# 程序简介

超级SQL注入工具（SSQLInjection）是一款基于HTTP协议自组包的SQL注入工具，支持出现在HTTP协议任意位置的SQL注入，支持各种类型的SQL注入，支持HTTPS模式注入。

超级SQL注入工具目前支持Bool型盲注、错误显示注入、Union注入，支持Access、MySQL5以上版本、SQLServer、Oracle等数据库。

超级SQL注入工具采用C#开发，底层采用Socket发包进行HTTP交互，极大的提升了发包效率，相比C#自带的HttpWebRequest速度提升2-5倍。

超级SQL注入工具支持盲注环境获取世界各国语言数据，解决了各种常见注入工具在盲注环境下无法支持中文等多字节编码的数据。

工具特点：

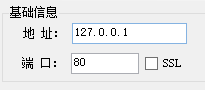
1. 支持任意地点出现的任意SQL注入
2. 支持各种语言环境。大多数注入工具在盲注下，无法获取中文等多字节编码字符内容，本工具可完美解决。
3. 支持注入数据发包记录。让你了解程序是如何注入，有助于快速学习和找出注入问题。
4. 依靠关键字进行盲注，可通过HTTP相应状态码判断，还可以通过关键字取反功能，反过来取关键字。

程序运行需要安装. Net Framework 4.0。运行环境Win7,Win8环境已测试，其他环境请自测。

# 基础信息配置

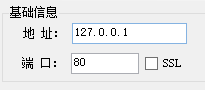
## 地址

打开程序在基础配置里面填写需要注入地址的域名或IP地址。



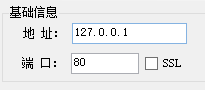
## 端口

打开程序在基础配置里面填写需要注入的域名或IP地址的端口。



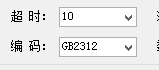
## SSL

如果是HTTPS传输的网页需要选择SSL，选择SSL会切换到443端口，如果是其他端口请修改端口地址。



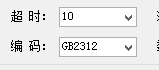
## 超时

程序使用Socket传输，每个HTTP请求的超时时间，如果超过此时间，程序将放弃本次请求，单位是秒。



## 编码

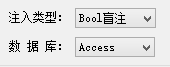
程序在获取HTML网页时，采用的解码方式，程序会自动识别网页编码，如果识别失败，将采用此编码进行解码。编码可以在HTTP相应头的Content-Type或HTML头里面查看“charset=xxx”。



## 注入类型

程序目前支持Bool盲注、Union注入、错误显示注入,已基本覆盖全部注入获取数据方式。

延时注入可以选择Bool盲注，在判断方式中选择时间判断，目前延时只支持mysql数据库。



### Bool盲注

Bool盲注利用数据库的阿斯特码（ASCII码）或Unicode、Hex等编码将目标数据转换成数字，利用And条件来判断此数字的值，判断完成后根据对应编码转换成字符串。

目前Bool盲注支持Access、MySQL5、SQLServer、Oracle等数据库，并支持中文、日文、繁体中文等世界各种语言。解决了各种注入工具在盲注环境无法获取中文等多字节编码的字符。

### Union注入

Union注入利用数据库的Union联合查询来获取数据。

目前盲注环境支持Access、MySQL5、SQLServer、Oracle等数据库。

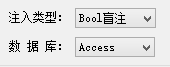
### 错误显示注入

错误显示注入利用数据库在完成某项操作时，在程序出现错误时将错误信息显示出来进行注入获取目标数据。由于错误显示的字符有长度限制，所以比Union注入方式稍慢，MySQL错误信息显示大约最长为64个字符，Oracle错误显示大约最长约为256个字符，SQLServer错误信息显示大约最长为2030个字符。

