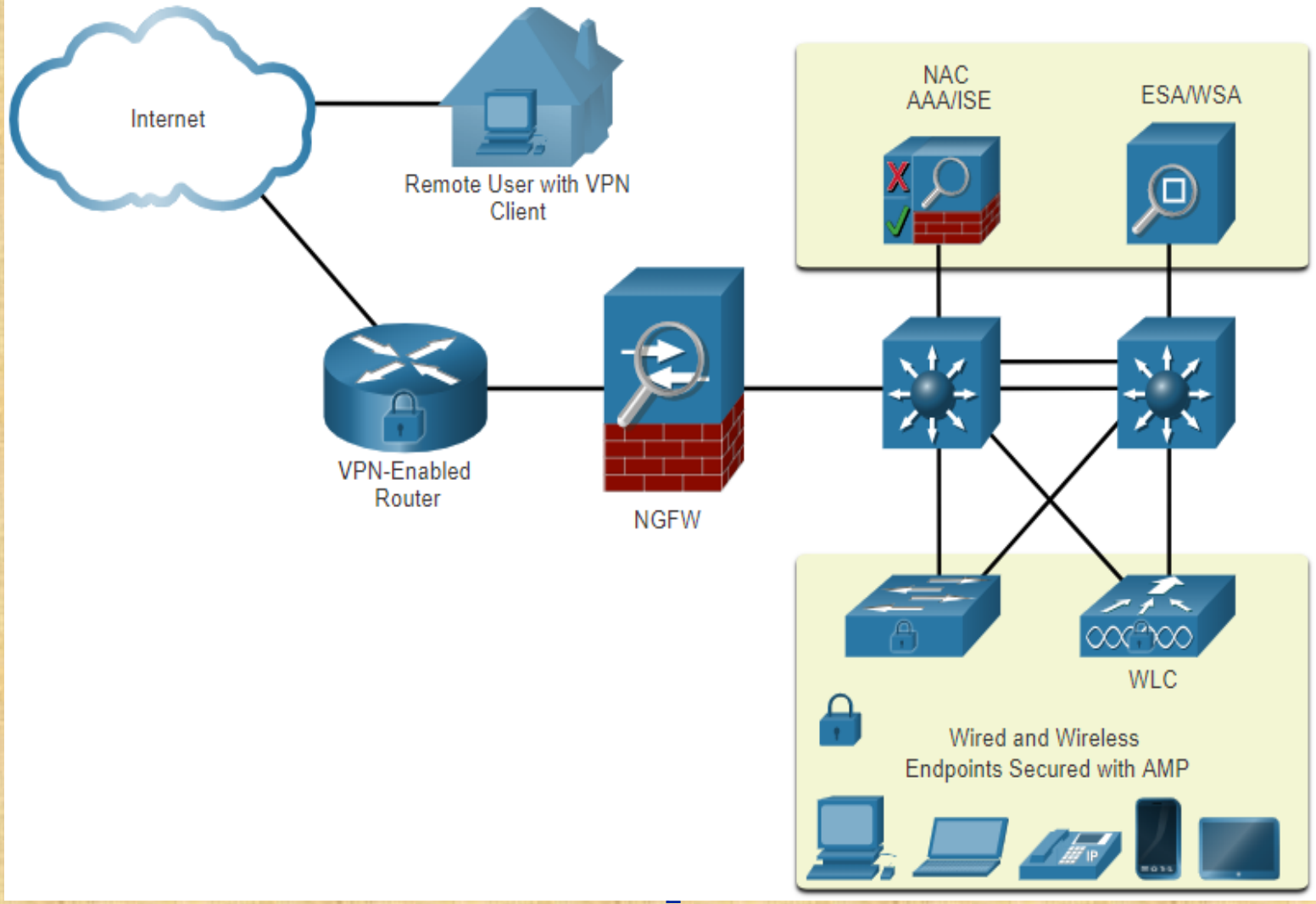
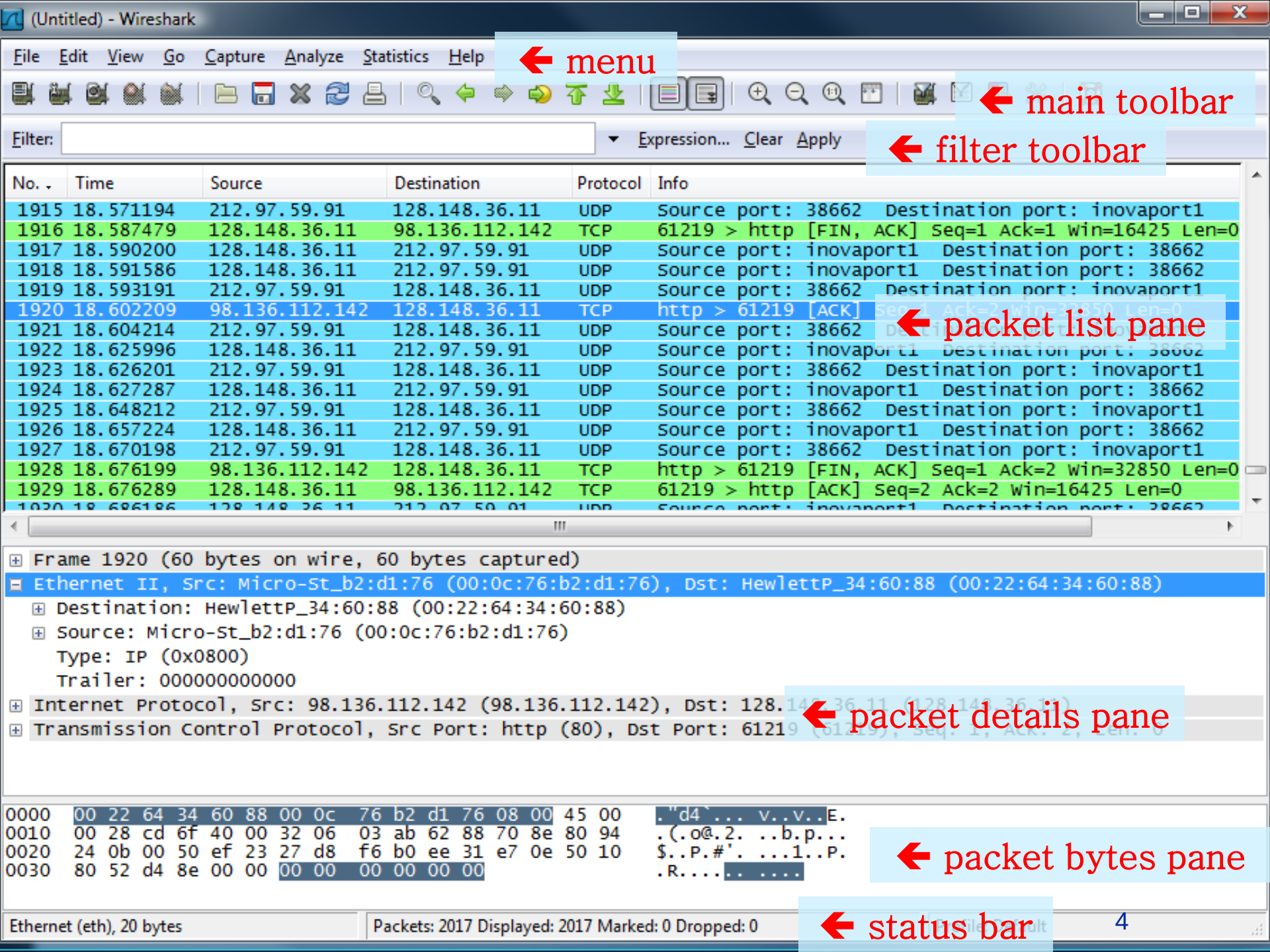
Chapter 5 : Network Security I

