4 April 2025

OFFSEC FUNBOXROOKIE

Kein System ist sicher
Dreistigkeit siegt - Strebe das Unmögliche an

PantherBear

Certified GreyHat Hacker

Table of Contents

1.0 Offensive Security Exam Penetration Test Report

1.1 Introduction

The Offensive Security Exam penetration test report contains all efforts that were conducted in order to pass the Offensive Security exam. This report will be graded from a standpoint of correctness and fullness to all aspects of the exam. The purpose of this report is to ensure that the student has a full understanding of penetration testing methodologies as well as the technical knowledge to pass the qualifications for the Offensive Security Certified Professional.

1.2 Objective

The objective of this assessment is to perform an internal penetration test against the Offensive Security Exam network. The student is tasked with following a methodical approach in obtaining access to the objective goals. This test should simulate an actual penetration test and how you would start from beginning to end, including the overall report. An example page has already been created for you at the latter portions of this document that should give you ample information on what is expected to pass this course. Use the sample report as a guideline to get you through the reporting.

1.3 Requirements

The student will be required to fill out this penetration testing report fully and to include the following sections:

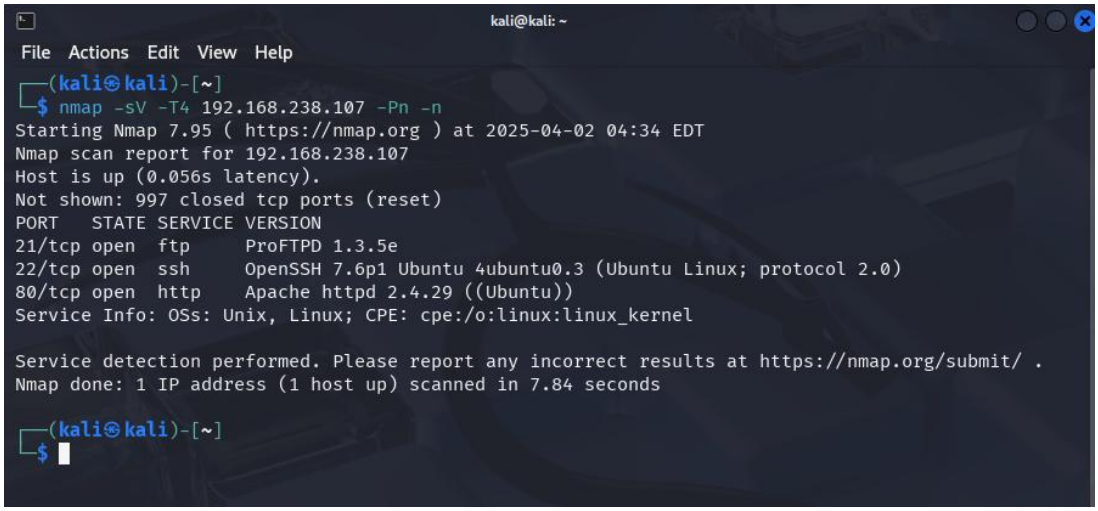
- Overall High-Level Summary and Recommendations (non-technical)
- Methodology walkthrough and detailed outline of steps taken
- Each finding with included screenshots, walkthrough, sample code, and proof.txt if applicable.
- Any additional items that were not included

2.0 System IP: 192.168.238.107

2.1 Service Enumeration

Port no.	Service	Version
21	ftp	ProFTPD 1.3.5e
22 80	ssh http	OpenSSH 7.6p1 Apache httpd 2.4.29

