Attack Methods

1. Break-in attacks
2. Denial-of-service

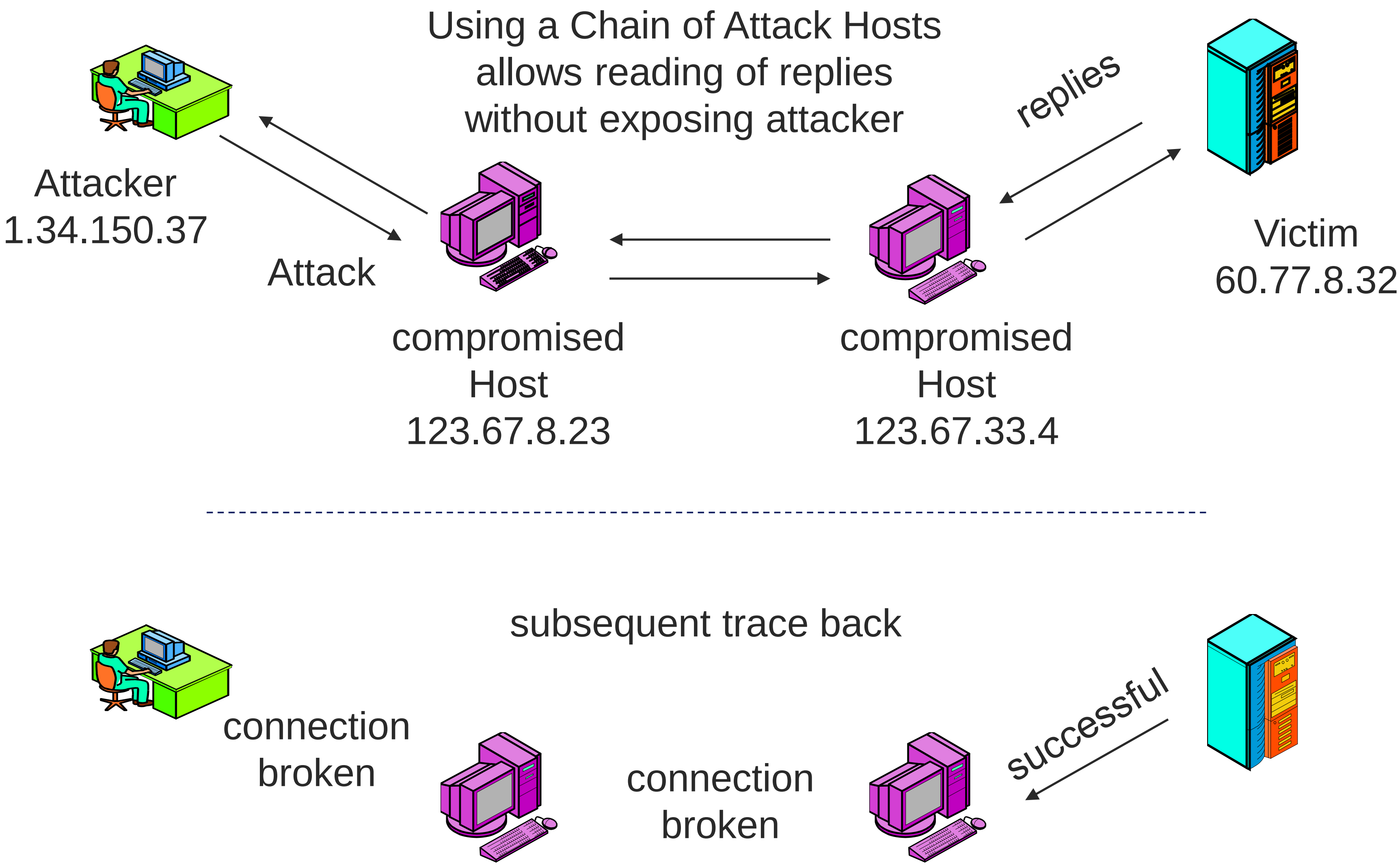
1. Break-in attacks

Break-in attacks

- Targeted attacks
  - Aim at a specific firm
- Starts with a not aggressive information collection
  - Network scanning
  - Look at web site, DNS info about network addresses, key persons, …
- Do a selected scan of servers for open ports/services
- Hide behind other compromised hosts if possible

DNS (Domain Name System) translates domain names to IP addresses

Break-in attacks



Break-in attacks

- (Password guessing; rare)

A. Scanning attack

B. TCP sequence number prediction

C. Session Hijacking

D. Man-in-the-middle attack

(A) Scanning attacks

- (A1) Host and network scanning
  - SYN/ACK scanning
- (A2) Port scanning
  - TCP port scanning
    - Stealth scanning
    - Half-open scanning
  - UDP port scanning
- (A3) Fingerprinting

