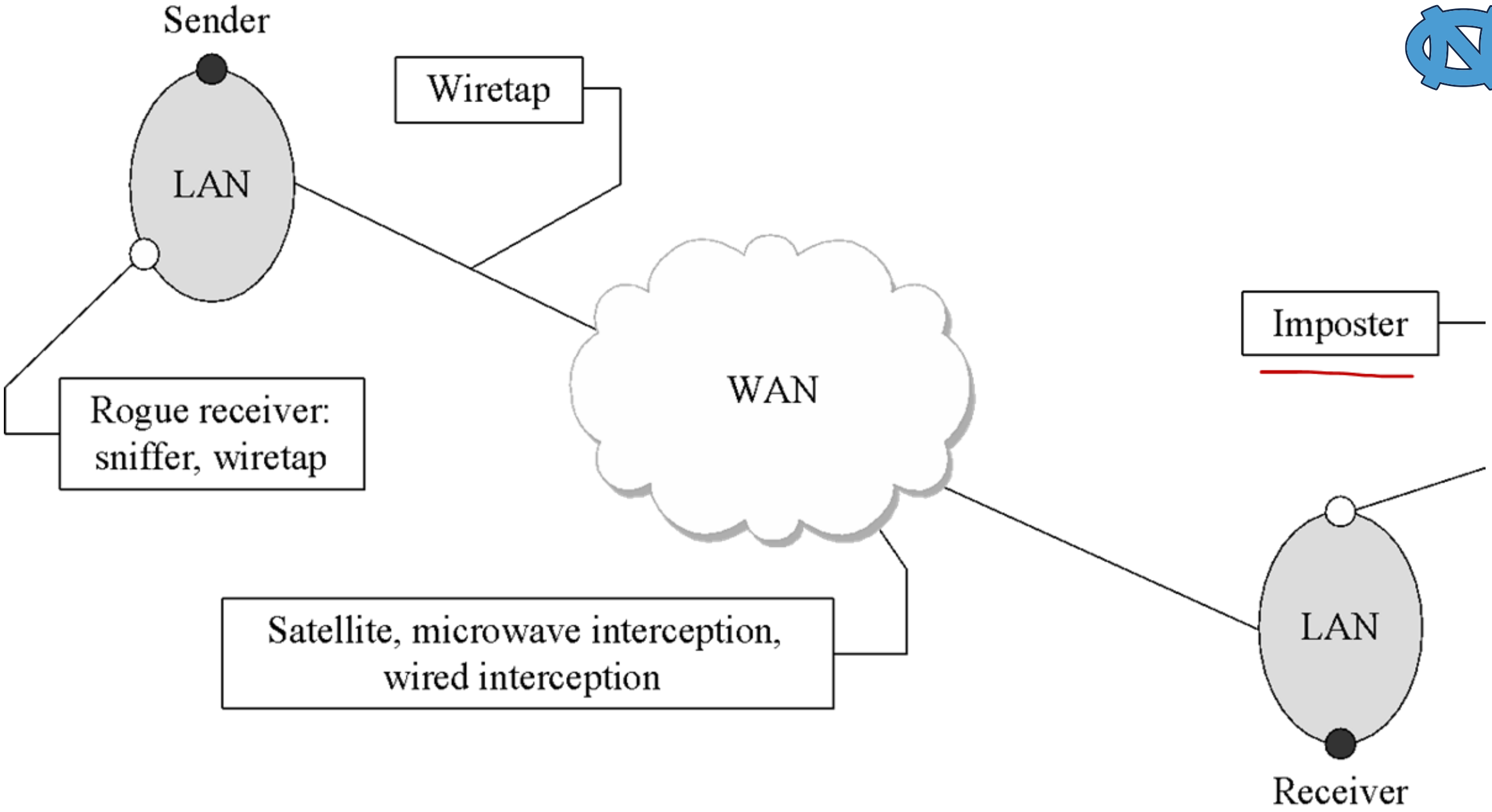




Network Security Concepts

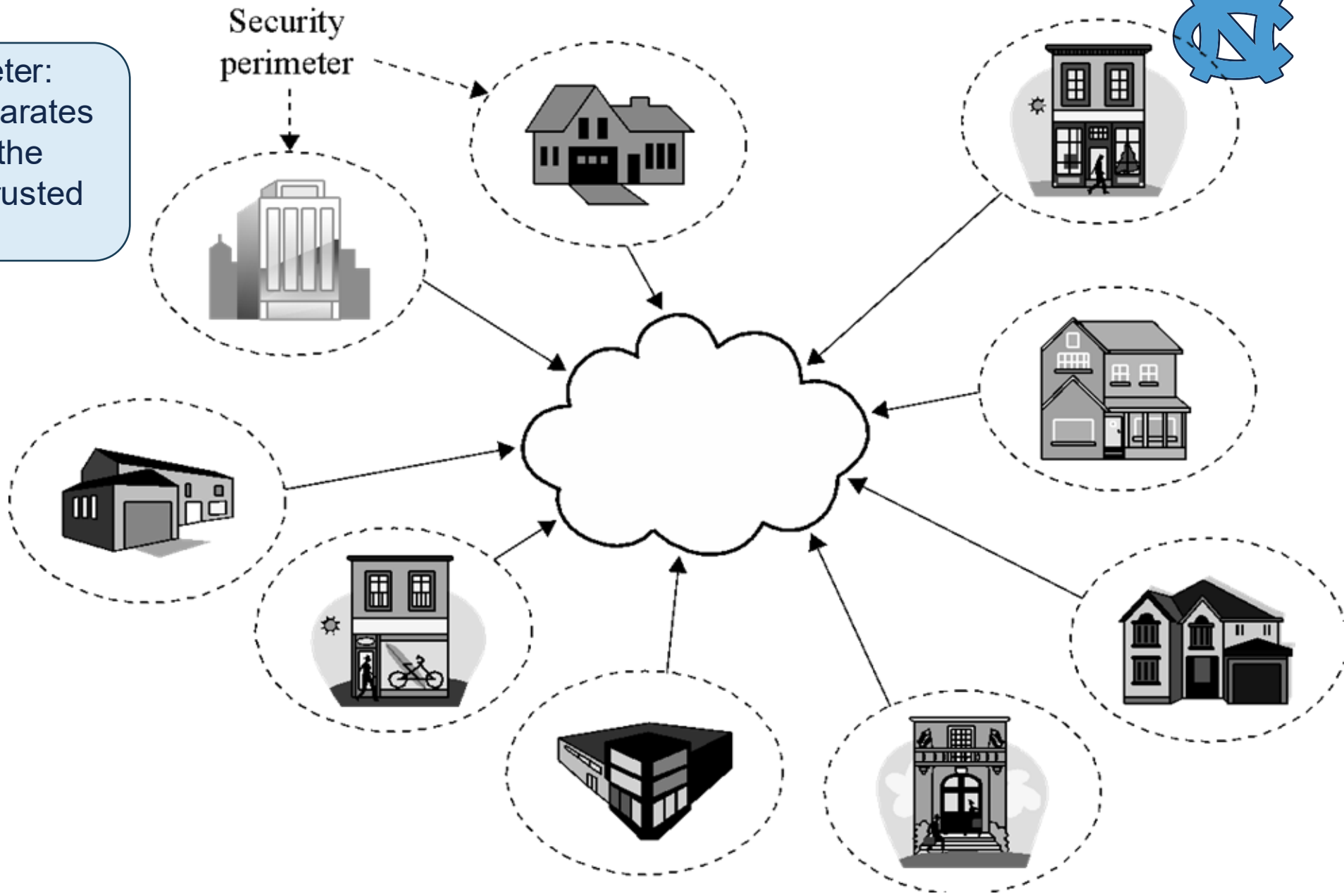
Challenges

- Anonymity
- Many points of attack
- Sharing
- Complexity
- Unknown perimeter
- Unknown path



From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

Security perimeter: boundary that separates trusted part of the network from untrusted areas



From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

Medium	Strengths	Weaknesses
Wire	• Cheap • Ubiquitous	• Signal emanation • Physical wiretapping
Optical Fiber	• No emanation • No wiretapping	• Weak at connection points
Microwave	• Strong signal	• Interception possible • Line of sight needed • Needs repeaters
Wireless	• Ubiquitous	• Interception possible • Short range
Satellite	• Strong signal	• Delay (long distance) • Interception possible

From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

Threats



- Interception
- Modification
- Fabrication
- Interruption

Threats



- ***Interception***  confidentiality
- Modification
- Fabrication
- Interruption

Threats

- Interception
- ***Modification***  integrity
- ***Fabrication*** 
- Interruption

Threats

- Interception
- Modification
- Fabrication
- ***Interruption***  availability



Dolev-Yao Model

Dolev-Yao Model



- IEEE Transactions on Information Theory, 1983
- Active attacker
  - Is a legitimate user of the network
  - Can obtain any message on the network
  - Can be a receiver to any user