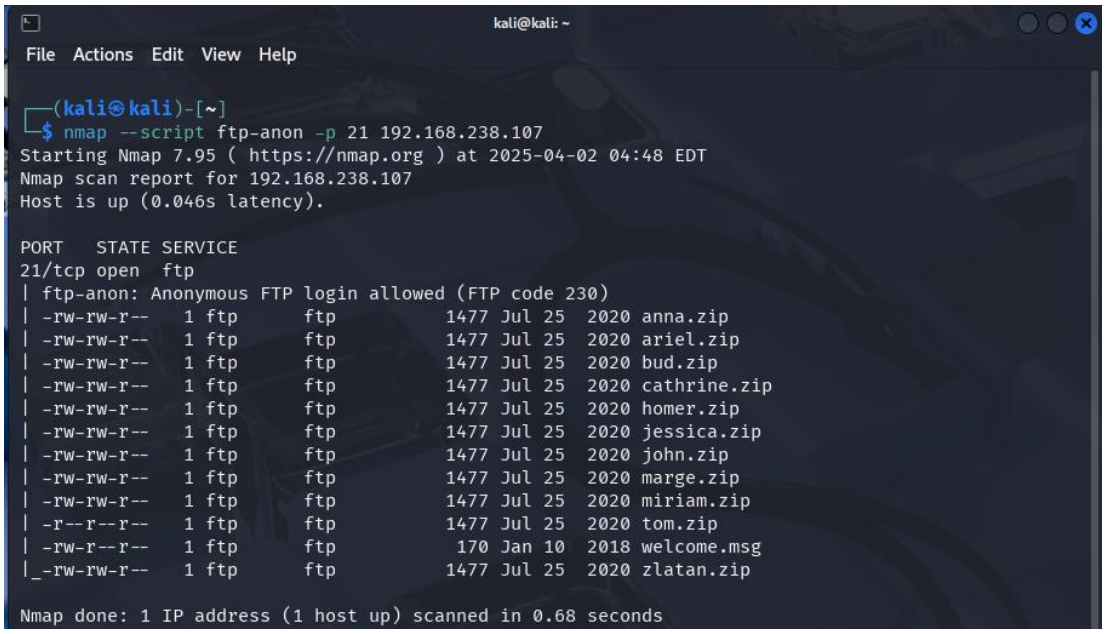
Nmap Scan Results:



Command: **nmap -sV -T4 192.168.238.107 -Pn -n**

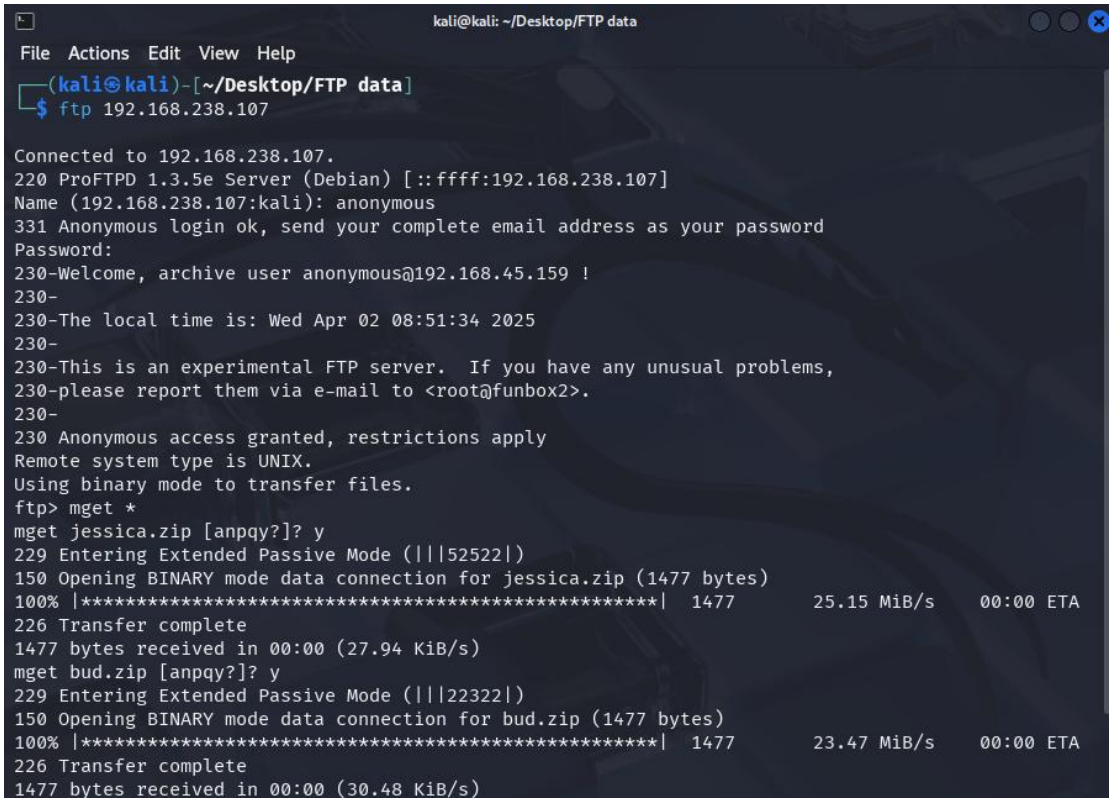
The command lists the service versions running on open ports, and the aggressive network scan is accelerated by disabling host discovery and DNS resolution.

