University of Connecticut

Computer Science and Engineering

CSE 4402/5095: Network Security

Vulnerabilities, Firewalls, Packet Filtering

and a bit on IDS/IPS and Honeypots

Last updated: Sunday, 08 December 2024

© Prof. Amir Herzberg

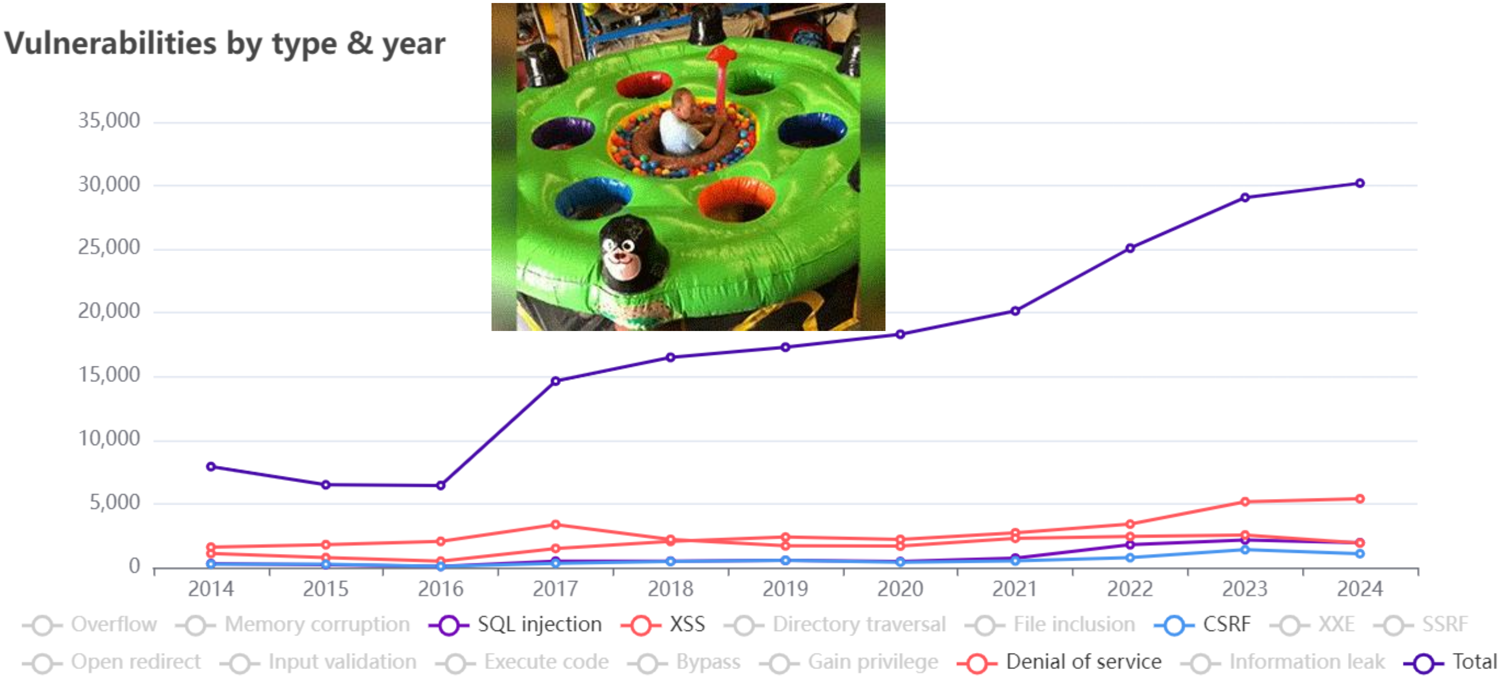
Vulnerabilities, FW & more: Agenda

- **Vulnerabilities**
- Firewalls: protecting the perimeter
  - Packet filtering FW
- Intrusion Detection/Prevention
- Honeypots

Most network attacks exploit Vulnerabilities.

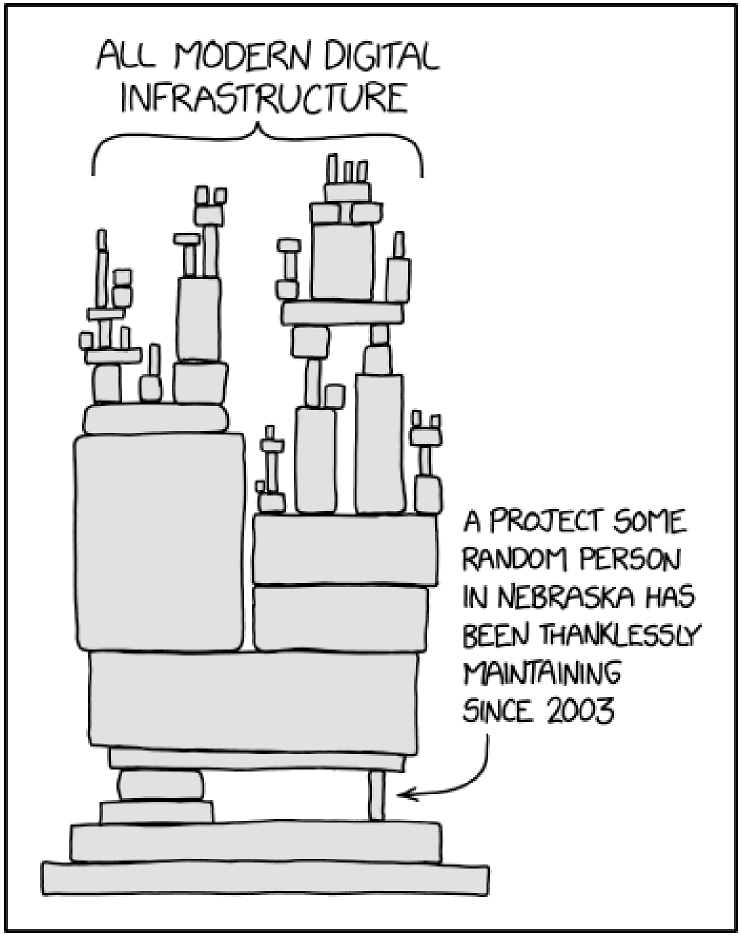
Why not fix all these vulnerabilities?

We find, fix vulnerabilities... And find more!



Why are Vulnerabilities so Common?

- Systems are complex (large `attack surface')
  - Complexity→ more errors, harder to detect/find/fix
  - Vulnerabilities Love Complexity
- Lots of code-reuse, open code
  - →Lots of vulnerabilities-reuse
- **Insufficient motivation** to find, fix:
  - Vendors: limited liability/reputation risk
    - Patching and versioning 'lock' clients
  - Gov'ts find/buy vulnerabilities – to abuse
    - Esp. Zero-Day (ZD) vulnerabilities
    - Snowden: NSA buys ZD for 25M$/year
  - Others?