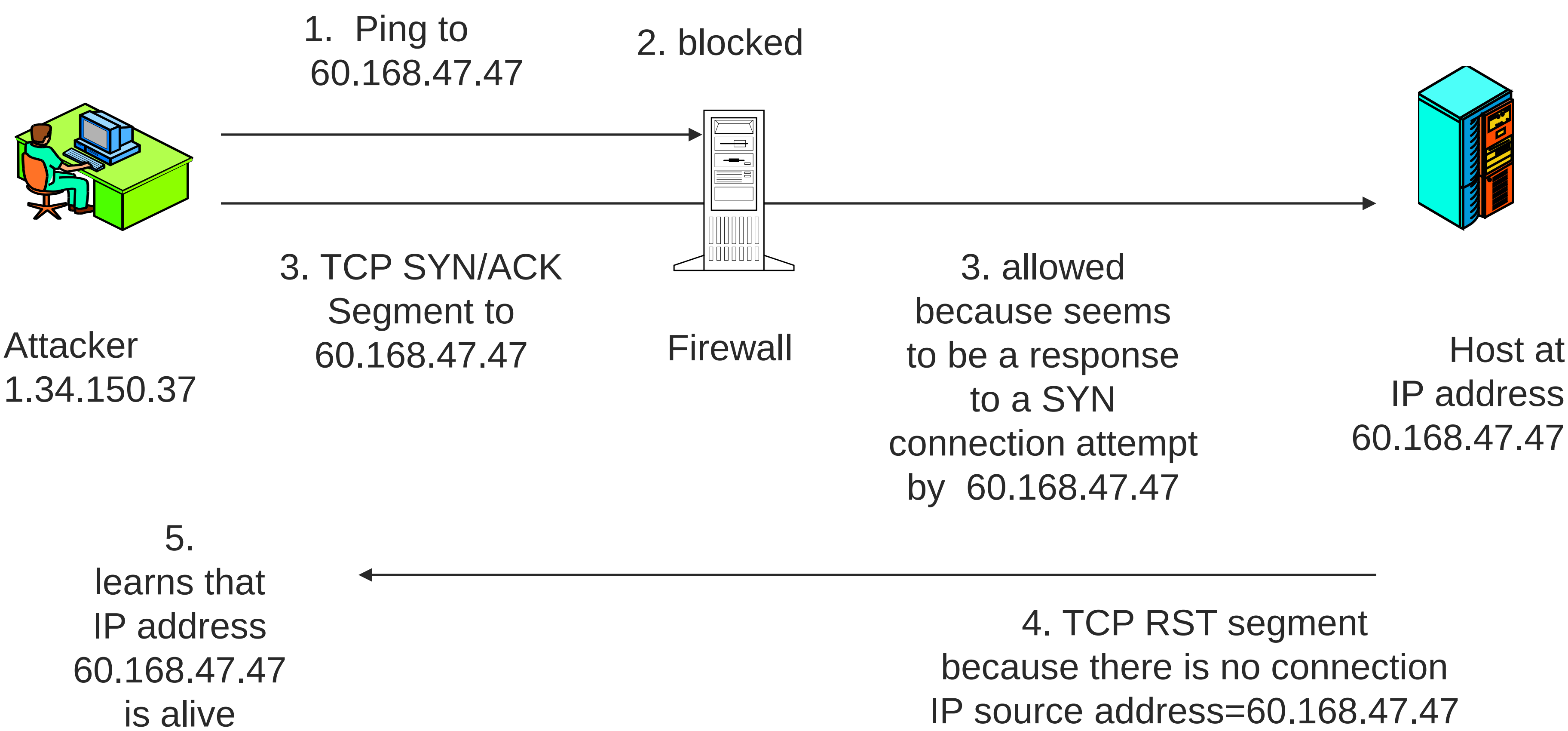
A1: Scanning attacks

- Host scanning
  - To identify possible victims
  - Ping range of IP addresses or use alternative scanning messages
- Network scanning
  - To learn a network's structure (routers, subnets, etc.)
  - Tracert shows all routers along the route to a destination host

A1: Scanning attacks

- Ping and Tracert are often blocked by firewalls
- Send SYN/ACK to generate RST responses
  - these might be blocked as well
  - if it works, log files will probably not show it
- Other RST-generating attacks
  - Send a TCP segment with SYN and FIN flag set on

A1: SYN/ACK scanning attack



A2: Port scanning

- Once a host is identified, do port scanning
- Most break-ins exploit specific services
  - Needed to find services on identified hosts
  - Example: services listen for connections on specific TCP or UDP ports (HTTP=80)
- Noisy process: >65,000 ports
  - Testing a subnet of 254 machines → >16,500,000 packets
  - If well-known services are searched → scan for well-known TCP ports (1024) and all well-known UDP ports (1024)
- This is a good reason for having an IDS!

IDS = Intrusion detection System

A2: TCP port scanning

- Scan servers for open TCP ports
  - Send SYN segments to a particular TCP port number
  - Observe SYN/ACK or RST response

- Stealth scanning
  - Scan fewer systems and ports
  - Scan more slowly to avoid detection
  - Or scan one host from different systems
  - May fool an IDS

A2: TCP port scanning

- <u>Half-open scanning</u>
  - Another possible scan is to begin the 3-ways handshake but never complete it

```
SYN ------>

      <------ SYN/ACK
```

  - Idea: uncompleted connections will not be logged
  - Programs are available to detect this type of scans

A2: UDP port scanning

- Harder because we will get no reply as with TCP
- Send 0 byte UDP packet to each port
  - Does not interfere with application
  - If ICMP port unreachable is received, port is closed
  - If no reply, we don’t know

A3: Fingerprinting

- Learn the victim's operating system (OS), application programs and (if possible) versions
  - Useful because most exploits are specific to particular application programs and versions
- Active fingerprinting
  - Send odd messages and observe replies (may trigger alarms)
  - Uses TCP, IP or ICMP messages, possibly malformed
  - Most OS and application programs respond differently
  - Can be detected by IDS (as most other active attacks)

A3: Fingerprinting

- Passive fingerprinting

  Read packets and look at parameters (TTL, window size, etc.)
  - If TTL is 113, probably originally 128 $\rightarrow$ Windows 2000, or Cisco 12.0
  - Window size field is 18,000. Must be Windows 2000

OS	Version	TTL	Window
Windows	9x/NT	32	8192
Windows	2000	128	17000 -18000
Solaris	8	64	24820
Linux	2.2	64	32120
Cisco	12.0	255	3800-5000

- Countermeasure: users and applications may change TTL to confuse

Demo – scanning a network

- Ping
- Tracert
- Nmap

NMAP scanner

- Network mapper
- Freeware tool (www.nmap.org)
- Available for most OS
  - GUI as an add-on
- Can perform all major scanning methods
  - TCP SYN/ACK scanning
  - Stealth scanning

(B) TCP sequence number prediction

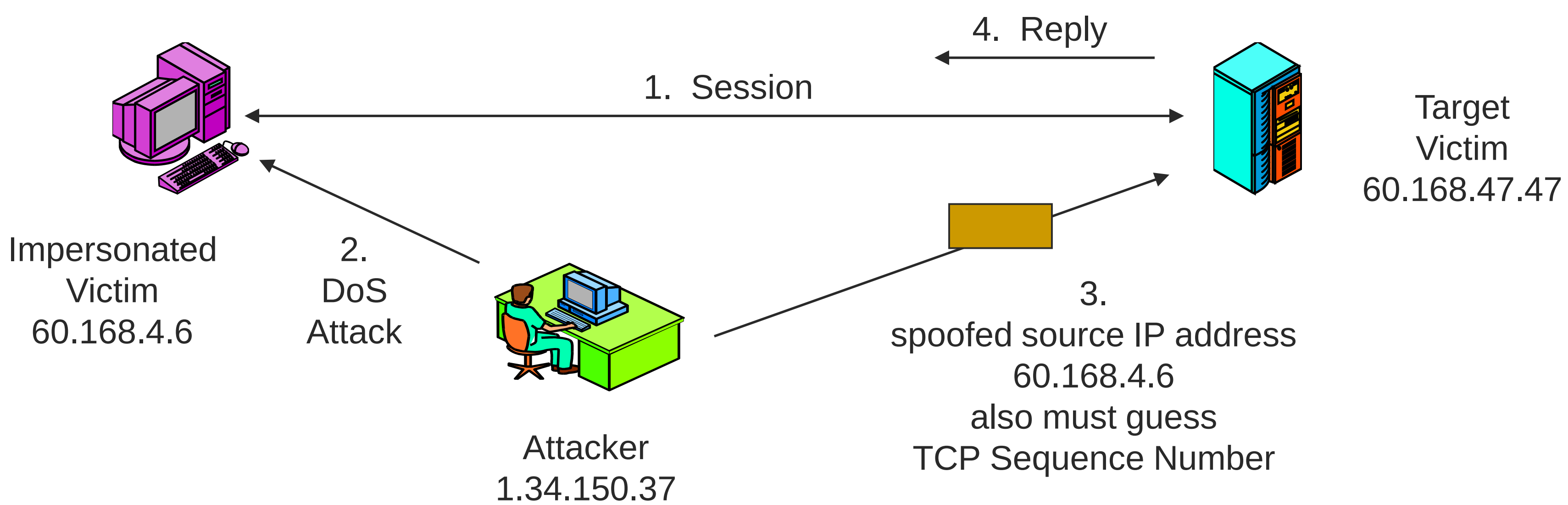
- Connecting several times to a service reveals how the OS selects TCP sequence numbers
- Some are completely random
- Other are more predictable
  - For example add a constant number for each new connection

