ESCUELA COLOMBIANA DE INGENIERIA
JULIO GARAVITO

IT SECURITY AND PRIVACY
GRUPO 1L

ETHIC HACKING PRACTICE

LAB 3

SUBMITTED BY:
JUAN PABLO FERNANDEZ GONZALES
MARIA VALENTINA TORRES MONSALVE

SUBMITTED TO:
Ing. DANIEL ESTEBAN VELA LOPEZ

Security Assessment Findings Report

Business Confidential

Date: February 10th, 2025
Project: 123-25
Version: 1.0

Contents

Confidentiality Statement

Disclaimer

This security assessment provides a snapshot in time. Findings and recommendations reflect information gathered during the assessment period and do not account for changes made afterward.

Contact Information

Name	Title	Contact Information
Maria Valentina Torres	Spti Student	Maria.torres-m@mail.escuelaing.edu.co
Juan Pablo Fernández Gonzalez	Spti Student	Juan.fernandez-g@mail.escuelaing.edu.co

Assessment Overview

From February 1st, 2025 to February 10th, 2025, SPTI class conducted a security assessment focused on external penetration testing. The methodology followed industry best practices including:

- **Planning** – Defining goals and scope.
- **Discovery** – Scanning and enumeration.
- **Attack** – Exploitation and validation of vulnerabilities.
- **Reporting** – Documenting findings and recommendations.

Assessment Components

External Penetration Test

A penetration test was performed to identify vulnerabilities in publicly accessible services. The following tests were executed:

1. **Port Scanning**
   - **Command:** nmap 10.10.11.48 -sSCV -Pn -T4
   - **Findings:** Open ports 22 (SSH) and 80 (HTTP) were detected.
2. **Advanced Port Scanning (TCP & UDP)**
   - **Command:** nmap -p --open -sS -sU --min-rate 5000 -n -Pn -vvv 10.10.11.48 -oG scan
   - **Findings:** Further enumeration detected additional UDP services.
3. **SNMP Analysis**
   - **Command:** snmp-check 10.10.11.48
   - **Findings:** A user named "steve" was found, along with a "daloradius" service.
4. **Web Directory Enumeration**
   - **Command:** dirsearch -u "http://10.10.11.48/daloradius/" -t 50
   - **Findings:** Several directories and files were identified.
5. **Configuration File Retrieval**
   - **Command:** wget http://10.10.11.48/config.yml
   - **Findings:** Credentials and sensitive configurations were extracted.
6. **Sensitive File Analysis**
   - **Command:** cat config.yml
   - **Findings:** Identified additional vulnerabilities within the system.
7. **Password Cracking**

    - **Command:** john hash.txt --wordlist=/usr/share/wordlists/rockyou.txt --format=raw-MD5
    - **Findings:** Recovered password "underwaterfriends".
8. **SSH Connection**
    - **Command:** ssh svcmosh@10.10.11.48
    - **Findings:** Established access to the target system.
9. **Privilege Escalation Check**
    - **Command:** sudo -l
    - **Findings:** Identified potential privilege escalation vectors.
10. **Root Access via Mosh**
    - **Command:** mosh --server="sudo /usr/bin/mosh-server" localhost
    - **Findings:** Successfully gained root access.

Finding Severity Ratings

Severity	CVSS V3 Score Range	Definition
Critical	9.0-10.0	Immediate action required. Full system compromise possible.
High	7.0-8.9	Exploitation could lead to significant data loss or downtime.
Moderate	4.0-6.9	Exploitable vulnerabilities with added complexity.
Low	0.1-3.9	Minimal impact vulnerabilities that should be patched.
Info	N/A	Informational findings without direct security risk.

Executive Summary

The security assessment identified multiple critical vulnerabilities that could allow unauthorized access to the target system. The most significant weaknesses included exposed services, weak authentication mechanisms, and configuration files containing sensitive information. The testing methodology used a structured approach to assess the security posture of the system, leveraging both automated scanning tools and manual testing techniques.

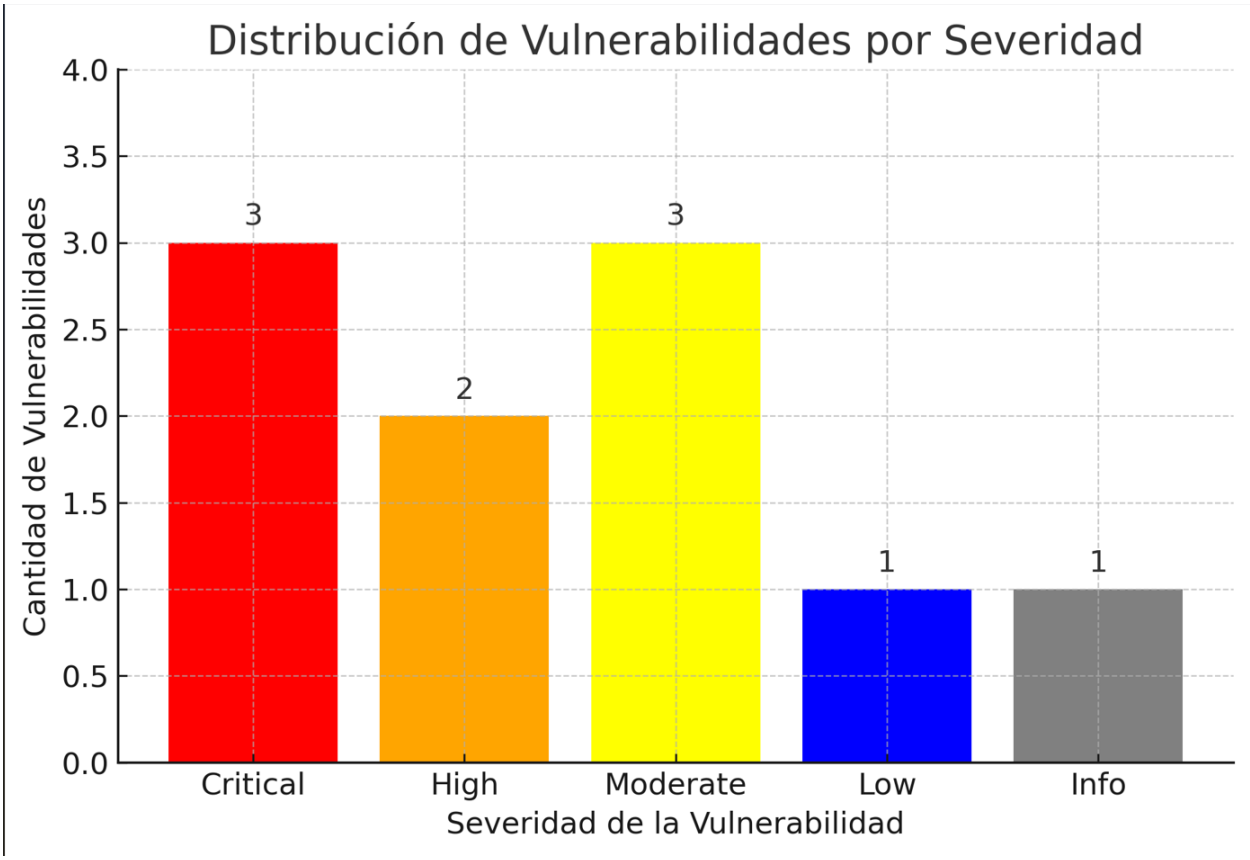
During the assessment, open ports were discovered that exposed SSH and web services to potential attackers. Further enumeration using SNMP analysis revealed an active user account, increasing the attack surface. By retrieving configuration files through directory brute-forcing, sensitive credentials were uncovered, enabling unauthorized access. Password cracking techniques successfully revealed user credentials, which facilitated an SSH connection to the

target system. Subsequently, privilege escalation techniques were applied, ultimately leading to full root access.

The vulnerabilities identified in this assessment pose significant risks to the confidentiality, integrity, and availability of the system. Immediate remediation is strongly recommended to mitigate these risks. Suggested actions include enforcing strong authentication mechanisms, restricting unnecessary service exposure, implementing robust password policies, and applying proper access controls to sensitive configuration files.

Attack Summary:

1. Enumerated open ports using nmap.
2. Extracted user and service information via snmp-check.
3. Identified sensitive configuration files through dirsearch.
4. Cracked user credentials with john and gained SSH access.
5. Escalated privileges to root using misconfigured permissions.
6. Documented vulnerabilities and provided remediation recommendations.



