【PHP代码审计】入门之路——第二篇-宽字节注入

phpoop / 2017-12-05 16:16:00 / 浏览数 3047 新手 入门资料 顶(0) 踩(0)

0x00 简介

首先先在这里说一声不好意思，时隔几个月没更新了，这个漏洞笔者认为按照现在的年代来说，价值不高了，但是本着努力学习积极提升自我的的观念，我认为我们还是应该
测试代码与sql：链接：http://pan.baidu.com/s/1dFiw2HV 密码：aabx

0x01概述

首先我们了解下宽字节注入，宽字节注入主要是源于程序员设置数据库编码与PHP编码设置为不同的两个编码那么就有可能产生宽字节注入

例如说PHP的编码为 UTF-8而MySql的编码设置为了
SET NAMES 'gbk'或是 SET character_set_client =gbk，这样配置会引发编码转换从而导致的注入漏洞。
这里要说明一小点的是：
SET NAMES 'x'语句与这三个语句等价：

```
mysql>SET character_set_client =x;
mysql>SET character_set_results =x;
mysql>SET character_set_connection =x;
```

也就是说你设置了 SET NAMES 'x' 时就等于同时执行了上面的3条语句

而我认为的宽字节注入就是PHP发送请求到MySql时使用了语句
SET NAMES 'gbk' 或是SET character_set_client =gbk 进行了一次编码，但是又由于一些不经意的字符集转换导致了宽字节注入

0x02 宽字节注入原理

1,在我们正常情况下使用addslashes函数或是开启PHPGPC（注：■php5.4■■■■■，并且需要说明特别说明一点，GPC无法过滤$_SERVER提交的参数）时过滤GET、P
提交的参数时，黑客们使用的预定义字符会给转义成添加反斜杠的字符串如下面的例子
例子：

```
■■■■'■= ■\'■
■■■■"■ = ■\"■
■■■■\■ = ■\\■
```

2,假如这个网站有宽字节注入那么我们提交：

```
http://127.0.0.1/unicodeSqlTest?id=%df%27
```

这时,假如我们现在使用的是addslashes来过滤,那么就会发生如下的转换过程
例子：

```
%df%27===(addslashes)===>%df%5c%27===(■■■GBK)===>■'
```

这里可能有一些人没看懂，我可以粗略的解释一下。
前端输入%df%27时首先经过上面addslashes函数转义变成了%df%5c%27（%5c■■■■\），之后在数据库查询前因为设置了GBK编码，即是在汉字编码范围内两个字节都会

干这样看我们可能也没能很清楚的看懂，我们可以来几个例子：

例子1：
在PHP中使用$pdo->query('set names
gbk');指定三个字符集（客户端、连接层、结果集）都是GBK编码。而PHP的编码等于UTF-8编码时造成的宽字节注入
例子代码：

那么如何逃过addslashes的限制呢？addslashes函数产生的效果就是，让'变成\'，让单双引号变得不再是'单双引号'，只是一撇而已。一般绕过方式就是，想办法处理

1.想办法给\ 前面再加一个\，变成\\'，这样\被转义了， ' 逃出了限制
2.想办法把\弄没有。
我们这里的宽字节注入是利用mysql的一个特性，mysql在使用GBK编码的时候，会认为两个字符是一个汉字（前一个ascii码要大于128，才到汉字的范围）。根据这个我们

我们可以看到，页面已经报错了。看到报错，说明这句sql语句出错，说明我们已经绕过了addslashes那么就可以正常的进行注入了。
我们只是在%27前面加了一个%df为什么就报错了？而且从上图中可以看到，报错的原因是多了一个单引号，而单引号前面的反斜杠已经不见了。
这就是mysql的特性，因为gbk是多字节编码，他认为两个字节代表一个汉字，所以%df和后面的\也就是%5c变成了一个汉字■，而'逃逸了出来,导致了注入。

例子2：

使用set names
UTF-8指定了UTF-8字符集，并且也使用转义函数进行转义。有时候在程序运行的时候，为了避免乱码，会将一些用户提交的GBK字符使用iconv函数（或mb_convert_en

例子代码：

转换过程：

```
%df%27===(addslashes)===>%df%5c%27===(iconv)===>%e5%5c%5c%27
```

`$id =iconv('GBK','UTF-8', $id)`;如果内容是utf8编码的，将自动转成gbk编码的. ■的`utf-8`编码是`0xe98ca6`，它的gbk编码是`0xe55c`。
有的同学可能就领悟了。`\`的`ascii`码正是`5c`。那么，当我们的■被`iconv`从`utf-8`转换成`gbk`后，变成了`%e5%5c`，而后面的`'`被`addslashes`变成了`%5c%27`，这样组合起

从上面的介绍中可以看出，宽字节注入的关键点有两个：
（1）需要将数据库编码与PHP编码设置为不同的两个编码那么就有可能产生宽字节注入；
（2）设置的宽字符集可能吃掉转义符号\（对应的编码为0x5c，即低位中包含正常的0x5c就行了）。

上一篇：企业安全项目架构实践分享 下一篇：【PHP代码审计】——黑白盒导图

1. 1 条回复


komas 2017-12-07 18:00:02

常规的审计文章太多了，需要来点框架审计正能量啊

0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录