ALL MODERN DIGITAL INFRASTRUCTURE
A PROJECT SOME RANDOM PERSON IN NEBRASKA HAS BEEN THANKLESSLY MAINTAINING SINCE 2003

Vulnerabilities Markets

- If vendors aren’t motivated to find vulnerabilities, and government abuse the ones they find/buy, who find them? And what’s their motivation?
  - To secure our world
  - For ‘fun and profit’
  - Profit: money and/or credit
- Financial profit:
  - Black markets (sell to anyone)
  - Grey markets: vendors, companies
  - Bug-bounty programs

Bug Bounty Programs

- Pay researchers for disclosed ZD vulnerabilities
  - Based on severity
- Run by many vendors – and some markets
- From CEO of the HackerOne market (2018):
  - Bounties from 100$ to 100,000$, typical ~750$
  - Most well paid hacker: 1M$, total: over 40M$
- Proposals:
  - Governments / international bounty program
    - Argument: $ in damage from attacks >> $ in profit to atkr
  - Compulsory bounty program
- Is it ethical to sell ZD without disclosing/patching?

Cybersecurity Ethics

- Basic cyber-sec ethics:
  - Do no harm
    - Intentional – or by negligence (e.g., experiment `in wild')
- But there are dilemmas…
  - Not disclosing/fixing vulnerabilities, using them for law enforcement, e.g., against terrorists
    - One man's terrorist is another man's journalist
  - To help national security?
    - US Cyber Command:
    - …The two swords represent the dual nature: to defend and **engage our enemies in the cyber domain**.
    - Which nation?

Disclosures: Types and Ethics

- What to disclose
  - Everything (full), partial (only to defend), none
- Who to disclose to (if at all)?
  - Vendor, bug-bounty program, ‘market’, public
- When to disclose?
  - Immediate, after patch/fix, after ‘reasonable time’
- ‘Responsible disclosure’:
  - Full, immediate to vendor
  - Partial or full, after delay/fix, to public
    - Expected from academic papers

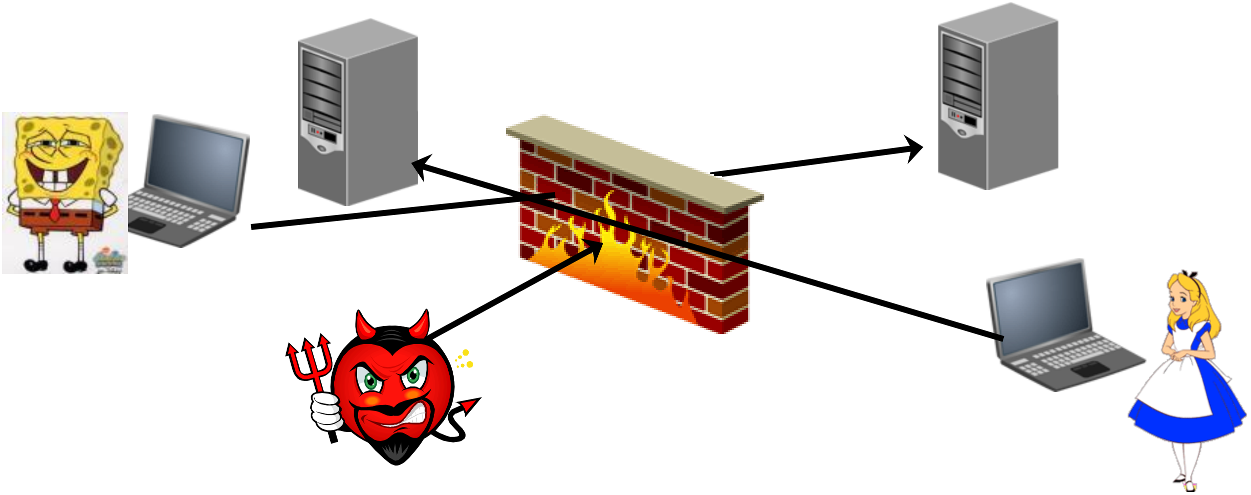
Vulnerabilities, FW & more: Agenda

- Vulnerabilities
- **Firewalls and packet filters**
- Intrusion Detection/Prevention
- Honeypots

Firewalls – Keeping Attackers Out

Secure / trusted machine/module:

- On path between two or more networks / host(s)
  - Avoid damage from outside - or from spreading
  - We focus on Intranet (‘behind’ FW) vs. Internet (‘outside’)
- Controls, inspects and filters the communication
  - Prevents / limits reconnaissance, exploits
  - ‘Fixes’ traffic, e.g., translate addresses (NAT), fuzzing



Packet-Filtering Firewall

- Most basic and common firewall: a router/switch
- Filters packets to block/detect attacks
  - Between network and ISP (aka AS – 'Autonomous System'), or between two ASes (typically, customer and provider)
- Filtering policy: ordered list of 'access control rules'
  - Rule: an action – and which packets to apply action to
  - Actions: allow, drop, **reject**, alert, redirect
- Selection of packets that rule applies to:
  - Typically: conditions on header fields
  - Stateless (efficient) or Stateful (more powerful)
    - Stateful FW can filter on existing (TCP) connection
  - Usually: abort on first match (skip remaining rules)
    - Order rules correctly for security and performance!

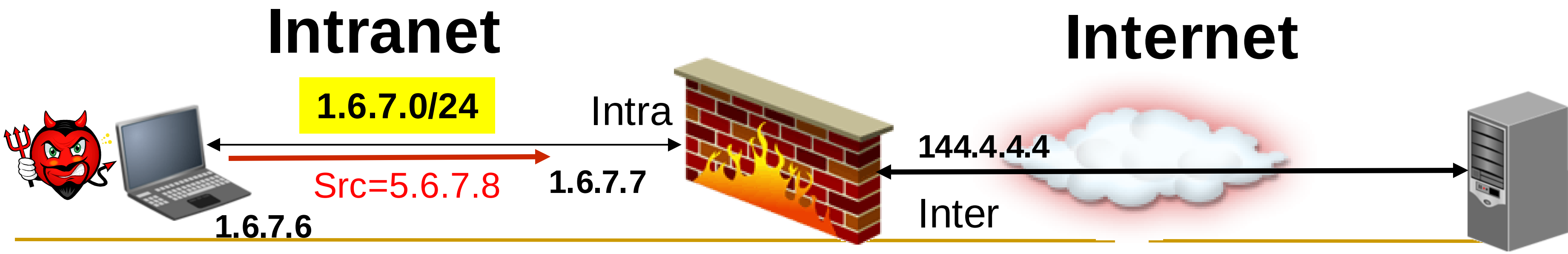
Typical Filtering Rules

- Anti IP spoofing rules:
  - Ingress filtering (packets sent into the Internet)
    - Allow only packets with assigned source IP addresses
  - Egress filtering (packets from (exiting) the Internet)
    - Drop incoming pkts with internal IPs and `forbidden' IPs
    - Drop other spoofed packets (Source Address Validation – SAV)
- Attack blocking rules:
  - Block connections/requests from Internet (except to servers)
  - Block (and detect) suspect packets sent to Internet

Ingress Filtering: Prevent Spoofed IP Packets

- Spoofing enables DoS and other off-path attacks, e.g. ???
- Ingress filtering: ISPs should drop spoofed packets from customers [BCP38,RFCs 2827, 3013, 3704,…]