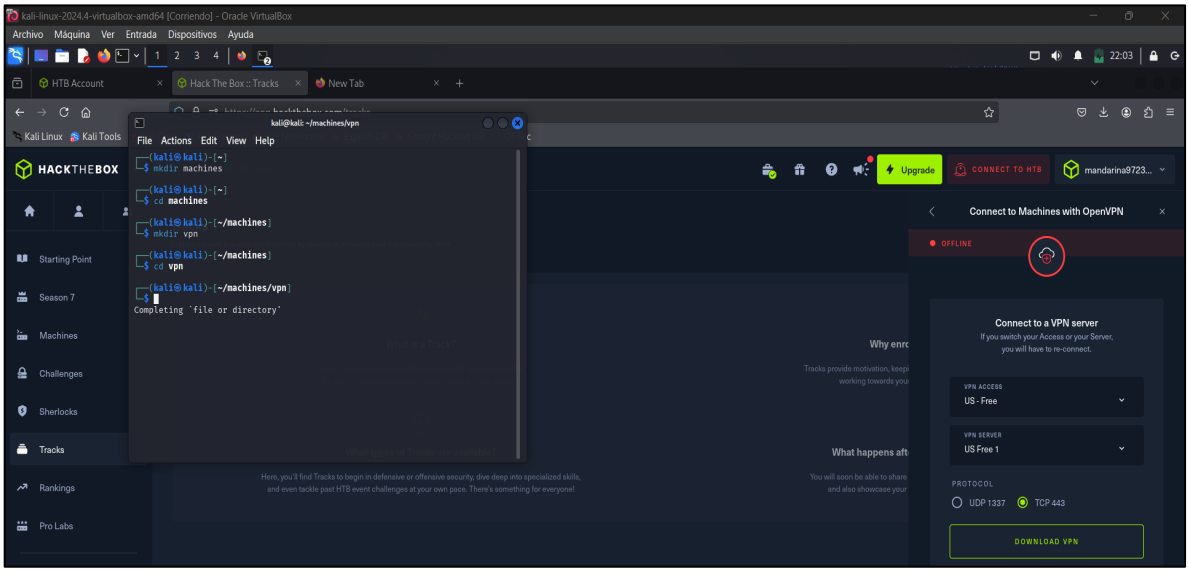
The security posture of the system requires significant improvements to prevent unauthorized access and potential exploitation. It is recommended to conduct periodic security assessments and implement security best practices to maintain a resilient infrastructure.

Security Weaknesses & Recommendations

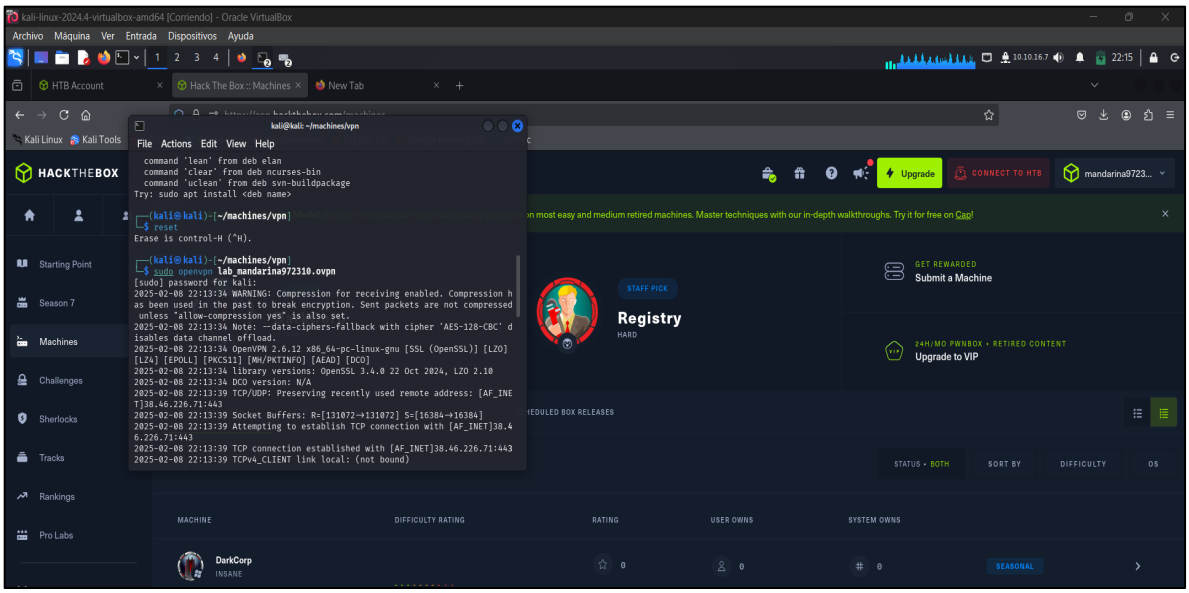
1. **Weak Password Policy**
    - Implement password complexity requirements.
    - Enforce Multi-Factor Authentication (MFA).
2. **Exposed Configuration Files**
    - Restrict access to sensitive files.
    - Use secure storage solutions.
3. **Privilege Escalation Vulnerabilities**
    - Restrict unnecessary sudo privileges.
    - Monitor user activity for anomalies.
4. **Unrestricted Directory Access**
    - Implement access controls.
    - Regularly audit exposed directories.

Detailed Step-by-Step Compromise Process

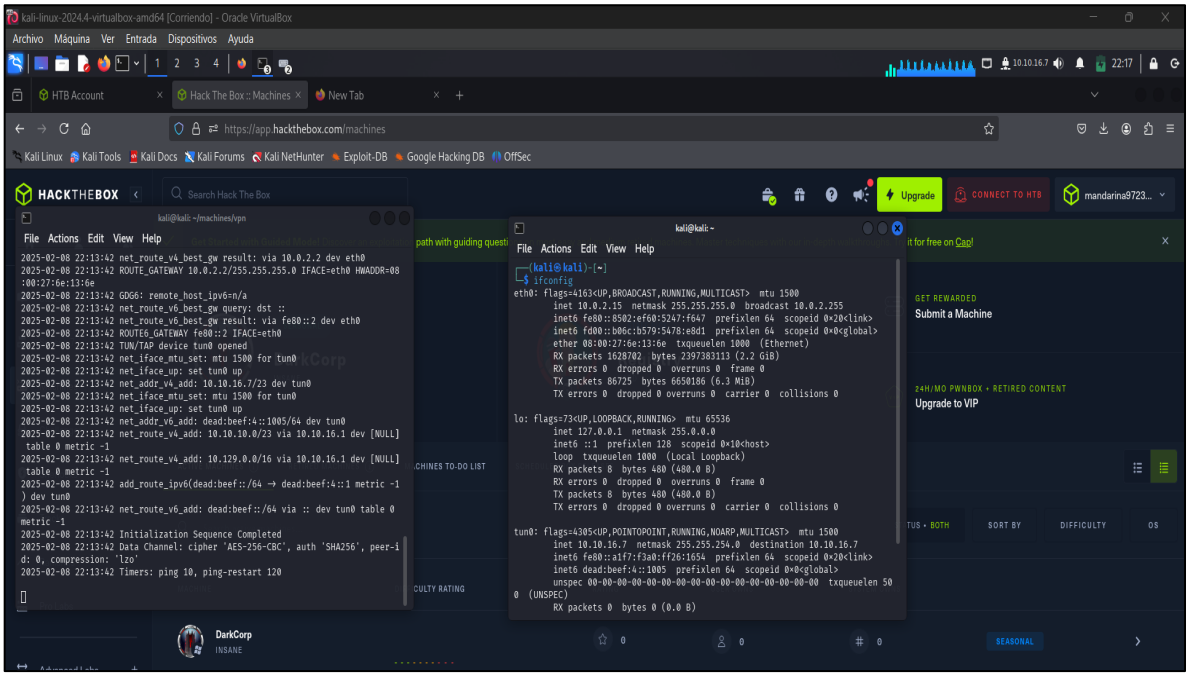
The first thing we will do is download the VPN from the browser of our Kali Linux virtual machine and create the directory to store all the information related to the UnderPass machine.

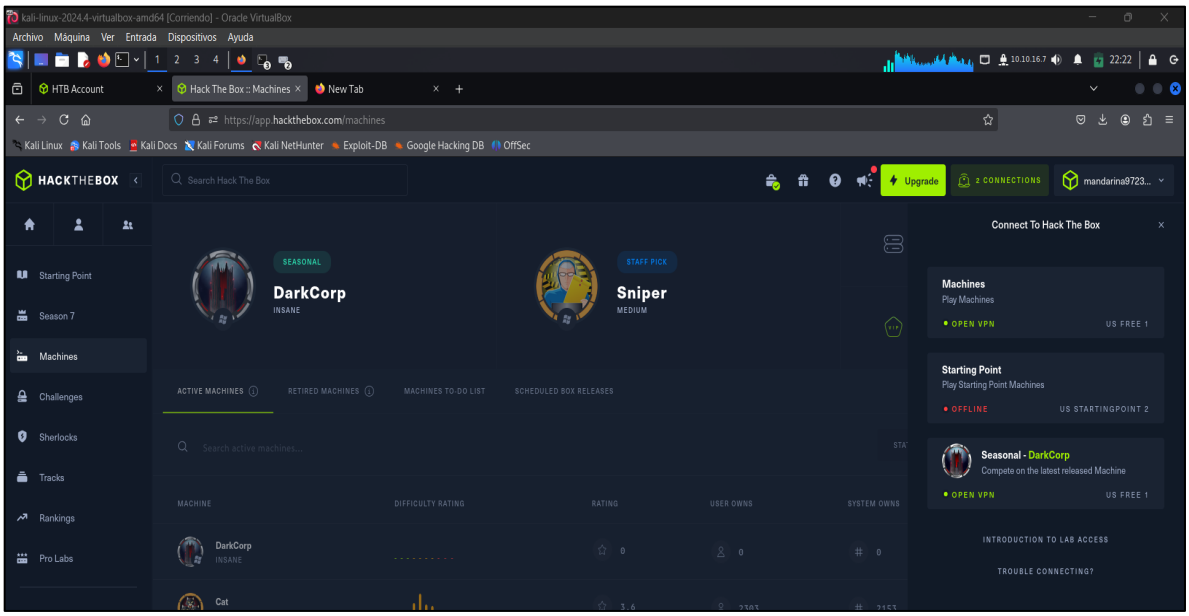


To connect to the Hack The Box servers, we will use the command **sudo openvpn lab_mandarina972310.ovpn**, which follows the structure **openvpn filename**.