Dolev-Yao Model

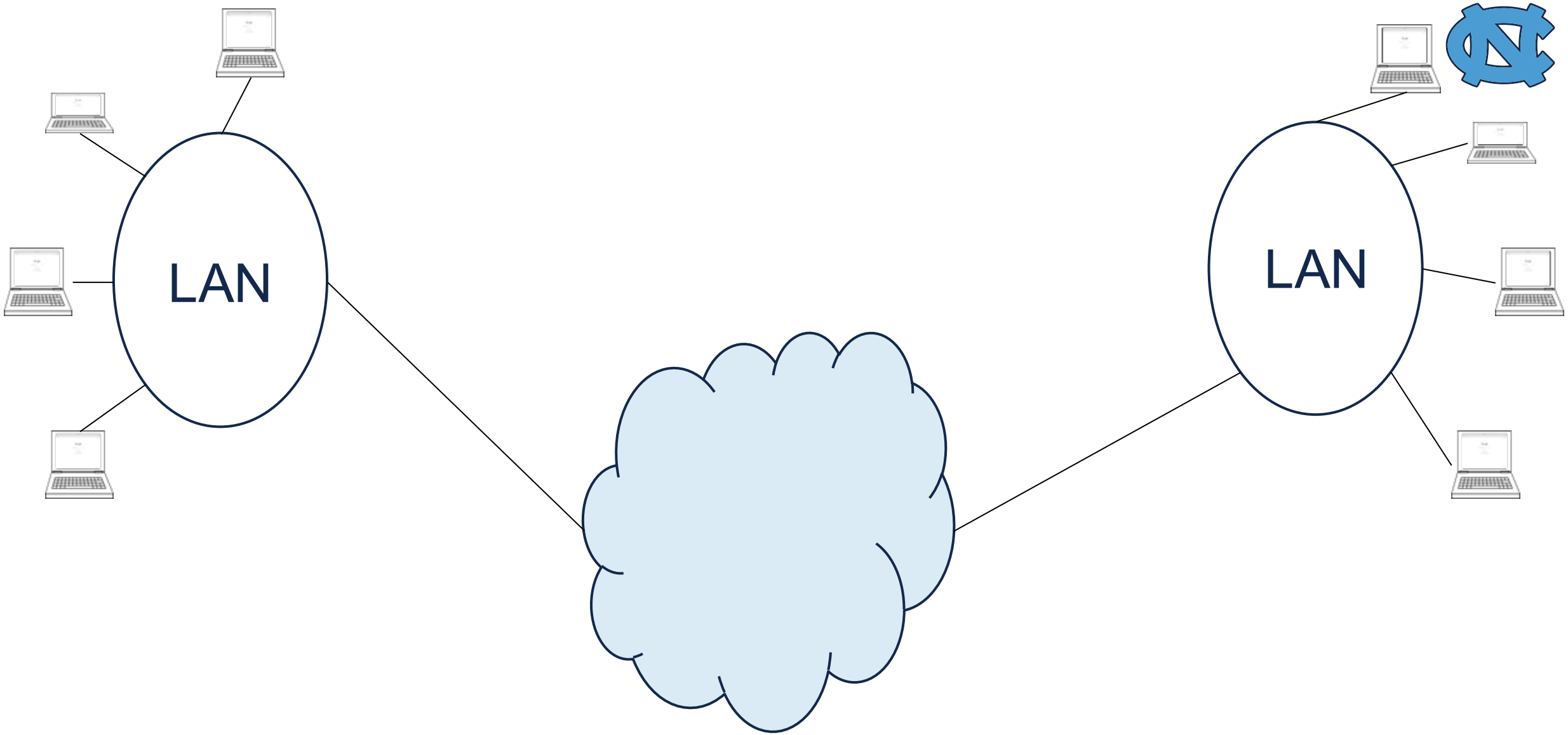


- IEEE Transactions on Information Theory, 1983
- Active attacker
  - Is a legitimate user of the network
  - Can obtain any message on the network
  - Can be a receiver to any user

