doi: 10.7690/bgzdh.2015.08.011

SQL 自动注入攻击框架研究与设计

李洪敏[1]，黄晓芳[2]，张建平[1]
（1. 中国工程物理研究院总体工程研究所，四川 绵阳 621900；
2. 西南科技大学计算机科学与技术学院，四川 绵阳 621000）

摘要：针对目前 SQL 注入攻击检测手段单一化，不能做网站全面自动分析的问题，构建一种 SQL 注入的自动化检测框架。在分析 SQL 注入的基本原理及常见攻击方式基础上，设计了自动注入攻击工具总体框架，以及 2 条识别和抓取目标 Web 系统的链接。测试结果表明：该工具能够自动扫描识别 2 种以上类型的 SQL 注入攻击，并可对数据库进行攻击检测。该工具还可以进行进一步拓展，以满足灵活的 SQL 注入变形攻击的需求。

关键词：SQL 注入；安全性；攻击；自动检测
中图分类号：TP393　　**文献标志码：**A

Research and Design of Framework to SQL Automatic Injection Attack

Li Hongmin[1], Huang Xiaofang[2], Zhan Jianping[1]
(1. *Institute of System Engineering*, *China Academy of Engineering Physics*, *Mianyang* 621900, *China*;
2. *School of Computer Science & Technology*, *Southwest University of Science & Technology*, *Mianyang* 621000, *China*)

Abstract: The detection mean of SQL injection attack is single and it can't carry out website automatic analysis, establish SQL injection automatic detection framework. Based on analyzing SQL injection basic principle and attack means, design automatic attack injection tool general framework, and 2 diagnosis and catching target Web system linkages. The test results show that the tool can automatically scanning more than 2 SQL injection attack types, then carry out attack detection for database. The tool can be further expanding for requirements of SQL flexible injection deformation attack.

Keywords: SQL injection; security; attack; automatic detection

0　引言

随着 Web 应用的发展越来越成熟，应用的各项技术发展得也越来越复杂。网络安全成了业界关注的热点。据开放式 Web 应用程序安全项目[1](open Web application security project，OWASP)每隔 3 年更新一次的“十大安全隐患列表”，近几次公布的总结 Web 应用程序最可能、最常见、最危险的十大安全隐患中，SQL 注入攻击一直排列靠前。在以 B/S 模式构建的 Web 应用程序中，如果不对网络用户提交的数据进行合法性判断，很容易导致 SQL 注入攻击。由于 SQL 注入是从正常的 Web 平台入口进行请求访问，和请求正常的 Web 页面没有什么区别，所以目前市面上除了专门的 Web 防御防火墙外，大部分防火墙都不会对这类 SQL 注入攻击进行监控和发出警报，若网站管理员没有经常查看 IIS 日志的经验和习惯，有可能被入侵很长一段时间后都没有任何发觉。因此，笔者针对目前 SQL 注入攻击检测手段单一化，不能做网站全面自动分析的弱点，在研究 SQL 注入攻击的关键技术基础上，构建 SQL 注入的自动化检测框架。

1　SQL 注入攻击分析

1.1　注入攻击常见方式

SQL 注入是一种源于 SQL 语句的漏洞，利用应用程序对输入数据的检验不足和程序自身对变量处理不当造成的，应用程序通常有表单输入、URL 参数重写、修改 Cookies 文件等方式接受用户输入数据。为了防止 SQL 注入攻击，通常会在 Web 应用的代码层中或通过第三方的外部实现，如 Web 应用防火墙(WAF)或入侵防御系统(IPS)中采用关键词过滤的方式，如 select|delete|update|count|等；以及代码层级别的过滤器，使用的方法包括编码、消除敏感字符等等各种方法，用来过滤掉畸形的数据串。但是，目前对这种过滤机制也有相应的 SQL 注入攻击技术绕过该机制，常见方法有：大小写变换、使用编码的方式、构造畸形数据包的方式、使用空字节绕过过滤机制、构造超长数据包的方式等[2]。

