Module 3
Network Security

Submodule 2: Network Security Basics

Network Security Issues-I

- Confidentiality
  - Network communication needs to be encrypted
  - Encryption can be done at the application layer or by revising a lower layer protocol to include encryption
- Integrity
  - Checksums in headers and footers can validate integrity of data, but not cryptographically secure
  - True integrity needs to be provided at application layer
- Availability
  - The scale of the Internet is a challenge
  - Availability solutions need to scale with the increases in communication requests

Network Security Issues-II

- Assurance
  - Permissions and policies that control the flow of data in a network
  - For instance, firewall can block unwanted traffic
- Authenticity
  - There is no notion of user identities in the Internet Protocol stack
  - Needs to be introduced explicitly at the application layer using alternative protocol
- Anonymity
  - User anonymity is default

Link Layer Security

Switch

- In a small local area network, we can use switch to connect computers.
- A switch is a network device:
  - Operates at the link layer
  - Has multiple ports, with each port connected to a computer
  - Is capable of forwarding frame to only intended destination by:
    - Learn the MAC address of each computer connected to it
    - Forward frames only to the destination computer
  - Can also do broadcasting

Network Interfaces

- Network interface: device connecting 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)

MAC Addresses-I

- Media Access Control (MAC) address:
  - Is a hardware specific identifier
  - Can identify network interfaces
  - Predefined by manufacturer

- MAC address format:
  - 48-bit number
  - Usually in hex such as: 00-1A-92-D4-BF-86

MAC Addresses-II

- The first three octets of any MAC address are IEEE-assigned Organizationally Unique Identifiers
  - For example: Cisco 00-1A-A1, D-Link 00-1B-11, ASUSTek 00-1A-92
- The second three octets of a MAC address can be used by the manufacturer for purpose such as identifying different model instance
  - Need to be mindful of the uniqueness of the address
- MAC address can be reconfigured by network interface driver software
  - They cannot be used as a reliable means of identifying an untrusted source of network traffic

Manipulating MAC Addresses-I

- Viewing the MAC addresses of the interfaces of a machine
  - Linux: `ifconfig`
  - Windows: `ipconfig /all`

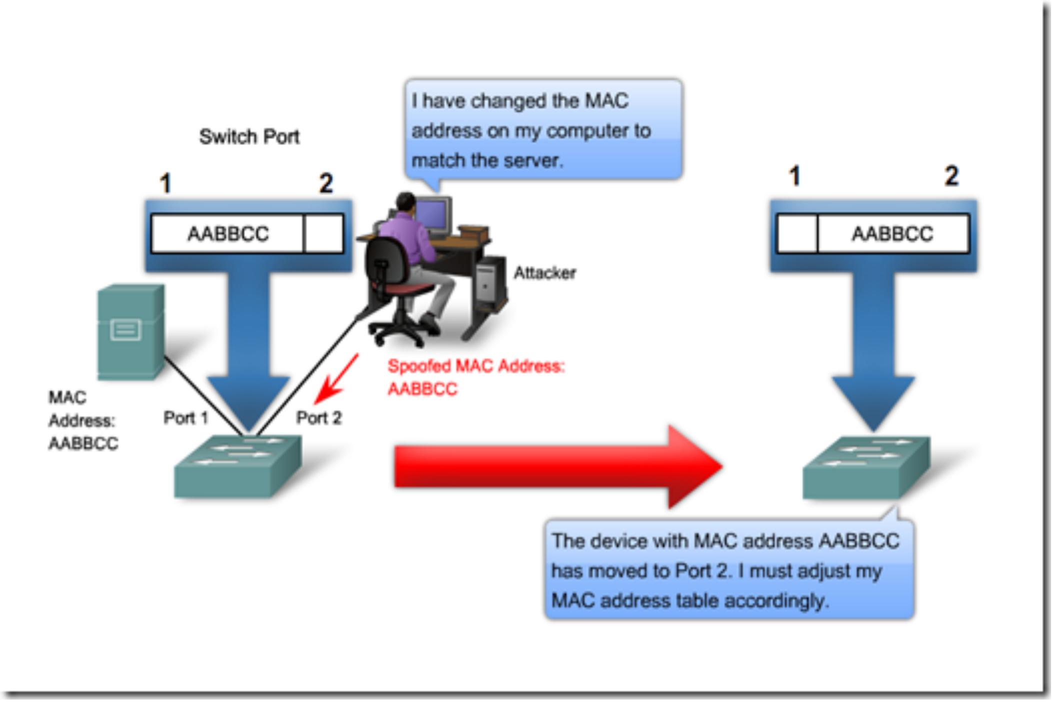
- Changing a MAC address in Linux
  - Stop the networking service: `/etc/init.d/network stop`
  - Change the MAC address: `ifconfig eth0 hw ether <MAC-address>`
  - Start the networking service: `/etc/init.d/network start`

Manipulating MAC Addresses-II

- Changing a MAC address in Windows
  - Open the Network Connections applet
  - Access the properties for the network interface
  - Click “Configure …”
  - In the advanced tab, change the network address to the desired value
- Changing a MAC address requires administrator privileges

MAC Address Filtering

- A switch can be configured to provide service only to machines with specific MAC addresses
- Allowed MAC addresses need to be registered with a network administrator


Switch Port
1
2
AABBCC
I have changed the MAC address on my computer to match the server.
Attacker
MAC Address: AABBCC
Port 1
Port 2
Spoofed MAC Address: AABBCC
1
2
AABBCC
The device with MAC address AABBCC has moved to Port 2. I must adjust my MAC address table accordingly.

MAC Spoofing Attack

- A MAC spoofing attack impersonates another machine
  - Find out MAC address of target machine
  - Reconfigure MAC address of rogue machine
  - Turn off or unplug target machine
- Countermeasures
  - Block port of switch when machine is turned off or unplugged
  - Disable duplicate MAC addresses

Address Resolution Protocol (ARP)

- ARP is a link layer protocol, it provides service to its upper layer—Network Layer
- This protocol is used to find a host’s hardware address given its network layer address, i.e., it is a translator between a MAC address and a given IP address.
- ARP spoofing is a man-in-the-middle attack against this protocol

ARP’s Working Mechanism

- How is the resolution of an IP address into a MAC address done:
  - A computer broadcasting a message of the form

    who has <IP address1> tell <IP address2>
  - The machine with <IP address1> or an ARP server receives this message sends the reply

    <IP address1> is <MAC address>
  - <IP address2> is used so the reply can be sent only to the machine who made the request
  - The machine stores the IP-MAC address pair in ARP Cache after receives the reply.
- Problem? Authentication is lacking.