Network Security Products



Network Monitoring Tool: Wireshark

- Wireshark is a packet sniffer and protocol analyzer
  - Captures and analyzes frames
  - Supports plugins
- Usually required to run with administrator privileges
- Setting the network interface in promiscuous mode captures traffic across the entire LAN segment and not just frames addressed to the machine
- Freely available on www.wireshark.org

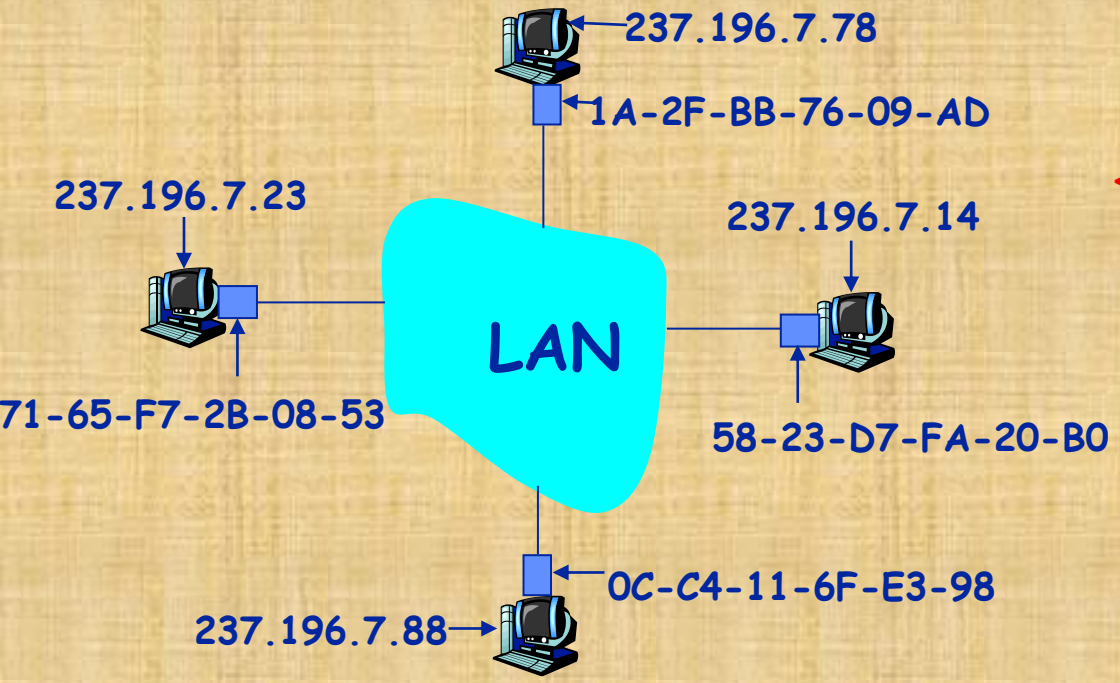

(Untitled) - Wireshark
File Edit View Go Capture Analyze Statistics Help
menu
main toolbar
Filter:
Expression... Clear Apply
filter toolbar
No. Time Source Destination Protocol Info
1915 18.571194 212.97.59.91 128.148.36.11 UDP Source port: 38662 Destination port: inovaport1
1916 18.587479 128.148.36.11 98.136.112.142 TCP 61219 > http [FIN, ACK] Seq=1 Ack=1 Win=16425 Len=0
1917 18.590200 128.148.36.11 212.97.59.91 UDP Source port: inovaport1 Destination port: 38662
1918 18.591586 128.148.36.11 212.97.59.91 UDP Source port: inovaport1 Destination port: 38662
1919 18.593191 212.97.59.91 128.148.36.11 UDP Source port: 38662
1920 18.602209 98.136.112.142 128.148.36.11 TCP http > 61219 [ACK]
packet list pane
1921 18.604214 212.97.59.91 128.148.36.11 UDP Source port: 38662
1922 18.625996 128.148.36.11 212.97.59.91 UDP Source port: inovaport1
1923 18.626201 212.97.59.91 128.148.36.11 UDP Source port: 38662 Destination port: inovaport1
1924 18.627287 128.148.36.11 212.97.59.91 UDP Source port: inovaport1 Destination port: 38662
1925 18.648212 212.97.59.91 128.148.36.11 UDP Source port: 38662 Destination port: inovaport1
1926 18.657224 128.148.36.11 212.97.59.91 UDP Source port: inovaport1 Destination port: 38662
1927 18.670198 212.97.59.91 128.148.36.11 UDP Source port: 38662 Destination port: inovaport1
1928 18.676199 98.136.112.142 128.148.36.11 TCP http > 61219 [FIN, ACK] Seq=1 Ack=2 Win=32850 Len=0
1929 18.676289 128.148.36.11 98.136.112.142 TCP 61219 > http [ACK] Seq=2 Ack=2 Win=16425 Len=0
Frame 1920 (60 bytes on wire, 60 bytes captured)
Ethernet II, Src: Micro-St_b2:d1:76 (00:0c:76:b2:d1:76), Dst: HewlettP_34:60:88 (00:22:64:34:60:88)
Destination: HewlettP_34:60:88 (00:22:64:34:60:88)
Source: Micro-St_b2:d1:76 (00:0c:76:b2:d1:76)
Type: IP (0x0800)
Trailer: 000000000000
Internet Protocol, Src: 98.136.112.142 (98.136.112.142), Dst: 128.
Transmission Control Protocol, Src Port: http (80), Dst Port: 6121
packet details pane
0000 00 22 64 34 60 88 00 0c 76 b2 d1 76 08 00 45 00 ."d4`... v..v..E.
0010 00 28 cd 6f 40 00 32 06 03 ab 62 88 70 8e 80 94 .(.o@.2. ..b.p...
0020 24 0b 00 50 ef 23 27 d8 f6 b0 ee 31 e7 0e 50 10 $..P.#'. ...1..P.
0030 80 52 d4 8e 00 00 00 00 00 00 00 00 .R...... ....
packet bytes pane
Ethernet (eth), 20 bytes
Packets: 2017 Displayed: 2017 Marked: 0 Dropped: 0
status bar
4

MAC Addresses and ARP

- 32-bit IP address:
  - *network-layer* address
  - used to get datagram to destination IP subnet
- MAC (or LAN or physical or Ethernet) address:
  - Data link layer address
  - used to get datagram from one interface to another physically-connected interface (same network)
  - 48 bit MAC address (for most LANs) burned in the adapter ROM
  - Some Network interface cards (NICs) can change their MAC

ARP: Address Resolution Protocol

Question: how to determine MAC address of host B when knowing B's IP address?



