A close up of a logo

Description automatically generatedA close-up of a logo

Description automatically generatedComplete Lab – Using OSINT Tools

**Practical Lab3B: Exploiting Wired and Wireless Networks**

CSE420- Ethical Hacking (Spring 23-24)

**Deadline:** May 5th, 2024 @ 23:59

# Background

Network-based vulnerabilities and exploits can be catastrophic because of the types of damage and impact they can cause in an organization. The following are some examples of network-based attacks and exploits:

Here are definitions for each of the mentioned security topics:

* **Windows name resolution-based attacks and exploits:** These are attacks targeting the Windows operating system's name resolution mechanisms, such as DNS (Domain Name System) and NetBIOS (Network Basic Input/Output System). Attackers exploit vulnerabilities in these protocols to perform various attacks, including DNS spoofing, DNS amplification attacks, and NetBIOS enumeration.
* **DNS cache poisoning attacks:** DNS cache poisoning attacks involve corrupting or poisoning the DNS cache of a DNS resolver with malicious DNS records. This can lead to users being redirected to malicious websites or services, resulting in information theft, phishing, or other malicious activities.
* **Attacks and exploits against Server Message Block (SMB) implementations:** SMB is a network file-sharing protocol commonly used in Windows environments. Attacks against SMB implementations involve exploiting vulnerabilities in the protocol to gain unauthorized access to file shares, execute remote commands, or launch denial-of-service attacks.
* **Simple Network Management Protocol (SNMP) vulnerabilities and exploits:** SNMP is a protocol used for network management and monitoring. SNMP vulnerabilities and exploits involve attacks targeting weaknesses in SNMP implementations, allowing attackers to gain unauthorized access to network devices, retrieve sensitive information, or disrupt network operations.
* **Simple Mail Transfer Protocol (SMTP) vulnerabilities and exploits:** SMTP is the standard protocol used for sending and receiving email messages. SMTP vulnerabilities and exploits involve attacks targeting weaknesses in SMTP servers or clients, allowing attackers to send spam, intercept email messages, or perform other malicious activities.
* **File Transfer Protocol (FTP) vulnerabilities and exploits:** FTP is a protocol used for transferring files between a client and a server on a computer network. FTP vulnerabilities and exploits involve attacks targeting weaknesses in FTP servers or clients, allowing attackers to gain unauthorized access to files, upload malware, or execute arbitrary commands on FTP servers.
* **Pass-the-hash attacks:** Pass-the-hash attacks involve stealing hashed user credentials from a compromised system and using them to authenticate and access other systems on the network without needing to crack the passwords. These attacks exploit weaknesses in authentication protocols like NTLM (NT LAN Manager) used in Windows environments.
* **On-path attacks (previously known as man-in-the-middle [MITM] attacks):** On-path attacks involve intercepting and manipulating communication between two parties without their knowledge. Attackers position themselves between the communicating parties, allowing them to eavesdrop, tamper with, or inject malicious content into the communication.
* **SSL stripping attacks:** SSL stripping attacks involve downgrading secure HTTPS connections to unencrypted HTTP connections, allowing attackers to intercept and view sensitive information exchanged between a client and a server. Attackers exploit vulnerabilities in web applications or network configurations to perform these attacks.
* **Denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks:** DoS and DDoS attacks involve flooding a target system or network with an overwhelming amount of traffic or requests, causing it to become unavailable to legitimate users. DoS attacks are launched from a single source, while DDoS attacks involve multiple sources coordinated to amplify the attack.
* **Network access control (NAC) bypass:** Network access control (NAC) systems are designed to enforce security policies and control access to network resources. NAC bypass attacks involve exploiting weaknesses in NAC implementations to circumvent access controls and gain unauthorized access to network resources.
* **Virtual local area network (VLAN) hopping attacks:** VLAN hopping attacks involve exploiting vulnerabilities in network switches to gain unauthorized access to traffic on different VLANs. Attackers exploit weaknesses in switch configurations or protocols like IEEE 802.1Q to bypass VLAN isolation and intercept sensitive information.

The following sections cover one particular On-Path attack in detail using ARP Spoofing. In an on-path attack (previously known as a man-in-the-middle [MITM] attack), an attacker places himself or herself in-line between two devices or individuals that are communicating in order to eavesdrop (that is, steal sensitive data) or manipulate the data being transferred (such as by performing data corruption or data modification). On-path attacks can happen at Layer 2 or Layer 3. Below figure illustrates an on-path attack.