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
```

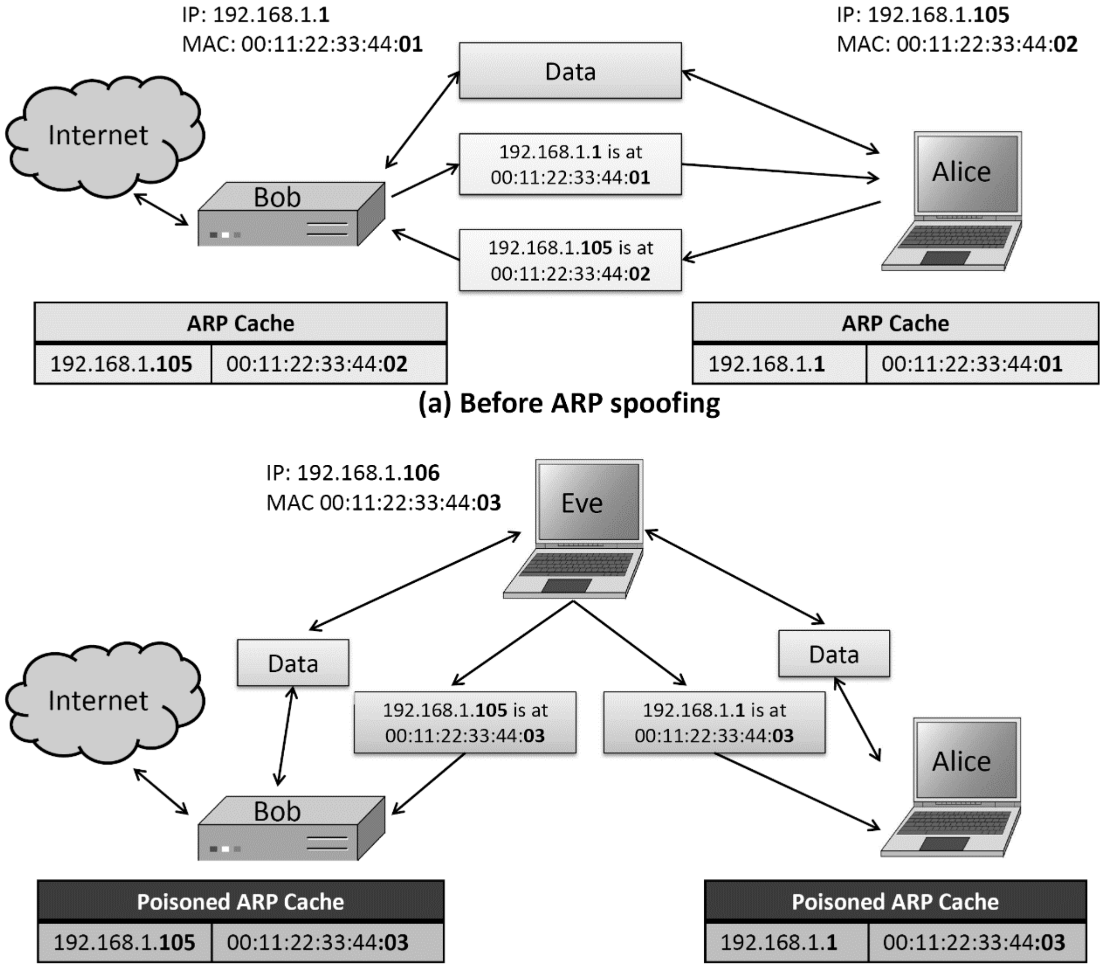
APR Spoofing

- What make this possible?
  - 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

- 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
- Using static entries solves the problem but it is almost impossible to manage!

Man-in-the-Middle 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**

(a) Before ARP spoofing

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**

(b) After ARP spoofing

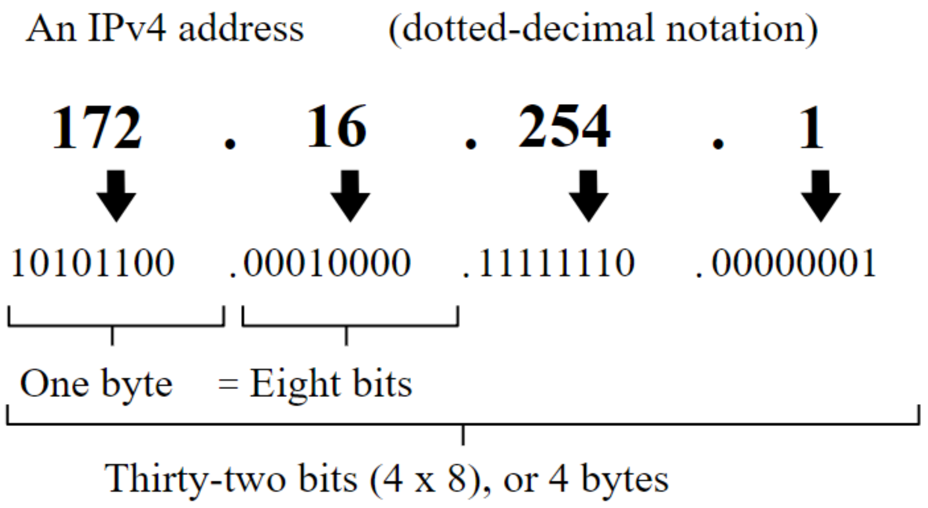
Figure 5.8: ARP spoofing enables a man-in-the-middle attack: (a) Before the ARP spoofing attack. (b) After the attack.

Source: Introduction to Computer Security by Goodrich & Tamassia

Network Layer Security

Internet Protocol

- Network layer is responsible for transmitting packets from one host to another.
- This layer use protocols such as the Internet Protocol (IP) to route the packet from source to destination.
  - Each node has a unique address—a numerical IP address.
  - IPv4 uses 32 bits for address
  - IPv6 uses 128 bits for address



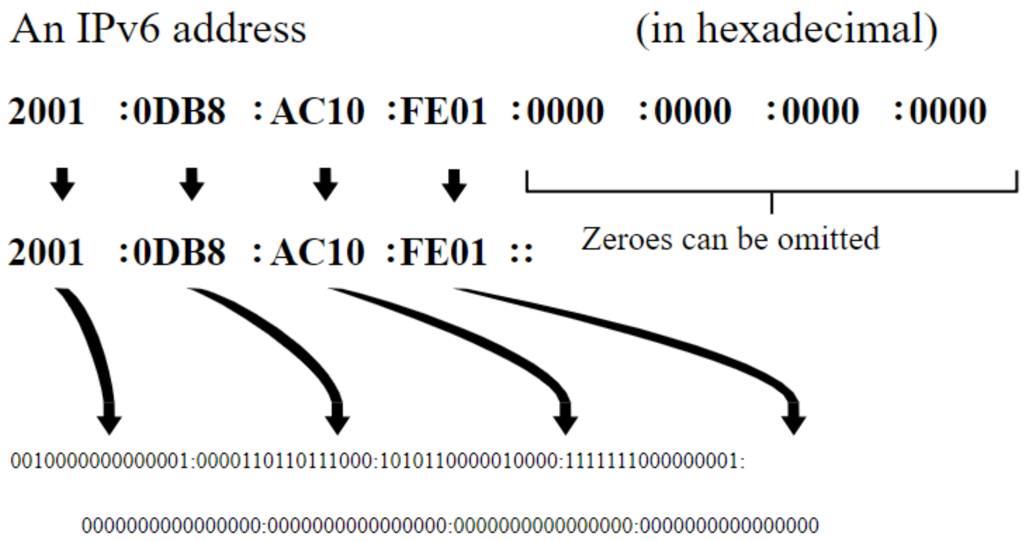
An IPv4 address (dotted-decimal notation)

172 . 16 . 254 . 1

10101100 .00010000 .11111110 .00000001

One byte = Eight bits

Thirty-two bits (4 x 8), or 4 bytes



An IPv6 address (in hexadecimal)

2001 :0DB8 :AC10 :FE01 :0000 :0000 :0000 :0000

2001 :0DB8 :AC10 :FE01 ::

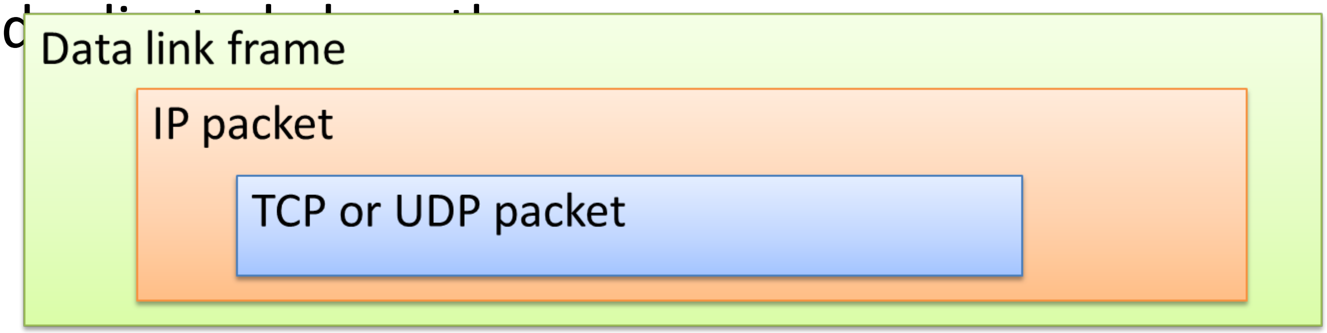
Zeroes can be omitted

0010000000000001:0000110110111000:1010110000010000:1111111000000001:

0000000000000000:0000000000000000:0000000000000000:0000000000000000

Characteristics of Internet Protocol

- It is connectionless
  - Each packet is transmitted separately from other packets
- It is not reliable
  - The delivery of packets are carried out on the “best effort” basis.
  - Sender will not received acknowledgements from the receiver of the packet.
  - Packets can get lost, reordered, compromised, or



Data link frame

IP packet

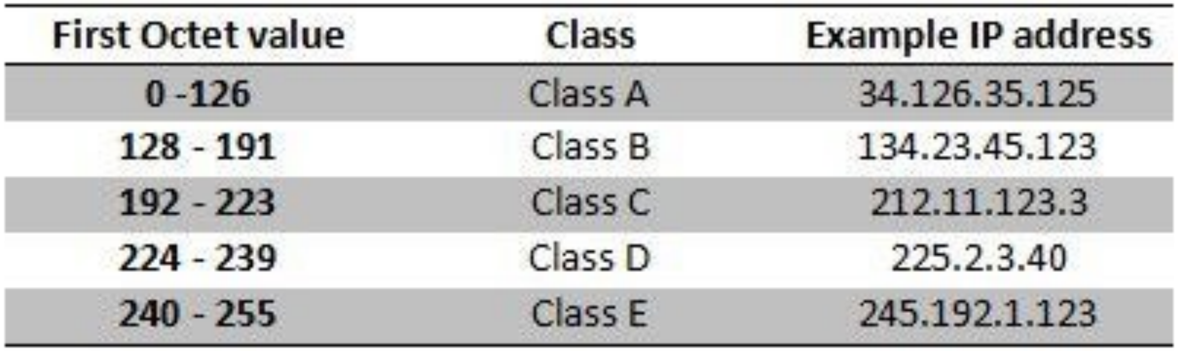
TCP or UDP packet

IP Address Basics-I

- All hosts on the Internet must have unique IP addresses.
- Internet Corporation for Assigned Names and Numbers (ICANN) is responsible for coordinating the maintenance of the namespaces and numerical spaces of the Internet.