1.2　注入攻击流程

在常规的 SQL 注入攻击基础上，二阶 SQL 注入攻击是另外一种高级攻击手法，属于一种更细微

收稿日期：2015-03-22；修回日期：2015-05-04
作者简介：李洪敏(1968—)，女，辽宁人，硕士，高级工程师，从事网络信息安全研究。

的漏洞，通常更难被检查到[3]。

该方法攻击过程如下所示：

1) 攻击者在 HTTP 请求中插入了精心构造的参数；

2) 应用服务器存储该输入的参数准备处理；

3) 攻击者进行第 2 次 HTTP 请求，同时插入了精心构造的参数；

4) 应用服务器将处理第 2 次请求，应用服务器将会检索已存储的参数并处理，因此导致了 SQL 查询被执行；

5) 应用服务器有可能会对第 2 次请求的参数响应返回输出结果。

以在应用系统中修改已存在用户信息为例说明。用户应用系统将会先使用 SELECT 检索该联系人的信息，并保存到内存中；然后使用用户提供的新信息更新相关数据，并在此对该输入中的引号进行重新编码，而用户没有更新的数据项将在内存中将保持不变；最后使用 UPDATE 语句将内存中的数据回写到数据中。如图 1 所示。

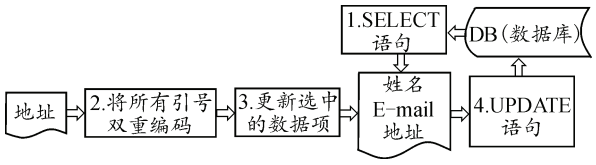

图 1 用户数据更新流程

上例中，引号双重编码可以有效地防止常规的 SQL 注入，但仍然易受到二阶 SQL 注入攻击。若攻击者想利用该漏洞，首先需要使用某个字段中的攻击 payload 创建一个联系人。现假设数据库系统是 Microsoft SQL Server，输入中的引号被双重编码，最终 INSERT 语句如下所示，联系人的姓名被安全地保存到数据库中了。

INSERT INTO webContacts VALUES ('a''+@@version+''a','test@ test.org',…)

接下来更新创建联系人，只需要提供一个新值即可。

应用首先检索已存在联系人的信息：

SELECT * RROM webUsers where contactid=123

检索出来的信息保存在内存中。应用系统使用新提供的数据值替换内存中的值，并将引号双重编码。然后执行 UPDATE 语句，将信息保存到数据库中：

UPDATE webUsers

SET

name='a'+@@version+'a',address='mianyang test ',...

WHERE contactid = 123

到上面这个语句后，攻击已成功执行并修改了应用的查询。当查看更新过的联系人的细节信息时，显示出来的是：

Name:aMicrosoft SQL Server ....Copyright (c)....(Build 2600:)a

Address:mianyang test

应用系统成功地执行了攻击者构造的 SQL 语句，这就证明该应用存在 SQL 注入攻击漏洞，并且可被攻击利用。

2 自动注入攻击工具框架设计

2.1 设计思路

笔者根据 SQL 注入漏洞的利用过程，构建 SQL 注入的自动化检测及注入攻击利用框架，能够自动扫描识别 2 种以上类型的 SQL 注入攻击；至少具有对 4 种类型的数据库进行进行攻击检测，能够对数据库的结构信息进行猜解，能对数据库进行数据操作，能一定程度上自定义注入攻击以绕过简单防注入机制[4-7]。

2.2 自动注入攻击工具框架总体设计

本系统分为 3 个主体部分：自动抓取目标 Web 网站链接；检测是否存在 SQL 注入漏洞；进行 SQL 注入攻击[8-10]。自动注入攻击工具框架总体架构图如图 2 所示。

2.3 遍历链接设计

对于识别和抓取目标 Web 系统的链接，笔者设计有 2 条路线：一是通过 Web 爬虫实现对于 Web 系统进行链接爬行和抓取，然后判断是否存在 SQL 注入；二是通过搜索引擎收录的目标 Web 链接，将之抓取过来进行注入分析。

1) 解析 URL。

通过 Python 的 urlparse 模块可以很轻松地分解 URL 并重新组装。当解析了 URL 链接后，更重要的是分析页面的内容，因为页面中的内容有可能还包含着该 Web 网站的其他链接，这样通过一层层的爬取，就能将 Web 站点中的页面基本爬取完毕。

2) 页面内容分析。

页面中内容还包含其他链接，这样通过一层层的爬取，就能将 Web 站点中的页面基本爬取完毕。首先，定义一个 HTMLParser 的新的类，使用覆盖 handle_starttag()的方法来显示所有链接中的 href 标

