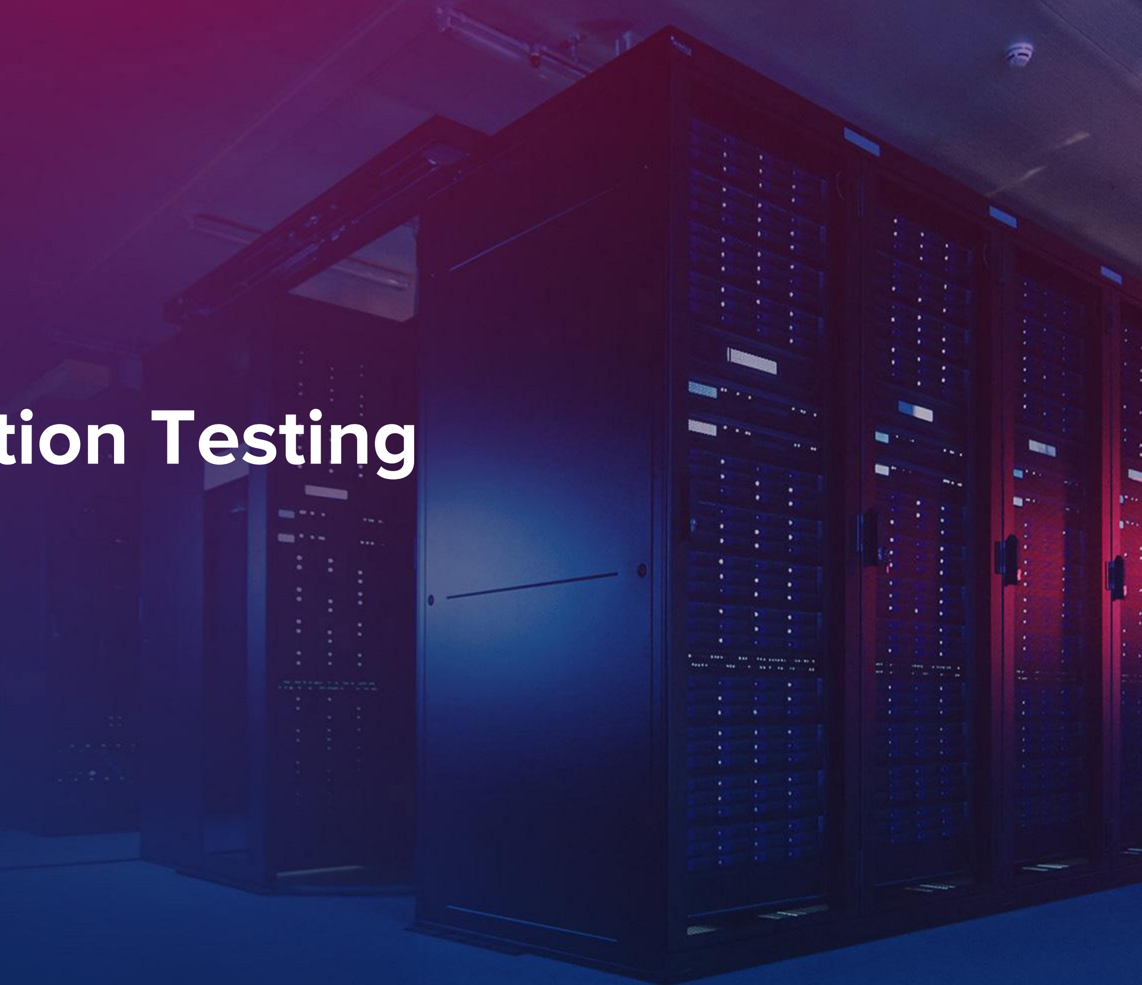
Network Penetration Testing

Course Introduction



Alexis Ahmed

Senior Penetration Tester @HackerSploit

Offensive Security Instructor @INE

Course Topic Overview

- Host Discovery & Port Scanning
- Service Enumeration
- MITM & Network-Based Attacks
- Windows Exploitation & Post-Exploitation
- Linux Exploitation & Post-Exploitation

Prerequisites

- Basic Understanding of Computer Networking
  - Knowledge of IP addresses, subnetting, routing, and network devices (switches, routers, firewalls).
  - Familiarity with common network protocols (TCP, UDP, HTTP, DNS, etc.).
- Fundamentals of Operating Systems
  - Basic knowledge of Windows and Linux operating systems, including their command-line interfaces.
  - Understanding of system processes, file systems, and user permissions.
- Experience with Exploitation and Post-Exploitation
  - Knowledge and experience in exploitation and post-exploitation on Windows and Linux.
  - Ability to target OS specific ports, protocols and services (SMB, RDP, WinRM etc)
  - Ability to identify and exploit vulnerabilities/misconfigurations in Windows and Linux systems.
- Experience with Penetration Testing Tools
  - Some experience using common penetration testing tools (e.g., Metasploit, Nmap, Wireshark).
  - Knowledge and understanding of penetration testing methodologies.

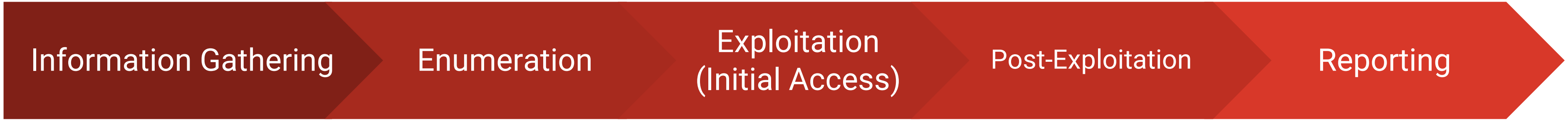
Learning Objectives:

1. Host Discovery & Port Scanning
   - Demonstrate competency in identifying hosts on a target network through various host discovery techniques applicable to both Windows and Linux.
   - Utilize network mapping and port scanning tools to identify open ports on target systems and the services running on the open ports.
2. Service Enumeration
   - Demonstrate competency in enumerating important information from services running on both Windows and Linux systems.
   - Leverage enumeration tools and techniques like Nmap Scripts and other protocol specific tools to enumerate information from specific network protocols (SMB, NetBIOS, SMTP, FTP etc).
3. MITM & Network-Based Attacks
   - Demonstrate competency in performing ARP Spoofing and DNS Spoofing attacks.
   - Demonstrate competency in performing ARP Poisoning and NBT-NS Poisoning attacks.
   - Leverage tools like arpspoof, dnsspoof and Responder to facilitate MITM Attacks.
4. Exploitation & Post-Exploitation
   - Demonstrate competency in exploiting Windows and Linux specific protocols and services for initial access.
   - Demonstrate competency in performing advanced network-based Windows exploitation techniques like SMB Relaying.
   - Demonstrate competency in performing post-exploitation activities on Windows and Linux systems.

Let’s Get Started!

Active Information Gathering

Penetration Testing Methodology



Information Gathering

Enumeration

Exploitation
(Initial Access)

Post-Exploitation

Reporting

<u>Passive Information Gathering</u>
OSINT

<u>Active Information Gathering</u>
Network Mapping
Host Discovery
Port Scanning
Service Detection & OS Detection

<u>Service & OS Enumeration</u>
Service Enumeration
User Enumeration
Share Enumeration

<u>Vulnerability analysis and threat modeling</u>
Vulnerability Analysis
Vulnerability Identification

<u>Exploitation</u>
Developing/Modifying Exploits
Service Exploitation

<u>Post Exploitation</u>
Local Enumeration
Privilege Escalation
Credential Access
Persistence
Defense Evasion
Lateral Movement

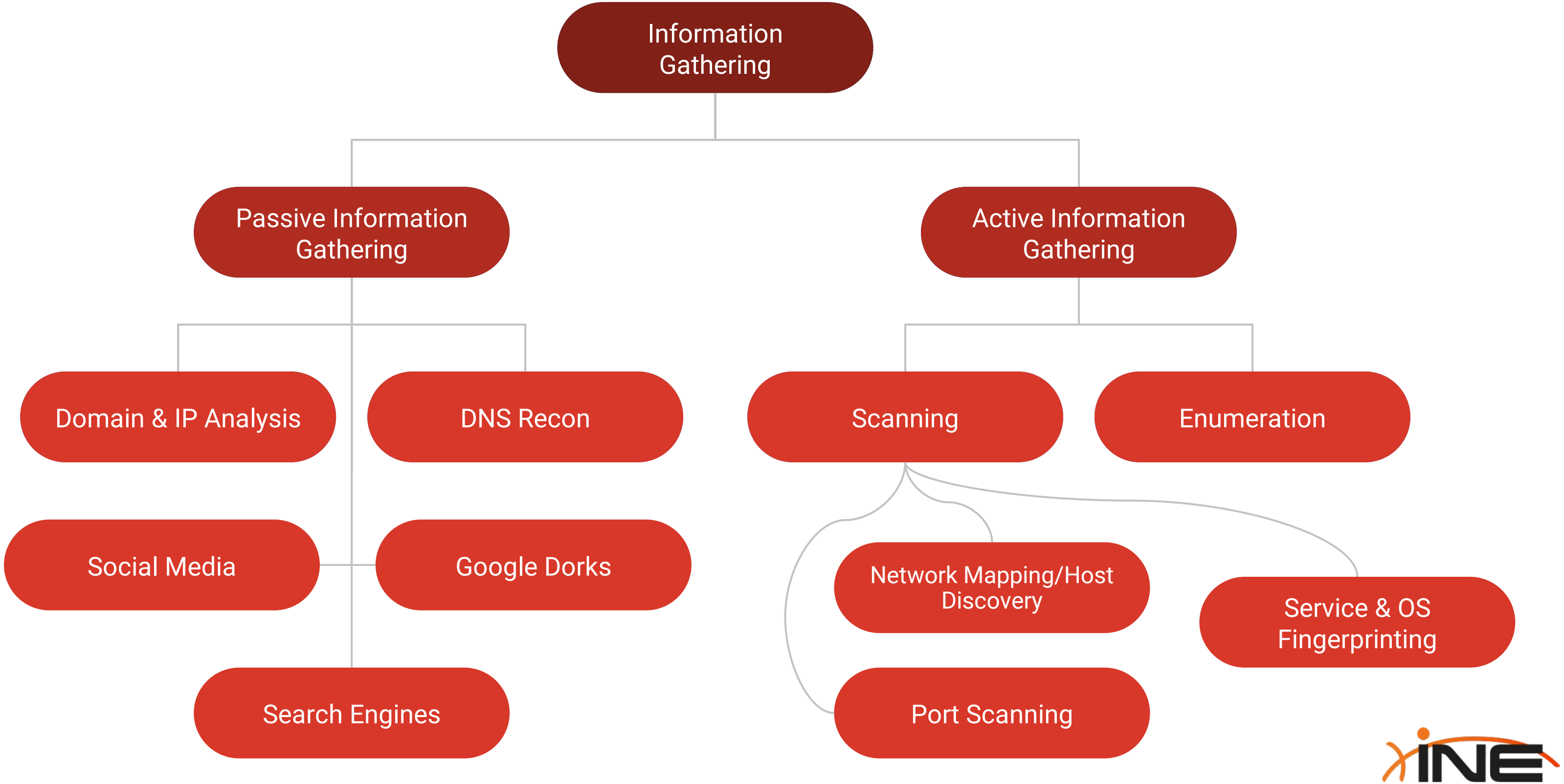
<u>Reporting</u>
Report Writing
Recommendations

Active Information Gathering

- Active information gathering, in the context of penetration testing, refers to the phase of the assessment where the tester actively interacts with the target system or network to collect data and identify potential vulnerabilities.

- This phase involves techniques that go beyond passive reconnaissance (where information is gathered without directly interacting with the target) and may include activities such as scanning, probing, and direct interaction with network services.

Scanning/Network Mapping



Networking Fundamentals

Network Protocols

- In computer networks, hosts communicate with each other through the use of network protocols.