(B) TCP sequence number prediction

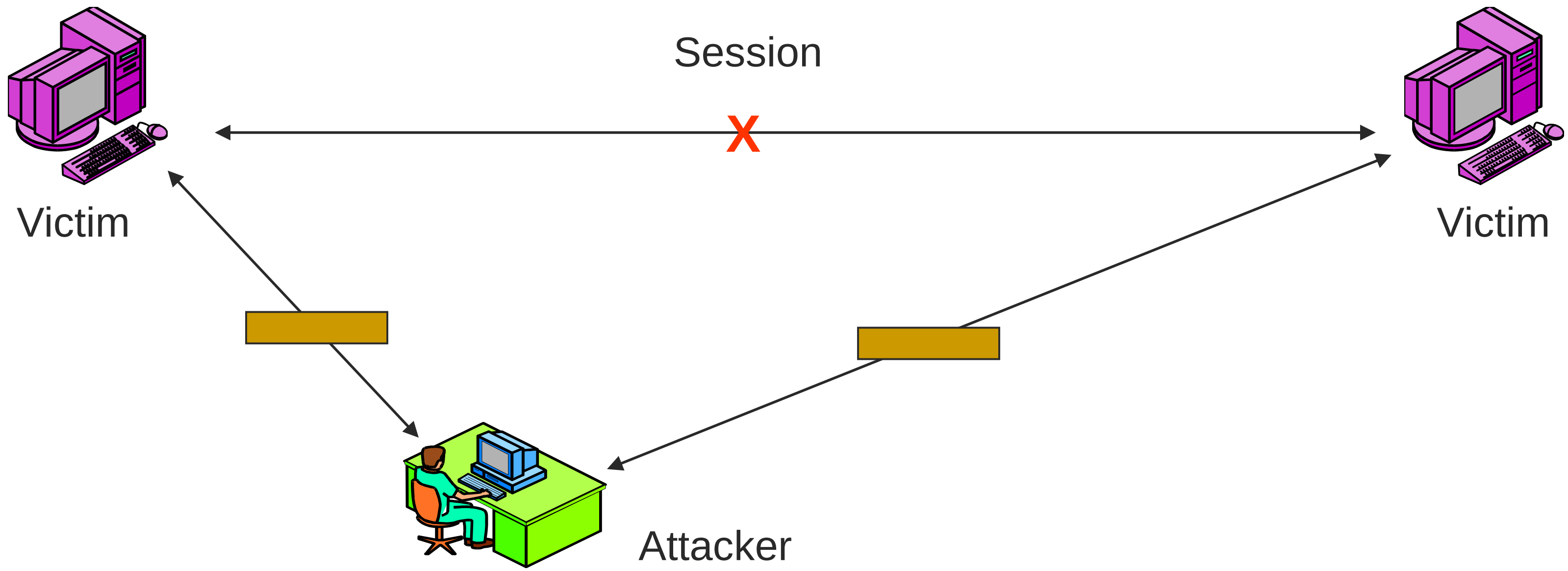
- If an attacker can determine possible sequence numbers used in another session, it is possible to:
  - Make client and server unsynchronized
  - Insert TCP segments into the session
  - The attacker may not see the result, but that might not be necessary
    - Change settings on a server, mail something, etc.

(C) Session Hijacking attack

- Take over an existing TCP session
- A DoS attack against the client makes it silent
- no DoS attack if aims at making client and server out of synchronization
- Difficult to do (must guess TCP Sequence Numbers), <u>today is rare</u>



(D) Man-in-the-middle attack



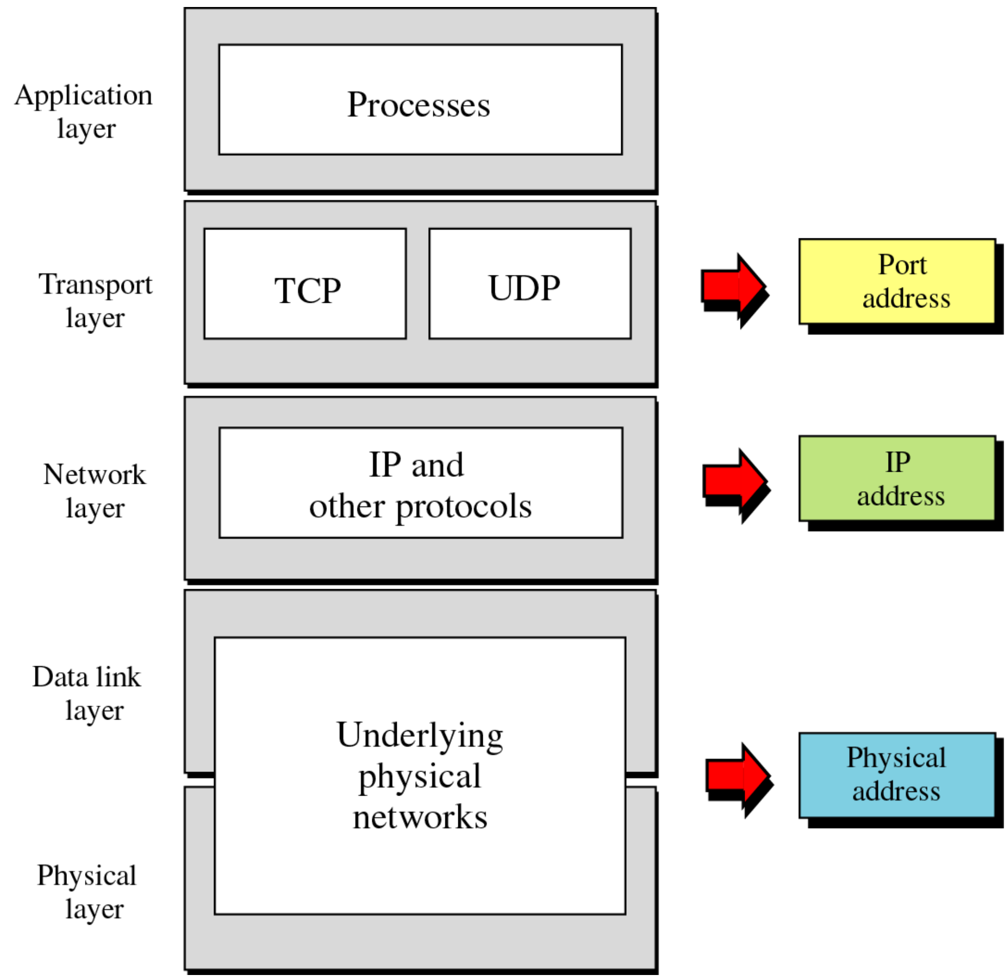
- Attacker is able to read, insert and modify messages between two parties without either party knowing that the link between them has been compromised
- The attacker must be able to observe and intercept messages going between the two victims
- Defence: authentication techniques (Lec 5)