Rule name / goal	Intf	Src IP, port	Dst IP, port	Protocol	Flags	Action
Ingress filtering (filter traffic from Intranet, sent to the Internet)						



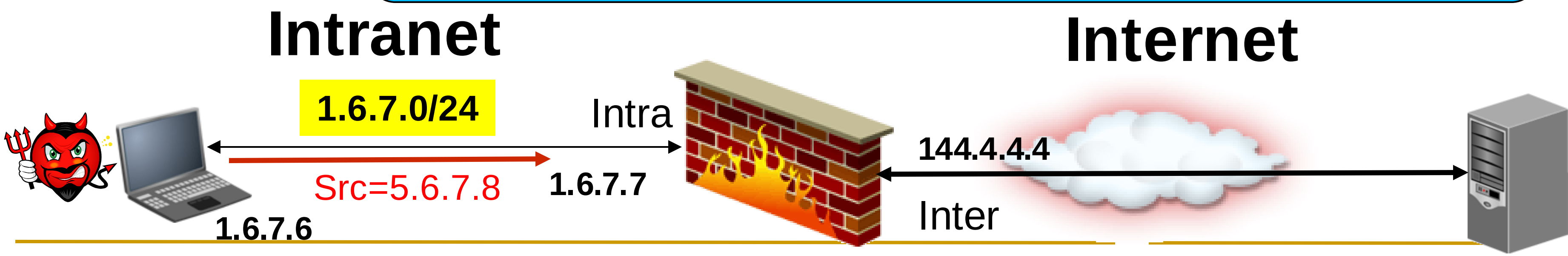
Ingress Filtering: Prevent Spoofed IP Packets

- Spoofing enables DoS and other off-path attacks, e.g. DNS
- [BCP38, ingress filtering]: ISPs <u>should</u> filter spoofed packets from customers

Rule name / goal	Intf	Src IP, port	Dst IP, port	Protocol	Flags	Action
Ingress filtering (filter traffic from Intranet, sent to the Internet)	Intra	Not in 1.6.7.*				Drop

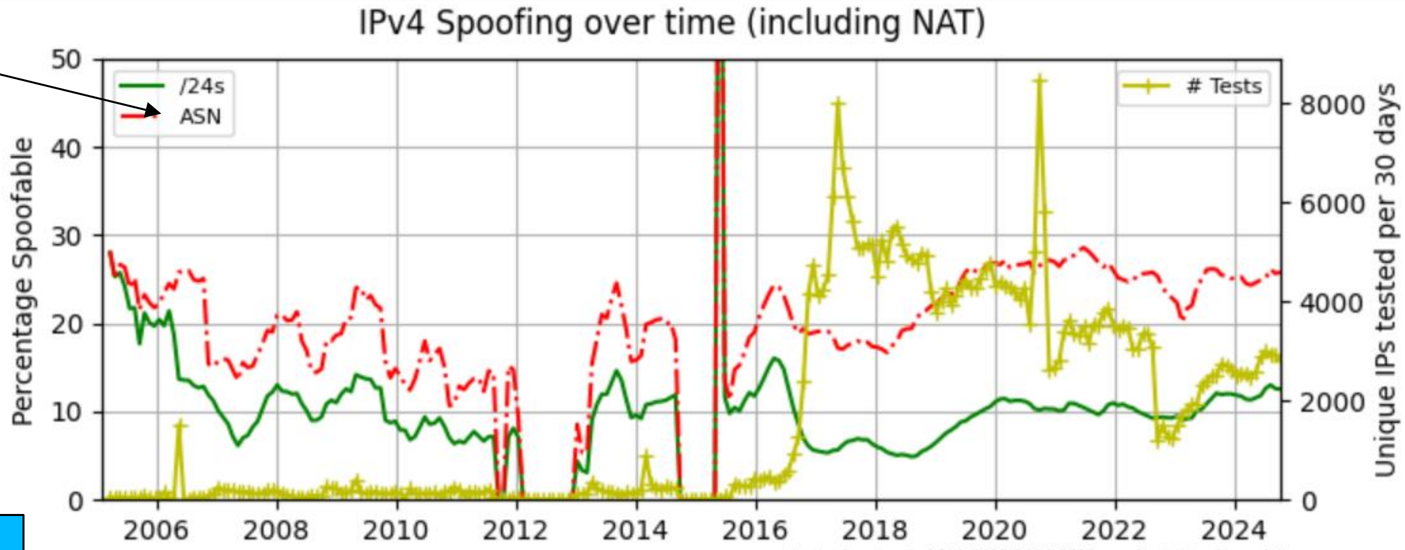
Unfortunately, not all ISPs do ingress filtering

Few incentives vs. real costs



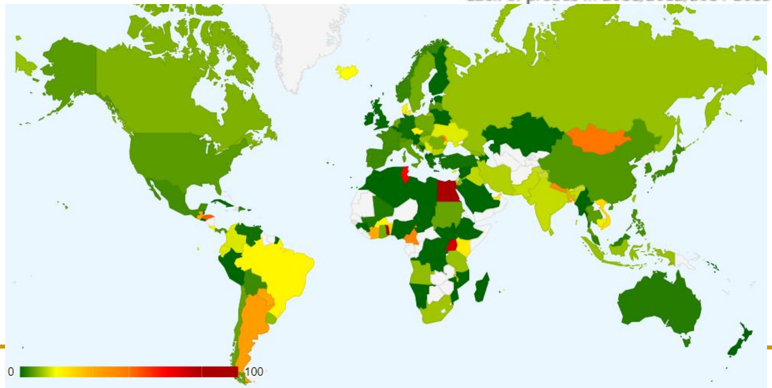
How many IPs are ingress-filtered?

- As measured by CAIDA's spoofer project
  - Only end-users: participation bias, no hosted servers/nets



Lack of probes in 2011/2012/2014-2015 are due to hardware failure

ASN=
AS
number

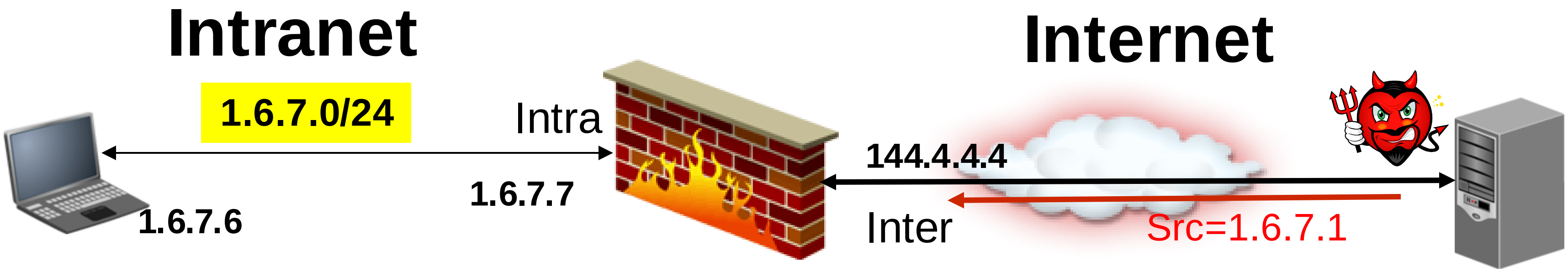


Typical Filtering Rules

- Anti IP spoofing rules:
  - Ingress filtering (packets sent into the Internet)
    - Allow only packets with assigned source IP addresses
  - Egress filtering (packets from (exiting) the Internet)
    - Drop incoming pkts with internal IPs and `forbidden' IPs
    - Drop other spoofed packets (Source Address Validation – SAV)
- Attack blocking rules:
  - Block connections/requests from Internet (except to servers)
  - Block (and detect) suspect packets sent to Internet