目前盲注支持MySQL5、SQLServer、Oracle等数据库。Access不支持错误显示注入。

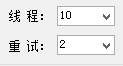
## 数据库

判断数据库类型，选择对应的数据库。



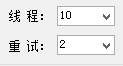
## 线程

选择程序同时能运行的线程数量，默认10个线程。



## 重试

程序发送一个HTTP包失败后，尝试继续发包的次数。



## 自动识别

填好数据包，填写地址和端口即可自动进行注入识别，支持GET和POST参数自动识别注入。

## 导出配置

点击导出配置，将选择需要导出程序配置信息的路径，程序将导出配置信息到一个XML文件中，以后可以使用菜单中的导入配置来加载配置信息。

# 注入中心

## HTTP请求数据包

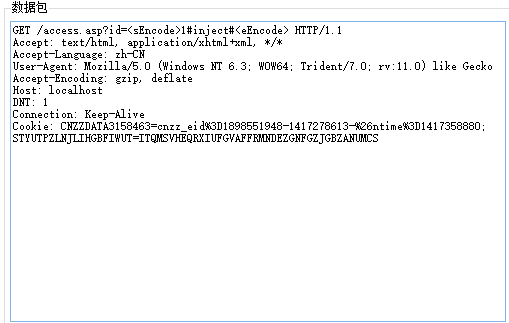
**方法一：**

在数据包中输入URL地址http://127.0.0.1:8090/mysql.jsp?id=1，在数据包空白处单击鼠标右键，选择“根据URL生成数据包”，即可自动配置GET型请求数据包，并自动设置IP地址和端口。

**方法二：**

使用抓包工具将HTTP发包数据抓取填写到这里，可使用Fiddler、Burp Suite等抓包工具，或手工配置数据包。

注意事项：如果是POST提交数据，必须有Content-Length属性，程序才能自动计算长度。



## Token随机值处理

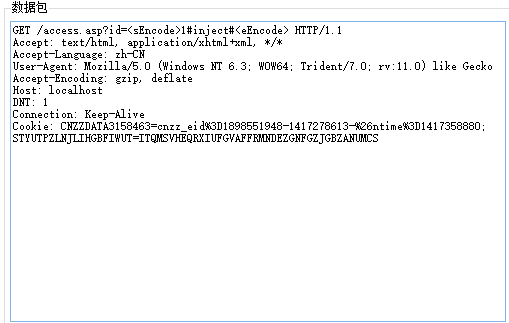
**方法一：**

在数据包中输入URL地址http://127.0.0.1:8090/mysql.jsp?id=1，在数据包空白处单击鼠标右键，选择“根据URL生成数据包”，即可自动配置GET型请求数据包，并自动设置IP地址和端口。

**方法二：**

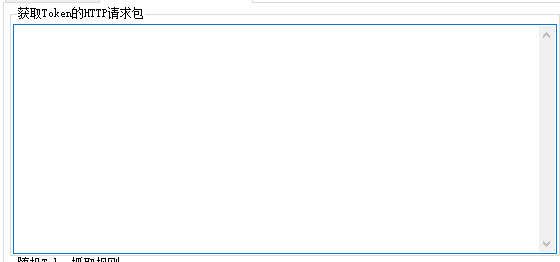
使用抓包工具将HTTP发包数据抓取填写到这里，可使用Fiddler、Burp Suite等抓包工具，或手工配置数据包。

注意事项：如果是POST提交数据，必须有Content-Length属性，程序才能自动计算长度。



### 获取Token的HTTP请求包

如果在注入数据包中标记了Token，需要在此设置抓取Token的HTTP包，程序将在每次注入发包时，首先访问此页面，获取Token。



### Token抓取规则

开始字符设置为Token前面的一段字符，结束字符设置为Token后面一段字符，程序通过开始字符和结束字符来抓取中间的Token字符。

## 注入设置

### 开启URL编码

开启URL编码后，程序将会把编码标记中的数据进行URL编码操作，建议选择，因为请求参数中如果有特殊字符，可能导致发包结果不一致。



### 302跟踪

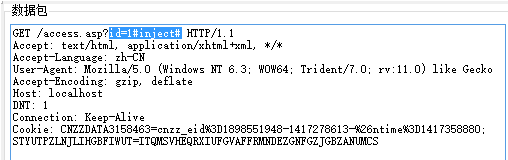
默认未开启，开启后，程序遇到302重定向时，将请求重定向的目标地址。如果可以根据状态码判断注入时，可选中，下面关键字配置状态码，进行注入。