Network Interface

- Device to connect a computer to a network
  - Ethernet card
  - WiFi adapter
- A computer may have multiple network interfaces
- Packets transmitted between network interfaces
- Most local area networks, (including Ethernet and WiFi) broadcast frames
- In regular mode, each network interface gets the frames intended for it
- Traffic sniffing can be accomplished by configuring the network interface to read all frames (promiscuous mode)

Types of Addresses

- Three different levels of addresses are used
- Each belongs to a specific layer in the system architecture



Address Resolution Protocol (ARP)

- ARP connects the network layer to the data link layer by converting IP addresses to MAC addresses
- ARP works by broadcasting requests and caching responses for future use
- The protocol begins with a computer broadcasting a message of the form

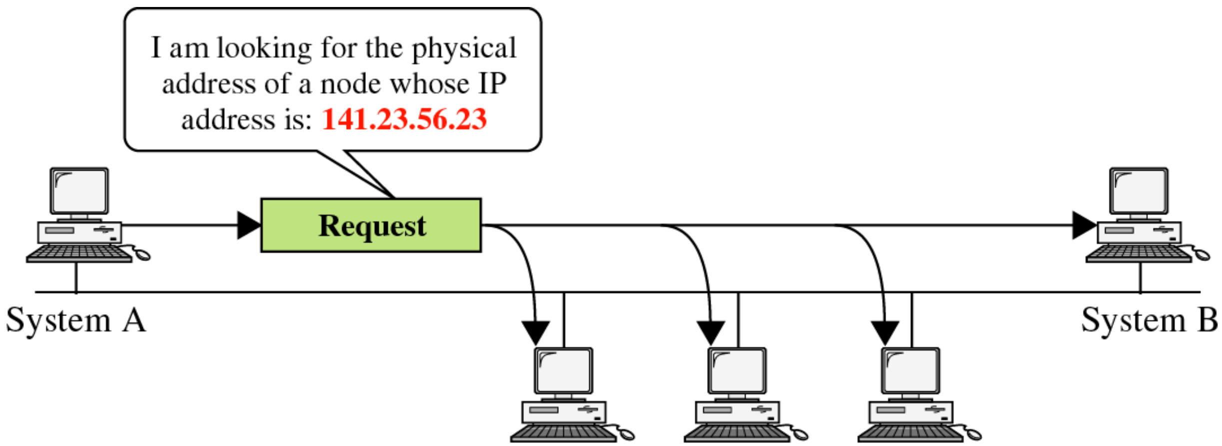
  who has <IP address1> tell <IP address2>

- When the machine with <IP address1> or an ARP server receives this message, it replies

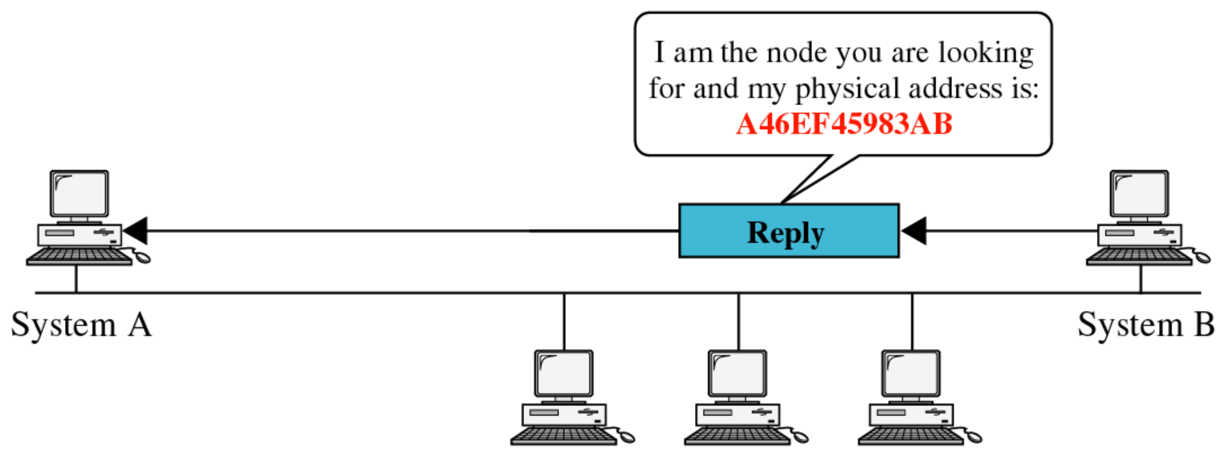
  <IP address1> is <MAC address>

Connecting Physical and Network addresses

How can a host find out the physical address of another host on the same network?



