Network Security

Network Security Issues

- Confidentiality
  - No requirement in any layer that packet contents to be kept confidential.
  - Encryption could be done at application layer (HTTPS) or revising a lower-layer protocol (IPsec)
- Integrity
  - Simple checksums are used at each layer to determine whether a small number of bits have been altered.
  - Not cryptographically secure
- Availability
  - Scale of the Internet makes network available on a 24/7 basis difficult

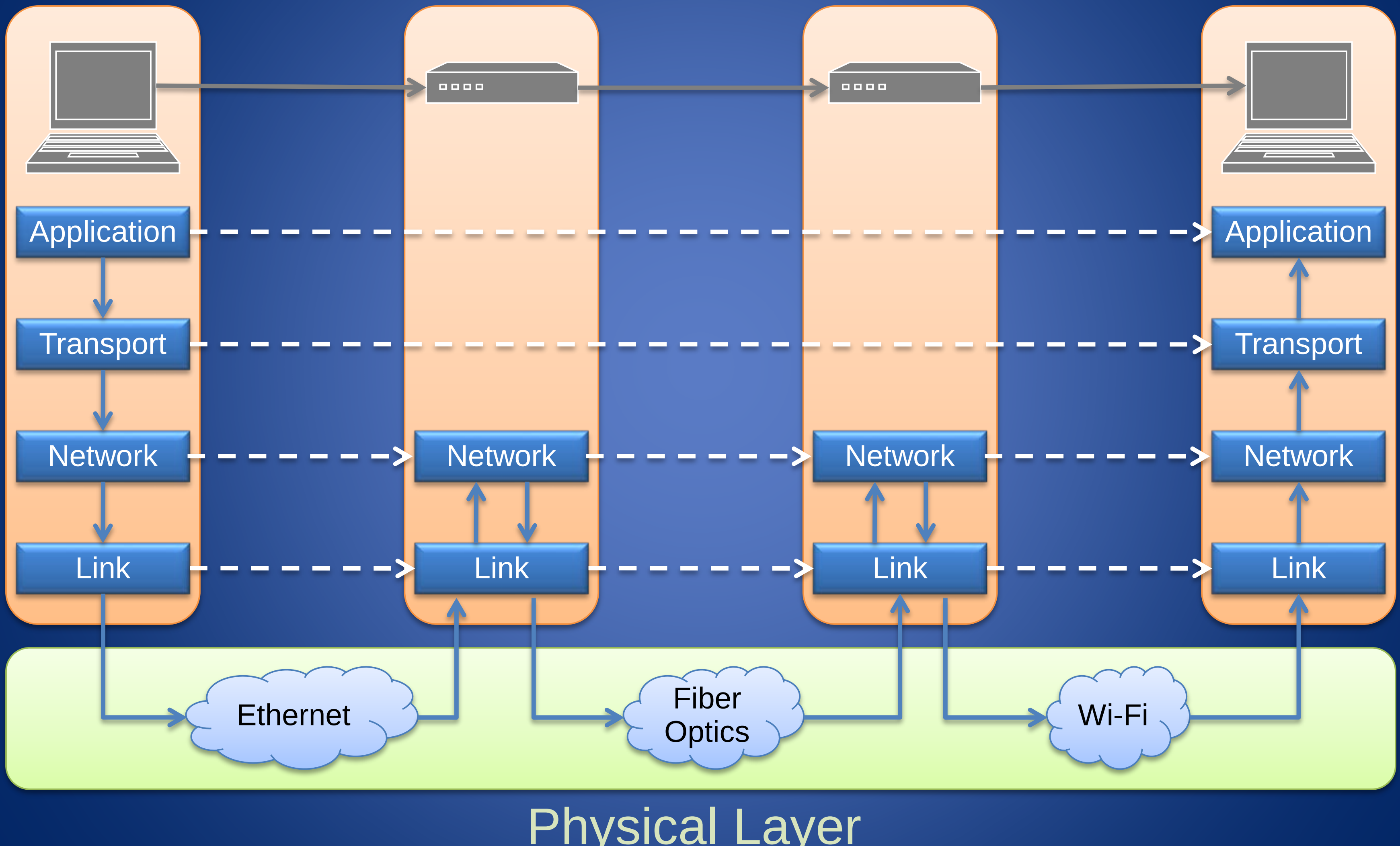
Network Security Issues

- Assurance
  - Permissions and policies to control data flows should be implemented
  - Firewalls to block traffic in and out of a network domain
- Authenticity
  - No digital signatures to identify user ID in standard protocols
  - Identities and signatures to be added at application layer
- Anonymity
  - A built-in anonymity feature of the Internet