- Network protocols ensure that different computer systems, using different hardware and software can communicate with each other.

- There are a large number of network protocols used by different services for different objectives/functionality.

- Communication between different hosts via protocols is transferred/facilitated through the use of packets.

Packets

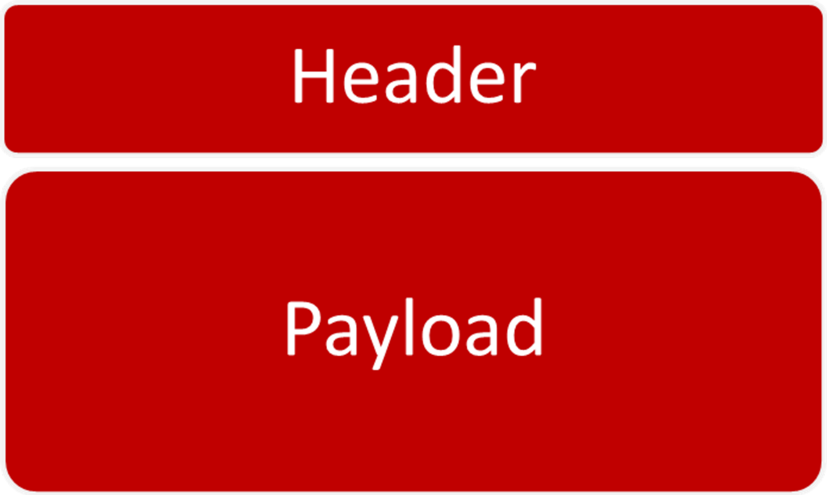
- The primary goal of networking is the exchange information between networked computers; this information is transferred by packets.

- Packets are nothing but streams of bits running as electric signals on physical media used for data transmission. (Ethernet, Wi-Fi etc)

- These electrical signals are then interpreted as bits (zeros and ones) that make up the information.

Packets

Every packet in every protocol has the following structure.

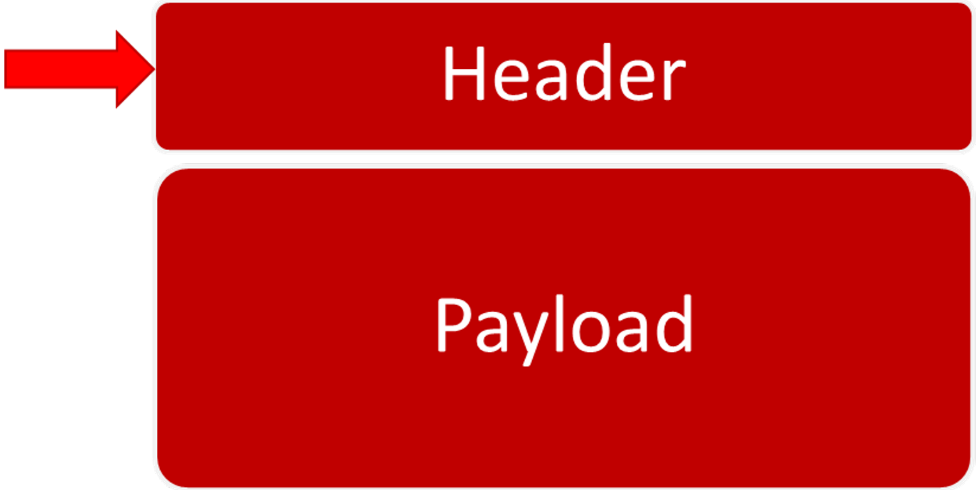


Header

Payload

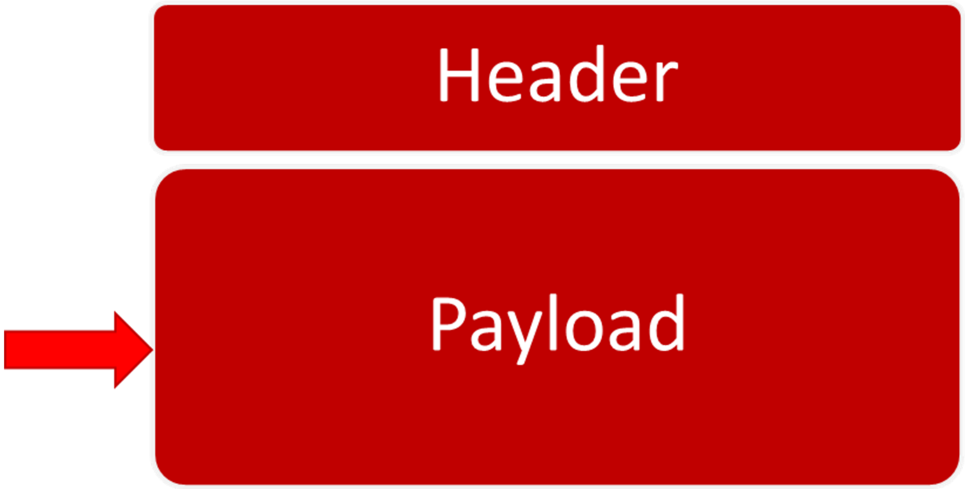
Packets

The header has a protocol-specific structure: this ensures that the receiving host can correctly interpret the payload and handle the overall communication.



Packets

The payload is the actual information being sent . It could be something like part of an email message or the content of a file during a download.



The OSI Model

- The OSI (Open Systems Interconnection) model is a conceptual framework that standardizes the functions of a telecommunication or computing system into seven abstraction layers.
- It was developed by the International Organization for Standardization (ISO) to facilitate communication between different systems and devices, ensuring interoperability and understanding across a wide range of networking technologies.
- The OSI model is divided into seven layers, each representing a specific functionality in the process of network communication.

The OSI Model

#	OSI LAYER	FUNCTION	EXAMPLES
7	APPLICATION LAYER	Provides network services directly to end-users or applications.	HTTP, FTP, IRC, SSH, DNS
6	PRESENTATION LAYER	Translates data between the application layer and lower layers. Responsible for data format translation, encryption, and compression to ensure that data is presented in a readable format.	SSL/TLS, JPEG, GIF, SSH, IMAP
5	SESSION LAYER	Manages sessions or connections between applications. Handles synchronization, dialog control, and token management. (Interhost communication)	APIs, NetBIOS, RPC
4	TRANSPORT LAYER	Ensures end-to-end communication and provides flow control.	TCP, UDP
3	NETWORK LAYER	Responsible for logical addressing and routing.(Logical Addressing)	IP, ICMP, IPSec
2	DATA LINK LAYER	Manages access to the physical medium and provides error detection. Responsible for framing, addressing, and error checking of data frames. (Physical addressing)	Ethernet, PPP, Switches etc
1	PHYSICAL LAYER	Deals with the physical connection between devices.	USB, Ethernet Cables, Coax, Fiber, Hubs etc

The OSI Model

- The OSI model serves as a guideline for developing and understanding network protocols and communication processes.

- While it is a conceptual model, it helps in organizing the complex task of network communication into manageable and structured layers.

- NOTE: The OSI model is not a strict blueprint for every networking system but rather a reference model that aids in understanding and designing network architectures.

Network Layer

Network Layer

- The Network Layer (Layer 3) of the OSI model is responsible for logical addressing, routing, and forwarding data packets between devices across different networks.
- Its primary goal is to determine the optimal path for data to travel from the source to the destination, even if the devices are on separate networks.
- The network layer abstracts the underlying physical network, allowing for the creation of a cohesive internetwork.

Network Layer Protocols

- Several key protocols operate at the network layer (Layer 3) of the OSI model. Here are some prominent network layer protocols:
- Internet Protocol (IP):
  - IPv4 (Internet Protocol version 4): The most widely used version of IP, employing 32-bit addresses and providing the foundation for communication on the Internet.
  - IPv6 (Internet Protocol version 6): Developed to address the limitations of IPv4, it uses 128-bit addresses and offers an exponentially larger address space.
- Internet Control Message Protocol (ICMP):
  - Used for error reporting and diagnostics. ICMP messages include ping (echo request and echo reply), traceroute, and various error messages.

Internet Protocol (IP)

- The Internet Protocol (IP) is a central protocol in the suite of protocols that form the foundation of the Internet.

- It operates at the network layer (Layer 3) of the OSI model and is responsible for logical addressing, routing, and the fragmentation and reassembly of data packets.

- IP enables communication between devices on different networks by providing a standardized way to identify and locate hosts.

Internet Protocol (IP) Versions

- IPv4 (Internet Protocol version 4):
  - IPv4 is the most widely used version of IP and employs 32-bit addresses. Each IPv4 address is represented as four sets of octets separated by dots (e.g., 192.168.0.1).
  - IPv4 provides a finite address space, which has led to the adoption of IPv6 to address the exhaustion of available IPv4 addresses.

- IPv6 (Internet Protocol version 6):
  - IPv6 was developed to overcome the limitations of IPv4 and provides a significantly larger address space using 128-bit addresses.
  - IPv6 addresses are represented in hexadecimal notation (e.g., 2001:0db8:85a3:0000:0000:8a2e:0370:7334).

Internet Protocol (IP) Functionality

- Logical Addressing:
  - IP addresses serve as logical addresses assigned to network interfaces. These addresses uniquely identify each device on a network.
  - IP addresses are hierarchical and structured based on network classes, subnets, and CIDR (Classless Inter-Domain Routing) notation.

- Packet Structure:
  - IP organizes data into packets for transmission across networks. Each packet consists of a header and payload.
  - The header contains essential information, including the source and destination IP addresses, version number, time-to-live (TTL), and protocol type.

Internet Protocol (IP) Functionality

- Fragmentation and Reassembly:
  - IP allows for the fragmentation of large packets into smaller fragments when traversing networks with varying Maximum Transmission Unit (MTU) sizes.
  - The receiving host reassembles these fragments to reconstruct the original packet.
- IP Addressing Types:
  - IP addresses can be classified into three types: unicast (one-to-one communication), broadcast (one-to-all communication within a subnet), and multicast (one-to-many communication to a selected group of devices).
- Subnetting:
  - Subnetting is a technique that divides a large IP network into smaller, more manageable sub-networks. It enhances network efficiency and security.

Internet Protocol (IP) Functionality

- Internet Control Message Protocol (ICMP):
  - ICMP is closely associated with IP and is used for error reporting and diagnostics. Common ICMP messages include echo request and echo reply, which are used in the ping utility.

- Dynamic Host Configuration Protocol (DHCP):
  - DHCP is often used in conjunction with IP to dynamically assign IP addresses to devices on a network, simplifying the process of network configuration.

IP Header Format

- The IP protocol defines many different fields in the packet header. These fields contain binary values that the IPv4 services reference as they forward packets across the network.
  - IP Source Address - Packet Source
  - IP Destination Address - Packet Destination
  - Time-to-Live (TTL) - An 8-bit value that indicates the remaining life of the packet.
  - Type-of-Service (ToS) - The Type-of-Service field contains an 8-bit binary value that is used to determine the priority of each packet.
  - Protocol - This 8-bit value indicates the data payload type that the packet is carrying

IPv4 Header Fields

Field	Purpose
Version (4 bits)	Indicates the version of the IP protocol being used. For IPv4, the value is 4.
Header Length (4 bits)	Specifies the length of the IPv4 header in 32-bit words. The minimum value is 5, indicating a 20-byte header, and the maximum is 15, indicating a 60-byte header.
Type of Service (8 bits)	Originally designed for specifying the quality of service, it includes fields such as Differentiated Services Code Point (DSCP) and Explicit Congestion Notification (ECN) to manage packet priority and congestion control.