- Each IP node (Host, Router) on LAN has ARP table
- ARP Table: IP/MAC address mappings for some LAN nodes

  < IP address; MAC address; TTL>
  - TTL (Time To Live): time after which address mapping will be forgotten (typically 20 min)

ARP

- 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, its broadcasts the response

  <IP address1> is <MAC address>
- The requestor's IP address <IP address2> is contained in the link header
- The Linux and Windows command arp - a displays the ARP table

```
  Internet Address      Physical Address      Type
  128.148.31.1          00-00-0c-07-ac-00     dynamic
  128.148.31.15         00-0c-76-b2-d7-1d     dynamic
  128.148.31.71         00-0c-76-b2-d0-d2     dynamic
  128.148.31.75         00-0c-76-b2-d7-1d     dynamic
  128.148.31.102        00-22-0c-a3-e4-00     dynamic
  128.148.31.137        00-1d-92-b6-f1-a9     dynamic
```

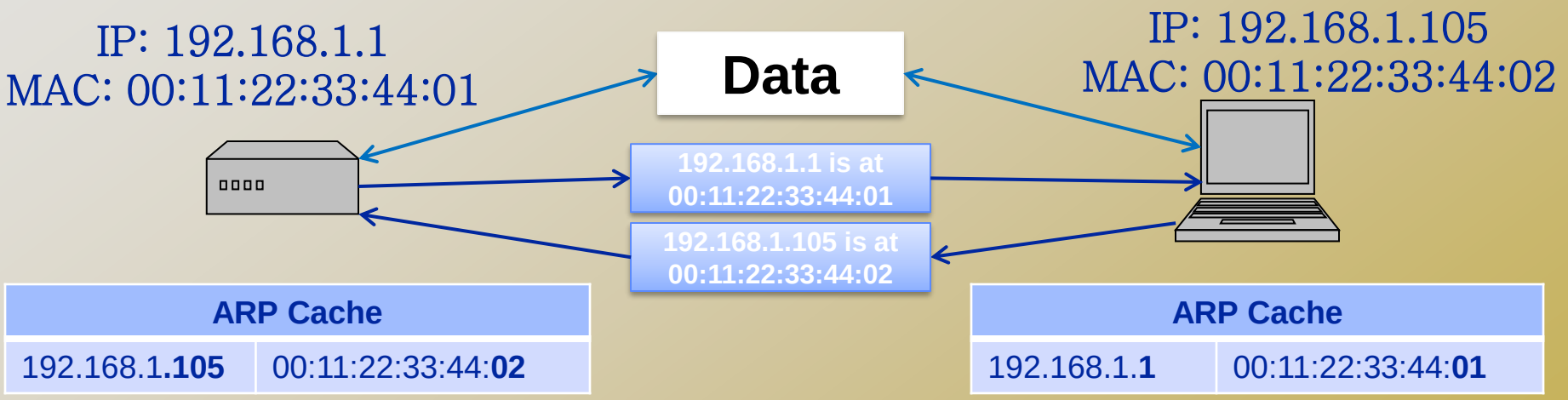
ARP Spoofing

- The ARP table is updated whenever an ARP response is received
- Requests are not tracked
- ARP announcements are not authenticated
- Machines trust each other
- A rogue machine can spoof other machines

ARP Poisoning (ARP Spoofing)

- According to the standard, almost all ARP implementations are stateless
- An arp cache updates every time that it receives an arp reply… even if it did not send any arp request!
- It is possible to "poison" an arp cache by sending gratuitous arp replies