签属性，然后创建好一个实例用来返回 HTMLParser 的对象，这样，就可以使用 urllib.urlopen(url)打开并读取 HTML 文档中的内容了。对于 HTML 文件中包含的链接内容，可以使用 read()函数读取将数据传递给 HTMLParser 对象，HTMLParser 的 feed 函数可以接收到数据，这样就可以抓取到页面内的链接了。

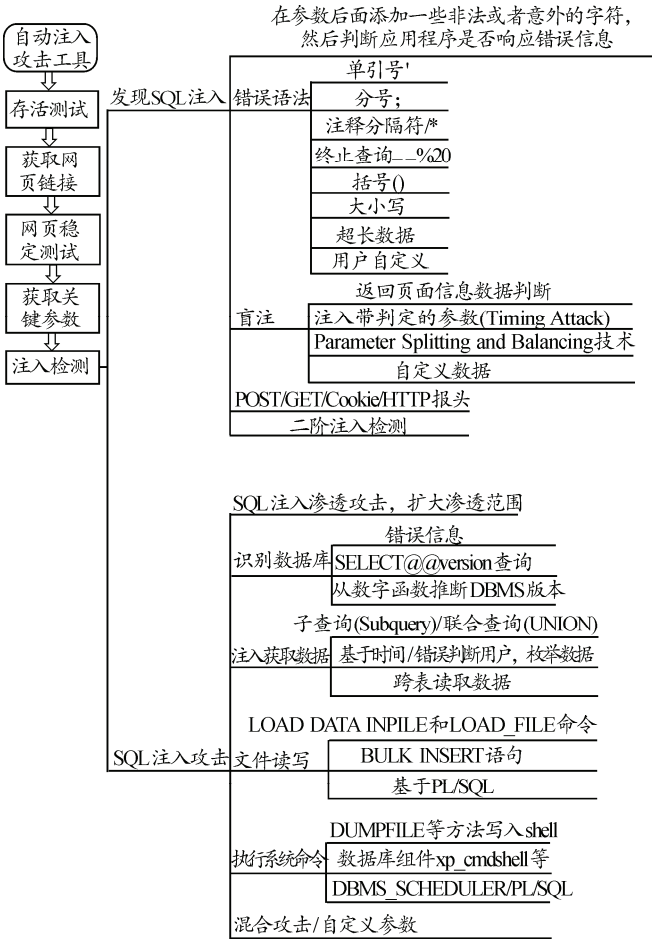

图 2　自动注入攻击工具框架总体架构图

利用爬虫技术来对 Web 目标的链接进行抓取确实不失为一个好方法，但很多时候，由于服务器安全设置或者是防火墙的设置，并不允许同一台服务器的与目标服务器建立大量的连接请求。很多时候网络爬虫的效率和效果并不太好。这时就可以利用搜索引擎搜索的关键字来定位与目标相关的 Web 链接，然后直接将抓取的链接进行 SQL 注入检测。

2.4　Cookie 信息的提取

笔者还采用从页面中提取 Cookies 信息的方法，可以使用 Cookielib 模块中的 LWPCookieJar()函数创建一个 Cookie jar 来实现。LWPCookieJar()函数可以返回一个从硬盘加载和存储的 Cookie 的对象。然后使用 urllib2 模块的 build_opener([handler, . . .])函数创建 opener 对象来处理 Cookies。之后使用 urlopen(Request)函数来打开 HTML 文件，HTML 文件中的 Cookie 就会被存放在 LWPCookieJar 对象中，之后，使用 save(filename)函数就获取到 Cookie 了。

2.5　SQL 注入检测过程

以 Mysql 数据库为示例，检测流程如图 3 所示。

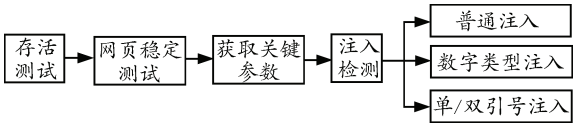

图 3　SQL 注入检测流程

存活测试是通过发送数据包 GET 或 POST 的方式，根据服务器的响应来判断目标 Web 是否存活。对于网页的稳定性测试，因为只有动态的网页才会有数据的交互，纯粹的 html 页面没有直接的数据库交互，所以可通过多次访问来确定服务器响应的网页的 md5 值是否完全匹配，也可以直接通过寻找关键字来确定是否是动态网页。通过 GET/POST 方式请求来发送非法或者意外的数据请求来进行 SQL 注入攻击检测。