  - It is a nonprofit organization
  - Incorporated in the US: historical bias in favor of US corporations and nonprofit organizations
  - Allocates IP address space
  - Manages top-level domains
    - Examples of top-level domains?

IP Address Basics-II

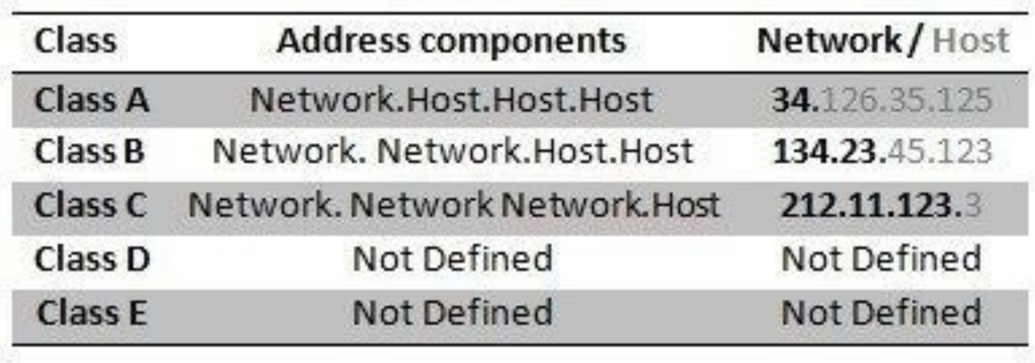
- Format of an IP address (IPv4)
  - Use the dotted decimal format—four octets, E.g., 128.148.32.110
  - Three numbers separated by decimal point, value of the number is between 0 and 255
  - An IP address belongs to a class depending on the number in the first octet:



First Octet value	Class	Example IP address
0 -126	Class A	34.126.35.125
128 - 191	Class B	134.23.45.123
192 - 223	Class C	212.11.123.3
224 - 239	Class D	225.2.3.40
240 - 255	Class E	245.192.1.123

IP Address Basics-III

- Structure of an IP address varies depending on its class:

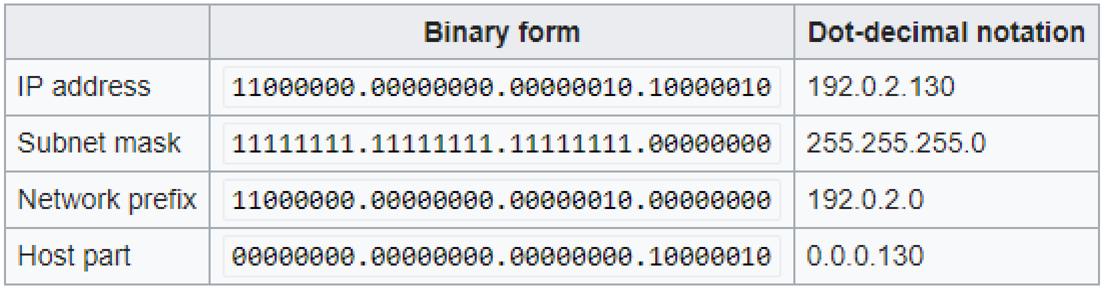


Class	Address components	Network / Host
Class A	Network.Host.Host.Host	34.126.35.125
Class B	Network. Network.Host.Host	134.23.45.123
Class C	Network. Network Network.Host	212.11.123.3
Class D	Not Defined	Not Defined
Class E	Not Defined	Not Defined

- Special IP addresses:
  - Loopback IP address: 127.0.0.1—does not reach outside the LAN
  - Broadcast address:
    - Limited broadcast: 255.255.255.255
    - Directed broadcast: 1.1.1.255

Subnetting-I

- What: Use subnet mask (32 bits, sequence of ones followed by block of zeros) with IP addresses to partition a network into logical groups, i.e., subnetworks (subnets).
- How:
  - Network portion of the IP address: perform bitwise AND on the subnet mask and the IP address
  - Host portion of the IP address: perform XOR on the result from previous step and the IP address



	Binary form	Dot-decimal notation
IP address	11000000.00000000.00000010.10000010	192.0.2.130
Subnet mask	11111111.11111111.11111111.00000000	255.255.255.0
Network prefix	11000000.00000000.00000010.00000000	192.0.2.0
Host part	00000000.00000000.00000000.10000010	0.0.0.130

Subnetting-II

- Subnet mask and network Class:
  - The smaller number of bits in a subnet mask, the higher number of hosts (unique IP addresses), the larger the network.
