Network Security - Assignment 4

Securing your Ubuntu server and exploring the network environment is crucial to identifying potential vulnerabilities and unauthorized access points. By executing these steps, you can gain a comprehensive understanding of your server's network environment and identify potential security issues. This knowledge is crucial for implementing effective security measures and maintaining a secure network. Run the following commands in your virtual machine that you set up last module. Take screenshots of your output as you complete each task. Describe anything you find interesting or useful in each output.

You will then set up a GitHub repo that will be your "Network Security Home Lab" to add to your portfolio when applying for jobs. Each assignment pertaining to your VM will be a separate "chapter".

GitHub Directions:

Most of you should have Git on your computer and a GitHub account. If not, Module 4 has a PDF going over how to set that up. Any issues, let me know.

1. In your computer, create a folder for all of your labs (network_security)
2. Open up Visual Studio Code (or use another text editor of your choice) and open the folder.
3. Create a README.md file which will include a brief explanation of what this repo is for (You can use mine as an example).
4. Create an md file for this assignment. You can create it in your text editor. I personally use Obsidian to create all of my assignments (including this one!!), pdfs, handouts, etc. and then just paste it into my README depending on what makes sense (Obsidian is nice because you can have all of your markdown files in one localized "Vault").
5. Create another md file for this lab/assignment. DON'T name it "Assignment 4"! Make it interesting for employers/other users. You can name it like "Exploring Ubuntu Home Lab", or something else that grabs people's attention.
6. Complete the lab below. For each tool you use, explain briefly what each tool does and take screenshots. You can do the basics, or be as complex as you would like. If you want a job in cybersecurity, I suggest being as detailed as possible. DO NOT copy paste info from chatGPT or another language model; people in the industry are not dumb and will be able to tell. Use it as a resource, but use your own words! Show your passion and expertise.
7. In your original README, add this assignment as a "chapter". See my repo as an example: https://github.com/kaitlinchoffmann/cybersecurity_home_lab. In markdown, spaces need %20 see below:

8. **YOUR SUBMISSION**: Submit the link to your GitHub repo.

Exploring Ubuntu Home Lab

NOTE: You will have to install some of the tools below via the command line. To install a package in ubuntu, we can use Advanced Package Tools (APT). Use the following syntax to install a package: sudo install apt *packagename*

1. **Identify Network Interfaces and IP Addresses**

- **Command:**

```
ip a or ifconfig
```

- **Purpose:** This command displays all network interfaces and their associated IP addresses on your server. Knowing which interfaces are active and their IP addresses helps you understand your server's network configuration.
- **Tool Explanation:** ip a and ifconfig are utilities that provide detailed information about network interfaces, including their status (up or down), IP addresses, and more.
- **NOTE:** You may have to install net-tools in order to run ifconfig. To do so, run the command: ``sudo apt install net-tools

2. **Check Open Ports**

- **Command:**

```
sudo netstat -tuln or ss -tuln
```

- **Purpose:** Lists all open ports on the server along with the services listening on them. This helps you identify unnecessary open ports that could be potential entry points for attackers.
- **Tool Explanation:** netstat and ss show network connections, routing tables, interface statistics, masquerade connections, and multicast memberships. The -tuln options restrict the output to show only TCP (t) and UDP (u) ports in listening (l) state without resolving names (n).

3. **Analyze Network Connections**

- **Command:**

```
sudo lsof -i -P -n
```

- **Purpose:** Lists all open network connections, which can help you identify unexpected or unauthorized connections to your server.
- **Tool Explanation:** lsof stands for 'list open files'. With the -i flag, it lists all network files, including their associated processes. The -P and -n flags prevent the resolution of port numbers and IP addresses, making the output easier to read and faster to generate.

4. **Perform Network Scanning with Nmap**

- **Command:**

```
sudo nmap -sS -O localhost
```

- **Purpose:** Scans your server to identify open ports, running services, and the operating system. This can help you discover services that are unintentionally exposed.
- **Tool Explanation:** Nmap (Network Mapper) is a powerful network scanning tool used to discover hosts and services on a network. The -sS option performs a stealth TCP SYN scan, and -O attempts to determine the operating system of the target.
- **NOTE:** You will have to install Nmap. To do so, run: sudo apt install nmap Nmap can be a little slow on the VM, so some of the commands may take a bit to complete. Be patient!

5. **Check for Open Ports on the Server's Network**

- **Command:**

```
sudo nmap -sP 192.168.1.0/24
```

- **Purpose:** Identifies all live hosts on your local network. This helps you understand the devices present in your network and ensures there are no unauthorized devices connected.

- **Tool Explanation:** The -sP option in Nmap is a Ping Scan, which discovers which hosts on a network are up without performing a port scan.

6. **Check for Services and Versions**

- **Command:**

```
sudo nmap -sV localhost
```

- **Purpose:** Scans for open ports and attempts to determine the service and version running on each port. This helps identify outdated or vulnerable software that might need updating.
- **Tool Explanation:** The -sV option in Nmap enables version detection, providing detailed information about the services running on open ports.

7. **Identify Potential Vulnerabilities**

- **Command:**

```
sudo nmap --script vuln localhost
```

- **Purpose:** Uses Nmap's vulnerability scanning scripts to identify known vulnerabilities on the server. This step is useful for finding common security issues in installed software.
- **Tool Explanation:** Nmap has a scripting engine that allows for a wide range of scans. The --script vuln option runs scripts that check for various vulnerabilities.

8. **Inspect Network Traffic**

- **Command:**

```
sudo tcpdump -i eth0
```

- **Purpose:** Monitors network traffic on a specific interface (e.g., eth0). This is helpful to observe real-time traffic and detect suspicious activities or anomalies.
- **Tool Explanation:** tcpdump is a packet analyzer that captures and displays packet headers of network traffic passing through a specified interface.
- **NOTE:** To stop process, hit ctrl+c on your keyboard.

9. **Monitor Network Connections in Real-Time**

- **Command:**

```
sudo watch -n 1 netstat -tulnp
```

- **Purpose:** Continuously monitors network connections, updating every second (-n 1). This helps in real-time observation of network activities, such as new connections or services starting.
- **Tool Explanation:** watch runs a specified command at regular intervals. In this case, it runs netstat to keep you updated about network connections in real time.
- **NOTE:** To stop process, hit ctrl+c on your keyboard.

10. **Check Firewall Rules**

- **Command:**

```
sudo ufw status verbose
```

- **Purpose:** Displays the current firewall rules configured on your server, showing which ports and services are allowed or blocked. This helps ensure that only necessary ports are open.
- **Tool Explanation:** ufw (Uncomplicated Firewall) is a front-end for managing iptables, designed to make it easier to configure a firewall. The status verbose option provides a detailed view of the current firewall configuration.
- **NOTE:** You don’t have a firewall set up yet, but we will fix that next module.