

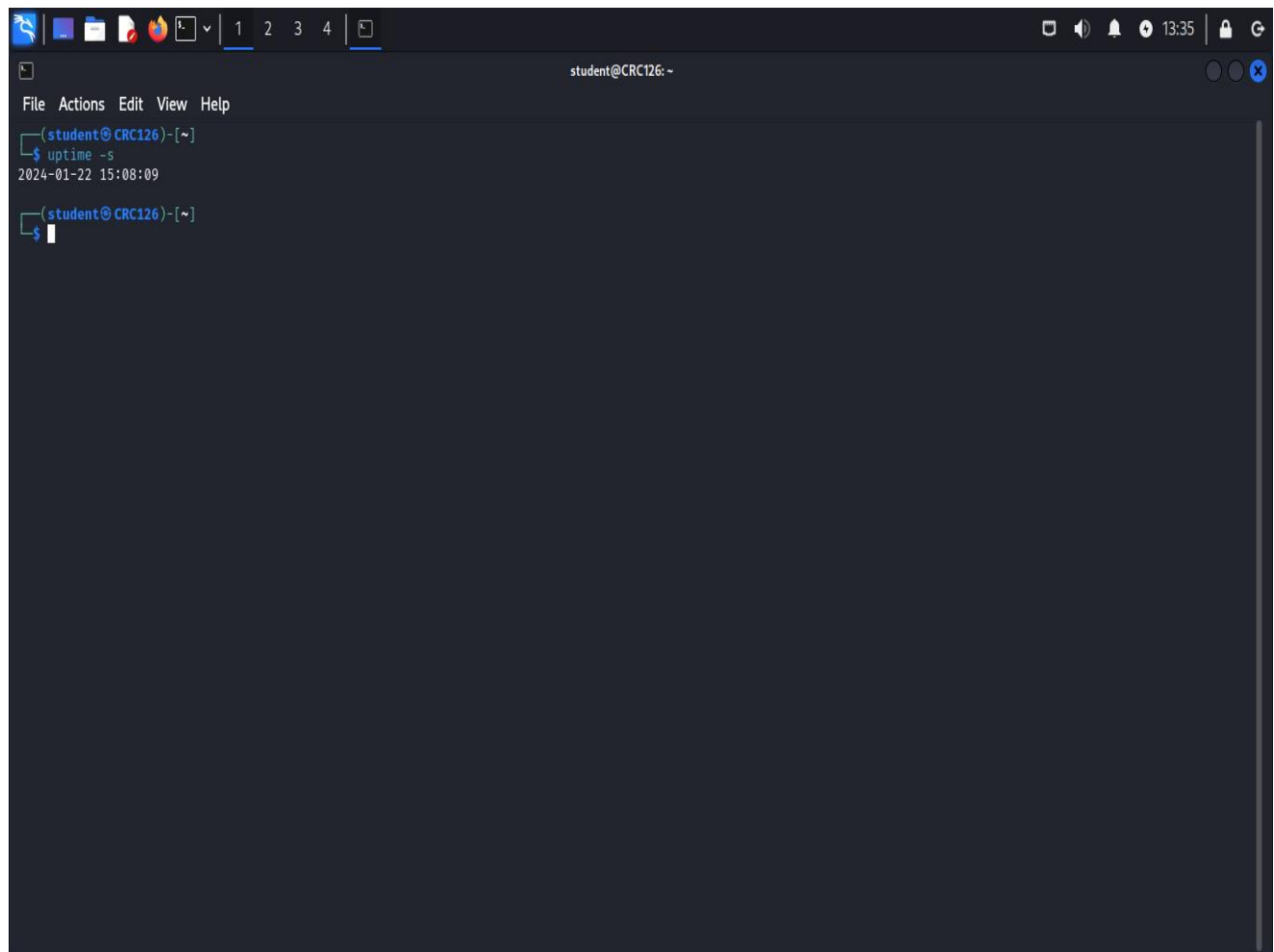
Lab 1 – Getting Acquainted with Kali Linux

Part 1: -

Q1. When your VM was last booted?

A: - Command: uptime -s

Explanation: This command tells when the system was last booted up.