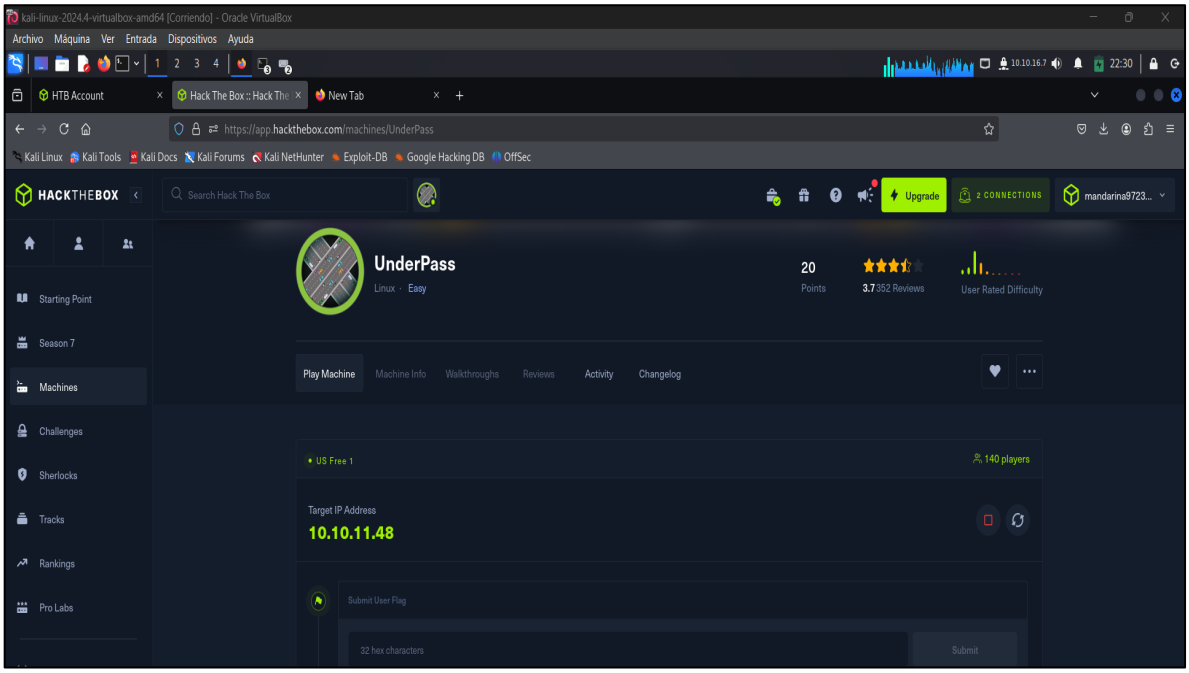


With the **ifconfig** command, we can verify that we are indeed connected. Additionally, we can check the active connections on the website, where they will be displayed.

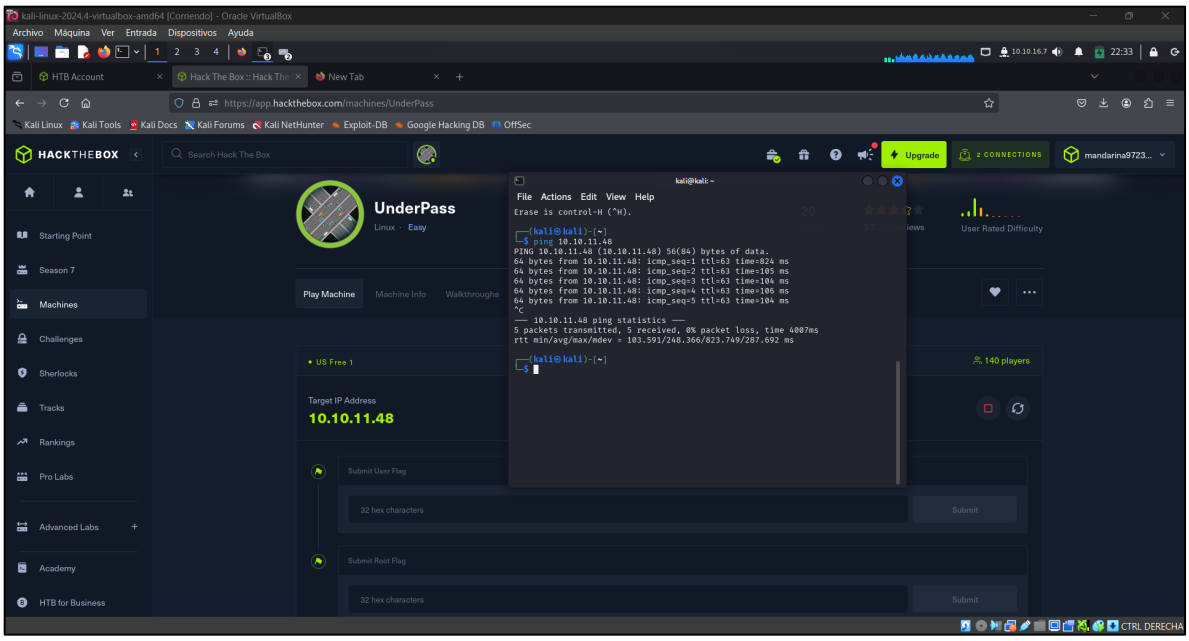




For this hacking practice, we selected the UnderPass machine, which has an easy difficulty level. To proceed with the first step, we need to obtain the machine's IP address (**10.10.11.48**).

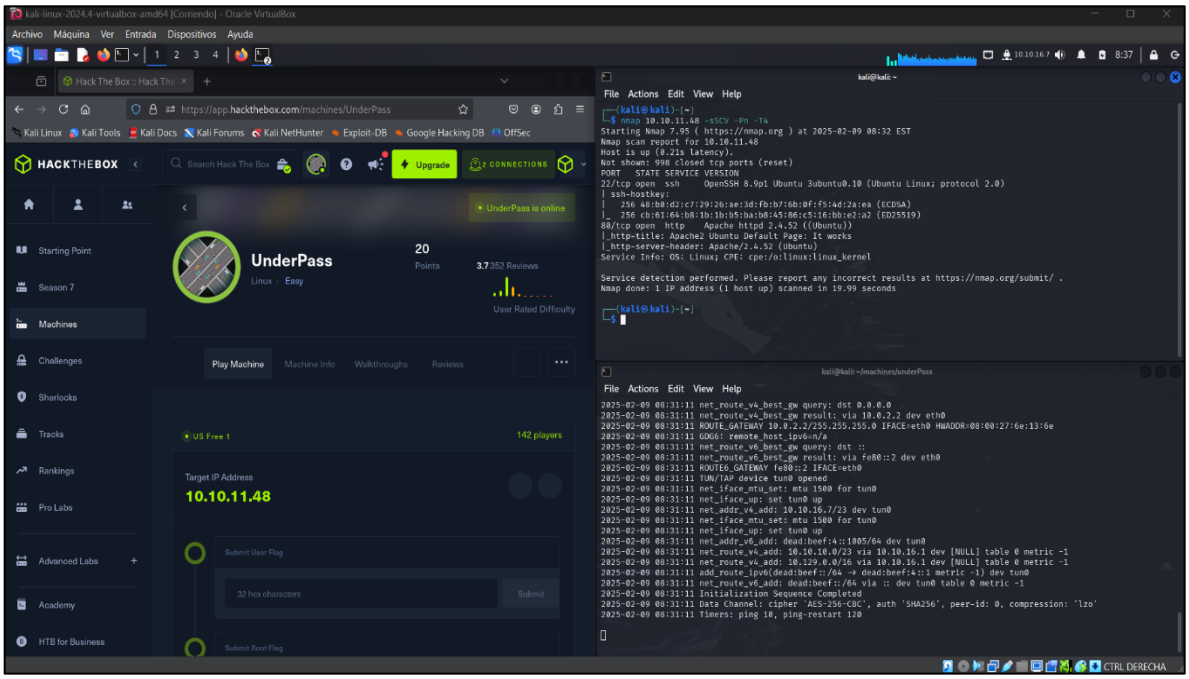


We will perform a **ping** to the address **10.10.11.48** to verify that it is still active and ready for the practice.