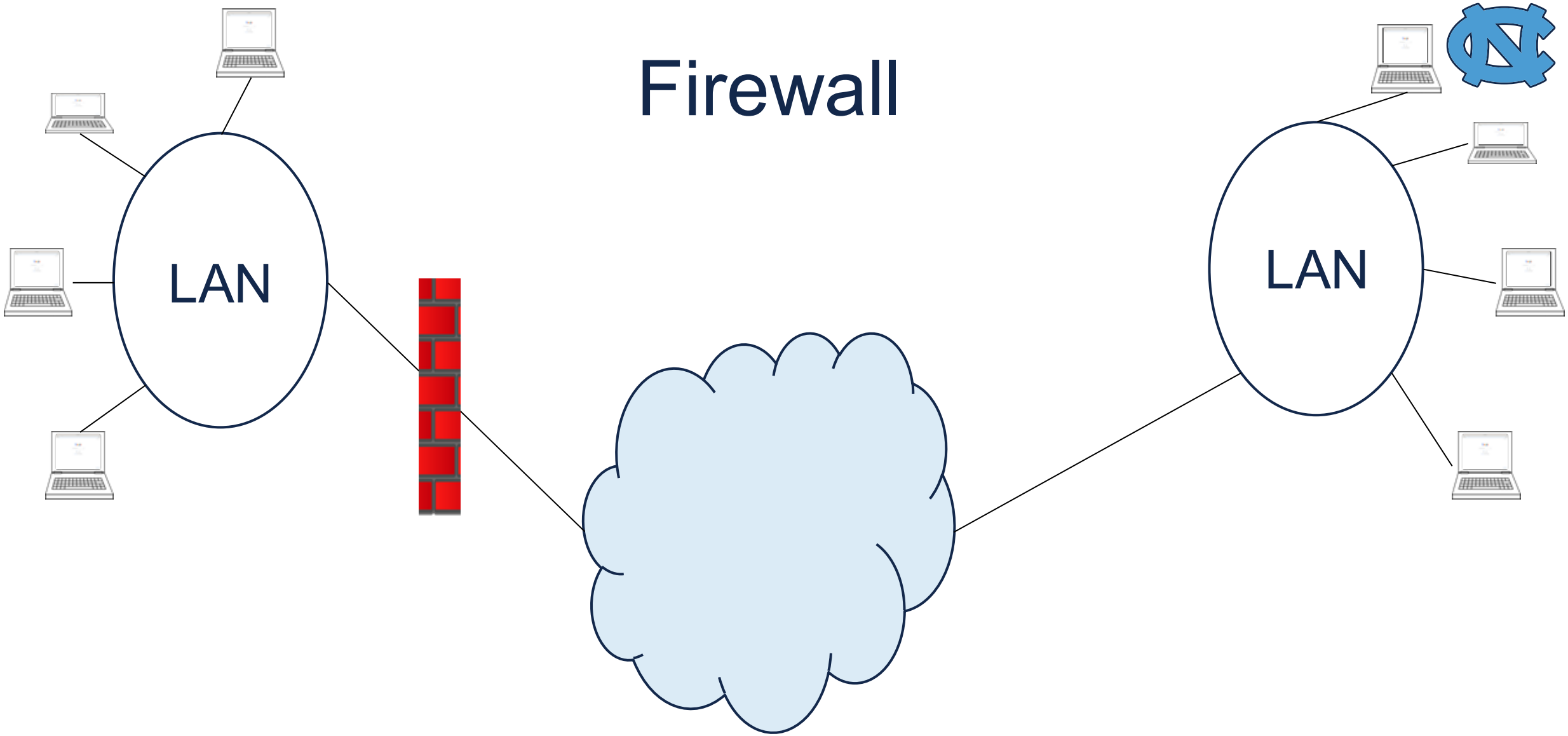
The attacker carries the message



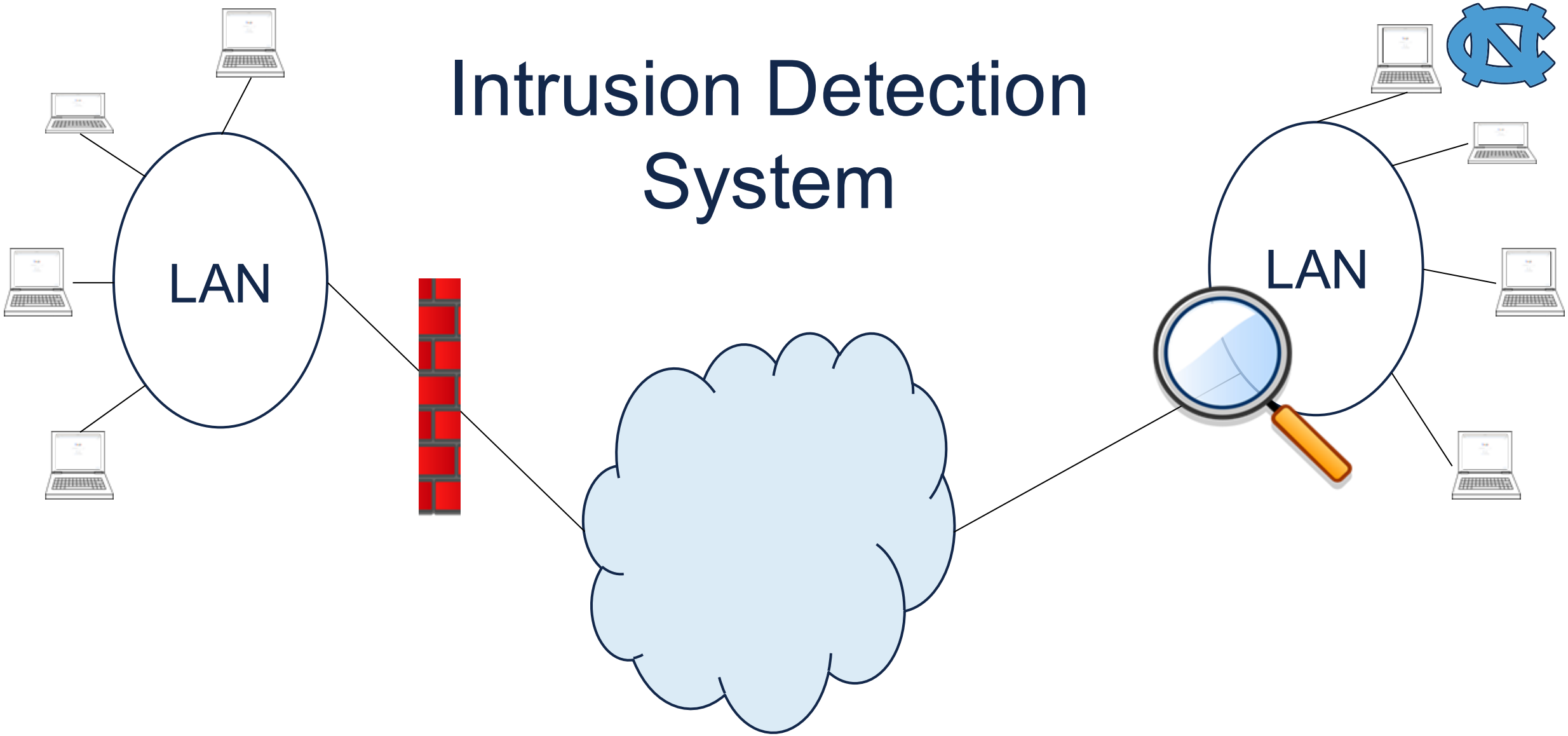
Defense Strategies


LAN
LAN

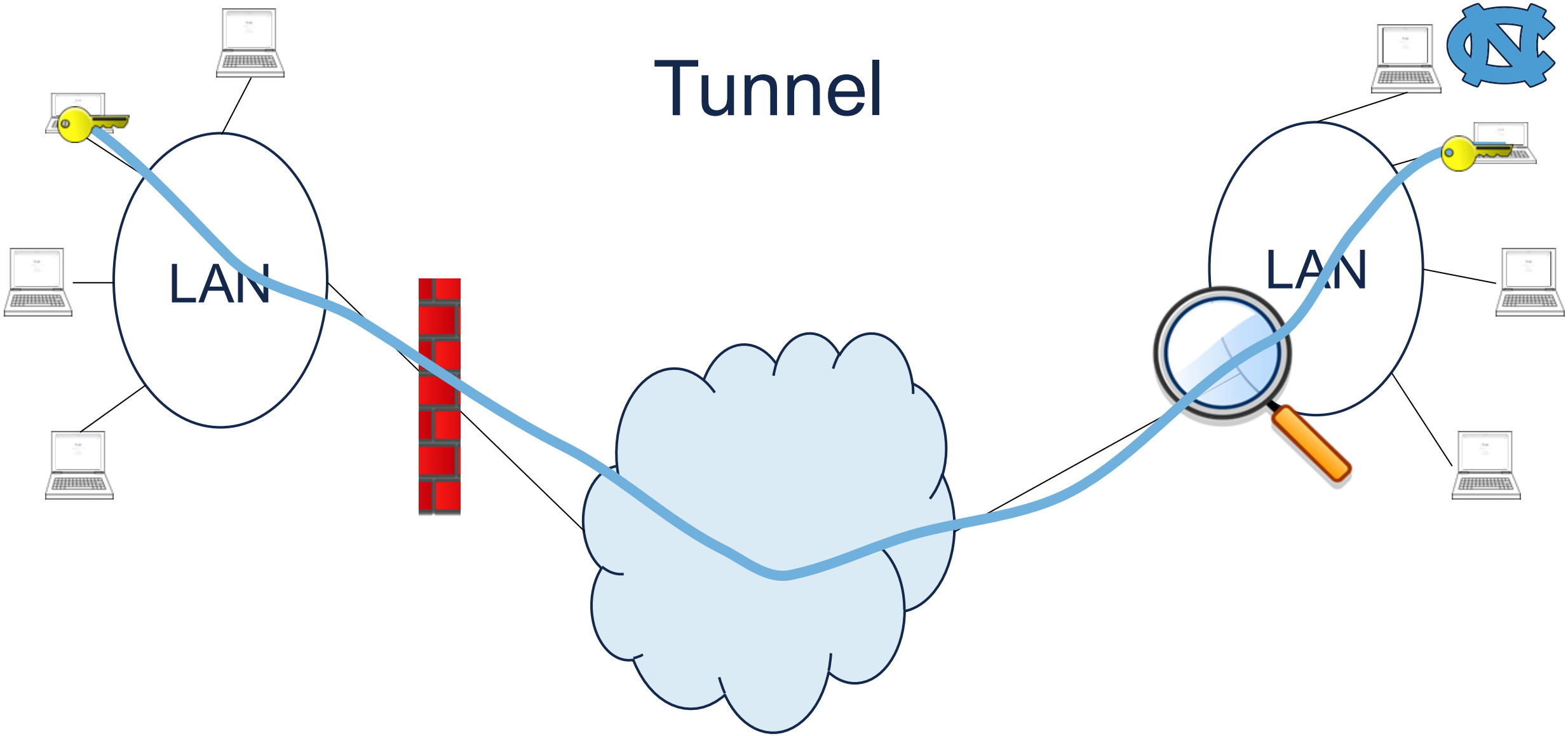
Firewall



Intrusion Detection System



Tunnel



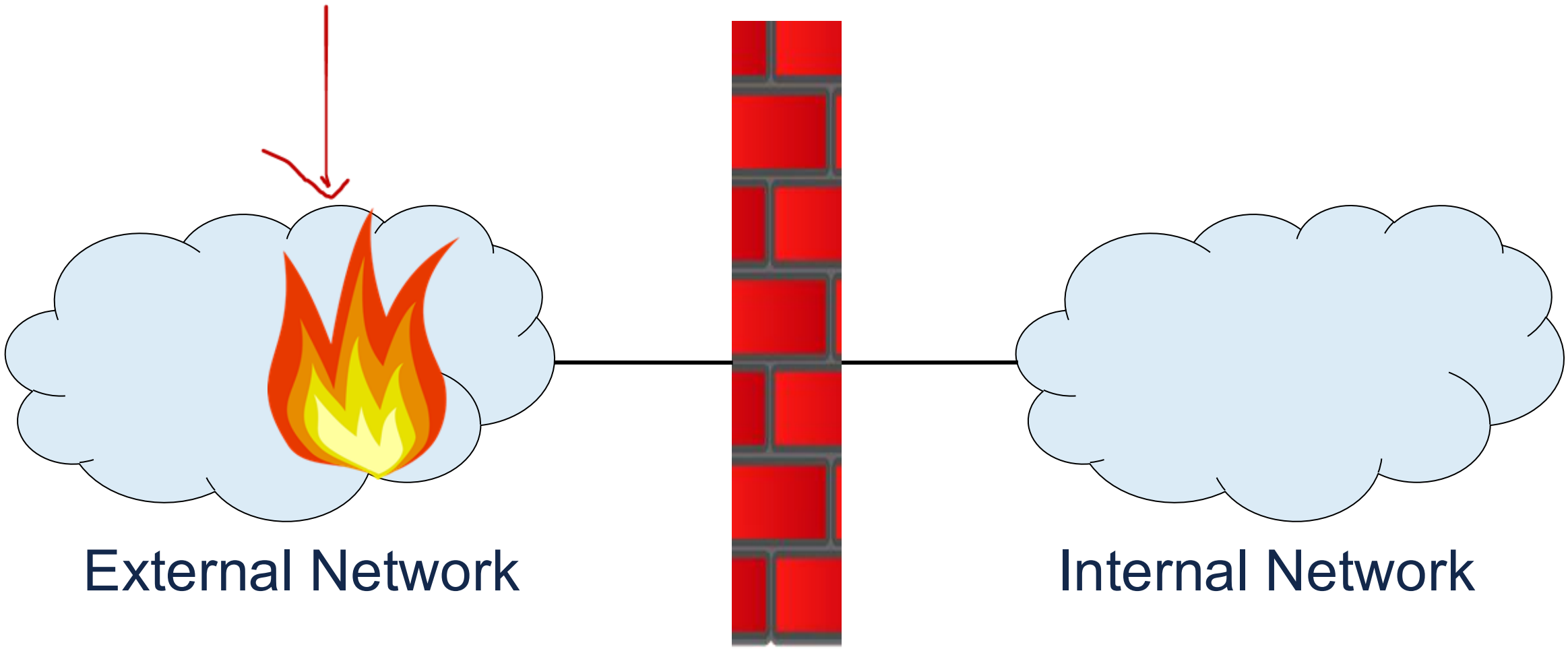


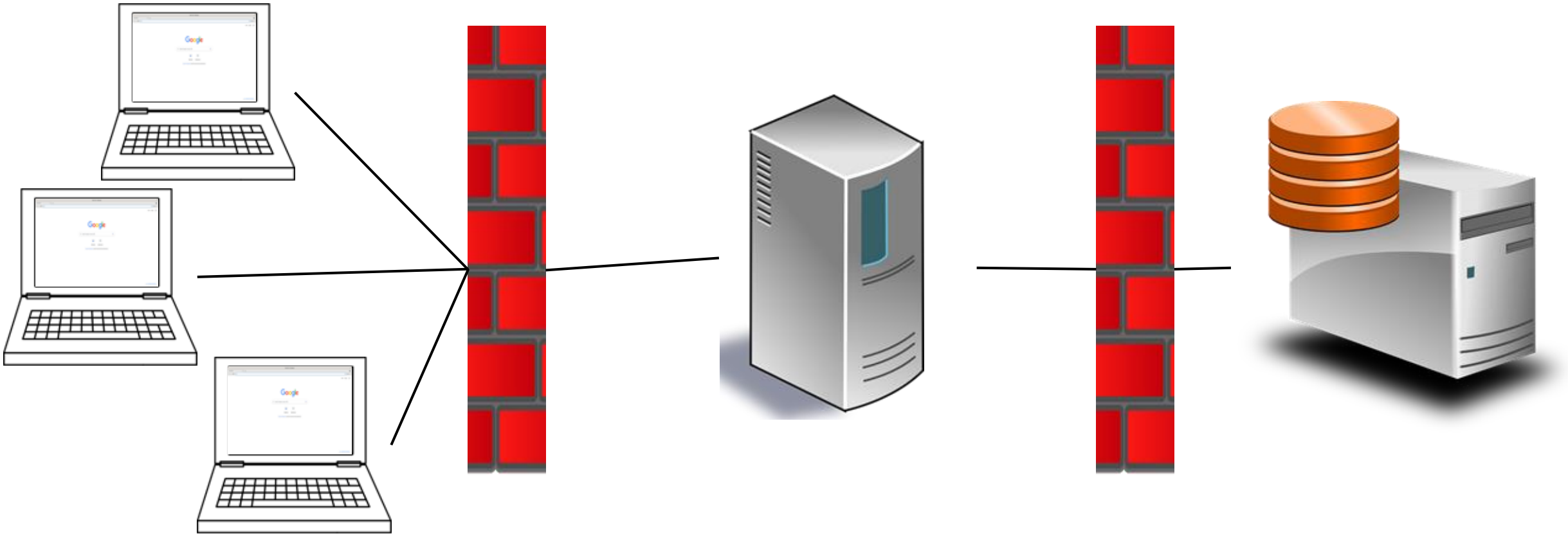
Firewalls








External Network
Internal Network



Internet

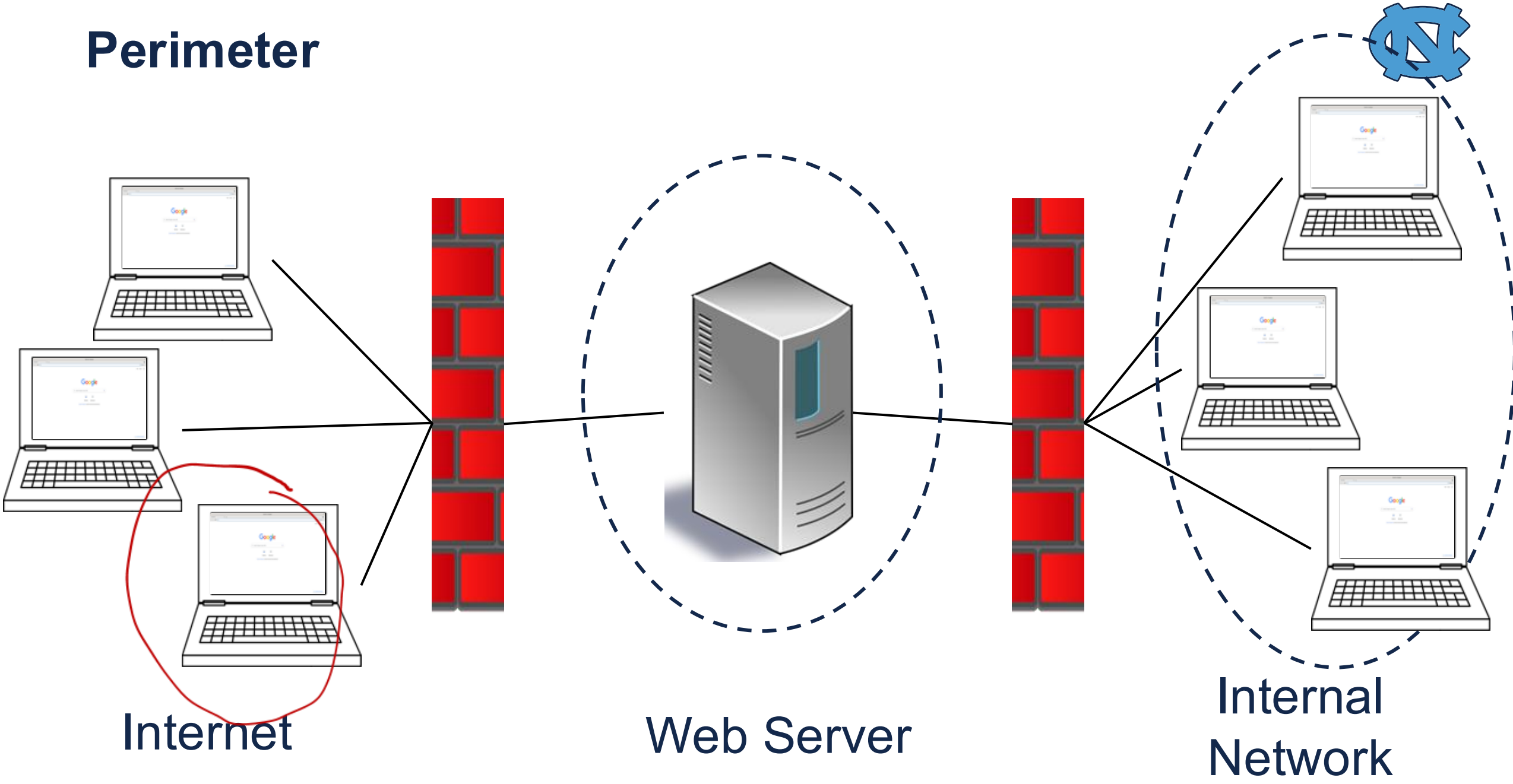
Web Server

Database Server

Firewall

Def’n: a gateway between two networks that can allow, deny, or modify data passing from one to the other

Perimeter



Firewall Properties



- security perimeter
- defense in depth: *used in conjunction w/ other methods*
- chokepoint for security analysis: *all communication funnels through*
- reference monitor
  - complete mediation
  - immune to attack
  - proven correct