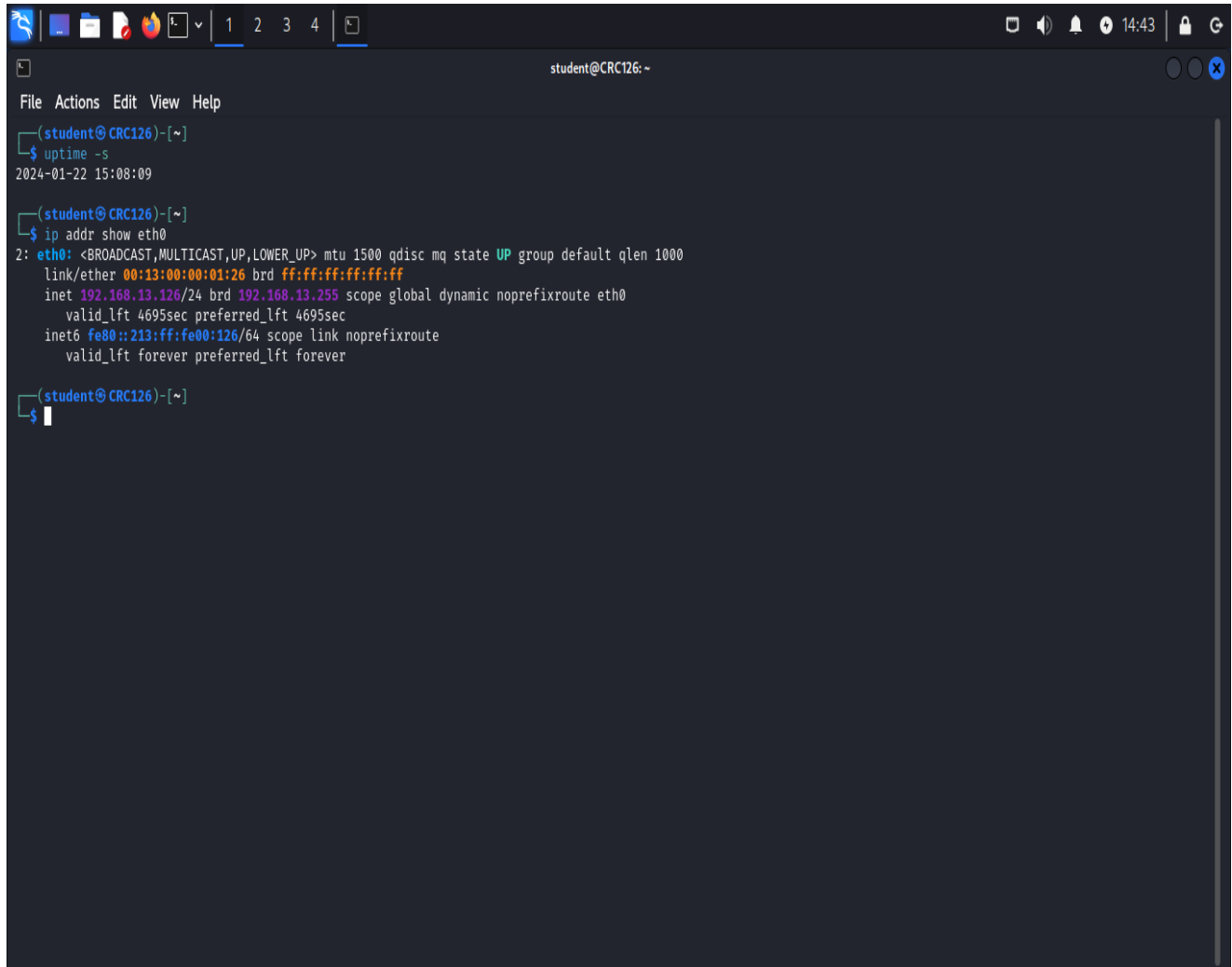
The screenshot shows a terminal window titled "student@CRC126: ~". The terminal has a menu bar with "File", "Actions", "Edit", "View", and "Help". The prompt is "(student@CRC126)-[~]". The user has entered the command "uptime -s", and the output is "2024-01-22 15:08:09". The prompt is now "(student@CRC126)-[~]" with a cursor on the next line.

```
(student@CRC126)-[~]  
$ uptime -s  
2024-01-22 15:08:09  
(student@CRC126)-[~]  
$
```

Q2. What is your VM's local (eth0) IP address?