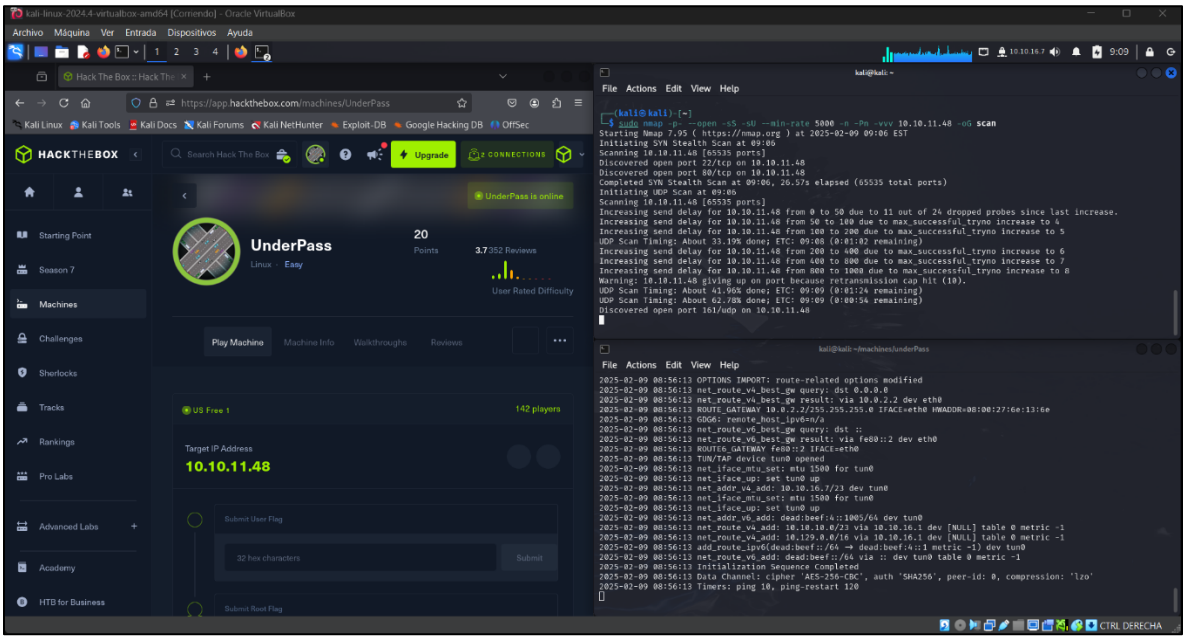
We will perform a deep scan with **Nmap** using the machine's IP address. A basic scan will not provide valuable information, so we will use specific flags such as **-sS** and **-sU** to search for open ports on the machine.

Using the commands **nmap 10.10.11.48 -sSCV -Pn -T4** and **nmap -p --open -sS -sU --min-rate 5000 -n -Pn -vvv 10.10.11.48 -oG scan**, we will identify that **TCP ports 22 and 80** are open, as well as **UDP port 161**, which is running a protocol.



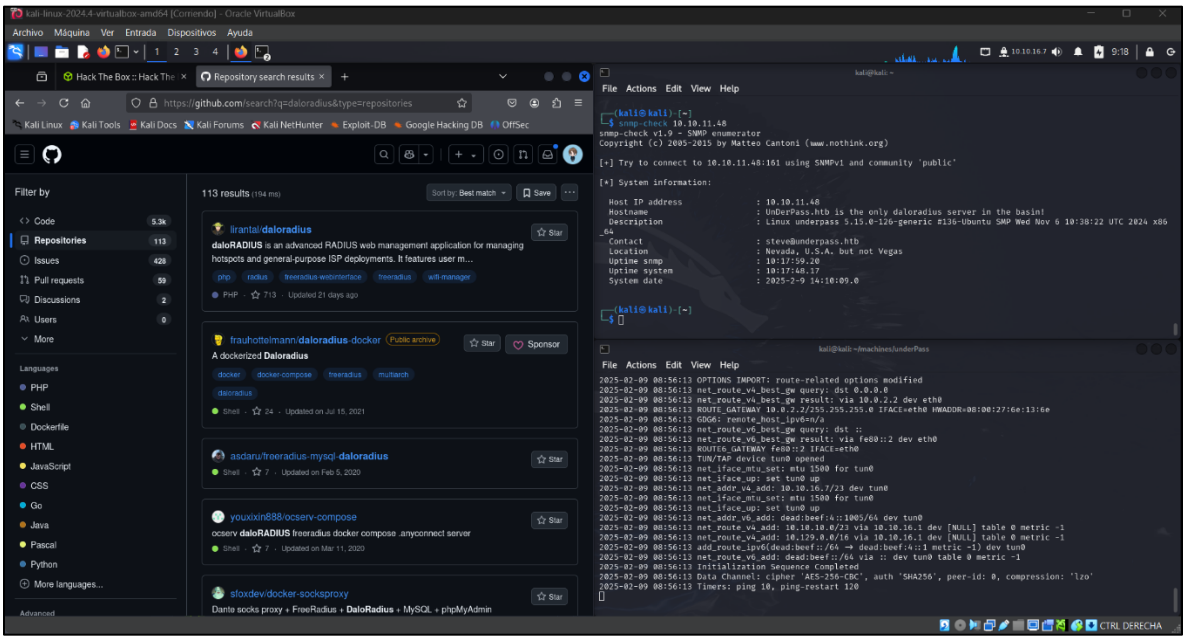
- ***Nmap:*** A tool for network scanning and open port detection.

- ***10.10.11.48:*** IP address of the target machine.
- ***-sS:*** TCP SYN scan (half-open, does not complete the connection).
- **-sC:** Uses default scanning scripts.
- **-sV:** Detects versions of services running on open ports.
- ***-Pn:*** Skips host discovery (assumes the machine is active).
- ***-T4:*** Uses an aggressive scanning speed.



- **-p --open**: Searches only for open ports.
- **-sS**: SYN scan (TCP).
- **-sU**: UDP port scan.
- **--min-rate 5000**: Ensures a minimum rate of 5000 packets per second.
- **-n**: Disables DNS resolution.
- **-Pn**: Skips host discovery.
- **-vvv**: Displays results in verbose mode.
- **-oG scan**: Saves the results in a grepable format file.

Interestingly, the **SNMP port** is available and open. This usually indicates misconfigurations on the device, so let's see if we can gather some data from this endpoint using the command **snmp-check 10.10.11.48**.

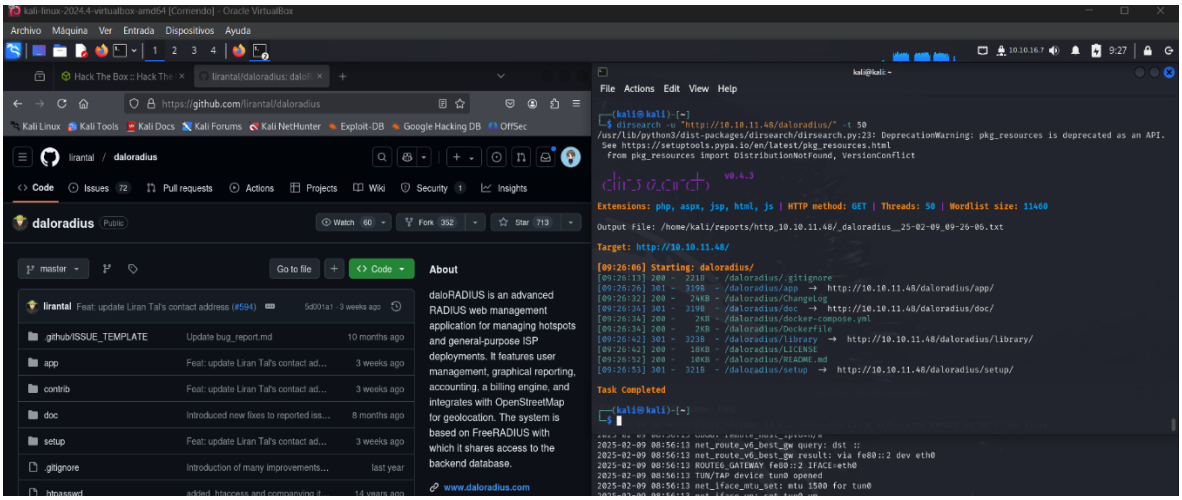


- **snmp-check**: A tool for gathering information from SNMP.
- **10.10.11.48**: Target machine's IP address.

We can see that there is a **daloRADIUS server** and a user on **underpass.htb** named **Steve**. Let's take a look at what **daloRADIUS** is. It appears to be a framework for deploying **FreeRADIUS** servers.

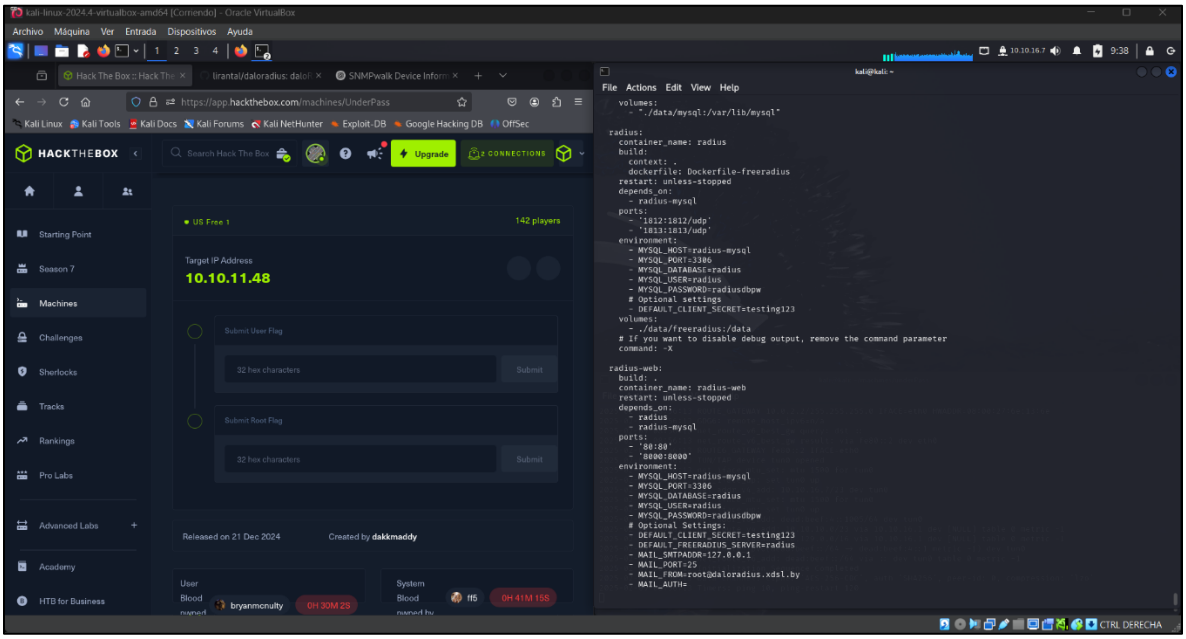
RADIUS: The **Remote Authentication Dial-In User Service** is a network protocol that provides centralized authentication, authorization, and accounting management for users connecting to and using a network service.