Network Layers

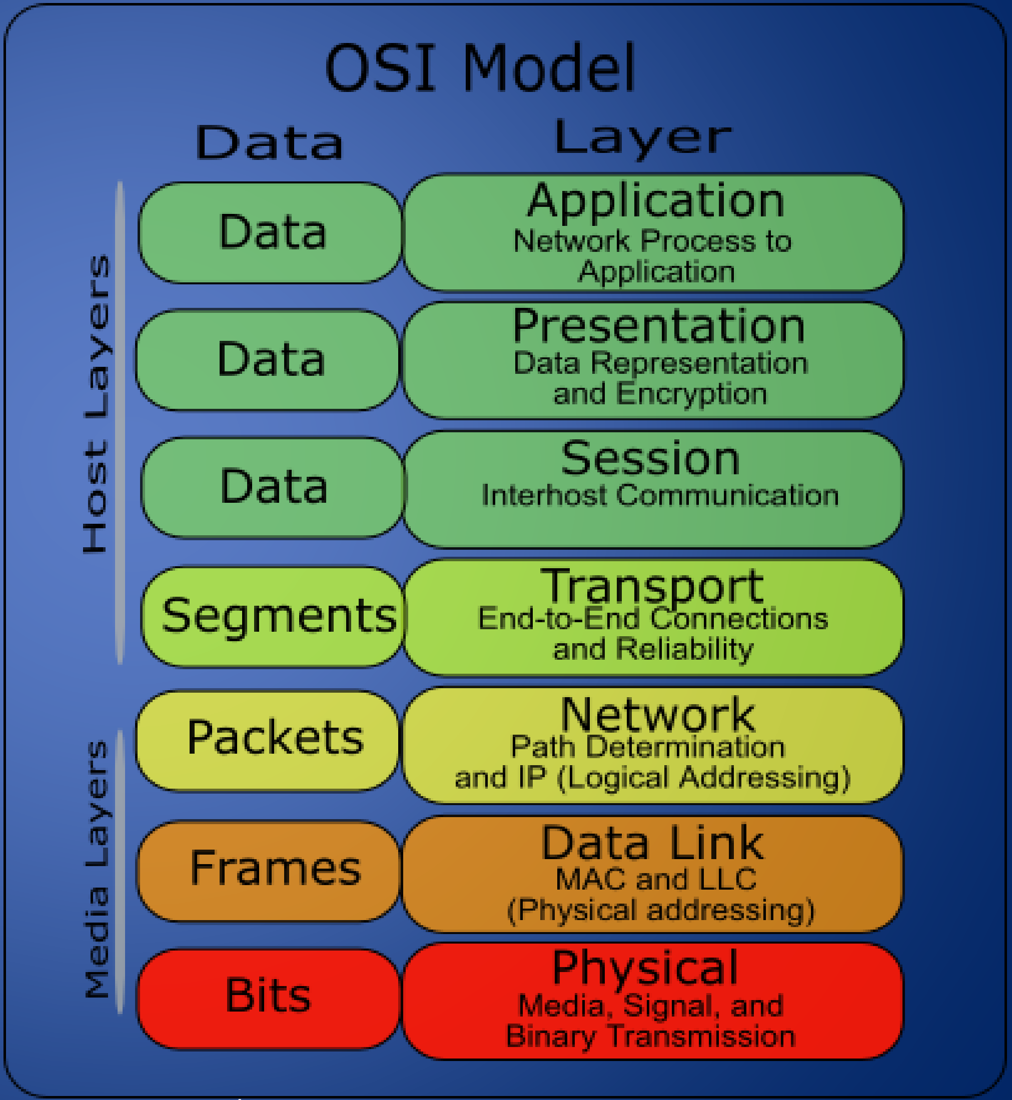
- Network models typically use a stack of layers
  - Higher layers use the services of lower layers via encapsulation
  - A layer can be implemented in hardware or software
  - The bottommost layer must be in hardware
- A network device may implement several layers
- A communication channel between two nodes is established for each layer
  - Actual channel at the bottom layer
  - Virtual channel at higher layers

