HGAME week2 write up

队伍名称：re35T 队伍id：#000026

CRYPTO

Ancient Recall

读代码先还原异或后的id,然后根据正负异或还原得到id

```
Major_Arcana = ["The Fool", "The Magician", "The High Priestess","The Empress",
"The Emperor", "The Hierophant","The Lovers", "The Chariot", "Strength","The
Hermit", "Wheel of Fortune", "Justice","The Hanged Man", "Death",
"Temperance","The Devil", "The Tower", "The Star","The Moon", "The Sun",
"Judgement","The World"]
wands = ["Ace of Wands", "Two of Wands", "Three of Wands", "Four of Wands", "Five
of Wands", "Six of Wands", "Seven of Wands", "Eight of Wands", "Nine of Wands",
"Ten of Wands", "Page of Wands", "Knight of Wands", "Queen of Wands", "King of
Wands"]
cups = ["Ace of Cups", "Two of Cups", "Three of Cups", "Four of Cups", "Five of
Cups", "Six of Cups", "Seven of Cups", "Eight of Cups", "Nine of Cups", "Ten of
Cups", "Page of Cups", "Knight of Cups", "Queen of Cups", "King of Cups"]
swords = ["Ace of Swords", "Two of Swords", "Three of Swords", "Four of Swords",
"Five of Swords", "Six of Swords", "Seven of Swords", "Eight of Swords", "Nine of
Swords", "Ten of Swords", "Page of Swords", "Knight of Swords", "Queen of
Swords", "King of Swords"]
pentacles = ["Ace of Pentacles", "Two of Pentacles", "Three of Pentacles", "Four
of Pentacles", "Five of Pentacles", "Six of Pentacles", "Seven of Pentacles",
"Eight of Pentacles", "Nine of Pentacles", "Ten of Pentacles", "Page of
Pentacles", "Knight of Pentacles", "Queen of Pentacles", "King of Pentacles"]
Minor_Arcana = wands + cups + swords + pentacles
tarot = Major_Arcana + Minor_Arcana

value =
[25329519520662917748904983691141959172407947049182105205710670852311474675019,
25329519520662917748903276660741003578980230131054431788812947003815097952 70,
25329519520662917748905544592872766049031303158592585441730683769670723357 30,
25329519520662917748908653282415328853915101626115345140144091742842991390 15,
25329519520662917748908306626081341560179463763099899341758339139211426093 34]


def restore(a):
    s = (a[0]+a[1]+a[2]+a[3]+a[4])//2
    b = [0]*5
    b[0]=s-a[1]-a[3]
    b[1]=s-a[2]-a[4]
    b[2]=s-a[0]-a[3]
    b[3]=s-a[4]-a[1]
    b[4]=s-a[0]-a[2]
    return b

for _ in range(250):
    value = restore(value)
print(value)
```

```
res=[]

for i in range(len(value)):
    if value[i]<0:
        value[i] = value[i] ^ (-1)
        res.append("re-"+tarot[value[i]])
    else:
        value[i] = value[i] ^ 0
        res.append(tarot[value[i]])

print(res)

FLAG=("hgame{"+"&".join(res)+"}").replace(" ","_")
print(FLAG)
```

