Title: Unveiling the Array of Attacks Targeting the Data Link Layer of the OSI Model: Real-World Case Studies and Effective Mitigation Strategies

Introduction:

The Data Link Layer, as a fundamental component of the OSI model, provides critical functionalities for ensuring efficient and reliable data transmission. However, it is not impervious to security threats. In this comprehensive article, we will delve into various types of attacks that specifically target the Data Link Layer. Additionally, we will analyze two real-world case studies to understand the ramifications of these attacks and explore effective mitigation strategies to bolster network security at this layer.

1. Address Resolution Protocol (ARP) Spoofing:

ARP spoofing entails manipulating the Address Resolution Protocol, which facilitates the mapping of IP addresses to MAC addresses in local networks. Attackers forge falsified ARP messages, associating their MAC address with a legitimate IP address. This enables them to intercept network traffic, potentially leading to unauthorized access or data theft.

Case Study 1: In 2014, a prominent financial institution fell victim to a large-scale ARP spoofing attack. The attacker successfully intercepted and manipulated network traffic, resulting in unauthorized access to confidential data, substantial financial losses, and severe reputational damage.

Mitigation Strategies for ARP Spoofing:

- Employ static ARP entries or enable port security to bind IP addresses with specific MAC addresses.
- Implement ARP inspection or dynamic ARP inspection to validate ARP messages and detect anomalies.
- Deploy intrusion detection and prevention systems (IDS/IPS) to identify and block suspicious ARP activity.

2. VLAN Hopping:

VLAN hopping exploits vulnerabilities in VLAN configurations or network trunking protocols to gain unauthorized access to data on different Virtual Local Area Networks (VLANs). Attackers can bypass VLAN separation, compromising the confidentiality and integrity of sensitive information.

Case Study 2: In 2008, a security researcher discovered a vulnerability in Cisco switches, allowing attackers to circumvent VLAN separation. This breach underscored the importance of robust VLAN configurations and timely firmware updates to address vulnerabilities and enhance security measures.

Mitigation Strategies for VLAN Hopping:

- Utilize VLAN access control lists (VACLs) to restrict traffic between VLANs.

- Enable VLAN pruning to prevent unnecessary broadcast and multicast traffic from traversing VLANs.
- Implement port security features to ensure that only authorized devices can access specific VLANs.
- Employ Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) to monitor and respond to suspicious VLAN hopping activity.

3. MAC Address Spoofing:

MAC address spoofing involves altering the Media Access Control (MAC) address of a network device to impersonate another device on the network. Attackers can exploit this technique to bypass network access controls, execute unauthorized activities, or launch further attacks within the network.

Mitigation Strategies for MAC Address Spoofing:

- Implement port security measures, such as MAC address filtering, to restrict network access based on authorized MAC addresses.
- Enable dynamic MAC address learning and aging mechanisms to prevent spoofed MAC addresses from persisting in network devices.
- Utilize 802.1X authentication protocols to validate the identity of network devices and prevent unauthorized access.

4. Denial-of-Service (DoS) Attacks:

DoS attacks targeting the Data Link Layer aim to overwhelm network devices or exhaust their resources, rendering them inaccessible to legitimate users. Attackers exploit vulnerabilities in network protocols or employ techniques such as flooding the network with an excessive volume of traffic, leading to disruptions and impairing network performance.

Mitigation Strategies for DoS Attacks:

- Configure rate-limiting and traffic shaping mechanisms to control and prioritize network traffic, preventing excessive traffic from overwhelming devices.
- Employ traffic anomaly detection systems to identify and mitigate abnormal traffic patterns associated with DoS attacks.
- Implement ingress and egress filtering to block spoofed or illegitimate traffic at the network perimeter.

5. Man-in-the-Middle (MitM) Attacks:

MitM attacks involve intercepting and altering communication between two parties without their knowledge. At the Data Link Layer, attackers position themselves between the sender and receiver, intercepting and forwarding network traffic. This enables eavesdropping on sensitive information, data manipulation, or unauthorized actions.

Mitigation Strategies for MitM Attacks:

- Implement strong encryption mechanisms, such as Virtual Private Networks (VPNs), to secure communication channels and protect against interception.

- Utilize cryptographic protocols, such as Secure Shell (SSH) or Transport Layer Security (TLS), to authenticate and establish secure connections.
- Employ mutual authentication mechanisms to ensure both parties are verified before establishing communication.

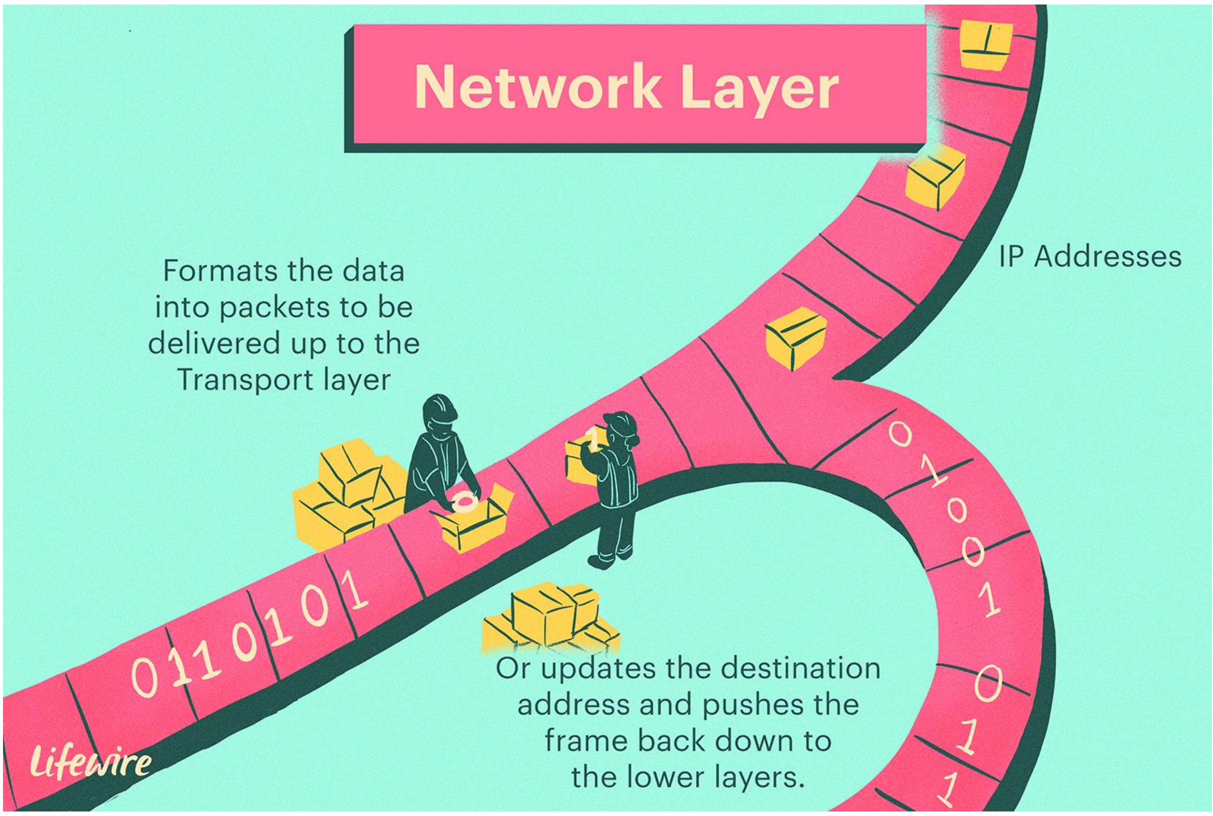
Conclusion:

Securing the Data Link Layer is of utmost importance to maintain the integrity, confidentiality, and availability of network communications. Understanding the various attacks targeting this layer, such as ARP spoofing, VLAN hopping, MAC address spoofing, DoS attacks, and MitM attacks, empowers organizations to implement effective mitigation strategies. By deploying a combination of preventative measures, including robust network configurations, advanced monitoring systems, and thorough security awareness training, organizations can fortify their network defenses and mitigate the risk of Data Link Layer attacks. Regular security assessments, firmware updates, and staying abreast of emerging threats are also crucial for ensuring a resilient and secure network infrastructure.

Resources :

https://www.makeuseof.com/what-is-mac-spoofing/
https://www.techtarget.com/searchsecurity/definition/VLAN-hopping
https://www.imperva.com/learn/application-security/man-in-the-middle-attack-mitm/
https://www.crowdstrike.com/cybersecurity-101/spoofing-attacks/arp-spoofing/
https://www.paloaltonetworks.com/cyberpedia/what-is-a-denial-of-service-attack-dos

OSI Layer 3 - Network Layer


Network Layer
IP Addresses
Formats the data into packets to be delivered up to the Transport layer
Or updates the destination address and pushes the frame back down to the lower layers.
0110101
0100101011
Lifewire

The OSI Model is a network protocol that stands for Open Systems Interconnection Model. This model divides network communication into seven different layers, with each layer addressing a specific aspect of data transmission and processing. The Network Layer is the third layer of the OSI Model and is responsible for the routing of data packets from source to destination.

Services Offered by Network Layer

The services which are offered by the network layer protocol are as follows:

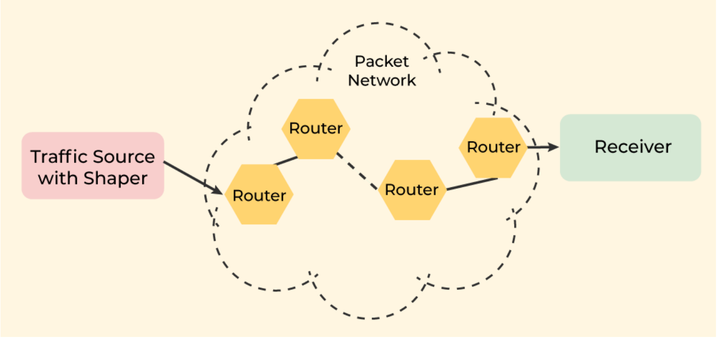
Packetizing

Routing

Forwarding

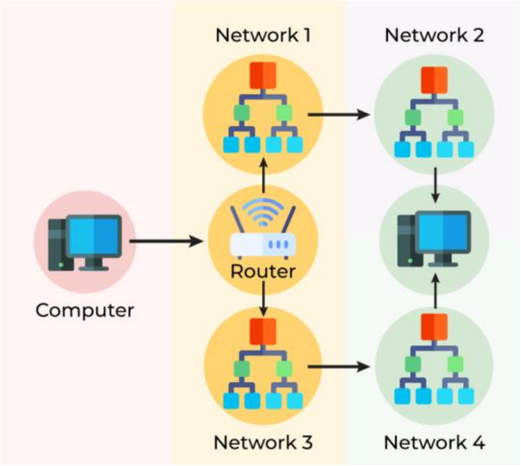
1. Packetizing

The process of encapsulating the data received from the upper layers of the network (also called payload) in a network layer packet at the source and decapsulating the payload from the network layer packet at the destination is known as packetizing.



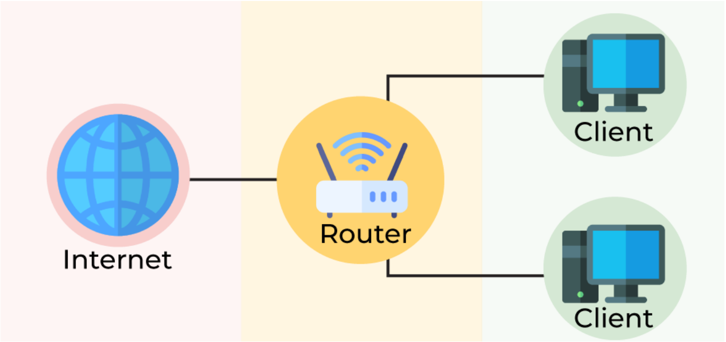
2. Routing

Routing is the process of moving data from one device to another device. These are two other services offered by the network layer. In a network, there are a number of routes available from the source to the destination. The network layer specifies some strategies which find out the best possible route. This process is referred to as routing. There are a number of routing protocols that are used in this process and they should be run to help the routers coordinate with each other and help in establishing communication throughout the network.



3. Forwarding

Forwarding is simply defined as the action applied by each router when a packet arrives at one of its interfaces. When a router receives a packet from one of its attached networks, it needs to forward the packet to another attached network (unicast routing) or to some attached networks (in the case of multicast routing). Routers are used on the network for forwarding a packet from the local network to the remote network. So, the process of routing involves packet forwarding from an entry interface out to an exit interface.



The source host adds a header that contains the source and destination address and some other relevant information required by the network layer protocol to the payload received from the upper layer protocol and delivers the packet to the data link layer.

The destination host receives the network layer packet from its data link layer, decapsulates the packet, and delivers the payload to the corresponding upper layer protocol. The routers in the path are not allowed to change either the source or the destination address. The routers in the path are not allowed to decapsulate the packets they receive unless they need to be fragmented.

The Network Layer plays a crucial role in communication networks. Its task is to ensure that packets can be delivered from the source network to the destination network. Additionally, it is responsible for controlling data traffic, performing routing processes, and determining appropriate paths for reaching the destination on the network.

The Network Layer performs the following functions:

Addressing and Routing: The Network Layer assigns a unique address (IP address) to each device and enables communication using these addresses. It also determines the optimal path for routing packets from the source network to the destination network.

Packetization and Fragmentation: Data is divided into packets at the Network Layer. Each packet is labeled with source and destination addresses to facilitate communication.

Routing Control: The Network Layer ensures the proper routing of data packets through router devices. Routers use routing tables at the network layer to direct packets to their destinations.

The Network Layer is based on the Internet Protocol (IP). IP provides addressing and routing functions for transmitting and directing data packets. IP addresses are unique identifying numbers used in the network layer.

The vulnerabilities and types of cyber attacks that the Network Layer is exposed to include:

Denial-of-Service (DoS) Attacks: The Network Layer can be vulnerable to DoS attacks, where resources are made inaccessible. Attackers typically overload network resources by sending a high volume of IP packets to disrupt the target network's services.

IP Spoofing: IP spoofing is an attack where attackers modify the source IP address of IP packets with a fake IP address. This allows attackers to conceal their true identities and facilitates attacks such as phishing or bypassing security measures on the network.

Smurf Attacks: Smurf attacks are typically DoS attacks that occur at the network layer. Attackers send ICMP echo requests (ping) with a spoofed source IP address to the IP broadcast address. This causes responses from all systems in the network to be directed towards the target, leading to resource depletion.

IP Fragmentation Attacks: These attacks exploit vulnerabilities in the fragmentation and reassembly process of IP packets. Attackers can disrupt the target network or compromise the security of resources by depriving the network layer of proper packet reassembly.

Routing Protocol Attacks: These attacks exploit vulnerabilities in routing protocols. Attackers can manipulate routing tables or send incorrect routing information to redirect network traffic to unwanted paths or destinations.

These are some of the vulnerabilities and types of cyber attacks that can affect the Network Layer of the OSI Model. Ensuring the security of the Network Layer should be addressed in conjunction with security measures implemented at other layers of the network.