Internet Layers



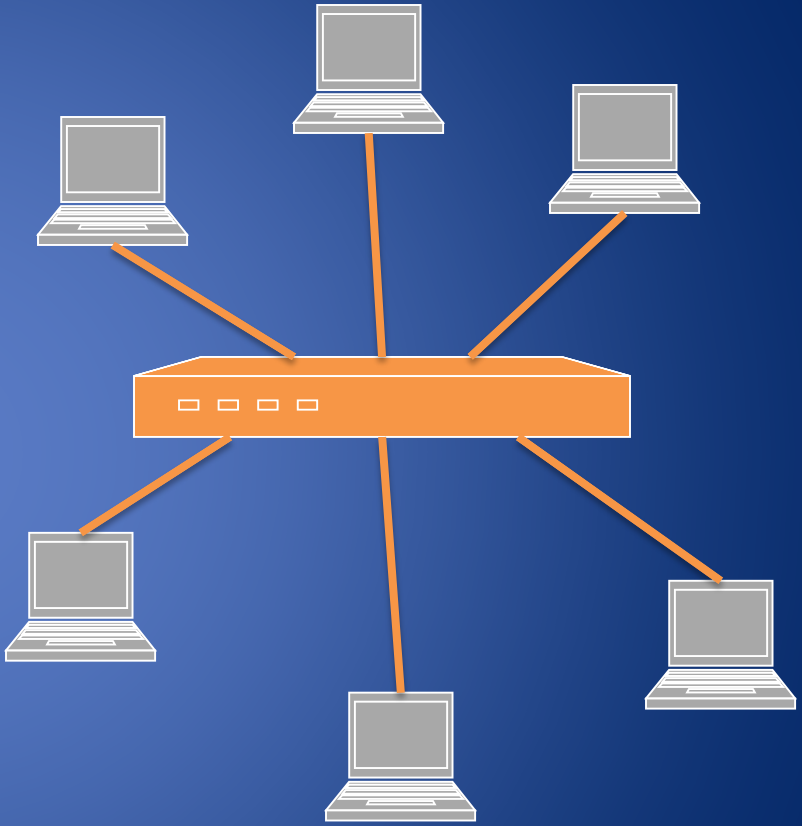
The OSI Model

- The OSI (Open System Interconnect) Reference Model is a network model consisting of seven layers
- Created in 1983, OSI is promoted by the International Standard Organization (ISO)



Switch

- A switch is a common network device
  - Operates at the link layer
  - Has multiple ports, each connected to a computer
- Operation of a switch
  - Learn the MAC address of each computer connected to it
  - Forward frames only to the destination computer



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
- 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

Viewing and Changing MAC Addresses

- 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

ARP

- The address resolution protocol (ARP) connects the network layer to the data 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, 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

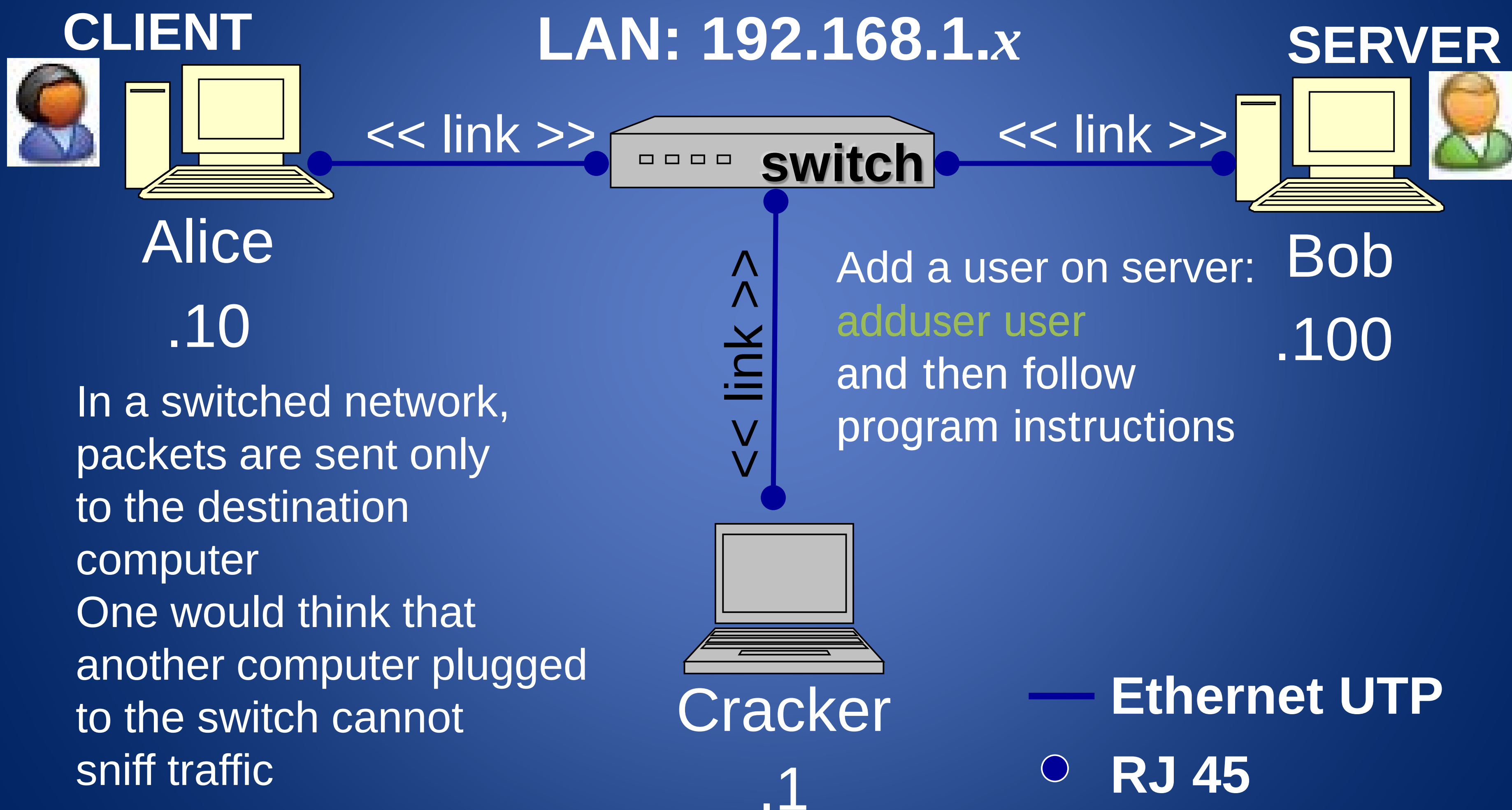
ARP Poisoning (ARP Spoofing)