2.2 FootHold on FTP and cracking the ZIPs



Command: **nmap --script ftp-anon -p 21 192.168.238.107**

The Nmap command uses the **ftp-anon** script to scan the target machine on port 21 to check if the FTP server permits anonymous access. The scan result shows FTP code 230, indicating that the server does allow anonymous access. Additionally, zip files of different users are found on the server.



Using FTP Anonymous login and exfiltrating every single zip file into ‘FTP Data’ directory

FTP Command: **mget ***

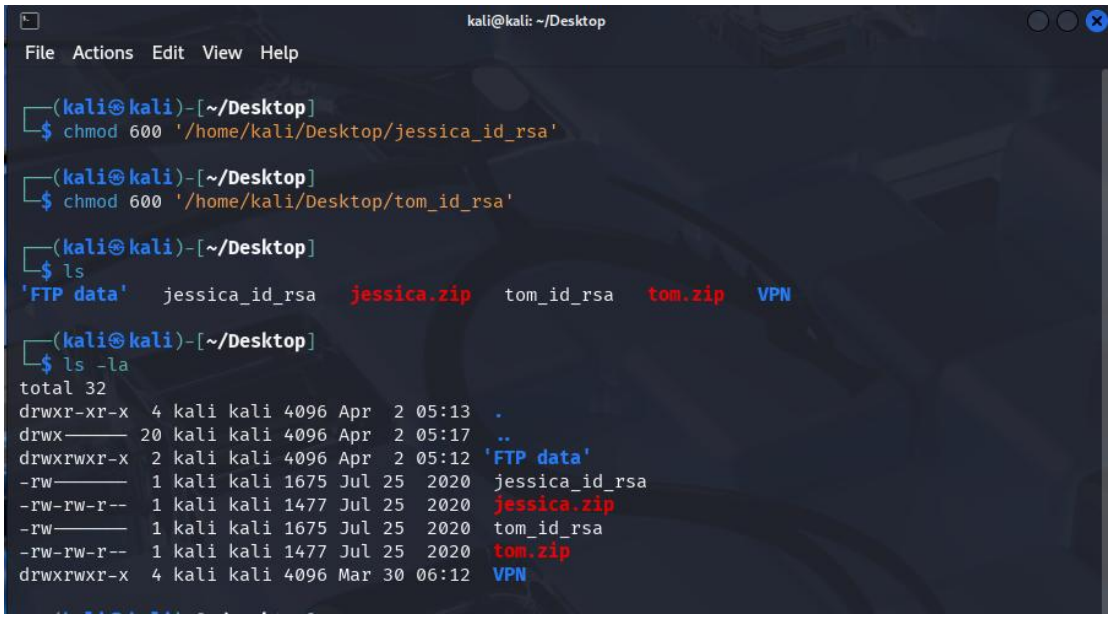
Command: **sudo john --wordlist=‘/usr/share/wordlists/rockyou.txt’ tomhash**

I converted the password-protected ZIP files into hash file format using the tool **zip2john** and cracked them using **John the Ripper**