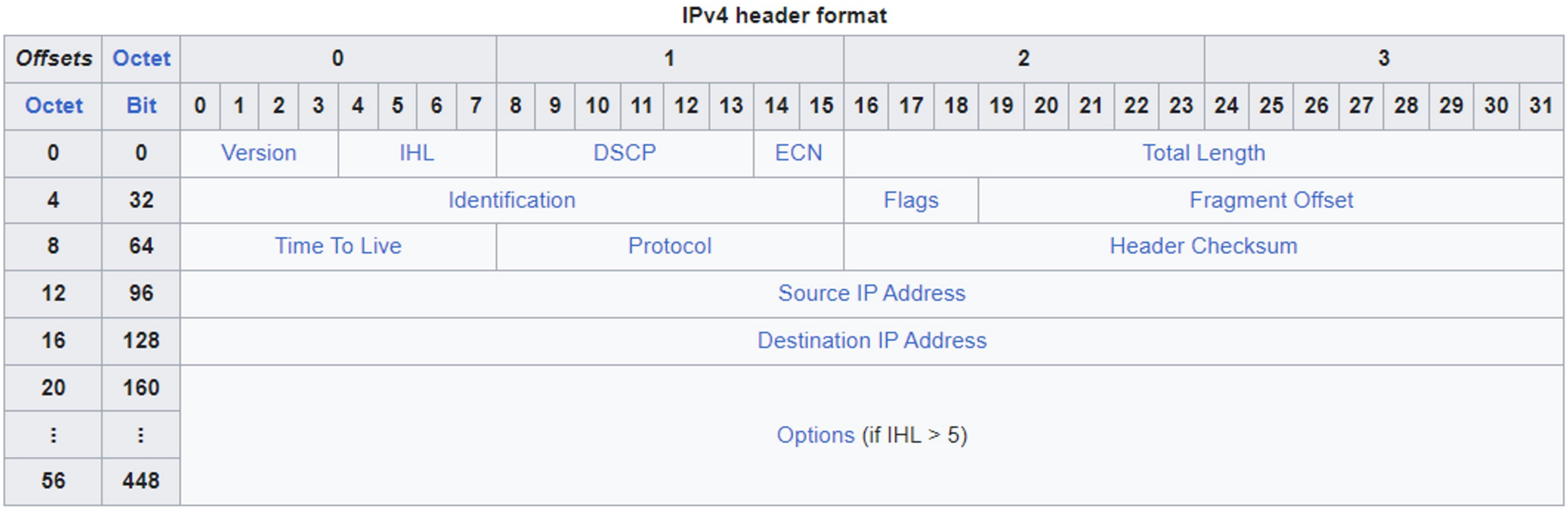
IPv4 Header Fields

Field	Purpose
Total Length (16 bits)	Represents the total size of the IP packet, including both the header and the payload (data). The maximum size is 65,535 bytes.
Identification (16 bits)	Used for reassembling fragmented packets. Each fragment of a packet is assigned the same identification value.
Flags (3 bits)	Includes three flags related to packet fragmentation: Reserved (bit 0): Always set to 0. Don't Fragment (DF, bit 1): If set to 1, indicates that the packet should not be fragmented. More Fragments (MF, bit 2): If set to 1, indicates that more fragments follow in a fragmented packet.

IPv4 Header Fields

Field	Purpose
Time-to-Live (TTL, 8 bits)	Represents the maximum number of hops (routers) a packet can traverse before being discarded. It is decremented by one at each hop.
Protocol (8 bits)	Identifies the higher-layer protocol that will receive the packet after IP processing. Common values include 6 for TCP, 17 for UDP, and 1 for ICMP.
Source IP Address (32 bits)	Specifies the IPv4 address of the sender (source) of the packet.
Destination IP Address (32 bits)	Specifies the IPv4 address of the intended recipient (destination) of the packet.

IP Header Format

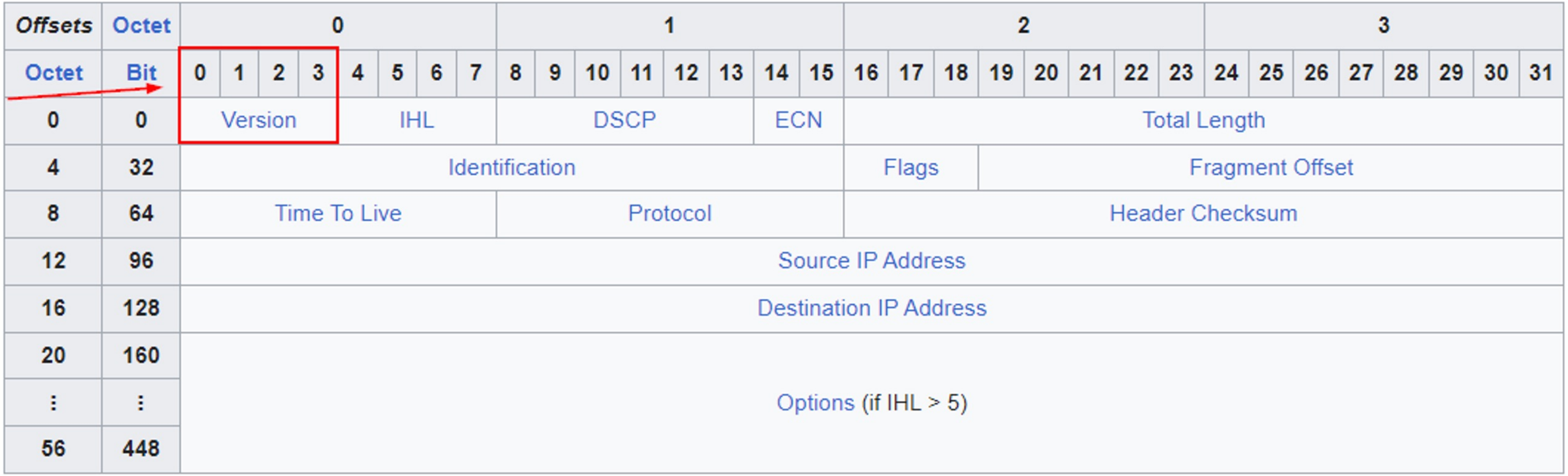


IPv4 header format

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				IHL				DSCP						ECN		Total Length															
4	32	Identification																Flags			Fragment Offset												
8	64	Time To Live								Protocol								Header Checksum															
12	96	Source IP Address																															
16	128	Destination IP Address																															
20	160	Options (if IHL > 5)																															
⋮	⋮																																
56	448																																

IP Header Format

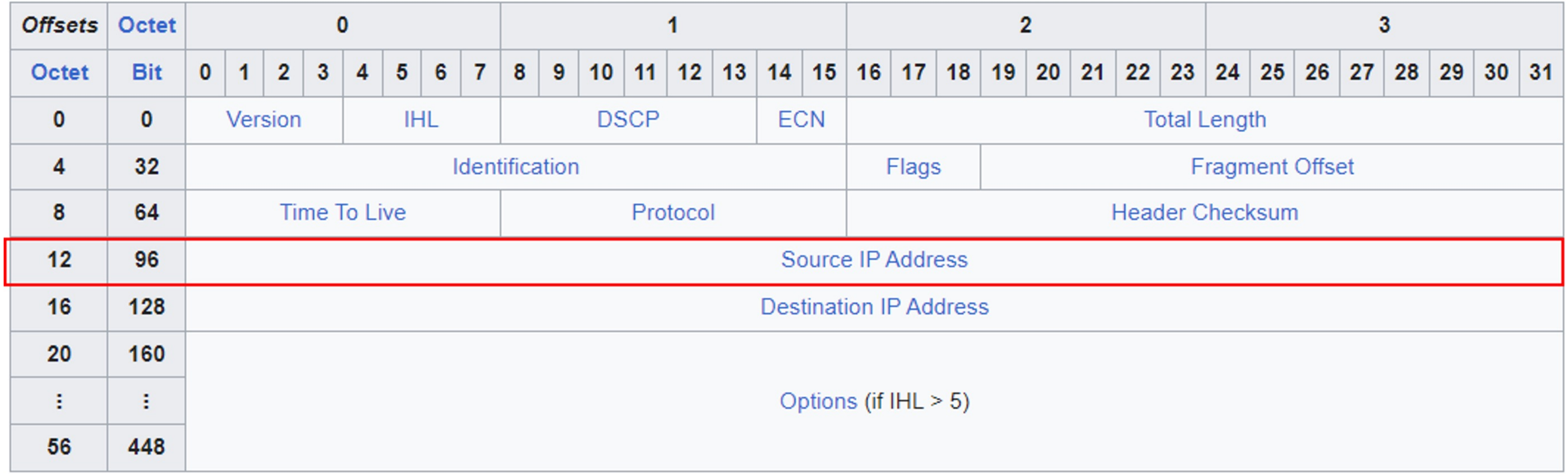
The first four bits identify the IP version. They can be used to represent IP version 4 or 6.



Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				IHL				DSCP						ECN		Total Length															
4	32	Identification																Flags			Fragment Offset												
8	64	Time To Live								Protocol								Header Checksum															
12	96	Source IP Address																															
16	128	Destination IP Address																															
20	160	Options (if IHL > 5)																															
⋮	⋮																																
56	448																																

IP Header Format

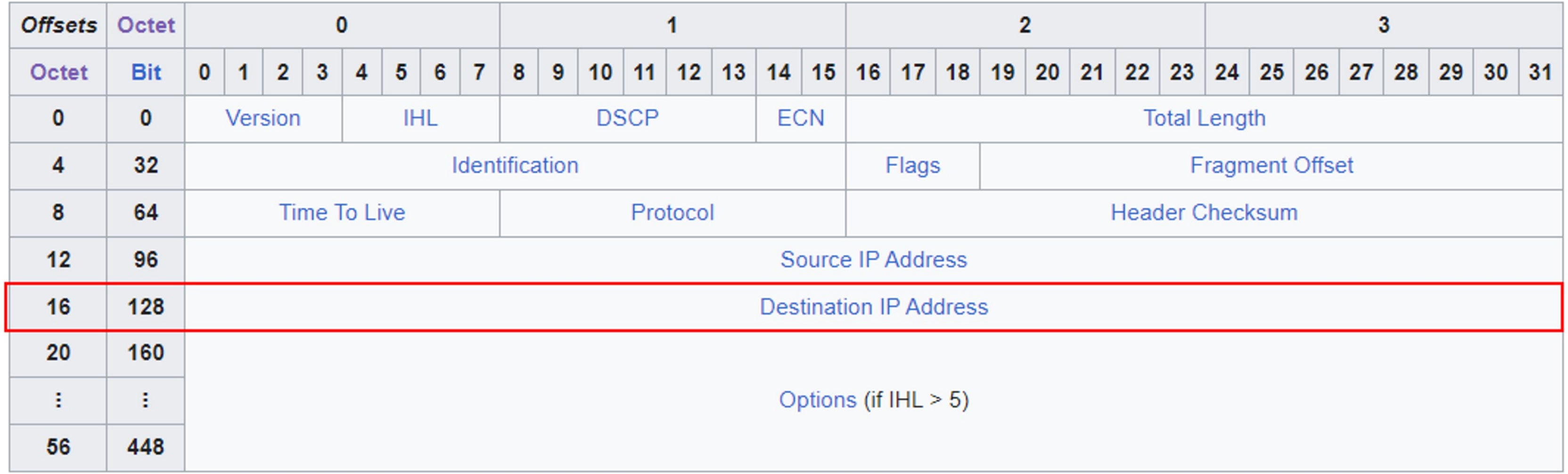
The 32 bits/4 bytes starting at the bit position 96 are allocated for the specification of the source address.



Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				IHL				DSCP						ECN		Total Length															
4	32	Identification																Flags			Fragment Offset												
8	64	Time To Live								Protocol								Header Checksum															
12	96	Source IP Address																															
16	128	Destination IP Address																															
20	160	Options (if IHL > 5)																															
⋮	⋮																																
56	448																																

IP Header Format

The following four bytes represent the destination address.



Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				IHL				DSCP						ECN		Total Length															
4	32	Identification																Flags			Fragment Offset												
8	64	Time To Live								Protocol								Header Checksum															
12	96	Source IP Address																															
16	128	Destination IP Address																															
20	160	Options (if IHL > 5)																															
⋮	⋮																																
56	448																																

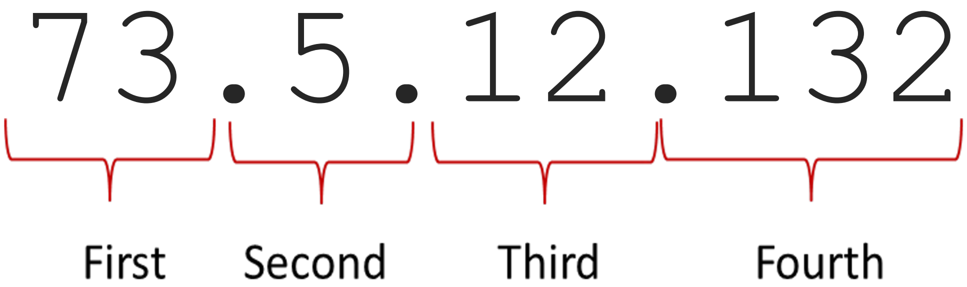
IPv4 Addresses

- The vast majority of networks run IP version 4 (IPv4).
- An IPv4 address consists of four bytes, or octets; a byte consists of 8 bits.

73.5.12.132

IPv4 Addresses

- A dot delimits every octet in the address.



73.5.12.132

First Second Third Fourth

Reserved Ipv4 Addresses

- For example, some reserved intervals are:
  - 0.0.0.0 – 0.255.255.255 representing "this" network.
  - 127.0.0.0 – 127.255.255.255 representing the local host (e.g., your computer).
  - 192.168.0.0 – 192.168.255.255 is reserved for private networks.

- You can find the details about the special use of IPv4 addresses in RFC5735.

Demo: IP Header Analysis

Transport Layer

Transport Layer

- The Transport Layer is the fourth layer of the OSI (Open Systems Interconnection) model, and it plays a crucial role in facilitating communication between two devices across a network.

- This layer ensures reliable, end-to-end communication, handling tasks such as error detection, flow control, and segmentation of data into smaller units.

- The Transport Layer, is responsible for providing end-to-end communication and ensuring the reliable and ordered delivery of data between two devices on a network.

Transport Layer Protocols

- TCP (Transmission Control Protocol): Connection-oriented protocol providing reliable and ordered delivery of data.

- UDP (User Datagram Protocol): Connectionless protocol that is faster but provides no guarantees regarding the order or reliability of data delivery.

TCP

- TCP, or Transmission Control Protocol, is one of the main protocols operating at the Transport Layer (Layer 4) of the OSI model.

- It is a connection-oriented, reliable protocol that provides a dependable and ordered delivery of data between two devices over a network.

- TCP ensures that data sent from one application on a device is received accurately and in the correct order by another application on a different device.

TCP

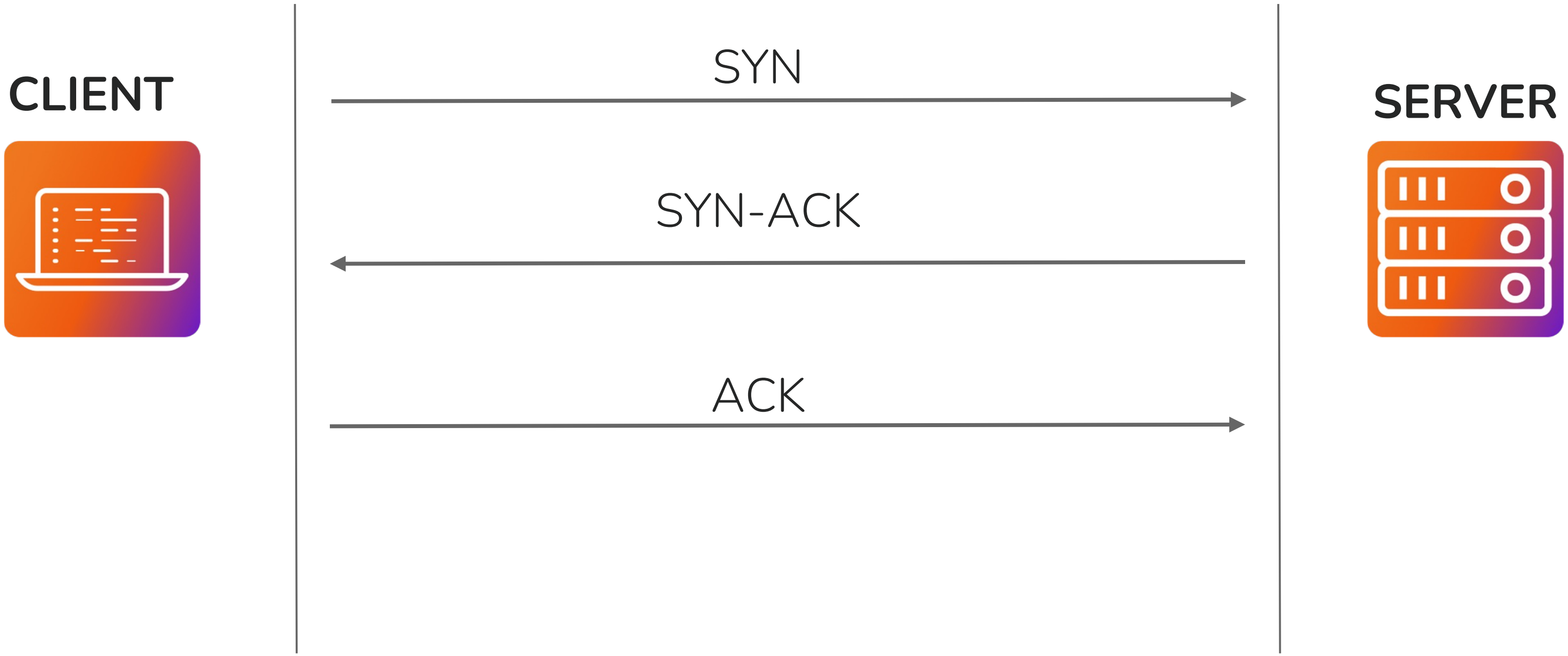
- Connection-Oriented:
  - TCP establishes a connection between the sender and receiver before any data is exchanged. This connection is a virtual circuit that ensures reliable and ordered data transfer.
- Reliability:
  - TCP guarantees reliable delivery of data. It achieves this through mechanisms such as acknowledgments (ACK) and retransmission of lost or corrupted packets. If a segment of data is not acknowledged, TCP automatically resends the segment.
- Ordered Data Transfer:
  - TCP ensures that data is delivered in the correct order. If segments of data arrive out of order, TCP reorders them before passing them to the higher-layer application.

TCP 3-Way Handshake

- The TCP three-way handshake is a process used to establish a reliable connection between two devices before they begin data transmission.

- It involves a series of three messages exchanged between the sender (client) and the receiver (server).

TCP 3-Way Handshake



TCP 3-Way Handshake

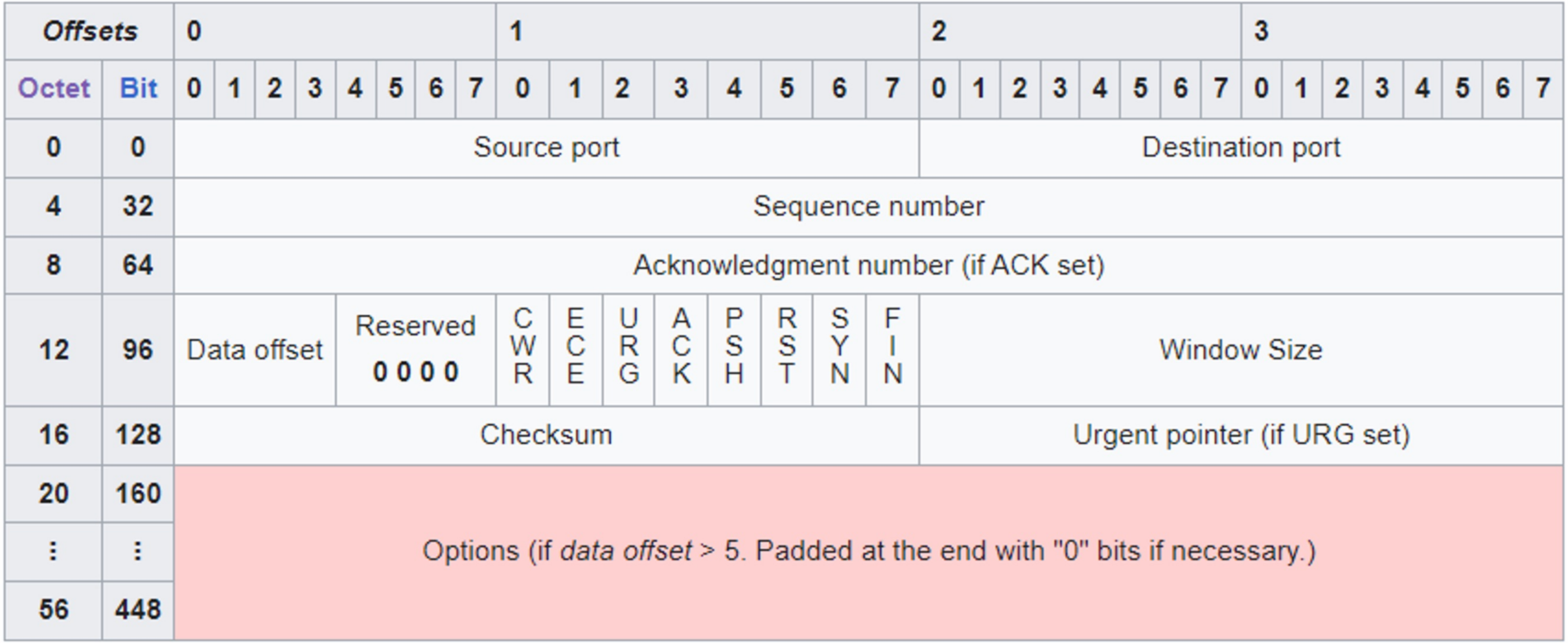
- SYN (Synchronize): The process begins with the client sending a TCP segment with the SYN (Synchronize) flag set. This initial message indicates the client's intention to establish a connection and includes an initial sequence number (ISN), which is a randomly chosen value.

- SYN-ACK (Synchronize-Acknowledge): Upon receiving the SYN segment, the server responds with a TCP segment that has both the SYN and ACK (Acknowledge) flags set. The acknowledgment (ACK) number is set to one more than the initial sequence number received in the client's SYN segment. The server also generates its own initial sequence number.

TCP 3-Way Handshake

- ACK (Acknowledge): Finally, the client acknowledges the server's response by sending a TCP segment with the ACK flag set. The acknowledgment number is set to one more than the server's initial sequence number.
- At this point, the connection is established, and both devices can begin transmitting data.
- After the three-way handshake is complete, the devices can exchange data in both directions. The acknowledgment numbers in subsequent segments are used to confirm the receipt of data and to manage the flow of information.