A: - Command: ip addr show eth0

Explanation: This command displays the details of the eth0 network interface.



```
(student@CRC126)-[~]
$ uptime -s
2024-01-22 15:08:09

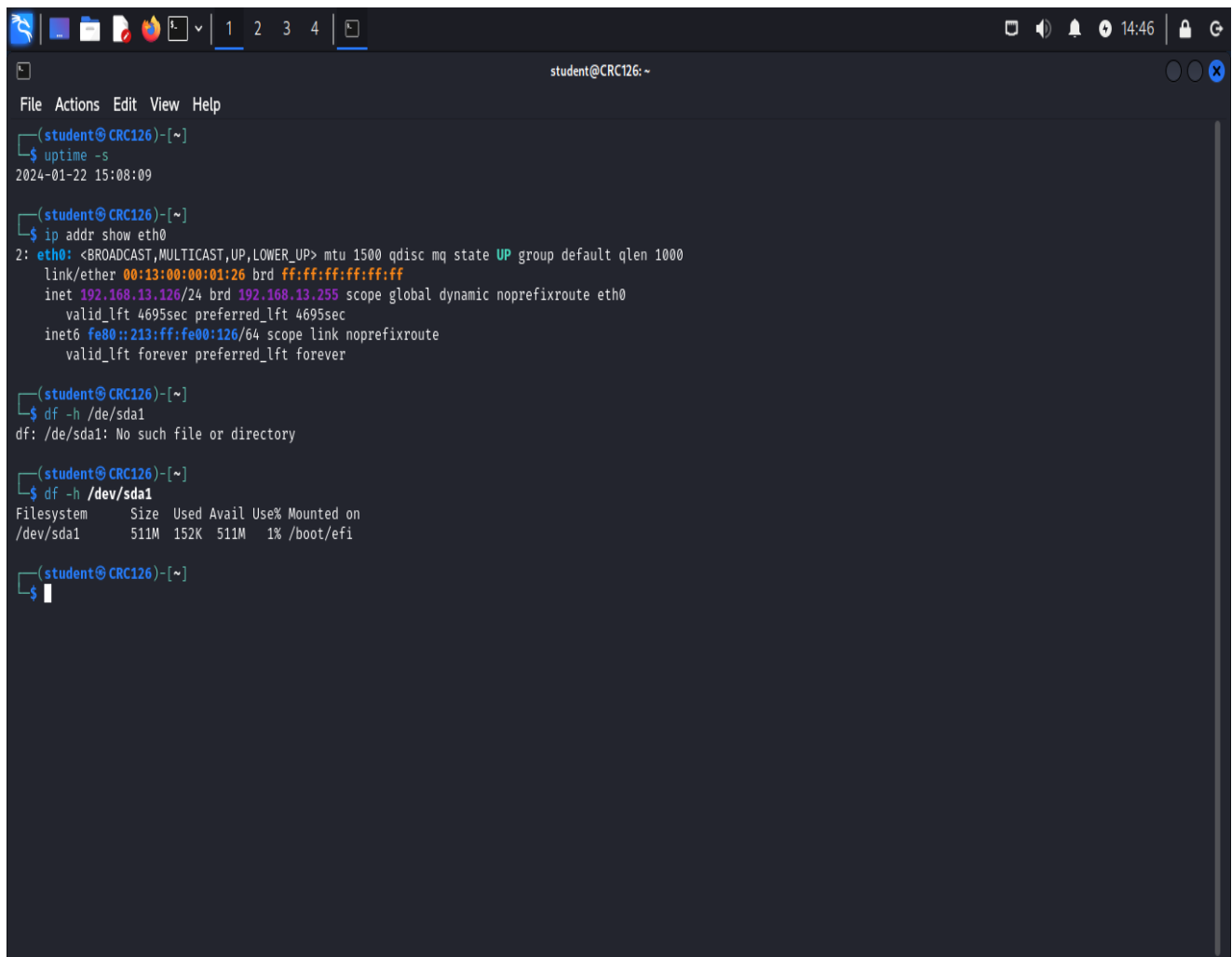
(student@CRC126)-[~]
$ ip addr show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:13:00:00:01:26 brd ff:ff:ff:ff:ff:ff
    inet 192.168.13.126/24 brd 192.168.13.255 scope global dynamic noprefixroute eth0
        valid_lft 4695sec preferred_lft 4695sec
    inet6 fe80::213:ff:fe00:126/64 scope link noprefixroute
        valid_lft forever preferred_lft forever

(student@CRC126)-[~]
$
```

Q3. How much hard disk space (on sda1) is available in your VM?

A: - Command: `df -h /dev/sda1`

Explanation: This command shows the disk space usage of `/dev/sda1` in a human-readable format, including the available space.