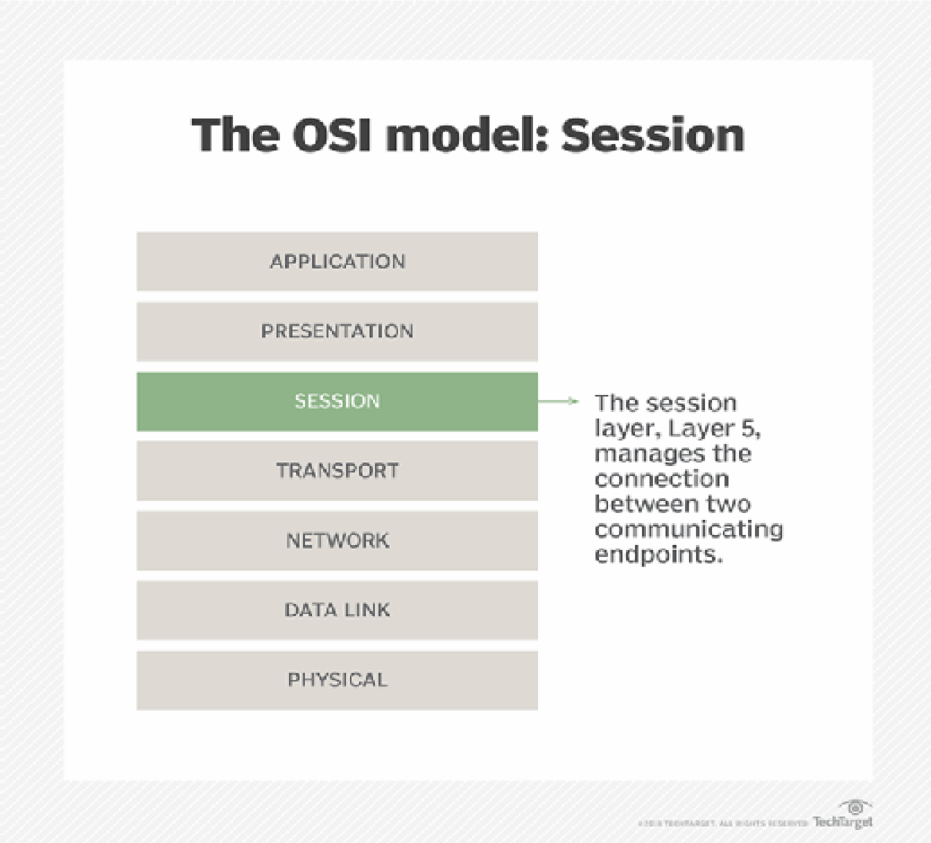
Resources :

https://osi-model.com/network-layer/
https://www.imperva.com/learn/application-security/osi-model/
https://www.geeksforgeeks.org/network-layer-services-packetizing-routing-and-forwarding/
https://en.wikipedia.org/wiki/OSI_model
https://ipwithease.com/network-vulnerabilities-and-the-osi-model/

Understanding the Session Layer: Enhancing Cybersecurity through Effective Communication

Introduction:

In the realm of cybersecurity, establishing secure and reliable communication channels is of utmost importance. One crucial component of the networking stack that plays a pivotal role in ensuring secure communication is the session layer. The session layer acts as a bridge between the transport layer and the application layer, facilitating the establishment, maintenance, and termination of communication sessions. In this article, we will delve into the session layer, exploring its functions, protocols, and its significance in enhancing cybersecurity.

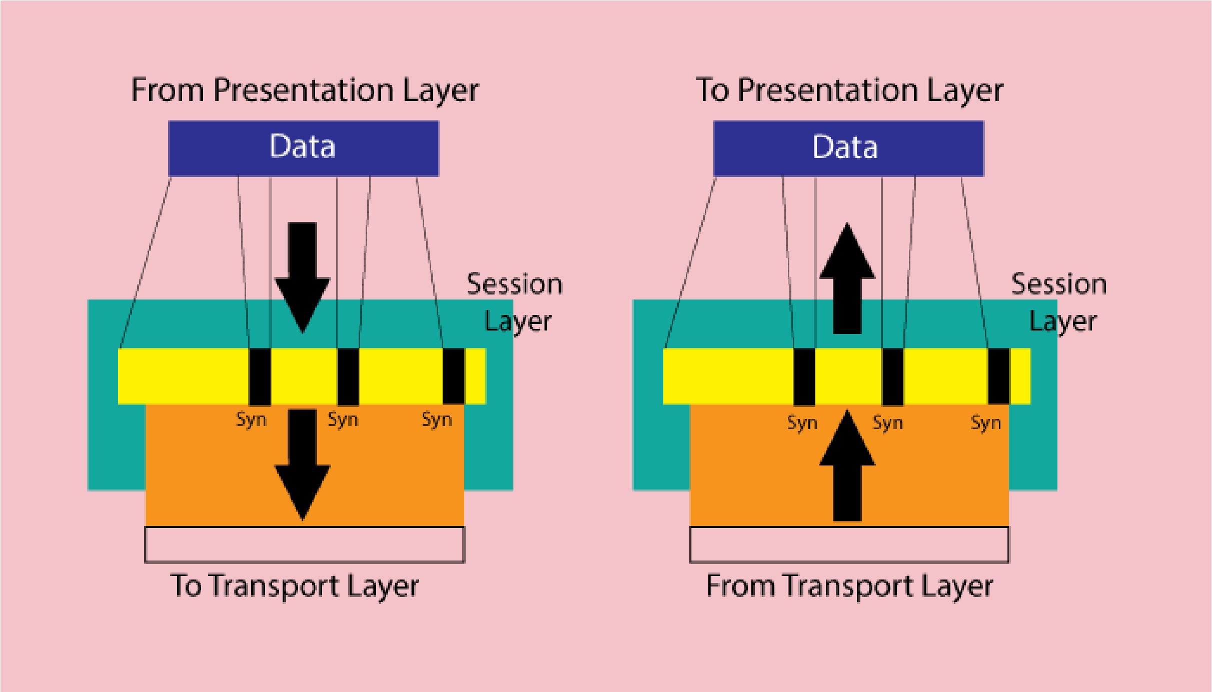


Understanding the Session Layer:

The session layer, as defined in the OSI (Open Systems Interconnection) model, is the fifth layer responsible for managing and coordinating communication sessions between hosts. Its primary objective is to establish a logical connection or session between two network applications. By providing synchronization, dialog control, and data exchange coordination, the session layer enables reliable and efficient communication.

Functions of the Session Layer:

Session Establishment: The session layer initiates and establishes sessions between communicating hosts, ensuring proper coordination and synchronization of data exchange. It defines how the session will be established, whether through a simple connection or a more complex negotiation process.



Session Maintenance:

Once a session is established, the session layer ensures its uninterrupted operation. It manages checkpoints, allowing retransmission of lost data, and handles session recovery in the event of failures, disruptions, or network congestion.

Session Termination:

The session layer gracefully terminates communication sessions by coordinating the release of resources, notifying the participating applications, and ensuring any pending data is transmitted and received. Protocols at the Session Layer: Several protocols operate at the session layer, each designed to fulfill specific communication requirements. Some notable session layer protocols include:

Remote Procedure Call (RPC):

RPC enables a program on one computer to execute code on a remote computer over a network. It establishes a session between the client and server, allowing the client to invoke procedures or functions on the remote server as if they were executed locally.

Session Control Protocol (SCP):

SCP is a protocol used to manage the establishment, maintenance, and termination of sessions. It ensures proper synchronization, data flow control, and session recovery mechanisms.

Session Description Protocol (SDP):

SDP is a protocol used for describing multimedia sessions. It provides information such as session timing, media types, codecs, and network addresses, facilitating the negotiation and establishment of multimedia sessions.

Significance in Cybersecurity:

The session layer plays a crucial role in enhancing cybersecurity in multiple ways:

Authentication and Authorization:

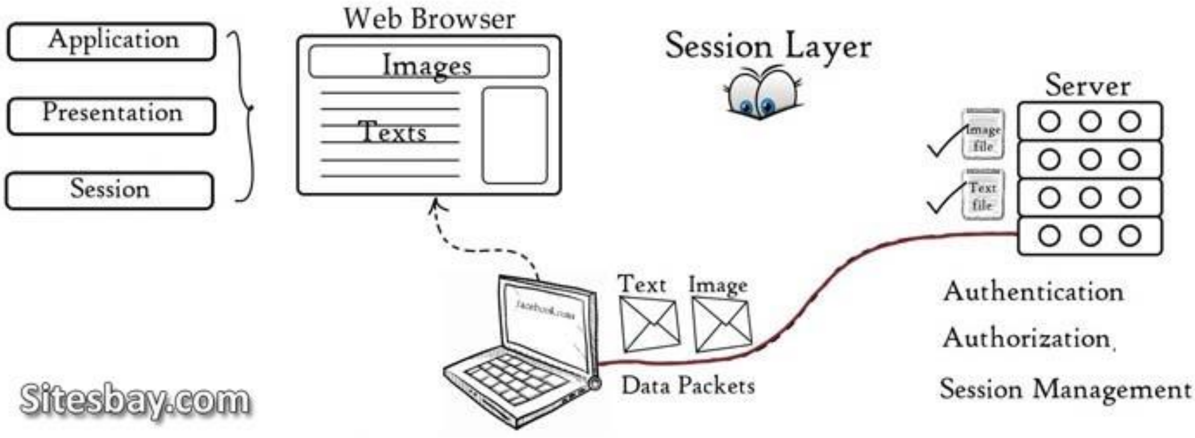
The session layer enables authentication and authorization mechanisms, ensuring that only authorized entities can establish sessions and access network resources. This helps prevent unauthorized access and protects against malicious activities.

Encryption and Data Integrity:

By establishing secure communication sessions, the session layer facilitates the implementation of encryption algorithms, ensuring confidentiality and integrity of data transmitted over the network. Encryption prevents eavesdropping and data tampering, safeguarding sensitive information.

Session Monitoring and Intrusion Detection:

The session layer allows for monitoring and inspection of session-related data, facilitating intrusion detectection and prevention systems. Monitoring session-related parameters, such as session duration, data transfer rates, and unusual behaviors, helps identify potential security threats and anomalous activities.



Conclusion:

In the realm of cybersecurity, the session layer serves as a critical component, facilitating secure and reliable communication between network applications. By establishing, maintaining, and terminating sessions, the session layer enhances authentication, encryption, data integrity, and intrusion detection, bolstering overall network security. Understanding the session layer and its protocols is vital for implementing robust cybersecurity measures and protecting sensitive information in today's interconnected world.

Title: **In-Depth Research on Attacks Targeting the Presentation 6th Layer of the OSI Model: Analysis of Real-World Case**

1. Abstract:

This research report dives into attacks targeting the Presentation Layer of the OSI (Open Systems Interconnection) model. It provides an in-depth exploration of various attack vectors, techniques, and their impact on network security. The report also includes an analysis of real-world case study, highlighting the consequences of presentation layer attacks and the countermeasures employed to mitigate them.

2. Introduction:

The presentation layer is the sixth layer of the OSI (Open Systems Interconnection) model. It is responsible for the formatting, encryption, and translation of data to ensure that it can be understood by the receiving application. The main goal of the presentation layer is to provide a common representation of data that can be shared between different systems. The primary objective of the presentation layer is to ensure that data sent by the application layer is properly formatted, secured, and understood by the receiving application. It acts as an intermediary between the application layer and the lower layers of the OSI model, such as the session layer and the transport layer.
Overall, the presentation layer acts as a bridge between the application layer and the underlying network infrastructure. It handles data formatting, encryption, translation, and protocol conversion to facilitate effective data communication and ensure that information is exchanged accurately and securely across networks.

3. Attack Vectors:

- **Malformed or manipulated data:** Attackers can attempt to send malformed or manipulated data to applications that rely on the presentation layer. By exploiting vulnerabilities in the parsing or processing of data, they may cause the application to behave unexpectedly or crash. This can be achieved by injecting specially crafted data that triggers buffer overflows, code injection, or other types of software vulnerabilities.

- **Code injection:** If an application relies on user input to generate or interpret data at the presentation layer, attackers may attempt to inject malicious code into the input fields. This code can exploit vulnerabilities in the application to gain unauthorized access, execute arbitrary commands, or compromise the system.

- **Format string attacks:** Applications that improperly handle format strings can be vulnerable to attacks. By manipulating the format specifiers in the input, attackers can trick the application into exposing sensitive information or executing unintended actions.

- **Cross-Site Scripting (XSS):** XSS attacks can indirectly impact the presentation layer by targeting web applications. By injecting malicious scripts into web pages or web application inputs, attackers can execute malicious code in the user's browser, leading to unauthorized access, session hijacking, or the theft of sensitive information.

- **Content spoofing:** Attackers may attempt to deceive users by manipulating the content displayed at the presentation layer. This can involve creating fake websites or altering the appearance of legitimate websites to trick users into divulging sensitive information.

4. Analysis Of Real-World Case Studies:

- **British Airways' Data Breach of 2018: A synopsis:**
  This case study examines the XSS vulnerability attack on the British airways Database During the summer of 2018, over 400,000 British Airways customers had their personal information breached, including their usernames, passwords, credit card details and other required flight information. The script is connected to the British Airways baggage claim information page; the last time it had been modified prior to the breach was December 2012. Klijnsma quickly noticed that attackers revised the component to include code—just 22 lines of it—often used in clandestine manipulations. The malicious code grabbed data that customers entered into a payment form, and sent it to an attacker-controlled server when a user clicked or tapped a submission button. The attackers even paid to set up a Secure Socket Layer certificate for their server, a credential that confirms a server has web encryption enabled to protect data in transit. Attackers of all sorts have increasingly used these certificates to help create an air of legitimacy—even though an encrypted site is not necessarily safe.
  The airline also said in its disclosure that the attack impacted its mobile users. Klijnsma found a part of the British Airways Android app built off of the same code as the compromised portion of the airline's website. It's normal for an app's functionality to be based in part on existing web infrastructure, but the practice can also create shared risk. In the case of the British Airways Android app, the malicious JavaScript component the attackers injected on the main site hit the mobile app as well. Attackers seem to have designed the script with this in mind by accommodating touchscreen inputs. While the attack wasn't elaborate, it was effective, because it was tailored to the specific scripting and data flow weaknesses of the British Airways site.

  RiskIQ says it is attributing the incident to Magecart because the skimmer code injected into the British Airways website is a modified version of the group's hallmark script. RiskIQ also views the attack as an evolution of the techniques used in the recent Ticketmaster breach, which RiskIQ linked to Magecart, though with the added innovation of directly targeting a victim's site rather than compromising a third party. And some of the attack infrastructure, like the web server hosting and domain name, point to the group as well.

  So far British Airways and law enforcement haven't publicly commented on this attribution, but Klijnsma says the other takeaway for now is the prevalence of tiny website vulnerabilities that can quickly turn into huge exposures.

5. Consequences and Impact of Attacks:

- **Identity Theft:** Phishing attacks may involve stealing personal information, including names, addresses, social security numbers, or dates of birth. Attackers can use this information to commit identity theft, where they impersonate the victim and carry out various fraudulent activities such as opening new credit accounts, applying for loans, or filing false tax returns. Identity theft can cause long-lasting financial and reputational damage.

- **Data Breaches:** In some cases, phishing attacks are aimed at gaining unauthorized access to sensitive data within organizations. If successful, attackers can infiltrate corporate networks, compromise databases, and gain access to confidential information, including customer data, intellectual property, or trade secrets. Data breaches can lead to legal consequences, loss of customer trust, financial penalties, and reputational damage for the affected organization.