Stateless Egress Filtering Rule

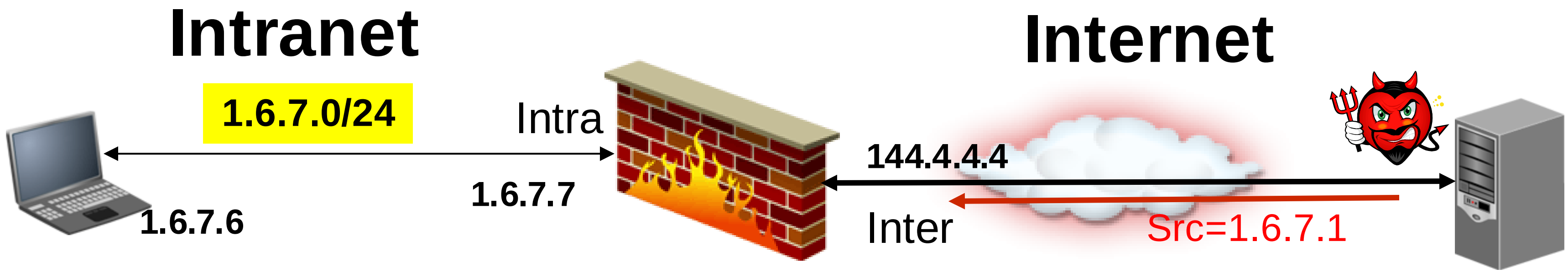
Rule name / goal	Intf	Src IP, port	Dst IP, port	Protocol	Flags	Action
Egress filtering Drop incoming pkts with internal IPs						



Stateless Egress Filtering Rule

Rule name / goal	Intf	Src IP, port	Dst IP, port	Protocol	Flags	Action
Egress filtering Drop incoming pkts with internal IPs	Inter	In 1.6.7.*				Drop

Doesn't prevent spoofing of other IP addresses



Typical Filtering Rules

- Anti IP spoofing rules:
  - Ingress filtering (packets sent into the Internet)
    - Allow only packets with assigned source IP addresses
  - Egress filtering (packets from (exiting) the Internet)
    - Drop incoming pkts with internal IPs and `forbidden’ IPs
    - Drop other spoofed packets (Source Address Validation – SAV)
- Attack blocking rules:
  - Block connections/requests from Internet (except to servers)
  - Block (and detect) suspect packets sent to Internet

IP Spoofing vs. filtering

- IP does not ensure Source Address Validation (SAV)
  - ➔ IP spoofing, off-path attacks
  - Done from non-ingress-filtering ISPs [BCP38]
- Source Address Validation (SAV) Filtering
  - Unicast Reverse Path Forwarding [RFC3704]
  - Enhanced Feasible Path uRPF (eFP-uRPF) [RFC8704]
  - In routing lecture: BAR-SAV filtering
  - Ad-hoc: learn, then filter on TTL (hop-count)
  - Tradeoffs: false positives (filter benign packets) and false negatives (allow spoofed packets)

SAV with uRPF

- uRPF: Unicast Reverse Path Forwarding
- Strict uRPF: allow packets **from** srcIP x via interface I, if there is a path **to** destIP x via interface I
  - Used for *stubs* (AS with one neighbor) or symmetric routing
- Feasible-path uRPF (FP-uRPF): allow packets **from** srcIP x via interface I, when some alt-route to x is via interface I
  - May work for (asymmetric) routing
- Loose uRPF: allow if there is any route to x
- Limited use of Feasible ; almost no value for Loose
- Later, improved variants: eFP-uRPF and BAR-SAV

Typical Filtering Rules

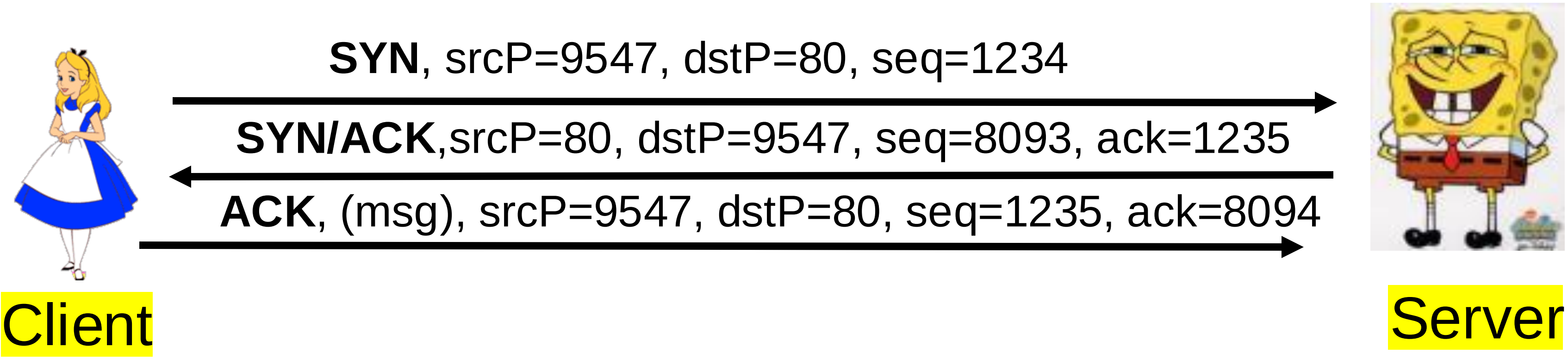
- Anti IP spoofing rules:
  - Ingress filtering (packets sent into the Internet)
    - Allow only packets with assigned source IP addresses
  - Egress filtering (packets from (exiting) the Internet)
    - Drop incoming pkts with internal IPs and `forbidden' IPs
    - Drop other spoofed packets (Source Address Validation – SAV)
- Attack blocking rules:
  - Block connections/requests from Internet (except to servers)
  - Block (and detect) suspect packets sent to Internet

Block **Connections**/Requests from Internet

- Clients **initiate connections** and send **requests**
  - **Exception:** FTP – server initiates `data` connection
- FW rules block incoming connections and requests
  - Except to (public) servers [see later – DMZ]
- First: **TCP (connection-based services)**
- Later: UDP & ICMP (connection-less services)

