# Network Scans

Ramin Sadre

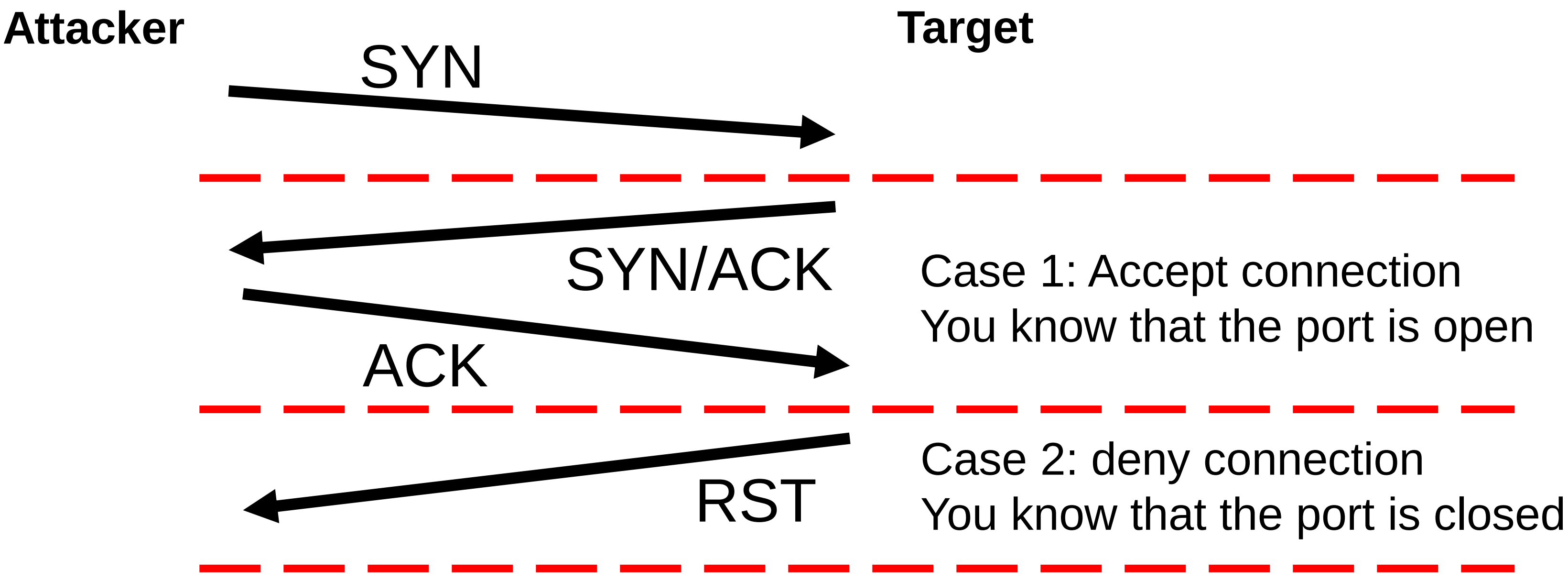
# Network Scans

- Scans are *information gathering attacks*:
  - Find vulnerable services/hosts
  - Discover network topology (used IP addresses,…)
  - System fingerprinting
  - …
- Scans are often performed as a preparation step for other attacks
- But sometimes also for legitimate reasons (research, network administration,…)!
- Can be combined with a "real" attack, e.g., a buffer overflow (Ping Of Death, 1997)
- Tool for scanning: `hping`, `nmap`, `zmap`, …
- Be careful and always ask first! Even a scan can crash the target

# Ping Sweeps

- Most simple scan:
  1. Send an ICMP echo request (“ping”) packet to the target IP address
  2. If you get an ICMP echo reply packet back, you know that the IP address is in use
  3. If the host does not exist, an intermediate router might also reply with a “host unreachable” ICMP message.
- Because ping sweeps are so easy to perform, network administrators often
  - configure hosts to ignore ICMP echo packets
  - configure their firewalls to block such packets
- So, it’s a simple but quite unreliable scan method

