**Command Injection**

**命令注入**

命令注入通常因为指Web应用在服务器上拼接系统命令而造成的漏洞。

该类漏洞通常出现在调用外部程序完成一些功能的情景下。比如一些Web管理界面的配置主机名/IP/掩码/网关、查看系统信息以及关闭重启等功能，或者一些站点提供如ping、nslookup、提供发送邮件、转换图片等功能都可能出现该类漏洞。

**判断命令注入流程**

是否调用系统命令

函数或函数的参数是否可控

是否拼接注入命令

**命令注入符号集**

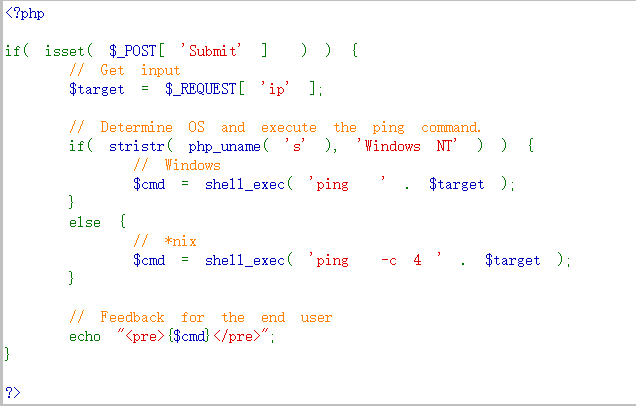
“&”也可以说是and符特性是可以前一句命令可以和后一句命令一起执行”