Types of Firewalls



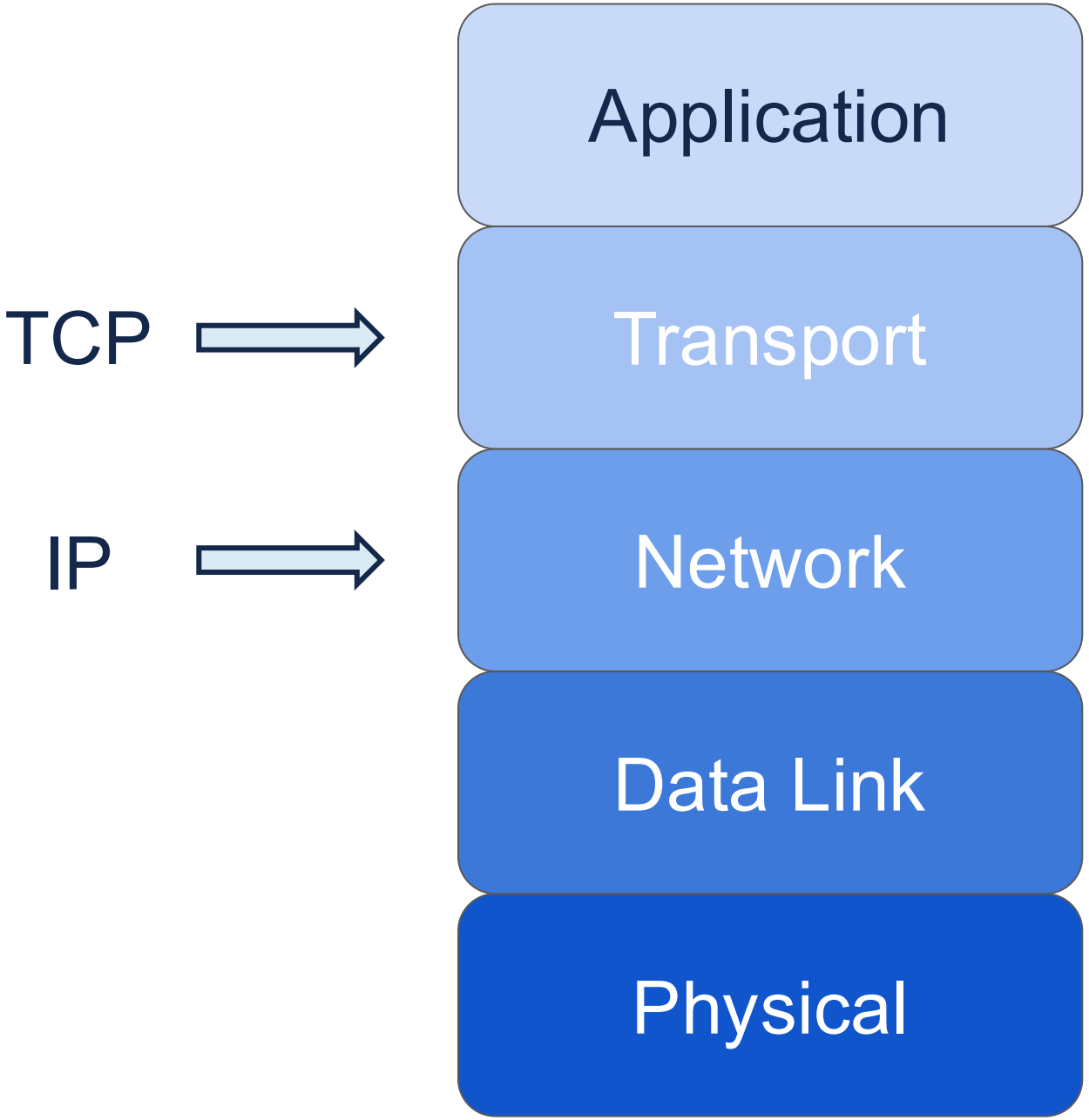
Packet Filters

Proxy



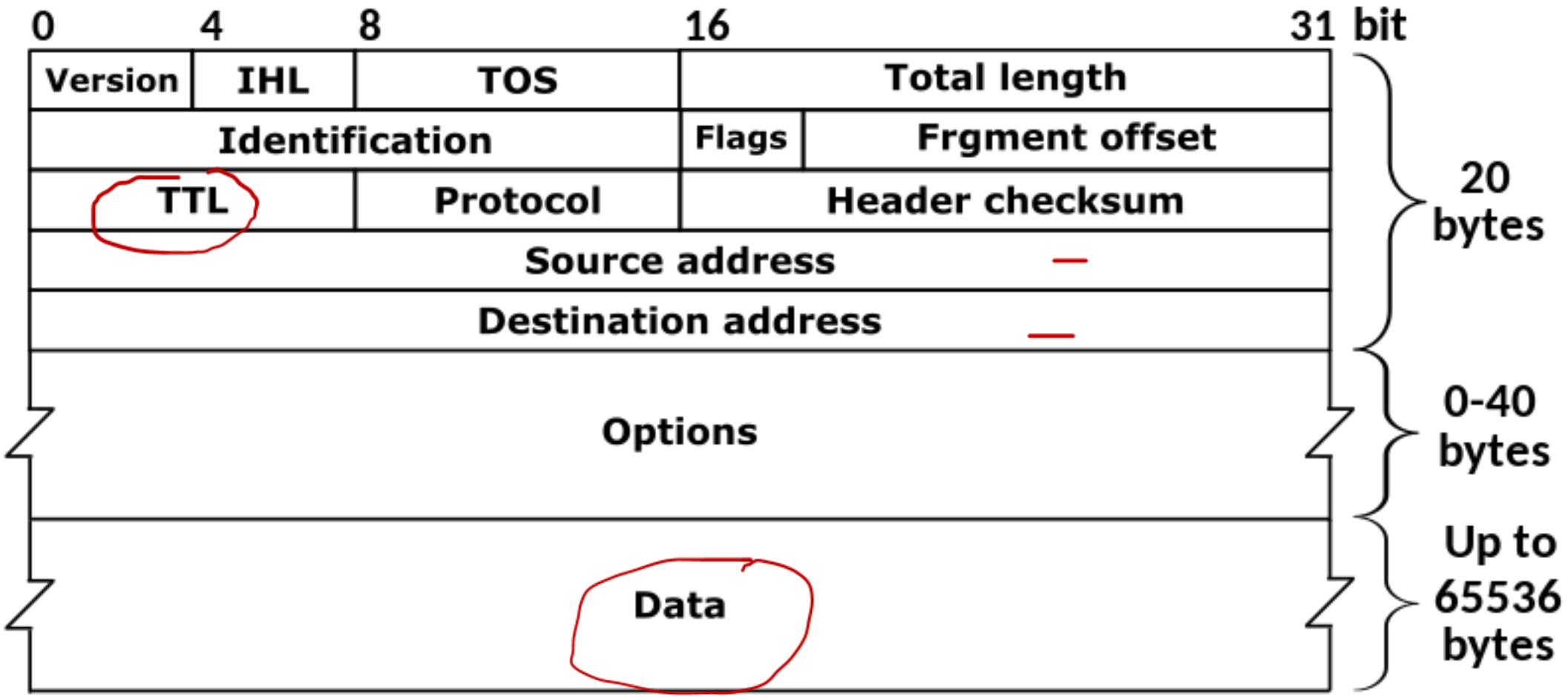
Some Background: TCP/IP

TCP/IP



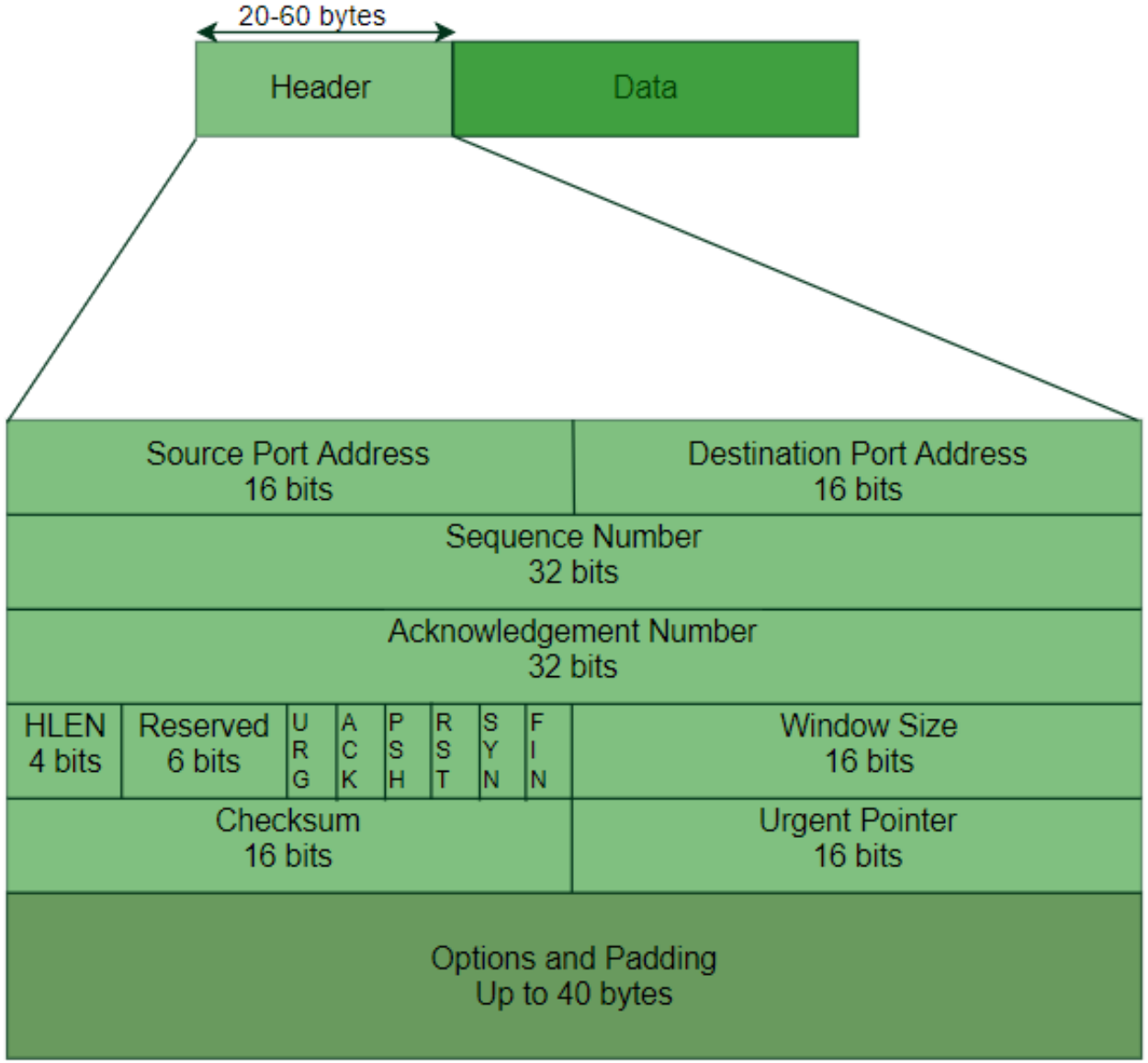
IPv4 Packet

Time to Live



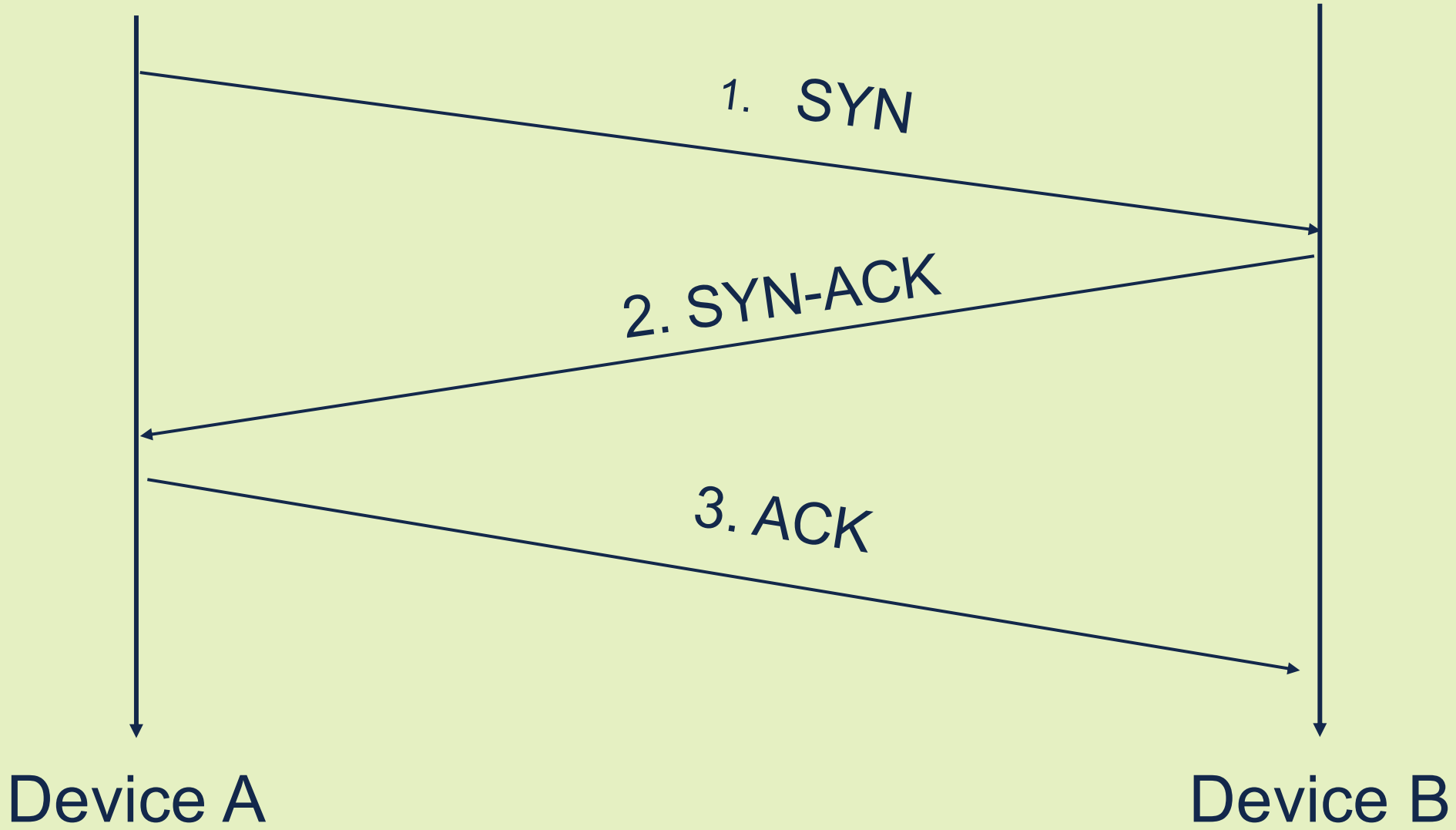
https://en.wikipedia.org/wiki/File:IPv4_Packet-en.svg

TCP Segment



https://www.geeksforgeeks.org/services-and-segment-structure-in-tcp/

TCP





Packet filters

Packet Filter



Def’n: IP packets are compared to a set of rules and either allowed, dropped, or rejected

Filtering Rules



- TCP/IP header fields
- IP header protocol
- Packet size, flags
- Application protocol
- User ID
- Network activity

Ways to Categorize Packet Filters



- Positive vs. Negative
  - Ex: Default Deny vs Default Allow
- Stateless vs Stateful
- Shallow View vs. Deep View

Firewall Limitations



- reliance on topology
- untrustworthy insiders
- tunneling
- encrypted content

An Example Packet Filter



Rule	Direction	Src Addr	Dest Addr	Protocol	Dest Port	Action
A	In	External	Internal	TCP	25	Allow
B	Out	Internal	External	TCP	>1023	Allow
C	Out	Internal	External	TCP	25	Allow
D	In	External	Internal	TCP	>1023	Allow
E	Either	Any	Any	Any	Any	Deny

Building Internet Firewalls: Internet and Web Security, by Elizabeth D. Zwicky, Simon Cooper, D. Brent Chapman, ch. 8.

An Example Packet Filter

People inside the organization can **receive** mail!

Rule	Direction	Src Addr	Dest Addr	Protocol	Dest Port	Action
A	In	External	Internal	TCP	25	Allow
B	Out	Internal	External	TCP	>1023	Allow
C	Out	Internal	External	TCP	25	Allow
D	In	External	Internal	TCP	>1023	Allow
E	Either	Any	Any	Any	Any	Deny

Building Internet Firewalls: Internet and Web Security, by Elizabeth D. Zwicky, Simon Cooper, D. Brent Chapman, ch. 8.

An Example Packet Filter

People inside the organization can **send** mail!