Command: **zip2john tom.zip > tomhash**

User	ZIP file Password
cathrine	catwoman
tom	iubire

2.3 Gaining Access



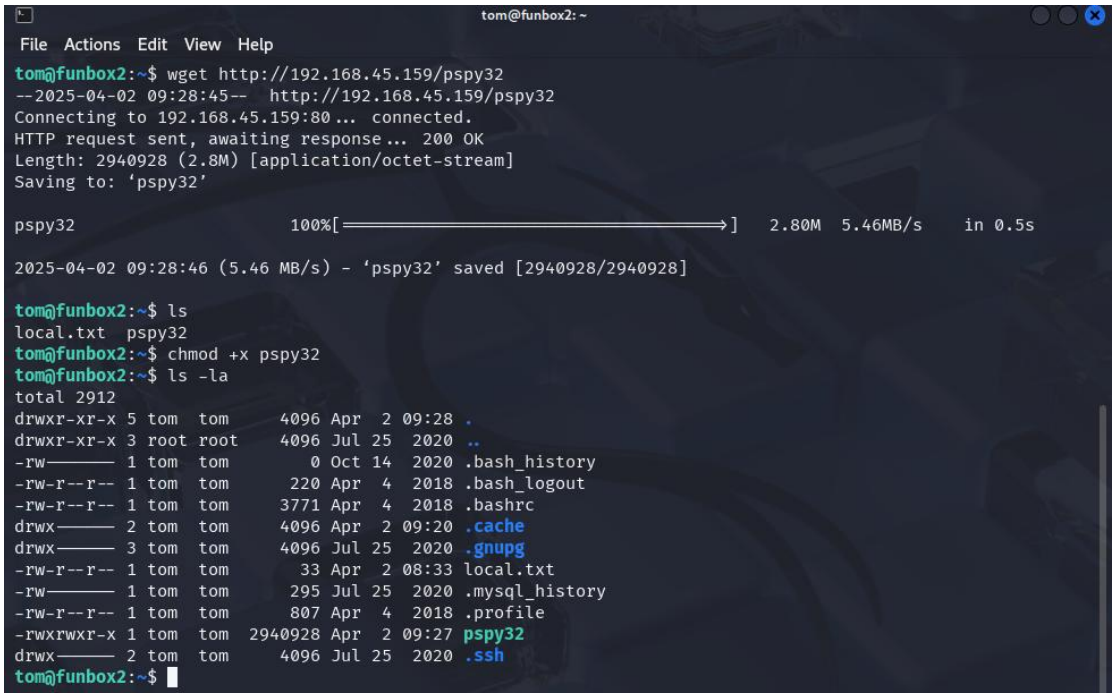
The ZIP files contains SSH private keys assigned to the respective users. Only the password hashes for user tom and cathrine are simple enough to be cracked. I changed the file permissions for their respective private keys

Command: **chmod 600 <SSH private key>**

Command: **ssh tom@192.168.238.107 -i <tom's SSH private key>**

I have managed to gain initial foothold on the machine by SSH into user tom's account using his SSH private key. User cathrine's SSH private key is unable to be used for SSH authentication.

2.4 Privilege Escalation - Enumeration and Exploitation

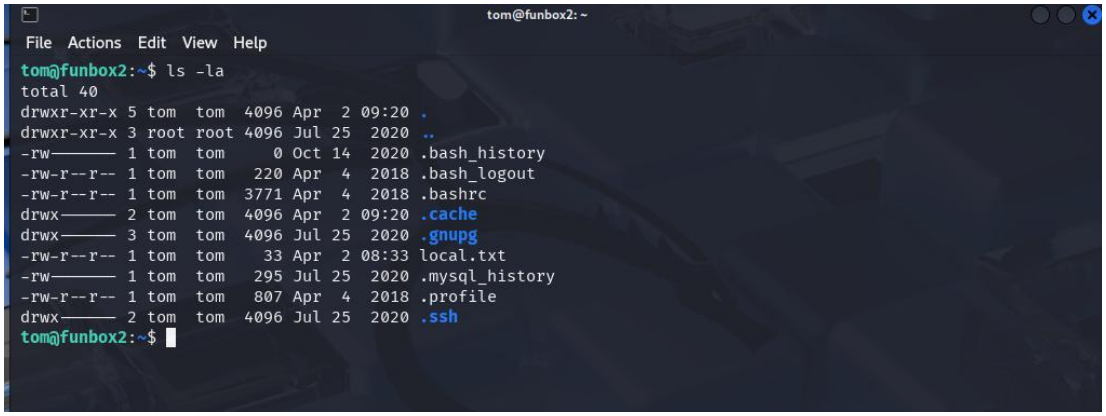


I have started a web server on my Kali Linux host machine so I can migrate pspy32 over to the compromised machine using **wget**. I changed the file permissions for pspy32 so that it can be executed.