    - Class A: at least 8 bits subnet mask➔ up to 2^24 IP addresses
      - For large government organizations and telecommunication companies
    - Class B: at least 16 bits subnet mask➔ up to 2^16 IP addresses
      - For ISPs and large businesses
    - Class C: at least 24 bits subnet mask➔ up to 2^8 IP addresses
      - For smaller organizations

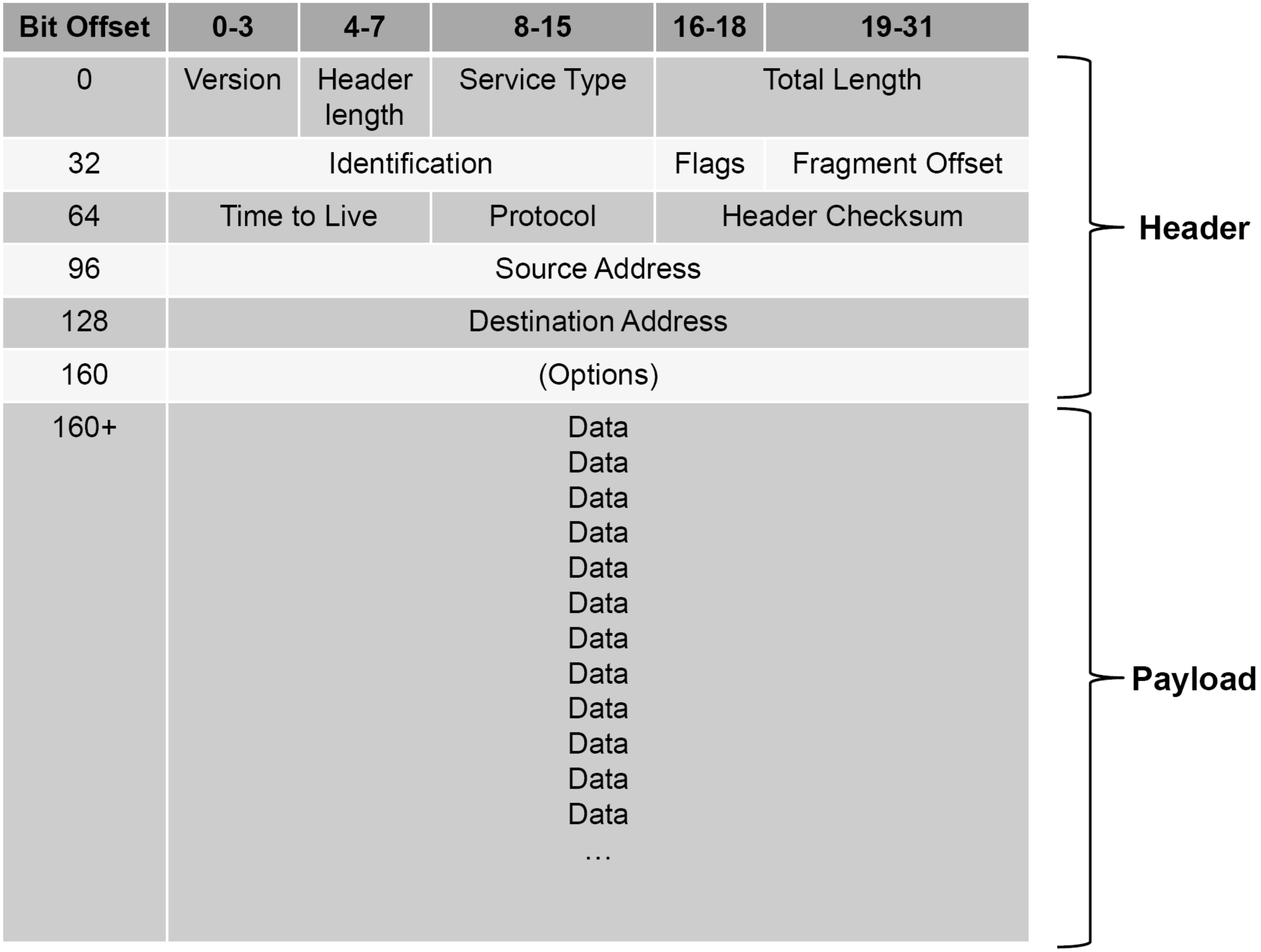
Subnetting Exercise

A Typical University’s IP Space

- Most universities separate their network connecting dorms and the network connecting offices and academic buildings
- Dorms
  - Class B network 138.16.0.0/16 (64K addresses)
- Academic buildings and offices
  - Class B network 128.148.0.0/16 (64K addresses)
- CS department
  - Several class C (/24) networks, each with 254 addresses

IP Addresses and Packets

- IP header includes
  - Source address
  - Destination address
  - Packet length (up to 64KB)
  - Time to live (up to 255)
  - IP protocol version
  - Fragmentation information
  - Transport layer protocol information (e.g., TCP)



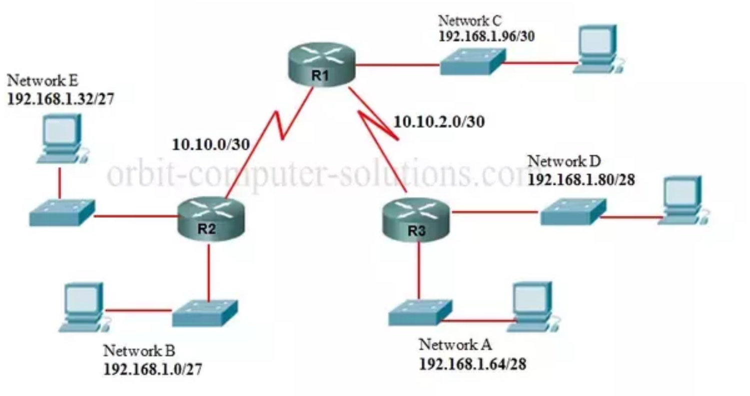
Bit Offset	0-3	4-7	8-15	16-18	19-31
0	Version	Header length	Service Type	Total Length	
32	Identification			Flags	Fragment Offset
64	Time to Live		Protocol	Header Checksum	
96	Source Address				
128	Destination Address				
160	(Options)				
160+	Data Data Data Data Data Data Data Data Data Data Data Data …				

Figure 5.10: Format of an IPv4 packet.

IP Routing

- A router bridges two or more networks
  - Operates at the network layer
  - Maintains tables to forward packets to the appropriate network
  - Forwarding decisions based solely on the destination address
- Router performs:
  - Drop
  - Deliver
  - Forward
- Routing table
  - Maps ranges of addresses to LANs or other gateway routers

IP Routing Process



Internet Routes

- Internet Control Message Protocol (ICMP)
  - Used for network testing, debugging, error notification
  - Simple messages encapsulated in single IP packets
    - Echo request
    - Echo response
    - Time exceeded
    - Destination unreachable
  - Considered a network layer protocol

- Tools based on ICMP
  - Ping: sends series of echo request messages and provides statistics on roundtrip times and packet loss