Rule	Direction	Src Addr	Dest Addr	Protocol	Dest Port	Action
A	In	External	Internal	TCP	25	Allow
B	Out	Internal	External	TCP	>1023	Allow
C	Out	Internal	External	TCP	25	Allow
D	In	External	Internal	TCP	>1023	Allow
E	Either	Any	Any	Any	Any	Deny

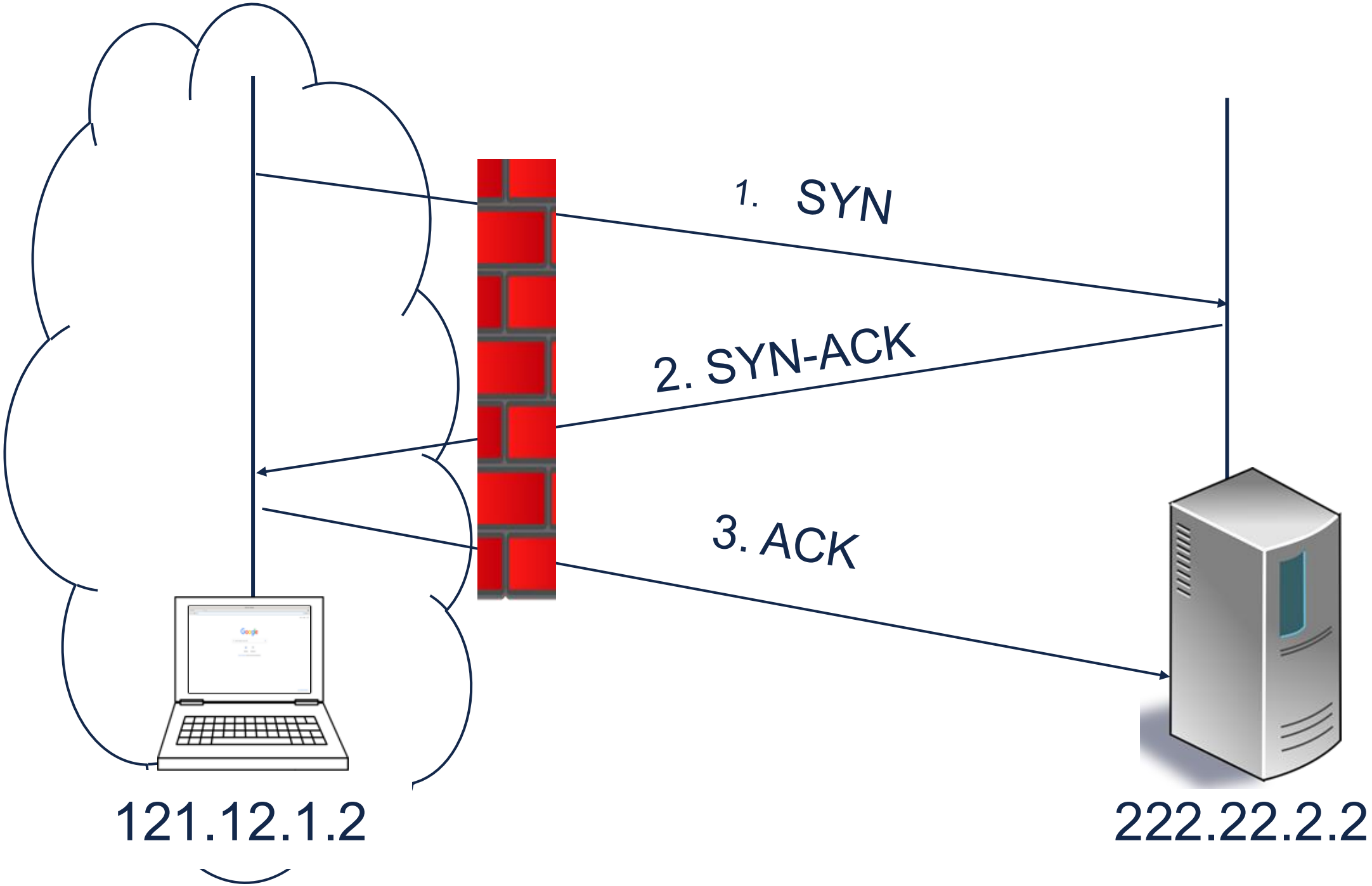
Building Internet Firewalls: Internet and Web Security, by Elizabeth D. Zwicky, Simon Cooper, D. Brent Chapman, ch. 8.

An Example Packet Filter

Default deny!

Rule	Direction	Src Addr	Dest Addr	Protocol	Dest Port	Action
A	In	External	Internal	TCP	25	Allow
B	Out	Internal	External	TCP	>1023	Allow
C	Out	Internal	External	TCP	25	Allow
D	In	External	Internal	TCP	>1023	Allow
E	Either	Any	Any	Any	Any	Deny

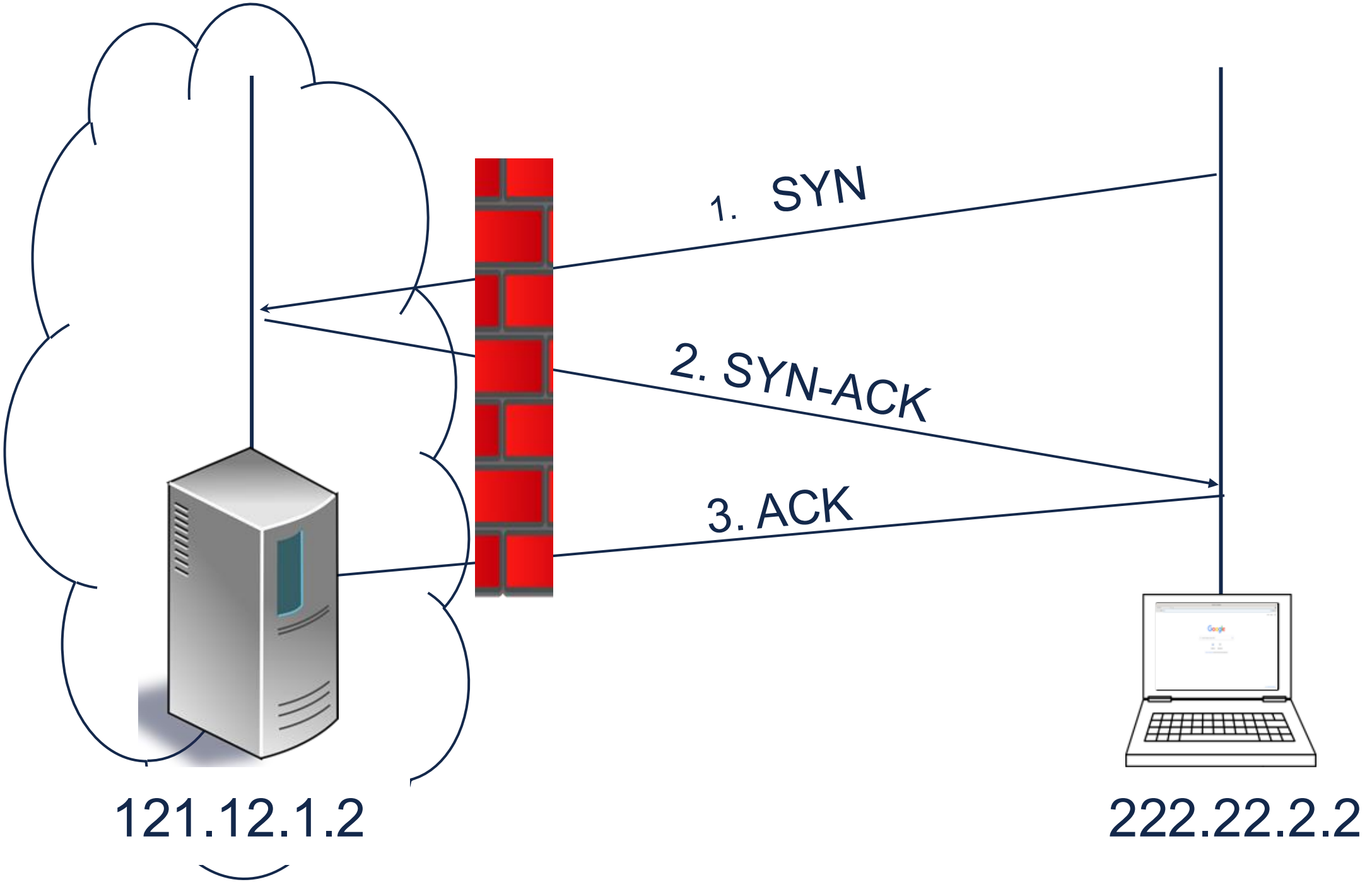
Building Internet Firewalls: Internet and Web Security, by Elizabeth D. Zwicky, Simon Cooper, D. Brent Chapman, ch. 8.


1. SYN
2. SYN-ACK
3. ACK
121.12.1.2
222.22.2.2

Rule	Dir	Src Addr	Dest Addr	Protocol	Dest Port	Action
A	In	External	Internal	TCP	25	Allow
B	Out	Internal	External	TCP	>1023	Allow
C	Out	Internal	External	TCP	25	Allow
D	In	External	Internal	TCP	>1023	Allow
E	Either	Any	Any	Any	Any	Deny

Packet	Direction	Src	Dest	Protocol	Dest Port	Action (rule)
1	In	121.12.1.2	222.2.2.2	TCP	25	Allow (A)
2	Out	222.2.2.2	121.12.1.2	TCP	5150	Allow (B)

Building Internet Firewalls: Internet and Web Security, by Elizabeth D. Zwicky, Simon Cooper, D. Brent Chapman, ch. 8.