- **Compromised Accounts:** Phishing attacks often involve tricking users into revealing their login credentials for various online accounts, such as email, social media, or cloud storage services. Once attackers gain access to these accounts, they can misuse them for various malicious purposes. They may send further phishing emails from compromised accounts, distribute malware, engage in identity theft, or even launch attacks on other individuals or organizations.

6. Countermeasures and mitigation Strategies:

- **Web Browsing and URL Protection:**
  Implement web filters and firewalls to block access to known phishing websites or malicious URLs.
  Utilize browser security features, such as anti-phishing toolbars or extensions, that can help identify and warn users about potentially malicious websites.
  Security Updates and Patch Management:

- **Regularly update operating systems**, applications, and software with the latest security patches to address known vulnerabilities that attackers might exploit.
  Incident Response and Reporting:

- **Develop and implement an incident response plan** to handle phishing incidents effectively. This includes steps for containing the attack, investigating the extent of the breach, and notifying affected individuals or organizations.
  Encourage users to report suspected phishing attempts promptly, and establish clear channels for reporting incidents to the appropriate IT or security teams.

- **Continuous Monitoring and Testing:**
  Conduct regular security assessments, vulnerability scans, and penetration testing to identify weaknesses in systems, networks, and applications.

- Monitor network traffic, log files, and user behaviour for signs of phishing activity or suspicious behaviour.

7. Conclusion:

Understanding attacks on the Presentation Layer of the OSI model is crucial for maintaining network security. By analyzing real-world case studies and exploring various attack vectors, techniques, and countermeasures, this research report provides valuable insights into the consequences of presentation layer attacks and offers effective strategies to mitigate their impact. Organizations can leverage this knowledge to enhance the security of their network infrastructure and protect critical assets.

8. References:

- https://www.cyberclaire.com/blog/BA2018
- https://brightsec.com/blog/xss/#:~:text=If%20successful%2C%20a%20cross%20site,compromise%20of%20the%20user's%20device.
- https://www.byos.io/blog/types-of-cyber-attacks-osi

RESEARCH ON ATTACKS TARGETING THE OSI MODEL

INTRODUCTION TO OSI MODEL

The OSI (Open Systems Interconnection) model is a conceptual framework that standardizes the functions of a communication system into seven different layers. Each layer has a specific role and interacts with adjacent layers to facilitate communication between different devices and systems. The OSI model helps in understanding and designing network protocols and provides a reference for troubleshooting network issues.

Here are the seven layers of the OSI model, from bottom to top:

1. Physical Layer:
   This layer deals with the physical aspects of data transmission. It defines the electrical, mechanical, and physical specifications for devices, cables, and signals. It includes concepts such as voltage levels, cables, connectors, and data transmission rates.

2. Data Link Layer:
   The data link layer provides error-free transmission of data frames between adjacent network nodes. It establishes and terminates the logical link between devices and ensures data integrity through error detection and correction mechanisms. It also handles flow control and defines protocols for accessing the physical media.

3. Network Layer:
   The network layer is responsible for addressing, routing, and forwarding data packets across different networks. It determines the best path for data transmission, selects appropriate routes, and handles logical addressing. IP (Internet Protocol) operates at this layer.

4. Transport Layer:
   The transport layer ensures reliable and transparent end-to-end communication between hosts. It breaks down data from the upper layers into smaller units, called segments, and provides mechanisms for error recovery, flow control, and congestion control. TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) are examples of transport layer protocols.

5. Session Layer:

   The session layer establishes, manages, and terminates sessions between applications. It provides mechanisms for session establishment, synchronization, and checkpointing. This layer is responsible for maintaining the communication session and managing the dialogue between applications.

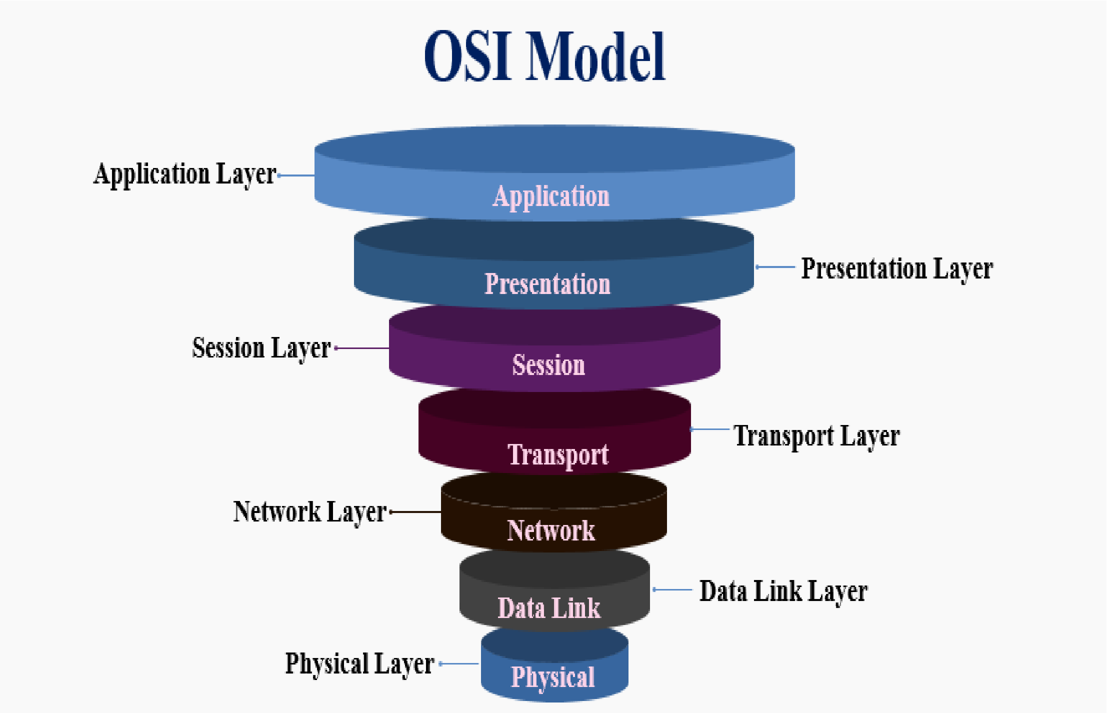
6. Presentation Layer:

   The presentation layer is responsible for data representation and ensures the compatibility of data exchanged between different systems. It handles data encryption, compression, and translation, as well as character encoding and syntax conversion.

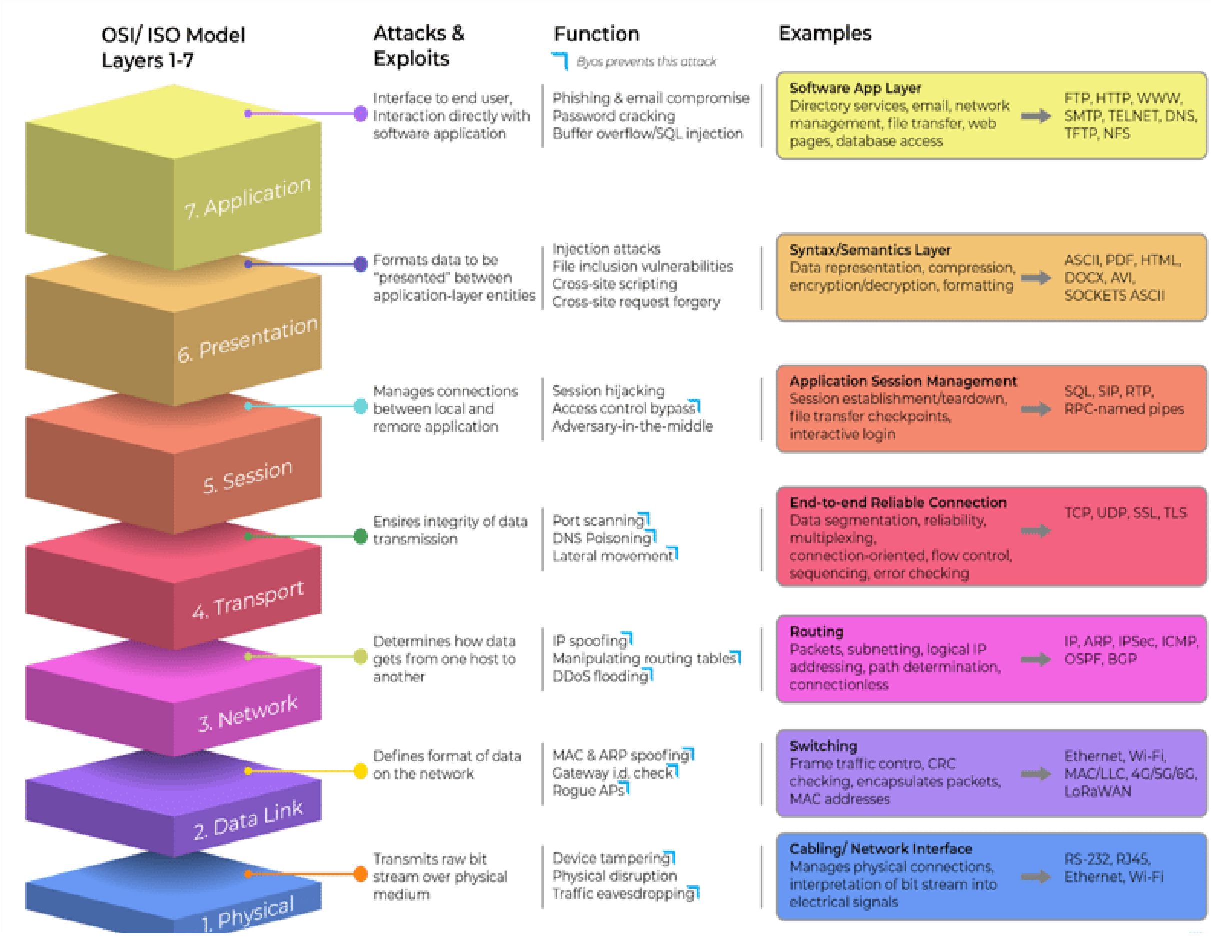
7. Application Layer:

   The application layer represents the user interface and provides services directly to end-user applications. It includes protocols such as HTTP (Hypertext Transfer Protocol), FTP (File Transfer Protocol), SMTP (Simple Mail Transfer Protocol), and DNS (Domain Name System).

By dividing the complex process of communication into these seven layers, the OSI model enables interoperability and standardization among different networking technologies and devices.



ATTACTS ON OSI MODEL LAYERS



⌝ Byos prevents this attack

OSI/ ISO Model Layers 1-7	Attacks & Exploits	Function	Examples	
7. Application	Interface to end user, Interaction directly with software application	Phishing & email compromise Password cracking Buffer overflow/SQL injection	Software App Layer Directory services, email, network management, file transfer, web pages, database access	FTP, HTTP, WWW, SMTP, TELNET, DNS, TFTP, NFS
6. Presentation	Formats data to be "presented" between application-layer entities	Injection attacks File inclusion vulnerabilities Cross-site scripting Cross-site request forgery	Syntax/Semantics Layer Data representation, compression, encryption/decryption, formatting	ASCII, PDF, HTML, DOCX, AVI, SOCKETS ASCII
5. Session	Manages connections between local and remore application	Session hijacking Access control bypass⌝ Adversary-in-the-middle	Application Session Management Session establishment/teardown, file transfer checkpoints, interactive login	SQL, SIP, RTP, RPC-named pipes
4. Transport	Ensires integrity of data transmission	Port scanning⌝ DNS Poisoning⌝ Lateral movement⌝	End-to-end Reliable Connection Data segmentation, reliability, multiplexing, connection-oriented, flow control, sequencing, error checking	TCP, UDP, SSL, TLS
3. Network	Determines how data gets from one host to another	IP spoofing⌝ Manipulating routing tables⌝ DDoS flooding⌝	Routing Packets, subnetting, logical IP addressing, path determination, connectionless	IP, ARP, IPSec, ICMP, OSPF, BGP
2. Data Link	Defines format of data on the network	MAC & ARP spoofing⌝ Gateway i.d. check⌝ Rogue APs⌝	Switching Frame traffic contro, CRC checking, encapsulates packets, MAC addresses	Ethernet, Wi-Fi, MAC/LLC, 4G/5G/6G, LoRaWAN
1. Physical	Transmits raw bit stream over physical medium	Device tampering⌝ Physical disruption Traffic eavesdropping⌝	Cabling/ Network Interface Manages physical connections, interpretation of bit stream into electrical signals	RS-232, RJ45, Ethernet, Wi-Fi

The OSI model defines seven abstraction layers computer systems use to communicate over a network, in order to enable the communication between users. Each OSI model layer has specific functions, which communicate and interact with the layers immediately above and below.

Layers are classified into two categories:

- Host layers: the application layer, the presentation layer, the session layer and the transport layer.
- Media layers: the network layer, the data link layer and the physical layer.

Application layer | 7

The application layer, also known as the “desktop layer”, is responsible for communicating with applications, both host-based and user-facing. This is the layer closest to the user.

It enables network access to application services and allows users to receive data. Besides, it specifies the shared communications protocols and interface methods hosts use in communication networks.

This OSI model layer communicates and interacts with: the presentation layer.

The most common security attack on the application layer is: an exploit attack.

Presentation layer | 6

The presentation layer, also known as the “syntax layer”, is responsible for formatting and translating data into the format the application layer specifies. It is to say, it acts as the network’s data translator to ensure that the data sent out by the application layer is readable by the receiving system’s application layer.

This OSI model layer communicates and interacts with: the application layer and the session layer.

The most common security attack on the presentation layer is: a phishing attack.

Session layer | 5

The session layer is responsible for opening, managing and closing sessions between end-user application processes. It establishes, manages and terminates the connections between local and remote applications.

This host layer creates the setup, controls the connection, ends the teardown between computers, and checkpoints and recovers sessions.

This OSI model layer communicates and interacts with: the presentation layer and the transport layer.

The most common security attack on the session layer is: a hijacking attack.

Transport layer | 4

The transport layer is responsible for providing means of transferring variable-length data sequences from a source host to a destination host. The protocols on this host layer provide end-to-end communication services for applications.

It recognizes two modes, connection-oriented and connectionless, to provide reliable transmission between points on a network.

This OSI model layer communicates and interacts with: the session layer and the network layer.

The most common security attacks on the transport layer are: reconnaissance and DoS attacks.

Network layer | 3

The network layer is responsible for providing means of transferring packets between connected nodes, via one or several networks. It structures and manages multi-node networks, using routers and switches to manage its traffic.

This OSI model layer communicates and interacts with: the transport layer and the data link layer.

The most common security attack on the network layer is: a man-in-the-middle attack.

Data link layer | 2

The data link layer is responsible for transferring data frames between two directly connected nodes, within the same local area network. It packages raw bits from the physical layer into frames. It might also perform error checking and correction.

This OSI model layer communicates and interacts with: the network layer and the physical layer.

The most common security attack on the data link layer is: a spoofing attack.