  - Traceroute: sends series ICMP packets with increasing TTL value to discover routes

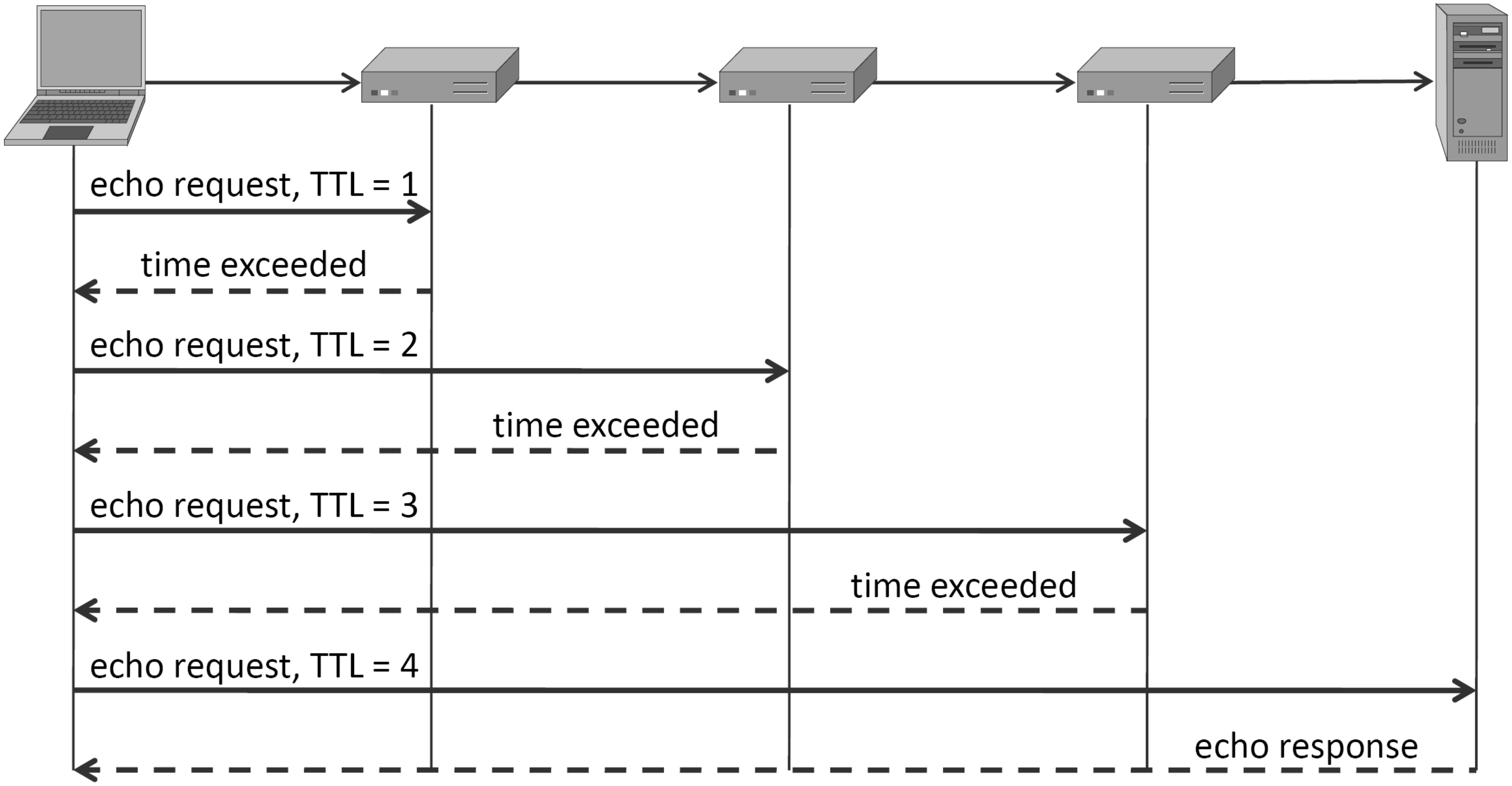
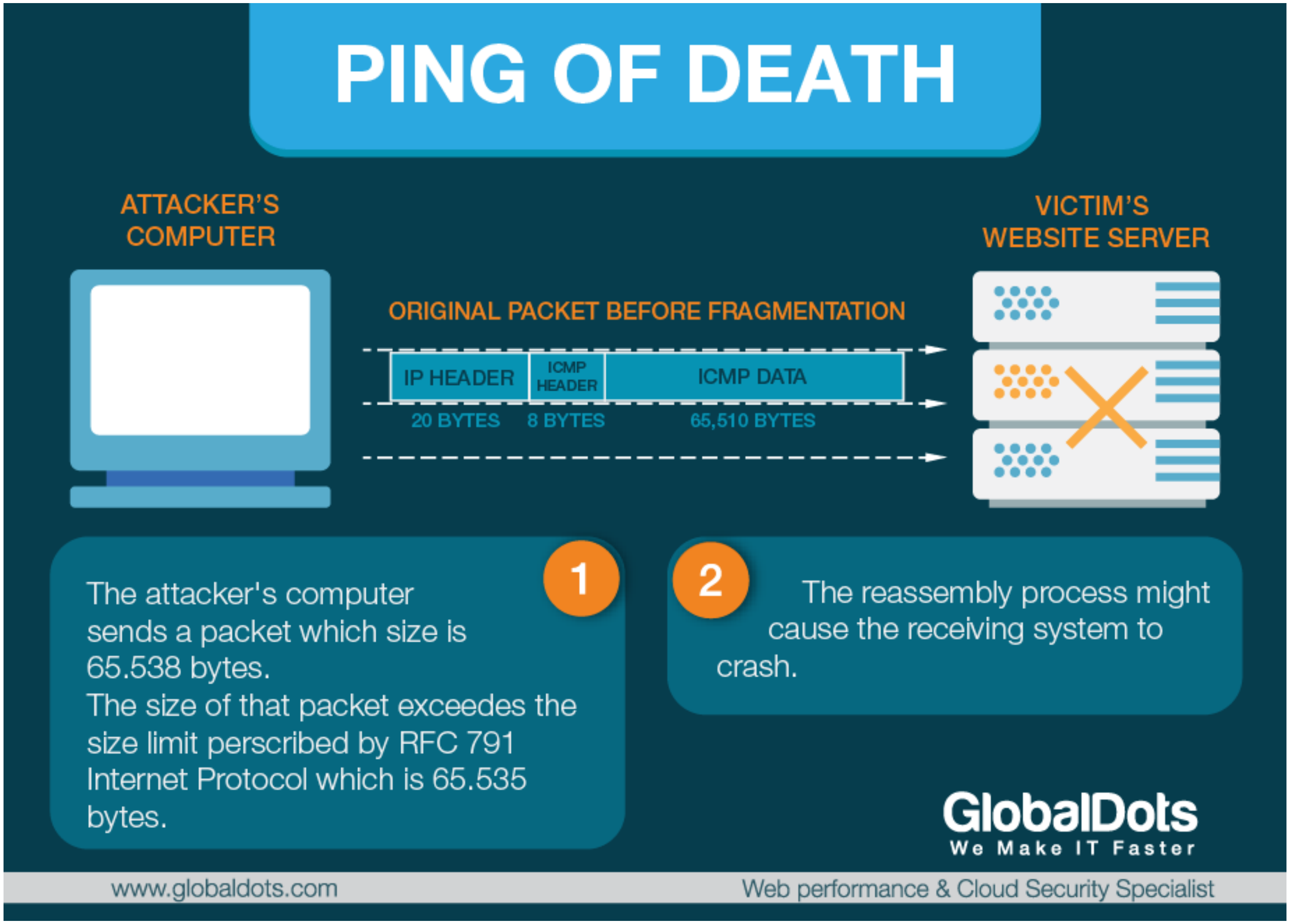


Figure 5.11: The traceroute utility.

ICMP Attacks

- Ping of death
  - ICMP specifies messages must fit a single IP packet (64KB)
  - Send a ping packet that exceeds maximum size using IP fragmentation
  - Reassembled packet caused several operating systems to crash due to a buffer overflow



PING OF DEATH

ATTACKER'S COMPUTER

VICTIM'S WEBSITE SERVER

ORIGINAL PACKET BEFORE FRAGMENTATION

IP HEADER	ICMP HEADER	ICMP DATA
20 BYTES	8 BYTES	65,510 BYTES

1 The attacker's computer sends a packet which size is 65.538 bytes. The size of that packet exceedes the size limit perscribed by RFC 791 Internet Protocol which is 65.535 bytes.

2 The reassembly process might cause the receiving system to crash.

GlobalDots
We Make IT Faster

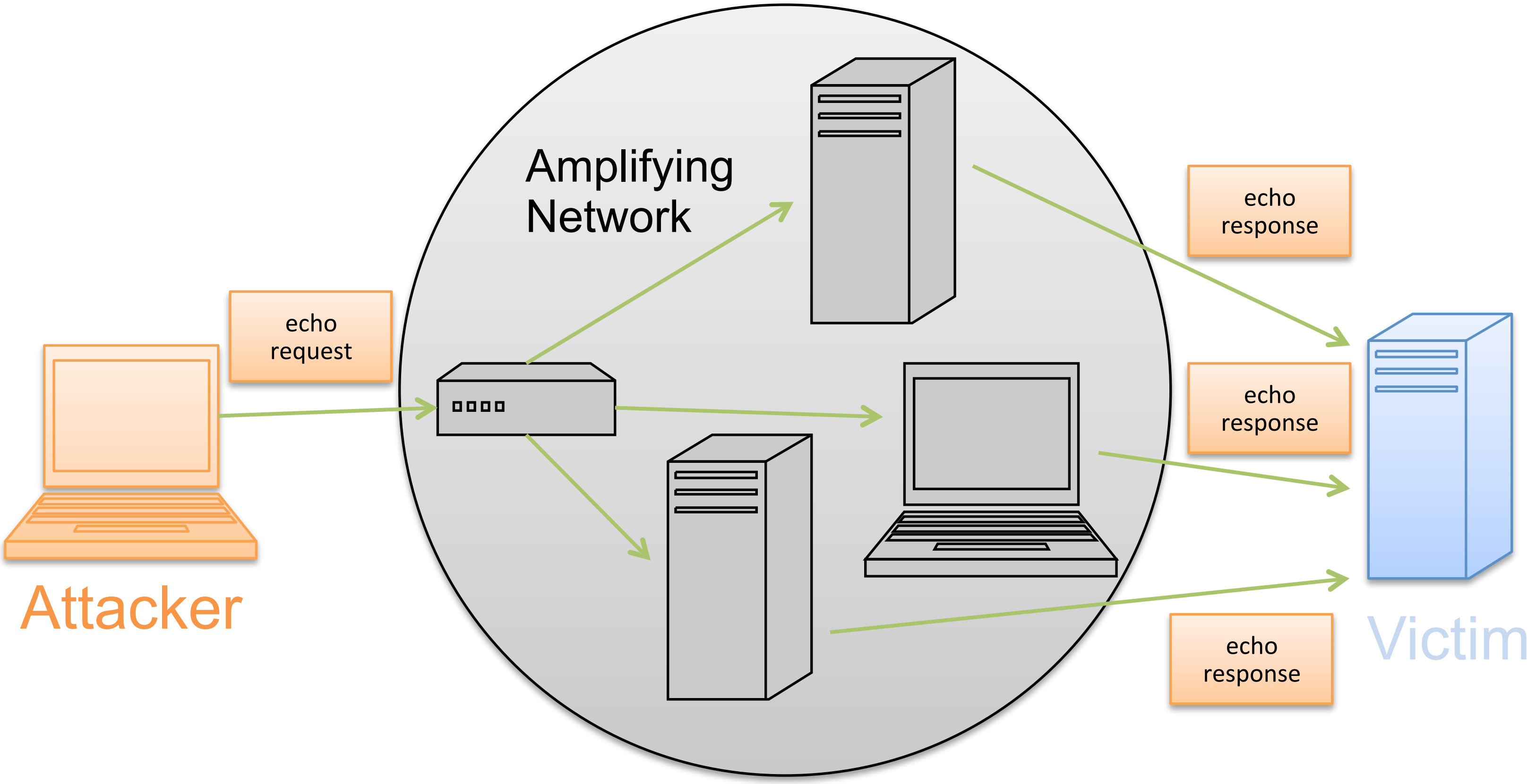
www.globaldots.com

Web performance & Cloud Security Specialist

Smurf Attack

Smurf

Ping a broadcast address using a spoofed source address



IP Vulnerabilities-I

- Unencrypted transmission
  - Eavesdropping possible at any intermediate host during routing
- No source authentication
  - Sender can spoof source address, making it difficult to trace packet back to attacker

IP Vulnerabilities-II

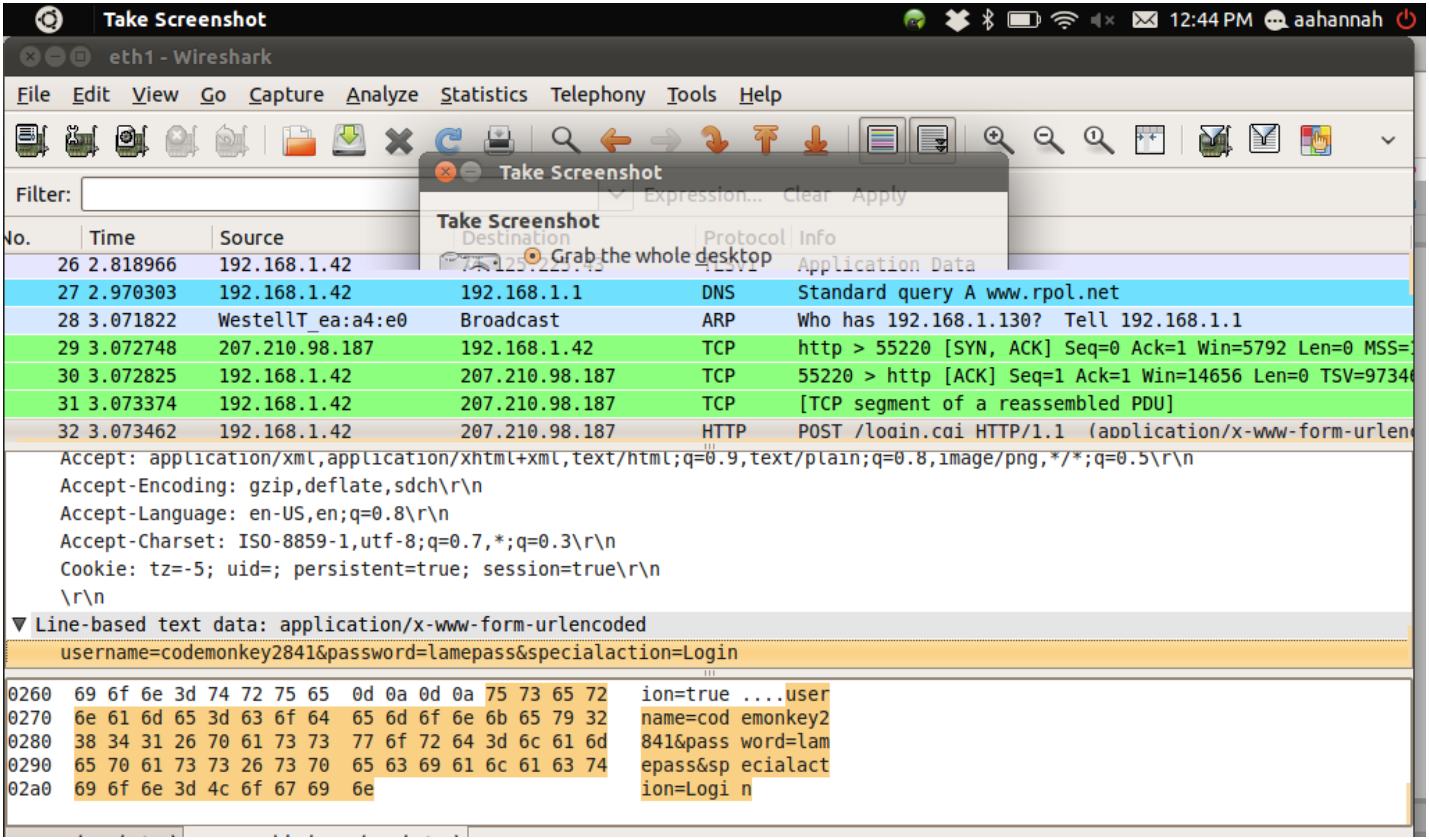
- No integrity checking
  - Entire packet, header and payload, can be modified while en route to destination, enabling content forgeries, redirections, and man-in-the-middle attacks
- No bandwidth constraints
  - Large number of packets can be injected into network to launch a denial-of-service attack
  - Broadcast addresses provide additional leverage

Deal with IP Spoofing

- Use border routers that can block packets from outside their domain that have source addresses from inside the domain.
- Implement IP traceback technique-tracing the path of a packet back to its actual source address

Packet Sniffing

- Eavesdropping can compromise the confidentiality of the packets.
- If a network interface is operating in promiscuous mode—an attacker could examine all data transmitted over a particular network segment
  - Could recover sensitive information