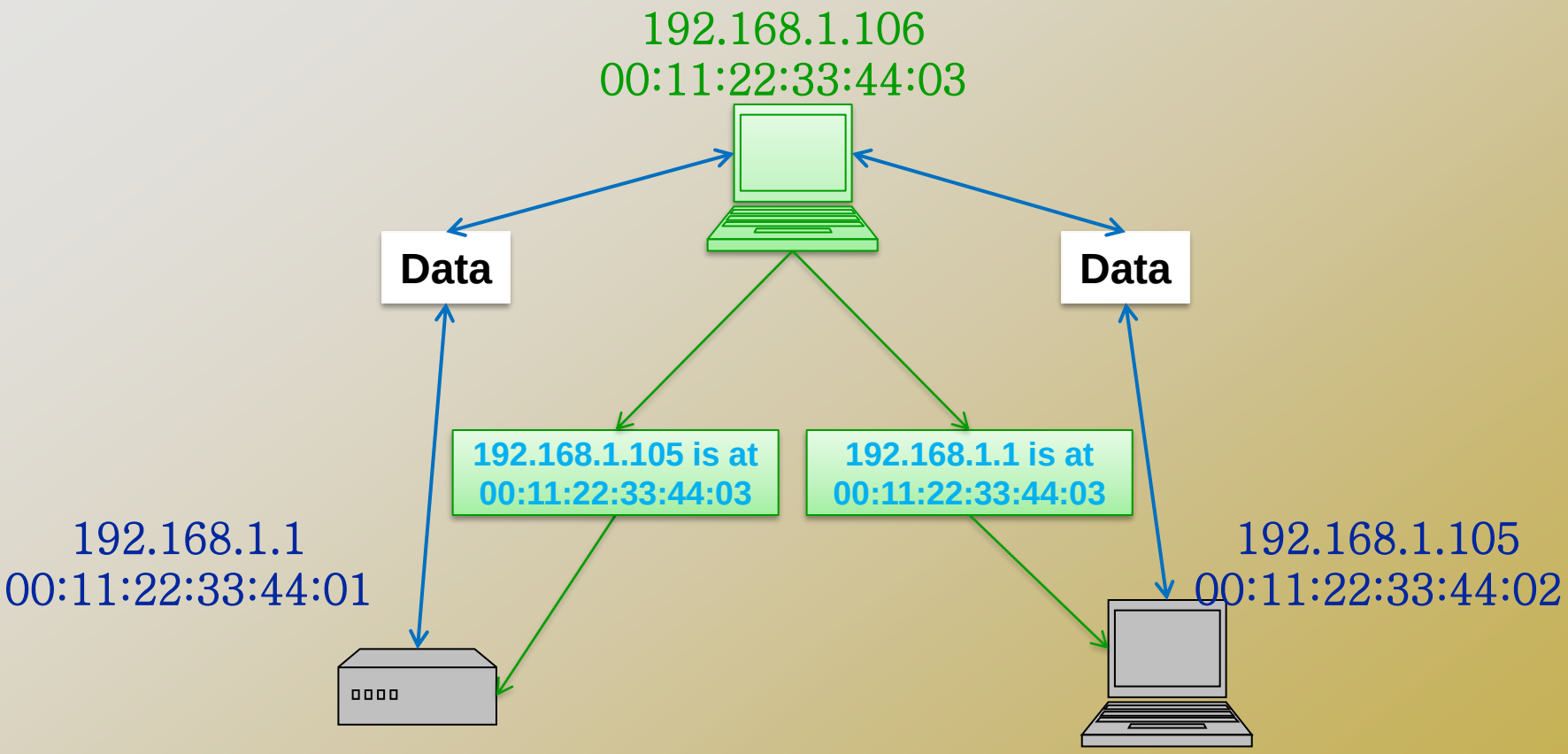
ARP Caches



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

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

Poisoned ARP Caches (man-in-the-middle attack)



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

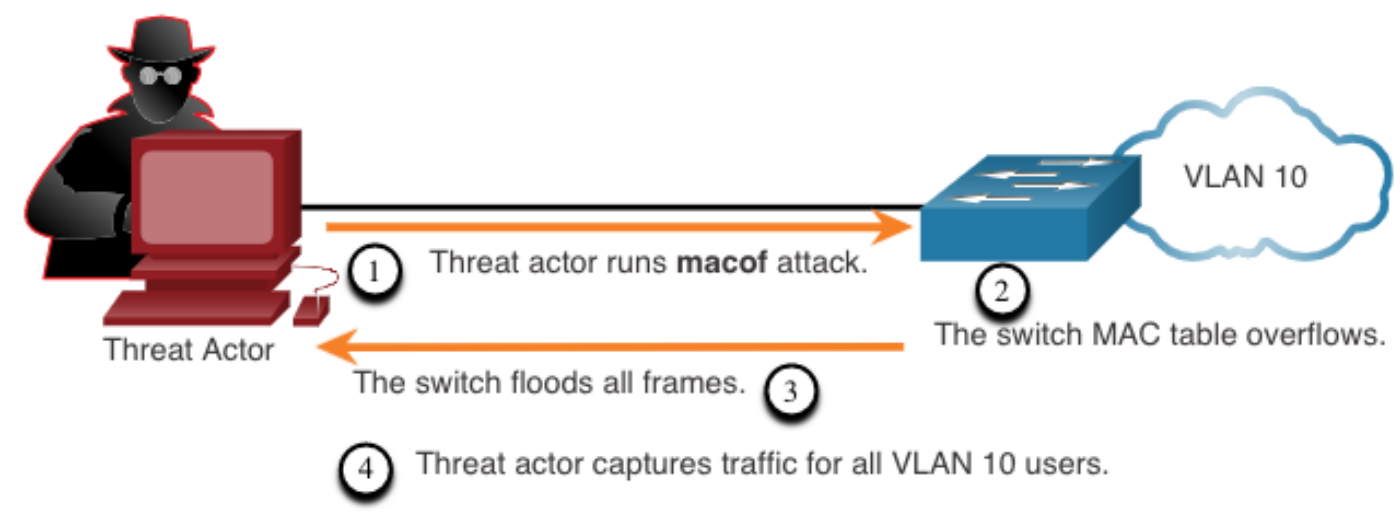
Poisoned ARP Cache	
192.168.1.**1**	00:11:22:33:44:**03**

ARP Spoofing

- Using static entries solves the problem but it is almost impossible to manage!
- Check multiple occurrence of the same MAC
  - i.e., One MAC mapping to multiple IP addresses (see previous slide's example)
- Software detection solutions
  - Anti-arpspoof, Xarp, Arpwatch

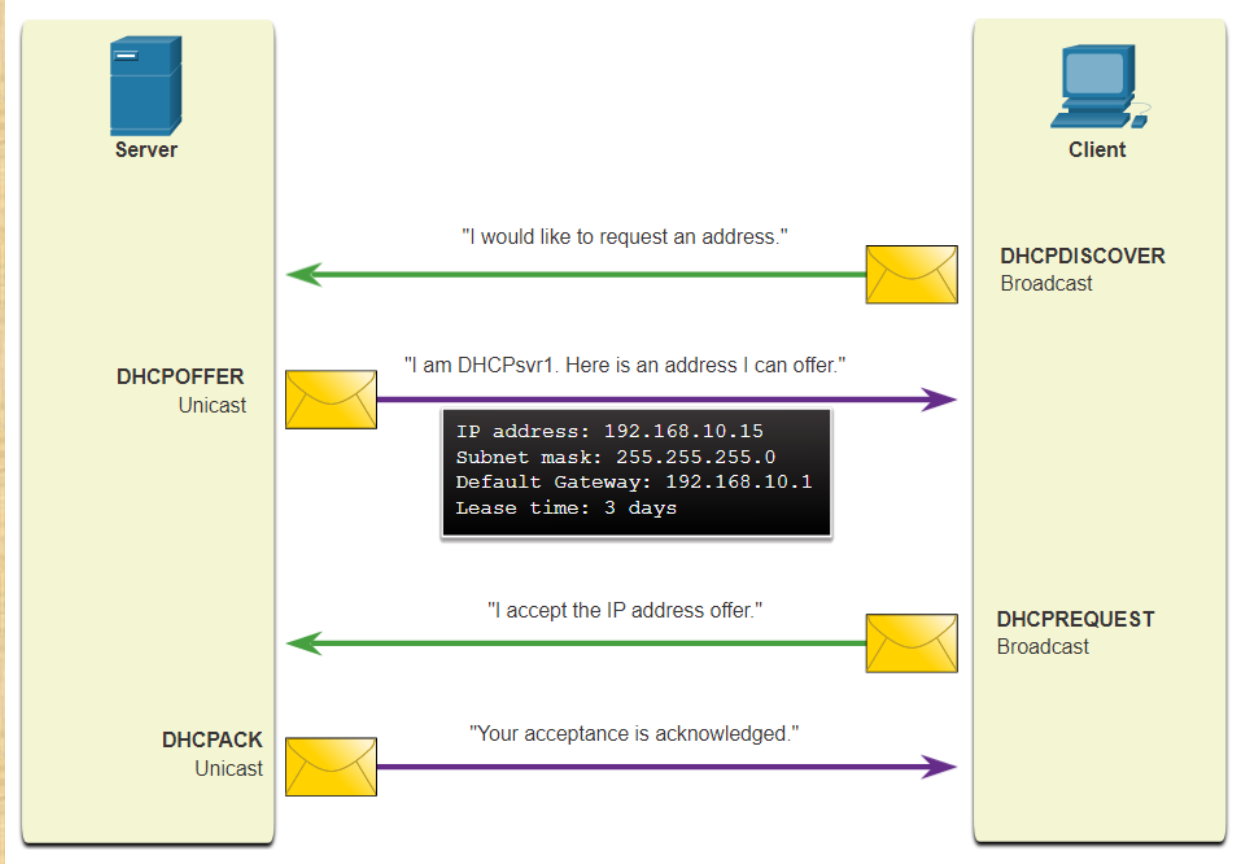
MAC Address Table Flooding

- All MAC tables have a fixed size and consequently, a switch can run out of resources in which to store MAC addresses.
- MAC address flooding attacks take advantage of this limitation by bombarding the switch with fake source MAC addresses until the switch MAC address table is full.
- When this occurs, the switch treats the frame as an unknown unicast and begins to flood all incoming traffic out all ports without referencing the MAC table.
- This condition now allows a threat actor to capture all of the frames sent from one host to another on the local LAN or local VLAN.