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

Telnet Protocol (RFC 854)

- Telnet is a protocol that provides a general, bi-directional, not encrypted communication
- **`telnet`** is a generic TCP client
  - Allows a computer to connect to another one
  - Provides remote login capabilities to computers on the Internet
  - Sends whatever you type
  - Prints whatever comes back
  - Useful for testing TCP servers (ASCII based protocols)

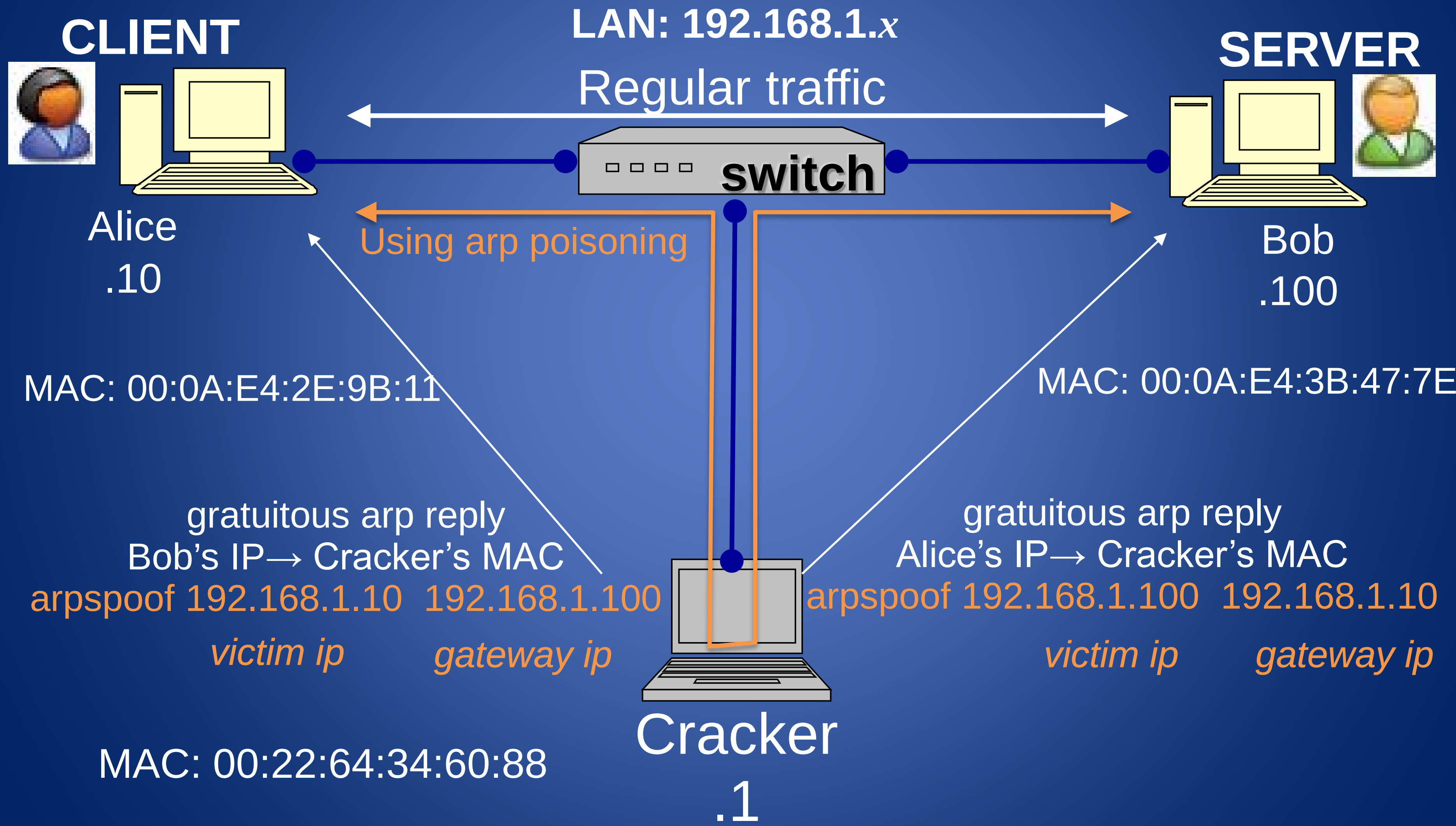
DEMO 1: Configuration using Telnet



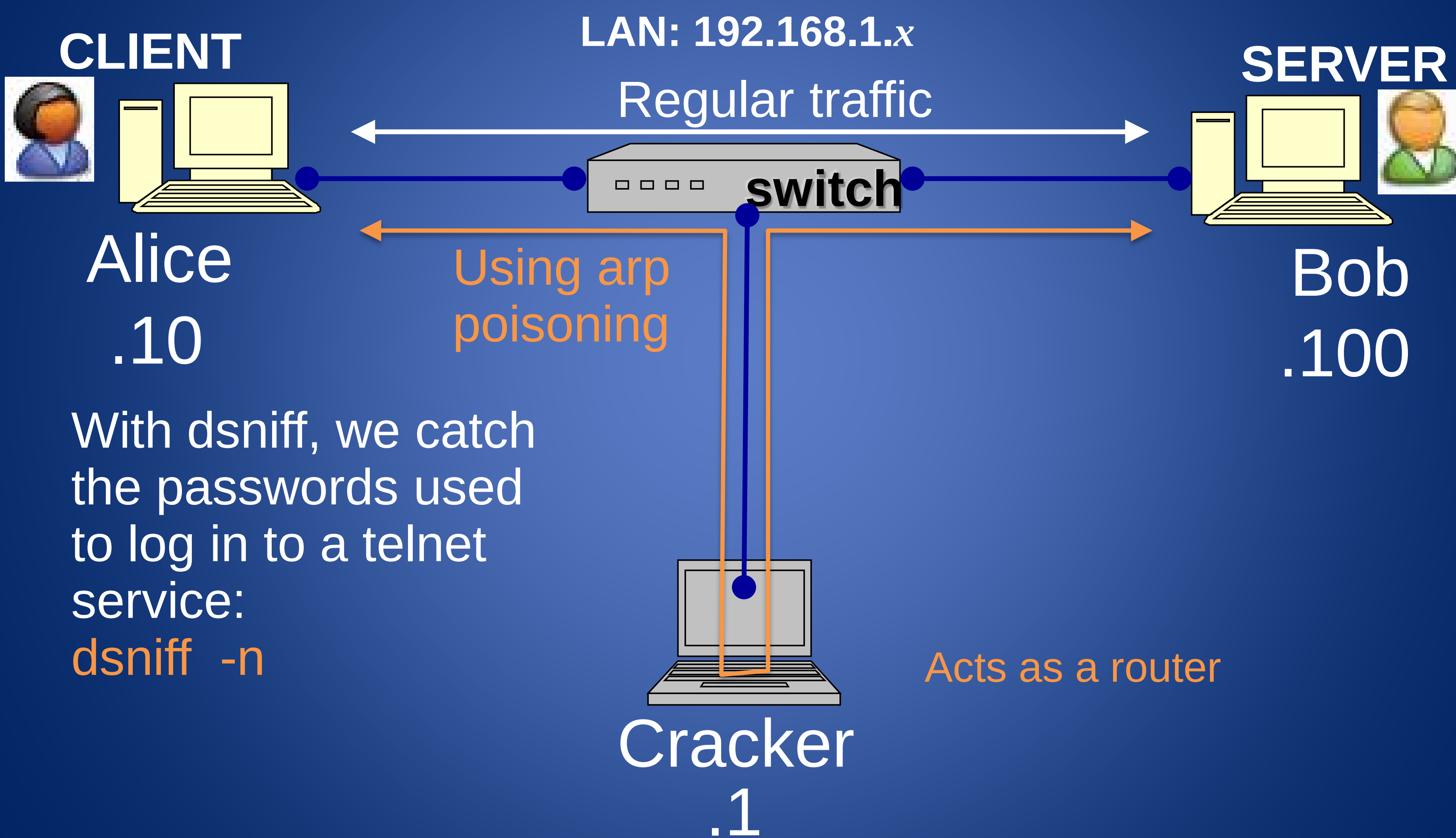
Add a user on server:
adduser user
and then follow program instructions

In a switched network, packets are sent only to the destination computer
One would think that another computer plugged to the switch cannot sniff traffic

