启航科技开发公司技术沙龙
Web安全专题

主讲：孙子谦

ubuntu

- SQL Injection
- XSS
- 文件上传
- 伪造请求

ubuntu

SQL Injection

原因：相当大一部分程序员在编写代码的时候，没有对用户输入数据的合法性进行判断，使应用程序存在安全隐患。用户可以提交一段数据库查询代码，根据程序返回的结果，获得某些他想得知的数据，这就是所谓的SQL Injection，即SQL注入。

看下面一段代码

```
<?php
    if(isset($_POST[‘feedback’]) {
        $dbc=@mysql_connect(‘127.0.0.1’,’DB
    _USER’,’DB_PSWD’) OR die ('Could not
    connect to MySQL');
        mysql_select_db(‘DB_NAME’);
        mysql_query("SET NAMES UTF8");
        $sql='insert into feedback (text)
    values(‘.$_POST[‘feedback’].’)‘;
        $res=mysql_query ($sql);
    //一个html
    } else { //一个表单 }
?>
```

后果：一旦有人在feedback这个文本框中恶意输入字符，就会拼接到sql语句中带来无法预料的结果。。。

例如输入 “’); drop table users;”
则原SQL语句将成为
insert into feedback (text) values(‘’);drop table users;....

此时，users表。。。。你懂得

解决办法：

如果能配置服务器的运行环境，可以修改magic_quotes_gpc设置为on并过滤部分字符

如果没有服务器权限，可以**get_magic_quotes_gpc。如果为on则只需要过滤部分特殊字符，如果为off则需要过滤全部特殊字符。**

可以用htmlentities()或者htmlspecialchars等函数把字符转换为 HTML 实体。

XSS

XSS又叫Cross Site Script，意为跨站脚本攻击。恶意攻击者往Web页面里插入恶意html代码，当用户浏览该页之时，嵌入其中Web里面的html代码会被执行，从而达到恶意用户的特殊目的。

假如我的src设定为一个恶意脚本。。。假如我的height和width均为0。。。

一旦用户访问到该网址，会自动调用远程端恶意脚本。而宽、高均为0，用户丝毫看不到任何调用。。。

此时，黑客可以通过恶意脚本挂马等。。。而用户最终感受到的是访问我们的网址时被挂马了，对我们的网站信誉有不良影响。

如果是用户通过一定方式将这种html脚本写入了数据库，则每次查看相应网页时都会直接显示相应的内容。这样影响到的人会更多。。

启航科技 QIHANG

解决方法：

入库和输出时过滤html标记

（这个我在信息学院新版网站里面确实没做处理。）

ubuntu

文件上传

上传恶意脚本

上传特意编制的有恶意功能的图片

（后者暂时没有完美的解决办法）

上传恶意脚本包含可执行脚本（Windows下的exe、com、bat、cmd文件，Linux下的sh文件等）可以通过过滤上传文件的扩展名，或者上传文件后更改（或者删除）扩展名来一定程度上预防。

建议上传的文件存储到不能通过浏览器直接访问的文件夹内，并编写专用的下载（或者加载）脚本来处理文件的下载。

例如

- Somedir
  - webdir
    - sondir(html or php etc.)
  - anotherDir
  - ...

有时候你无权访问其他目录，那样建议采用更改扩展名或者限制扩展名的方式。

Linux+PHP时还有一个办法：chmod函数，把上传的文件权限改为444（r--r--r--，只读）防止被系统执行（但是不能防止通过web搞破坏）

稍微高级一点的入侵：

伪造访问请求

（这还不是最高级的入侵。。）

伪造get请求

伪造html表单

伪造cookies

伪造post请求

伪造get请求

这种是最容易实现的，也是黑客们最常用的。就是把url里的参数手动修改掉。处理方案就是严格过滤通过get方式取得的内容。
例如XSS示例中的cid值

伪造html表单

这个也很容易实现。将表单另存为，然后用记事本打开，按需修改后就可以直接用了。

这类问题可以通过formhash、验证码等技术处理（不做详细讲解）

伪造cookies

COOKIE常用来标示用户的登陆状态等信息，黑客也没有放弃对它的利用。如果在cookie里存储了权限等级等信息，黑客可以很容易通过firebug等工具修改权限从而达到更高目的。

解决办法：使用服务器端技术Session来存储权限等用户状态信息

伪造post请求

这个算是比较复杂的了。此类黑客大多数谙熟HTTP协议甚至是TCP/IP协议。通常使用curl等类似方式模拟整个post请求。

此类黑客比较难预防。首要的是通过formhash等技术来过滤虚假表单或者请求。然后还要在必要的地方添加验证码等措施避免黑客通过简单的客户端事件来模拟post请求

以上介绍的是部分主流的web攻击方式。黑客的攻击还远远不止这些。有些如缓冲区溢出、DoS等以上没有列出。

构造安全的Web服务需要我们认真处理每一个细节。

谢谢观看
Thank you./Merci.