A diagram of a computer network

Description automatically generated

****ARP Spoofing and ARP Cache Poisoning****

**ARP cache poisoning** (also known as ARP spoofing) is an example of an attack that leads to an on-path attack scenario. An ARP spoofing attack can target hosts, switches, and routers connected to a Layer 2 network by poisoning the ARP caches of systems connected to the subnet and intercepting traffic intended for other hosts on the subnet. In the above figure, the attacker spoofs Layer 2 MAC addresses to make the victim believe that the Layer 2 address of the attacker is the Layer 2 address of its default gateway (10.2.3.4). The packets that are supposed to go to the default gateway are forwarded by the switch to the Layer 2 address of the attacker on the same network. The attacker can forward the IP packets to the correct destination in order to allow the client to access the web server (10.2.66.77).

**Media Access Control (MAC) spoofing** is an attack in which a threat actor impersonates the MAC address of another device (typically an infrastructure device such as a router). The MAC address is typically a hard-coded address on a network interface controller. In virtual environments, the MAC address could be a virtual address (that is, not assigned to a physical adapter). An attacker could spoof the MAC address of physical or virtual systems to either circumvent access control measures or perform an on-path attack.

**Note:** A common mitigation for ARP cache poisoning attacks is to use Dynamic Address Resolution Protocol (ARP) Inspection (DAI) on switches to prevent spoofing of the Layer 2 addresses.

Another example of a Layer 2 on-path attack involves placing a switch in the network and manipulating Spanning Tree Protocol (STP) to make it the root switch. This type of attack can allow an attacker to see any traffic that needs to be sent through the root switch.

An attacker can carry out an on-path attack at Layer 3 by placing a rogue router on the network and then tricking the other routers into believing that this new router has a better path than other routers. It is also possible to perform an on-path attack by compromising the victim’s system and installing malware that can intercept the packets sent by the victim. The malware can capture packets before they are encrypted if the victim is using SSL/TLS/HTTPS or any other mechanism. An attack tool called SSLStrip uses on-path functionality to transparently look at HTTPS traffic, hijack it, and return non-encrypted HTTP links to the user in response. This tool was created by a security researcher called Moxie Marlinspike. You can download the tool from <https://github.com/moxie0/sslstrip>.

The following are some additional Layer 2 security best practices for securing your infrastructure:

* Select an unused VLAN (other than VLAN 1) and use it as the native VLAN for all your trunks. Do not use this native VLAN for any of your enabled access ports. Avoid using VLAN 1 anywhere because it is the default.
* Administratively configure switch ports as access ports so that users cannot negotiate a trunk; also disable the negotiation of trunking (that is, do not allow Dynamic Trunking Protocol [DTP]).
* Limit the number of MAC addresses learned on a given port by using the port security feature.
* Control Spanning Tree to stop users or unknown devices from manipulating it. You can do so by using the BPDU Guard and Root Guard features.
* Turn off Cisco Discovery Protocol (CDP) on ports facing untrusted or unknown networks that do not require CDP for anything positive. (CDP operates at Layer 2 and might provide attackers information you would rather not disclose.)
* On a new switch, shut down all ports and assign them to a VLAN that is not used for anything other than a parking lot. Then bring up the ports and assign correct VLANs as the ports are allocated and needed.
* Use Root Guard to control which ports are not allowed to become root ports to remote switches.
* Use DAI.
* Use IP Source Guard to prevent spoofing of Layer 3 information by hosts.
* Implement 802.1X when possible to authenticate and authorize users before allowing them to communicate to the rest of the network.
* Use Dynamic Host Configuration Protocol (DHCP) snooping to prevent rogue DHCP servers from impacting the network.
* Use storm control to limit the amount of broadcast or multicast traffic flowing through a switch. An attacker could perform a **packet storm** (or broadcast storm) attack to cause a DoS condition. The attacker does this by sending excessive transmissions of IP packets (often broadcast traffic) in a network.
* Deploy access control lists (ACLs), such as Layer 3 and Layer 2 ACLs, for traffic control and policy enforcement.

## Part 1: On-Path Attacks with Ettercap

On-path attacks are very powerful ways to steal data that is travelling on a network. Without end-to-end encryption, as with much data travelling on local LANs, it is easy to capture clear text information, and even complete files, using on-path attack methods.

**Note:** On-path is replacing man-in-the-middle (MITM) as the name of this type of attack. The replacement process is incomplete; however, so you may still see MITM in many places, including some certification exam questions. Just be aware that the two terms are currently interchangeable.