### 注入标记

通常手工注入使用”xxxx.asp?id=1 and 1=1”进行注入判断，那么这里选中“ and 1=1”点击标记注入，程序将在有注入的位置设置注入标记，程序注入时将标记替换成注入获取数据的代码。核心就是将我们测试语句中的“and 1=1”替换成标记。





示例：

标记其实就是选择payload放在那个位置

数字型： 1 and 1=1 把and 1=1替换成标记

字符型： 1' and 1=1 and '1'='1 把and 1=1替换成标记

字符型： 1' and 1=1# 把and 1=1替换成标记

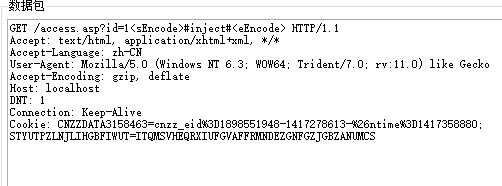
字符型： 1' and 1=1-- 把and 1=1替换成标记

搜索型： 1%' and 1=1 and '%'=' 把and 1=1替换成标记

### 编码标记

选择数据包中需要使用URL编码的字符，点击标记编码，程序将插入编码标记。程序将自动一配置好的编码方式进行编码。





### 注入取数据配置

参考我是如何开始注入章节的自动模式中的获取数据配置。

### 标记Token随机值

在数据包中，将随机值的位置选中，并单机“标记Token随机值”按钮，即可标记，标记格式为：<Token>xxxxx</Token>，也可参数标记格式，人工填写标记。

注：标记Token时，只能单线程获取数据，因为多线程访问，或打乱Token生成的顺序。

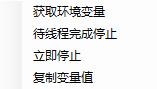
# 数据中心

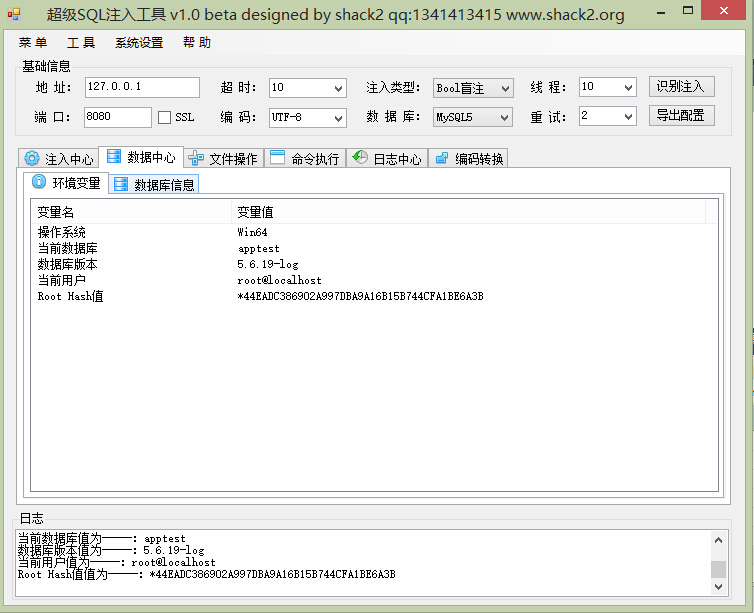
## 环境变量

### 获取环境变量

点击右键弹出菜单，点击获取环境变量程序将获取数据库的基础信息，Access数据库不支持此功能。







### 复制变量值

选中对应的变量，按右键复制值。

## 数据库信息

### 获取数据库

点击获取数据库，程序将获取全部数据库列表，Oracle将获取所有用户列表，Access没有库。

### 获取表

点击获取数据表，程序将获取对应数据库的表。

### 获取列

点击获取数据列，程序将获取对应表的列。

### 获取数据

获取开始数据，默认开始下标为0，即重第一条数据开始获取，获取条数可设置，但不能超过重下标开始的剩余行数。双击列名可进行排序操作。



### 编码设置

这里的编码设置用于Bool盲注取数据的配置，盲注如出现乱码，可以选择其他编码方式尝试获取数据，查看是否有乱码。这里编码主要用于数据库中Hex、Ascw、Unicode等编码函数的解码。