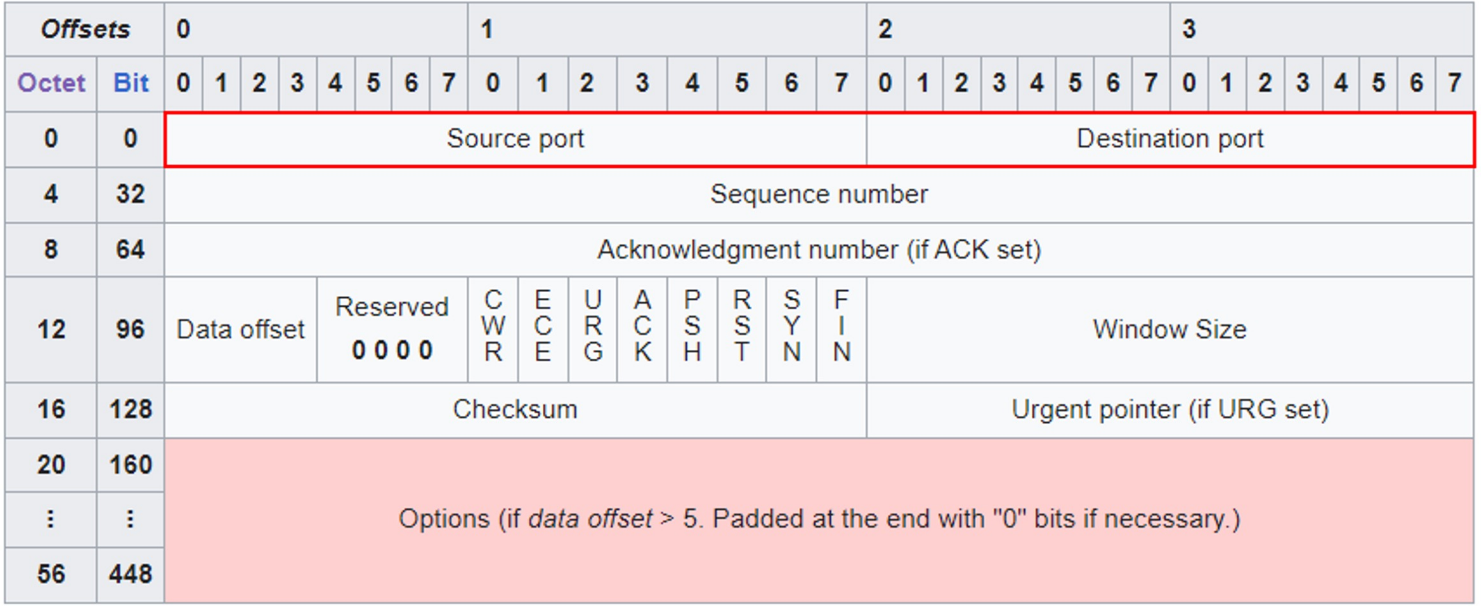
TCP Header Fields



Offsets		0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
0	0	Source port																Destination port															
4	32	Sequence number																															
8	64	Acknowledgment number (if ACK set)																															
12	96	Data offset				Reserved 0 0 0 0				CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	Window Size															
16	128	Checksum																Urgent pointer (if URG set)															
20	160	Options (if *data offset* > 5. Padded at the end with "0" bits if necessary.)																															
⋮	⋮																																
56	448																																

TCP Header Fields

- The SRC(16 bits) & DST(16 bits) Port identifies the source and destination port.



Offsets		0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
0	0	Source port																Destination port															
4	32	Sequence number																															
8	64	Acknowledgment number (if ACK set)																															
12	96	Data offset				Reserved 0 0 0 0				CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	Window Size															
16	128	Checksum																Urgent pointer (if URG set)															
20	160	Options (if *data offset* > 5. Padded at the end with "0" bits if necessary.)																															
⋮	⋮																																
56	448																																

TCP Control Flags

- TCP (Transmission Control Protocol) uses a set of control flags to manage various aspects of the communication process.

- These flags are included in the TCP header and control different features during the establishment, maintenance, and termination of a TCP connection.

TCP Control Flags

- Establishing a Connection:
  - SYN (Set): Initiates a connection request.
  - ACK (Clear): No acknowledgment yet.
  - FIN (Clear): No termination request.

- Establishing a Connection (Response):
  - SYN (Set): Acknowledges the connection request.
  - ACK (Set): Acknowledges the received data.
  - FIN (Clear): No termination request.

TCP Control Flags

- Terminating a Connection:
  - SYN (Clear): No connection request.
  - ACK (Set): Acknowledges the received data.
  - FIN (Set): Initiates connection termination.

TCP Port Range

- TCP (Transmission Control Protocol) uses port numbers to distinguish between different services or applications on a device.

- Port numbers are 16-bit unsigned integers, and they are divided into three ranges.

- The maximum port number in the TCP/IP protocol suite is 65,535.

TCP Port Range

- Well-Known Ports (0-1023): Port numbers from 0 to 1023 are reserved for well-known services and protocols. These are standardized by the Internet Assigned Numbers Authority (IANA). Examples include:
  - 80: HTTP (Hypertext Transfer Protocol)
  - 443: HTTPS (HTTP Secure)
  - 21: FTP (File Transfer Protocol)
  - 22: SSH (Secure Shell)
  - 25: SMTP (Simple Mail Transfer Protocol)
  - 110: POP3 (Post Office Protocol version 3)

TCP Port Range

- Registered Ports (1024-49151): Port numbers from 1024 to 49151 are registered for specific services or applications. These are typically assigned by the IANA to software vendors or developers for their applications. While they are not standardized, they are often used for well-known services. Examples include:
  - 3389: Remote Desktop Protocol (RDP)
  - 3306: MySQL Database
  - 8080: HTTP alternative port
  - 27017: MongoDB Database

UDP

- UDP, or User Datagram Protocol, is a connectionless and lightweight transport layer protocol that provides a simple and minimalistic way to transmit data between devices on a network.

- UDP does not establish a connection before sending data and does not provide the same level of reliability and ordering guarantees. Instead, it focuses on simplicity and efficiency, making it suitable for certain types of applications.

UDP

- Connectionless: UDP is a connectionless protocol, meaning that it does not establish a connection before sending data. Each UDP packet (datagram) is treated independently, and there is no persistent state maintained between sender and receiver.

- Unreliable: UDP does not provide reliable delivery of data. It does not guarantee that packets will be delivered, and there is no mechanism for retransmission of lost packets. This lack of reliability makes UDP faster but less suitable for applications that require guaranteed delivery.

UDP

- Used for Real-Time Applications: UDP is commonly used in real-time applications where low latency is crucial, such as audio and video streaming, online gaming, and voice-over-IP (VoIP) communication.

- Simple and Stateless: UDP is a stateless protocol, meaning that it does not maintain any state information about the communication.

- Each UDP packet is independent of previous or future packets.

TCP vs UDP

Feature	UDP	TCP
Connection	Connectionless	3-Way Handshake
Reliability	Unreliable, no guaranteed delivery of packets	Reliable, guarantees delivery and order of packets and supports retransmission
Header Size	Smaller header size, lower overhead	Larger header size
Applications	VOIP, streaming, gaming	HTTP, Email
Examples	DNS, DHCP, SNMP, VoIP (e.g., SIP), online gaming.	HTTP, FTP, Telnet, SMTP (email), HTTPS.

Demo: TCP 3-Way Handshake

Network Mapping

Network Mapping

- After collecting information about a target organization during the passive information gathering stage, a penetration tester typically moves on to active information gathering phase which involves discovering hosts on a network, performing port scanning and enumeration.

- As you know, every host connected to the Internet or a private network must have a unique IP address that uniquely identifies it on said network.

- How can a penetration tester determine what hosts, within an in-scope network are online? what ports are open on the active hosts? and what operating systems are running on the active hosts? Answer - Network Mapping.

Network Mapping

- Network mapping in the context of penetration testing (pentesting) refers to the process of discovering and identifying devices, hosts, and network infrastructure elements within a target network.

- Pentesters use network mapping as a crucial initial step to gather information about the network's layout, understand its architecture, and identify potential entry points for further exploitation.

Example - Why Map a Network?

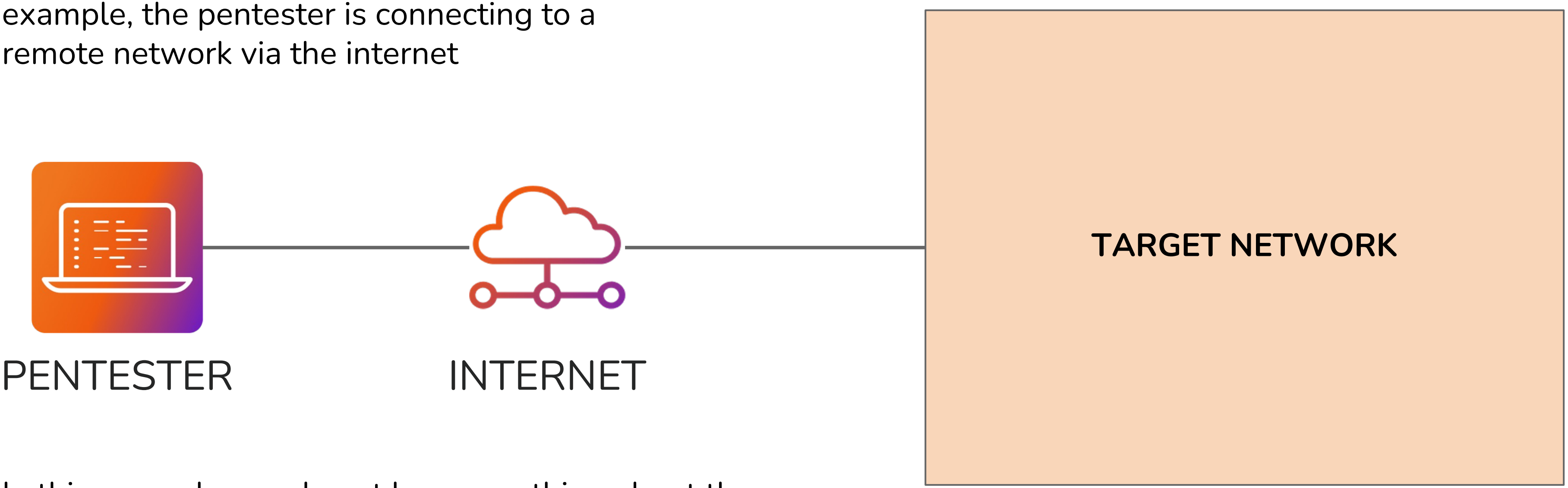
- A company asks for you/your company to perform a penetration test, and the following address block is considered in scope: 200.200.0.0/16.

- A sixteen-bit long netmask means the network could contain up to 216 (65536) hosts with IP addresses in the 200.200.0.0 - 200.200.255.255 range.

- The first job for the penetration tester will involve determining which of the 65536 IP addresses are assigned to a host, and which of those hosts are online/active.

Example - Why Map a Network?

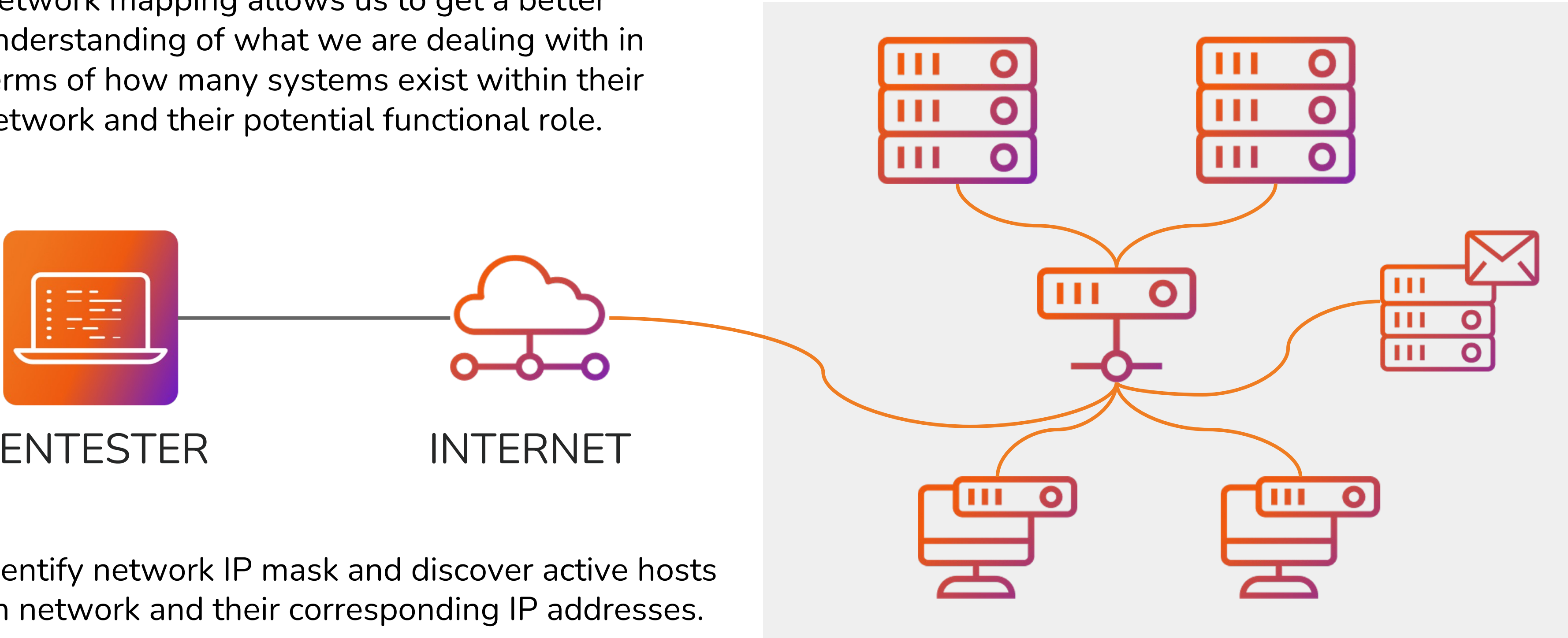
We need a way to map out an unknown network into something more useful, In this example, the pentester is connecting to a remote network via the internet



In this example, we do not know anything about the target network, the objective of network mapping is to develop a picture of the network architecture and the systems that make up the network as a whole.