Ettercap is used to perform on-path (MITM) attacks. The goal of an on-path attack is to intercept traffic between devices to obtain information that can be used to impersonate the target or to alter data being transmitted. The attacker is situated” between” two communicating hosts. In on-path attacks, the hacker doesn’t need to compromise the target device, but can just sniff traffic passing back and forth between the target and destination. Ettercap is used as an on-path tool, and the attack machine is on the same IP network as the victim.

Objectives

In this lab, you will complete the following objectives:

* Launch Ettercap and Explore Its Capabilities
* Perform the On-Path (MITM) Attack
* Use Wireshark to observe the ARP Spoofing

Required Resources

* Kali VM customized for Ethical Hacker course

Instructions

Step 1.1: Set up an ARP spoofing attack.

In this attack, you will use ARP spoofing to redirect traffic on the local virtual network to your Kali Linux system at 10.6.6.1. ARP spoofing is often used to impersonate the default gateway router to capture all traffic entering or leaving the local IP network. Because your lab environment uses an internal virtual network, instead of spoofing the default gateway, you will use ARP spoofing to redirect traffic that is destined for a local server with the address 10.6.6.13.

1. Load Kali Linux using the username **kali** and the password **kali**. Open a terminal session from the menu bar at the top of the screen.
2. A screenshot of a computer

   Description automatically generatedWe need first to change the prompt to include the date and time, and your student ID. Run the command: PS1=”\d \@ [Your\_Student-ID]”. Use your own student ID. The prompt will change as shown below.
3. The target host in this lab is the Linux device at 10.6.6.23. To view the network from the target perspective, and initiate traffic between the target and the server, use SSH to log in to this host. The username is **labuser** and the password is **Cisco123**.

The user of the 10.6.6.23 host is communicating with the server at 10.6.6.13. The on-path attacker at 10.6.6.1 (your Kali VM) will intercept and relay traffic between these hosts.

**Note**: The password will not display on the screen.

┌──(kali㉿Kali)-[~]

└─# **ssh -l labuser 10.6.6.23**

If you get the following message, answer **yes** to continue.

The authenticity of host '10.6.6.23 (10.6.6.23)' can't be established.

ED25519 key fingerprint is SHA256:u3Yjj1imvIGFFU6uLfJlAyM+BC1AXhLyO45oPedjNk8.

This key is not known by any other names.

Are you sure you want to continue connecting (yes/no/[fingerprint])? **Yes**

Warning: Permanently added '10.6.6.23' (ED25519) to the list of known hosts.

labuser@10.6.6.23’s password: **Cisco123**

1. Because you are creating an on-path attack that uses ARP spoofing, you will be monitoring the ARP mappings on the victim host. The attack will cause changes to those mappings.

Use the command **ip neighbor** to view the current ARP cache on the target computer. **Note**: The hostname 3fb0515ea2f7 maybe different for your Kali VM environment.

labuser@3fb0515ea2f7:/$ **ip neighbor**

10.6.6.1 dev eth0 llanddr 02:42:17:81:d2:45 REACHABLE (output may vary)

**Note**: If you are using the ARM CPUs (Apple M1/M2) version of the VM, you will need to switch to use the root user with the password **Cisco123** and use the command **arp -a** in place of **ip neighbor** to view the current ARP cache throughout this activity.

labuser@gravemind:/$ **su -**

Password: **Cisco123**

root@gravemind:/$ **arp -a**

? (10.6.6.1) at 02:42:17:d5:bb:2b:ab [ether] on eth0

How many entries are there in the current ARP cache?

**←1**

Answer below.

What is the MAC address of the Kali attacker machine?

**←2**

Answer below.

Step 1.2: Load Ettercap to begin scanning and Explore its Capabilities.

1. Open a new terminal session from the menu bar in Kali Linux. Do not close the SSH-terminal that is running the session with 10.6.6.23.
2. Use the **ettercap -h** command to view the help file for the Ettercap application.

┌──(kali㉿kali)-[~]

└─# **ettercap -h**

1. Examine the help file content.

How many user interfaces are available for the Ettercap tool? What are the options used to specify the user interfaces?

What is the MAC address of the Kali attacker machine?

**←3**

Answer below.

1. In this part, you will use a GUI interface to access Ettercap. Start Ettercap GTK+ graphical user interface using the **ettercap -G** command. Most Ettercap functions require root permissions, so use the **sudo** command to obtain the required permissions.