DHCP

- DHCP servers dynamically provide IP configuration information including IP address, subnet mask, default gateway, DNS servers, and more to clients.



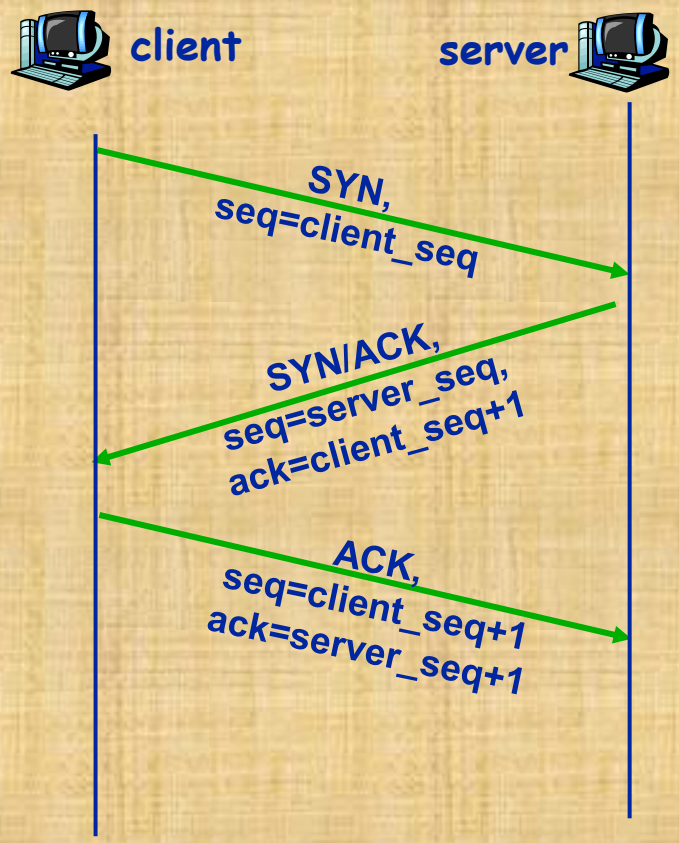
DHCP Attacks

- **DHCP Starvation Attack:**
- The goal of this attack is to create a DoS for connecting clients. Such Attack tool (e.g. Gobbler) creates DHCP discovery messages with bogus MAC addresses has the ability to look at the entire scope of leasable IP addresses and tries to lease them all.

- **DHCP Spoofing Attack –** This occurs when a rogue DHCP server is connected to the network and provides false IP configuration parameters to legitimate clients. A rogue server can provide a variety of misleading information, including the following:
  - Wrong default gateway - Can lead to a man-in-the-middle attack. This may go entirely undetected
  - Wrong DNS server - pointing the user to a nefarious website.
  - Wrong IP address - An invalid IP address effectively creating a DoS attack on the DHCP client.

TCP Session Hijacking

- TCP connection has both sequence number and acknowledge number in each packet.
- The two ends negotiate what seq. and ack. Numbers to be used in TCP set up stage.
- seq and ack number size: 2^{32}
  - Makes seq/ack guessing very hard to achieve
  - Very hard to hijack an already setup TCP connection!



TCP Session Hijacking

- Possible when an attacker is on the same network segment as the target machine.
  - Attacker can sniff all back/forth tcp packets and know the seq/ack numbers.
  - Attacker can inject a packet with the correct seq/ack numbers with the spoofed IP address.
    - IP spoofing needs low-level packet programming, OS-based socket programming cannot be used!