We will further investigate the server's directory structure using the command: **dirsearch -u "http://10.10.11.48/daloradius/" -t 50** . This will help us search for **hidden directories** on the web server.

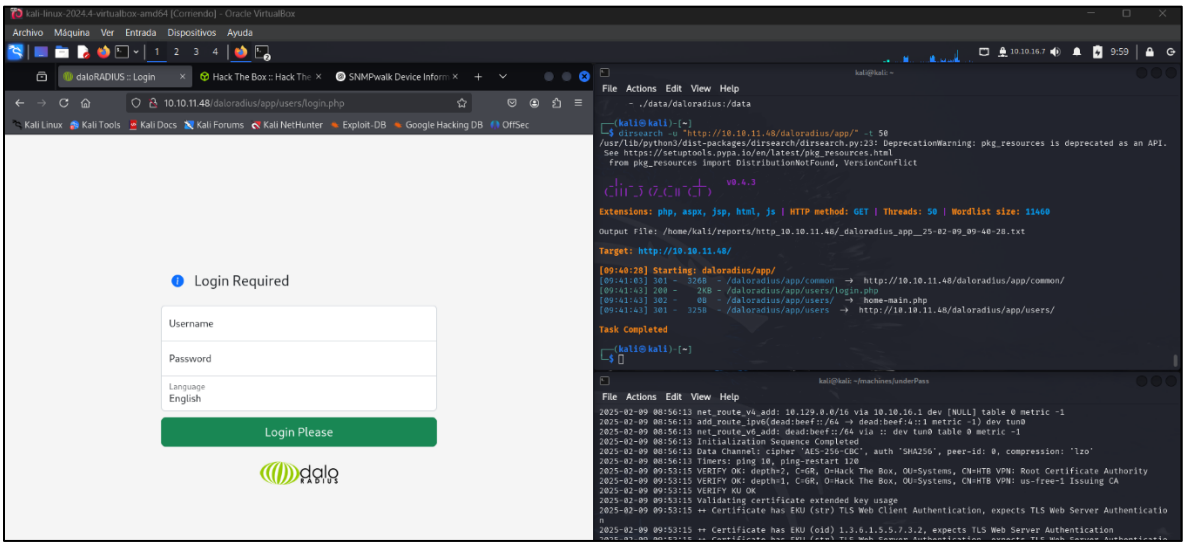


- **dirsearch**: Searches for hidden directories on web servers.
- **-u**: Specifies the target URL.
- **-t 50**: Uses 50 threads to speed up the scan.

We found several exposed endpoints that we can continue analyzing to gather more information. However, first, we will download the **Docker-compose.yml** file using the command **wget http://10.10.11.48/daloradius/docker-compose.yml** .Then, we will review the **environment configurations**.

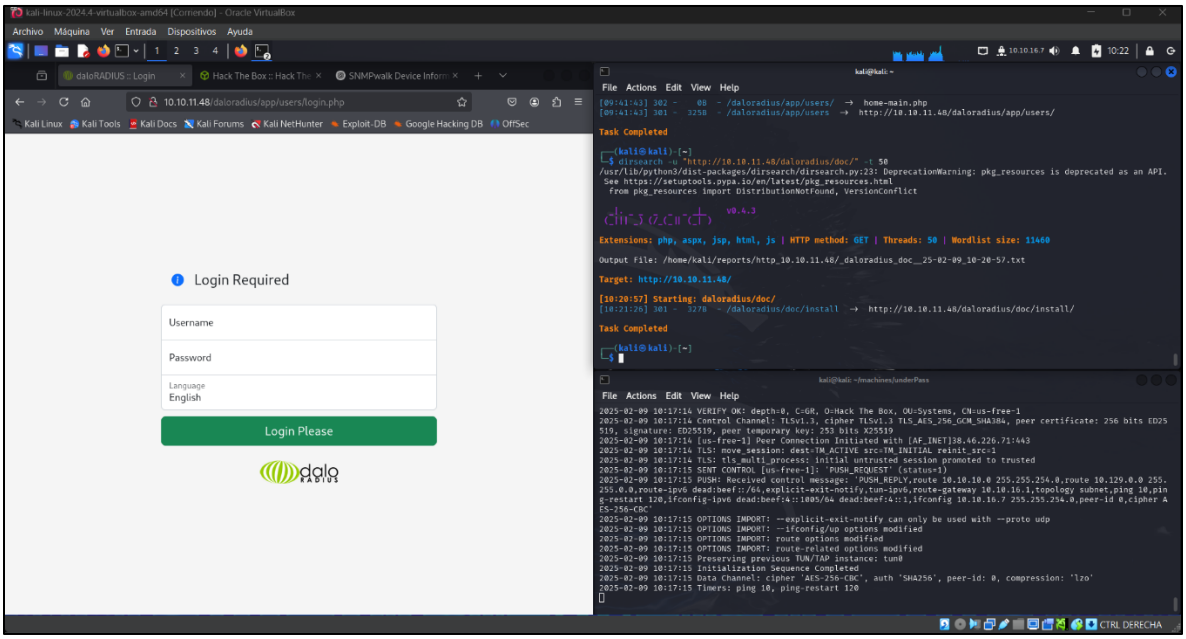


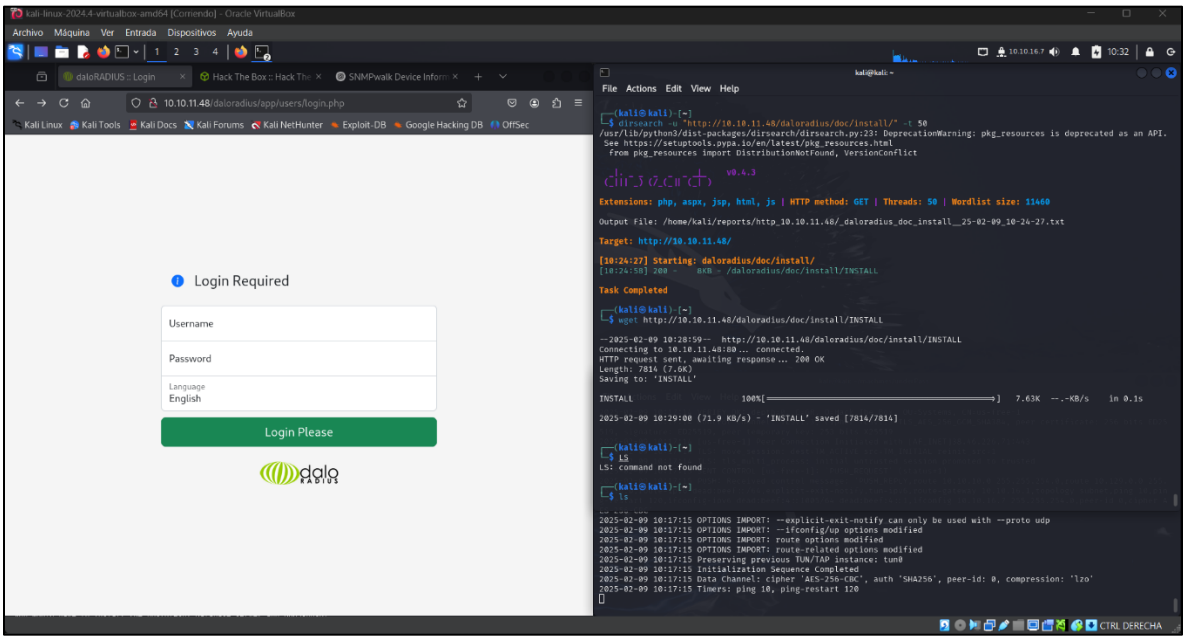
We run **dirsearch** again on the application to look for more directories and files that are not visible at first glance. Analyzing the results, we find the **source file of a web page** that contains a **login** to access the **RADIUS service**.