a. ARP request is broadcast



b. ARP reply is unicast

ARP table

- In Windows the command arp displays the ARP table

```
C:\>arp -a

Interface: 130.236.145.189 --- 0xc
  Internet Address      Physical Address      Type
  130.236.145.129       cc-ef-48-84-f4-c0     dynamic
  130.236.145.191       ff-ff-ff-ff-ff-ff     static
  224.0.0.2             01-00-5e-00-00-02     static
  224.0.0.13            01-00-5e-00-00-0d     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

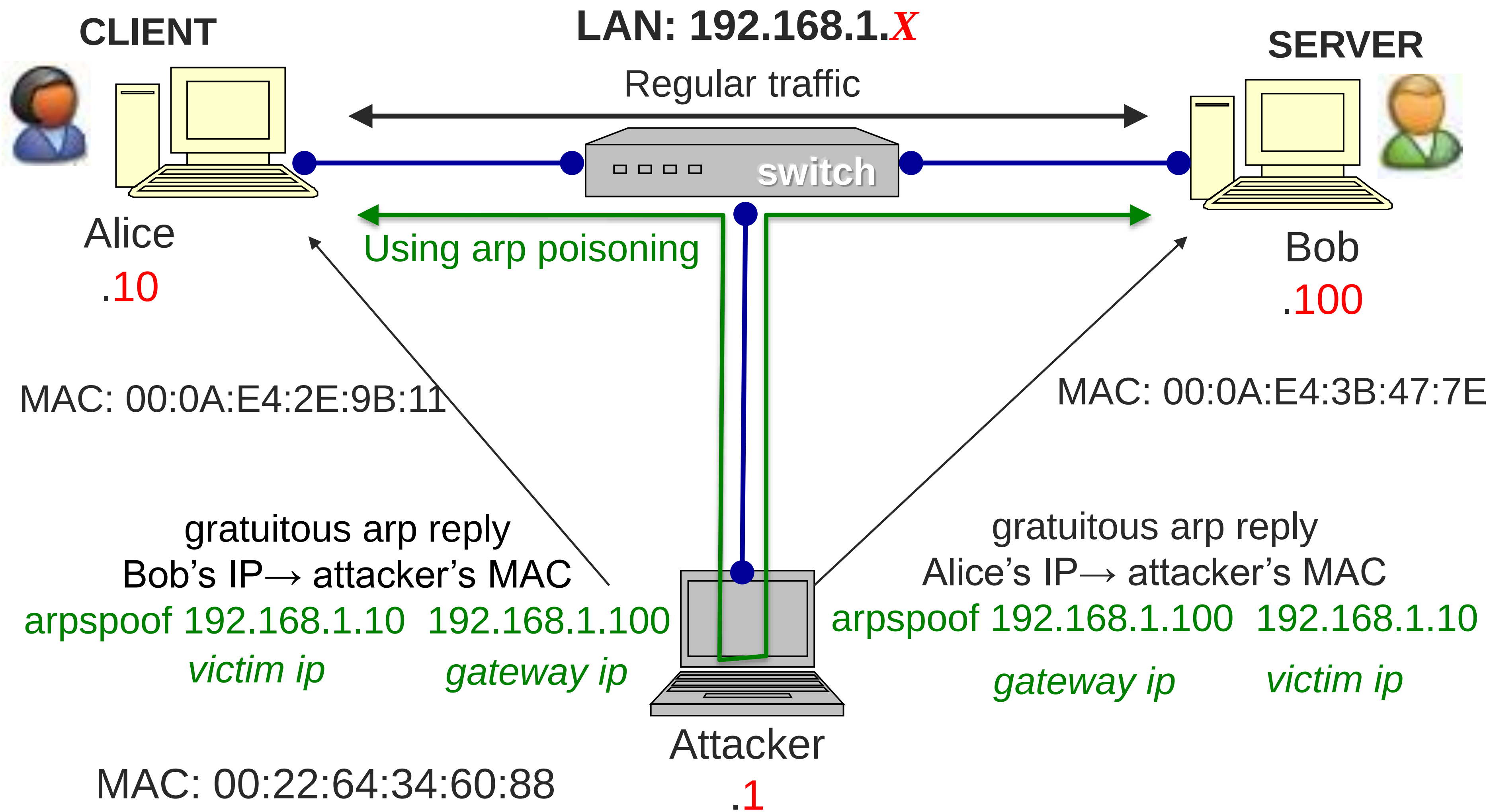
ARP spoofing attack

- Man-in-the-middle attack
- ARP table is updated whenever an ARP response is received
- Requests are not tracked
- ARP announcements are not authenticated
- Machines trust each other
- A malicious machine can spoof other machines

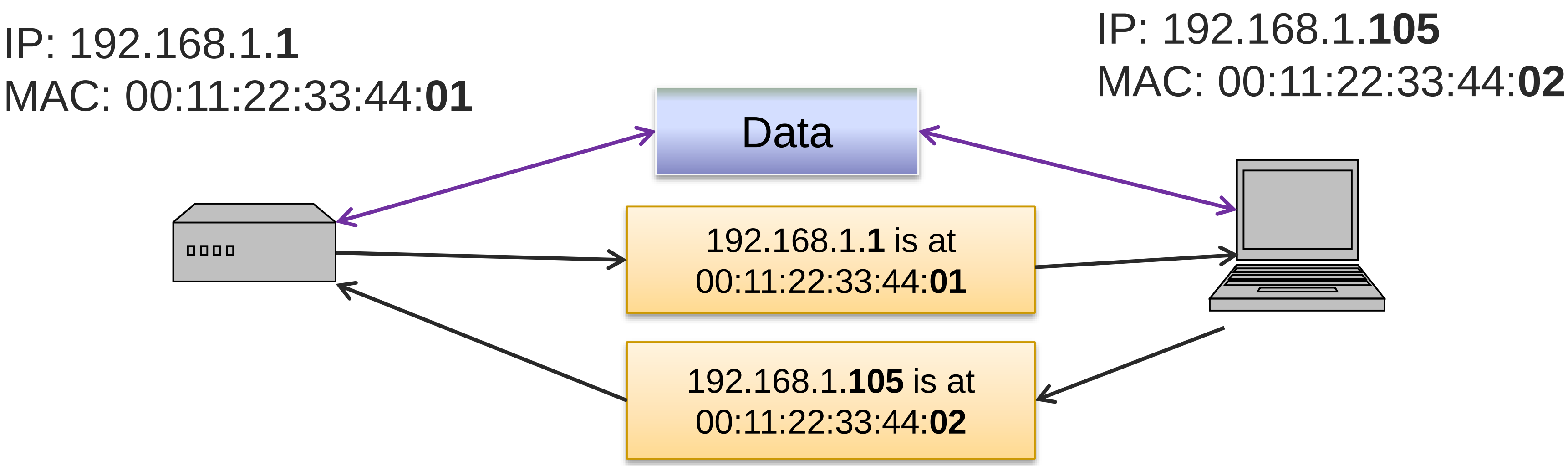
ARP spoofing attack

- According to the standard, almost all ARP implementations are stateless
- ARP table updates every time it receives an ARP reply
  - even if it did not send any ARP request
- It is possible to poison an ARP table by sending (unrequested) ARP replies
- Using static entries solves the problem

ARP spoofing attack



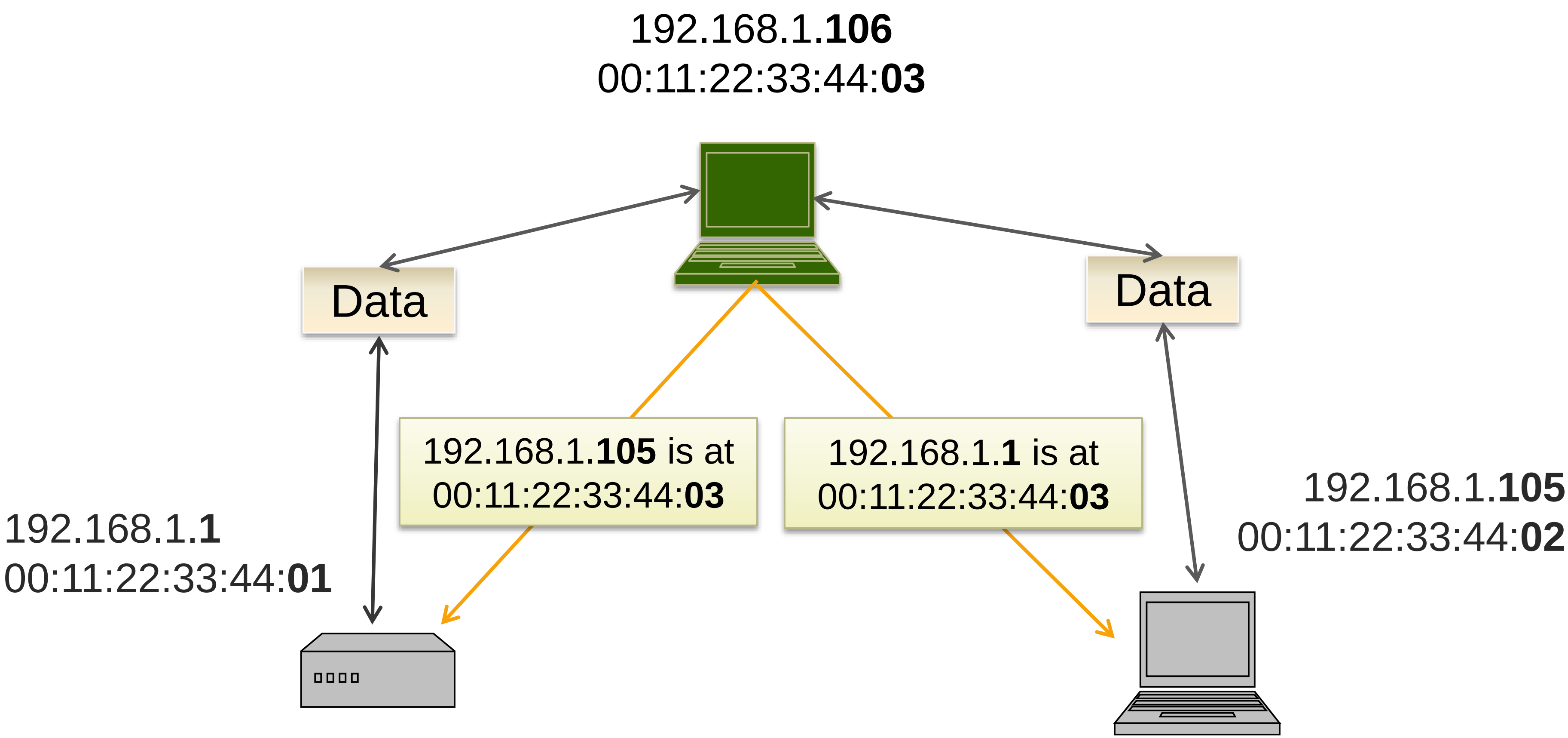
ARP spoofing attack



ARP Cache	
192.168.1.**105**	00:11:22:33:44:**02**

ARP Cache	
192.168.1.**1**	00:11:22:33:44:**01**

ARP spoofing attack



Poisoned ARP Cache	
192.168.1.**105**	00:11:22:33:44:**03**

Poisoned ARP Cache	
192.168.1.**1**	00:11:22:33:44:**03**

After knowing the target

- Now it’s time to break in
  - Known on victim’s computer:
    - OS