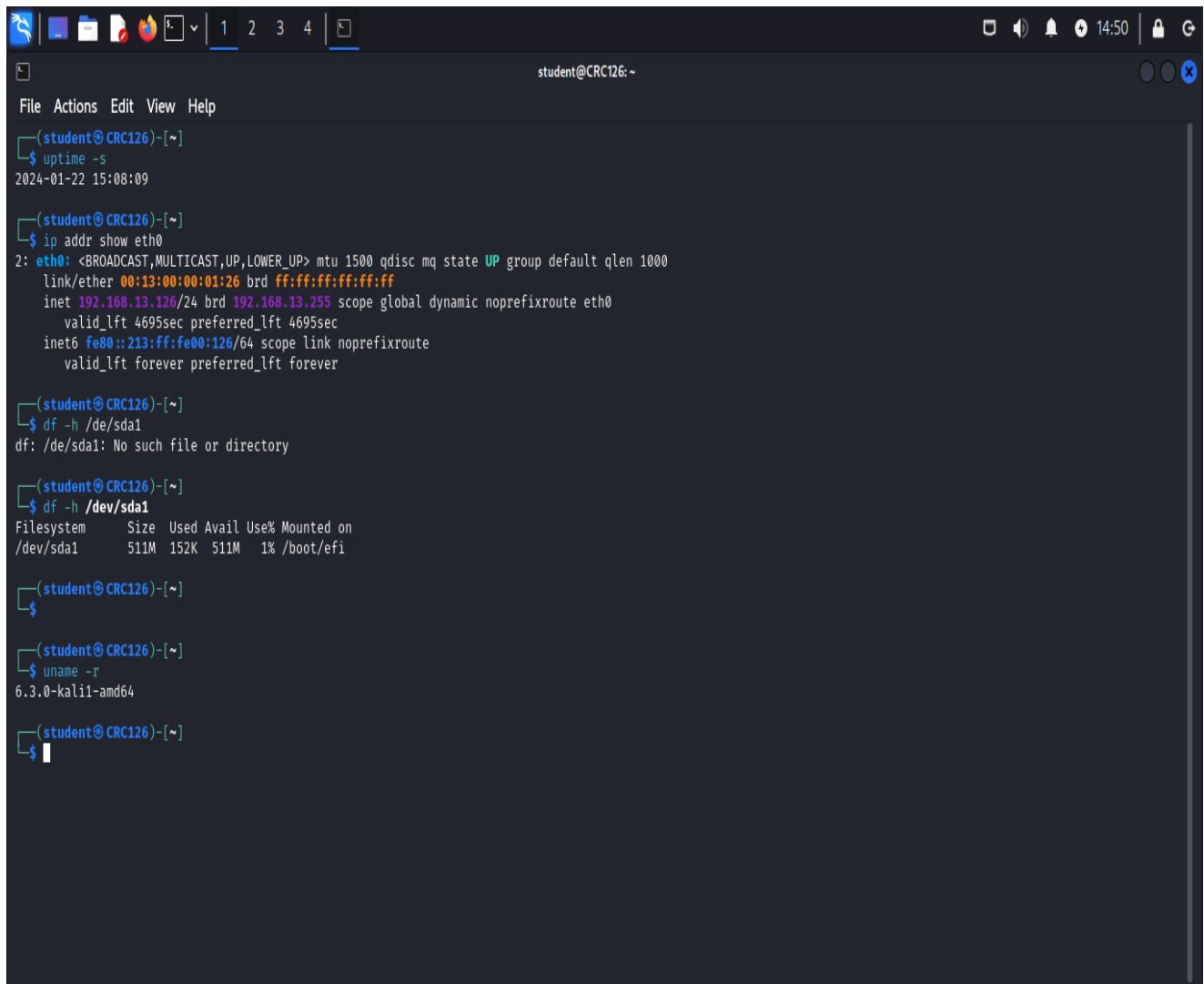


```
student@CRC126: ~  
File Actions Edit View Help  
(student@CRC126)-[~]  
$ uptime -s  
2024-01-22 15:08:09  
(student@CRC126)-[~]  
$ ip addr show eth0  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000  
    link/ether 00:13:00:00:01:26 brd ff:ff:ff:ff:ff:ff  
    inet 192.168.13.126/24 brd 192.168.13.255 scope global dynamic noprefixroute eth0  
        valid_lft 4695sec preferred_lft 4695sec  
    inet6 fe80::213:ff:fe00:126/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever  
(student@CRC126)-[~]  
$ df -h /de/sda1  
df: /de/sda1: No such file or directory  
(student@CRC126)-[~]  
$ df -h /dev/sda1  
Filesystem      Size  Used Avail Use% Mounted on  
/dev/sda1       511M  152K  511M   1% /boot/efi  
(student@CRC126)-[~]  
$
```

Q4. Which version of Linux kernel is your Kali Linux VM using?

