Politecnico di Milano
Dip. Elettronica e Informazione
Milano, Italy

Network Protocol attack examples

Ing. Stefano Zanero

Recall: taxonomy of some attacks

- Denial of service – against availability
- Sniffing (abusive reading of packets) – against confidentiality
- Spoofing or hijacking (forging packets "similar" to legitimate ones) – against integrity/authenticity

- Most of them can happen at network protocol level, in particular at the lowest levels
- We will see **examples**, not an exhaustive list!

Denial of Service examples

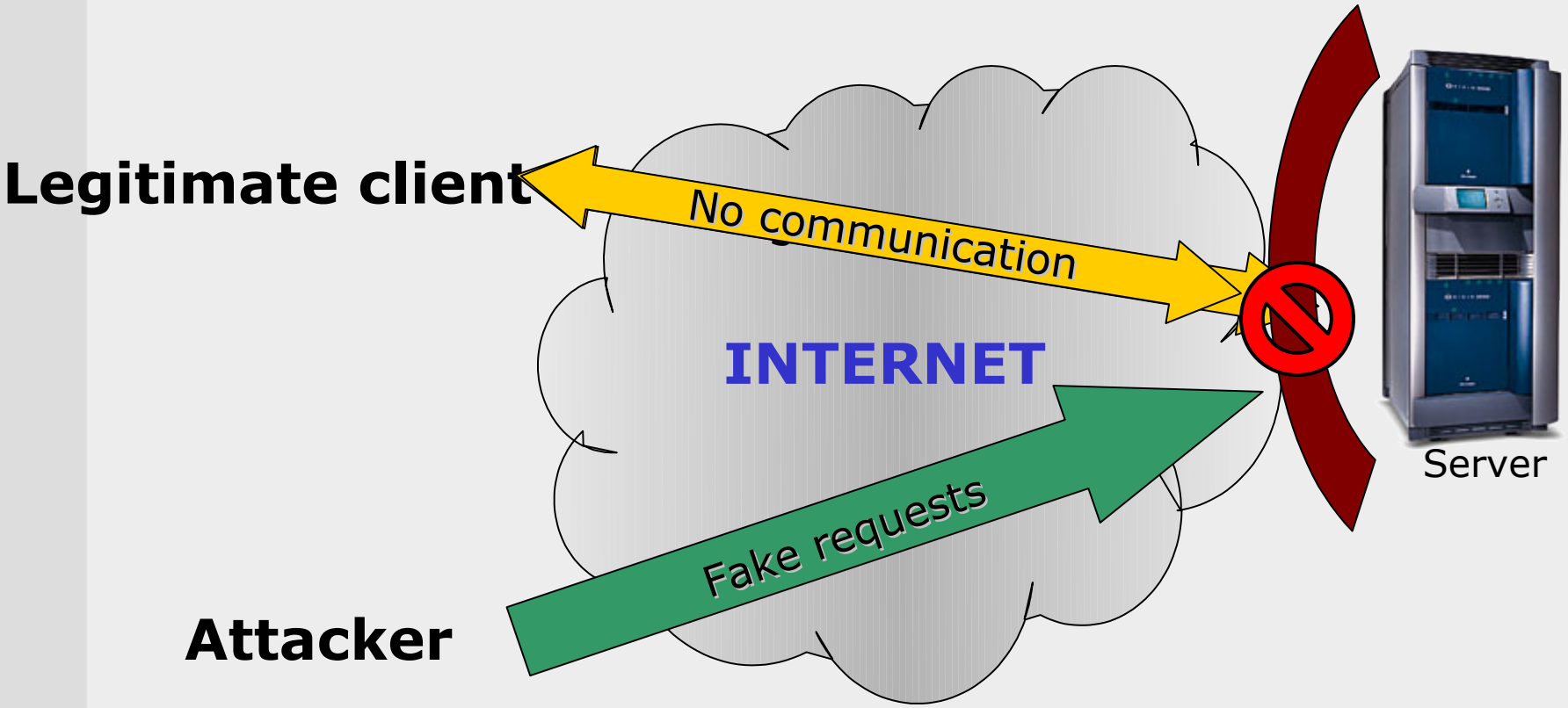
- "Killer packets"
- SYN flood
- Smurf/multiplication attacks
- Distributed DoS