Intergalactic Bound

根据THcurve信息查到是一种特殊曲线，方程为：

$$ax^3 + y^3 + 1 = dxy$$

然后现在未知a和d,通过G和Q的信息可以得到a和d二元一次方程组，解得a和d

然后搜索找到羊城杯2024-TH_curve题目，实现从TH curve到Weierstrass型曲线的转换，利用已有代码映射完剩下解决dlp问题

```
from sage.all import *
import gmpy2
import libnum
import hashlib
from Crypto.Cipher import AES
from Crypto.Util.Padding import pad
# 定义质数 p
p = 55099055368053948610276786301  # 替换为实际的质数 p

Gx,Gy = 196634467629629276330379267 40, 350744124309156560717770153 20
Qx,Qy = 268051376735366358258844330180, 263768331126093094759511868 83



A = (Qx**3 - Qx * Qy * Gx**3 * inverse_mod(Gx * Gy, p)) % p
B = (Qx * Qy * (Gy**3 + 1) * inverse_mod(Gx * Gy, p) - Qy**3 - 1) % p

a = (inverse_mod(A, p) * B) % p

d = (a*Gx**3+Gy**3+1)%p*inverse_mod(Gx*Gy,p)%p


a0 = 1
a1 = - 3 *(d/3) / (a - (d/3) *(d/3) *(d/3))
a3 = - 9 / ((a - (d/3)* (d/3) *(d/3)) *(a - (d/3) *(d/3)* (d/3)))
a2 = - 9 *(d/3) *(d/3) / ((a - (d/3) *(d/3) *(d/3)) * (a - (d/3)* (d/3)* (d/3)))
```

```
a4 = - 27 *(d/3) / ((a - (d/3)* (d/3)* (d/3))* (a - (d/3) *(d/3) *(d/3)) *(a -
(d/3)* (d/3)* (d/3)))
a6 = - 27 / ((a - (d/3) *(d/3)* (d/3)) *(a - (d/3) *(d/3) *(d/3))* (a - (d/3) *
(d/3)* (d/3)) *(a - (d/3)* (d/3)* (d/3)))

#print(d)
#print(a0,a1,a2,a4,a6)

E = EllipticCurve(GF(p), [a1, a2, a3, a4, a6])

#u = (-3 / (a - d* d *d/27)) *x / (d *x/3 - (-y) + 1)
#v = (-9 / ((a - d *d *d/27) *(a - d* d *d/27))) (-y) / (d* x/3 - (-y) + 1)

gx =(-3 / (a - d* d *d/27)) *Gx / (d *Gx/3 - (-Gy) + 1)
gy =(-9 / ((a - d *d *d/27) *(a - d* d *d/27))) *(-Gy) / (d* Gx/3 - (-Gy) + 1)
qx =(-3 / (a - d* d *d/27)) *Qx / (d *Qx/3 - (-Qy) + 1)
qy =(-9 / ((a - d *d *d/27) *(a - d* d *d/27))) *(-Qy) / (d* Qx/3 - (-Qy) + 1)

G = E(gx,gy)
Q = E(qx,qy)

x = discrete_log(Q,G,operation='+')
print(x)

key = hashlib.sha256(str(x).encode()).digest()
print(key)


cipher = AES.new(key, AES.MODE_ECB)

ciphertext=b"k\xe8\xbe\x94\x9e\xfc\xe2\x9e\x97\xe5\xf3\x04'\x8f\xb2\x01T\x06\x88\
x04\xeb3Jl\xdd Pk$\x00:\xf5"
flag = cipher.decrypt(ciphertext)
print(f"ciphertext={flag}")
```

CRYPTO

Computer cleaner plus

打开虚拟机开始检查，在执行ps指令查看进程时发现/bin/ps存在，但root账户仍然不可执行感觉可疑，修改权限执行发现B4ck_D0_oR.elf 当作flag提交成功