```
No.   Time        Source              Destination       Protocol  Info
 26 2.818966     192.168.1.42         74.125.225.43     TLSv1     Application Data
 27 2.970303     192.168.1.42         192.168.1.1       DNS       Standard query A www.rpol.net
 28 3.071822     WestellT_ea:a4:e0    Broadcast         ARP       Who has 192.168.1.130?  Tell 192.168.1.1
 29 3.072748     207.210.98.187       192.168.1.42      TCP       http > 55220 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=
 30 3.072825     192.168.1.42         207.210.98.187    TCP       55220 > http [ACK] Seq=1 Ack=1 Win=14656 Len=0 TSV=9734
 31 3.073374     192.168.1.42         207.210.98.187    TCP       [TCP segment of a reassembled PDU]
 32 3.073462     192.168.1.42         207.210.98.187    HTTP      POST /login.cgi HTTP/1.1  (application/x-www-form-urlen
```

```
    Accept: application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5\r\n
    Accept-Encoding: gzip,deflate,sdch\r\n
    Accept-Language: en-US,en;q=0.8\r\n
    Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.3\r\n
    Cookie: tz=-5; uid=; persistent=true; session=true\r\n
    \r\n
▼ Line-based text data: application/x-www-form-urlencoded
    username=codemonkey2841&password=lamepass&specialaction=Login
```

