WEB安全

常规漏洞

WEB安全我把它分两类：

- 常规WEB漏洞
- 业务逻辑漏洞

常规WEB漏洞还有哪些？

1、SQL注入

2、XSS

3、代码注入

3、命令执行

4、本地文件包含

5、远程文件包含

6、跨站请求伪造(CSRF)攻击

7、服务端请求伪造(SSRF)攻击

8、更多...

SQL注入

1、什么是SQL注入？

在参数中加入SQL语法，改变原来的SQL结构，达到我们编写的程序时意料之外结果的一种

SQL注入

2、SQL注入的类型

可显注入：攻击者可以直接在当前界面内容中获取想要获得的内容

乐视云官方接口泄漏(账户信息接口含密码&SQL注入)

报错注入：数据库查询返回结果并没有在页面中显示，但是应用程序将数据库报错信息打印到了页面中，所以攻击者可以构造数据库报错语句，从报错信息中获取想要获得的内容

17173游戏某站点MySQL报错注入(不带逗号注入的精简过程)

盲注：数据库查询结果无法从直观页面中获取，攻击者通过使用数据库逻辑或使数据库库执行延时等方法获取想要获得的内容

淘宝网某站点存在MySQL注射(附验证脚本)

SQL注入

3、SQL注入的防范

1、接收到的参数一定要检查，拦截带有SQL语法的参数传入到程序

2、使用预编译(PDO)的处理方式处理拼接了用户参数的SQL语句

3、定期查看数据库执行日志，查看有没有正常逻辑之外的SQL语句执行了

XSS

1、什么是xss?

2、XSS可以导致那些危害?

3、XSS有哪几种类型?

4、面对XSS攻击我们有哪些防范措施?

1、什么是XSS?

恶意攻击者往WEB页面里插入恶意HTML代码(一般是JavaScript的脚步)当用户浏览该页之时，嵌入其中Web里面的HTML代码会被执行，从而达到恶意攻击用户的特殊目的。

2、XSS可以导致的危害

1 XSS常与CSRF漏洞结合起来使用，可在用户不知不觉中用用户的账号进行转账，加关注等操作

2 其它一些少见的利用方式包括利用XSS进行DDOS，

SOHU视频XSS漏洞导致其用户成为DDOS肉鸡

3、常见的几种XSS

1. 反射型XSS，非永久化，需要欺骗用户自己去点击链接才能触发XSS代码（服务器中没有这样的页面和内容）。

2. 存储型XSS，永久化，代码是存储在服务器中的，如在个人信息或发表文章等地方，加入代码，如果没有过滤或过滤不严，那么这些代码将储存到服务器中，用户访问该页面的时候触发代码执行。

3. DOM型XSS，DOM—based XSS漏洞是基于文档对象模型Document Object Model，DOM)的一种漏洞

3.1反射型XSS案例

1. 新浪微博存在多处反射型XSS漏洞（FIREFOX、CHROME均可触发）
2. 百度地图持久形XSS漏洞
3. 金山逍遥存在SQL注射等安全问题

3.2存储型XSS案例

腾讯群空间存储型XSS

3.3DOM型XSS案例

淘宝主域名下多处Dom XSS

腾讯微博一处两用DOM-XSS，能反射，能后门

4、XSS注入的防范

1　对XSS的防御需要根据实际情况对用户的输入进行严格的过滤

2　在服务端设置COOKIE中加入HTTPONLY属性可以防止JAVASCRIPT获取COOKIE

代码注入

```
eval("\$ret = $data;");
eval("\$ret = deal('$data');");
eval("\$ret = deal("$data");");
```

例子

```
<?php
$string = "beautiful";
$time = "winter";

$str = 'This is a $string $time morning!';
echo $str. "<br />";

eval("\$str = \"$str\";");
echo $str;
?>
```

输出:

```
This is a $string $time morning!
This is a beautiful winter morning!
```

1、什么是代码注入?

调用一些能将字符串转化成代码的函数（如PHP中的EVAL、ASSERT ）时没有考虑用户是否能控制这个字符串，就有可能造成代码注入漏洞

代码注入

2、代码注入的案例和危害

PHPCMS前台任意代码执行（有PHP版本限制）

PHPCMS V9 后台远程代码执行漏洞（第三弹）

51CTO技术网站存在PHP代码注入可WEBSHELL

代码注入

3、代码注入的防范

1 尽量使用JSON保存数组、对象就使用JSON，不要把PHP对象保存成字符串，不然否则读取的时候需要使用EVAL

2 如果必须使用EVAL的情况，一定要保证用户不能轻易接触EVAL的参数（或用正则严格判断输入的数据格式）

总结

1. 过滤入口参数
2. 检查执行参数
3. 屏蔽敏感返回值