WEB

Level 21096 HoneyPot

代码审计发现可疑部分

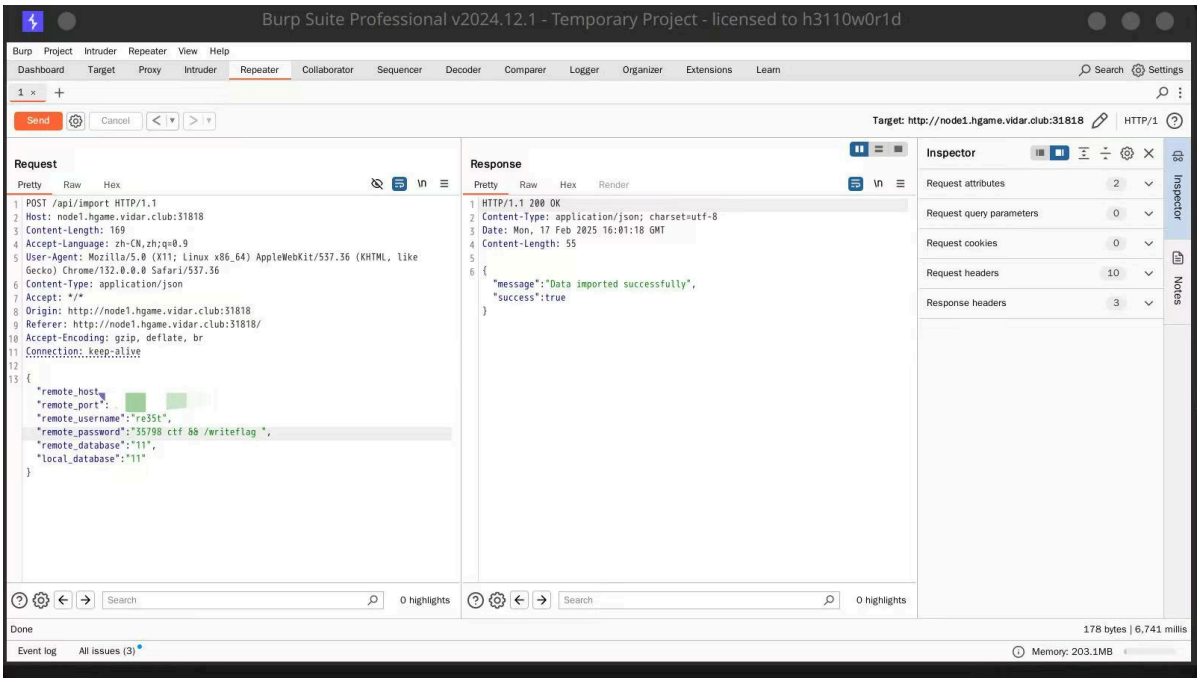
```
//Never able to inject shell commands,Hackers can't use this,HaHa

    command := fmt.Sprintf("/usr/local/bin/mysqldump -h %s -u %s -p%s %s
|/usr/local/bin/mysql -h 127.0.0.1 -u %s -p%s %s",
        config.RemoteHost,
        config.RemoteUsername,
        config.RemotePassword,
```

```
        config.RemoteDatabase,
        localConfig.Username,
        localConfig.Password,
        config.LocalDatabase,
    )
    fmt.Println(command)
    cmd := exec.Command("sh", "-c", command)
    if err := cmd.Run(); err != nil {
        c.JSON(http.StatusInternalServerError, gin.H{
            "success": false,
            "message": "Failed to import data: " + err.Error(),
        })
        return
    }
```

hint得到利用CVE-2024-21096:Mysqldump命令注入漏洞，可以注入实现rce但是有过滤

```
func validateImportConfig(config ImportConfig) error {
    if config.RemoteHost == "" ||
        config.RemoteUsername == "" ||
        config.RemoteDatabase == "" ||
        config.LocalDatabase == "" {
        return fmt.Errorf("missing required fields")
    }

    if match, _ := regexp.MatchString(`^[a-zA-Z0-9\.\-]+$`, config.RemoteHost);
!match {
        return fmt.Errorf("invalid remote host")
    }

    if match, _ := regexp.MatchString(`^[a-zA-Z0-9_]+$`, config.RemoteUsername);
!match {
        return fmt.Errorf("invalid remote username")
    }

    if match, _ := regexp.MatchString(`^[a-zA-Z0-9_]+$`, config.RemoteDatabase);
!match {
        return fmt.Errorf("invalid remote database name")
    }

    if match, _ := regexp.MatchString(`^[a-zA-Z0-9_]+$`, config.LocalDatabase);
!match {
        return fmt.Errorf("invalid local database name")
    }

    return nil
}
```

有过滤但是password没有检查，于是注入构造payload



然后访问/flag就得到了flag

hgame{e0ca38e8-d3f9-9b32-7202-37b312957dbe}