TCP: Transmission Control Protocol

- TCP is the Internet’s main transport layer protocol
- TCP server application (e.g., http) **listens** to a port
- TCP client (e.g., browser) **connects** to server port
    - Using an arbitrary client port (not connected to this server IP:port)
- TCP sends packets using the Internet Protocol (IP)
    - Packets of a connection identified by (clientIP:port,serverIP:port)
- Connections begin with **three-way handshake:**



SYN, srcP=9547, dstP=80, seq=1234

SYN/ACK,srcP=80, dstP=9547, seq=8093, ack=1235

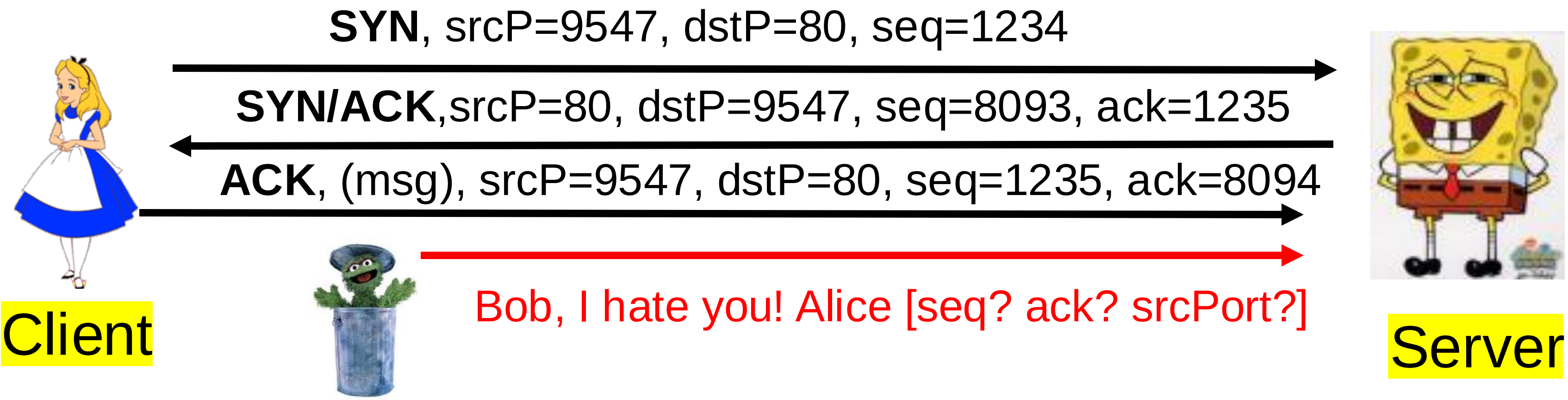
ACK, (msg), srcP=9547, dstP=80, seq=1235, ack=8094

Client

Server

TCP Services

- Like every transport protocol, TCP ensures port-based communication between applications in different hosts
- TCP further ensures:
  - Reliability: messages received as sent (or connection RST)
  - Congestion control: slow down if path is congested
  - Flow control: don’t overfill recipient’s buffers
  - Challenge-response authentication against off-path attackr



SYN, srcP=9547, dstP=80, seq=1234

SYN/ACK,srcP=80, dstP=9547, seq=8093, ack=1235

ACK, (msg), srcP=9547, dstP=80, seq=1235, ack=8094

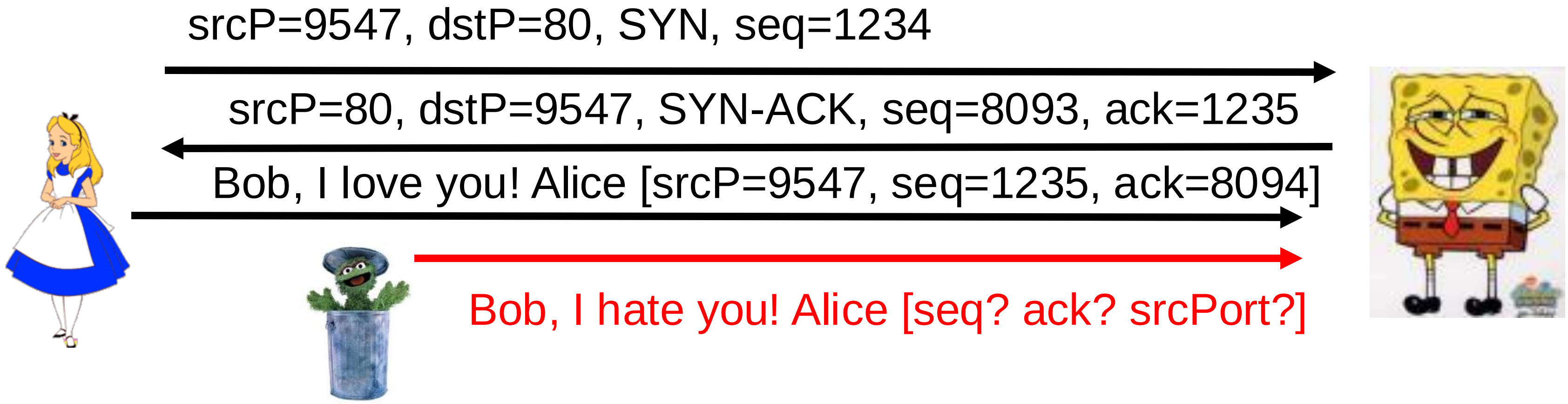
Bob, I hate you! Alice [seq? ack? srcPort?]

Client

Server

Off-path TCP inject challenges

- No explicit off-path defenses in TCP
- But… TCP injection requires:
  - 4-tuple: (clientIP:**port**, serverIP:port)
    - IPs and server port are often known
  - And sequence/ack numbers
  - Initialized randomly (since the 1990s)