┌──(kali㉿kali)-[~]

└─# **sudo ettercap -G**

1. The Ettercap GUI opens in a new window. You are sniffing traffic on an internal, virtual network. The default setup is to scan using interface eth0. Change the sniffing interface to **br-internal**, which is the interface that is configured on the 10.6.6.0/24 virtual network, by changing the value in the **Setup > Primary** **Interface** dropdown.
2. Click the **checkbox** icon at the top right of the Ettercap screen to continue. A message appears at the bottom of the screen indicating that Unified sniffing has started.

Step 1.3: Perform the On-Path (MITM) Attack: Select the Target Devices.

1. In the Ettercap GUI window, open the Hosts List window by clicking the Ettercap menu (three dots icon). Select the **Hosts** entry and then **Hosts List**. Click the **Scan for Hosts** icon (magnifying glass) at top left in the menu bar. A list of the hosts that were discovered on the 10.6.6.0/24 network appears in the Host List window.

How many hosts were discovered?

**←4**

Answer below.

At least one of the MAC addresses should be familiar.

1. Define the source and destination devices for the attack. To do so, click the IP address **10.6.6.23** in the window to highlight the target user host. Click the **Add to Target 1** button at the bottom of the Host List window. This defines the user’s host as Target 1.
2. Click the IP address of the destination web server at **10.6.6.13** to highlight the line. Click the **Add to Target 2** button at the bottom of the host window.

Any IP/MAC address specified as a Target 1 will have all its traffic diverted through the attacking computer that is running Ettercap. In this lab, the attacking computer is the Kali Linux machine at 10.6.6.1. All other computers on the subnet, other than the targets, will communicate normally.

1. Click the MITM icon on the menu bar (the first circular icon on top right). Select **ARP Poisoning…** from the dropdown menu. Verify that **Sniff remote connections** is selected. Click **OK**.
2. The MITM exploit is started. If sniffing does not start immediately, click the **Start** option (play button) at left in the top menu.

Step 1.4: Perform the ARP spoofing attack.

1. Return to the terminal window that is running the SSH session with the target user host at 10.6.6.23. Repeat the ping to 10.6.6.13

labuser@3fb0515ea2f7:/$ **ping -c 5 10.6.6.13**

1. Use the **ip neighbor** command to view the ARP table on 10.6.6.23 again. Note the MAC address listed for 10.6.6.13.

Is the MAC address associated with 10.6.6.13 the same as the one you recorded in Step 1.1.d?

**←5**

Answer below.

What is strange about this?

**←6**

Answer below.

What is the effect of this change?

**←7**

Answer below.

1. Close the Ettercap graphical user interface. Leave the SSH connection to 10.6.6.23 active.

Step 1.5: Select the Target Devices and Perform the MITM attack using the CLI

In this step, you will use the command line interface in Ettercap to perform ARP spoofing and write a .pcap file that can be opened in Wireshark. Refer to the help information for Ettercap to interpret the options used in the commands.

1. Return to the terminal session that is connected via SSH to 10.6.6.23. Ping the IP addresses 10.6.6.11 and 10.6.6.13. 10.6.6.11 is another host on the LAN that we will verify is unaffected by the attack. Then, use the **ip neighbor** command to find the MAC addresses associated with the IP addresses of the two systems.

labuser@3fb0515ea2f7:/$ **ping -c 5 10.6.6.11**

labuser@3fb0515ea2f7:/$ **ping -c 5 10.6.6.13**

labuser@3fb0515ea2f7:/$ **ip neighbor**

Complete the table below:

**←8**

| **IP Address** | **MAC Address** | **Role** |
| --- | --- | --- |
| 10.6.6.1 |  |  |
| 10.6.6.11 |  |  |
| 10.6.6.13 |  |  |
| 10.6.6.23 |  |  |

**Note**: To find the MAC of 10.6.6.23, go to the SSH session terminal and enter the **ip address** command. Determine the MAC address of the interface that is addressed on the 10.6.6.0/24 network.

1. The **ettercap -T** command runs Ettercap in text mode, instead of using the GUI interface. The syntax to start Ettercap and specify the targets is: **sudo ettercap -T [options] -q -i [interface] --write [file name] -- mitm arp /[target 1]// /[target 2]//**.

Open a new terminal window as necessary.