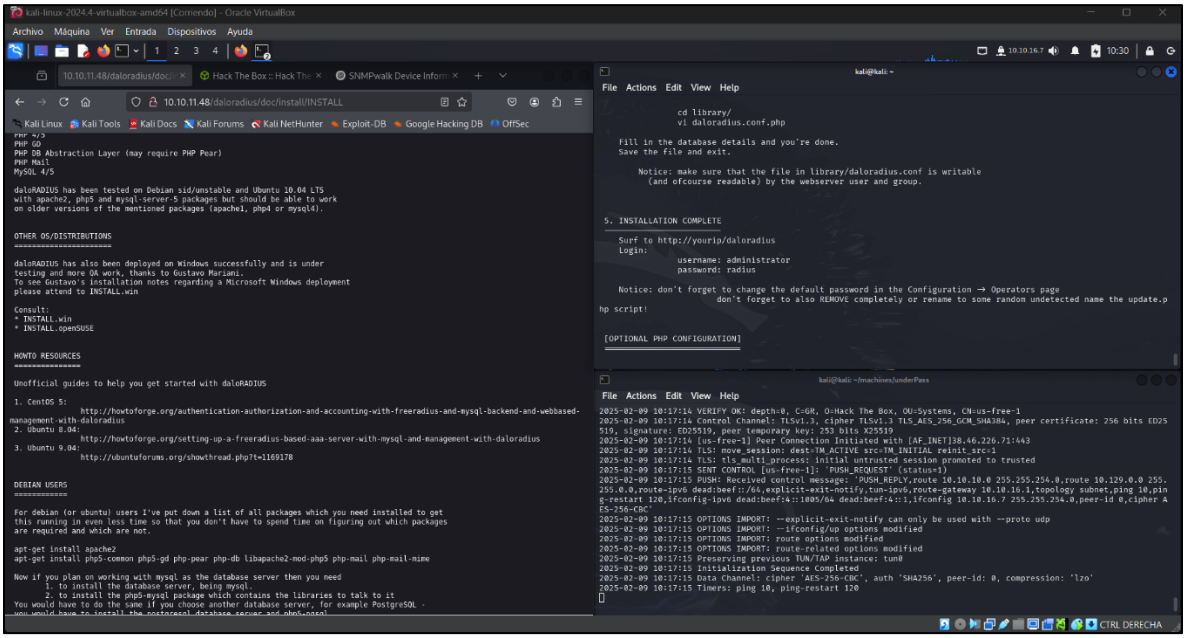
We analyze the **docs directory** to see what information it contains about installation, usage, and execution.

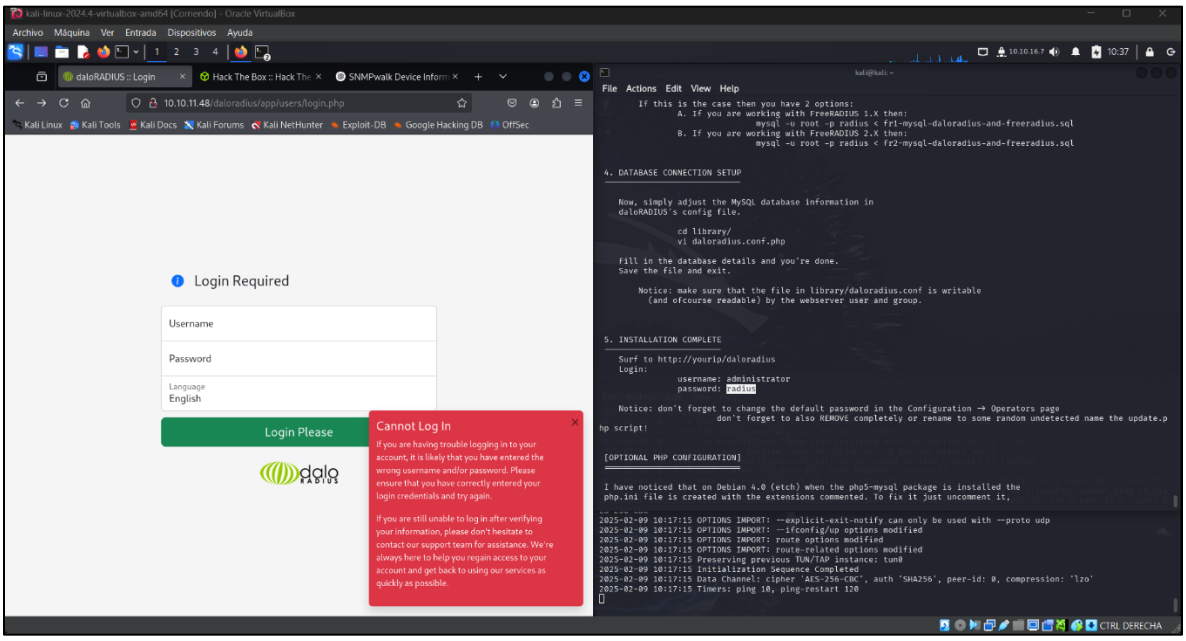
During this process, we find another **directory** and repeat the analysis. In this new directory, we discover an **INSTALL file** that contains **sensitive details** about the web application.



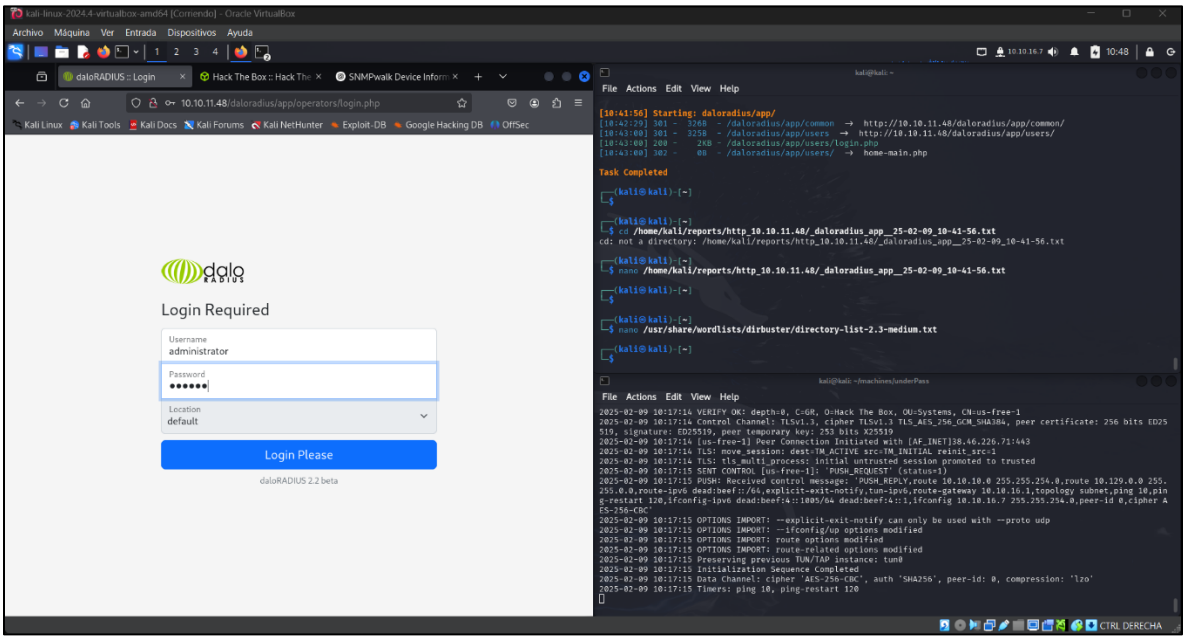


We download the **INSTALL file**, where we confirm that it contains **all the sensitive information** about the application. Additionally, we verify that, through its path, we can access it from a browser without any restrictions. Inside the file, we find a **default user**, but when we try to log in, we receive an **error message**.

