Basic Penetration

Organization: TryHackMe	Type: online CTF	
Categories: ✓ Network Security, ☐ Cryptography, ☐ Mobile Applications	☐ Reverse Engineering, ✓ Web Applications, ☐ Forensics	Difficulty: Easy
Name: Kasper Verhulst	Release date:23-07-2019 Completing date:29-12-2022	

Scanning & Reconaissance

First, let us start scanning the machine to see which services are running. As usual, let's start by running an nmap command.

```
nmap -A -sS -p1-1024 $BOX_IP -oN nmap.out
```

We find the following services running on the machine

Port	Service	Version
22/tcp open	SSH	OpenSSH 7.2p2 Ubuntu 4ubuntu2.4
80/tcp open	HTTP	Apache httpd 2.4.18
139/tcp	NetBIOS	
445/tcp	Samba	4.3.11

The nmap also shows the netBIOS name of the host is BASIC2.

Gaining Access

Let's start with examining the web server. We can access the web server on http://BOX_IP. Nothing interesting is shown on the start page or in the source. Let's run gobuster to find other paths on the webserver

```
gobuster dir -x html,php -u http://$BOX_IP -w
    /usr/share/dirbuster/wordlists/apache-user-enum-2.0.txt
```

We find the path /development that allows for some directory listing and we find two notes under this directory. In the note *j.txt*, K warns J he has a poor password. In the note *dev.txt*, both developers are talking about the programs they are installing: Apache httpd server, Apache struts and SMB.

So we already know J has a weak password, but we need to find a username too. Brute-forcing username and password at the same time would simply take too much time. So let us scan the Samba server. There are many different ways to do so but here let us try with smbmap:

```
smbmap -H $BOX_IP -R
```

We see there is a file called *staff.txt* available tu any user. Let's grab this file:

```
smbmap -H $BOX_IP --download ./Anonymous/staff.txt
```

In this file, we can find the usernames for both users: jan and kay.
Another way to have found these usernames would have been:

```
enum4linux $BOX_IP
```

Now since we know that Jan has a weak password, let us try to brute-force it using Hydra.

```
hydra -l jan -P /usr/share/wordlists/rockyou.txt ssh://$BOX_IP -t 4
```

We managed to brute-force our way into the machine using the password **armando**. We can now connect to the machine with

```
ssh jan@$BOX_IP
```

Privilege Escalation

Now that we have access to the machine, we need to further extend our privileges. To start we can try a couple of things and take a look around what we can do. First let's try whether our user can simply use sudo

```
sudo -l
```

Let's check which users there are

```
cat /etc/passwords
```

No luck so far, so let's use an automated privileged escalation script that can help us. Basically they check the machine for one of the following

1. Kernel exploits: exploiting vulnerabilities in the Linux Kernel we can sometimes escalate our privileges
2. Programs running as root: specific service is running as root and you can make that service execute commands you can execute commands as root
3. Installed software Third-party software that might be vulnerable
4. Weak/reused/plaintext passwords
5. Inside service
6. Suid/guid misconfiguration: programs can be run with the privilege of the owner
7. Abusing sudo-rights: access to some programs using sudo
8. World writable scripts invoked by root: If you find a script that is owned by root but is writable by anyone you can add your own malicious code in that script
9. Bad path configuration
10. Cronjobs: With privileges running script that are editable for other users
11. Unmounted filesystems here we are looking for any unmounted filesystems. If we find one we mount it and start the priv-esc process over again

The LinEnum privilege escalation script finds a file */home/jay/pass.bak.*. Furthermore, we see that the binary `vim` has the SUID bit set. This means that the `vim` program is always ru with the privileges of the owner (root in this case) independently of who is calling the binary. Hence, even though the user *jan* doesn't have read access to the file */home/jay/pass.bak.*, he can read the content using the `vim` program. We can find the final flag:

```
vim /home/jay/pass.bak
```

Alternatively, we could have found the private key from Kay. The file /home/user/jay/.ssh/id_rsa was world readable. You can scp the private key to the local machine and use it to authenticate to the server as Kay.

```
ssh kay@$BOX_IP -i id_rsa
```

The private key is password protected, but the password can be cracked using John the Ripper:

```
/usr/share/john/ssh2john.py id_rsa | john --wordlist /usr/share/wordlists/rockyou.txt
```