DEMO 1: ARP Spoofing



DEMO 1: catch telnet password



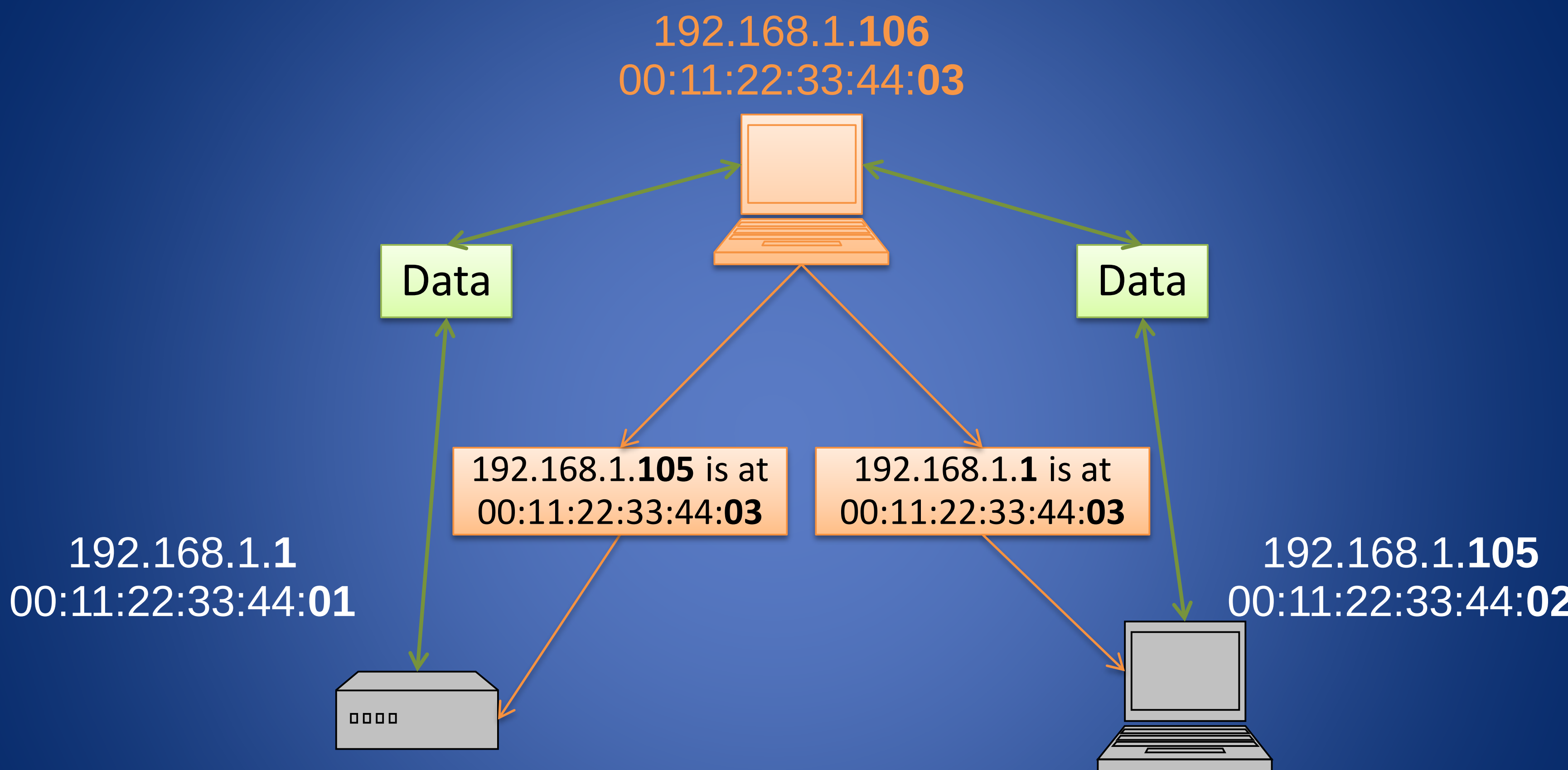
With dsniff, we catch the passwords used to log in to a telnet service:
dsniff -n

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



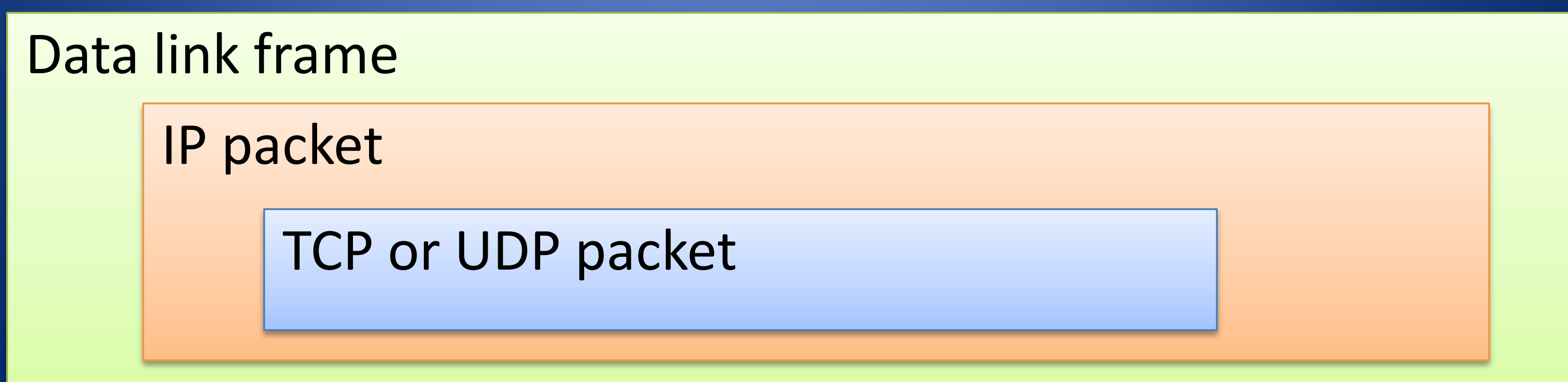
Poisoned ARP Cache	
192.168.1.**105**	00:11:22:33:44:**03**

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

Security Issues with IP and TCP

Internet Protocol

- Connectionless
  - Each packet is transported independently from other packets
- Unreliable
  - Delivery on a best effort basis
  - No acknowledgments
  - Packets may be lost, reordered, corrupted, or duplicated
- IP packets
  - Encapsulate TCP and UDP packets
  - Encapsulated into link-layer frames