    - running applications
    - network structure may also be known
  - Known vulnerabilities are regularly published
  - Attackers can test and plan attacks off-line
    - Can set up similarly configured systems
    - Make sure it works the first time – without traces

After the break-in

- The attacker can weaken security:
  - Install rootkit and erase audit logs
  - Download password files
  - Create backdoors for re-entry if original hacking vulnerability is fixed
    - Backdoor accounts
    - Trojanized programs that permit re-entry

After the break-in

- Steal information, do damage
- Install software:
  - Spyware
  - Remote Administration Trojans (RATs)
  - Attack software to use against other hosts

Break-in attacks defenses

- If someone successfully breaks in
  - All software on all affected machines must be reinstalled
  - How do we know which machines are affected?
- Make sure all software is patched
  - OS, firewalls and applications
  - Harden hosts, disable unused network services

Break-in attacks defenses

- Run personal firewalls on all clients and servers
  - Can filter out many TCP and IP attacks
  - Remove strange options and fragmented packets before reach OS
  - See Home Network Security (Cert)
- Run IDSs to discover attacks

2. Denial-of-Service attacks

(DoS attacks)

DoS attacks

- Attempt to make a computer resource unavailable to its legitimate users
  - Crash a system or make it unavailable to others
- Types of DoS attacks
  - Single-message
  - Flooding
    - SYN flooding
    - Smurf flooding
    - Distributed DoS attack