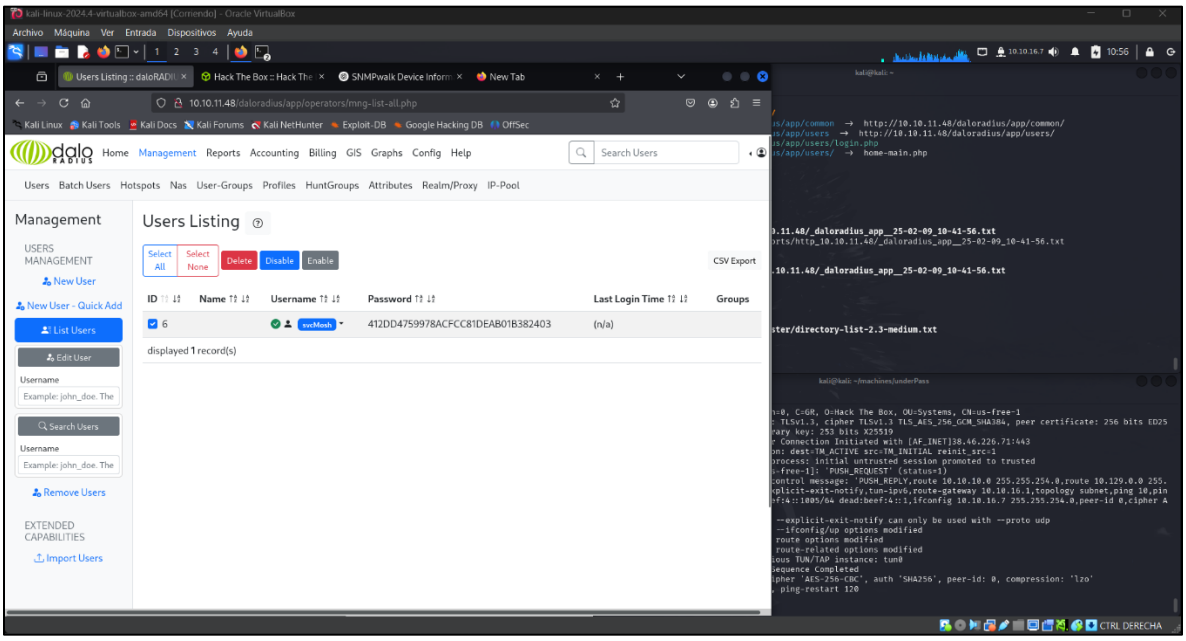




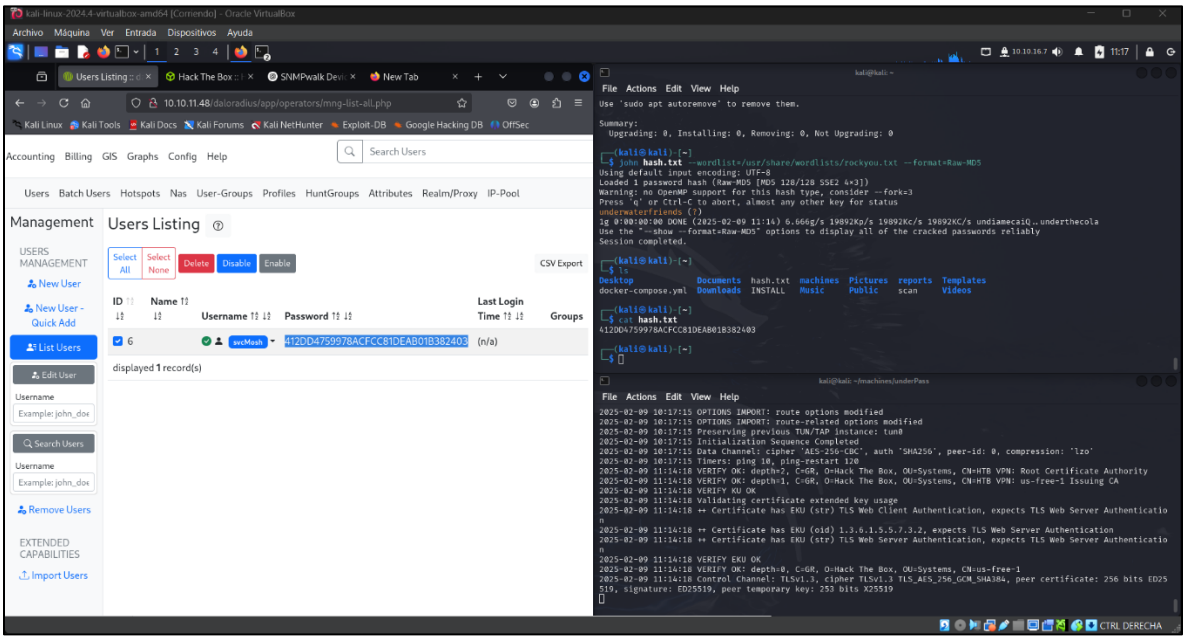
In a second, deeper analysis, we found another **exposed endpoint** for **operators**. Upon accessing it, we discovered that it also leads to a **login page**. We tested the previously found credentials and successfully **gained access**.



In the **users section**, we were able to list and view the **password** of the only user. However, it is **encrypted in MD5 format**.

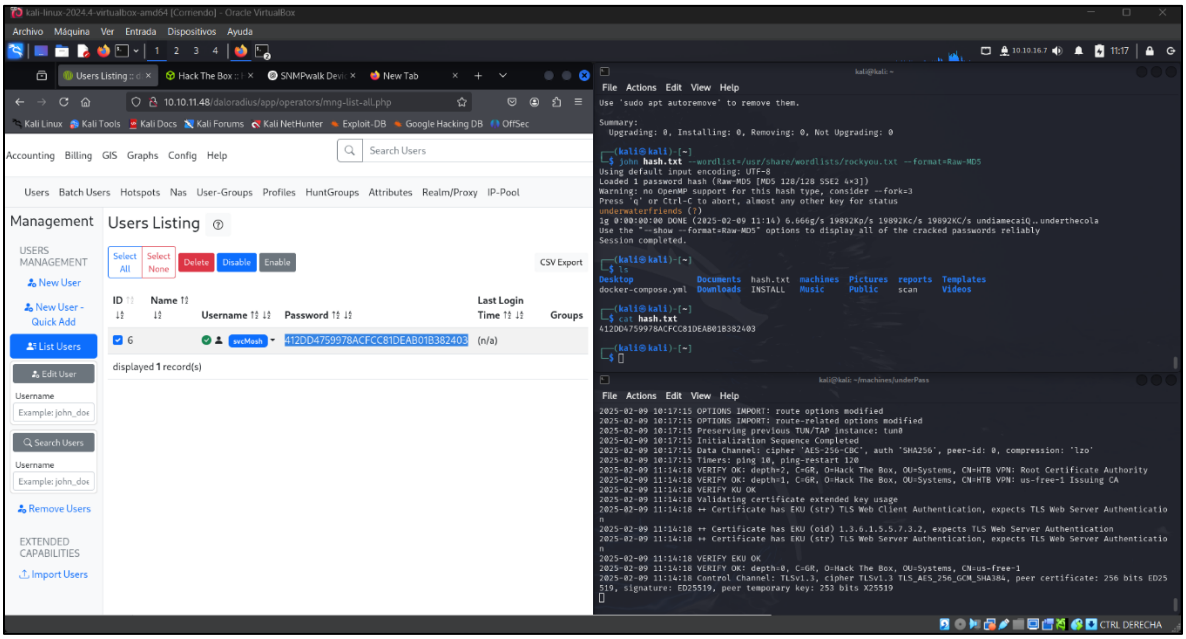


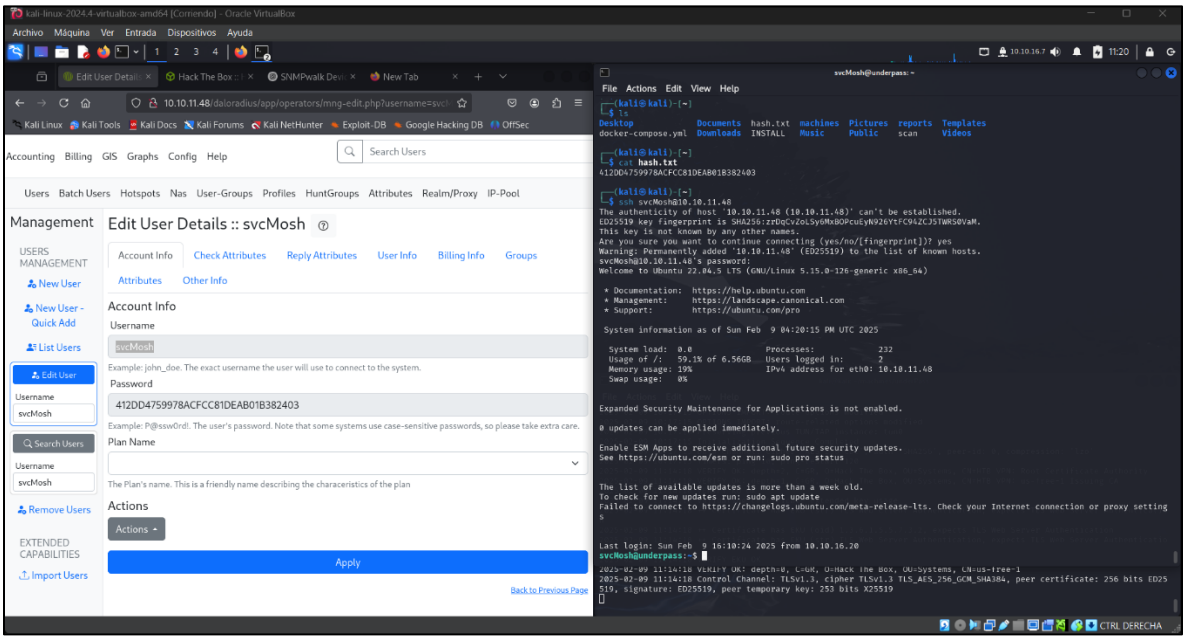
To decrypt the password, we use the following command with **John the Ripper**: **john hash.txt --wordlist=/usr/share/wordlists/rockyou.txt --format=raw-MD5**. This will attempt to crack the **MD5 hash** using the **RockYou** wordlist.



- **john**: A tool for cracking passwords.
- **hash.txt**: File containing the MD5 hash.
- **--wordlist=/usr/share/wordlists/rockyou.txt**: Specifies the password dictionary.
- **--format=raw-MD5**: Defines the hash format as raw MD5.

The result obtained with this command is **underwaterfriends**. Now, we will connect via **SSH** using the username and password we just found with the command **ssh** **svcmosh@10.10.11.48**.

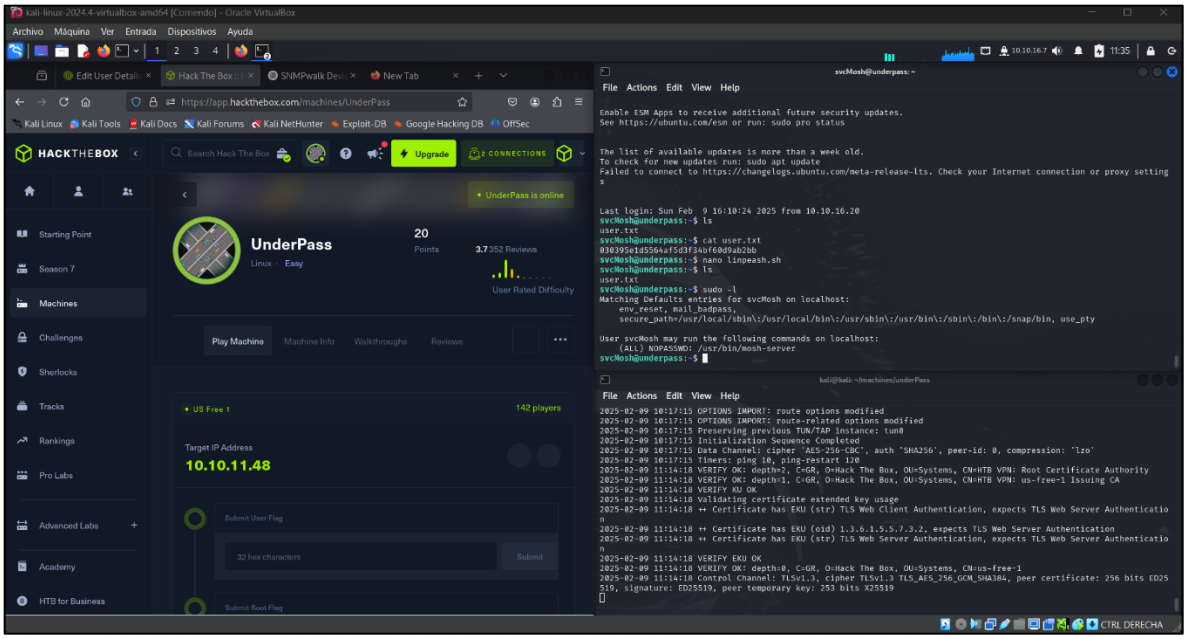




- **ssh**: SSH client used to connect to a remote machine.
- **svcmosh**: Username of the target machine.
- **10.10.11.48**: IP address of the target machine.