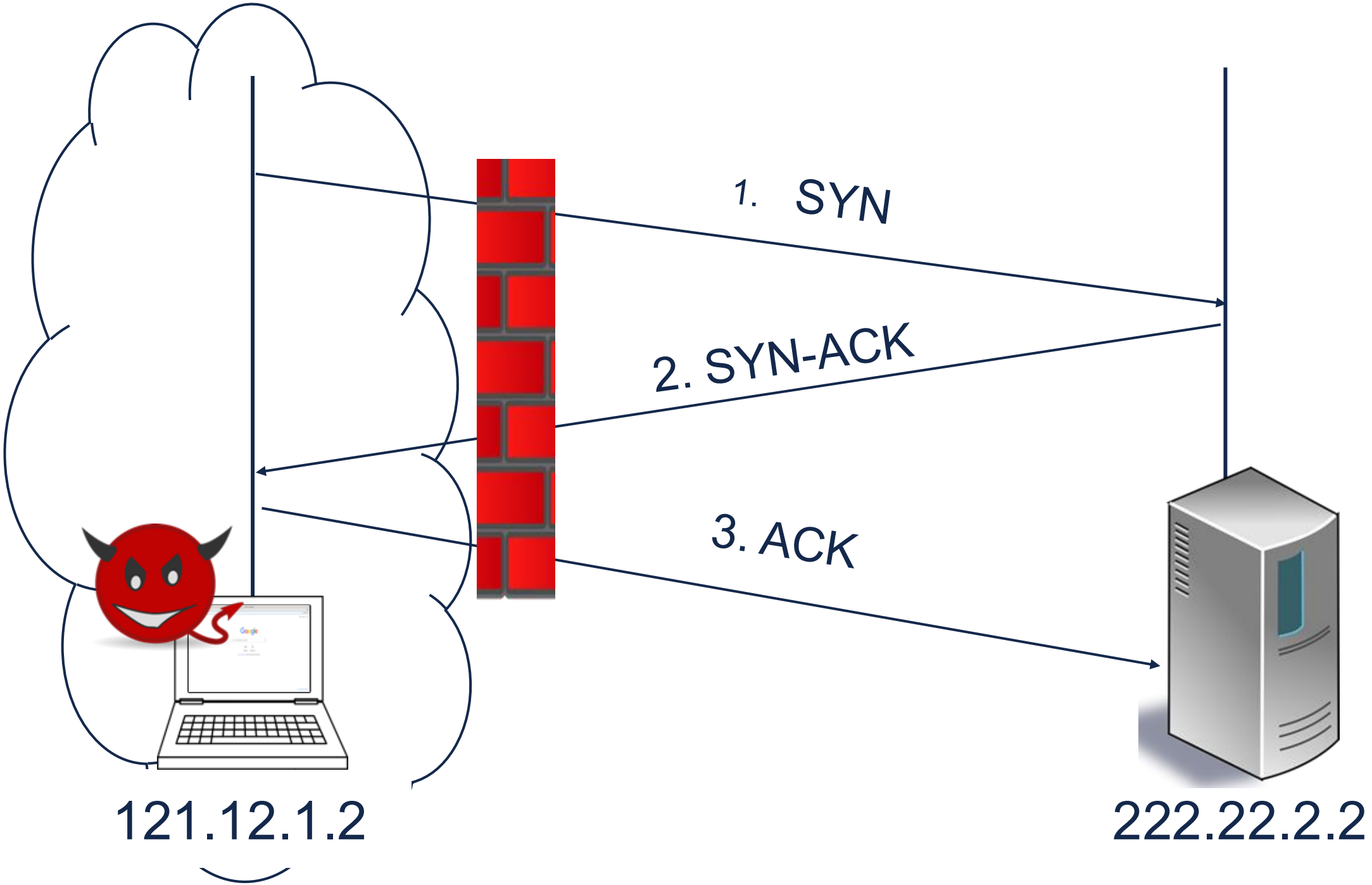

1. SYN
2. SYN-ACK
3. ACK
121.12.1.2
222.22.2.2

An example of what outgoing mail might look like. AKA mail being sent from someone inside org to internet.

Internal mail client uses port 5150

			Dest Addr	Protocol	Dest Port	Action
A	In	External	Internal	TCP	25	Allow
B	Out	Internal	External	TCP	>1023	Allow
C	Out	Internal	External	TCP	25	Allow
D	In	External	Internal	TCP	>1023	Allow
E	Either	Any	Any	Any	Any	Deny

Packet	Direction	Src	Dest	Protocol	Dest Port	Action (rule)
3	Out	222.22.2.2	121.12.1.2	TCP	25	Permit (C)
4	In	121.12.1.2	222.22.2.2	TCP	5150	Permit (D)


1. SYN
2. SYN-ACK
3. ACK
121.12.1.2
222.22.2.2

A high destination port does not prove a packet is a reply.
Rule D lets an attacker connect to port 8080 from a web proxy server; rule B lets the server reply.

			Dest Addr	Protocol	Dest Port	Action
A	In	External	Internal	TCP	25	Permit
B	Out	Internal	External	TCP	>1023	Permit
C	Out	Internal	External	TCP	25	Permit
D	In	External	Internal	TCP	>1023	Permit
E	Either	Any	Any	Any	Any	Deny

Packet	Direction	Src	Dest	Protocol	Dest Port	Action (rule)
1	In	121.12.1.2	222.2.2.2	TCP	8080	Allow (D)
2	Out	222.2.2.2	121.12.1.2	TCP	5150	Allow (B)

Updated Packet Filter

One fix: look at source port # as well as destination port #.



Rule	Dir	Src Addr	Dest Addr	Prot	Src Port	Dest Port	Action
A	In	External	Internal	TCP	> 1023	25	Allow
B	Out	Internal	External	TCP	25	> 1023	Allow
C	Out	Internal	External	TCP	> 1023	25	Allow
D	In	External	Internal	TCP	25	> 1023	Allow
E	Either	Any	Any	Any	Any	Any	Deny

Rule	Dir	Src Addr	Dest Addr				
A	In	External	Internal				
B	Out	Internal	External	TCP	25	> 1023	Allow
C	Out	Internal	External	TCP	> 1023	25	Allow
D	In	External	Internal	TCP	25	> 1023	Allow
E	Either	Any	Any	Any	Any	Any	Deny

Here are the malicious packets from before. Now, they get denied.



Packet	Dir	Src Addr	Dest Addr	Prot	Src Port	Dest Port	Action (rule)
1	In	121.12.1.2	222.2.2.2	TCP	5150	8080	Deny (E)
2	Out	222.2.2.2	121.12.1.2	TCP	8080	5150	Deny (E)

Another attacker: Source port 25 does not prove a packet is a mail server's reply.
An attacker can choose port 25: rule D lets them in, and rule C allows the response.

					Src Port	Dest Port	Action
A	In	External	Internal	TCP	> 1023	25	Allow
B	Out	Internal	External	TCP	25	> 1023	Allow
C	Out	Internal	External	TCP	> 1023	25	Allow
D	In	External	Internal	TCP	25	> 1023	Allow
E	Either	Any	Any	Any	Any	Any	Deny

Packet	Dir	Src Addr	Dest Addr	Prot	Src Port	Dest Port	Action (rule)
1	In	121.12.1.2	222.2.2.2	TCP	25	8080	Allow (D)
2	Out	222.2.2.2	121.12.1.2	TCP	8080	25	Allow (C)

Fix: including the ACK flag in the rules

			Dest Addr	Prot	Src Port	Dest Port	ACK Set	Action
A	In	External	Internal	TCP	> 1023	25	Any	Allow
B	Out	Internal	External	TCP	25	> 1023	Yes	Allow
C	Out	Internal	External	TCP	> 1023	25	Any	Allow
D	In	External	Internal	TCP	25	> 1023	Yes	Allow
E	Either	Any	Any	Any	Any	Any	Any	Deny

Packet	Dir	Src Addr	Dest Addr	Prot	Src Port	Dest Port	ACK Set	Action (rule)
1	In	121.12.1.2	222.2.2.2	TCP	25	8080	No	Deny (E)



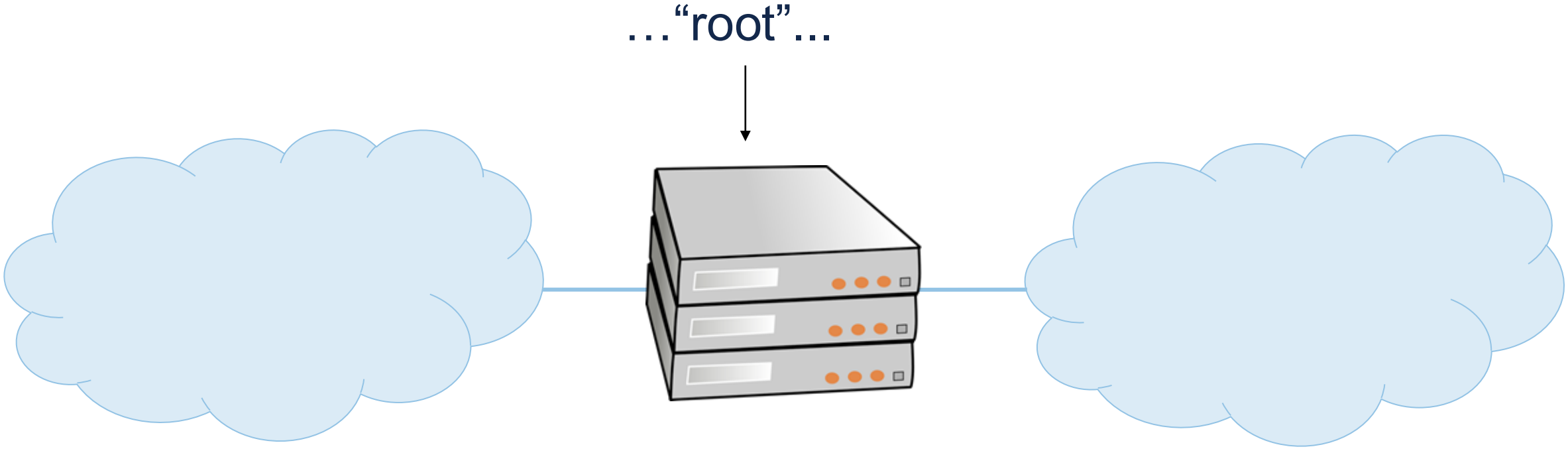
Problem: packet filters have an imperfect view