TCP Session Hijacking

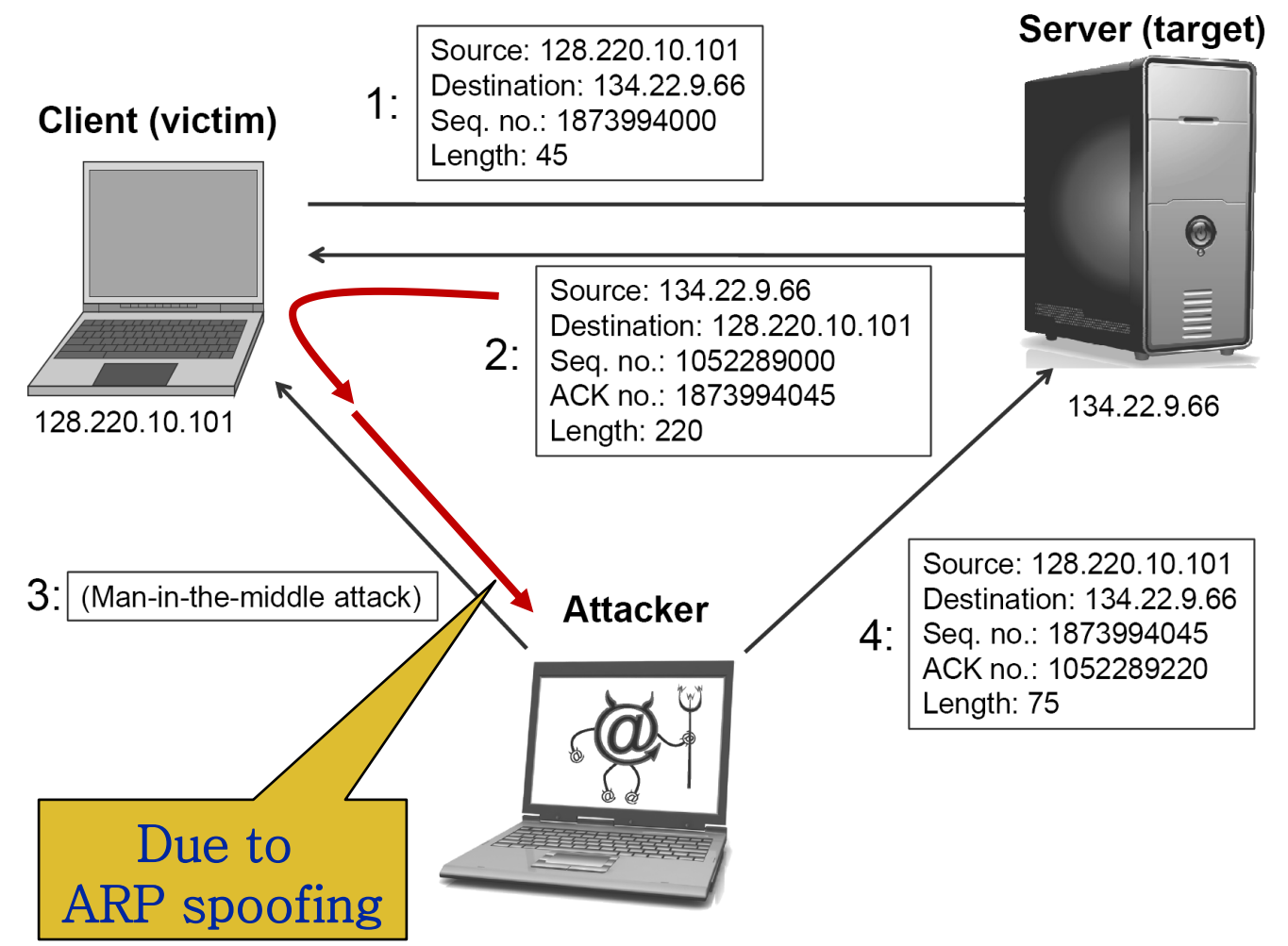
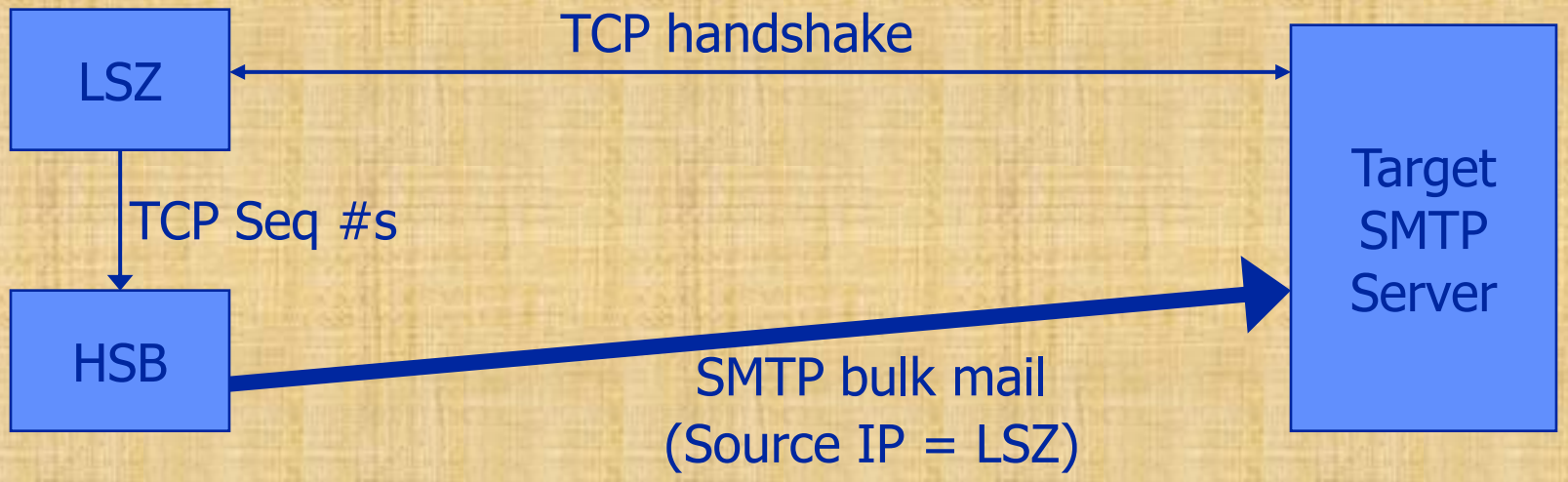


Figure 5.18: A TCP session hijacking attack.

TCP Session Hijacking

- Another way is "coordinated IP spoofing" by using two computers, such as the "Thin pipe / Thick pipe method":
  - High Speed Broadband connection (HSB)
  - Controls a Low Speed Zombie (LSZ)
  - Assumes no egress filtering at HSB's ISP
  - Hides IP address of HSB. LSZ is blacklisted.



Denial-of-Service (DoS) Attack

- An attempt to make a computer or network resource unavailable to its intended users
  - DoS to the network bandwidth of targeted server
  - DoS to the computing resource of targeted server
    - Memory, CPU
  - DoS to the vulnerability in targeted server
    - Causing server OS crash (buffer overflow bug, logic bug, etc)
    - Causing server program crash (e.g., Apache, Sendmail, SQL)
- Distributed Denial-of-Service (DDoS) attack
  - Sending attack packets from multiple computers
  - Botnet is the root cause for DDoS attacks