Killer Packet: alcuni esempi

- Ping of Death
  - Pathological ICMP echo request
    - ping -l 65527 (Windows)
    - ping -s 65527 (UNIX)
- Teardrop
  - Fragmented packets with overlapping offsets
  - During reassembly “some OS” lock up with a funny BSOD
- Land
  - A long time ago, in a Windows 95 far, far away, a packet with src IP = dst IP and SYN flag set could loop and lock up a TCP/IP stack
  - Back to the future, same happened with SP2 in Windows XP
  - “This thing is like Dracula: it just won't stay dead”

Denial Of Service (DOS)

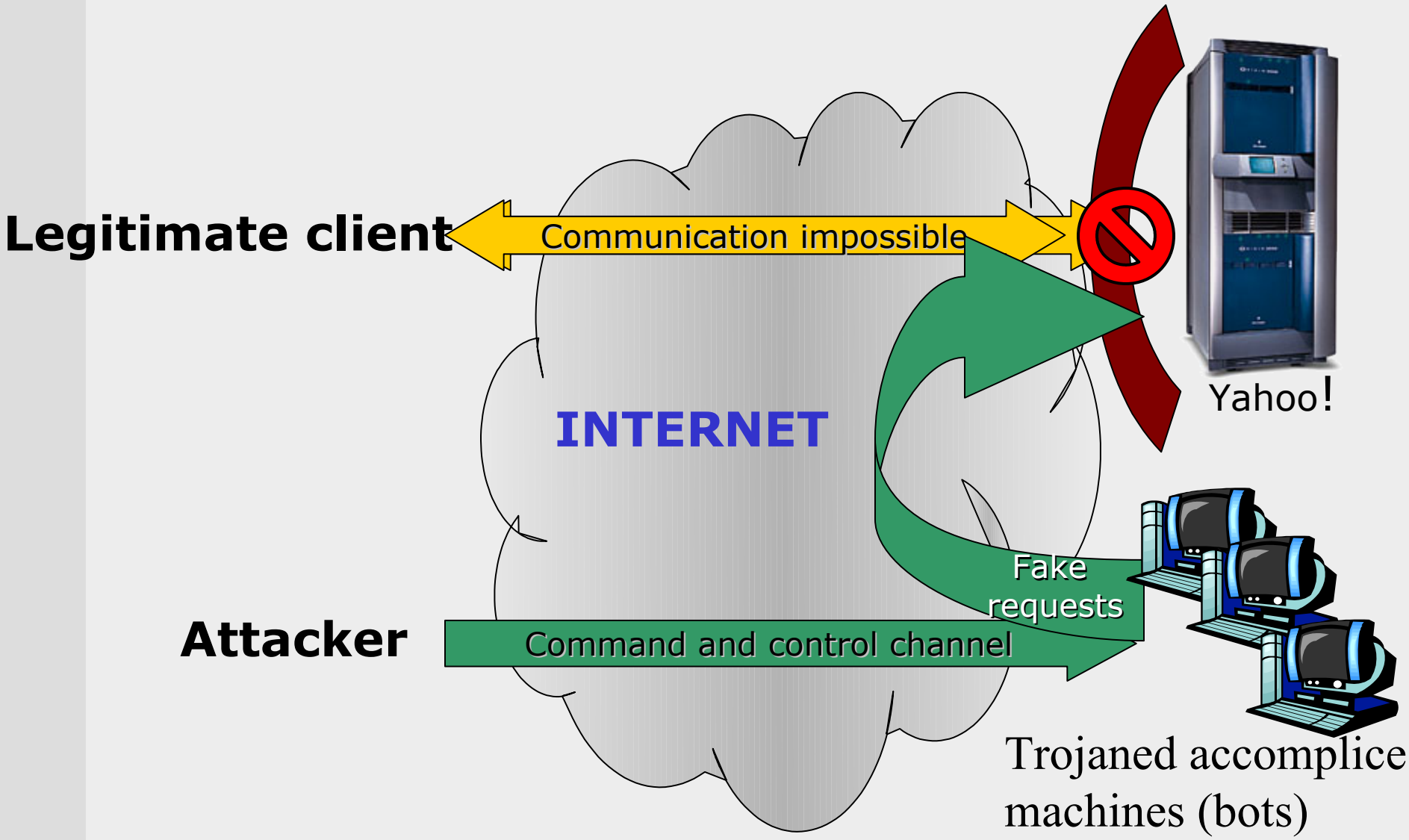
- Denial of Service via *flood*:



SYN Flood

- Attacker generates a high volume of SYN requests with spoofed (fake) src address (back to this in a few slides)
- Half-open queue in the TCP/IP stack of the kernel saturates
- Victim begins to drop SYN reqs from legitimate clients
- Modern stacks employ mechanisms such as SYN-cookies to avoid this

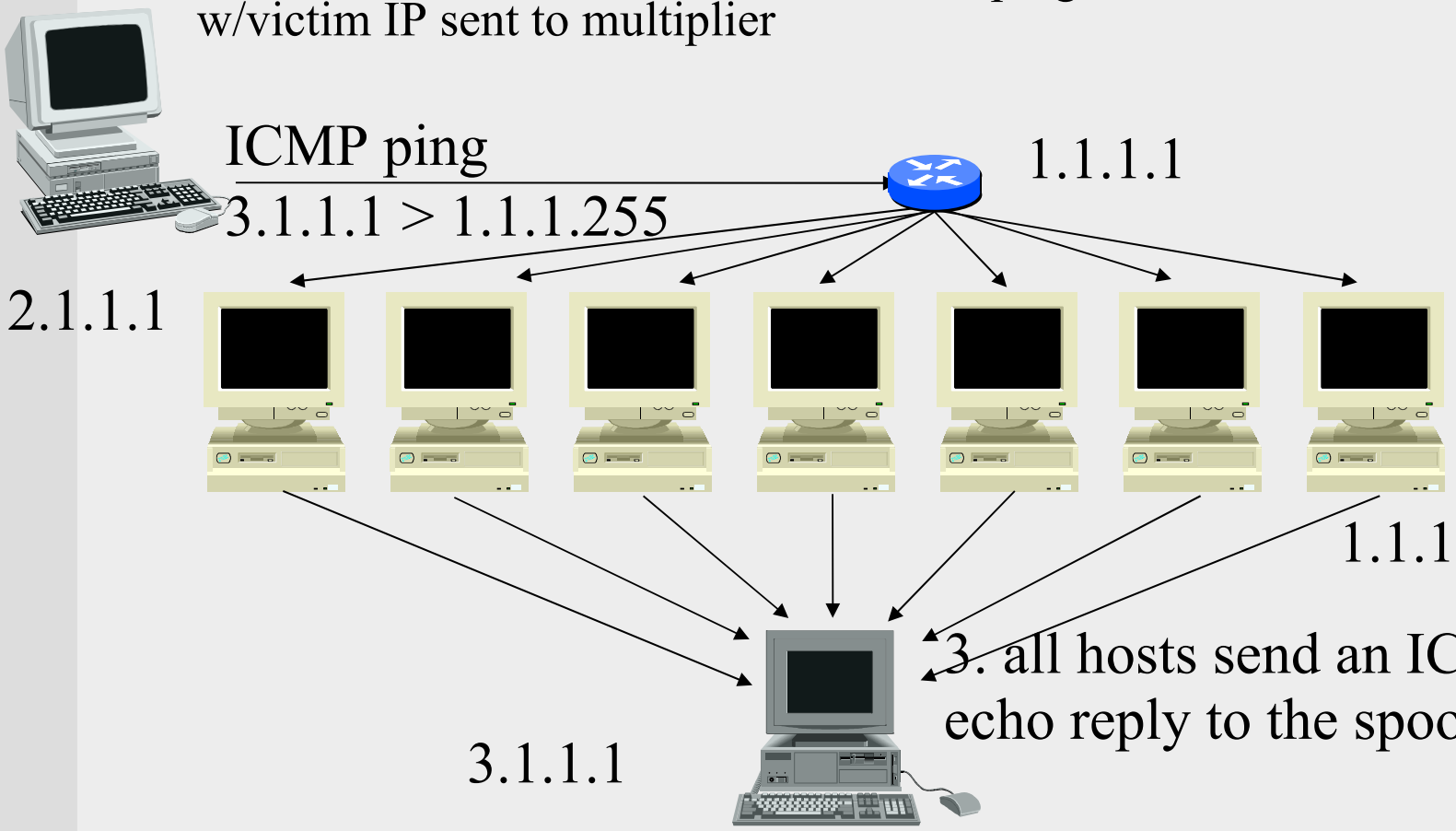
Variant: a DDOS



Smurf: a multiplier attack

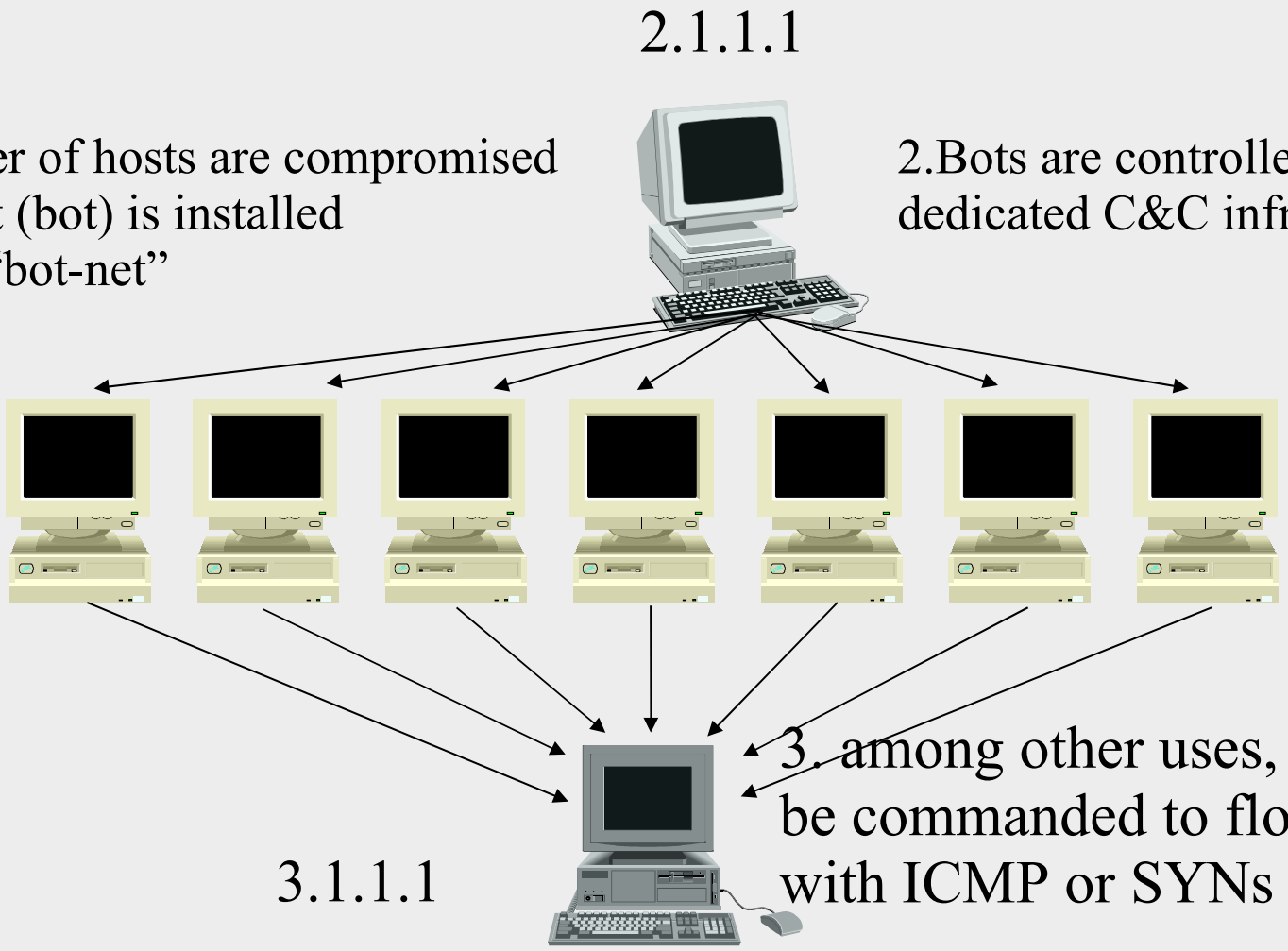
1. ICMP echo request with spoofed source w/victim IP sent to multiplier

2. Multiplier = a router able to ping network broadcast



Note: nowadays Smurf is less of a danger because of safer default router configurations

Botnet case


2.1.1.1
1. A number of hosts are compromised and a client (bot) is installed creating a “bot-net”
2.Bots are controlled using a dedicated C&C infrastructure
3. among other uses, they can be commanded to flood a target with ICMP or SYNs
3.1.1.1