Single-message DoS attacks

- Crash a host with a single attack packet (see Lec3)
  - Ping-of-Death
  - Teardrop
  - LAND
- Even firewalls can crash
- Send unusual input to applications
  - Common bug: buffer overflow

Buffer Overflow

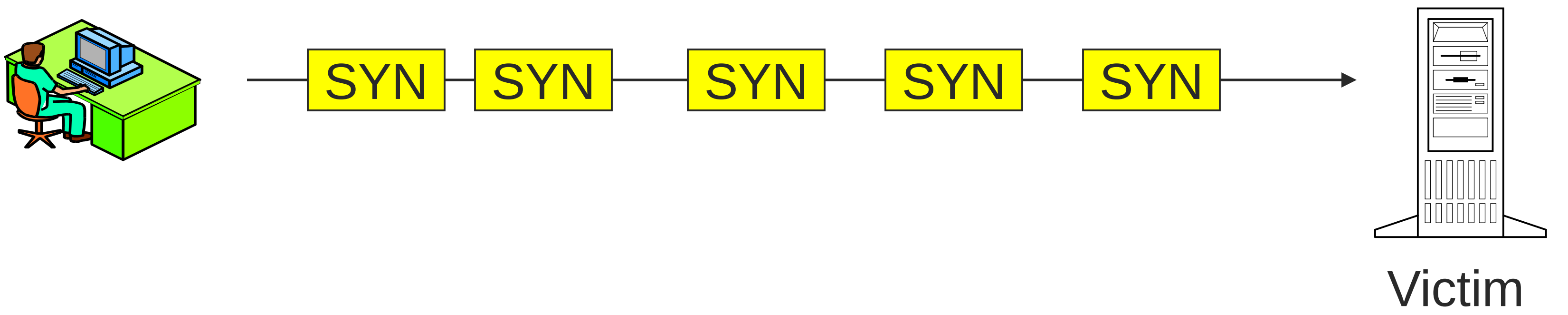
- Anomaly where a process designed to store data in a certain area of allocated memory (buffer) allows the caller to supply more data
  - extra data overwrites the process' own executable memory (out of buffer's bounds)
  - this may result in erratic program behavior:
    - memory access errors
    - incorrect results
    - program termination (crash)
- Can be triggered by inputs that are designed to execute code
  - they are the basis of many software vulnerabilities
  - bounds checking can prevent buffer overflows
  - programming languages associated with buffer overflows include C and C++

    do not automatically check that data written to arrays (out of bounds)
- Malware can force the system to execute malicious code by replacing legitimate code with its own payload of instructions copied into memory outside the buffer area

Flooding DoS attacks

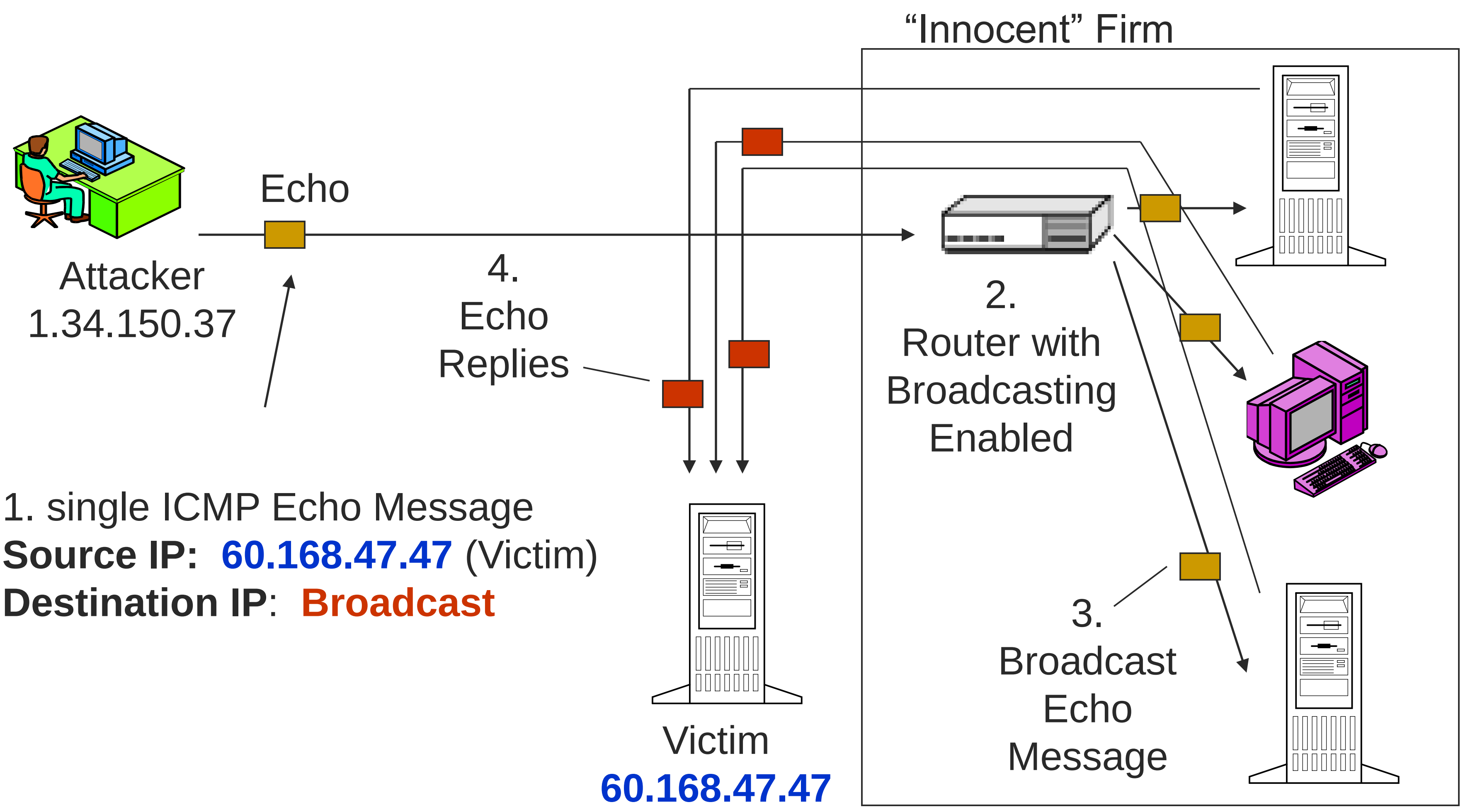
- Overload a host with many messages to make the host crash or very busy
- SYN flooding attack
  - Try to open many connections with SYN segments
  - Victim must prepare to work with many connections
  - Victim crashes if runs out of resources (at least slows down)
  - More computationally expensive for the victim than the attacker

SYN Flooding attack



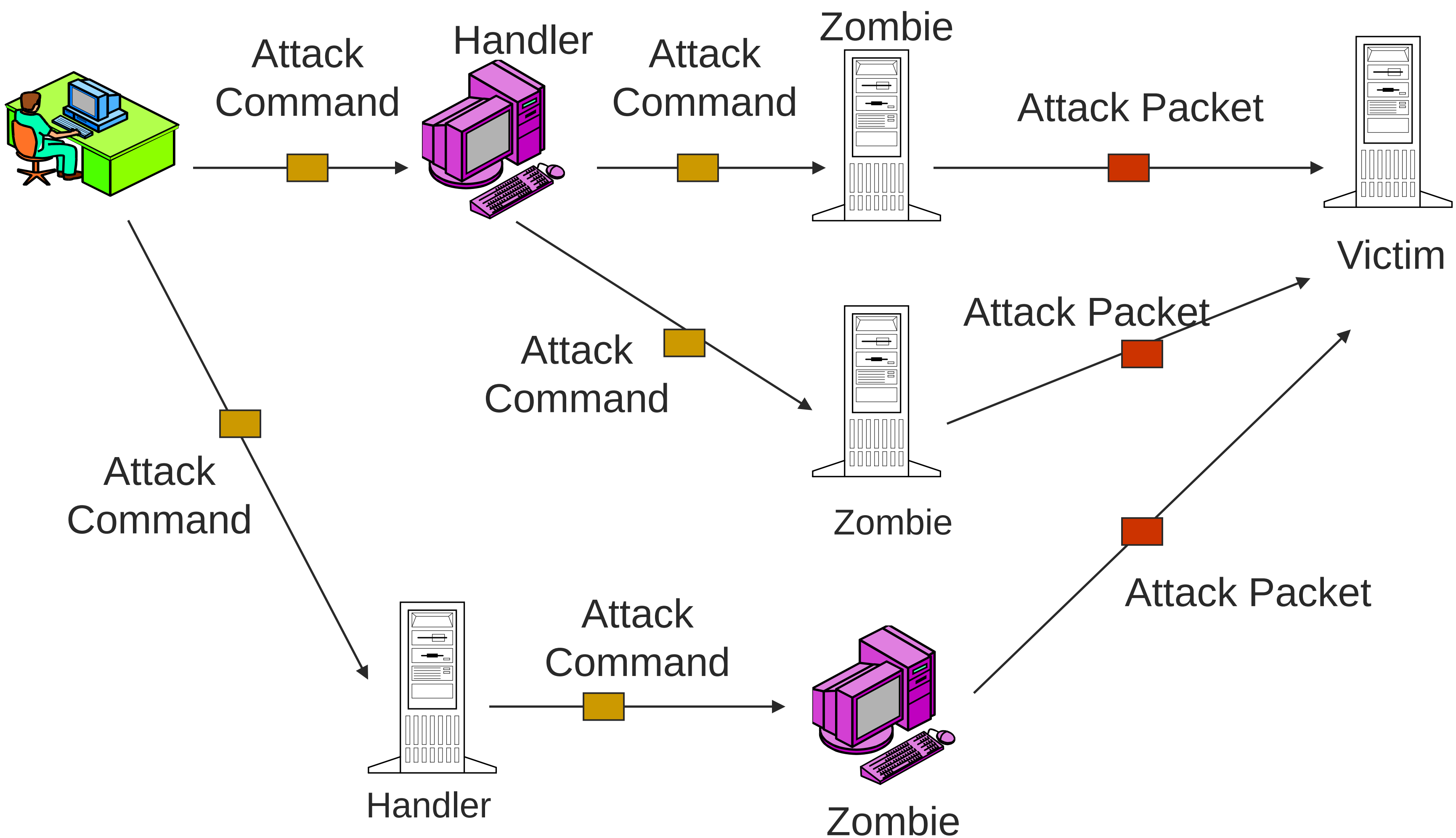