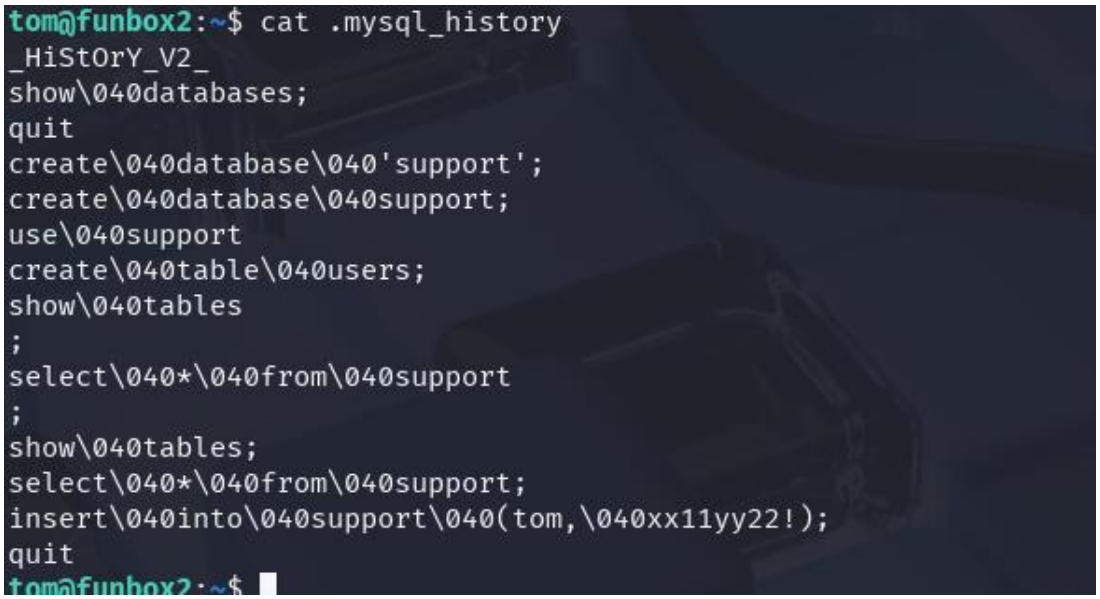
Common Restrictions in rbash include:

i. Cannot change directories using cd.
ii. Cannot set or unset the PATH variable.
iii. Cannot use commands like exec, which can replace the current shell with another command.
iv. Cannot redirect output using >, >>, or 2>&1 in most cases.
v. Cannot run commands with ; (semicolon) or & to run multiple commands in a single line.

In this case, the slash symbol '/' cannot be used in command names due to rbash restriction. Hence, no files or scripts can be executed from within current directory. Pspy and Linpeas cannot be used in this case.



Reading the contents of local.txt grants the user flag. I have listed out all files within current directory to see potential entry points leading to privilege escalation. The files **.bash_history, .mysql_history** and **.ssh** directory are of interest.