# TCP port scan with regular connections



Case 1: Accept connection
You know that the port is open

Case 2: deny connection
You know that the port is closed

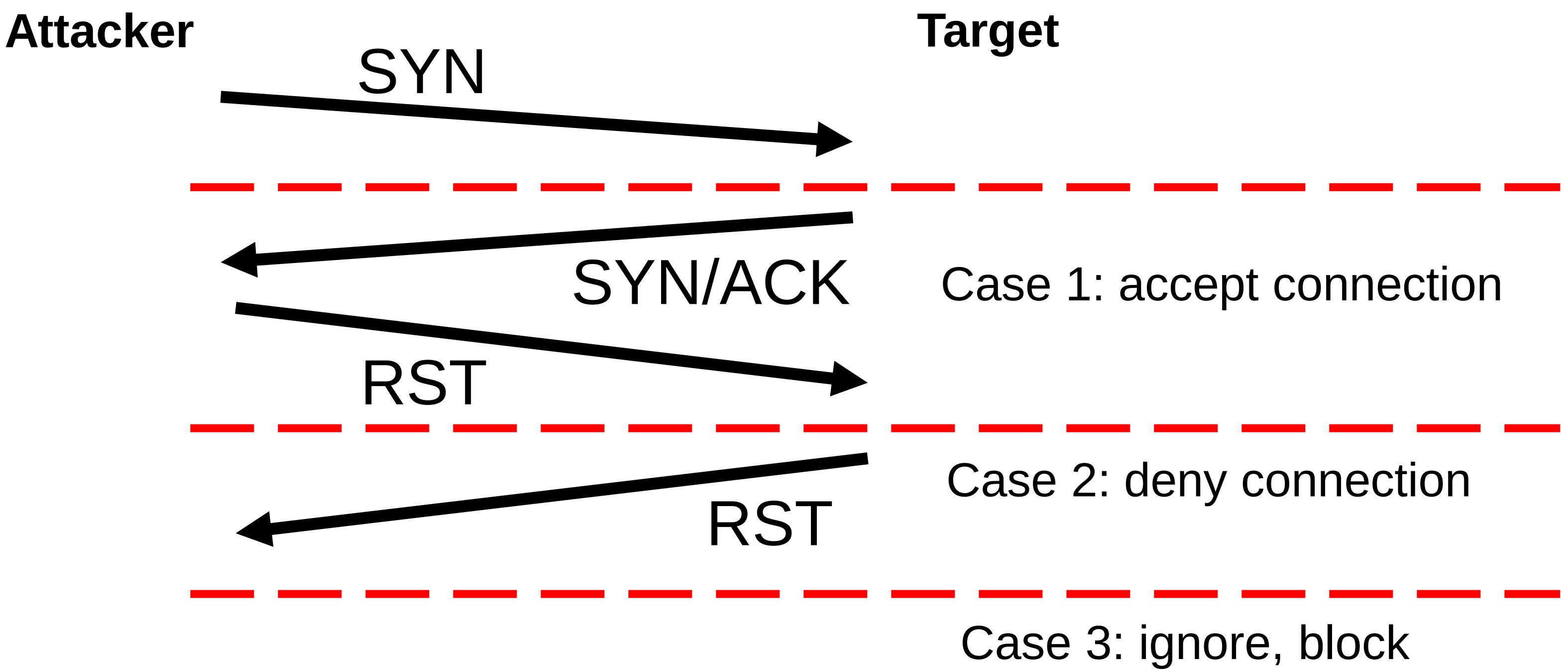
Case 3: ignore, block
You know nothing…

+ Easy to implement

– Slow

– Consumes resources (open connections) on the scanner host

# TCP port scan with SYN packets only



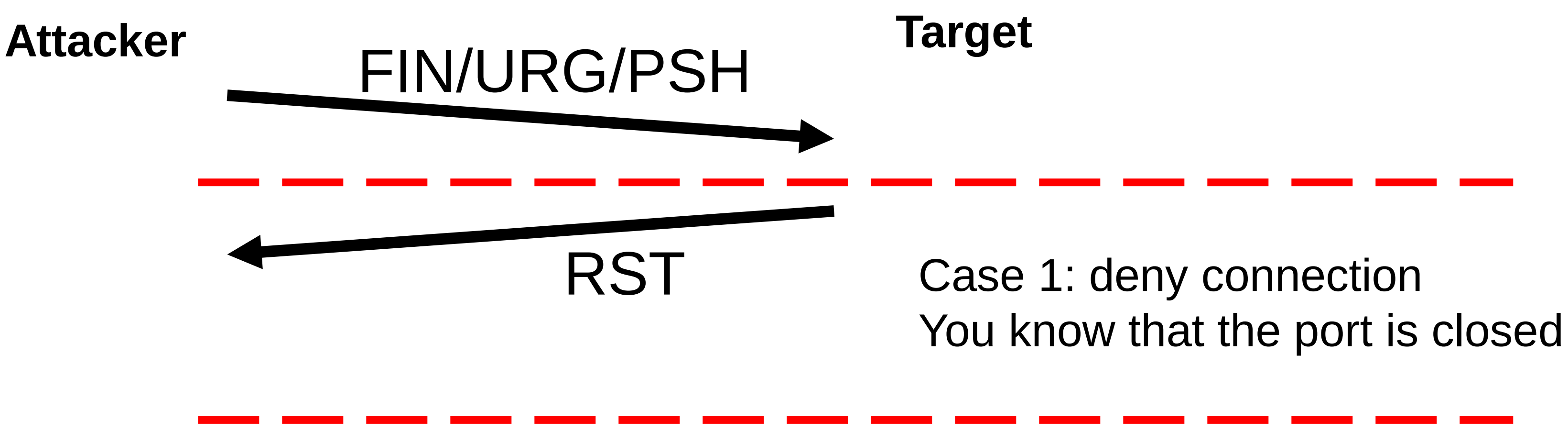
+ Fast

– Not supported by the OS. You have to write your own code (or use existing tools and libraries)

Example:

https://github.com/jgamblin/Mirai-Source-Code/blob/master/mirai/bot/scanner.c

# TCP port scan: Xmas-tree scan



Attacker

Target

FIN/URG/PSH

RST

Case 1: deny connection
You know that the port is closed

Case 2: ignore, block
You know that the port is open
or that the packet was blocked

# UDP port scan

- UDP is connectionless, so the TCP approach does not work
- Two approaches: Send packet to target port and …