### 导出数据

点击导出数据将选择导出数据的磁盘路径，程序将自动导出数据。

### 添加节点

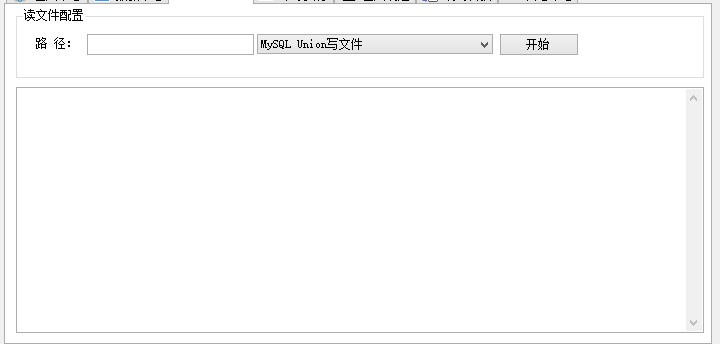
### 删除节点

### 修改节点

# 文件操作

## MySQL Load\_File读文件

在Mysql的账户拥有文件读写权限下（一般情况下只有ROOT账户拥有），可在盲注、错误显示、Union注入下，读取地址路径下的文本文件内容。填写路径，选择MySQL Load\_File读文件，点击开始即可。



## MySQL Union写文件

在Mysql的账户拥有文件读写权限下（一般情况下只有ROOT账户拥有），可在Union注入下，向磁盘写文本文件。

## SQLServer FileSystemObject写文件

在SA权限下，可利用FileSystemObject写文件，写文件内容注入不要超过4000字节，同时如何是GET请求提交注意，一般GET提交不能超过1024个字节，所以如果没写成功，看看你提交的数据是否太长。

## SQLServer Sp\_MakeWebTask写文件

在SA权限下，可利用Sp\_MakeWebTask写文件，写文件内容注入不要超过4000字节，同时如何是GET请求提交注意，一般GET提交不能超过1024个字节，所以如果没写成功，看看你提交的数据是否太长。相比FileSystemObject成功率较低。

## SQLServer利用备份数据库写文件

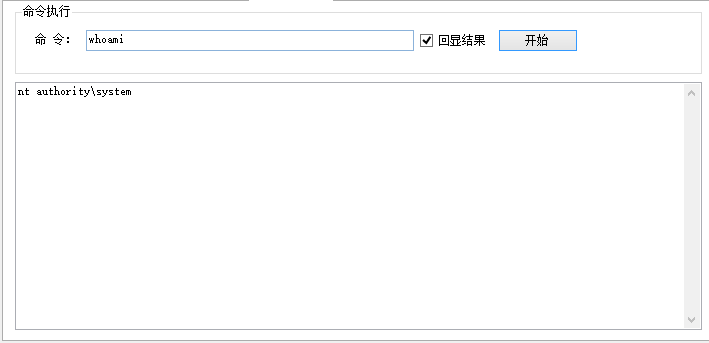
在SA权限下，可利用Back DataBase备份数据库方法，将内容备份到文件中。此方法写入文件后，此文件属于数据库备份文件，所以有一些其他多余数据。

## SQLServer FileSystemObject读文件

在SA权限下，可利用FileSystemObject读文件，可在盲注、显错注入、Union注入下读取文件。读取文件长度不能超过4000字节。

# 命令执行

目前此功能只支持SQLServer，在sa权限下，可利用xp\_cmdshell执行操作系统命令，可选择是否回显执行结果。可在盲注、显错注入、Union注入下获取回显结果。执行此功能需支持多语句。



# 注入绕过

## 字符替换

替换编辑编码内的字符，需注意的，默认系统设置在URL编码前处理绕过字符，如果没有选择在URL编码前处理绕过字符，那么替换的字符就是URL编码后的字符，如替换空格成/\*\*/，那么填写的就是“%20”替换成“%2f\*\*%2f”。