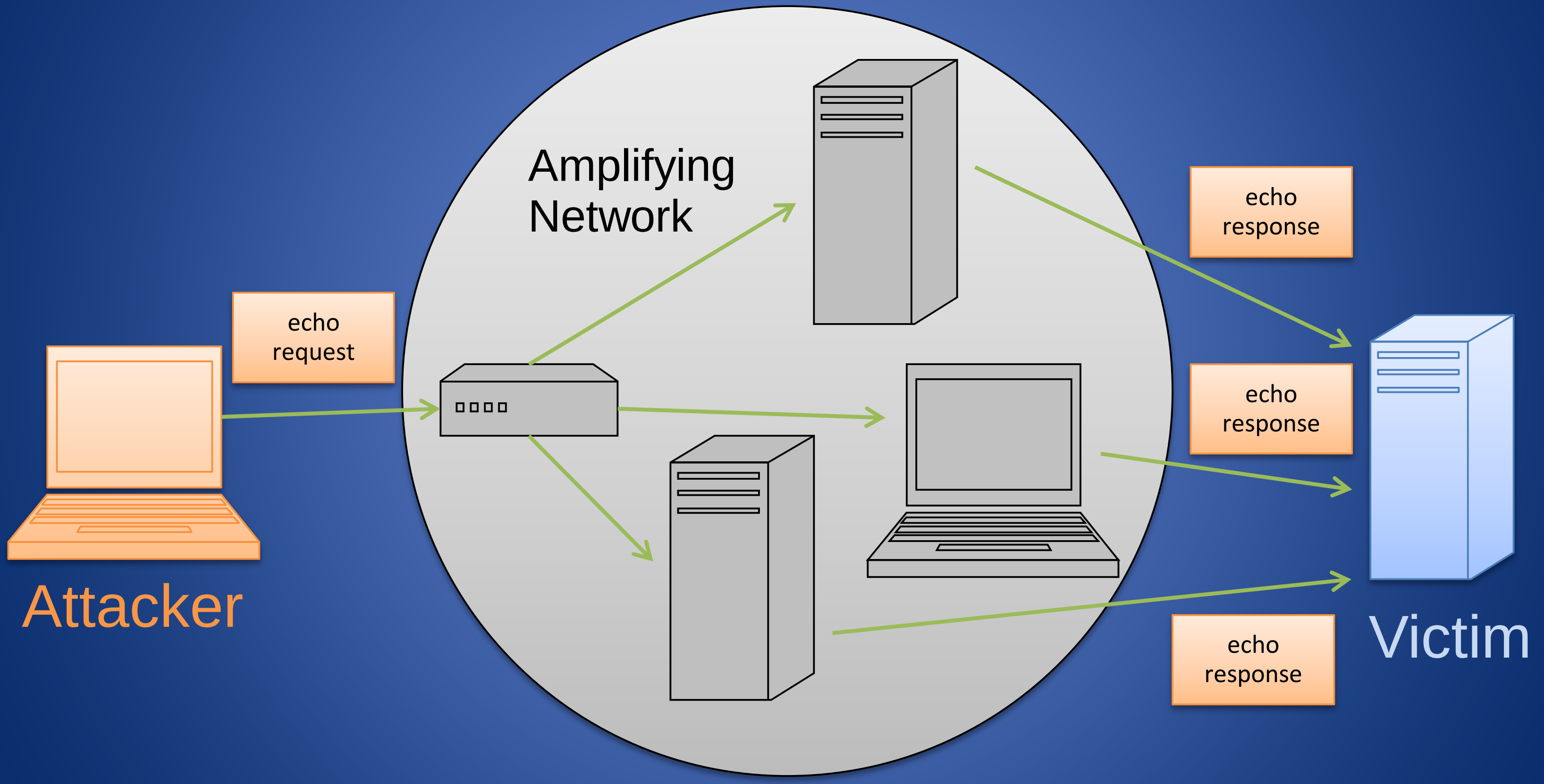
Internet Routes

- Internet Control Message Protocol (ICMP)
  - Used for network testing and debugging
  - Simple messages encapsulated in single IP packets
  - 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

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
- Smurf
  - Ping a broadcast address using a spoofed source address

Smurf Attack



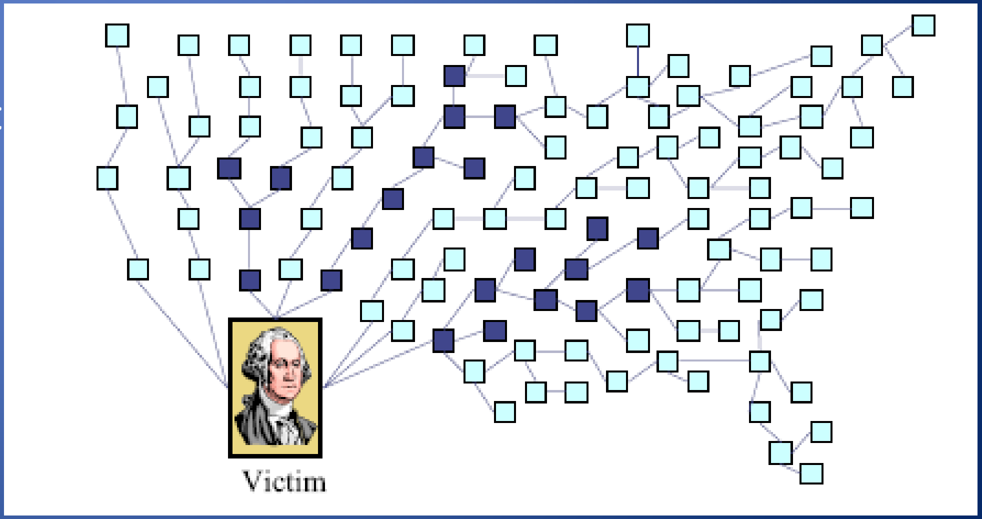
IP Vulnerabilities

- 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
- 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