```
0260  69 6f 6e 3d 74 72 75 65  0d 0a 0d 0a 75 73 65 72   ion=true ....user
0270  6e 61 6d 65 3d 63 6f 64  65 6d 6f 6e 6b 65 79 32   name=cod emonkey2
0280  38 34 31 26 70 61 73 73  77 6f 72 64 3d 6c 61 6d   841&pass word=lam
0290  65 70 61 73 73 26 73 70  65 63 69 61 6c 61 63 74   epass&sp ecialact
02a0  69 6f 6e 3d 4c 6f 67 69  6e                        ion=Logi n
```

Transport Layer Security

Transmission Control Protocol-I

- TCP is a transport layer protocol guaranteeing reliable data transfer, in-order delivery of messages and the ability to distinguish data for multiple concurrent applications on the same host
- Most popular application protocols, including WWW, FTP and SSH are built on top of TCP
- TCP takes a stream of 8-bit byte data, packages it into appropriately sized segment and calls on IP to transmit these packets

Transmission Control Protocol-II

- Delivery order is maintained by marking each packet with a sequence number

- Every time TCP receives a packet, it sends out an ACK to indicate successful receipt of the packet.

- TCP generally checks data transmitted by comparing a checksum of the data with a checksum encoded in the packet

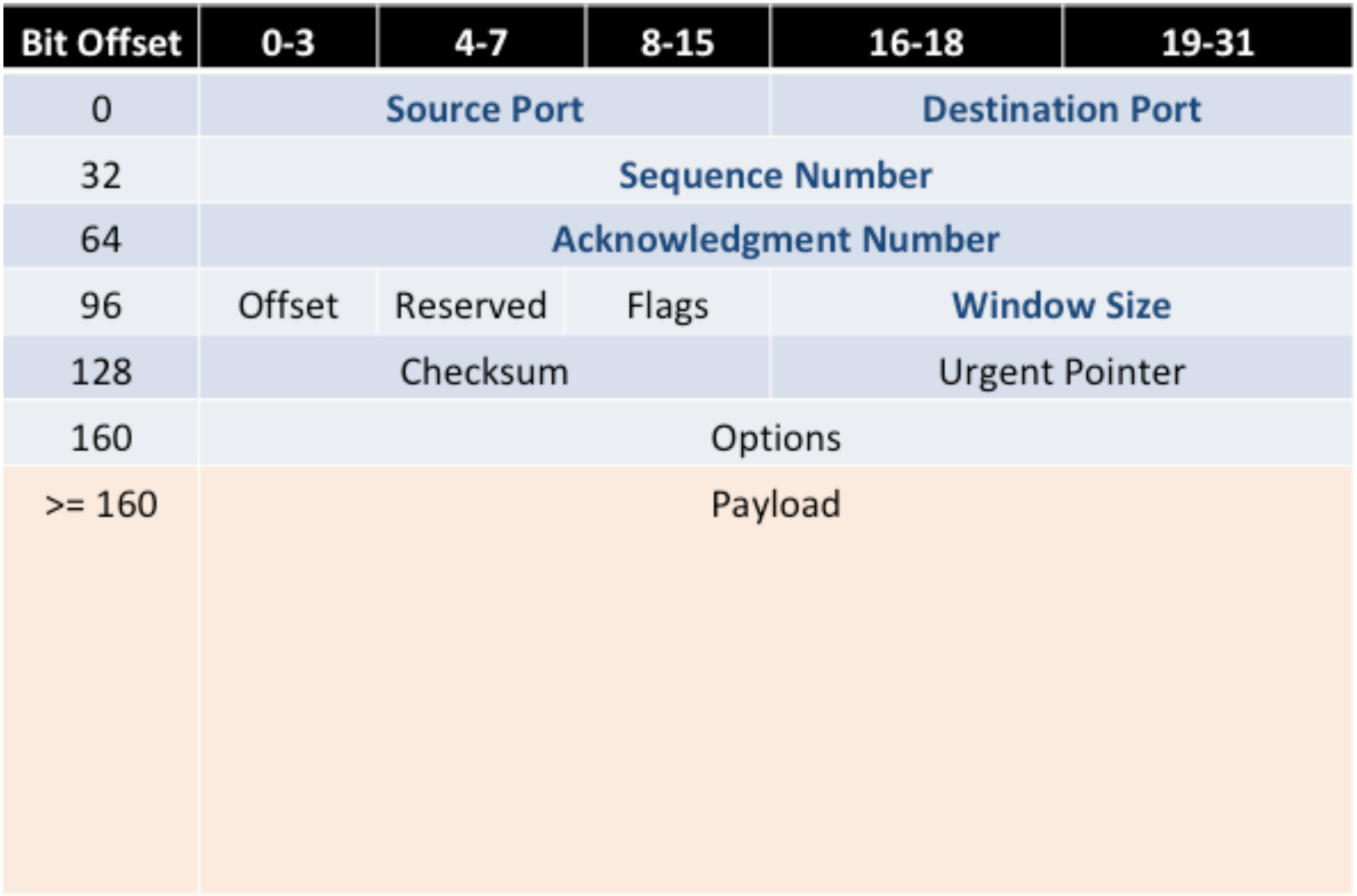
Ports-I

- TCP supports multiple concurrent applications on the same server
- Accomplishes this by having ports, 16 bit numbers identifying where data is directed
- The TCP header includes space for both a source and a destination port, thus allowing TCP to route all data
- In most cases, both TCP and UDP use the same port numbers for the same applications

Ports-II

- Ports 0 through 1023 are reserved for use by known protocols.
- Ports 1024 through 49151 are known as user ports, and should be used by most user programs for listening to connections and the like
- Ports 49152 through 65535 are private ports used for dynamic allocation by socket libraries
- For a list of common TCP/IP ports, click here

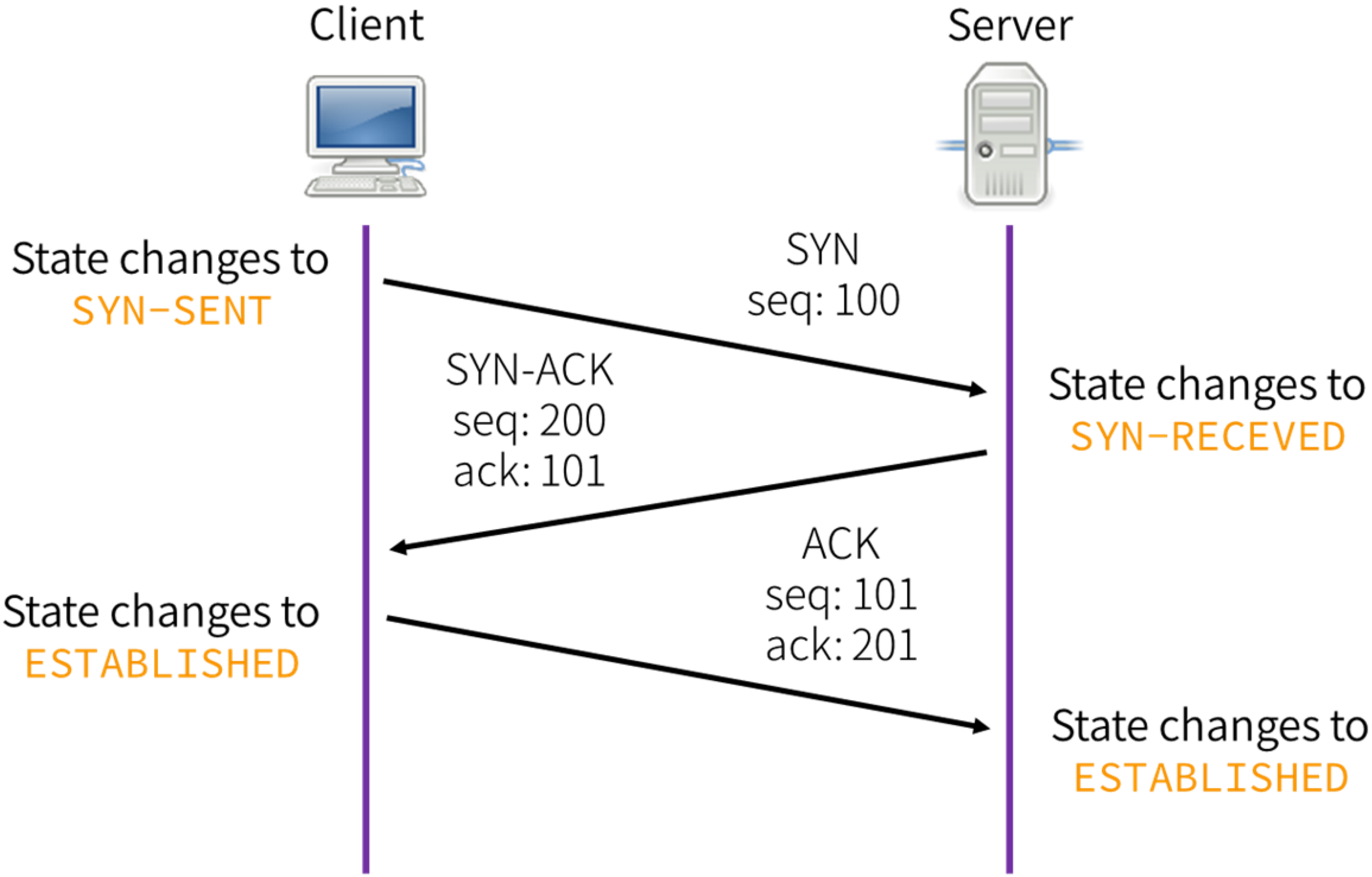
TCP Packet Format



Bit Offset	0-3	4-7	8-15	16-18	19-31
0	Source Port			Destination Port	
32	Sequence Number				
64	Acknowledgment Number				
96	Offset	Reserved	Flags	Window Size	
128	Checksum			Urgent Pointer	
160	Options				
>= 160	Payload				

Establishing TCP Connections

- TCP connections are established through a three way handshake.
- The server generally has a passive listener, waiting for a connection request
- The client requests a connection by sending out a SYN packet
- The server responds by sending a SYN/ACK packet, indicating an acknowledgment for the connection
- The client responds by sending an ACK to the server thus establishing connection



Client

Server

State changes to
SYN-SENT

SYN
seq: 100

State changes to
SYN-RECEVED

SYN-ACK
seq: 200
ack: 101

State changes to
ESTABLISHED

ACK
seq: 101
ack: 201

State changes to
ESTABLISHED

SYN Flood

- Typically DOS attack, though can be combined with other attack such as TCP hijacking
- Rely on sending TCP connection requests faster than the server can process them
- Attacker creates a large number of packets with spoofed source addresses and setting the SYN flag on these

SYN Flood-II

- The server responds with a SYN/ACK for which it never gets a response (waits for about 3 minutes each)
- Eventually the server stops accepting connection requests, thus triggering a denial of service.
- Can be solved in multiple ways
- One of the common way to do this is to use SYN cookies
  - Instead of allocating space on the connection table, the sequence number on the SYN/ACK packet is a carefully calculated hash of the connection requestors details, and when the server receives a response it adds the connection to the connection table after verifying information in the cookie.

TCP Data Transfer-I

- During connection initialization using the three way handshake, initial sequence numbers are exchanged
- The TCP header includes a 16 bit checksum of the data and parts of the header, including the source and destination
- Acknowledgment or lack thereof is used by TCP to keep track of network congestion and control flow and such

TCP Data Transfer-II

- TCP connections are cleanly terminated with a 4-way handshake
  - The client which wishes to terminate the connection sends a FIN message to the other client
  - The other client responds by sending an ACK
  - The other client sends a FIN
  - The original client now sends an ACK, and the connection is terminated

TCP Congestion Control-I

- During the mid-80s it was discovered that uncontrolled TCP messages were causing large scale network congestion

- TCP responded to congestion by retransmitting lost packets, thus making the problem worse

- What is predominantly used today is a system where ACKs are used to determine the maximum number of packets which should be sent out

- Most TCP congestion avoidance algorithms, avoid congestion by modifying a congestion window (cwnd) as more cumulative ACKs are received

TCP Congestion Control-II