Physical layer | 1

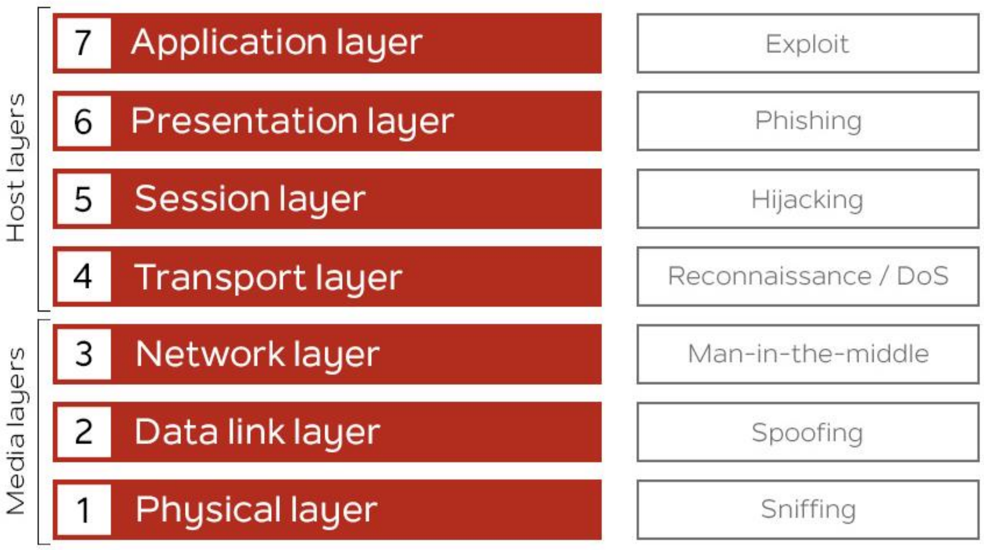
The physical layer is responsible for transmitting and receiving unstructured raw data between devices and physical transmission media. It can be implemented through diverse hardware technologies.

This OSI model layer communicates and interacts with: the data link layer. It translates logical communications requests from the data link layer into hardware-specific operations in order to transmit and receive signals.

The most common security attack on the physical layer is: a sniffing attack.

Types of attacks in the OSI model by layer

These are the different types of attacks that can affect each particular layer of the OSI model.



Host layers	7	Application layer	Exploit
	6	Presentation layer	Phishing
	5	Session layer	Hijacking
	4	Transport layer	Reconnaissance / DoS
Media layers	3	Network layer	Man-in-the-middle
	2	Data link layer	Spoofing
	1	Physical layer	Sniffing

Exploit on the application layer

An exploit consists of taking advantage of vulnerabilities in software applications to gain unauthorized access and take control over a system, and perform diverse types of attacks, such as a denial-of-service attack.

Phishing attacks on the presentation layer

Phishing attacks consist of deceiving individuals into revealing sensitive data through diverse techniques. It is one of the most commonly used cyberattacks nowadays and includes many types of attacks.

Hijacking attacks on the session layer

Hijacking attacks consist of intercepting and taking control of an established communication session either to access sensitive data or to gain unauthorized access to the targeted user’s computer or account.

Reconnaissance and DoS attacks on the transport layer

Reconnaissance attacks consist of gathering information about a system in order to identify its vulnerabilities. Although it was originally used as an ethical hacking technique to identify security loopholes and improve security, it has also become a mechanism to identify vulnerabilities before launching a cyberattack.

DoS attacks, short for “Denial-of-service attacks”, consist of making a resource unavailable to users by flooding the target with superfluous requests that intend to prevent legitimate requests from being fulfilled. The disruption can be either temporary or indefinite. When the attack originates from numerous sources at a time, it is known as Distributed denial-of-service attack or DDoS attack.

Man-in-the-middle attacks on the network layer

Man-in-the-middle attacks, abbreviated as “MitM attacks”, consist of an attacker placing himself between two communicating parties to monitor, relay and even alter the content of messages. While both parties believe to be communicating with each other directly and securely.

This attack is also known under many other names, such as:

- Machine-in-the-middle attack.
- Manipulator-in-the-middle attack.
- Meddler-in-the-middle attack.

Spoofing attacks on the data link layer

Spoofing attacks consist of a person or program falsifying data to identify as an authorized user or device. By impersonating authorized users or devices, attackers can bypass access control to systems, steal data and spread malware.

Sniffing attacks on the physical layer

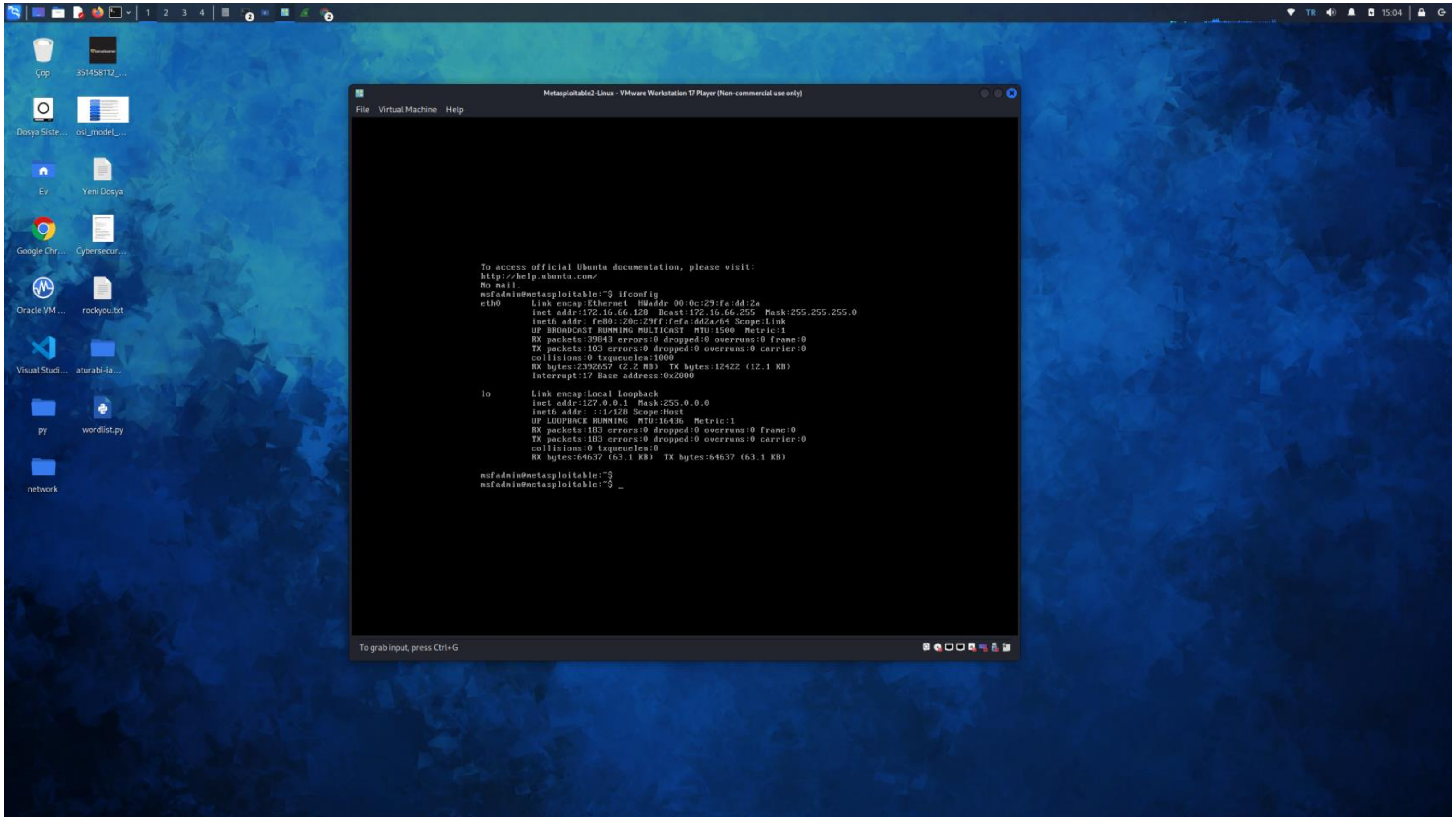
Sniffing attacks consist of intercepting data using a packet sniffer application. Then, if the captured packets are not encrypted, the packet sniffer can be used to read them. This allows attackers to analyze the network and gain information to corrupt it or even cause it to crash.

This 7-layers networking model and the common cyberattacks associated with each of them highlight the importance of assessing risks and vulnerabilities to protect corporate security at all levels. IT threats are commonplace nowadays and cannot be overlooked. As a result, strict approaches to security such as Zero Trust and Disaster Recovery solutions are becoming widely used among organizations to ensure business continuity.

OSI ANALYSIS

TEAM OF ACONİTUM

WE ARE ADDING METASPLOİTABLE 2 VIRTUAL MACHINE

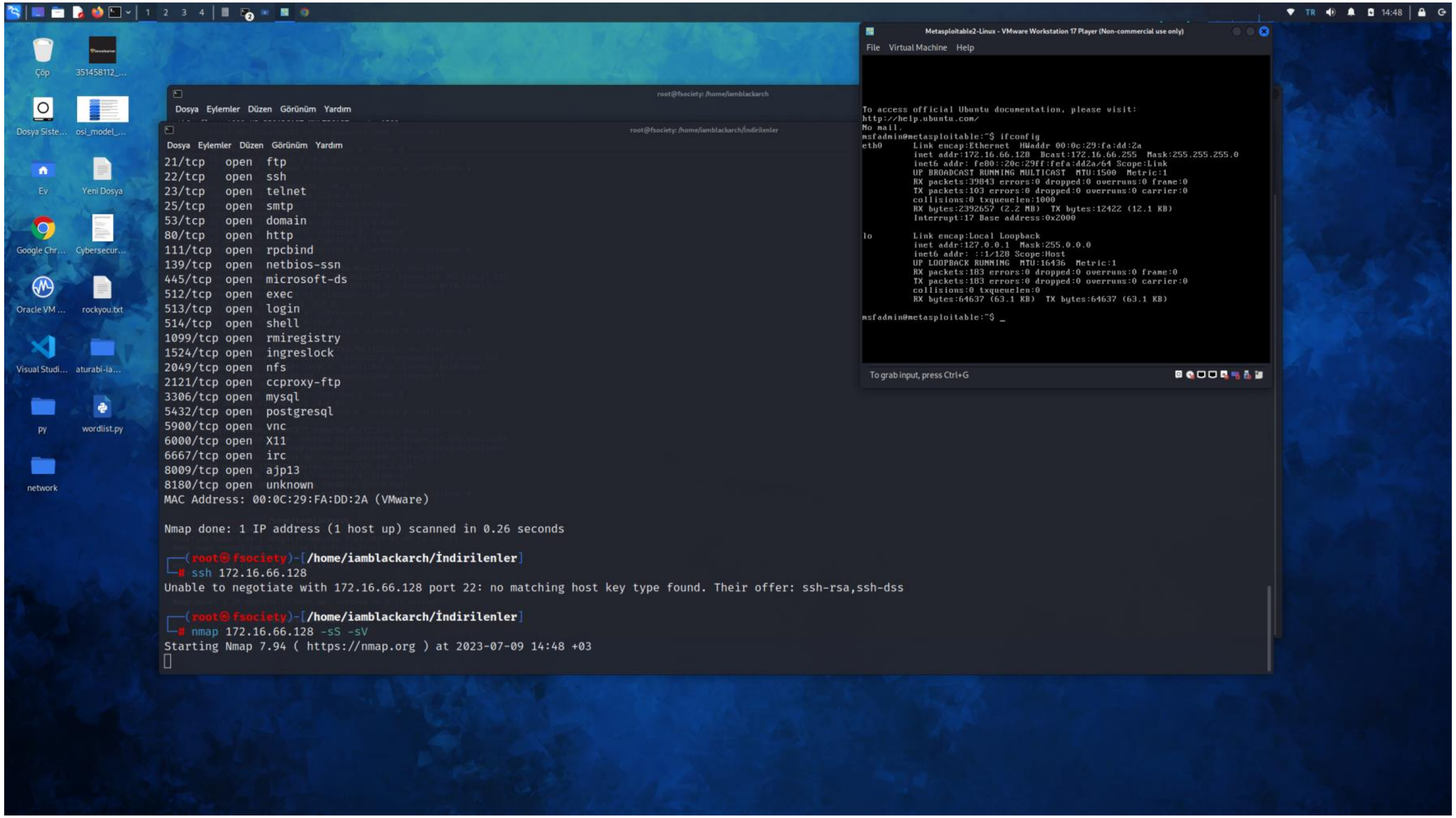


```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:fa:dd:2a
          inet addr:172.16.66.128  Bcast:172.16.66.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fefa:dd2a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:39843 errors:0 dropped:0 overruns:0 frame:0
          TX packets:103 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2392657 (2.2 MB)  TX bytes:12422 (12.1 KB)
          Interrupt:17 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:183 errors:0 dropped:0 overruns:0 frame:0
          TX packets:183 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:64637 (63.1 KB)  TX bytes:64637 (63.1 KB)

msfadmin@metasploitable:~$
msfadmin@metasploitable:~$ _
```

WE LEARN THE IP ADDRESS WITH THE IFCONFIG COMMAND



```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:fa:dd:2a
          inet addr:172.16.66.128  Bcast:172.16.66.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fefa:dd2a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:39843 errors:0 dropped:0 overruns:0 frame:0
          TX packets:103 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2392657 (2.2 MB)  TX bytes:12422 (12.1 KB)
          Interrupt:17 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:183 errors:0 dropped:0 overruns:0 frame:0
          TX packets:183 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:64637 (63.1 KB)  TX bytes:64637 (63.1 KB)

msfadmin@metasploitable:~$ _
```