srcP=9547, dstP=80, SYN, seq=1234

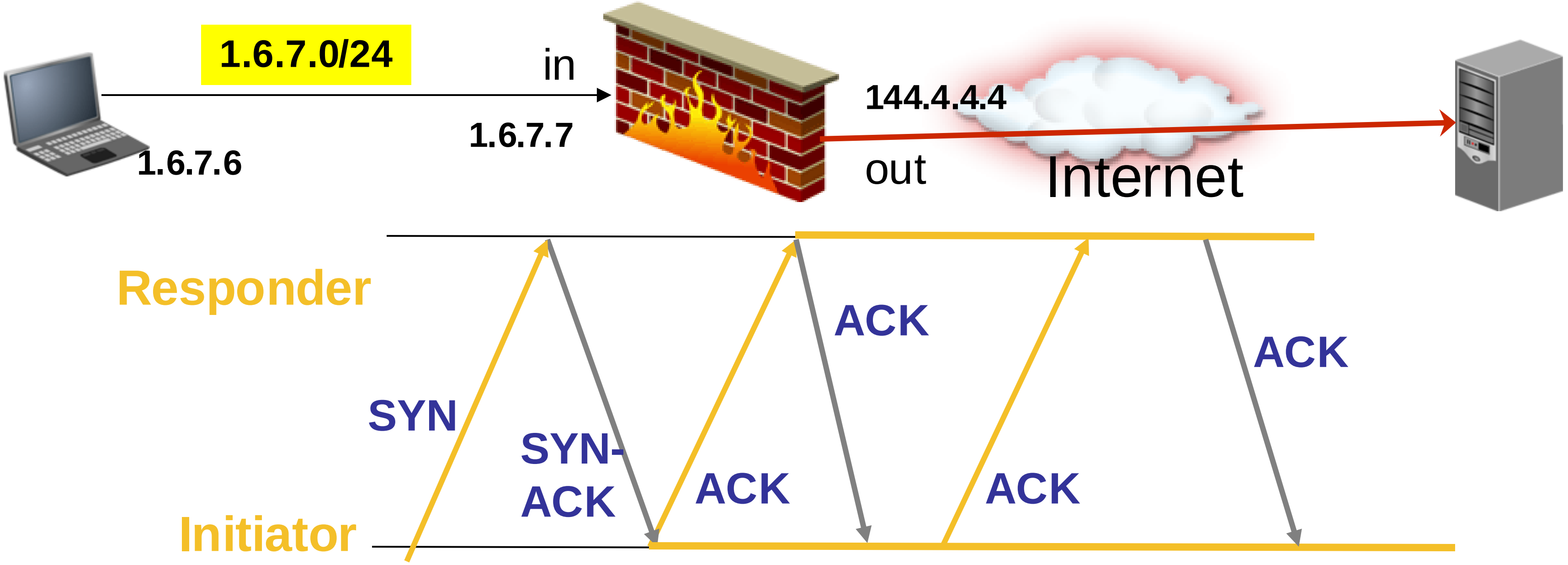
srcP=80, dstP=9547, SYN-ACK, seq=8093, ack=1235

Bob, I love you! Alice [srcP=9547, seq=1235, ack=8094]

Bob, I hate you! Alice [seq? ack? srcPort?]

Block Incoming TCP Requests

Rule name / goal	Intf	Src IP, port	Dst IP, port	Protocol	Flags	Action
No incoming TCP connections				TCP		



1.6.7.0/24

in

1.6.7.7

1.6.7.6

144.4.4.4

out

Internet

Responder

Initiator

SYN

SYN-ACK

ACK

ACK

ACK

ACK

Block Incoming TCP Requests

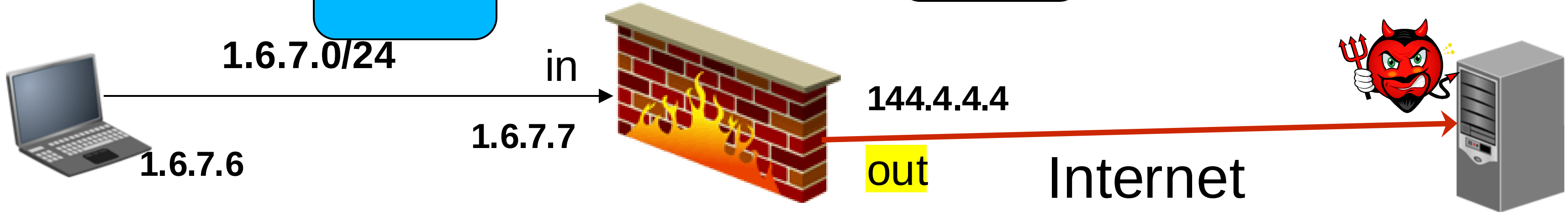
- TCP initiation is always by sending a SYN packet
  - Only legit TCP segment without ACK bit
- Responder sends back a SYN-ACK packet
- SYN bit is only set in these first two packets
- Hence: **Block incoming (SYN) packets without ACK bit**
  - **Or simply allow drop incoming pkts without ACK**

Rule	Intf	Src IP:p	Dst IP, port	Protocol	Flag	ion
No incoming TCP connections	Out	*:*	*:*	TCP	SYN+No ACK	DROP

???

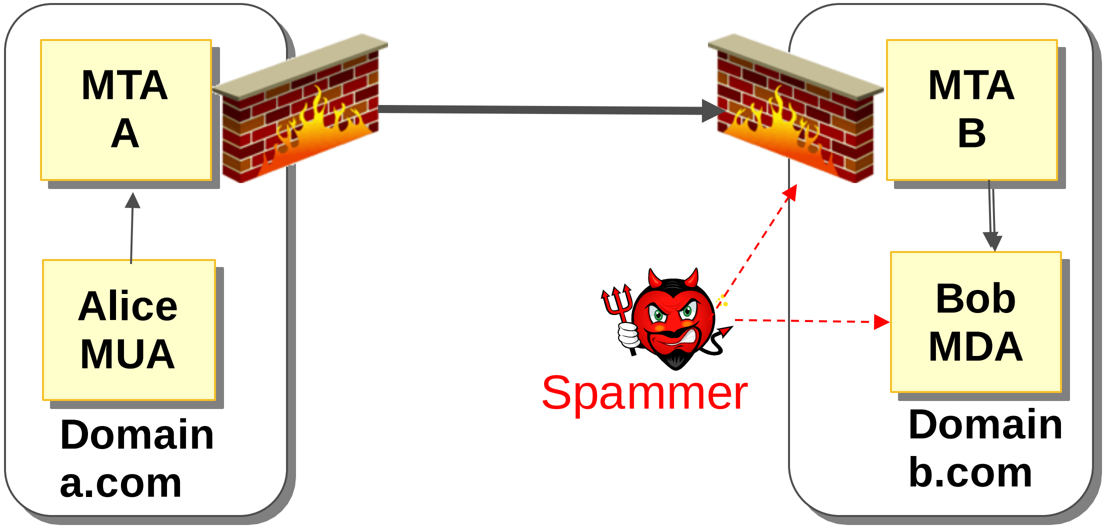
???

???



Example: filtering incoming SMTP

MTA A: a.com's mail transfer agent (prevents sending spam)



MTA B: b.com's mail transfer agent (blocks mail from blacklisted IPs)

- SMTP: simple mail transfer protocol, listens to port 25
- Blacklists identify suspect-spamming IP addresses
- MTA B block mail from blacklisted IP (e.g., spammer)
- Spammer could try to send directly to Bob's MDA
  - MDA : mail delivery agent ; MUA: mail user agent (client)
- But FW allows SYN from Internet to port 25 only to MTA