Denial-of-Service (DoS) Attack

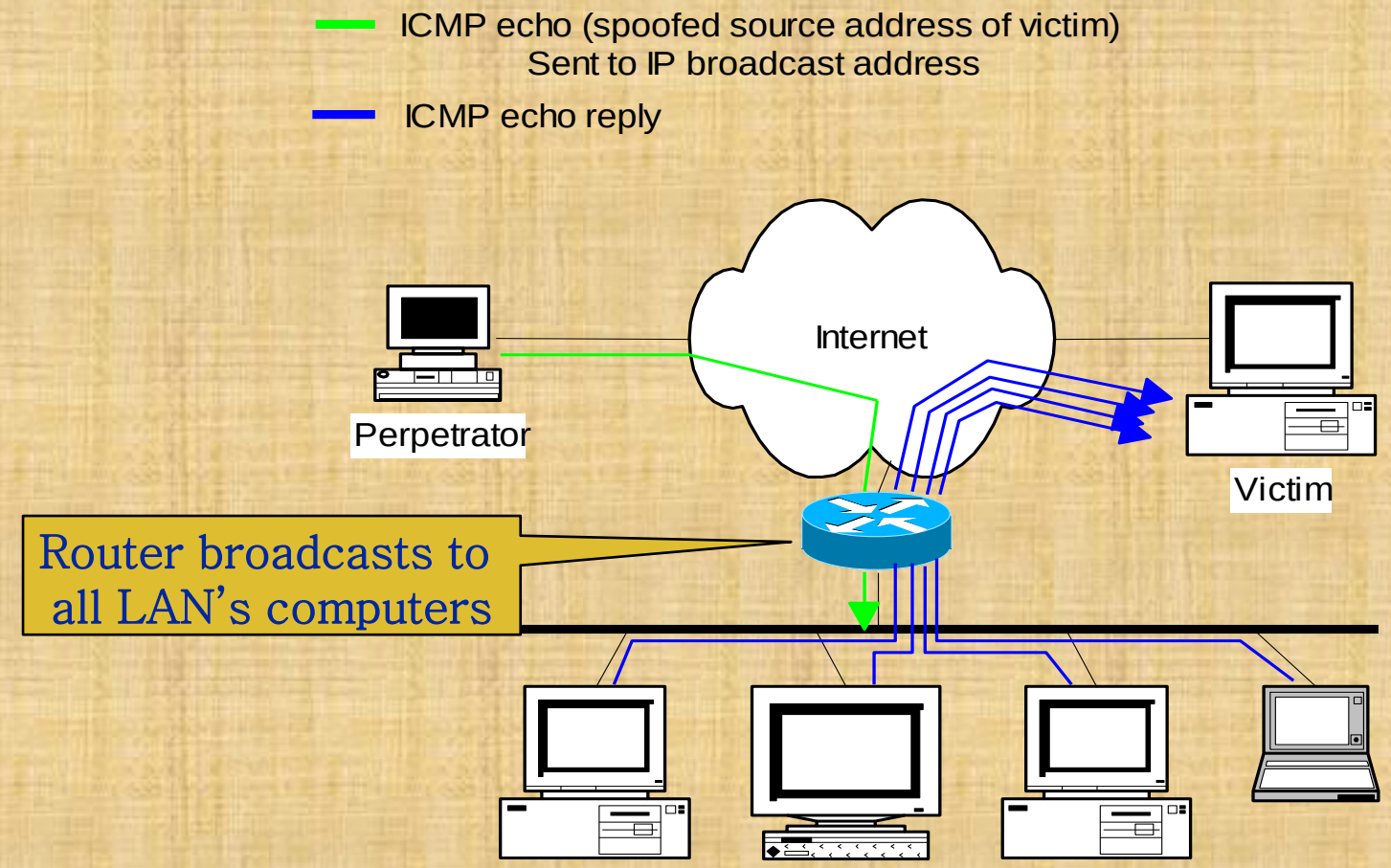
- Format:
  - Real IP-based attack using botnets
    - Attacker does not worry about exposing bots' IP addresses.
    - TCP flooding, UDP flooding, icmp flooding
  - Spoofed IP-based attack
    - SYN flooding with spoofed IPs.
  - Source address hiding attack
    - Smurf attack

Smurf Attack

- Uses ICMP echo/reply packets with broadcast networks to multiply traffic
- Requires the ability to send spoofed packets

- Abuses "bounce-sites" to attack victims
  - Traffic multiplied by a factor of 50 to 200

Description of Smurfing Attack



How to prevent being a "bounce site"

- Turn off directed broadcasts to subnets with 5 hosts or more
  - Cisco router: Interface command "no ip directed-broadcast"
- Use access control lists (if necessary) to prevent ICMP echo requests from entering your network
  - Probably not an elegant solution; makes troubleshooting difficult, but many networks are doing this now
- Encourage vendors to turn off replies for ICMP echos to broadcast addresses
  - Host Requirements RFC-1122 Section 3.2.2.6 states "An ICMP Echo Request destined to an IP broadcast or IP multicast address MAY be silently discarded."
  - Patches are available for free UNIX-ish operating systems.

SYN Flooding Attack

- An attacker sends a large number of SYN requests to a target's system
  - Target uses too much memory and CPU resources to process these fake connection requests
  - Target's bandwidth is overwhelmed
- Usually, SYN flood packets use spoofed source IPs
  - No TCP connection is set up (not like the TCP hijacking!)
  - Hide attacking source
  - Make the target very hard to decide which TCP SYN is attack and which TCP SYN is from legitimate users!

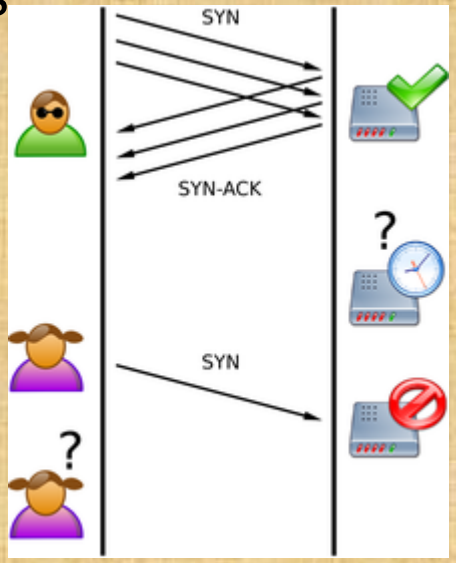


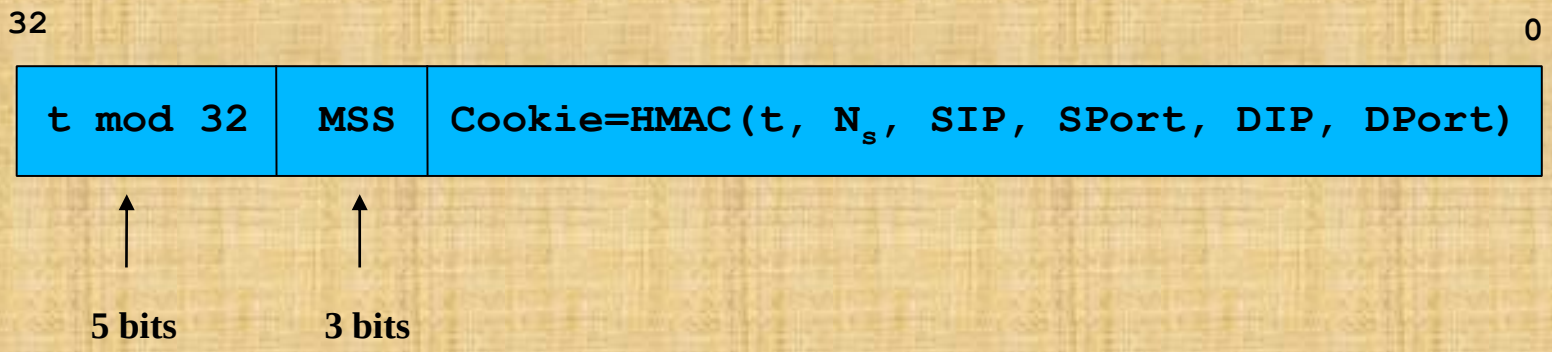
Image from wikipedia

SYN Flood Defense: SYN Cookie

- General idea
  - Client sends SYN to server (client_seq number only)
  - Server responds to Client with SYN-ACK cookie
    - Server_sqn = f(src addr, src port, dest addr, dest port, rand)
    - Ack number is normal value: client_seq +1
    - Server does not save state
  - Honest client responds with ACK(client_ack = server_sqn+1)
  - Server checks response using a reverse function whether this is valid
  - If matches SYN-ACK, establishes connection