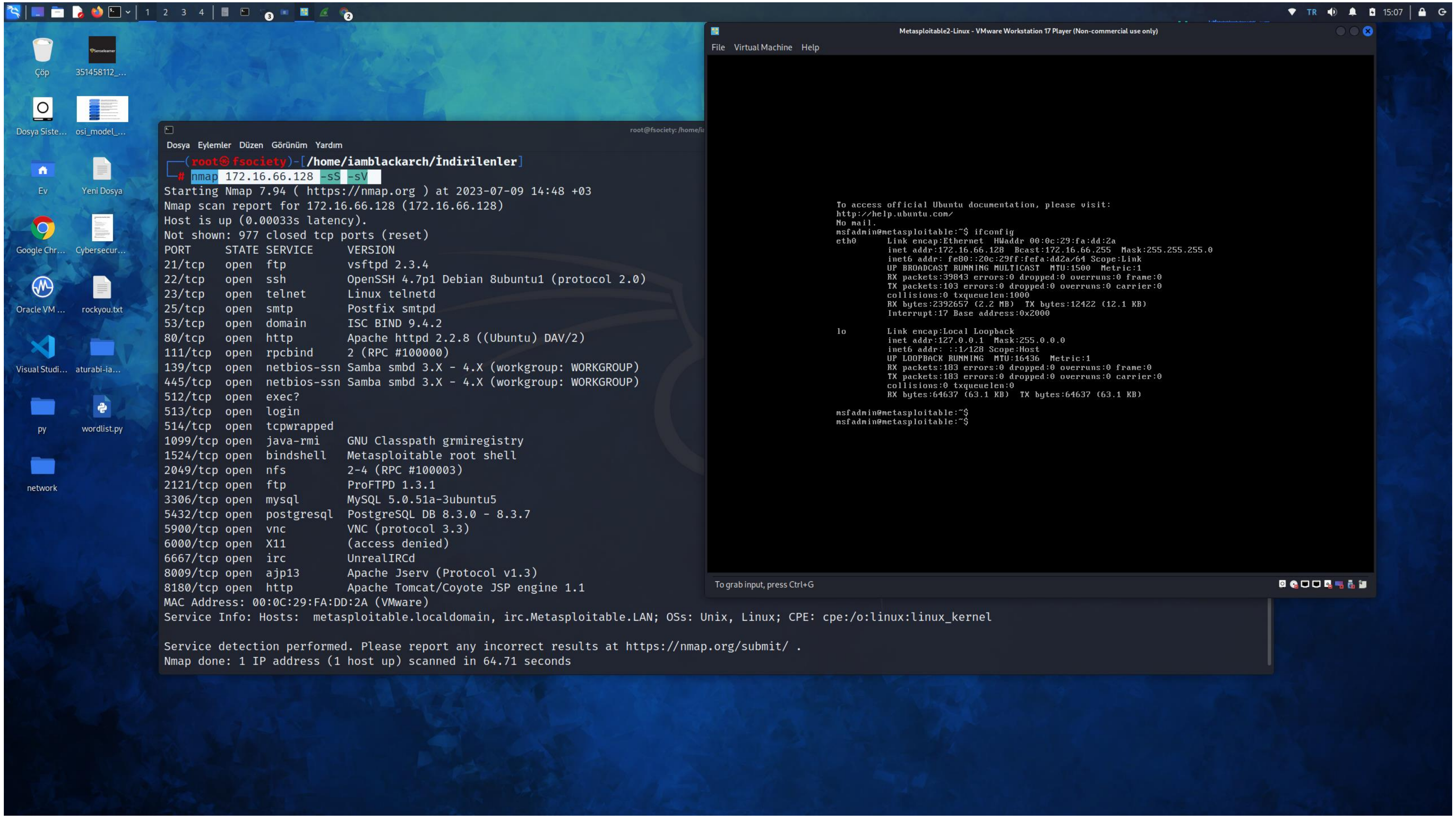
```
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 00:0C:29:FA:DD:2A (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.26 seconds

┌──(root㉿fsociety)-[/home/iamblackarch/İndirilenler]
└─# ssh 172.16.66.128
Unable to negotiate with 172.16.66.128 port 22: no matching host key type found. Their offer: ssh-rsa,ssh-dss

┌──(root㉿fsociety)-[/home/iamblackarch/İndirilenler]
└─# nmap 172.16.66.128 -sS -sV
Starting Nmap 7.94 ( https://nmap.org ) at 2023-07-09 14:48 +03
```

nmap 172.16.66.128 -sS -sV



```
┌──(root㉿fsociety)-[/home/iamblackarch/İndirilenler]
└─# nmap 172.16.66.128 -sS -sV
Starting Nmap 7.94 ( https://nmap.org ) at 2023-07-09 14:48 +03
Nmap scan report for 172.16.66.128 (172.16.66.128)
Host is up (0.00033s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE     VERSION
21/tcp    open  ftp         vsftpd 2.3.4
22/tcp    open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet      Linux telnetd
25/tcp    open  smtp        Postfix smtpd
53/tcp    open  domain      ISC BIND 9.4.2
80/tcp    open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind     2 (RPC #100000)
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec?
513/tcp   open  login
514/tcp   open  tcpwrapped
1099/tcp  open  java-rmi    GNU Classpath grmiregistry
1524/tcp  open  bindshell   Metasploitable root shell
2049/tcp  open  nfs         2-4 (RPC #100003)
2121/tcp  open  ftp         ProFTPD 1.3.1
3306/tcp  open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp  open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc         VNC (protocol 3.3)
6000/tcp  open  X11         (access denied)
6667/tcp  open  irc         UnrealIRCd
8009/tcp  open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp  open  http        Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 00:0C:29:FA:DD:2A (VMware)
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 64.71 seconds
```

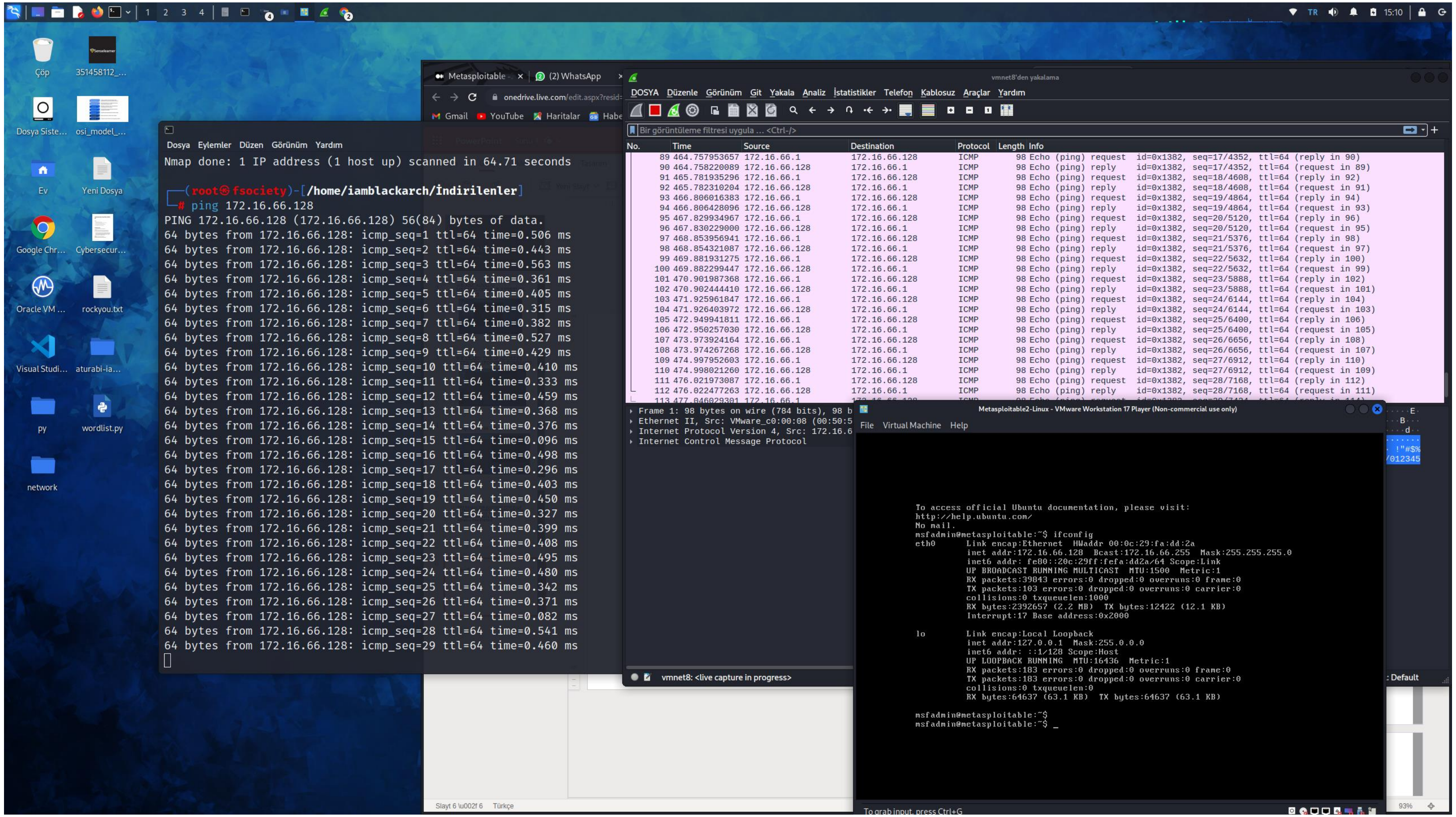
```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:fa:dd:2a
          inet addr:172.16.66.128  Bcast:172.16.66.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fefa:dd2a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:39843 errors:0 dropped:0 overruns:0 frame:0
          TX packets:103 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2392657 (2.2 MB)  TX bytes:12422 (12.1 KB)
          Interrupt:17 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:183 errors:0 dropped:0 overruns:0 frame:0
          TX packets:183 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:64637 (63.1 KB)  TX bytes:64637 (63.1 KB)

msfadmin@metasploitable:~$
msfadmin@metasploitable:~$
```

- THE REASON FOR SCANNING WITH NMAP IS THAT IF THE PORTS ARE OPEN, WE WILL HAVE THE CHANCE TO MONITOR THE OUTGOING PACKETS BY DIRECTLY PINGING THE SYSTEM.

Ping Start!



Metasploitable
(2) WhatsApp
onedrive.live.com/edit.aspx?resid=
Gmail
YouTube
Haritalar
Dosya Eylemler Düzen Görünüm Yardım
vmnet8'den yakalama
DOSYA Düzenle Görünüm Git Yakala Analiz İstatistikler Telefon Kablosuz Araçlar Yardım
Bir görüntüleme filtresi uygula … <Ctrl-/>
vmnet8: <live capture in progress>
Metasploitable2-Linux - VMware Workstation 17 Player (Non-commercial use only)
File Virtual Machine Help
To grab input, press Ctrl+G
Çöp
351458112_…
Dosya Siste…
osi_model_…
Ev
Yeni Dosya
Google Chr…
Cybersecur…
Oracle VM …
rockyou.txt
Visual Studi…
atarabi-ia…
py
wordlist.py
network
15:10
TR
Slayt 6 \u002f 6
Türkçe
93%
: Default

```
Nmap done: 1 IP address (1 host up) scanned in 64.71 seconds

┌──(root㉿fsociety)-[/home/iamblackarch/İndirilenler]
└─# ping 172.16.66.128
PING 172.16.66.128 (172.16.66.128) 56(84) bytes of data.
64 bytes from 172.16.66.128: icmp_seq=1 ttl=64 time=0.506 ms
64 bytes from 172.16.66.128: icmp_seq=2 ttl=64 time=0.443 ms
64 bytes from 172.16.66.128: icmp_seq=3 ttl=64 time=0.563 ms
64 bytes from 172.16.66.128: icmp_seq=4 ttl=64 time=0.361 ms
64 bytes from 172.16.66.128: icmp_seq=5 ttl=64 time=0.405 ms
64 bytes from 172.16.66.128: icmp_seq=6 ttl=64 time=0.315 ms
64 bytes from 172.16.66.128: icmp_seq=7 ttl=64 time=0.382 ms
64 bytes from 172.16.66.128: icmp_seq=8 ttl=64 time=0.527 ms
64 bytes from 172.16.66.128: icmp_seq=9 ttl=64 time=0.429 ms
64 bytes from 172.16.66.128: icmp_seq=10 ttl=64 time=0.410 ms
64 bytes from 172.16.66.128: icmp_seq=11 ttl=64 time=0.333 ms
64 bytes from 172.16.66.128: icmp_seq=12 ttl=64 time=0.459 ms
64 bytes from 172.16.66.128: icmp_seq=13 ttl=64 time=0.368 ms
64 bytes from 172.16.66.128: icmp_seq=14 ttl=64 time=0.376 ms
64 bytes from 172.16.66.128: icmp_seq=15 ttl=64 time=0.096 ms
64 bytes from 172.16.66.128: icmp_seq=16 ttl=64 time=0.498 ms
64 bytes from 172.16.66.128: icmp_seq=17 ttl=64 time=0.296 ms
64 bytes from 172.16.66.128: icmp_seq=18 ttl=64 time=0.403 ms
64 bytes from 172.16.66.128: icmp_seq=19 ttl=64 time=0.450 ms
64 bytes from 172.16.66.128: icmp_seq=20 ttl=64 time=0.327 ms
64 bytes from 172.16.66.128: icmp_seq=21 ttl=64 time=0.399 ms
64 bytes from 172.16.66.128: icmp_seq=22 ttl=64 time=0.408 ms
64 bytes from 172.16.66.128: icmp_seq=23 ttl=64 time=0.495 ms
64 bytes from 172.16.66.128: icmp_seq=24 ttl=64 time=0.480 ms
64 bytes from 172.16.66.128: icmp_seq=25 ttl=64 time=0.342 ms
64 bytes from 172.16.66.128: icmp_seq=26 ttl=64 time=0.371 ms
64 bytes from 172.16.66.128: icmp_seq=27 ttl=64 time=0.082 ms
64 bytes from 172.16.66.128: icmp_seq=28 ttl=64 time=0.541 ms
64 bytes from 172.16.66.128: icmp_seq=29 ttl=64 time=0.460 ms
```

```
No.  Time           Source         Destination    Protocol Length Info
  89 464.757953657 172.16.66.1    172.16.66.128  ICMP     98 Echo (ping) request  id=0x1382, seq=17/4352, ttl=64 (reply in 90)
  90 464.758220089 172.16.66.128  172.16.66.1    ICMP     98 Echo (ping) reply    id=0x1382, seq=17/4352, ttl=64 (request in 89)
  91 465.781935296 172.16.66.1    172.16.66.128  ICMP     98 Echo (ping) request  id=0x1382, seq=18/4608, ttl=64 (reply in 92)
  92 465.782310204 172.16.66.128  172.16.66.1    ICMP     98 Echo (ping) reply    id=0x1382, seq=18/4608, ttl=64 (request in 91)
  93 466.806016383 172.16.66.1    172.16.66.128  ICMP     98 Echo (ping) request  id=0x1382, seq=19/4864, ttl=64 (reply in 94)
  94 466.806428096 172.16.66.128  172.16.66.1    ICMP     98 Echo (ping) reply    id=0x1382, seq=19/4864, ttl=64 (request in 93)
  95 467.829934967 172.16.66.1    172.16.66.128  ICMP     98 Echo (ping) request  id=0x1382, seq=20/5120, ttl=64 (reply in 96)
  96 467.830229000 172.16.66.128  172.16.66.1    ICMP     98 Echo (ping) reply    id=0x1382, seq=20/5120, ttl=64 (request in 95)
  97 468.853956941 172.16.66.1    172.16.66.128  ICMP     98 Echo (ping) request  id=0x1382, seq=21/5376, ttl=64 (reply in 98)
  98 468.854321087 172.16.66.128  172.16.66.1    ICMP     98 Echo (ping) reply    id=0x1382, seq=21/5376, ttl=64 (request in 97)
  99 469.881931275 172.16.66.1    172.16.66.128  ICMP     98 Echo (ping) request  id=0x1382, seq=22/5632, ttl=64 (reply in 100)
 100 469.882299447 172.16.66.128  172.16.66.1    ICMP     98 Echo (ping) reply    id=0x1382, seq=22/5632, ttl=64 (request in 99)
 101 470.901987368 172.16.66.1    172.16.66.128  ICMP     98 Echo (ping) request  id=0x1382, seq=23/5888, ttl=64 (reply in 102)
 102 470.902444410 172.16.66.128  172.16.66.1    ICMP     98 Echo (ping) reply    id=0x1382, seq=23/5888, ttl=64 (request in 101)
 103 471.925961847 172.16.66.1    172.16.66.128  ICMP     98 Echo (ping) request  id=0x1382, seq=24/6144, ttl=64 (reply in 104)
 104 471.926403972 172.16.66.128  172.16.66.1    ICMP     98 Echo (ping) reply    id=0x1382, seq=24/6144, ttl=64 (request in 103)
 105 472.949941811 172.16.66.1    172.16.66.128  ICMP     98 Echo (ping) request  id=0x1382, seq=25/6400, ttl=64 (reply in 106)
 106 472.950257030 172.16.66.128  172.16.66.1    ICMP     98 Echo (ping) reply    id=0x1382, seq=25/6400, ttl=64 (request in 105)
 107 473.973924164 172.16.66.1    172.16.66.128  ICMP     98 Echo (ping) request  id=0x1382, seq=26/6656, ttl=64 (reply in 108)
 108 473.974267268 172.16.66.128  172.16.66.1    ICMP     98 Echo (ping) reply    id=0x1382, seq=26/6656, ttl=64 (request in 107)
 109 474.997952603 172.16.66.1    172.16.66.128  ICMP     98 Echo (ping) request  id=0x1382, seq=27/6912, ttl=64 (reply in 110)
 110 474.998021260 172.16.66.128  172.16.66.1    ICMP     98 Echo (ping) reply    id=0x1382, seq=27/6912, ttl=64 (request in 109)
 111 476.021973087 172.16.66.1    172.16.66.128  ICMP     98 Echo (ping) request  id=0x1382, seq=28/7168, ttl=64 (reply in 112)
 112 476.022477263 172.16.66.128  172.16.66.1    ICMP     98 Echo (ping) reply    id=0x1382, seq=28/7168, ttl=64 (request in 111)
 113 477.046029301 172.16.66.1
```

```
› Frame 1: 98 bytes on wire (784 bits), 98 b
› Ethernet II, Src: VMware_c0:00:08 (00:50:5
› Internet Protocol Version 4, Src: 172.16.6
› Internet Control Message Protocol
```