“&&”连接符只有前一句命令可以执行后一句命令才能执行

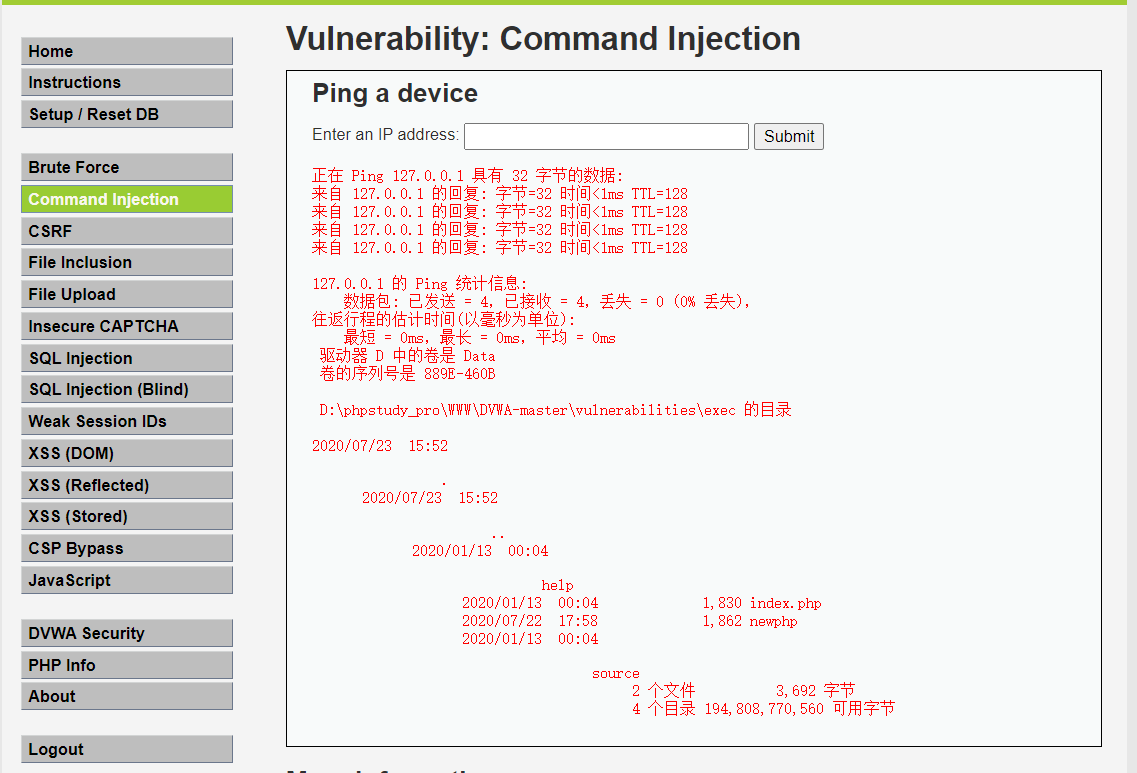
“|“管道符将前一句命令的输出作为后一句的输入，只打印后一句的输出

“||“前一句命令失败了才执行后一句

# Level:Low

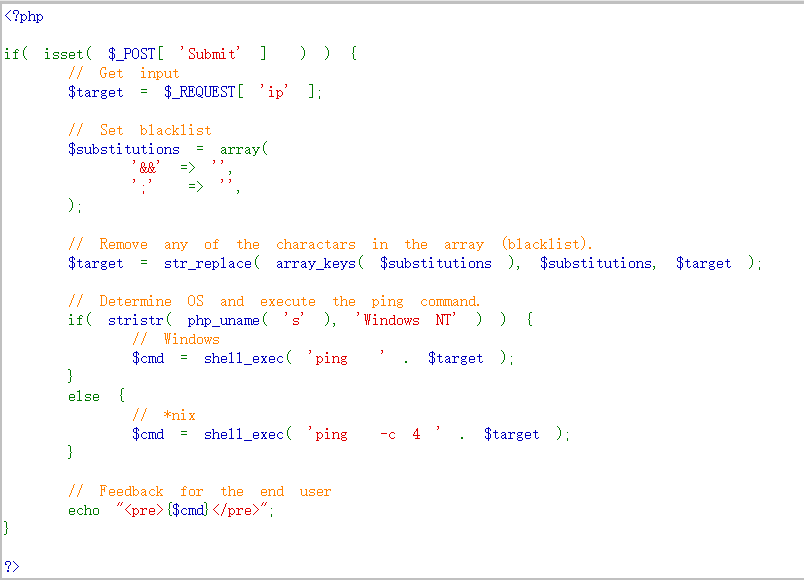


没有对命令注入符进行过滤



根据Low级别的源码得知并没有对命令字符进行过滤，可以直接用127.0.0.1 && dir输入命令拼接就可以进行命令注入

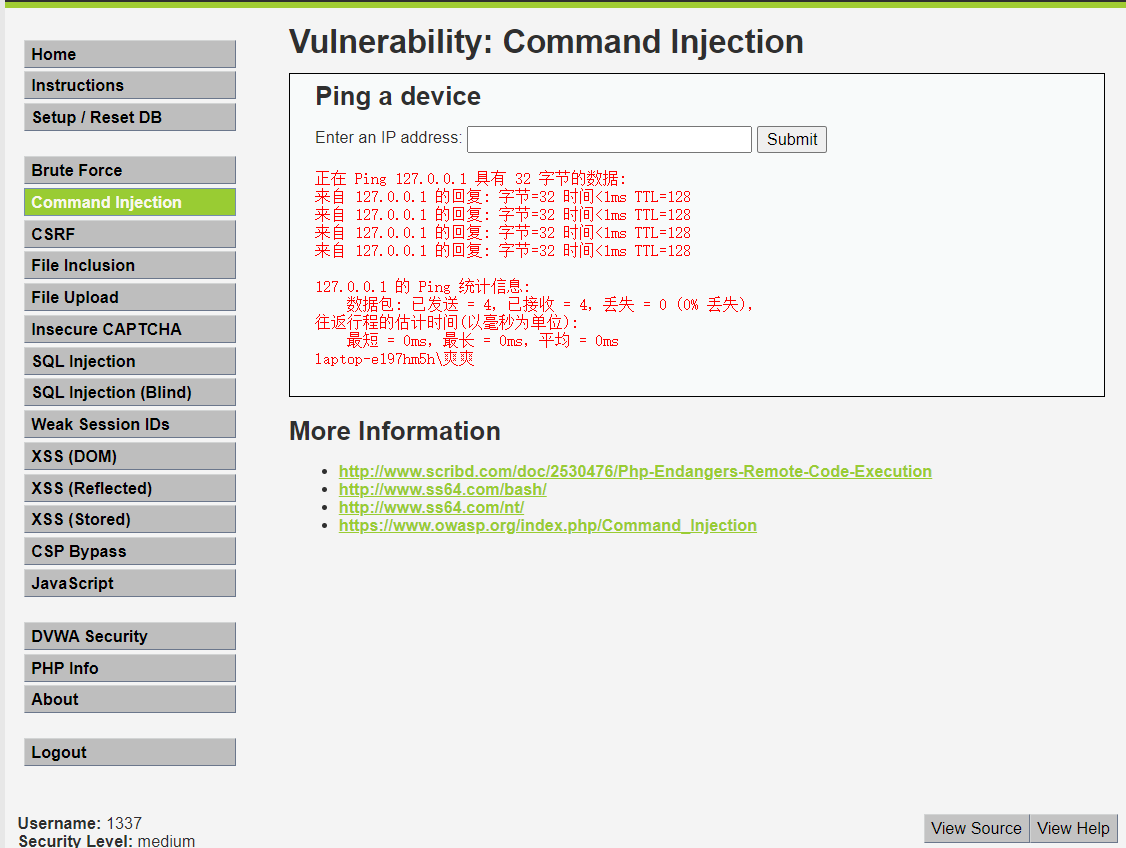
# Level: Medium



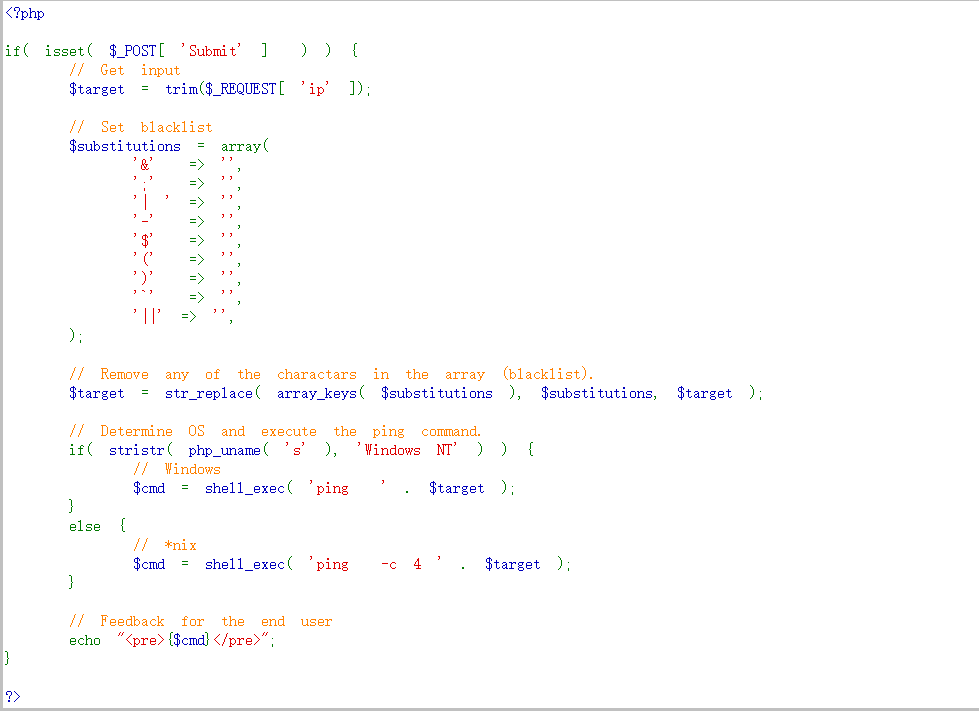
**str\_replace — 子字符串替换**

Medium的源码进行了一些字符过滤，输入的‘&&’和“；“字符会被替换成空置

这次可以用”&”字符来进行注入， 127.0.0.1 & whoami

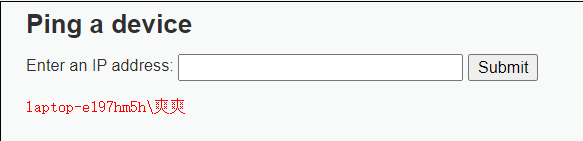


# Level：High



High这一次一些单字符被过滤了，但是源码漏掉了”|”这个字符没有过滤，所以这一次可以用“|“字符进行注入

127.0.0.1 |whoami



# Level:impossible



Impossible多了4个函数1个变量数组:

$\_SESSION[]

变量的数组.

Stripslashes()

反引用一个引用字符串

explode()

使用一个字符串分割另一个字符串

is\_numeric()

检测变量是否为数字或数字字符串

Sizeof()

[count()](https://www.php.net/manual/zh/function.count.php) 的别名 count()是计算数组中的单元数目，或对象中的属性个数