A: - Command: `uname -r`

Explanation: This command displays the kernel release version of your Linux system.

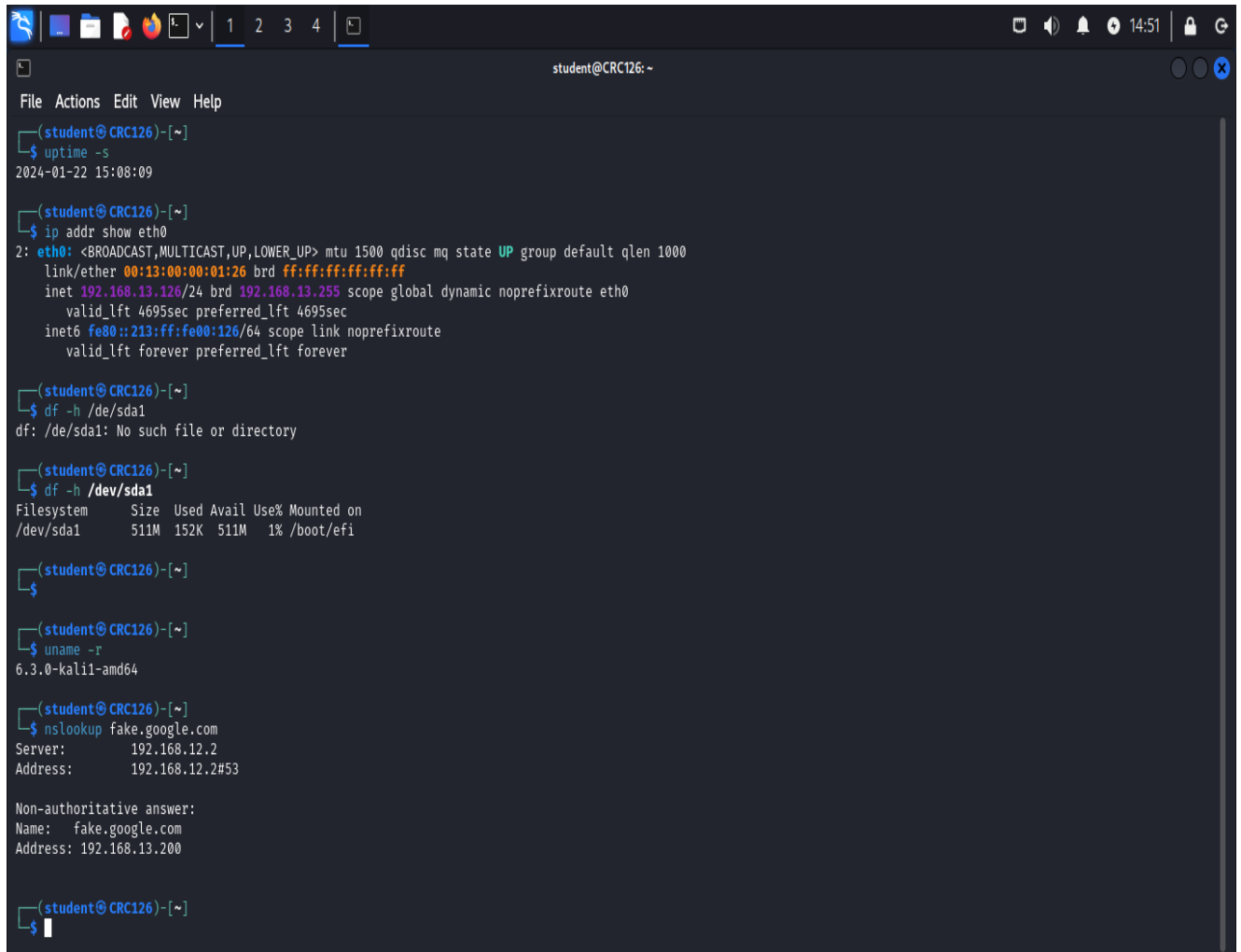


```
student@CRC126: ~  
File Actions Edit View Help  
(student@CRC126)-[~]  
$ uptime -s  
2024-01-22 15:08:09  
(student@CRC126)-[~]  
$ ip addr show eth0  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000  
    link/ether 00:13:00:00:01:26 brd ff:ff:ff:ff:ff:ff  
    inet 192.168.13.126/24 brd 192.168.13.255 scope global dynamic noprefixroute eth0  
        valid_lft 4695sec preferred_lft 4695sec  
    inet6 fe80::213:ff:fe00:126/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever  
(student@CRC126)-[~]  
$ df -h /de/sda1  
df: /de/sda1: No such file or directory  
(student@CRC126)-[~]  
$ df -h /dev/sda1  
Filesystem      Size  Used Avail Use% Mounted on  
/dev/sda1       511M  152K  511M   1% /boot/efi  
(student@CRC126)-[~]  
$  
(student@CRC126)-[~]  
$ uname -r  
6.3.0-kali1-amd64  
(student@CRC126)-[~]  
$
```