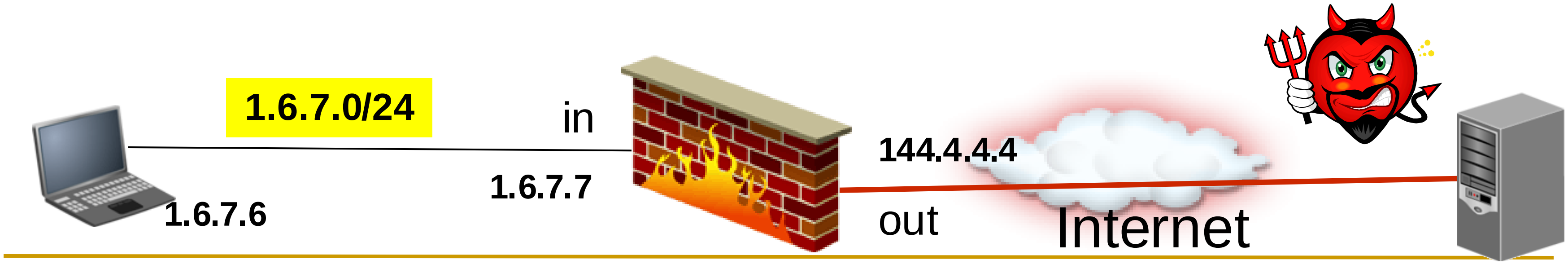
Block incoming connectionless requests

- **Connectionless transport protocols: UDP, ICMP**
  - QUIC provides connection-services over UDP
- How do we block incoming requests?
- **Solution 1**: **Drop all UDP, ICMP traffic**
  - Whitelist necessary, e.g., DNS (and QUIC, ICMP unreachable)
  - May whitelist only specific IPs (but IP could be spoofed)
- **Solution 2**: **Drop packets to `known service ports' (<1024,)**
- **Solution 3 (stateful)**:
  - Record Src=x.x.x.x:p, dst=y.y.y.y:q, time for outgoing packets
    - Allow responses packets: src=y.y.y.y:q, dst=x.x.x.x:p
    - For up to some time-limit after sending request (few seconds)
  - Similarly for NAT

Example of stateless UDP rules: Allow (only) DNS responses from Internet (egress)

Rule name / goal	Intf	Src IP, port	Dst IP, port	Proto col	Flags	Action
Allow DNS response	out	53	>1024	UDP		Allow
Block other incoming UDP	out	*	*	UDP		DROP

Note: ‘allow’ must be placed after egress SAV rule(s)

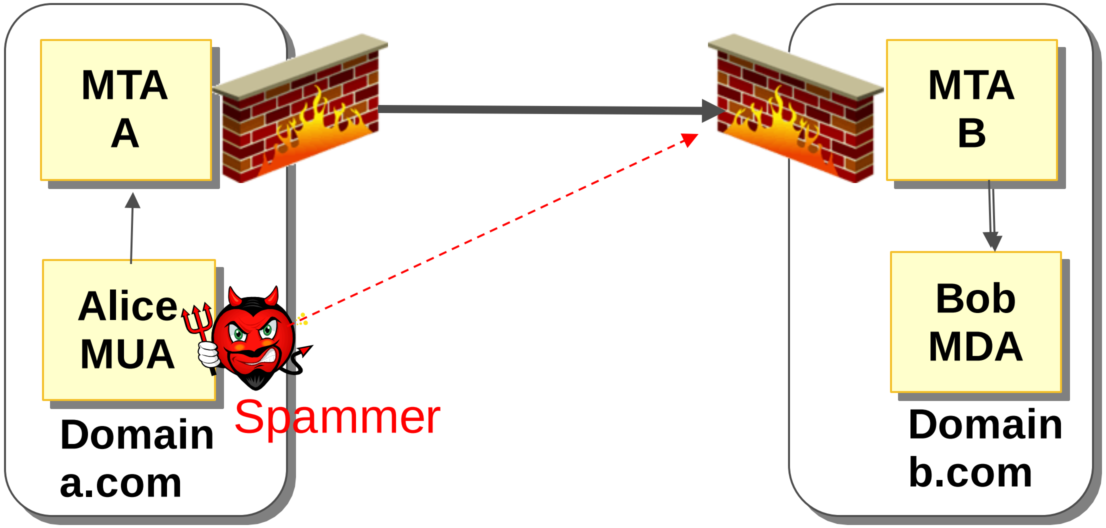


Typical Filtering Rules

- Anti IP spoofing rules:
  - Ingress filtering (packets sent into the Internet)
    - Allow only packets with assigned source IP addresses
  - Egress filtering (packets from (exiting) the Internet)
    - Drop incoming pkts with internal IPs and `forbidden' IPs
    - Drop other spoofed packets (Source Address Validation – SAV)
- Attack blocking rules:
  - Block connections/requests from Internet (except to servers)
  - Block (and detect) suspect packets sent to Internet

Example: filtering outgoing SMTP

MTA A:
a.com's mail
transfer agent
(prevents
sending spam)

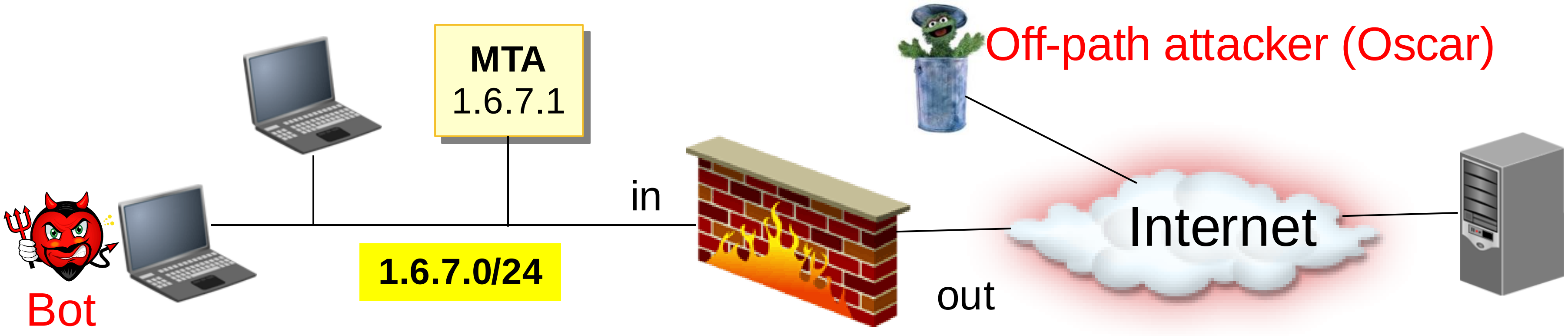


MTA B: b.com's
mail transfer
agent (blocks
mail from
blacklisted IPs)

- SMTP: simple mail transfer protocol, listens to port 25
- Blacklists identify suspect-spamming IP addresses
- MTA B block mail from blacklisted IP, blacklists spammers
- MTA A filters spam (and avoids getting blacklisted)
- Spammer controlling Alice's MUA can try to send directly
- But FW allows only MTA A to connect to port 25 (SMTP)