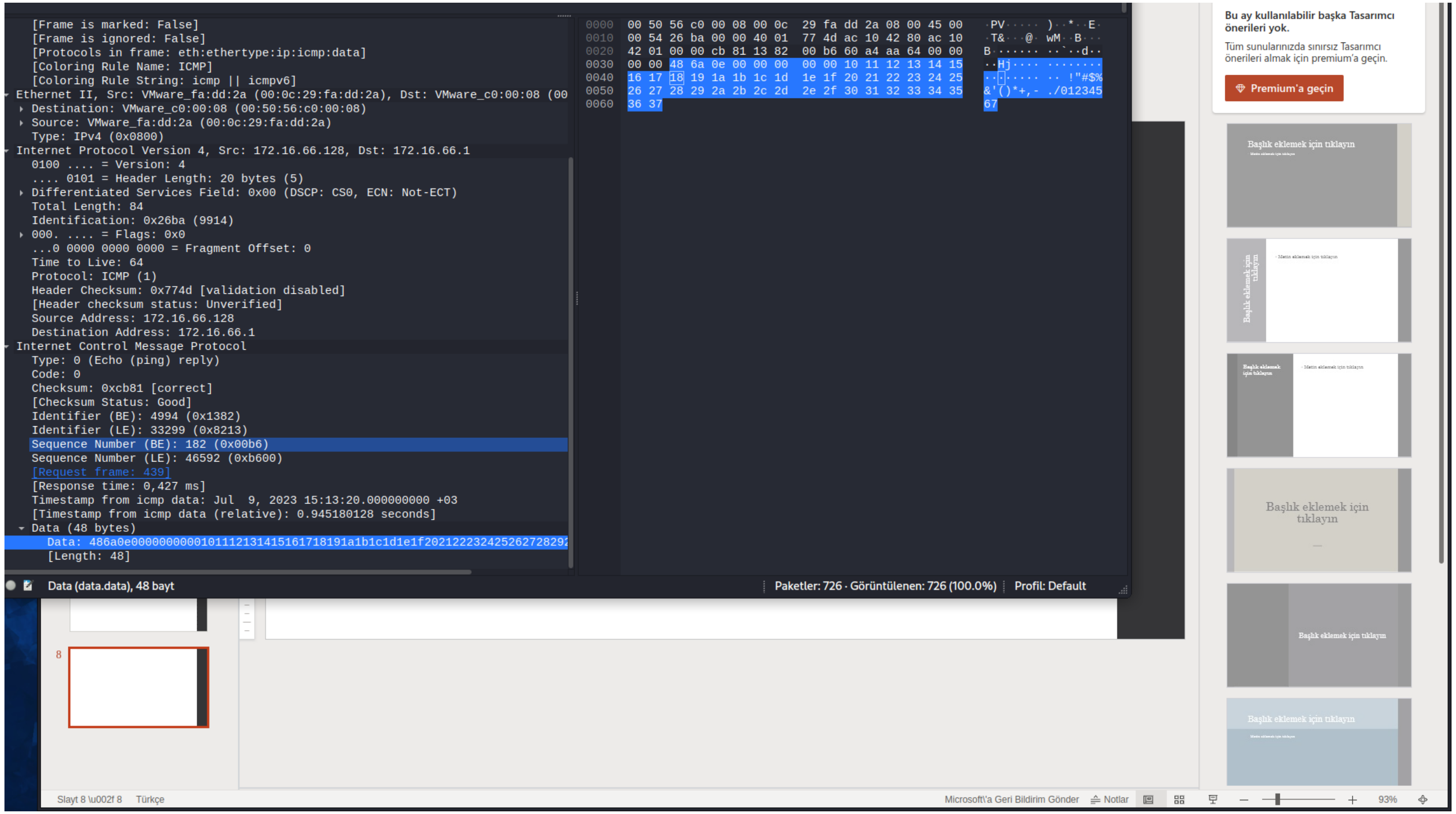
```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:fa:dd:2a
          inet addr:172.16.66.128  Bcast:172.16.66.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fefa:dd2a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:39843 errors:0 dropped:0 overruns:0 frame:0
          TX packets:103 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2392657 (2.2 MB)  TX bytes:12422 (12.1 KB)
          Interrupt:17 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:183 errors:0 dropped:0 overruns:0 frame:0
          TX packets:183 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:64637 (63.1 KB)  TX bytes:64637 (63.1 KB)

msfadmin@metasploitable:~$
msfadmin@metasploitable:~$ _
```

- it is possible to see ICMP packets at the network layer. After pinging we attack the 3rd layer, the network layer.

IN THE MARKED AREA YOU CAN SEE HOW MANY BYTES OF PACKAGE HAVE BEEN SENT. ALSO THE CONTENTS OF THE PACKAGE ARE AVAILABLE.



```
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:icmp:data]
    [Coloring Rule Name: ICMP]
    [Coloring Rule String: icmp || icmpv6]
 Ethernet II, Src: VMware_fa:dd:2a (00:0c:29:fa:dd:2a), Dst: VMware_c0:00:08 (00
    Destination: VMware_c0:00:08 (00:50:56:c0:00:08)
    Source: VMware_fa:dd:2a (00:0c:29:fa:dd:2a)
    Type: IPv4 (0x0800)
 Internet Protocol Version 4, Src: 172.16.66.128, Dst: 172.16.66.1
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 84
    Identification: 0x26ba (9914)
    000. .... = Flags: 0x0
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
    Protocol: ICMP (1)
    Header Checksum: 0x774d [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 172.16.66.128
    Destination Address: 172.16.66.1
 Internet Control Message Protocol
    Type: 0 (Echo (ping) reply)
    Code: 0
    Checksum: 0xcb81 [correct]
    [Checksum Status: Good]
    Identifier (BE): 4994 (0x1382)
    Identifier (LE): 33299 (0x8213)
    Sequence Number (BE): 182 (0x00b6)
    Sequence Number (LE): 46592 (0xb600)
    [Request frame: 439]
    [Response time: 0,427 ms]
    Timestamp from icmp data: Jul  9, 2023 15:13:20.000000000 +03
    [Timestamp from icmp data (relative): 0.945180128 seconds]
    Data (48 bytes)
       Data: 486a0e0000000000101112131415161718191a1b1c1d1e1f202122232425262728292
       [Length: 48]
```

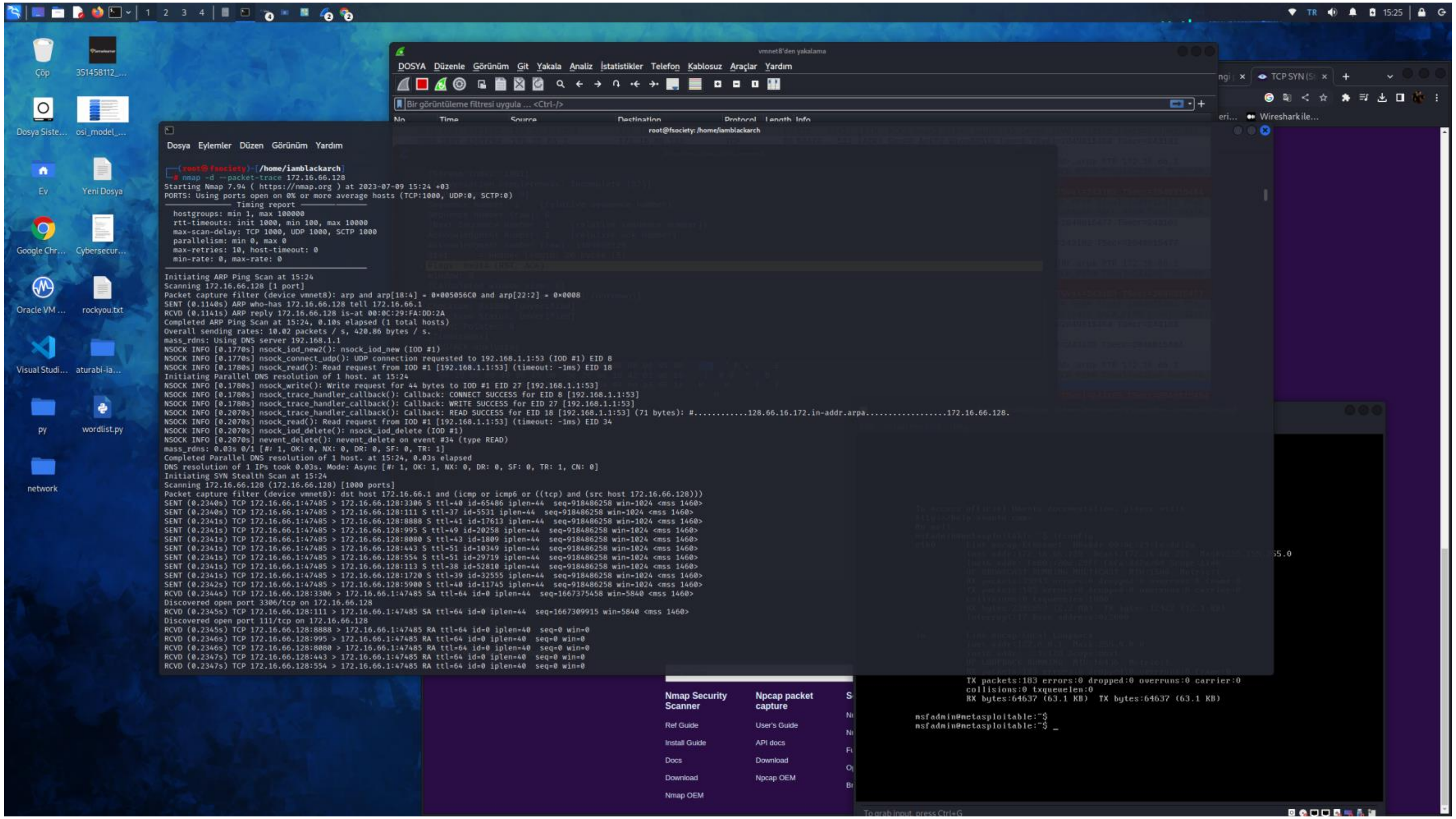
```
0000  00 50 56 c0 00 08 00 0c  29 fa dd 2a 08 00 45 00   ·PV····· )··*··E·
0010  00 54 26 ba 00 00 40 01  77 4d ac 10 42 80 ac 10   ·T&···@· wM··B···
0020  42 01 00 00 cb 81 13 82  00 b6 60 a4 aa 64 00 00   B······· ··`··d··
0030  00 00 48 6a 0e 00 00 00  00 00 10 11 12 13 14 15   ··Hj···· ········
0040  16 17 18 19 1a 1b 1c 1d  1e 1f 20 21 22 23 24 25   ········ ·· !"#$%
0050  26 27 28 29 2a 2b 2c 2d  2e 2f 30 31 32 33 34 35   &'()*+,- ./012345
0060  36 37                                              67
```

Data (data.data), 48 bayt

Paketler: 726 · Görüntülenen: 726 (100.0%) | Profil: Default

Bu ay kullanılabilir başka Tasarımcı önerileri yok.
Tüm sunularınızda sınırsız Tasarımcı önerileri almak için premium'a geçin.
Premium'a geçin
Başlık eklemek için tıklayın
Metin eklemek için tıklayın
8
Slayt 8 \u002f 8
Türkçe
Microsoft\'a Geri Bildirim Gönder
Notlar
93%

SYN ATTACK BY SENDING A TCP PACKET.