Example - Why Map a Network?

Network mapping allows us to get a better understanding of what we are dealing with in terms of how many systems exist within their network and their potential functional role.



Identify network IP mask and discover active hosts on network and their corresponding IP addresses.

Network Mapping Objectives

- Discovery of Live Hosts: Identifying active devices and hosts on the network. This involves determining which IP addresses are currently in use.
- Identification of Open Ports and Services: Determining which ports are open on the discovered hosts and identifying the services running on those ports. This information helps pentesters understand the attack surface and potential vulnerabilities.
- Network Topology Mapping: Creating a map or diagram of the network topology, including routers, switches, firewalls, and other network infrastructure elements. Understanding the layout of the network assists in planning further penetration testing activities.

Network Mapping Objectives

- Operating System Fingerprinting: Determining the operating systems running on discovered hosts. Knowing the operating system helps pentesters tailor their attack strategies to target vulnerabilities specific to that OS.
- Service Version Detection: Identifying specific versions of services running on open ports. This information is crucial for pinpointing vulnerabilities associated with particular service versions.
- Identifying Filtering and Security Measures: Discovering firewalls, intrusion prevention systems, and other security measures in place. This helps pentesters understand the network's defenses and plan their approach accordingly.

Nmap (Network Mapper)

- Nmap, or Network Mapper, is an open-source network scanning tool used for discovering hosts and services on a computer network, finding open ports, and identifying potential vulnerabilities.

- It is a powerful and versatile tool that has become a standard in the toolkit of security professionals, network administrators, and penetration testers.

- Nmap offers a range of features and functionalities that make it a valuable tool in various network security contexts:

Nmap Functionality

- Host Discovery: Nmap can identify live hosts on a network using techniques such as ICMP echo requests, ARP requests, or TCP/UDP probes.
- Port Scanning: It can perform various types of port scans to discover open ports on target hosts.
- Service Version Detection: Nmap can determine the versions of services running on open ports. This information helps in understanding the software stack and potential vulnerabilities associated with specific versions.
- Operating System Fingerprinting:Nmap can attempt to identify the operating systems of target hosts based on characteristics observed during the scanning process.

Host Discovery Techniques

Host Discovery

- In penetration testing, host discovery is a crucial phase to identify live hosts on a network before further exploration and vulnerability assessment.

- Various techniques can be employed for host discovery, and the choice of technique depends on factors such as network characteristics, stealth requirements, and the goals of the penetration test.

Host Discovery Techniques

- Ping Sweeps (ICMP Echo Requests): Sending ICMP Echo Requests (ping) to a range of IP addresses to identify live hosts. This is a quick and commonly used method.
- ARP Scanning: Using Address Resolution Protocol (ARP) requests to identify hosts on a local network. ARP scanning is effective in discovering hosts within the same broadcast domain.
- TCP SYN Ping (Half-Open Scan): Sending TCP SYN packets to a specific port (often port 80) to check if a host is alive. If the host is alive, it responds with a TCP SYN-ACK. This technique is stealthier than ICMP ping.

Host Discovery Techniques

- UDP Ping: Sending UDP packets to a specific port to check if a host is alive. This can be effective for hosts that do not respond to ICMP or TCP probes.
- TCP ACK Ping: Sending TCP ACK packets to a specific port to check if a host is alive. This technique expects no response, but if a TCP RST (reset) is received, it indicates that the host is alive.
- SYN-ACK Ping (Sends SYN-ACK packets): Sending TCP SYN-ACK packets to a specific port to check if a host is alive. If a TCP RST is received, it indicates that the host is alive.

Host Discovery Techniques

- The choice of the "best" host discovery technique in penetration testing depends on various factors, and there isn't a one-size-fits-all answer.

- The effectiveness of a host discovery technique can be influenced by the specific characteristics of the target network, the security controls in place, and the goals of the penetration test.

- Here are a few considerations:

Host Discovery Techniques

- ICMP Ping:
  - Pros: ICMP ping is a widely supported and quick method for identifying live hosts.
  - Cons: Some hosts or firewalls may be configured to block ICMP traffic, limiting its effectiveness. ICMP ping can also be easily detected.

- TCP SYN Ping:
  - Pros: TCP SYN ping is stealthier than ICMP and may bypass firewalls that allow outbound connections.
  - Cons: Some hosts may not respond to TCP SYN requests, and the results can be affected by firewalls and security devices.

Ping Sweeps

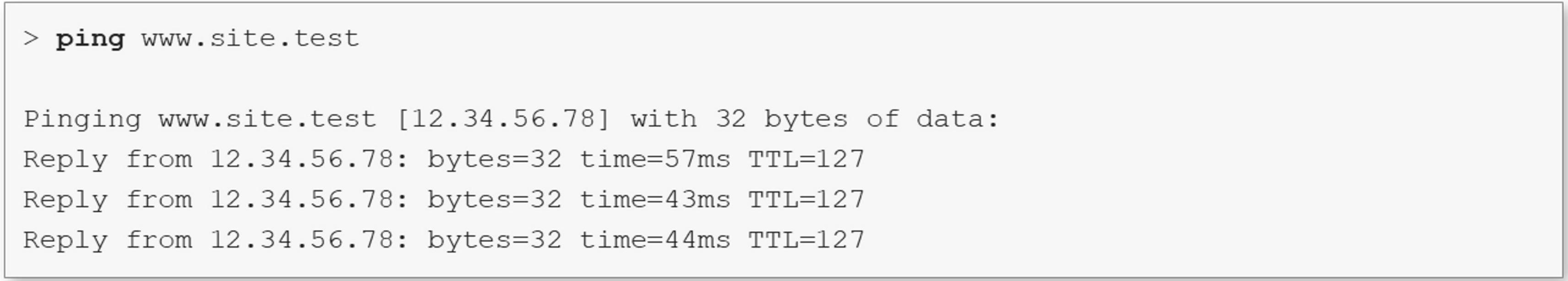
Ping Sweeps

- A ping sweep is a network scanning technique used to discover live hosts (computers, servers, or other devices) within a specific IP address range on a network.

- The basic idea is to send a series of ICMP Echo Request (ping) messages to a range of IP addresses and observe the responses to determine which addresses are active or reachable.

Ping Sweeps

- You probably already know the ping command; it is a utility designed to check if a host is alive/reachable.
- The ping command is available on every major operating system and can be invoked in the command line/terminal as follows:



```
> ping www.site.test

Pinging www.site.test [12.34.56.78] with 32 bytes of data:
Reply from 12.34.56.78: bytes=32 time=57ms TTL=127
Reply from 12.34.56.78: bytes=32 time=43ms TTL=127
Reply from 12.34.56.78: bytes=32 time=44ms TTL=127
```

Ping Sweeps

- Ping sweeps work by sending one or more specially crafted ICMP packets (Type 8 - echo request) to a host.
- If the destination host replies with an ICMP echo reply (Type 0) packet, then the host is alive/online.
- In the context of ICMP (Internet Control Message Protocol), the ICMP Echo Request and Echo Reply messages are used for the purpose of ping. These messages have specific ICMP type and code values associated with them.

Ping Sweeps

- ICMP Echo Request:
  - Type: 8
  - Code: 0

- ICMP Echo Reply:
  - Type: 0
  - Code: 0

Ping Sweeps

- The "Type" field in the ICMP header indicates the purpose or function of the ICMP message, and the "Code" field provides additional information or context related to the message type.

- In the case of ICMP Echo Request and Echo Reply, the Type value 8 represents Echo Request, and the Type value 0 represents Echo Reply.

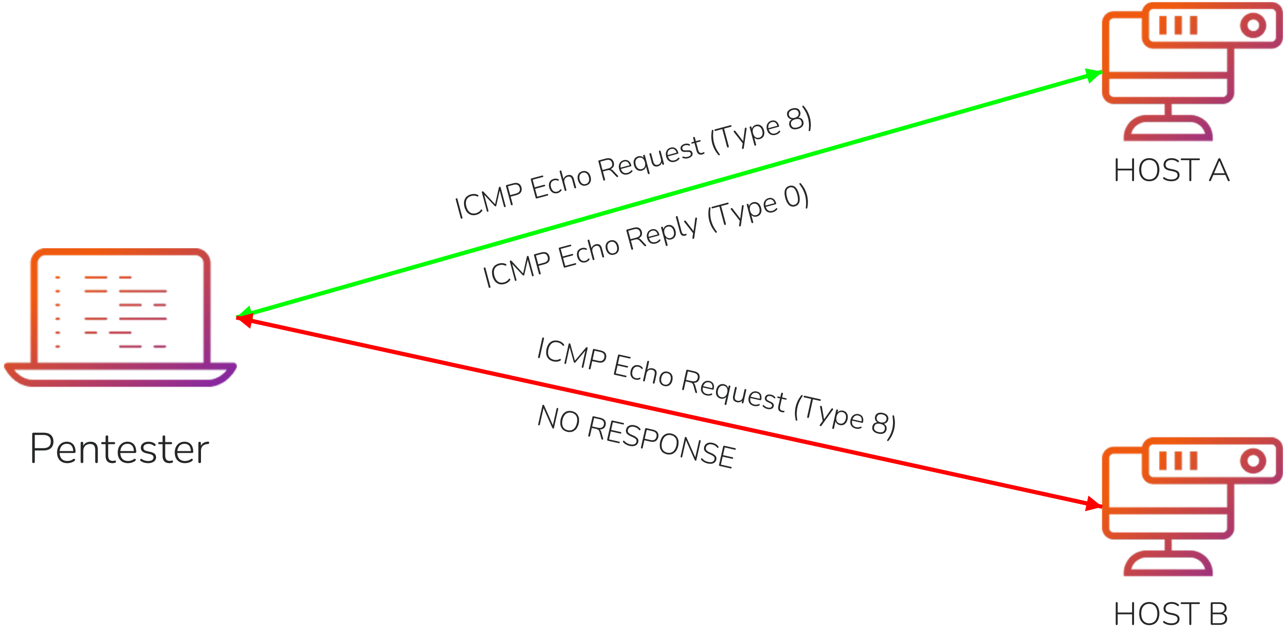
- So, when a device sends an ICMP Echo Request, it creates an ICMP packet with Type 8, Code 0.

- When the destination device receives the Echo Request and responds with an Echo Reply, it creates an ICMP packet with Type 0, Code 0.