其中，SQL 注入攻击模块，使用 Python 的 urllib，urllib2 cookielib 库的模块发送 GET/POST 请求对于注入功能语句进行遍历访问。图 4 为 POST 请求发送数据代码示例：

```
1  import urllib
2  import urllib2
3  import cookielib
4
5  cj = cookielib.CookieJar()
6  opener = urllib2.build_opener(urllib2.HTTPCookieProcessor(cj))
7  opener.addheaders = [('User-agent','Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1)')]
8  urllib2.install_opener(opener)
9  req = urllib2.Request("http://www.swust.edu.cn",urllib.urlencode({"username":"dllk","password":"dllk"}))
10 req.add_header("Referer","http://www.swust.edu.cn")
11 resp = urllib2.urlopen(req)
12 print resp.read()
```

图 4　POST请求发送数据示例

图 5　输入具有 SQL 注入的页面

3　实现及测试效果

在输入有 SQL 注入的 URL 页面，如图 5 所示。

在基本设置里可以先点击选择参数，参数控制界面如图 6，可以选择列举出数据库类型、操作系统、测试参数，同时用户也可以根据测试情况进一步自定义限制这些参数，以达到更好的测试效果。

图 6　参数控制界面

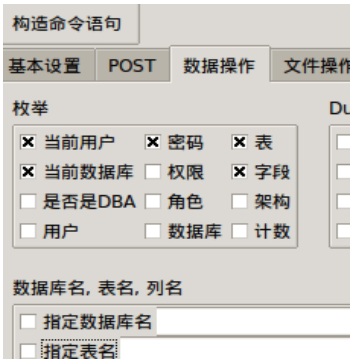

图 7　数据操作界面

设置 POST 方式、数据操作等，界面如图 7 所示，用户可以设置枚举当前用户、密码、表段、字段、数据结构等，也可自定义指定数据库名指定表名来进行效果更好的检测。

例如，当前测试的 URL 生成的命令，其测试结果如图 8、图 9 所示。

```
-u"http://**.swust.edu.cn/***.asp?ArticleID=279"
--random-agent --dbms=access --current-user
--current-db --tables --columns -v 2
```

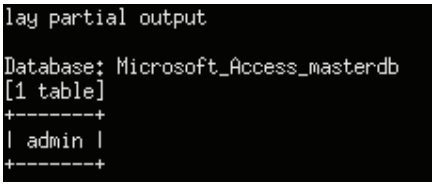

图 8　列举数据库表段结果

图 9　注入测试结果参数

可以在界面上实时看到得到的注入攻击结果，同时结果保存在 output 文件夹下，或通过历史记录，直接查看所获取的信息。

4　结束语

笔者设计的自动 SQL 注入检测工具通用框架融合了旁注、扫描、各种版本的提权等功能，并且工具的自定义功能可以进行进一步拓展，以满足灵活的 SQL 注入变形攻击的需求。

参考文献：

[1] Victor Chapela.Advanced SQL Injection[EB/OL]. www.owasp. org/index. php/Image:Advanced_SQL_Injection.ppt.

[2] 吴翰清. 白帽子讲 Web 安全[M]. 北京: 电子工业出版社, 2012.

[3] TinKode. MySQL. com Victim of SQL Injection Attack[C/OL]. http://www.acunetix.com/blog/web-security-zone/articles/mysql-com-victim-of-sql-injection/, 2011.

[4] James C. Foster Mike Price. 安全编程修炼之道[M]. 邓劲生, 译. 北京: 清华大学出版社, 2006: 5-6.

[5] Macro Ivaldi[EB/OL]. http://www.0xdeadbeef.info/exploits/.

[6] Chen Xueping. SQL injection attack and guard technical research[R]. Procedia Engineering, 2011.

[7] Indrani Balasundaram, Ramaraj E. An Authentication Mechanism to prevent SQL Injection Attacks[R]. 2011.

[8] 赵文龙, 朱俊虎, 王清贤. SQL Injection 分析与防范[J]. 计算机工程与设计, 2006, 27(2): 300-302.

[9] 陈小兵, 张汉煜, 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): 150-152.

[10] 余静, 高丰, 徐良华, 等. 基于 SQL 注入的渗透性测试技术研究[J]. 计算机工程与设计, 2007, 28(15): 3577-3579.