Q5. What is the IP address for the domain fake.google.com?

A: - Command: nslookup fake.google.com

Explanation: This command queries the DNS records for fake.google.com to find its associated IP address. Since fake.google.com only exists in cyber range and is a hypothetical domain, in a real scenario, this would return an error or no IP.



```
student@CRC126: ~  
File Actions Edit View Help  
(student@CRC126)-[~]  
$ uptime -s  
2024-01-22 15:08:09  
  
(student@CRC126)-[~]  
$ ip addr show eth0  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000  
    link/ether 00:13:00:00:01:26 brd ff:ff:ff:ff:ff:ff  
    inet 192.168.13.126/24 brd 192.168.13.255 scope global dynamic noprefixroute eth0  
        valid_lft 4695sec preferred_lft 4695sec  
    inet6 fe80::213:ff:fe00:126/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever  
  
(student@CRC126)-[~]  
$ df -h /de/sda1  
df: /de/sda1: No such file or directory  
  
(student@CRC126)-[~]  
$ df -h /dev/sda1  
Filesystem      Size  Used Avail Use% Mounted on  
/dev/sda1       511M  152K  511M   1% /boot/efi  
  
(student@CRC126)-[~]  
$  
  
(student@CRC126)-[~]  
$ uname -r  
6.3.0-kali1-amd64  
  
(student@CRC126)-[~]  
$ nslookup fake.google.com  
Server:         192.168.12.2  
Address:        192.168.12.2#53  
  
Non-authoritative answer:  
Name:   fake.google.com  
Address: 192.168.13.200  
  
(student@CRC126)-[~]  
$
```

Part 2: -

Q1. You temporarily got hold of a hard drive containing top-secret documents but protected by an unknown/encrypted filesystem. Your mission is to try and retrieve the documents in a covert fashion. What is the first Kali Linux tool that you will use on the hard drive and why?

A: - In a situation of having a hard drive with an unknown or encrypted filesystem, and where our mission is to retrieve documents covertly, the initial tool of choice from the Kali Linux arsenal would likely be cryptsetup. Here's why:

Cryptsetup: This tool is used for setting up encrypted filesystems using dm-crypt. It allows to manage the creation of LUKS (Linux Unified Key Setup) encrypted partitions, which is a standard for hard disk encryption. When facing an unknown or encrypted filesystem, the first step is often to identify and attempt to unlock the encryption, for which cryptsetup is ideally suited.

Why Cryptsetup? If the filesystem is encrypted with a common method like LUKS, cryptsetup can be used to open the encrypted volume by providing the correct passphrase or key file. Once the volume is unlocked, the filesystem can be mounted, and access to the files is gained. This approach is covert as it does not alter the encrypted volume (assuming you have the correct passphrase or key), and it allows for direct access to the files stored within.

In practice, we can start by attempting to identify the type of encryption used on the filesystem. If it's LUKS (a common choice for Linux systems), we can use `cryptsetup luksOpen <device> <name>` to try and open it. Success here depends on having the passphrase or a key file.

If the encryption type is unknown or not LUKS, or if cryptsetup is not immediately helpful, alternative steps might involve using forensic tools to analyze the disk for clues or trying to crack the encryption with tools like john (John the Ripper) if a passphrase is involved and can potentially be brute-forced or guessed based on available information.

However, the choice of cryptsetup as the first tool is based on a straightforward approach to dealing with known encryption schemes efficiently and covertly, provided we have some means of authentication.