Ping Sweeps

- When the host is offline or not reachable, the ICMP Echo Request message sent by the ping utility will not receive a corresponding ICMP Echo Reply.
- The absence of a response doesn't necessarily mean that the host is permanently offline; it could be due to various reasons, such as network congestion, temporary unavailability, or firewall settings that block ICMP traffic.
- The ping utility provides a quick and simple way to check the reachability of a host, but it's important to interpret the results in the context of the network conditions and host configuration.

Ping Sweeps Visualized



Demo: Ping Sweeps

Host Discovery With Nmap

Demo: Host Discovery With Nmap

Port Scanning With Nmap

Demo: Port Scanning With Nmap

Service Version & OS Detection

Demo: Service Version & OS Detection

Nmap Scripting Engine (NSE)

Demo: Nmap Scripting Engine (NSE)

Firewall Detection & IDS Evasion

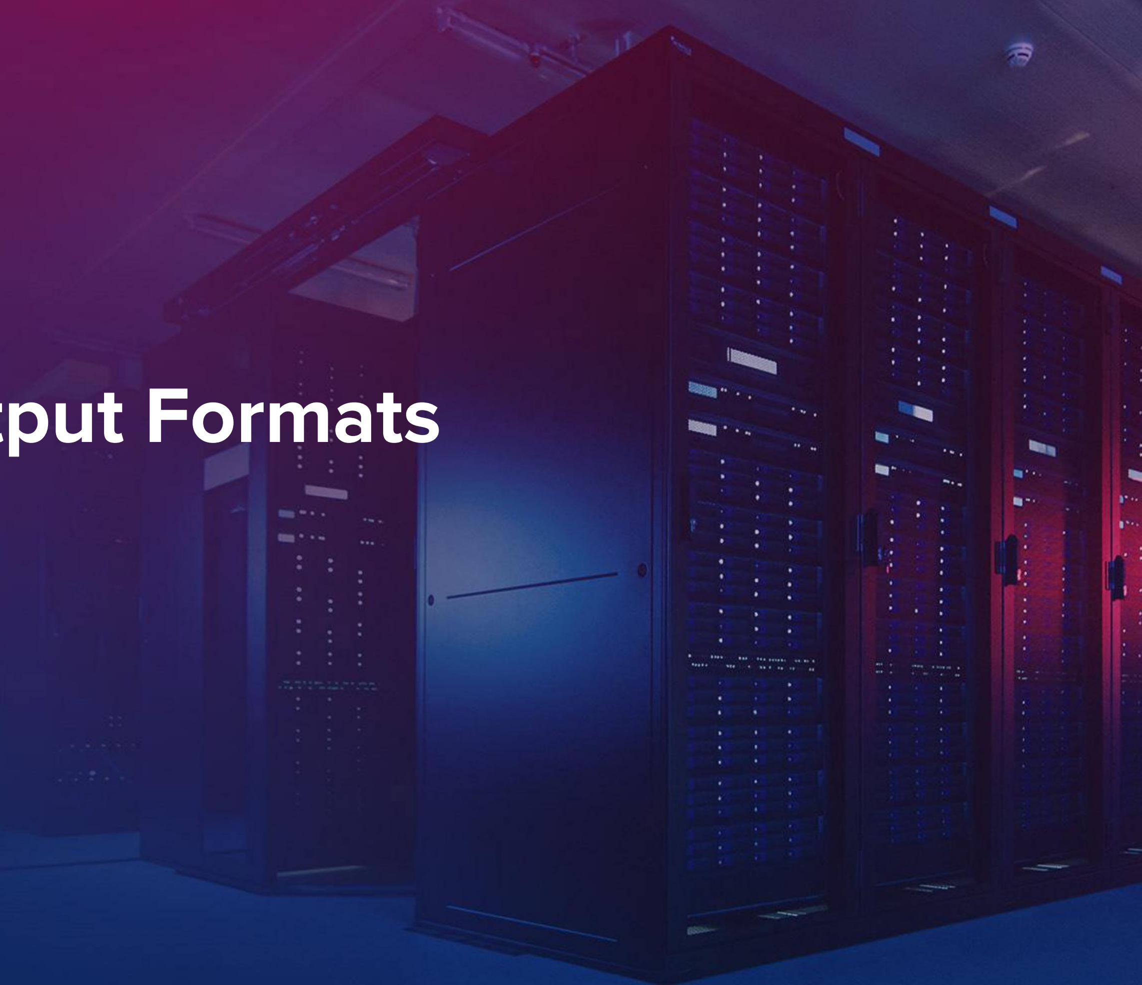
Demo: Firewall Detection & IDS Evasion

Optimizing Nmap Scans

Demo: Optimizing Nmap Scans

Nmap Output Formats

Demo: Nmap Output Formats



Introduction To Enumeration

Enumeration

- After the host discovery and port scanning phase of a penetration test, the next logical phase is going to involve service enumeration.

- The goal of service enumeration is to gather additional, more specific/detailed information about the hosts/systems on a network and the services running on said hosts.

- This includes information like account names, shares, misconfigured services and so on.

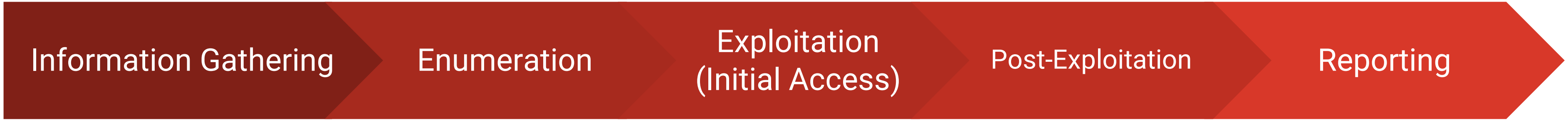
- Like the scanning phase, enumeration involves active connections to the remote devices in the network.

Enumeration

- There are many protocols on networked systems that an attacker can target if they have been misconfigured or have been left enabled.

- In this section of the course, we will be exploring the various tools and techniques that can be used to interact with these protocols, with the intent of eventually/potentially exploiting them in later phases.

Penetration Testing Methodology



Passive Information Gathering

OSINT

Active Information Gathering

Network Mapping
Host Discovery
Port Scanning
Service Detection & OS Detection

Service & OS Enumeration

Service Enumeration
User Enumeration
Share Enumeration

Vulnerability analysis and threat modeling

Vulnerability Analysis
Vulnerability Identification

Exploitation

Developing/Modifying Exploits
Service Exploitation

Post Exploitation

Local Enumeration
Privilege Escalation
Credential Access
Persistence
Defense Evasion
Lateral Movement

Reporting

Report Writing
Recommendations

SMB & NetBIOS Enumeration

SMB & NetBIOS Enumeration

- NetBIOS and SMB are two different technologies, but they're related in the context of networking and file sharing on Windows networks.

- Let's break down each of them to understand their roles and how they differ:

NetBIOS (Network Basic Input/Output System)

- NetBIOS is an API and a set of network protocols for providing communication services over a local network. It's used primarily to allow applications on different computers to find and interact with each other on a network.
- Functions: NetBIOS offers three primary services:
  - Name Service (NetBIOS-NS): Allows computers to register, unregister, and resolve names in a local network.
  - Datagram Service (NetBIOS-DGM): Supports connectionless communication and broadcasting.
  - Session Service (NetBIOS-SSN): Supports connection-oriented communication for more reliable data transfers.
- Ports: NetBIOS typically uses ports 137 (Name Service), 138 (Datagram Service), and 139 (Session Service) over UDP and TCP.

SMB (Server Message Block)

- SMB is a network file sharing protocol that allows computers on a network to share files, printers, and other resources. It is the primary protocol used in Windows networks for these purposes.
- Functions: SMB provides features for file and printer sharing, named pipes, and inter-process communication (IPC). It allows users to access files on remote computers as if they were local.
- Versions: SMB has several versions:
  - SMB 1.0: The original version, which had security vulnerabilities. It was used with older operating systems like Windows XP.
  - SMB 2.0/2.1: Introduced with Windows Vista/Windows Server 2008, offering improved performance and security.
  - SMB 3.0+: Introduced with Windows 8/Windows Server 2012, adding features like encryption, multichannel support, and improvements for virtualization.
- Ports: SMB generally uses port 445 for direct SMB traffic (bypassing NetBIOS) and port 139 when operating with NetBIOS.

SMB & NetBIOS Enumeration

- While NetBIOS and SMB were once closely linked, modern networks rely primarily on SMB for file and printer sharing, often using DNS and other mechanisms for name resolution instead of NetBIOS.

- Modern implementations of Windows primarily use SMB and can work without NetBIOS, however, NetBIOS over TCP 139 is required for backward compatibility and are often enabled together.

Lab Demo: SMB & NetBIOS Enumeration

SNMP Enumeration

Simple Network Management Protocol (SNMP)

- SNMP (Simple Network Management Protocol) is a widely used protocol for monitoring and managing networked devices, such as routers, switches, printers, servers, and more.

- It allows network administrators to query devices for status information, configure certain settings, and receive alerts or traps when specific events occur.

Simple Network Management Protocol (SNMP)

- SNMP is an application layer protocol that typically uses UDP for transport. It involves three primary components:
  - SNMP Manager: The system responsible for querying and interacting with SNMP agents on networked devices.
  - SNMP Agent: Software running on networked devices that responds to SNMP queries and sends traps.
  - Management Information Base (MIB): A hierarchical database that defines the structure of data available through SNMP. Each piece of data has a unique Object Identifier (OID).

Simple Network Management Protocol (SNMP)

- Versions of SNMP:
  - SNMPv1: The earliest version, using community strings (essentially passwords) for authentication.
  - SNMPv2c: An improved version with support for bulk transfers but still relying on community strings for authentication.
  - SNMPv3: Introduced security features, including encryption, message integrity, and user-based authentication.
- Ports:
  - Port 161 (UDP): Used for SNMP queries.
  - Port 162 (UDP): Used for SNMP traps (notifications).

SNMP Enumeration

- SNMP enumeration in penetration testing involves querying SNMP-enabled devices to gather information useful for identifying potential vulnerabilities, misconfigurations, or points of attack.

- Here are the key objectives and outcomes of SNMP enumeration during a pentest:

SNMP Enumeration

- Identify SNMP-Enabled Devices: Determine which devices on the network have SNMP enabled and whether they are vulnerable to information leakage or attacks.
- Extract System Information: Collect system-related data like device names, operating systems, software versions, network interfaces, and more.
- Identify SNMP Community Strings: Test for default or weak community strings, which can grant unauthorized access to device information.
- Retrieve Network Configurations: Gather information about routing tables, network interfaces, IP addresses, and other network-specific details.
- Collect User and Group Information: In some cases, SNMP can reveal user account information and access permissions.
- Identify Services and Applications: Find out which services and applications are running on the target devices, potentially leading to further attack vectors.

Lab Demo: SNMP Enumeration

Linux Service Enumeration

Lab Demo: Linux Service Enumeration

SMB Relay Attack

SMB Relay Attack

- An SMB relay attack is a type of network attack where an attacker intercepts SMB (Server Message Block) traffic, manipulates it, and relays it to a legitimate server to gain unauthorized access to resources or perform malicious actions.