## 包含关键字

适用于MySQL数据库的绕过，可人工测试可行后选择使用/\*!\*/包含关键字来绕过防护。

## 随机大写

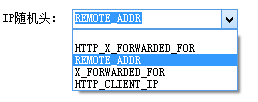
将编码标记内的字符进行随机大小写转换，绕过SQL注入防护。

## 发包延时

将每个包发送后休息一定时间在发送，绕过一些防护。

## IP随机头

在HTTP请求头中增加随机IP值，这里列出来常见的伪装IP头，可测试选择对应的头，程序会随机产生IP。



## Base64编码

将编码范围内参数进行base64编码。

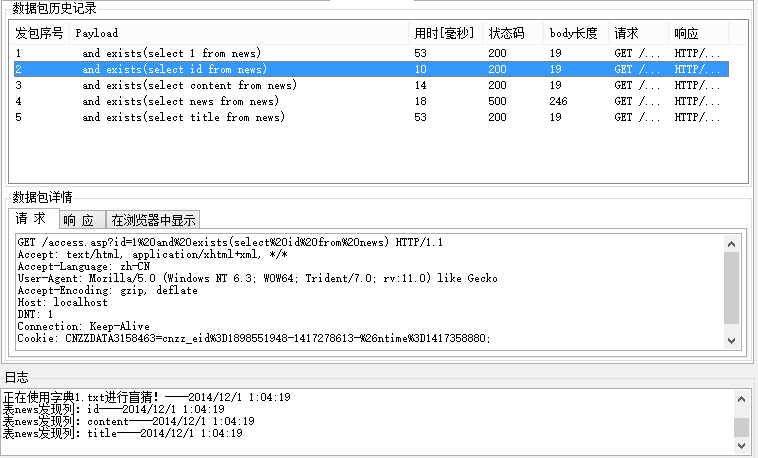
# 编码转换

填写需要转换的字符，选择对应编码解码方法就行。

# 日志中心

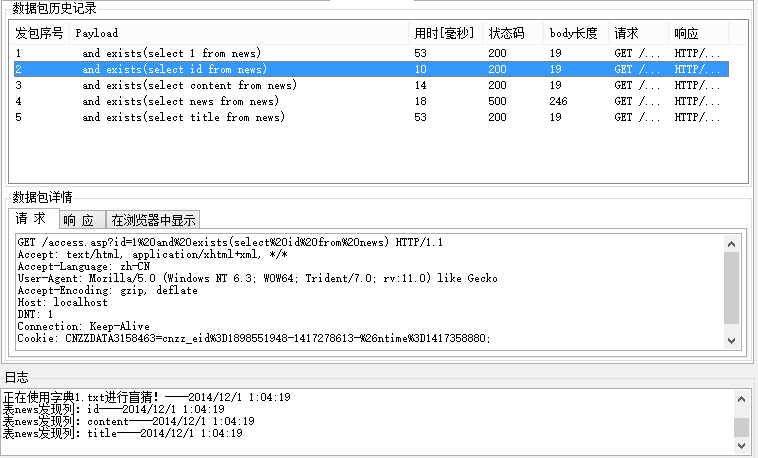
## 数据包历史记录

数据包历史记录记录了程序每一个HTTP的请求与相应，可用于错误排查。如需要提高性能，可在菜单中的系统设置中选择关闭HTTP发包日志记录。