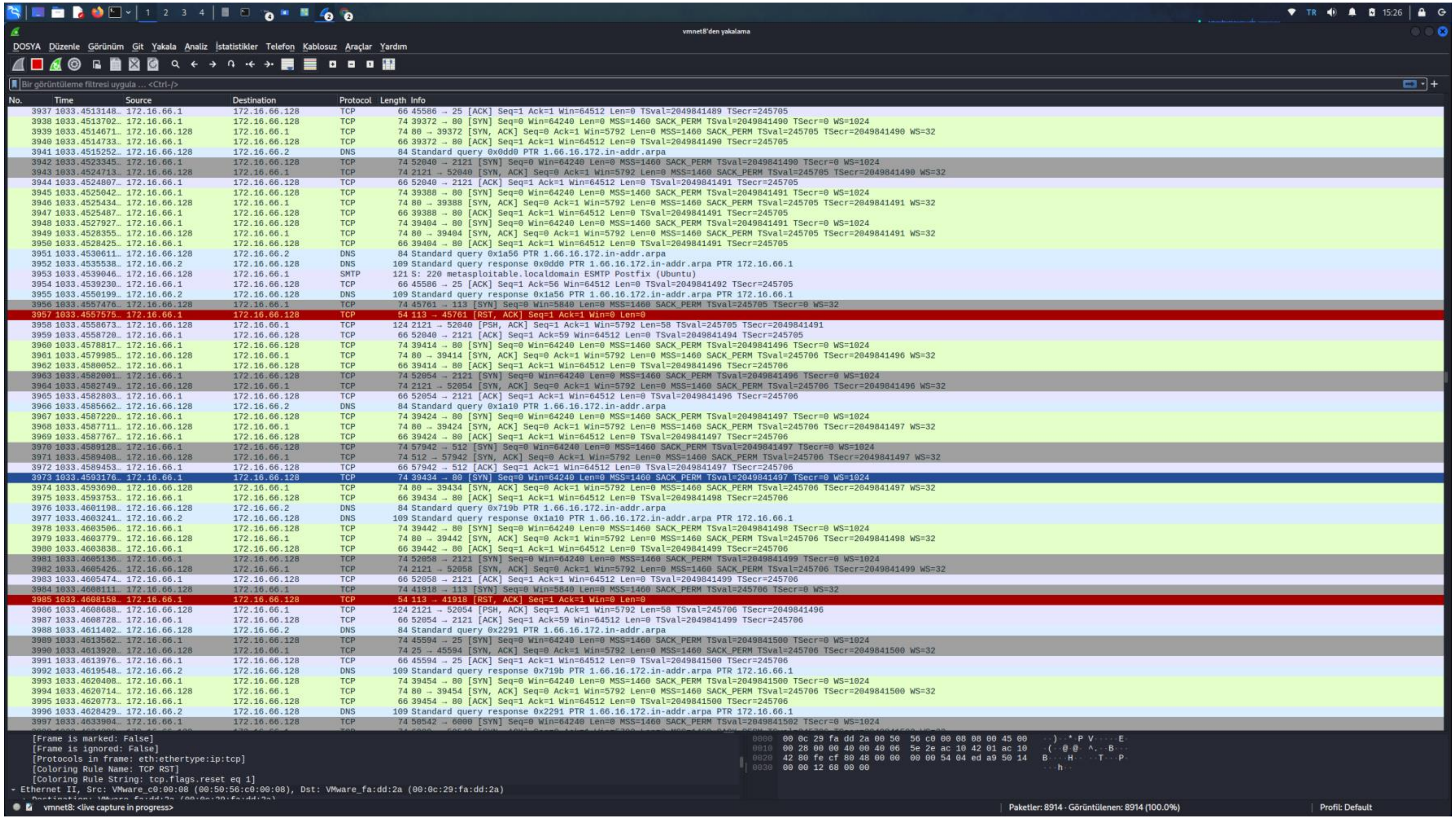
vmnet8'den yakalama
DOSYA Düzenle Görünüm Git Yakala Analiz İstatistikler Telefon Kablosuz Araçlar Yardım
Bir görüntüleme filtresi uygula ... <Ctrl-/>
No. Time Source Destination Protocol Length Info
root@fsociety: /home/iamblackarch
Dosya Eylemler Düzen Görünüm Yardım
TCP SYN
Wireshark ile...
Nmap Security Scanner
Ref Guide
Install Guide
Docs
Download
Nmap OEM
Npcap packet capture
User's Guide
API docs
Download
Npcap OEM
To grab input, press Ctrl+G

```
┌──(root㉿fsociety)-[/home/iamblackarch]
└─# nmap -d --packet-trace 172.16.66.128
Starting Nmap 7.94 ( https://nmap.org ) at 2023-07-09 15:24 +03
PORTS: Using ports open on 0% or more average hosts (TCP:1000, UDP:0, SCTP:0)
--------------- Timing report ---------------
  hostgroups: min 1, max 100000
  rtt-timeouts: init 1000, min 100, max 10000
  max-scan-delay: TCP 1000, UDP 1000, SCTP 1000
  parallelism: min 0, max 0
  max-retries: 10, host-timeout: 0
  min-rate: 0, max-rate: 0
---------------------------------------------
Initiating ARP Ping Scan at 15:24
Scanning 172.16.66.128 [1 port]
Packet capture filter (device vmnet8): arp and arp[18:4] = 0×005056C0 and arp[22:2] = 0×0008
SENT (0.1140s) ARP who-has 172.16.66.128 tell 172.16.66.1
RCVD (0.1141s) ARP reply 172.16.66.128 is-at 00:0C:29:FA:DD:2A
Completed ARP Ping Scan at 15:24, 0.10s elapsed (1 total hosts)
Overall sending rates: 10.02 packets / s, 420.86 bytes / s.
mass_rdns: Using DNS server 192.168.1.1
NSOCK INFO [0.1770s] nsock_iod_new2(): nsock_iod_new (IOD #1)
NSOCK INFO [0.1770s] nsock_connect_udp(): UDP connection requested to 192.168.1.1:53 (IOD #1) EID 8
NSOCK INFO [0.1780s] nsock_read(): Read request from IOD #1 [192.168.1.1:53] (timeout: -1ms) EID 18
Initiating Parallel DNS resolution of 1 host. at 15:24
NSOCK INFO [0.1780s] nsock_write(): Write request for 44 bytes to IOD #1 EID 27 [192.168.1.1:53]
NSOCK INFO [0.1780s] nsock_trace_handler_callback(): Callback: CONNECT SUCCESS for EID 8 [192.168.1.1:53]
NSOCK INFO [0.1780s] nsock_trace_handler_callback(): Callback: WRITE SUCCESS for EID 27 [192.168.1.1:53]
NSOCK INFO [0.2070s] nsock_trace_handler_callback(): Callback: READ SUCCESS for EID 18 [192.168.1.1:53] (71 bytes): #...........128.66.16.172.in-addr.arpa.................172.16.66.128.
NSOCK INFO [0.2070s] nsock_read(): Read request from IOD #1 [192.168.1.1:53] (timeout: -1ms) EID 34
NSOCK INFO [0.2070s] nsock_iod_delete(): nsock_iod_delete (IOD #1)
NSOCK INFO [0.2070s] nevent_delete(): nevent_delete on event #34 (type READ)
mass_rdns: 0.03s 0/1 [#: 1, OK: 0, NX: 0, DR: 0, SF: 0, TR: 1]
Completed Parallel DNS resolution of 1 host. at 15:24, 0.03s elapsed
DNS resolution of 1 IPs took 0.03s. Mode: Async [#: 1, OK: 1, NX: 0, DR: 0, SF: 0, TR: 1, CN: 0]
Initiating SYN Stealth Scan at 15:24
Scanning 172.16.66.128 (172.16.66.128) [1000 ports]
Packet capture filter (device vmnet8): dst host 172.16.66.1 and (icmp or icmp6 or ((tcp) and (src host 172.16.66.128)))
SENT (0.2340s) TCP 172.16.66.1:47485 > 172.16.66.128:3306 S ttl=40 id=65486 iplen=44  seq=918486258 win=1024 <mss 1460>
SENT (0.2340s) TCP 172.16.66.1:47485 > 172.16.66.128:111 S ttl=37 id=5531 iplen=44  seq=918486258 win=1024 <mss 1460>
SENT (0.2341s) TCP 172.16.66.1:47485 > 172.16.66.128:8888 S ttl=41 id=17613 iplen=44  seq=918486258 win=1024 <mss 1460>
SENT (0.2341s) TCP 172.16.66.1:47485 > 172.16.66.128:995 S ttl=49 id=20258 iplen=44  seq=918486258 win=1024 <mss 1460>
SENT (0.2341s) TCP 172.16.66.1:47485 > 172.16.66.128:8080 S ttl=43 id=1809 iplen=44  seq=918486258 win=1024 <mss 1460>
SENT (0.2341s) TCP 172.16.66.1:47485 > 172.16.66.128:443 S ttl=51 id=10349 iplen=44  seq=918486258 win=1024 <mss 1460>
SENT (0.2341s) TCP 172.16.66.1:47485 > 172.16.66.128:554 S ttl=51 id=29719 iplen=44  seq=918486258 win=1024 <mss 1460>
SENT (0.2341s) TCP 172.16.66.1:47485 > 172.16.66.128:113 S ttl=38 id=52810 iplen=44  seq=918486258 win=1024 <mss 1460>
SENT (0.2341s) TCP 172.16.66.1:47485 > 172.16.66.128:1720 S ttl=39 id=32555 iplen=44  seq=918486258 win=1024 <mss 1460>
SENT (0.2342s) TCP 172.16.66.1:47485 > 172.16.66.128:5900 S ttl=40 id=11745 iplen=44  seq=918486258 win=1024 <mss 1460>
RCVD (0.2344s) TCP 172.16.66.128:3306 > 172.16.66.1:47485 SA ttl=64 id=0 iplen=44  seq=1667375458 win=5840 <mss 1460>
Discovered open port 3306/tcp on 172.16.66.128
RCVD (0.2345s) TCP 172.16.66.128:111 > 172.16.66.1:47485 SA ttl=64 id=0 iplen=44  seq=1667309915 win=5840 <mss 1460>
Discovered open port 111/tcp on 172.16.66.128
RCVD (0.2345s) TCP 172.16.66.128:8888 > 172.16.66.1:47485 RA ttl=64 id=0 iplen=40  seq=0 win=0
RCVD (0.2346s) TCP 172.16.66.128:995 > 172.16.66.1:47485 RA ttl=64 id=0 iplen=40  seq=0 win=0
RCVD (0.2346s) TCP 172.16.66.128:8080 > 172.16.66.1:47485 RA ttl=64 id=0 iplen=40  seq=0 win=0
RCVD (0.2347s) TCP 172.16.66.128:443 > 172.16.66.1:47485 RA ttl=64 id=0 iplen=40  seq=0 win=0
RCVD (0.2347s) TCP 172.16.66.128:554 > 172.16.66.1:47485 RA ttl=64 id=0 iplen=40  seq=0 win=0
```

```
          TX packets:183 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:64637 (63.1 KB)  TX bytes:64637 (63.1 KB)

msfadmin@metasploitable:~$
msfadmin@metasploitable:~$ _
```

WİRESHARK SHOW



```
No.   Time            Source         Destination    Protocol Length Info
 3937 1033.4513148… 172.16.66.1    172.16.66.128  TCP       66 45586 → 25 [ACK] Seq=1 Ack=1 Win=64512 Len=0 TSval=2049841489 TSecr=245705
 3938 1033.4513702… 172.16.66.1    172.16.66.128  TCP       74 39372 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2049841490 TSecr=0 WS=1024
 3939 1033.4514671… 172.16.66.128  172.16.66.1    TCP       74 80 → 39372 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TSval=245705 TSecr=2049841490 WS=32
 3940 1033.4514733… 172.16.66.1    172.16.66.128  TCP       66 39372 → 80 [ACK] Seq=1 Ack=1 Win=64512 Len=0 TSval=2049841490 TSecr=245705
 3941 1033.4515252… 172.16.66.128  172.16.66.2    DNS       84 Standard query 0x0dd0 PTR 1.66.16.172.in-addr.arpa
 3942 1033.4523345… 172.16.66.1    172.16.66.128  TCP       74 52040 → 2121 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2049841490 TSecr=0 WS=1024
 3943 1033.4524713… 172.16.66.128  172.16.66.1    TCP       74 2121 → 52040 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TSval=245705 TSecr=2049841490 WS=32
 3944 1033.4524807… 172.16.66.1    172.16.66.128  TCP       66 52040 → 2121 [ACK] Seq=1 Ack=1 Win=64512 Len=0 TSval=2049841491 TSecr=245705
 3945 1033.4525042… 172.16.66.1    172.16.66.128  TCP       74 39388 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2049841491 TSecr=0 WS=1024
 3946 1033.4525434… 172.16.66.128  172.16.66.1    TCP       74 80 → 39388 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TSval=245705 TSecr=2049841491 WS=32
 3947 1033.4525487… 172.16.66.1    172.16.66.128  TCP       66 39388 → 80 [ACK] Seq=1 Ack=1 Win=64512 Len=0 TSval=2049841491 TSecr=245705
 3948 1033.4527927… 172.16.66.1    172.16.66.128  TCP       74 39404 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2049841491 TSecr=0 WS=1024
 3949 1033.4528355… 172.16.66.128  172.16.66.1    TCP       74 80 → 39404 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TSval=245705 TSecr=2049841491 WS=32
 3950 1033.4528425… 172.16.66.1    172.16.66.128  TCP       66 39404 → 80 [ACK] Seq=1 Ack=1 Win=64512 Len=0 TSval=2049841491 TSecr=245705
 3951 1033.4530611… 172.16.66.128  172.16.66.2    DNS       84 Standard query 0x1a56 PTR 1.66.16.172.in-addr.arpa
 3952 1033.4535538… 172.16.66.2    172.16.66.128  DNS      109 Standard query response 0x0dd0 PTR 1.66.16.172.in-addr.arpa PTR 172.16.66.1
 3953 1033.4539046… 172.16.66.128  172.16.66.1    SMTP     121 S: 220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
 3954 1033.4539230… 172.16.66.1    172.16.66.128  TCP       66 45586 → 25 [ACK] Seq=1 Ack=56 Win=64512 Len=0 TSval=2049841492 TSecr=245705
 3955 1033.4550199… 172.16.66.2    172.16.66.128  DNS      109 Standard query response 0x1a56 PTR 1.66.16.172.in-addr.arpa PTR 172.16.66.1
 3956 1033.4557476… 172.16.66.128  172.16.66.1    TCP       74 45761 → 113 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM TSval=245705 TSecr=0 WS=32
 3957 1033.4557575… 172.16.66.1    172.16.66.128  TCP       54 113 → 45761 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 3958 1033.4558673… 172.16.66.128  172.16.66.1    TCP      124 2121 → 52040 [PSH, ACK] Seq=1 Ack=1 Win=5792 Len=58 TSval=245705 TSecr=2049841491
 3959 1033.4558720… 172.16.66.1    172.16.66.128  TCP       66 52040 → 2121 [ACK] Seq=1 Ack=59 Win=64512 Len=0 TSval=2049841494 TSecr=245705
 3960 1033.4578817… 172.16.66.1    172.16.66.128  TCP       74 39414 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2049841496 TSecr=0 WS=1024
 3961 1033.4579985… 172.16.66.128  172.16.66.1    TCP       74 80 → 39414 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TSval=245706 TSecr=2049841496 WS=32
 3962 1033.4580052… 172.16.66.1    172.16.66.128  TCP       66 39414 → 80 [ACK] Seq=1 Ack=1 Win=64512 Len=0 TSval=2049841496 TSecr=245706
 3963 1033.4582001… 172.16.66.1    172.16.66.128  TCP       74 52054 → 2121 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2049841496 TSecr=0 WS=1024
 3964 1033.4582749… 172.16.66.128  172.16.66.1    TCP       74 2121 → 52054 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TSval=245706 TSecr=2049841496 WS=32
 3965 1033.4582803… 172.16.66.1    172.16.66.128  TCP       66 52054 → 2121 [ACK] Seq=1 Ack=1 Win=64512 Len=0 TSval=2049841496 TSecr=245706
 3966 1033.4585662… 172.16.66.128  172.16.66.2    DNS       84 Standard query 0x1a10 PTR 1.66.16.172.in-addr.arpa
 3967 1033.4587220… 172.16.66.1    172.16.66.128  TCP       74 39424 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2049841497 TSecr=0 WS=1024
 3968 1033.4587711… 172.16.66.128  172.16.66.1    TCP       74 80 → 39424 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TSval=245706 TSecr=2049841497 WS=32
 3969 1033.4587767… 172.16.66.1    172.16.66.128  TCP       66 39424 → 80 [ACK] Seq=1 Ack=1 Win=64512 Len=0 TSval=2049841497 TSecr=245706
 3970 1033.4589128… 172.16.66.1    172.16.66.128  TCP       74 57942 → 512 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2049841497 TSecr=0 WS=1024
 3971 1033.4589408… 172.16.66.128  172.16.66.1    TCP       74 512 → 57942 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TSval=245706 TSecr=2049841497 WS=32
 3972 1033.4589453… 172.16.66.1    172.16.66.128  TCP       66 57942 → 512 [ACK] Seq=1 Ack=1 Win=64512 Len=0 TSval=2049841497 TSecr=245706
 3973 1033.4593176… 172.16.66.1    172.16.66.128  TCP       74 39434 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2049841497 TSecr=0 WS=1024
 3974 1033.4593690… 172.16.66.128  172.16.66.1    TCP       74 80 → 39434 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TSval=245706 TSecr=2049841497 WS=32
 3975 1033.4593753… 172.16.66.1    172.16.66.128  TCP       66 39434 → 80 [ACK] Seq=1 Ack=1 Win=64512 Len=0 TSval=2049841498 TSecr=245706
 3976 1033.4601198… 172.16.66.128  172.16.66.2    DNS       84 Standard query 0x719b PTR 1.66.16.172.in-addr.arpa
 3977 1033.4603241… 172.16.66.2    172.16.66.128  DNS      109 Standard query response 0x1a10 PTR 1.66.16.172.in-addr.arpa PTR 172.16.66.1
 3978 1033.4603506… 172.16.66.1    172.16.66.128  TCP       74 39442 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2049841498 TSecr=0 WS=1024
 3979 1033.4603779… 172.16.66.128  172.16.66.1    TCP       74 80 → 39442 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TSval=245706 TSecr=2049841498 WS=32
 3980 1033.4603838… 172.16.66.1    172.16.66.128  TCP       66 39442 → 80 [ACK] Seq=1 Ack=1 Win=64512 Len=0 TSval=2049841499 TSecr=245706
 3981 1033.4605136… 172.16.66.1    172.16.66.128  TCP       74 52058 → 2121 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2049841499 TSecr=0 WS=1024
 3982 1033.4605426… 172.16.66.128  172.16.66.1    TCP       74 2121 → 52058 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TSval=245706 TSecr=2049841499 WS=32
 3983 1033.4605474… 172.16.66.1    172.16.66.128  TCP       66 52058 → 2121 [ACK] Seq=1 Ack=1 Win=64512 Len=0 TSval=2049841499 TSecr=245706
 3984 1033.4608111… 172.16.66.128  172.16.66.1    TCP       74 41918 → 113 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM TSval=245706 TSecr=0 WS=32
 3985 1033.4608158… 172.16.66.1    172.16.66.128  TCP       54 113 → 41918 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 3986 1033.4608688… 172.16.66.128  172.16.66.1    TCP      124 2121 → 52054 [PSH, ACK] Seq=1 Ack=1 Win=5792 Len=58 TSval=245706 TSecr=2049841496
 3987 1033.4608728… 172.16.66.1    172.16.66.128  TCP       66 52054 → 2121 [ACK] Seq=1 Ack=59 Win=64512 Len=0 TSval=2049841499 TSecr=245706
 3988 1033.4611402… 172.16.66.128  172.16.66.2    DNS       84 Standard query 0x2291 PTR 1.66.16.172.in-addr.arpa
 3989 1033.4613562… 172.16.66.1    172.16.66.128  TCP       74 45594 → 25 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2049841500 TSecr=0 WS=1024
 3990 1033.4613920… 172.16.66.128  172.16.66.1    TCP       74 25 → 45594 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TSval=245706 TSecr=2049841500 WS=32
 3991 1033.4613976… 172.16.66.1    172.16.66.128  TCP       66 45594 → 25 [ACK] Seq=1 Ack=1 Win=64512 Len=0 TSval=2049841500 TSecr=245706
 3992 1033.4619548… 172.16.66.2    172.16.66.128  DNS      109 Standard query response 0x719b PTR 1.66.16.172.in-addr.arpa PTR 172.16.66.1
 3993 1033.4620408… 172.16.66.1    172.16.66.128  TCP       74 39454 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2049841500 TSecr=0 WS=1024
 3994 1033.4620714… 172.16.66.128  172.16.66.1    TCP       74 80 → 39454 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TSval=245706 TSecr=2049841500 WS=32
 3995 1033.4620773… 172.16.66.1    172.16.66.128  TCP       66 39454 → 80 [ACK] Seq=1 Ack=1 Win=64512 Len=0 TSval=2049841500 TSecr=245706
 3996 1033.4628429… 172.16.66.2    172.16.66.128  DNS      109 Standard query response 0x2291 PTR 1.66.16.172.in-addr.arpa PTR 172.16.66.1
 3997 1033.4633904… 172.16.66.1    172.16.66.128  TCP       74 50542 → 6000 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2049841502 TSecr=0 WS=1024
```

```
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:tcp]
    [Coloring Rule Name: TCP RST]
    [Coloring Rule String: tcp.flags.reset eq 1]
- Ethernet II, Src: VMware_c0:00:08 (00:50:56:c0:00:08), Dst: VMware_fa:dd:2a (00:0c:29:fa:dd:2a)
```