Sniffing

- Ordinarily, a NIC will pass to the host OS only traffic with dst set to the host
- Promiscuous mode = telling the NIC to pass on any traffic on the wire
- **Sniffing**
- Partial solution: switched network, as opposed to hub network
  - Switch relays on a cable only traffic needed
- DSniff: www.monkey.org/~dugsong/dsniff
  - ARP spoofing
  - MAC flooding
  - Sniffing

ARP Spoofing

- ARP, Address Resolution Protocol, maps 32-bit logical Ipv4 addresses to 48-bit HW ETH addresses
- Two messages:
  - ARP request
  - ARP reply
- First come first trusted :(
- If an attacker sends a spoofed ARP reply, any host receiving it first will believe it
- Replies are usually stored in an ARP cache for performance reasons
- Unsolicited replies may be stored in cache, improving performance and worsening security

Arpspoofing…

```
[root@sconvery-lnx dsniff-2.3]# ./arpspoof 15.1.1.1
0:4:43:f2:d8:1 ff:ff:ff:ff:ff:ff 0806 42: arp reply
15.1.1.1 is-at 0:4:4e:f2:d8:1
0:4:43:f2:d8:1 ff:ff:ff:ff:ff:ff 0806 42: arp reply
15.1.1.1 is-at 0:4:4e:f2:d8:1
0:4:43:f2:d8:1 ff:ff:ff:ff:ff:ff 0806 42: arp reply
15.1.1.1 is-at 0:4:4e:f2:d8:1
0:4:43:f2:d8:1 ff:ff:ff:ff:ff:ff 0806 42: arp reply
15.1.1.1 is-at 0:4:4e:f2:d8:1
```

```
C:\>test

C:\>arp -d 15.1.1.1

C:\>ping -n 1 15.1.1.1

Pinging 15.1.1.1 with 32 bytes

Reply from 15.1.1.1: bytes=32 time<10ms TTL=255

C:\>arp -a

Interface: 15.1.1.26 on Interface 2
  Internet Address      Physical Address      Type
  15.1.1.1              00-10-83-34-29-72     dynamic
  15.1.1.25             00-04-4e-f2-d8-01     dynamic
C:\>arp -a

Interface: 15.1.1.26 on Interface 2
  Internet Address      Physical Address      Type
  15.1.1.1              00-04-4e-f2-d8-01     dynamic
  15.1.1.25             00-04-4e-f2-d8-01     dynamic
```

Filling up a CAM table

- Switches use CAM tables to know which MAC addresses are on which ports
- Typical CAM tables have 128k lines
- Dsniff (macof) can generate ~155k spoofed packets a minute: fills the CAM table in roughly 70 seconds from experience
- CAM table full: the switch cannot cache ARP responses any more and must forward everything to every port (like a hub does)
- Obviously, blinking light on the flooded port may give us away
- Port security features could also kill the attack

MAC Flooding

```
[root@sconvery-lnx dsniff-2.3]# ./macof
101.59.29.36 -> 60.171.137.91 TCP D=55934 S=322 Syn Seq=1210303300 Len=0 Win=512
145.123.46.9 -> 57.11.96.103 TCP D=44686 S=42409 Syn Seq=1106243396 Len=0 Win=52
109.40.136.24 -> 51.158.227.98 TCP D=59038 S=21289 Syn Seq=2039821840 Len=0 Win2
126.121.183.80 -> 151.241.231.59 TCP D=7519 S=34044 Syn Seq=310542747 Len=0 Win2
211.28.168.72 -> 91.247.223.23 TCP D=62807 S=53618 Syn Seq=2084851907 Len=0 Win2
183.159.196.56 -> 133.10.138.87 TCP D=23929 S=51034 Syn Seq=1263121444 Len=0 Wi2
19.113.88.77 -> 16.189.146.61 TCP D=1478 S=56820 Syn Seq=609596358 Len=0 Win=512
237.162.172.114 -> 51.32.8.36   TCP D=38433 S=31784 Syn Seq=410116516 Len=0 Win2
 118.34.90.6 -> 61.169.58.50 TCP D=42232 S=31424 Syn Seq=1070019027 Len=0 Win=52
46.205.246.13 -> 72.165.185.7 TCP D=56224 S=34492 Syn Seq=937536798 Len=0 Win=52
105.109.246.116 -> 252.233.209.72 TCP D=23840 S=45783 Syn Seq=1072699351 Len=0 2
60.244.56.84 -> 142.93.179.59 TCP D=3453 S=4112 Syn Seq=1964543236 Len=0 Win=512
151.126.212.86 -> 106.205.161.66 TCP D=12959 S=42911 Syn Seq=1028677526 Len=0 W2
9.121.248.84 -> 199.35.30.115 TCP D=33377 S=31735 Syn Seq=1395858847 Len=0 Win=2
226.216.132.20 -> 189.89.89.110 TCP D=26975 S=57485 Syn Seq=1783586857 Len=0 Wi2
124.54.134.104 -> 235.83.143.109 TCP D=23135 S=55908 Syn Seq=852982595 Len=0 Wi2
 27.54.72.62 -> 207.73.65.108 TCP D=54512 S=25534 Syn Seq=1571701185 Len=0 Win=2
246.109.199.72 -> 1.131.122.89 TCP D=61311 S=43891 Syn Seq=1443011876 Len=0 Win2
 251.49.6.89 -> 18.168.34.97 TCP D=25959 S=956 Syn Seq=6153014 Len=0 Win=512
51.105.154.55 -> 225.89.20.119 TCP D=33931 S=1893 Syn Seq=116924142 Len=0 Win=52
82.2.236.125 -> 210.40.246.122 TCP D=43954 S=49355 Syn Seq=1263650806 Len=0 Win2
21.221.14.15 -> 9.240.58.59  TCP D=61408 S=26921 Syn Seq=464123137 Len=0 Win=512
70.63.102.43 -> 69.88.108.26 TCP D=61968 S=53055 Syn Seq=682544782 Len=0 Win=512
```