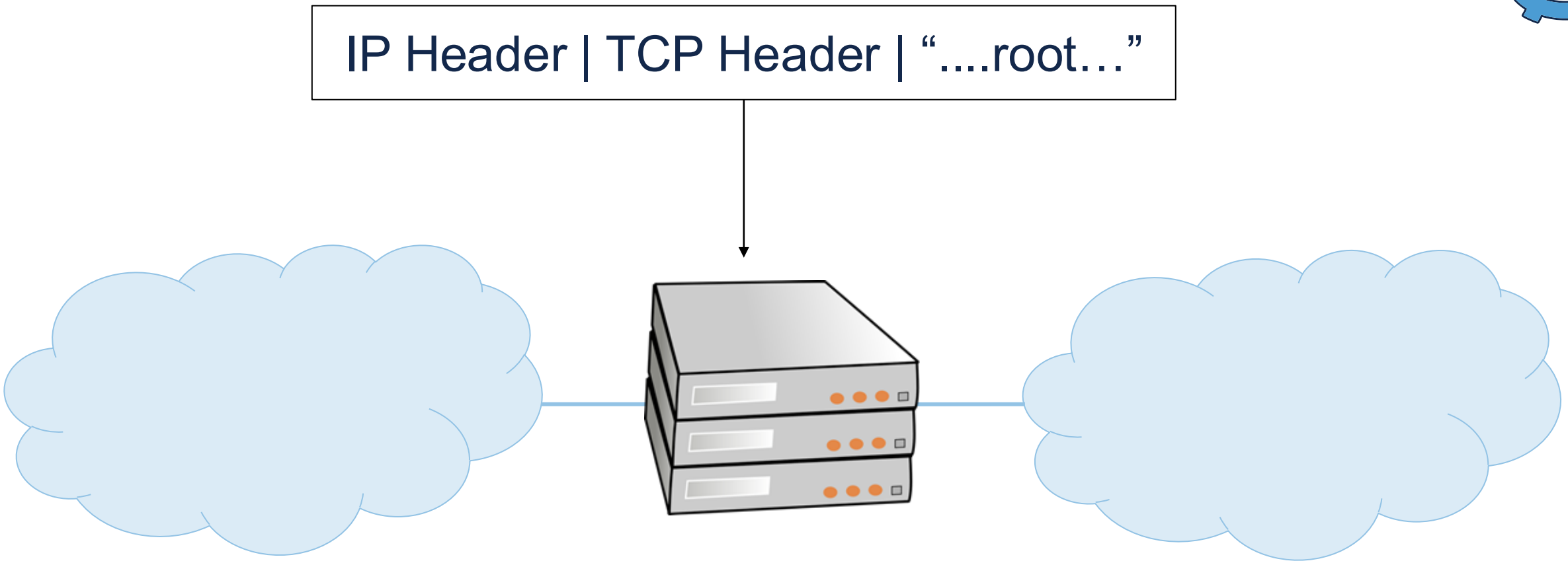
Signature-Based Filters



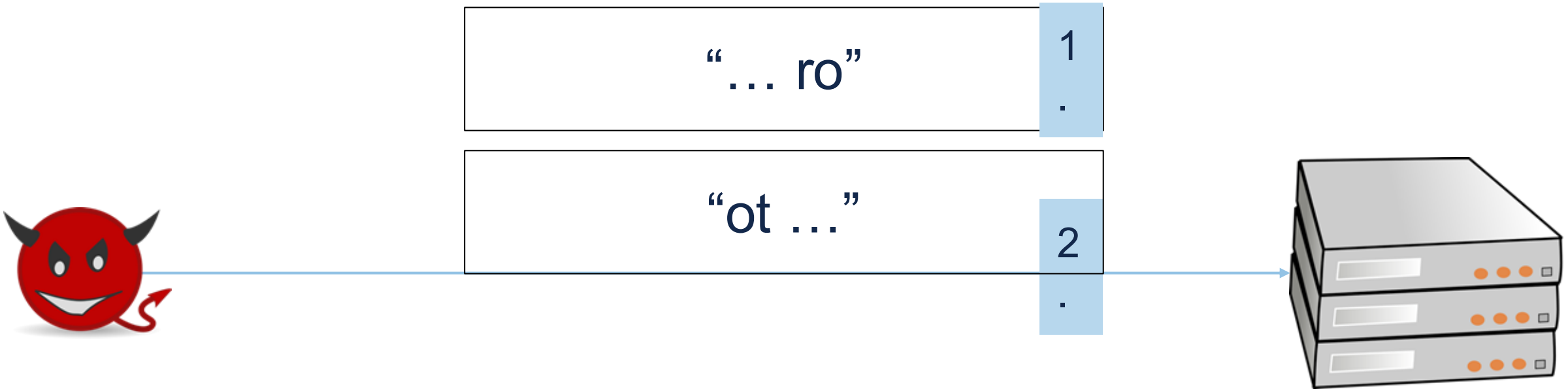
Def'n: a packet filter that inspects the packet payload to find patterns of known attacks

E.g., look for "root"; look for "/etc/passwd"


…“root”...
External Network
Internal Network

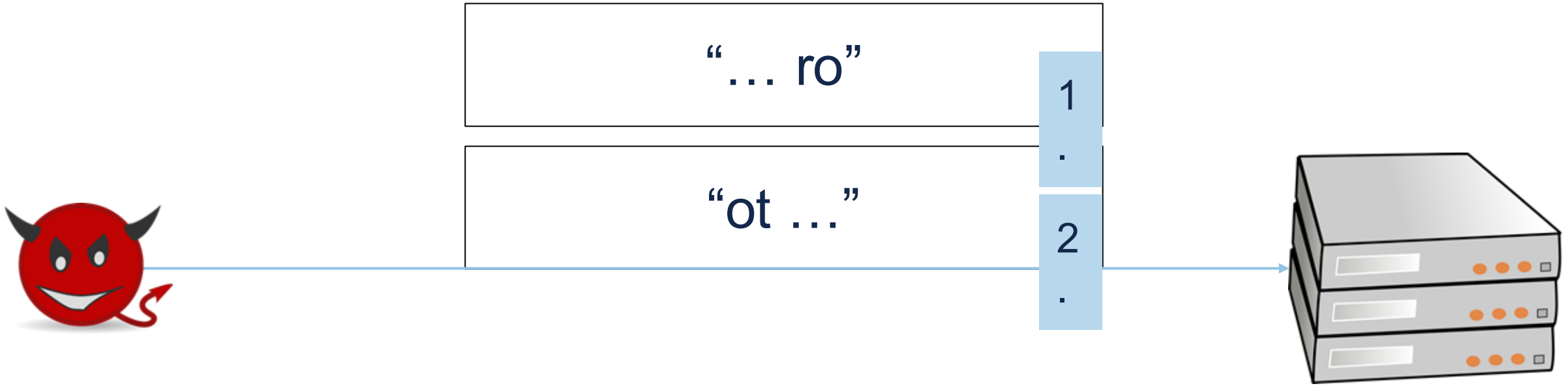

IP Header | TCP Header | “....root…”
External Network
Internal Network

Attack



Neither segment contains “root,” but the reassembled stream does. Inspecting packets individually can miss content split across boundaries.

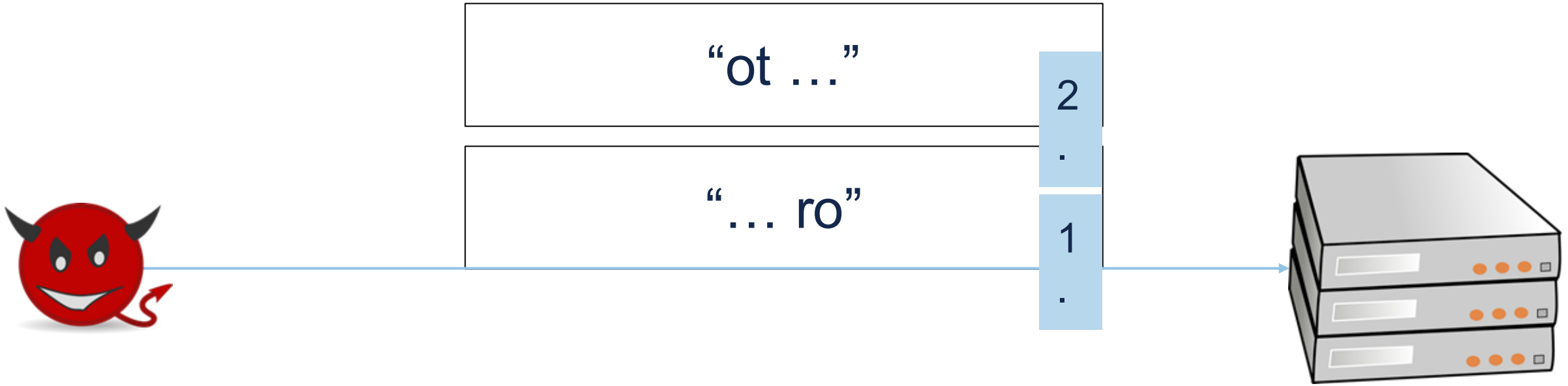
Attack



Fix: Track sequences of packets

Downside…. Have to track more state.

Attack



Attacker can easily ensure segments sent out of order.

Another Attack

TTL=25	“r”
TTL=19	“i”
TTL=25	“o”
TTL=19	“p”
TTL=25	“o”
TTL=25	“t”



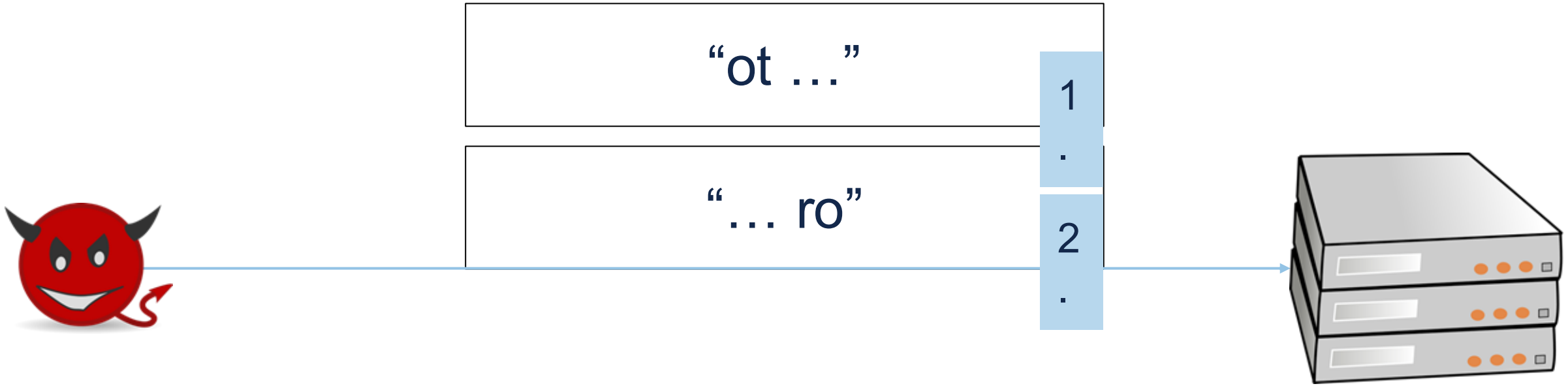


r

o

o

t



Fix: Reassemble the TCP stream