- Lost packets are taken to be a sign of network congestion
- TCP begins with an extremely low cwnd and rapidly increases the value of this variable to reach bottleneck capacity
- At this point it shifts to a collision detection algorithm which slowly probes the network for additional bandwidth
- TCP congestion control is a good idea in general but allows for certain attacks.

Optimistic ACK Attack-I

- An optimistic ACK attack takes advantage of the TCP congestion control
- It begins with a client sending out ACKs for data segments it hasn’t yet received
- This flood of optimistic ACKs makes the servers TCP stack believe that there is a large amount of bandwidth available and thus increase cwnd

Optimistic ACK Attack-II

- This leads to the attacker providing more optimistic ACKs, and eventually bandwidth use beyond what the server has available

- This can also be played out across multiple servers, with enough congestion that a certain section of the network is no longer reachable

- There are no practical solutions to this problem

Session Hijacking

- Also commonly known as TCP Session Hijacking
- A security attack over a protected network
- Attempt to take control of a network session
- Sessions are server keeping state of a client’s connection
- Servers need to keep track of messages sent between client and the server and their respective actions
- Most networks follow the TCP/IP protocol
- IP Spoofing is one type of hijacking on large network

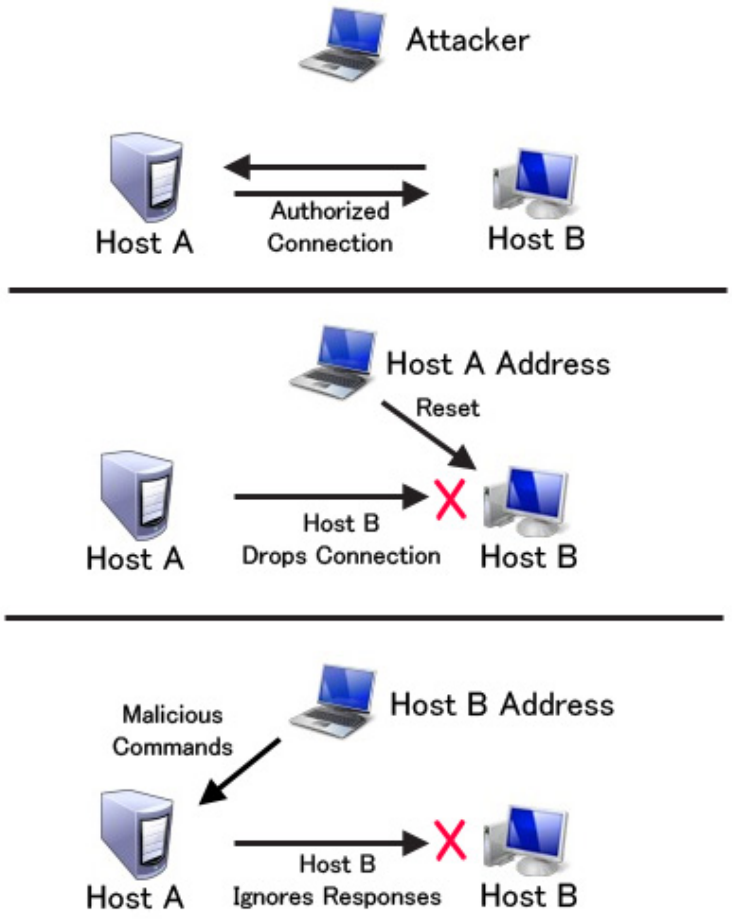


Figure 2: Session Hijacking

More on Packet Sniffers

- Packet sniffers "read" information traversing a network
  - Packet sniffers intercept network packets, possibly using ARP cache poisoning
  - Can be used as legitimate tools to analyze a network
    - Monitor network usage
    - Filter network traffic
    - Analyze network problems
  - Can also be used maliciously
    - Steal information (i.e. passwords, conversations, etc.)
    - Analyze network information to prepare an attack
- Packet sniffers can be either software or hardware based
  - Sniffers are dependent on network setup

Stop Packet Sniffing-I

- The best way is to encrypt packets securely
  - Sniffers can capture the packets, but they are meaningless
    - Capturing a packet is useless if it just reads as garbage
  - SSH is also a much more secure method of connection
    - Private/Public key pairs makes sniffing virtually useless
  - On switched networks, almost all attacks will be via ARP spoofing
    - Add machines to a permanent store in the cache
    - This store cannot be modified via a broadcast reply
    - Thus, a sniffer cannot redirect an address to itself

Stop Packet Sniffing-II

- The best security is to not let them in in the first place
  - Sniffers need to be on your subnet in a switched hub in the first place
  - All sniffers need to somehow access root at some point to start themselves up

User Datagram Protocol-I

- UDP is a stateless, unreliable datagram protocol built on top of IP, that is it lies on level 4

- It does not provide delivery guarantees, or acknowledgments, but is significantly faster

- Can however distinguish data for multiple concurrent applications on a single host.

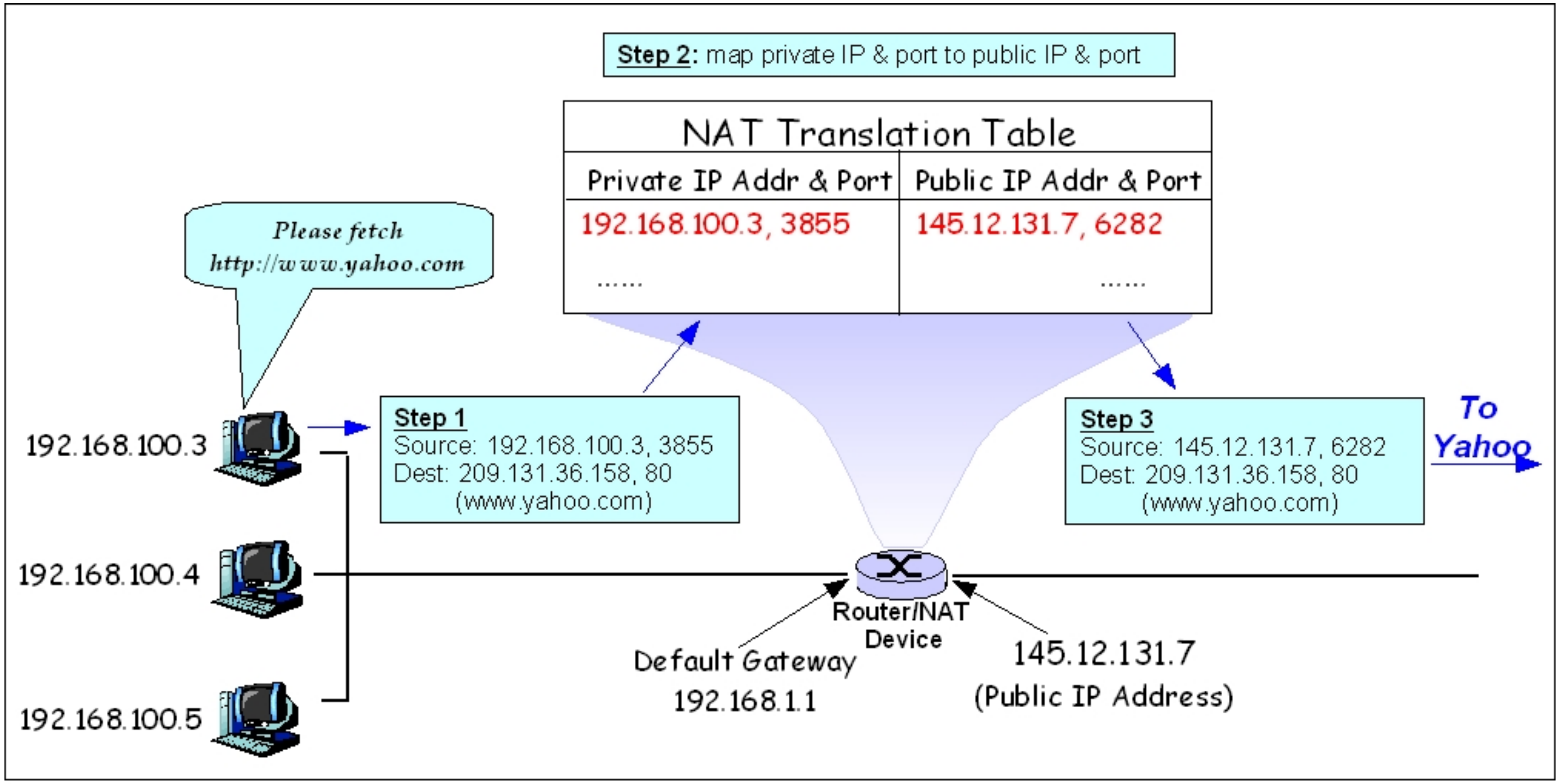
User Datagram Protocol-II

- A lack of reliability implies applications using UDP must be ready to accept a fair amount of error packages and data loss. Some application level protocols such as TFTP (Trivial File Transfer Protocol) build reliability on top of UDP.
  - Most applications used on UDP will suffer if they have reliability. VoIP, Streaming Video and Streaming Audio all use UDP.

- UDP does not come with built in congestion protection, so while UDP does not suffer from the problems associated with optimistic ACK, there are cases where high rate UDP network access will cause congestion.

Network Address Translation

- Introduced in the early 90s to alleviate IPv4 address space congestion
- Relies on translating addresses in an internal network, to an external address that is used for communication to and from the outside world
- NAT is usually implemented by placing a router in between the internal private network and the public network.
- Saves IP address space since not every terminal needs a globally unique IP address, only an organizationally unique one
- While NAT should really be transparent to all high level services, this is sadly not true because a lot of high level communication uses things on IP



Step 2: map private IP & port to public IP & port

NAT Translation Table

Private IP Addr & Port	Public IP Addr & Port
192.168.100.3, 3855	145.12.131.7, 6282
……	……

Please fetch http://www.yahoo.com

Step 1
Source: 192.168.100.3, 3855
Dest: 209.131.36.158, 80
(www.yahoo.com)

Step 3
Source: 145.12.131.7, 6282
Dest: 209.131.36.158, 80
(www.yahoo.com)

To Yahoo

192.168.100.3

192.168.100.4

192.168.100.5

Router/NAT Device

Default Gateway
192.168.1.1

145.12.131.7
(Public IP Address)