Use the man page for Ettercap and complete the table below:

**←9**

| **Options and Values** | **Meaning** |
| --- | --- |
| -T |  |
| -q |  |
| -i |  |
| --write |  |
| --mitm arp |  |
| /target1// |  |
| /target2// |  |

1. In a terminal window, enter the command as follows to save the pcap file in the current working directory:

┌──(kali@kali)-[~]

└─$**sudo** **ettercap -T -q -i br-internal --write mitm-saved.pcap --mitm arp /10.6.6.23// /10.6.6.13//**

When Ettercap starts, you will receive output similar to that shown:

ettercap 0.8.3.1 copyright 2001-2020 Ettercap Development Team

Listening on:

br-internal -> 02:42:14:BB:18:BD

10.6.6.1/255.255.255.0

fe80::42:14ff:febb:18bd/64

SSL dissection needs a valid 'redir\_command\_on' script in the etter.conf file

Privileges dropped to EUID 65534 EGID 65534...

34 plugins

42 protocol dissectors

57 ports monitored

28230 mac vendor fingerprint

1766 tcp OS fingerprint

2182 known services

Lua: no scripts were specified, not starting up!

Scanning for merged targets (2 hosts)...

\* |==================================================>| 100.00 %

2 hosts added to the hosts list...

ARP poisoning victims:

GROUP 1 : 10.6.6.23 02:42:0A:06:06:17

GROUP 2 : 10.6.6.11 02:42:0A:06:06:0B

Starting Unified sniffing...

Text only Interface activated...

Hit 'h' for inline help

1. Return to the SSH terminal session to 10.6.6.23. Ping the two IP addresses, 10.6.6.11 and 10.6.6.13, again. Use the **ip neighbor** command to view the associated MAC addresses.

Are the MAC addresses that are associated with the IP addresses the same as you recorded in substep a?

**←10**

Answer below.

1. Close the SSH terminal session that is connected to 10.6.6.23 and return to the terminal session running Ettercap in text mode. Enter **q** to quit Ettercap.

Step 1.6: Use Wireshark to Observe the ARP Spoofing Attack.

In this step, you will examine the .pcap file that Ettercap created.

1. Review the MAC addresses that you recorded in Step 1c. The MAC address for 10.6.6.23 can be found in the output of the Ettercap text interface in Target Group 1.

What is now true of the MAC addresses of these three systems?

**←11**

Answer below.

1. In the Kali terminal window, start Wireshark with the **mitm-saved.pcap** file that you created with Ettercap.

┌──(kali@kali)-[~]

└─$ **wireshark mitm-saved.pcap**

1. The Ettercap attack computer first broadcasts ARP requests to obtain the actual MAC addresses for the two target hosts, 10.6.6.23 and 10.6.6.11. The attacking machine then begins to send ARP responses to both target hosts using its own MAC for both IP addresses. This causes the two target hosts to address the Ethernet frames to the attacker’s computer, which enables it to collect data as an on-path attacker.

Why must the computer executing the Ettercap attack be located on the same IP network as the target system?

**←12**

Answer below.

## Part 2: Self-Learning (Exploration of Additional On-Path Attack Techniques)

In Part 1, we explored an example of an on-path attack utilizing ARP spoofing. For Part 2, your task is to investigate and execute a different type of on-path attack of your choice. Your assignment includes the following steps:

* + - Provide a detailed description of the scenario and network topology involved.
    - Explain the methodology employed to carry out the attack, including the tools and commands utilized.
    - Screenshots included must show the Student-ID in the prompt using the command: **PS1=”\d \@ [Your\_Student-ID]”**. Use your own student ID. The prompt will change as shown below

A screenshot of a computer

Description automatically generated

Grading Criteria and Rules

1. This Practical Lab2B counts for 2.5%.
2. This assignment is an individual work.
3. Deadline for submission is Sunday 5th May 2024 @ 23:59.
   * Any late submission will be penalized (-0.25/ day).
   * Any submission via email or MS Teams will NOT be accepted.
4. **Deliverables:** You should upload your work via Blackboard on time before submission Deadline.
   * Word file with all answers and screenshots showing that you did the practical Lab. You are required to insert all commands and outputs (screenshots) in the document provided, including the answers to all questions. Use references where appropriate.
5. **Evaluation Rubric:**

Your work will be assessed based on the following criteria:

|  |  |  |
| --- | --- | --- |
| **Step** | | **Max** |
| Part 1- On-Path Attacks with Ettercap | | 1.25 |
| Part 2- Self-Learning | | 1.25 |
| **Total marks:** | **2.5 marks** | |