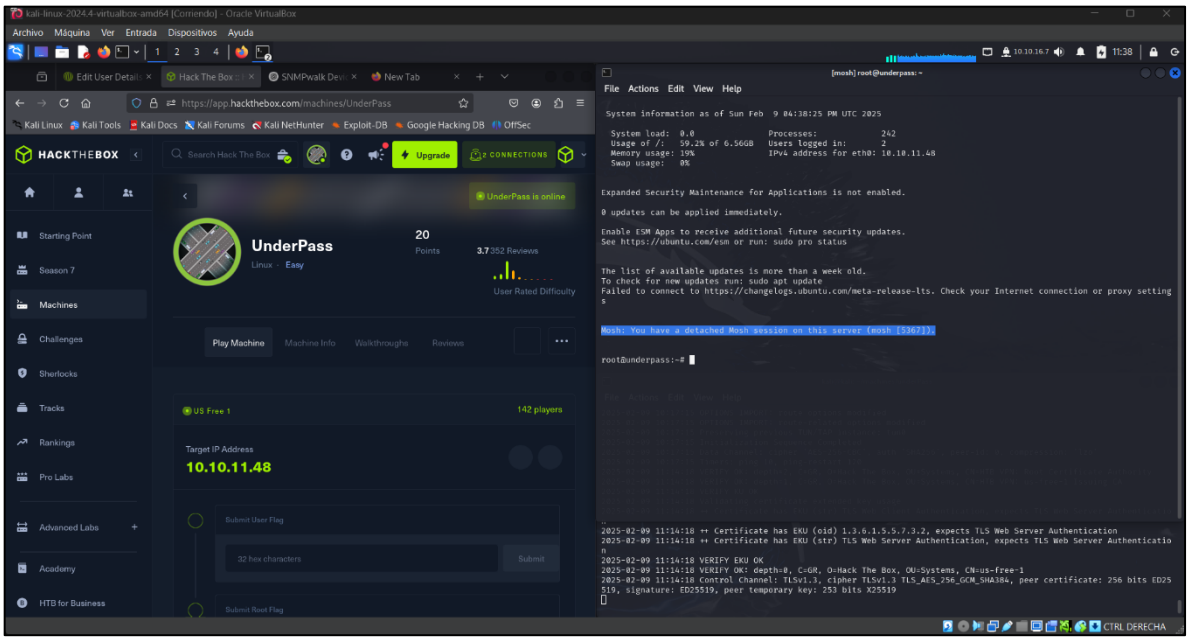
In the **root directory**, we find a **file** containing one of the **flags** that we need to submit in **Hack**

The Box. Additionally, by running **sudo -l**, we can check the **permissions** of the user we are connected with via **SSH**.



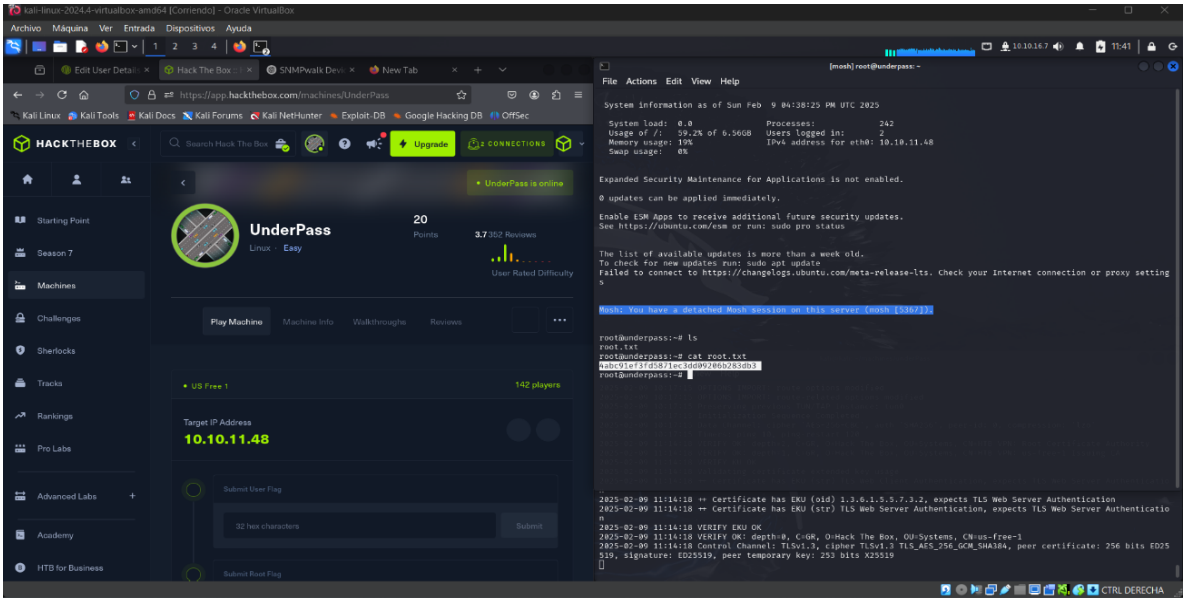
- **sudo -l**: Displays the commands the user can execute with elevated privileges.

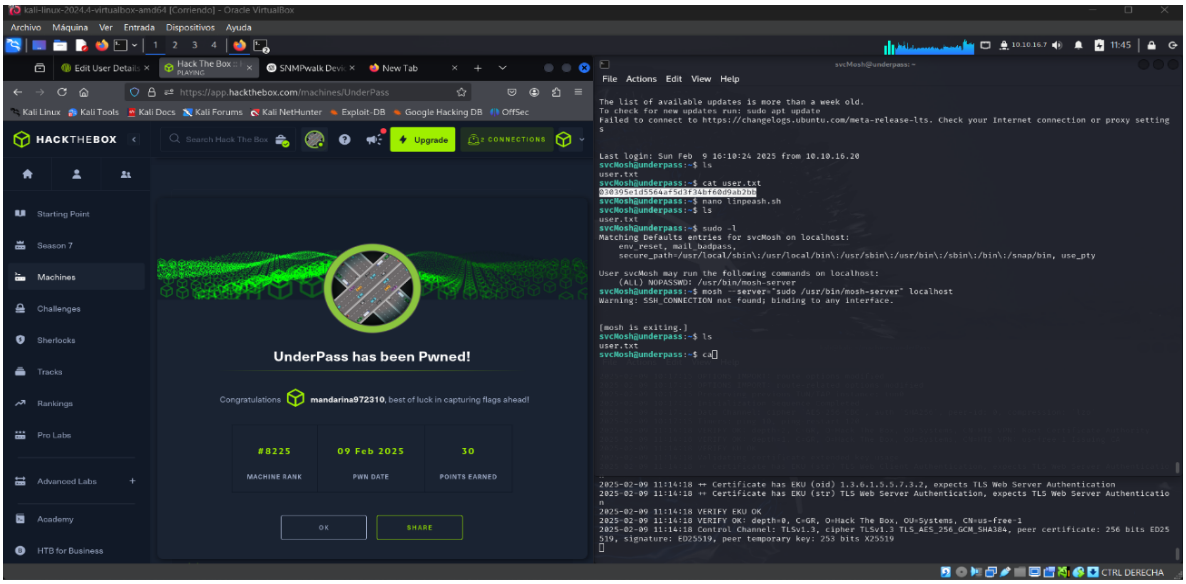
We can see mentions indicating that **no password is required** for the binary **/usr/bin/mosh-server**. **Mosh** is familiar since it also appeared in the **username**, suggesting that this is a **service account** for a specific server. **Mosh (Mobile Shell)** is an alternative to **SSH**, generating shell sessions on a **random port above 60000** and creating a **key for user authentication**. To escalate privileges, we use the command **mosh --server="sudo /usr/bin/mosh-server" localhost**.



- **mosh**: Remote connection tool.
- **--server="sudo /usr/bin/mosh-server"**: Runs the Mosh server with elevated privileges.
- **localhost**: Connects to the local machine.

In the **root directory**, we find a **root.txt** file, which contains the **second flag** needed to submit on the **Hack The Box**website.





Conclusion

The security assessment demonstrated that multiple vulnerabilities exist within the system, allowing unauthorized access and potential privilege escalation. Remediation steps should be prioritized to secure the environment effectively.

Additional Reports:

- Full vulnerability scan details.
- Raw penetration testing logs.