- attacker sends flood of SYN segments
- victim sets aside resources for each
- victim crashes or becomes too overloaded
  to respond to the SYN segments from legitimate users
- can it be solved by using SYN cookies?

Smurf Flooding attack



1. single ICMP Echo Message
Source IP: **60.168.47.47** (Victim)
Destination IP: **Broadcast**

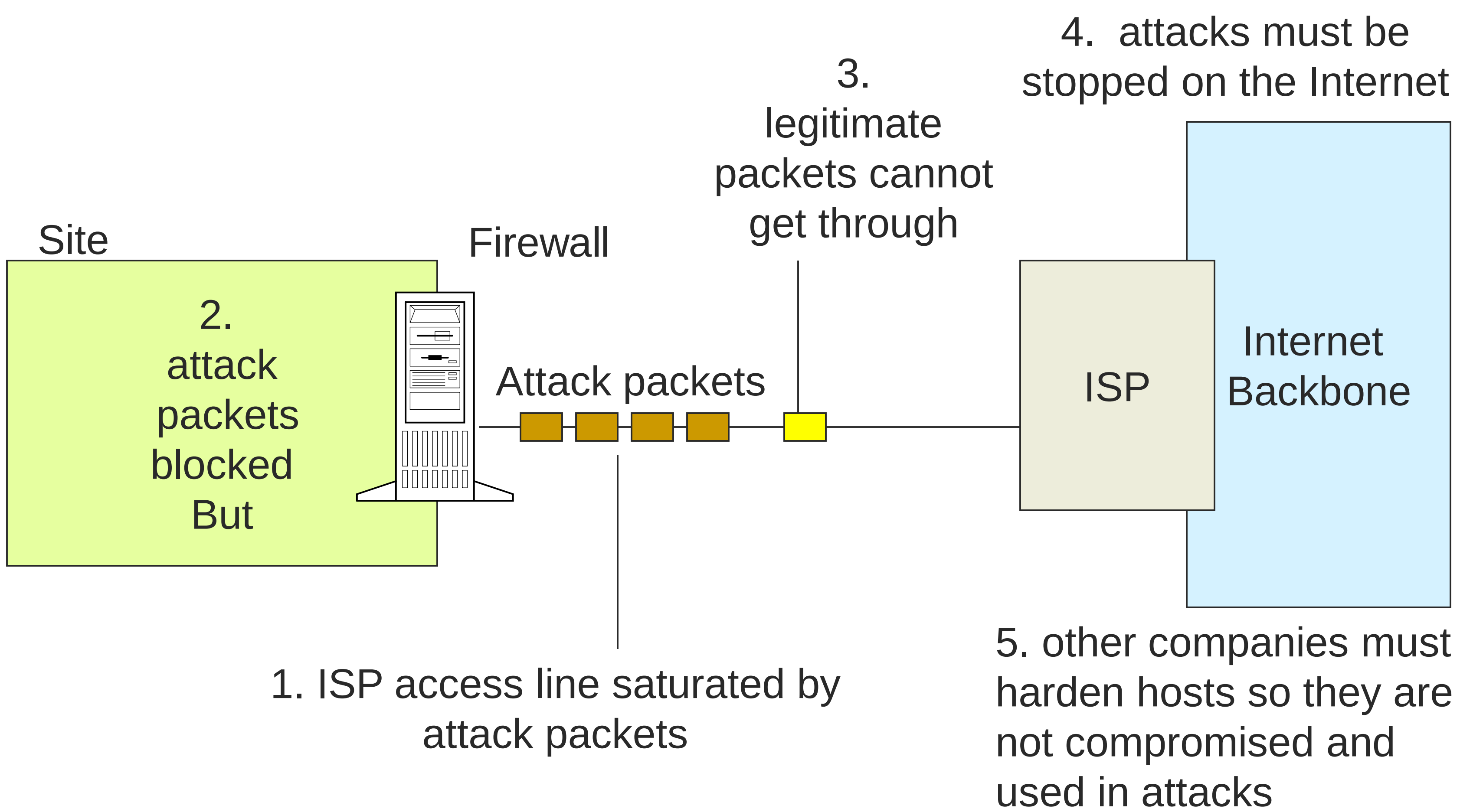
Distributed DoS attack



Stopping DoS attacks

- Ingress filtering to stop attack packets
  - Limited ability of ingress filtering because link to Internet Service Provider (ISP) might become overloaded
- Distributed DoS attacks are even harder
  - May involve lots of zombies all over the Internet
  - Can be hard to find them (e.g. false src IP addresses)
  - Requires cooperating from many companies and ISPs
- Egress filtering by companies or ISPs
  - Prevents src IP address spoofing
- Victim cannot do it alone $\rightarrow$ requires a community response

Stopping DoS attacks



ISP = internet service provider

DoS attacks defenses

- Always do ingress and egress filtering in border routers
  - Antispoofing rules on all interfaces
- Filter incoming ICMP messages
- Filter outgoing ICMP messages
  - Port/host/network unreachable (type 3)
- Never rely on IP addresses when authorizing connections
- Make firewalls identify port scans
  - Disable traffic from that host for a period of time (?)

DoS attacks defenses

- Disable IP Options (e.g. source routing)
- Consider disabling fragmented IP packets
  - Always discard short fragments
- Run scanning tools against your own network