STP protocol woes

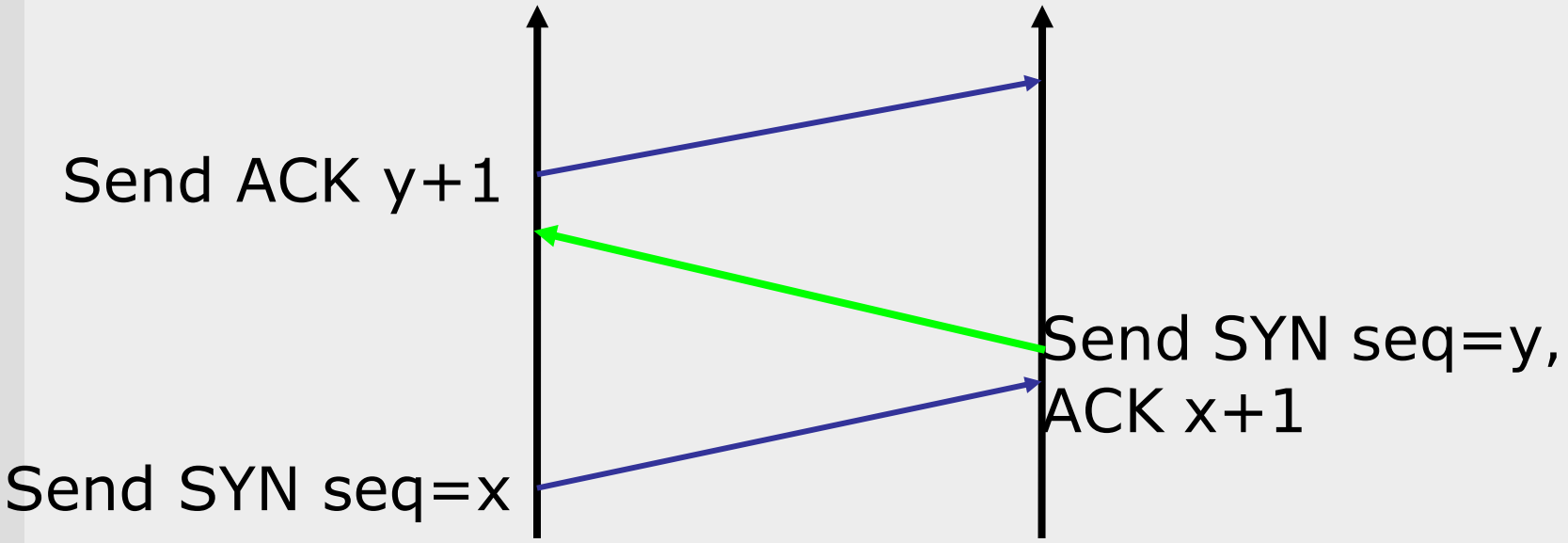
- STP: Spanning Tree Protocol (802.1d), a layer 2 protocol used to avoid loops on switched network
- A loop could cause a broadcast storm
- STP builds a ST by exchanging BPDU (bridge protocol data unit) packets
- Core idea: a broadcast received from the root is flooded to the children or viceversa but not both
- I need to elect a root switch to start building the tree. Guess what? BPDU are not authenticated
- Changing the shape of the tree may change my chances at ARP spoofing or sniffing