Q2. You somehow gained access to a corporate network with Class-C private IP addressing. This corporation is known to be hosting a vulnerable web service (on port 443) only for users connected to the corporate network. Your mission is to exploit this vulnerability in order to download sensitive data from the web server, but first you must locate it within the network. What Kali Linux network scanning tool will you use to find the web server's IP address and why?

A: - For the task of locating a web server within a corporate network, particularly one that is known to be hosting a vulnerable web service on port 443, the ideal Kali Linux network scanning tool to use is Nmap (Network Mapper). Here's why Nmap is the preferred tool for this scenario:

Why Nmap?

Versatility: Nmap is an open-source tool for network exploration and security auditing. It is designed to discover hosts and services on a computer network, thereby building a "map" of the network. Nmap's versatility lies in its ability to detect open ports, identify running services and their versions, and even some vulnerabilities.

Port Scanning: Since the web service is on port 443, Nmap can be specifically instructed to scan for hosts with port 443 open. This is crucial because it directly targets the known port, making the scan more efficient and faster.

Stealth and Speed: Nmap offers various scanning techniques, including stealthy scans that can evade detection (to some extent) by intrusion detection systems. This makes it suitable for use in environments where discretion is necessary.

Scriptable: Nmap comes with the Nmap Scripting Engine (NSE), which allows for the execution of scripts for more advanced discovery and exploitation tasks. There are scripts specifically designed for discovering vulnerabilities, which could be beneficial once the web server is located.

Example Nmap Command

An example command to locate the web server on port 443 within a Class-C private network (assuming the network is 192.168.1.0/24) would be:

```
nmap -p 443 --open 192.168.1.0/24
```

This command does the following:

-p 443 specifies the port to scan for.

--open instructs Nmap to only show hosts with the specified port open.

192.168.1.0/24 defines the target network range, adjusting as necessary for the actual network you're scanning.

Nmap will then scan the entire subnet for devices with port 443 open, identifying potential targets for the next step of your mission.

Q3. Network scanning can trigger alerts on intrusion detection systems. An alternate method to Q2 for locating the vulnerable web server, is by passive network traffic monitoring. What Kali Linux tool can facilitate that and how?

A: - For passive network traffic monitoring, which is a method that can avoid triggering alerts on intrusion detection systems (IDS), the Kali Linux tool you would use is Wireshark. Wireshark is a powerful network protocol analyzer, enabling us to capture and interactively browse the traffic flowing across a network. It operates in a passive mode, meaning it only listens to the network traffic without sending packets or altering the flow in any way. This characteristic makes it ideal for use in environments where stealth is necessary.

Why Wireshark?

Deep Packet Inspection: Wireshark can analyze the packets in detail, including the protocols used and the payload of packets. This allows for the identification of the types of traffic on the network, including HTTP/HTTPS traffic to and from web servers.

Filtering Capabilities: Wireshark provides extensive filtering capabilities that allows to isolate traffic to and from specific ports, such as port 443 for secure web traffic. This can help in identifying servers that are actively being accessed over the network.

Graphical Interface: Wireshark's user-friendly graphical interface makes it easier to visualize the traffic and identify patterns or specific communications that could indicate the location and activity of the web server.

No Traffic Generation: Since Wireshark does not generate traffic but only captures existing traffic, it is less likely to be detected by network monitoring tools or IDS, making it a safer option for covertly monitoring network activities.

How to Use Wireshark for This Purpose

Capture Setup: Start Wireshark and set it up to capture traffic on the network interface that is connected to the corporate network.

Filter for HTTPS Traffic: Use a display filter such as `tcp.port == 443` to isolate HTTPS traffic, which is what we would expect from a web server communicating over port 443.

Analyze Traffic: Look for patterns or repeated communications to a specific IP address within the Class-C range of the corporate network. This could indicate the web server we are looking for.

Identify the Server: By examining the details of the captured packets, including the IP headers and any transmitted data, we can potentially identify the web server's IP address and any vulnerabilities it might be exposing through its traffic patterns or unencrypted data leaks.

Q4. You work with classified medical documents on your smartphone. You also suspect that your neighbor is a foreign spy, who may be trying to compromise your smartphone using Bluetooth. You still want to keep Bluetooth ON to use your Bluetooth headphones. Using which Kali Linux tool can you monitor if and when someone is trying to attack your smartphone via Bluetooth?