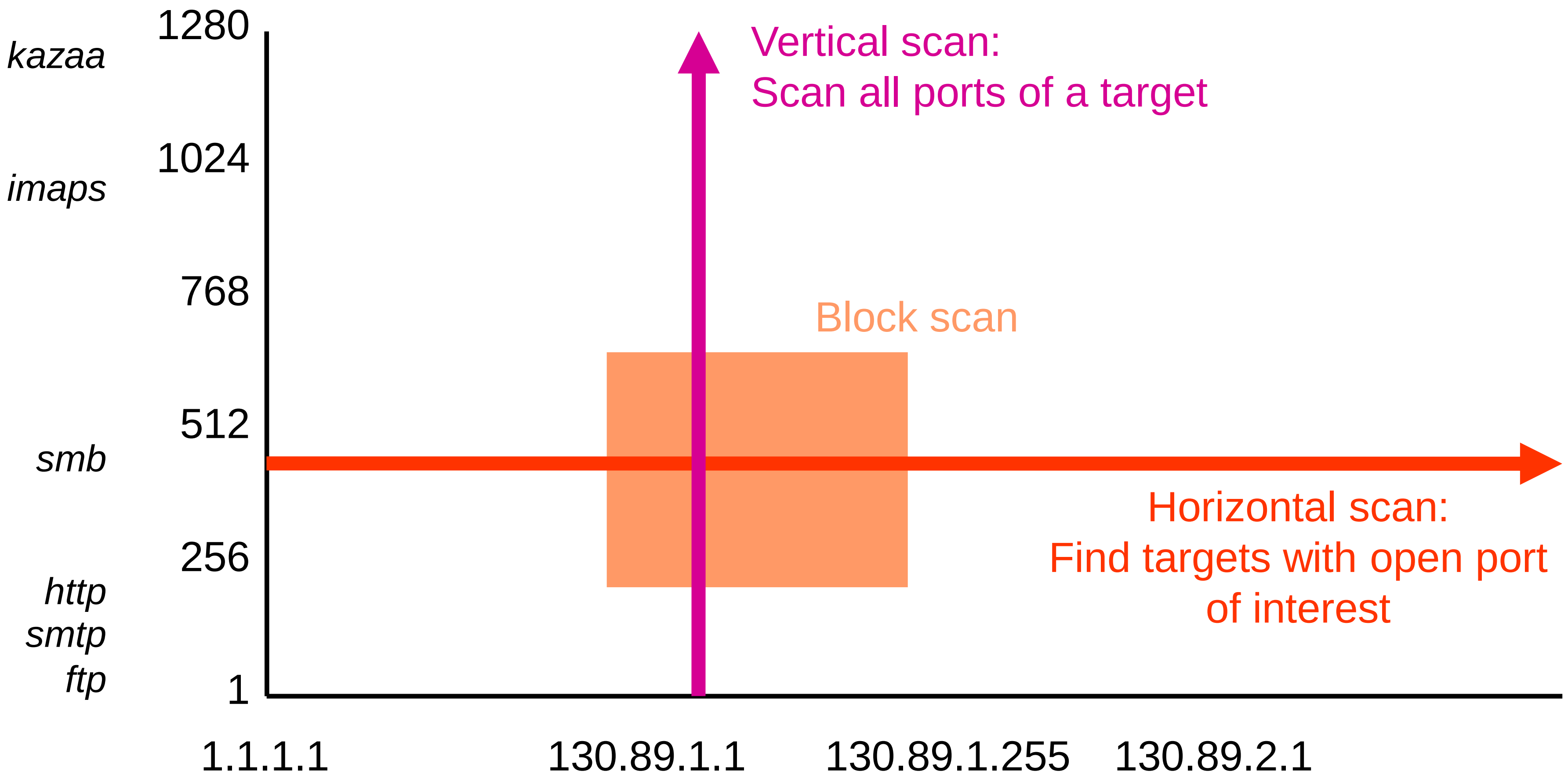
1. … wait for negative answer: If the UDP port is not open, the target will send an ICMP message “port unreachable”
2. … wait for positive answer
   Example: send DNS query to port 53 and wait for DNS response