SMTP filtering rules

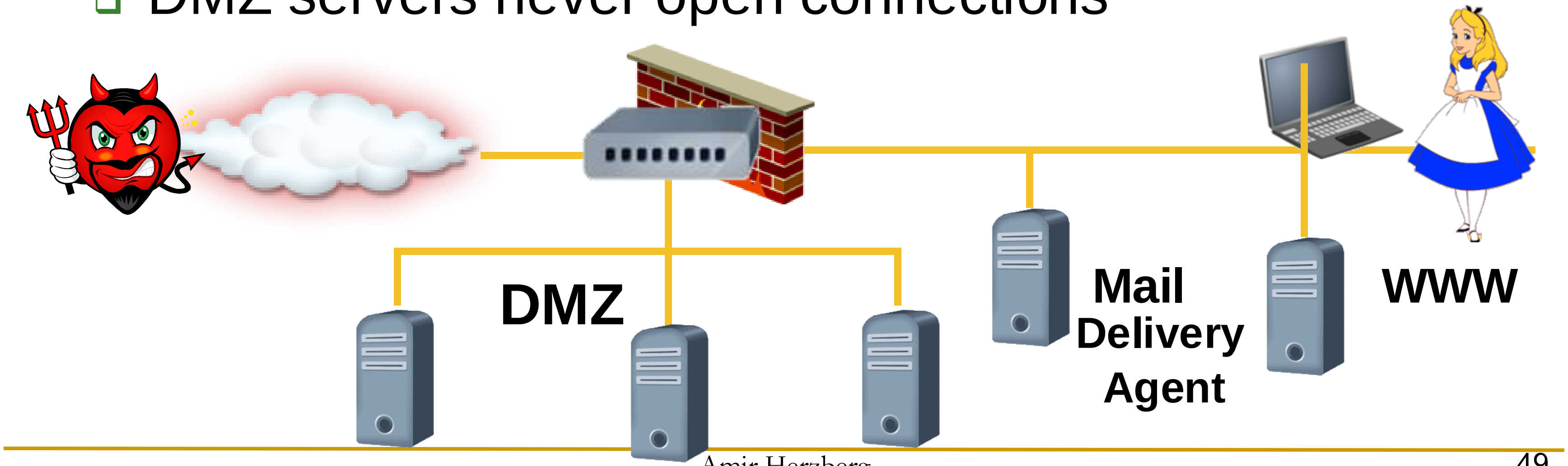
Rule	Intf	Src IP: port	Dst IP: port	Protocol	Flags	Action
Receive mail only via the MTA	out	*	1.6.7.1:25	TCP		Allow
			*:25			Drop
Send mail only via the MTA	in	Not MTA:*	*:25	TCP		DROP+ Alert



Block all traffic to/from port 25, except via MTA

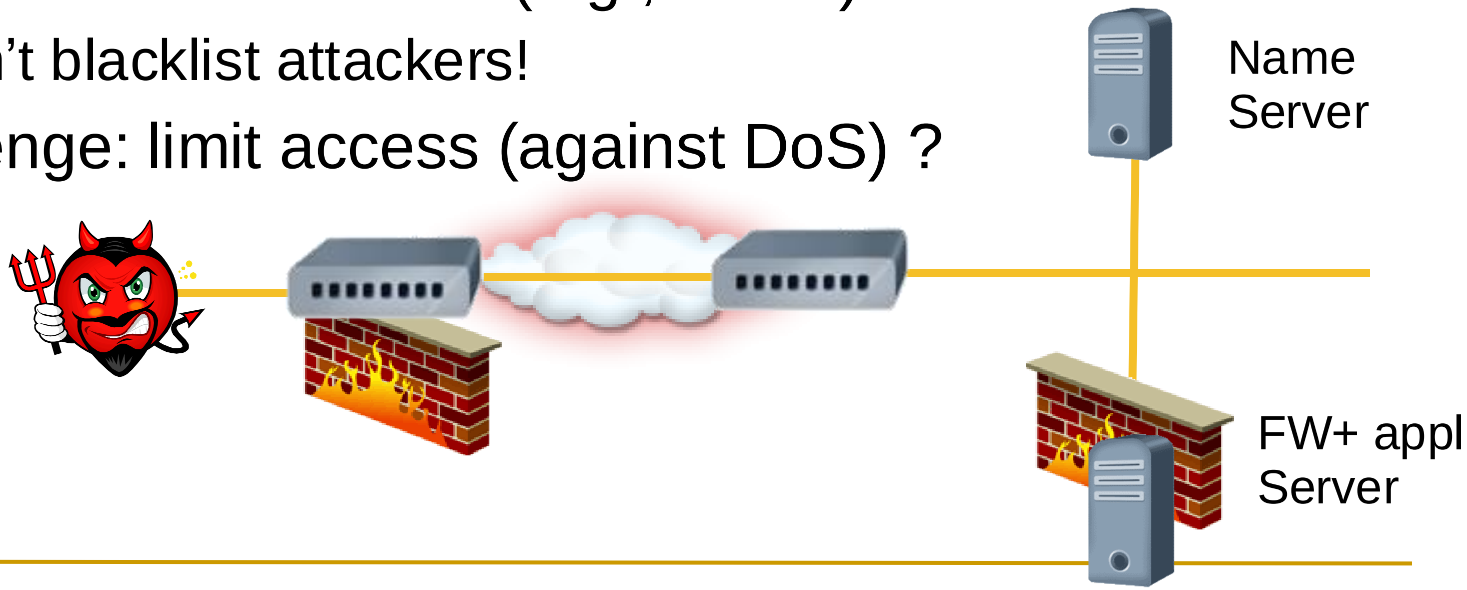
De-Militarized Zone (DMZ)

- Subnet for public services:
  - External web/FTP, Incoming mail server, DNS, …
    - Large attack surface ➔ Separate ‘Internal’ net, servers
  - Allow connections only from Internet
- Often: **separate interface of packet filter**
- Block outgoing connections and alert
  - DMZ servers never open connections



Cloud Firewalls and ‘free ACLs’

- Firewalls often deployed in clouds to protect hosted networks/hosts or physical network
- Also, clouds offer basic stateless firewall to guests
  - ‘Network ACLs’: basic stateless rules
  - Free of charge for rules, dropped (incoming) traffic
  - Limited number of rules (e.g., 20-50)
    - Can’t blacklist attackers!
  - Challenge: limit access (against DoS) ?



Few words on Intrusion Detection Systems

- IDS Goals: **detect, log, alert** [; IPS: also prevent]
  - For traffic that passed the FW’s filtering
- Detect known attack signatures / patterns
- Detect other attacks
  - Based on heuristics & statistics (anomaly detection)
- Critical: **minimize false alarms**
  - Many events * 1% → still too many → ignored !
    - 1% of 100M packets is still 1M!
- Attackers respond by different evasion techniques…

Decoys and Honeypots/nets

- Challenge: how do we detect new attacks?
- Idea: detect attack by <u>any</u> access to ‘decoy’
  - Object created (only) to detect access
  - Learn about attack(er): new malware, spam, IP, content, method…
  - Waste attacker resources (time)?
- Decoys:
  - Decoy host (honeypot), network (honeynet)
  - File or records in DB (detect access / modification)
  - User/password in password file
  - Email mailbox (detect spam messages)
  - Addresses in address-book (detect exposure)

Summary

- Vulnerabilities are an ongoing threat to security
- Firewalls provide an important line of defense
  - Defend the perimeter of a network from outsiders
  - Also, prevent attacks by insiders – and detect them
    - Good citizenship, preserve reputation (avoid blacklist)
  - Limited Source Address Validation (SAV) mechanisms
  - Block incoming requests, connections
  - Block suspect attacks
- Defense in depth: defend within perimeter
  - Internal firewalls protect against insiders
  - IDS/IPS, decoys/honeypots/honeynets, and more