## 数据包详情

在数据包历史记录中选择一条记录，程序将自动将详细信息进行显示，可以查看请求响应数据，在响应中同时按“Ctrl+Alt”键可以弹出查找关键字面板，进行关键字查找。



# 批量扫描注入

导入域名或链接，选择爬行或检测注入。

# 停止注入

## 立即停止

当前功能下点击右键，点击立即停止，程序将停止当前所有线程。

# 系统设置

开启MySQL多字节取数据，将判断数据是否有中文。

开启自动检测更新，程序将自动检测更新。

开启底部日志，程序将在底部显示日志信息。

开启发包日志，程序将记录HTTP每一个发包日志。

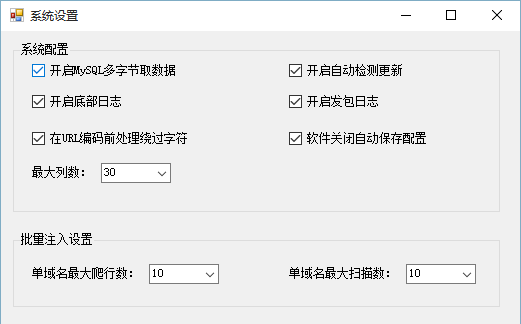
开启在URL编码前处理绕过字符，所有绕过字符处理将处理后再进行URL编码。

软件关闭自动保存配置，软件关闭时是否自动保存配置。

最大列数，自动识别注入时，自动进行Union测试时的最大测试列。

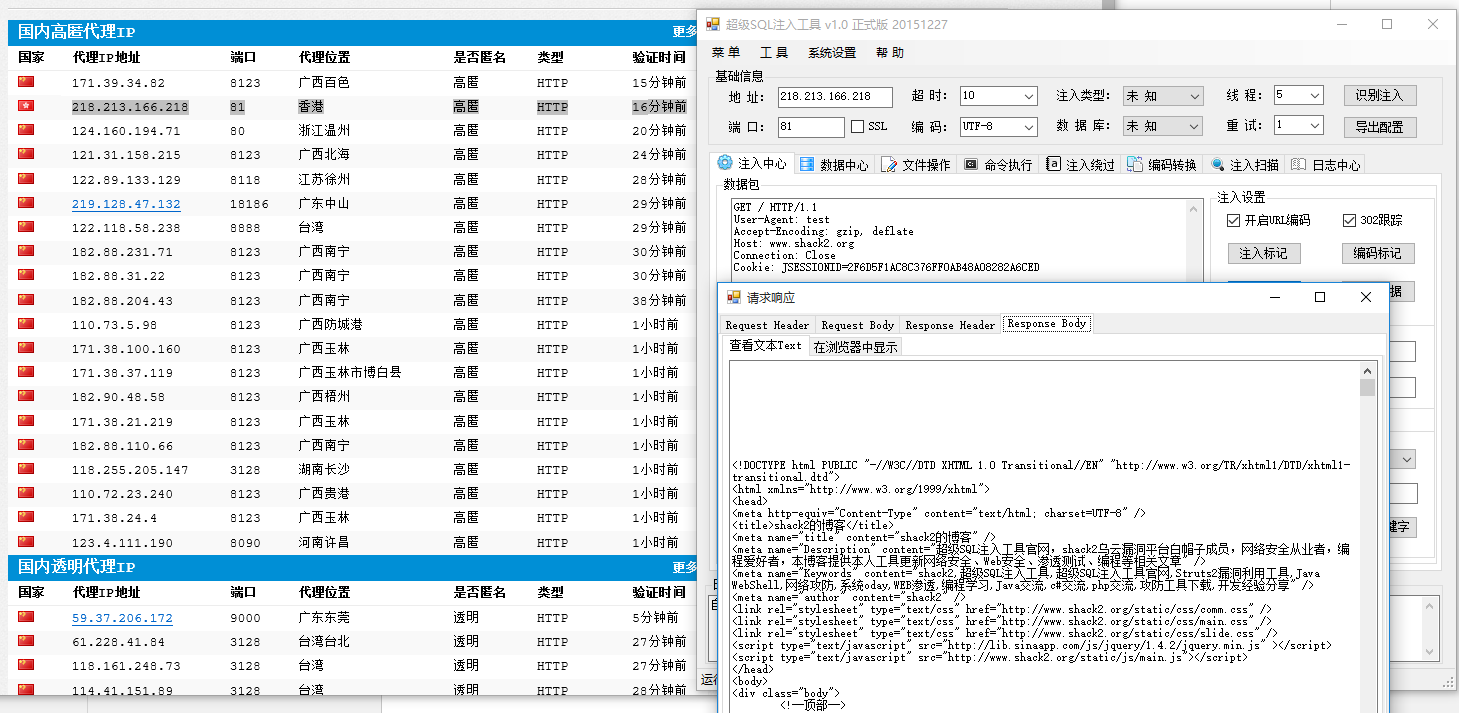
单域名最大爬行数，批量扫描注入时，如果选中爬行连接，一个域名爬行到最大链接数后将停止继续爬行链接。