+ Easy to implement

– Not very reliable because UDP packets might be lost

– For approach 1: ICMP might be disabled for security reasons

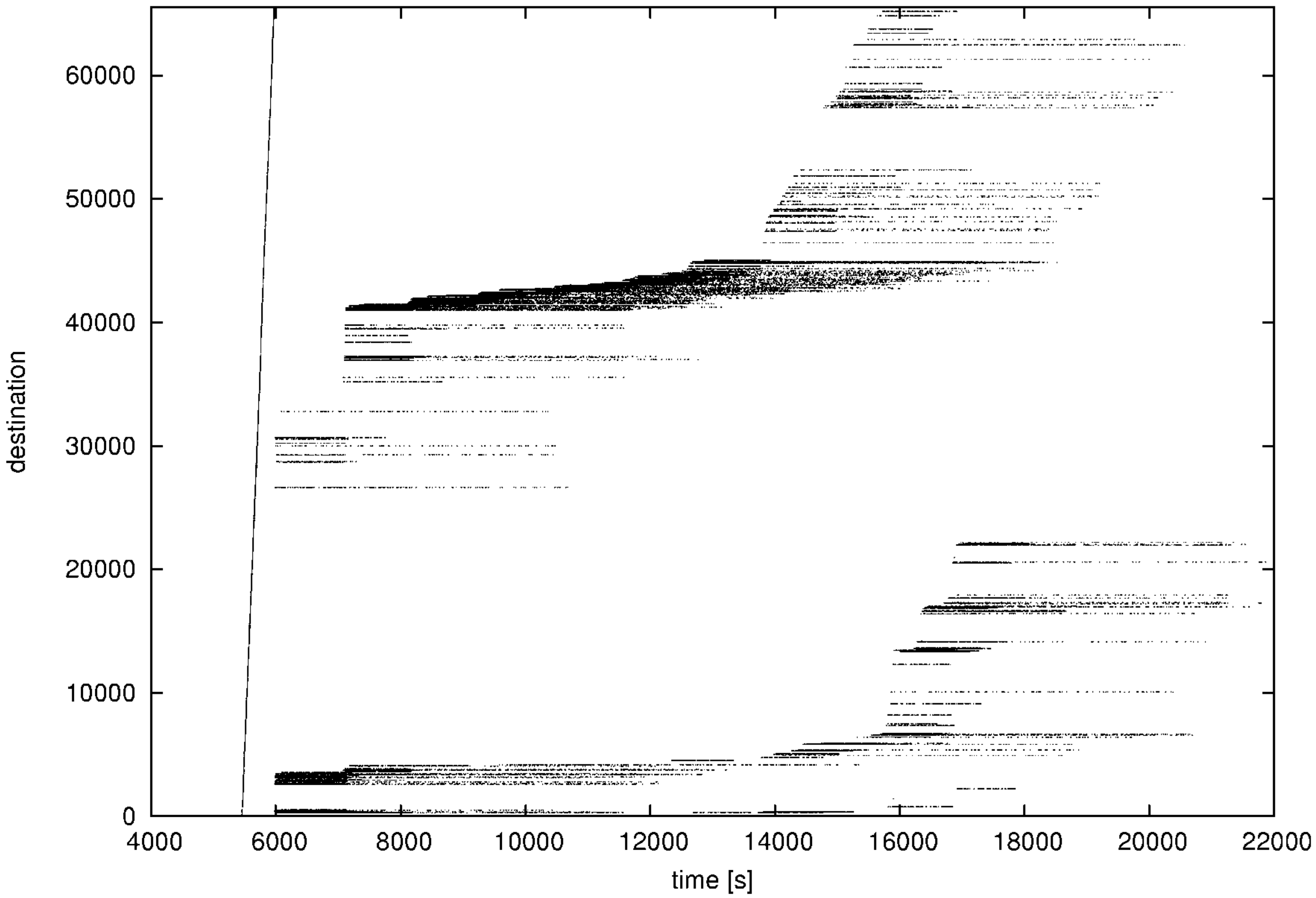
# Types


kazaa
imaps
smb
http
smtp
ftp
1280
1024
768
512
256
1
Vertical scan:
Scan all ports of a target
Block scan
Horizontal scan:
Find targets with open port of interest
1.1.1.1
130.89.1.1
130.89.1.255
130.89.2.1

# Remark

- The scans shown here have as goal to find open UDP and TCP ports
- Scans are also possible for other protocols
  - Example: a HTTP server
    The attacker can try different URLs to see what web applications are running on the server
  - Example: Smart Home automation
    The attack can send different commands (“open door”, “switch on light”,...) to the automation server to see what hardware has been installed
  - ...