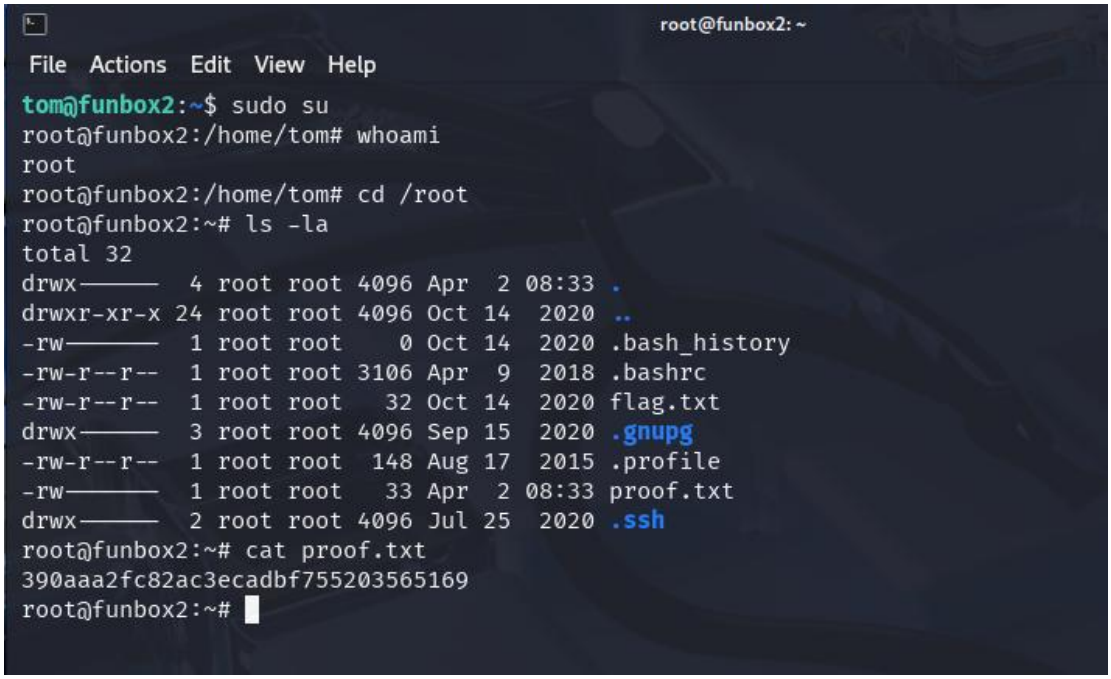


The .**mysql_history f**ile contains the history of all commands executed in the MySQL command-line client. A database named 'support' is created, and a table named 'users' is created within it. The record for the user 'tom' and their password, inserted via the CLI, can be observed.

User	Password
tom	**xx11yy22!**

Checking the sudo privileges for the user 'tom' requires a password, which is obtained from the contents of the .mysql_history file. The user tom can execute all commands with sudo privileges.

The commands that user 'tom' can execute in the current shell can be listed using the command: **compgen -c**



One of the commands that can be executed in the current shell is **su**. Since the user tom can execute any command with sudo, they can directly escalate privileges by switching to the root account without requiring to know the root user's password.

User ‘tom’ can be observed to execute python3 with sudo privileges as well. I executed python3 with sudo privileges to spawn a bash shell using Python’s pty module. It will spawn a new process that runs an interactive /bin/bash shell with root privileges.

2.5 Vulnerability Fix and Severity

Vulnerability 1: Anonymous FTP login

FTP server has poor security configurations in supporting FTP Anonymous login

Fix and Mitigations

- **Restricting access to specific directories that are intended for anonymous usage**
- **Using firewalls and network configurations to restrict who can access the anonymous FTP service**
- **Adopt Data Classification when uploading documents in terms of sensitivity of information**

Vulnerability 2: Exposure of sensitive files

SSH private keys are stored in password protected ZIP files on an easily-accessible FTP server.

Fix and Mitigations

- **Stop using FTP to store sensitive data**
- **Adopt Data Classification when uploading documents in terms of sensitivity of information**
- **Use stronger encryption and protection for SSH keys**
  i. Use a hardware security module (HSM): If possible, store the private keys in an HSM or a hardware token (YubiKey)
  ii. Adopt AES-256 encryption of SSH private keys
  iii. Adopt a more secure storage location instead of ZIP files

Vulnerability 3: Leak of information through history files

Tom’s password is leaked in .mysql_history file

Fix and Mitigations

- **Isolate tom’s device from the network and change their password**
- **Clear the contents of history files to prevent data leaks**

  > ~/.mysql_history

  > ~/.bash_history

Vulnerability 4: Vertical Privilege Escalation to root

The unrestricted sudo permissions in user tom's account allowed the abuse of commands to escalate privilege levels.

Fix and Mitigations

- **Review and correct the sudoers file**
- **Limit Sudo Access**

  Ensure users are only granted **specific, necessary privileges**. Limit what each user can do via sudo and explicitly specify which commands they can run.

- **Restriction of commands executed by sudo**

  The sudoers file can be edited to restrict **command arguments** and **environment variables** that can be passed to commands when they are executed with sudo.

- **Restrict sudo to only trusted users**