Denial of Service Attack

- Send large number of packets to host providing service
  - Slows down or crashes host
  - Often executed by botnet
- Attack propagation
  - Starts at zombies
  - Travels through tree of internet routers rooted
  - Ends at victim
- IP source spoofing
  - Hides attacker
  - Scatters return traffic from victim

Source:
M.T. Goodrich, Probabalistic Packet Marking for Large-Scale IP Traceback, IEEE/ACM Transactions on Networking 16:1, 2008.



IP Spoofing

- IP Spoofing is an attempt by an intruder to send packets from one IP address that appear to originate at another
- If the server thinks it is receiving messages from the real source after authenticating a session, it could inadvertently behave maliciously
- There are two basic forms of IP Spoofing
    - Blind Spoofing
        - Attack from any source
    - Non-Blind Spoofing
        - Attack from the same subnet

Blind IP Spoofing

- The TCP/IP protocol requires that “acknowledgement” numbers be sent across sessions
- Makes sure that the client is getting the server’s packets and vice versa
- Need to have the right sequence of acknowledgment numbers to spoof an IP identity

Non-Blind IP Spoofing

- IP Spoofing without inherently knowing the acknowledgment sequence pattern
  - Done on the same subnet
  - Use a packet sniffer to analyze the sequence pattern
    - Packet sniffers intercept network packets
    - Eventually decodes and analyzes the packets sent across the network
    - Determine the acknowledgment sequence pattern from the packets
    - Send messages to server with actual client's IP address and with validly sequenced acknowledgment number

IP Traceback

- Problem
  - How to identify leaves of DoS propagation tree
  - Routers next to attacker
- Issues
  - There are more than 2M internet routers
  - Attacker can spoof source address
  - Attacker knows that traceback is being performed
- Approaches
  - Filtering and tracing (immediate reaction)
  - Messaging (additional traffic)
  - Logging (additional storage)
  - Probabilistic marking

Probabilistic Packet Marking

- Method
  - Random injection of information into packet header
  - Changes seldom used bits
  - Forward routing information to victim
  - Redundancy to survive packet losses
- Benefits
  - No additional traffic
  - No router storage
  - No packet size increase
  - Can be performed online or offline

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

Detecting Sniffers

- Sniffers are almost always passive
  - They simply collect data
  - They do not attempt “entry” to “steal” data
- This can make them extremely hard to detect
- Most detection methods require suspicion that sniffing is occurring
  - Then some sort of “ping” of the sniffer is necessary
  - It should be a broadcast that will cause a response only from a sniffer
- Another solution on switched hubs is ARP watch
  - An ARP watch monitors the ARP cache for duplicate entries of a machine
  - If such duplicates appear, raise an alarm
  - Problem: false alarms
    - Specifically, DHCP networks can have multiple entires for a single machine

Stopping Packet Sniffing

- 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
- 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

Port Knocking

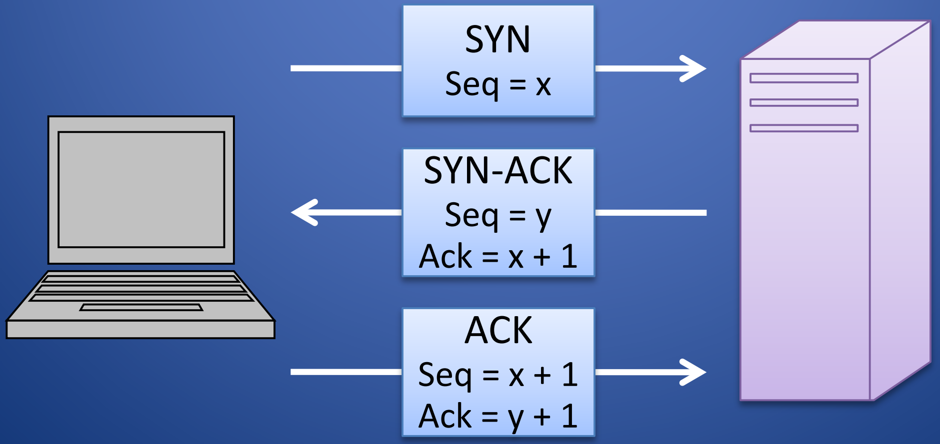
- Broadly port knocking is the act of attempting to make connections to blocked ports in a certain order in an attempt to open a port
- Port knocking is fairly secure against brute force attacks since there are 65536^k combinations, where k is the number of ports knocked
- Port knocking however if very susceptible to replay attacks. Someone can theoretically record port knocking attempts and repeat those to get the same open port again
- One good way of protecting against replay attacks would be a time dependent knock sequence.

Transmission Control Protocol

- 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
- 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

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



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
- 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

TCP Data Transfer

- 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 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

- 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 was 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
- 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

- 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
- 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