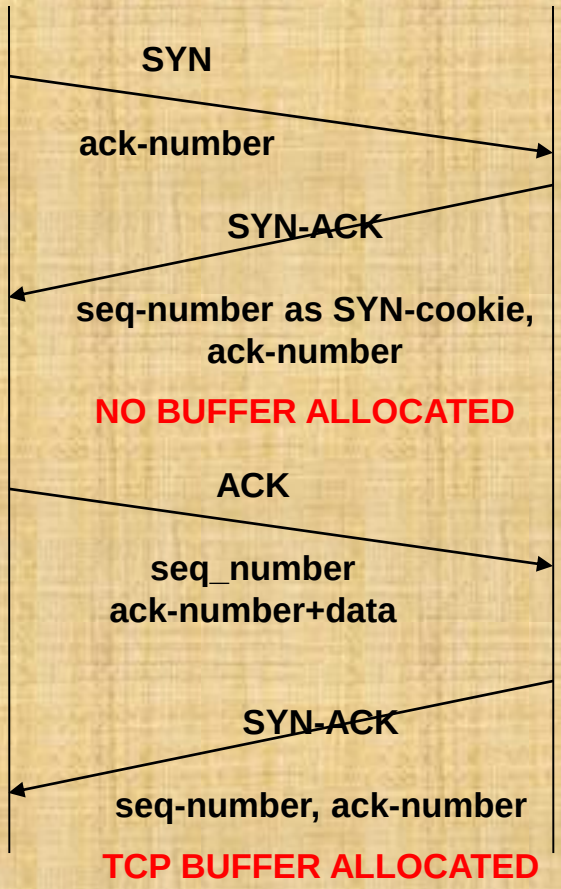
TCP SYN cookie

- TCP SYN/ACK server_seq encodes a cookie
  - 32-bit sequence number
    - **time mod 32:** counter to ensure sequence numbers increase every 64 seconds
    - **MSS:** encoding of server MSS (can only have 8 settings)
    - **Cookie:** easy to create and validate, hard to forge
      - Includes timestamp, nonce, 4-tuple



SYN Cookies

- client
  - sends SYN packet and ACK number to server
  - waits for SYN-ACK from server w/ matching ACK number
- server
  - responds w/ SYN-ACK packet w/ initial SYN-cookie sequence number
  - Sequence number is cryptographically generated value based on client address, port, and time.
- client
  - sends ACK to server w/ matching sequence number
- server
  - If ACK is to an unopened socket, server validates returned sequence number as SYN-cookie
  - If value is reasonable, a buffer is allocated and socket is opened

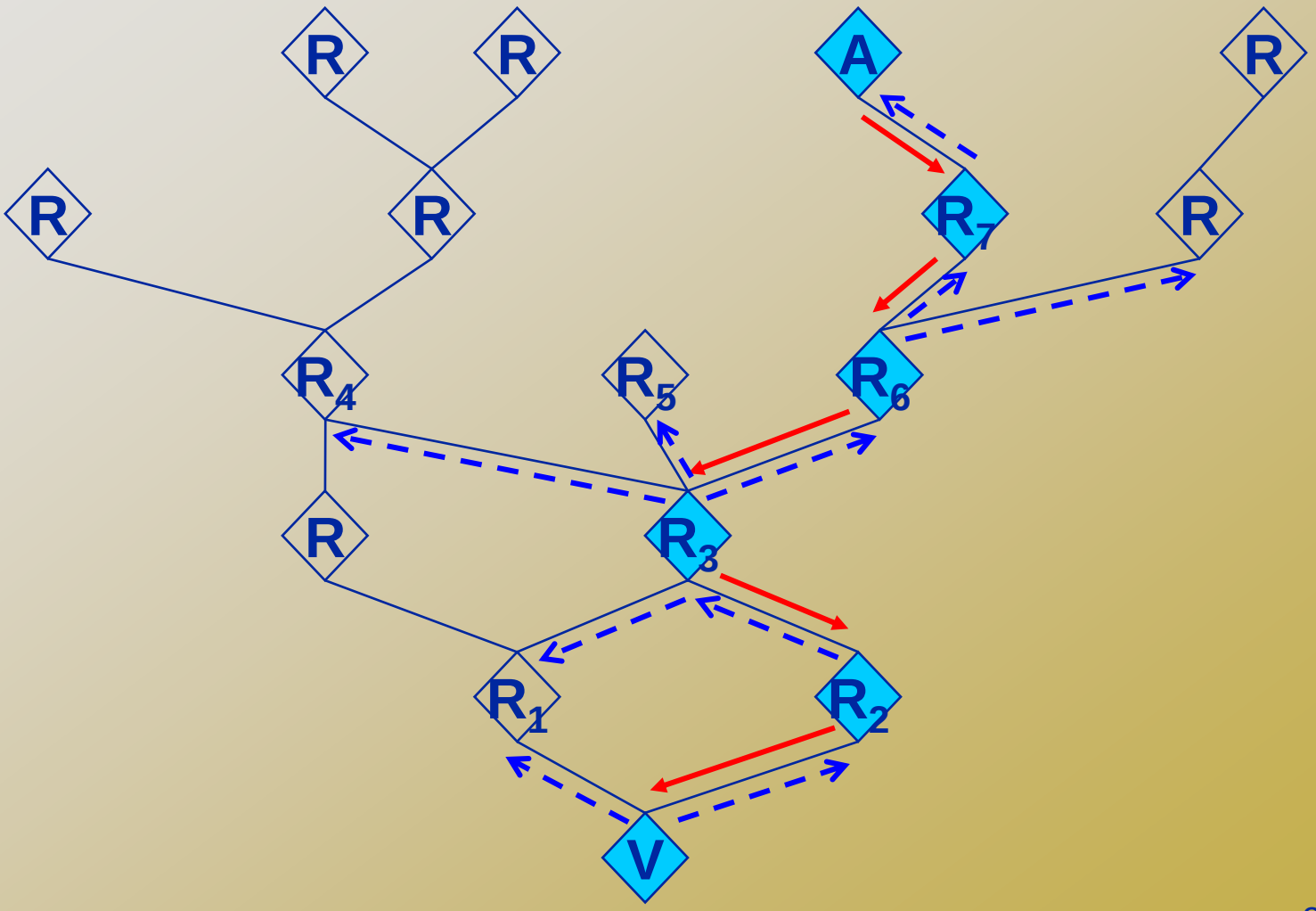


SYN Cookies Limitation

- Windows has not adopted SYN cookies
- Some Linux distributions have used it

- Maximum segment size can only be 8 possible values
- Do not allow the use of TCP option field
  - Many TCP option fields have been used by many programs

IP Traceback



Logging Challenges

- Attack path reconstruction is difficult
  - Packet may be transformed as it moves through the network
- Full packet storage is problematic
  - Memory requirements are prohibitive at high line speeds (OC-192 is ~10Mpkt/sec)
- Extensive packet logs are a privacy risk
  - Traffic repositories may aid eavesdroppers