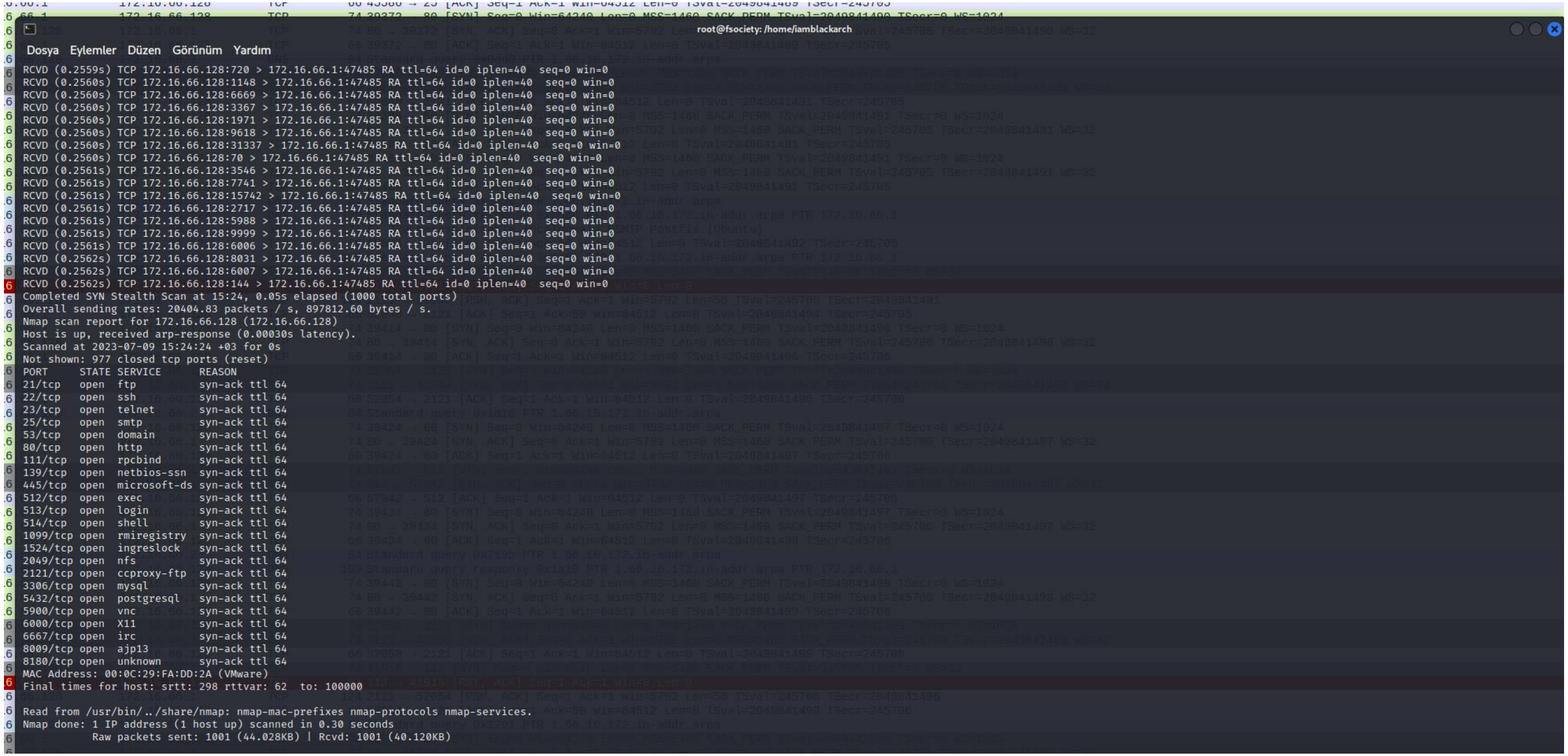
```
0000  00 0c 29 fa dd 2a 00 50  56 c0 00 08 08 00 45 00   ··)··*·P V·····E·
0010  00 28 00 00 40 00 40 06  5e 2e ac 10 42 01 ac 10   ·(··@·@· ^.··B···
0020  42 80 fe cf 80 48 00 00  00 00 54 04 ed a9 50 14   B····H·· ··T···P·
0030  00 00 12 68 00 00                                  ···h··
```

vmnet8: <live capture in progress>

Paketler: 8914 · Görüntülenen: 8914 (100.0%)

Profil: Default

THE PORTS AFFECTED BY THIS ATTACK ON THE TRANSPORT LAYER ARE AS FOLLOWS.


root@fsociety: /home/iamblackarch
Dosya Eylemler Düzen Görünüm Yardım

```
RCVD (0.2559s) TCP 172.16.66.128:720 > 172.16.66.1:47485 RA ttl=64 id=0 iplen=40  seq=0 win=0
RCVD (0.2560s) TCP 172.16.66.128:1148 > 172.16.66.1:47485 RA ttl=64 id=0 iplen=40  seq=0 win=0
RCVD (0.2560s) TCP 172.16.66.128:6669 > 172.16.66.1:47485 RA ttl=64 id=0 iplen=40  seq=0 win=0
RCVD (0.2560s) TCP 172.16.66.128:3367 > 172.16.66.1:47485 RA ttl=64 id=0 iplen=40  seq=0 win=0
RCVD (0.2560s) TCP 172.16.66.128:1971 > 172.16.66.1:47485 RA ttl=64 id=0 iplen=40  seq=0 win=0
RCVD (0.2560s) TCP 172.16.66.128:9618 > 172.16.66.1:47485 RA ttl=64 id=0 iplen=40  seq=0 win=0
RCVD (0.2560s) TCP 172.16.66.128:31337 > 172.16.66.1:47485 RA ttl=64 id=0 iplen=40  seq=0 win=0
RCVD (0.2560s) TCP 172.16.66.128:70 > 172.16.66.1:47485 RA ttl=64 id=0 iplen=40  seq=0 win=0
RCVD (0.2561s) TCP 172.16.66.128:3546 > 172.16.66.1:47485 RA ttl=64 id=0 iplen=40  seq=0 win=0
RCVD (0.2561s) TCP 172.16.66.128:7741 > 172.16.66.1:47485 RA ttl=64 id=0 iplen=40  seq=0 win=0
RCVD (0.2561s) TCP 172.16.66.128:15742 > 172.16.66.1:47485 RA ttl=64 id=0 iplen=40  seq=0 win=0
RCVD (0.2561s) TCP 172.16.66.128:2717 > 172.16.66.1:47485 RA ttl=64 id=0 iplen=40  seq=0 win=0
RCVD (0.2561s) TCP 172.16.66.128:5988 > 172.16.66.1:47485 RA ttl=64 id=0 iplen=40  seq=0 win=0
RCVD (0.2561s) TCP 172.16.66.128:9999 > 172.16.66.1:47485 RA ttl=64 id=0 iplen=40  seq=0 win=0
RCVD (0.2561s) TCP 172.16.66.128:6006 > 172.16.66.1:47485 RA ttl=64 id=0 iplen=40  seq=0 win=0
RCVD (0.2562s) TCP 172.16.66.128:8031 > 172.16.66.1:47485 RA ttl=64 id=0 iplen=40  seq=0 win=0
RCVD (0.2562s) TCP 172.16.66.128:6007 > 172.16.66.1:47485 RA ttl=64 id=0 iplen=40  seq=0 win=0
RCVD (0.2562s) TCP 172.16.66.128:144 > 172.16.66.1:47485 RA ttl=64 id=0 iplen=40  seq=0 win=0
Completed SYN Stealth Scan at 15:24, 0.05s elapsed (1000 total ports)
Overall sending rates: 20404.83 packets / s, 897812.60 bytes / s.
Nmap scan report for 172.16.66.128 (172.16.66.128)
Host is up, received arp-response (0.00030s latency).
Scanned at 2023-07-09 15:24:24 +03 for 0s
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE      REASON
21/tcp   open  ftp          syn-ack ttl 64
22/tcp   open  ssh          syn-ack ttl 64
23/tcp   open  telnet       syn-ack ttl 64
25/tcp   open  smtp         syn-ack ttl 64
53/tcp   open  domain       syn-ack ttl 64
80/tcp   open  http         syn-ack ttl 64
111/tcp  open  rpcbind      syn-ack ttl 64
139/tcp  open  netbios-ssn  syn-ack ttl 64
445/tcp  open  microsoft-ds syn-ack ttl 64
512/tcp  open  exec         syn-ack ttl 64
513/tcp  open  login        syn-ack ttl 64
514/tcp  open  shell        syn-ack ttl 64
1099/tcp open  rmiregistry  syn-ack ttl 64
1524/tcp open  ingreslock   syn-ack ttl 64
2049/tcp open  nfs          syn-ack ttl 64
2121/tcp open  ccproxy-ftp  syn-ack ttl 64
3306/tcp open  mysql        syn-ack ttl 64
5432/tcp open  postgresql   syn-ack ttl 64
5900/tcp open  vnc          syn-ack ttl 64
6000/tcp open  X11          syn-ack ttl 64
6667/tcp open  irc          syn-ack ttl 64
8009/tcp open  ajp13        syn-ack ttl 64
8180/tcp open  unknown      syn-ack ttl 64
MAC Address: 00:0C:29:FA:DD:2A (VMware)
Final times for host: srtt: 298 rttvar: 62  to: 100000

Read from /usr/bin/../share/nmap: nmap-mac-prefixes nmap-protocols nmap-services.
Nmap done: 1 IP address (1 host up) scanned in 0.30 seconds
           Raw packets sent: 1001 (44.028KB) | Rcvd: 1001 (40.120KB)
```

PORT CLOSE

- service iptables stop
- iptables -I INPUT -p tcp --dport -j REJECT
- iptables -I INPUT -p udp --dport -j REJECT
- service iptables save
- service iptables start

PORT OPENED

- service iptables stop
- iptables -I INPUT -p tcp --dport -j ACCEPT
- iptables -I INPUT -p udp --dport -j ACCEPT
- service iptables save
- service iptables start

To shut down requests from the inside to the outside

- service iptables stop
- iptables -I OUTPUT -p tcp --dport -j REJECT
- iptables -I OUTPUT -p udp --dport -j REJECT
- service iptables save
- service iptables start

INCOMING PACKETS CAN BE FILTERED USING THE FREWALL CONFIGURATION.

- THANK YOU

REFERENCES

- https://www.koraykey.com/?p=3642
- https://elfanet.com.tr/tr/main/article/osi-katmanlari/51
- https://kadergultekin.medium.com/ping-hangi-port-%C3%BCzerinde-%C3%A7al%C4%B1%C5%9F%C4%B1r-c386e44386b7
- https://www.youtube.com/watch?v=lTFomlqPIRg
- https://tr.wikipedia.org/wiki/OSI_modeli
- https://ahmetsakarr.medium.com/metasploitable-2-35408f2ce0b0
- https://docs.rapid7.com/metasploit/discovery-scan/
- https://blog.cliaweb.com/linux-port-acma-komutlari.html
- https://bidb.itu.edu.tr/seyir-defteri/blog/2013/09/07/osi-katmanlar%C4%B1