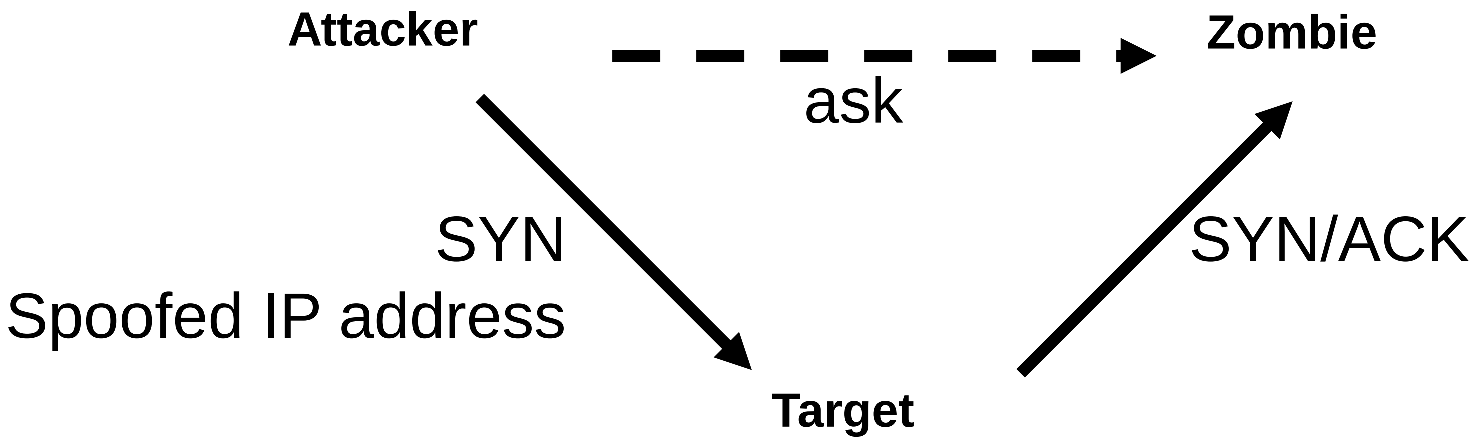
# Example: SSH attacker



# How to hide: Obfuscation

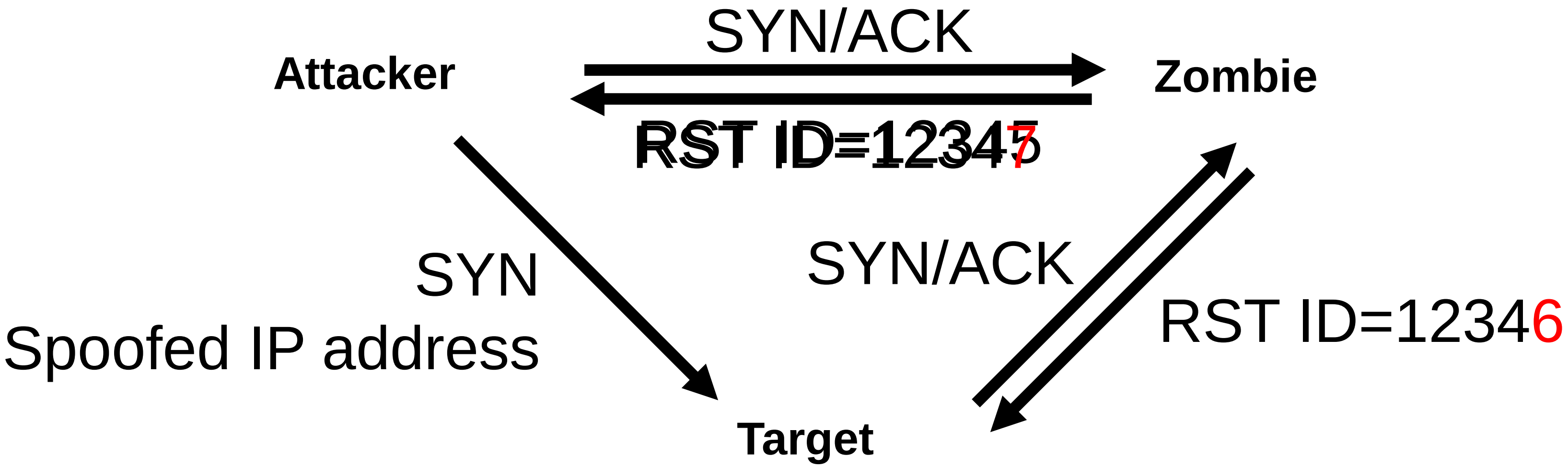
- The target system knows your IP address!
- How you can avoid to be detected/blocked by automatic systems or by human administrators?
- Obfuscation:
  - *Slow scan*: Scan very slowly. Most firewalls have automatic filters based on thresholds (we will see that later)
  - *Distributed scan*: Scan from multiple locations
  - *Indirect scan*: idle scan (1998),...
  - ...

# Idle scan



- How to ask the zombie?
- Fragment ID field in IP header

# Idle scan



Attacker

SYN/ACK

Zombie

RST ID=12345

RST ID=12347

SYN
Spoofed IP address

SYN/ACK

RST ID=12346

Target

# Implementing Idle scan

- Nowadays, many (but not all) operating systems randomize the ID field → difficult to find a zombie

- Implementations of idle scan can be found in various tools, for example nmap:

  https://nmap.org/book/idlescan.html