IP address spoofing

- IP src address is not authenticated
- So changing it for a UDP or ICMP packet is a piece of cake
- In general though, we will not see the answers to our packets, because they will be sent to the spoofed host (blind spoofing)
  - But if we are on the same network as the server or the spoofed client, we can sniff the rest
  - We can use source routing to have packets returned to us, or arp spoofing, or other tricks
- If we are blind, we must be able to predict answers
  - Which means we have a problem with TCP

TCP spoofing

Three-Way Handshake



TCP sequence guessing

- TCP connection use sequence numbers for reordering and acknowledging packets
- For each connection a semi-random initial sequence number (ISN) is chosen
- If a blind spoofer can predict the ISN, he can blindly complete the 3-way handshake without seeing the answers
- However, the spoofed source needs not to receive the response packets, otherwise it might answer with a RST and spoil the fun

TCP Session Hijacking

- Session Hijacking: taking over an active TCP session
- If I can sniff the packets, it is easy:
  - C follows the conversation of A and B recording the sequence numbers
  - C somehow disrupts B's connection (e.g. SYN Flood): B sees only a “random” disruption of service
  - C takes over the dialogue with A by spoofing B address and starting with a correct ISN. A suspects nothing
- hunt/dsniff implement this automatically
- I can avoid disrupting B's session and just inject things in the flow only if I am a MITM and can control/resync all the traffic flowing through

“Man in the middle”

- A wide-ranging cathegory comprising all the attacks where an attacker can impersonate the server wrt the client and vice-versa
- It can be
  - Physical or logical
  - Full or half-duplex (blind)
- E.g.: what happens if I'm able to arp-spoof the gateway of a LAN?

DNS poisoning

- If I intercept a DNS request, I can answer it spoofing the UDP packet, and the client will accept it (protocol not authenticated, once more)
- When a DNS server receives a request:
  - If it is authoritative for that domain, it answers
  - If not, if it cached the answer, it answers
  - If no answer in cache:
    - Recursion: resolves the name on behalf of the client
    - Iterative mode: gives the authoritative DNS address
- How to poison the cache of the DNS server?
  - Make a recursive query to the victim DNS
  - Spoof the answer of the authoritative DNS
  - Warning! In the forged answer we need to use the ID of the DNS transaction initiated by the victim... guess? bruteforce?

DHCP poisoning

- Once more, darling, with feeling
- DHCP not authenticated, blah blah
- Intercept request – first to answer – client will believe you
- I can set an IP address, DNS addresses, a default gateway...
- Like stealing candies from a baby...

ICMP redirect

- ICMP redirect: tells an host that a better route exists for a given destination, and gives the gateway for that route
- If we forge an ICMP redirect spoofing the src address as the gateway, and the best gateway as ourselves, we can trick the victim into sending us traffic
- We must sniff the original packet because ICMP redirect requests 64 bits + header IP of it in the packet as a weak "authentication" (and some OS – guess – do not even bother to check)
- Handling of ICMP redirect is OS-dependent
  - Windows 9x accepted them adding a temporary host entry in routing tables
  - Linux: default off, configured by value in /proc/sys/net/ipv4/<int>/accept_redirects

IRDP Poisoning

- IRDP (ICMP Router Discovery Protocol): ICMP based, automatic assignment of gateways
- Periodically, each router sends a multicast announcement with its IP address
- Non authenticated, etc, etc, usual stuff
- Advertisements have lifetime and priority: forge ones with high priority and long life
  - Windows 9x: accepted IRDP
  - Windows NT: used IRDP at boot
  - Windows 2k and later, Linux: ignore IRDP

Route mangling

- If I can announce routes to a router I can play a number of magical tricks
  - IGRP, RIP, OSPF: no or weak auth
  - EIGRP, BGP: authentication available but seldom used
- I can send routes with low metric and restrictive netmasks to have priority
- I can also play with multipath or QoS routing
- Static routes however usually have priority on dynamic ones