A: - To monitor potential Bluetooth-based attacks on your smartphone while still using our Bluetooth headphones, we can use the Kali Linux tool Wireshark, along with `hcitool` and `bluetoothctl` for Bluetooth device management and monitoring. However, for directly focusing on Bluetooth security and monitoring for attacks, `bettercap` stands out as a more specialized tool for this scenario.

Bettercap

Overview: Bettercap is a powerful, flexible, and portable tool designed for network attacks and monitoring. It includes a wide range of capabilities for network sniffing, attacks, and tests, including those for Wi-Fi, Ethernet, and Bluetooth.

Bluetooth Monitoring: For Bluetooth, Bettercap can scan the environment for devices, monitor the communications, and potentially identify malicious activities or vulnerabilities being exploited.

Usage: It can be used to actively monitor the air for Bluetooth packets, assess the security of our Bluetooth connection, and identify any unusual or potentially malicious activity that could indicate an attack on our smartphone.

How to Use Bettercap for Bluetooth Monitoring

Installation: Ensure Bettercap is installed on Kali Linux machine. It can typically be installed via package managers or from its GitHub repository.

Launch Bettercap: Run Bettercap with appropriate permissions (usually root) to access network interfaces and Bluetooth devices.

Enable Bluetooth Module: Within Bettercap, use the Bluetooth module to start scanning and monitoring Bluetooth traffic. We can use commands within Bettercap to focus on our smartphone's Bluetooth MAC address, monitor connections, and log any potential threats or unusual activities.

```
bettercap -eval "ble.recon on"
```

This command starts Bettercap and automatically begins Bluetooth Low Energy (BLE) reconnaissance.

Bettercap is a versatile tool that, when used correctly, can provide significant insights into Bluetooth security and potential vulnerabilities. However, the effectiveness of monitoring for Bluetooth attacks also depends on the attacker's techniques and the specific vulnerabilities they might exploit. Regularly updating your device's firmware and using strong, up-to-date security practices are also key to protecting against Bluetooth-based threats.

Q5. Your friends recommended to you a cool new Android application that can be easily downloaded from the app store. However, you (a security expert) do not trust the application developer, who also did not publish the application's source code. Which Kali Linux tool will you use to retrieve the equivalent source code, so that you can conduct a thorough security audit of the application, and why?

A: - For the scenario where we need to review the source code of an Android application without direct access to the source from the developer, the tool from the Kali Linux arsenal best suited for this task is Apktool.

Apktool

Purpose: Apktool is a tool for reverse engineering third-party, closed, binary Android apps. It can decode resources to nearly original form and rebuild them after making modifications, which is invaluable for security audits and research purposes.

Functionality: With Apktool, we can:

Decode APK files to extract all the components and resources, including XML files and the manifest file, which can be read and understood.

Rebuild decoded resources back to a binary APK/JAR.

Modify apps for customization or underpinning security assessments.

Use Case: Ideal for security professionals and researchers who want to inspect an app's code and resources for potential security flaws, privacy concerns, or malicious functionalities. It allows for an in-depth analysis of the app beyond what is readily visible on the surface.

Why Apktool for Source Code Retrieval?

While Apktool does not directly retrieve the "source code" in the format that developers write in (like Java for Android apps), it decodes the compiled .apk file into a form that is quite close to the original source. Specifically, it allows us to:

Extract and inspect the app's manifest and XML resources, which can reveal permissions, activities, service registrations, receivers, and providers that indicate how the app operates.

Decode the resources to a near-original state, allowing us to understand the app's layout, design, strings, and other resources.

Disassemble the DEX (Dalvik Executable) files to Smali code, which is an assembly language used in Android. This is as close as we can get to the original source code without having the actual Java source files. Smali code can be analyzed to understand the app's functionality, logic, and any potential malicious code.

Conducting a Security Audit

Decompile the APK: Use Apktool to decompile the APK file and extract its contents.

Review the Manifest: Check the app's manifest file for excessive permissions or intent filters that could be abused.

Inspect the Code: Analyze the Smali code or any decoded resource files for suspicious or malicious behavior.

Rebuild & Test: After understanding the app's components, we can modify them for testing purposes and rebuild the app to see how changes affect its behavior, which is useful in vulnerability research.

Conclusion

Apktool stands out for this purpose due to its ability to break down and reconstruct Android applications for a closer inspection of their operations and potential security flaws. This tool is crucial for anyone looking to conduct a thorough security audit of an Android app without access to its source code. However, it's important to note that analyzing Smali code requires a good understanding of Android app structure and programming concepts.