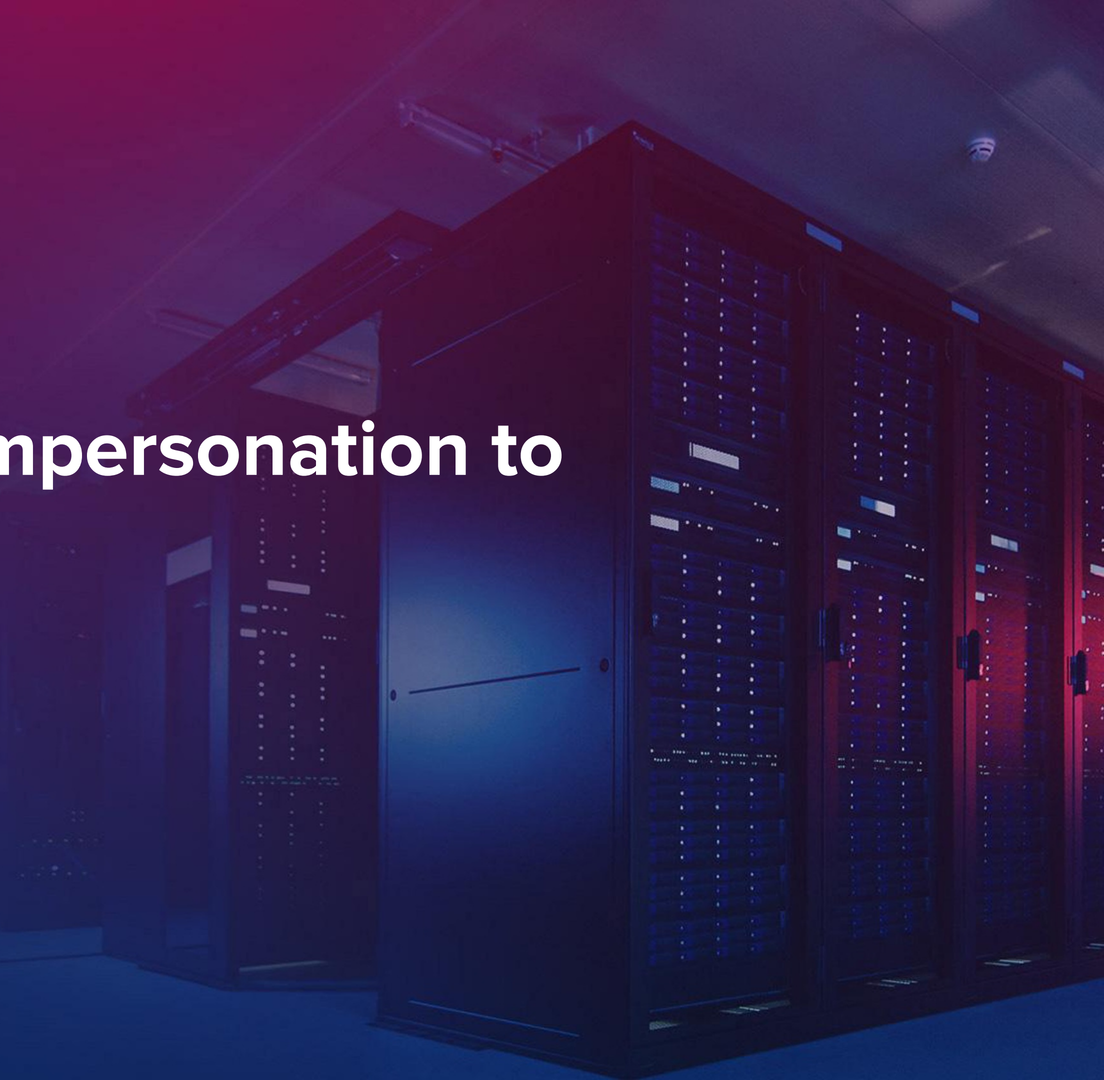
- This type of attack is common in Windows networks, where SMB is used for file sharing, printer sharing, and other network services.

How SMB Relay Attacks Work

- **Interception:** The attacker sets up a man-in-the-middle position between the client and the server. This can be done using various techniques, such as ARP spoofing, DNS poisoning, or setting up a rogue SMB server.
- **Capturing Authentication:** When a client connects to a legitimate server via SMB, it sends authentication data. The attacker captures this data, which might include NTLM (NT LAN Manager) hashes.
- **Relaying to a Legitimate Server:** Instead of decrypting the captured NTLM hash, the attacker relays it to another server that trusts the source. This allows the attacker to impersonate the user whose hash was captured.
- **Gain Access:** If the relay is successful, the attacker can gain access to the resources on the server, which might include sensitive files, databases, or administrative privileges. This access could lead to further lateral movement within the network, compromising additional systems.

Lab Demo: SMB Relay Attack

MSSQL DB User Impersonation to RCE



Lab Demo: MSSQL DB User Impersonation to RCE

Linux Black-Box Penetration Test

Lab Demo: Linux Black-Box Penetration Test

Dumping & Cracking NTLM Hashes

Windows Password Hashes

- The Windows OS stores hashed user account passwords locally in the SAM (Security Accounts Manager) database.

- Hashing is the process of converting a piece of data into another value. A hashing function or algorithm is used to generate the new value. The result of a hashing algorithm is known as a hash or hash value.

- Authentication and verification of user credentials is facilitated by the Local Security Authority (LSA).

- Windows versions up to Windows Server 2003 utilize two different types of hashes:
  - LM
  - NTLM
- Windows disables LM hashing and utilizes NTLM hashing from Windows Vista onwards.

SAM Database

- SAM (Security Account Manager) is a database file that is responsible for managing user accounts and passwords on Windows. All user account passwords stored in the SAM database are hashed.

- The SAM database file cannot be copied while the operating system is running.

- The Windows NT kernel keeps the SAM database file locked and as a result, attackers typically utilize in-memory techniques and tools to dump SAM hashes from the LSASS process.

- In modern versions of Windows, the SAM database is encrypted with a syskey.

Note: Elevated/Administrative privileges are required in order to access and interact with the LSASS process.

NTLM (NTHash)

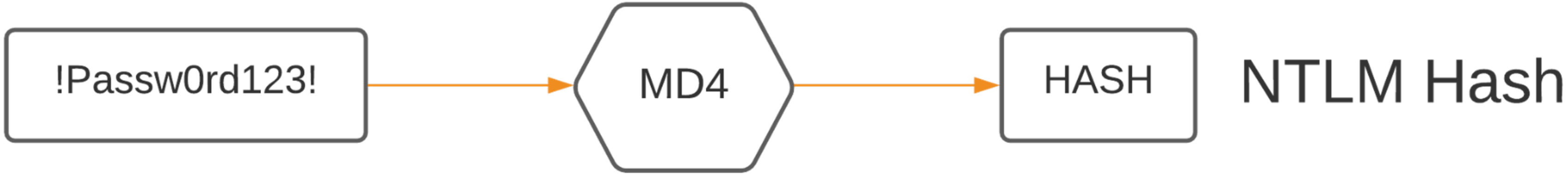
- NTLM is a collection of authentication protocols that are utilized in Windows to facilitate authentication between computers. The authentication process involves using a valid username and password to authenticate successfully.

- From Windows Vista onwards, Windows disables LM hashing and utilizes NTLM hashing.

- When a user account is created, it is encrypted using the MD4 hashing algorithm, while the original password is disposed of.

- NTLM improves upon LM in the following ways:
  - Does not split the hash in to two chunks.
  - Case sensitive.
  - Allows the use of symbols and unicode characters.

NTLM (NTHash)



Dumping & Cracking NTLM Hashes

- We can dump Windows password hashes by leveraging various utilities like:
  - The inbuilt meterpreter “hashdump” command
  - Mimikatz

- After we have dumped the hashes, we can crack them through the use of the following utilities:
  - John The Ripper
  - Hashcat

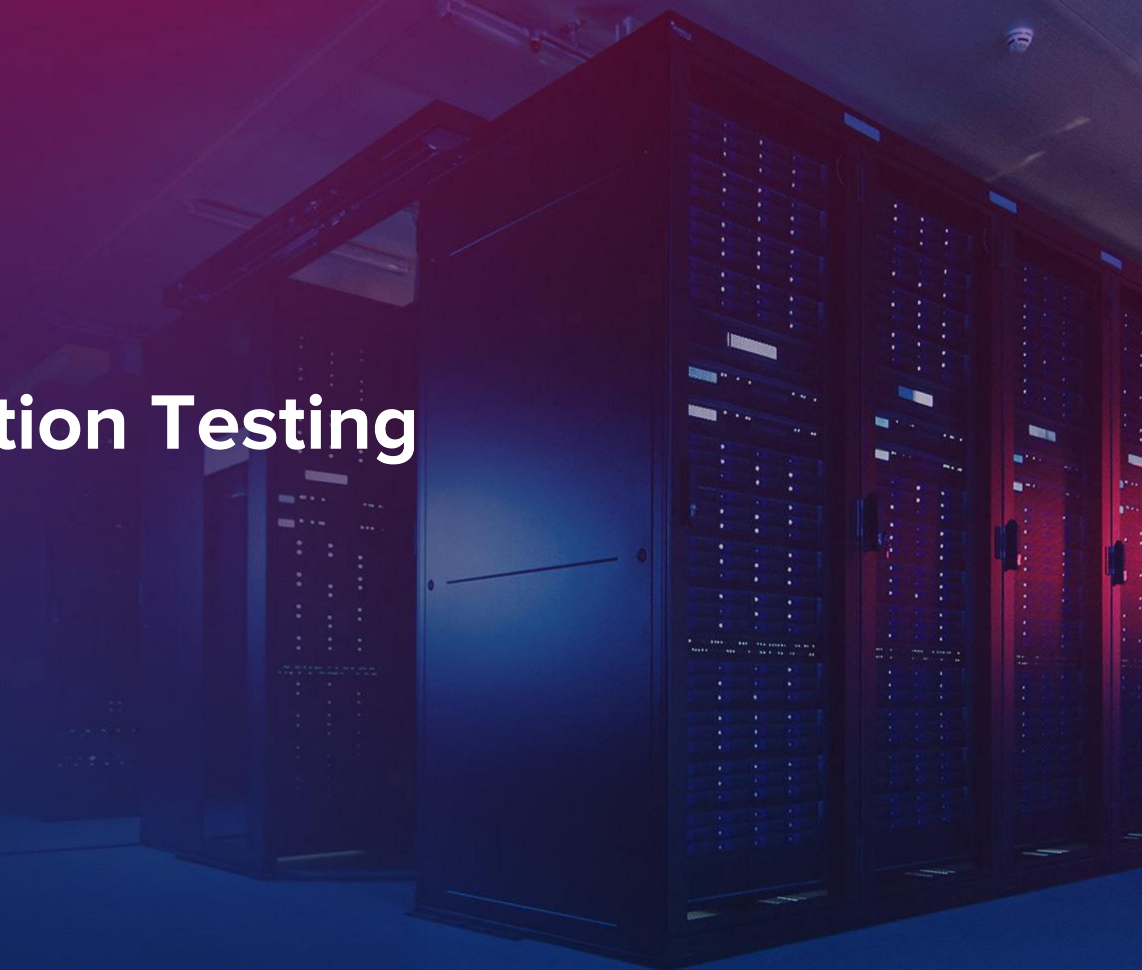
Lab Demo: Dumping & Cracking NTLM Hashes

Windows Post-Exploitation Lab

Lab Demo: Windows Post-Exploitation Lab

Network Penetration Testing

Course Conclusion



Learning Objectives:

1. Host Discovery & Port Scanning
   - Demonstrate competency in identifying hosts on a target network through various host discovery techniques applicable to both Windows and Linux.
   - Utilize network mapping and port scanning tools to identify open ports on target systems and the services running on the open ports.
2. Service Enumeration
   - Demonstrate competency in enumerating important information from services running on both Windows and Linux systems.
   - Leverage enumeration tools and techniques like Nmap Scripts and other protocol specific tools to enumerate information from specific network protocols (SMB, NetBIOS, SMTP, FTP etc).
3. MITM & Network-Based Attacks
   - Demonstrate competency in performing ARP Spoofing and DNS Spoofing attacks.
   - Demonstrate competency in performing ARP Poisoning and NBT-NS Poisoning attacks.
   - Leverage tools like arpspoof, dnsspoof and Responder to facilitate MITM Attacks.
4. Exploitation & Post-Exploitation
   - Demonstrate competency in exploiting Windows and Linux specific protocols and services for initial access.
   - Demonstrate competency in performing advanced network-based Windows exploitation techniques like SMB Relaying.
   - Demonstrate competency in performing post-exploitation activities on Windows and Linux systems.

Thank You!

EXPERTS AT MAKING YOU AN EXPERT

INE