单域名最大扫描数，批量扫描注入时，单个域名最大检测注入链接数。



# 隐藏功能之代理

工具一直没有提供设置代理的地址那是不是就不能使用代理了呢，使用代理很简单，在基础配置的地址和端口上输入代理的IP地址和端口就可以使用代理了，是不是很神奇呢？

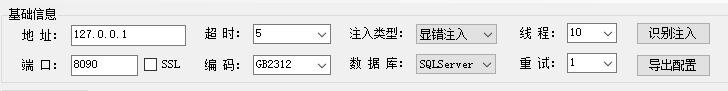


# 我如何开始注入？

## 自动模式

自动模式适合网页响应内容比较固定的注入检测，自动模式只支持GET或者POST传递的参数的注入检测，不支持HTTP请求属性的注入，如Referer注入。

首先开始自动识别注入时，需要配置基础信息，地址、端口、超时、编码、线程、重试次数。



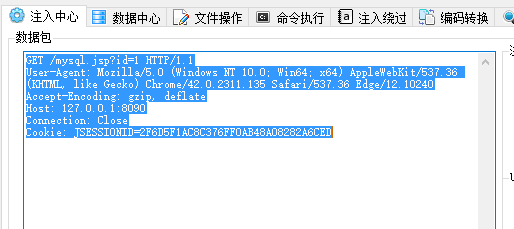
其次需要配置HTTP请求数据包，可以在数据包文本框里按鼠标右键选择生成GET或POST数据模板，然后修改请求的URL地址和HOST及提交数据。

* 注入http://127.0.0.1:8090/mysql.jsp?id=1 示例数据包配置：

1. 方法一

在数据包中输入URL地址<http://127.0.0.1:8090/mysql.jsp?id=1>，在数据包空白处单击鼠标右键，选择“根据URL生成数据包”，即可自动配置GET型请求数据包，并自动设置IP地址和端口。

首先生成GET数据包模板，修改请求URL地址，这里取URL的根目录“/mysql.jsp?id=1”，然后修改Host为“127.0.0.1:8090”即可。



最后点击识别注入，就可以开始自动识别注入了，如果识别成功，会自动设置数据库类型和注入类型，并自动进行注入标记，这时候只需要切换到数据中心选择对应功能去获取相关数据即可。

## 手动模式

### 基础配置

手动模式相对自动模式配置相对复杂，但自动模式可以对付各种情况各种位置的注入。

首先需要完成基本配置，基本配置参考自动模式的整个过程。

然后需要人工对注入进行判断后在选择对应数据库和注入类型，并进行标记，那么如何进行判断和标记呢？

首先需要判断注入的类型，比如数字型、字符型、搜索型等，然后判断对应的数据库。

判断完成后，需要对注入位置进行标记，如何判断和标记，请往下看！

### 判断注入类型

#### 通用数字型

* URL地址：

http://127.0.0.1:8090/JavaSQLInjection/mysql.jsp?id=1

* 判断注入：

改变参数id=1 and 1=1 和 id=1 and 1=2进行测试，如果1=1页面显示正常和原页面一样，并且1=2时页面报错或者页面部分数据显示不正常，那么可以确定此处为数字型注入。

* 判断数据库类型：

使用exists函数判断查询对应数据库的系统表是否存在，根据页面是否正常来判断对应数据库类型。

MySQL：http://127.0.0.1:8090/JavaSQLInjection/mysql.jsp?id=1 and exists(select 1 from information\_schema.tables)

SQLServer: http://127.0.0.1:8090/JavaSQLInjection/mysql.jsp?id=1 and exists(select 1 from sysobjects)

Access:http://127.0.0.1:8090/JavaSQLInjection/mysql.jsp?id=1 and exists(select 1 from MSysAccessObjects)

Oracle:http://127.0.0.1:8090/JavaSQLInjection/mysql.jsp?id=1 and exists(select 1 from user\_tables)

#### 通用字符型

* URL地址：

http://127.0.0.1:8090/JavaSQLInjection/mysql.jsp?id=1

* 判断注入：

改变参数id=1’ and 1=1 and ‘1’=’1 和 id=1’ and 1=1 and ‘1’=’2进行测试，或者利用注释符号“#”或者“— ”，当然特殊字符记得进行URL编码，“#”URL编码后是“%23”，“— ”编码后是“—%20”，利用注释符，那么可以改变参数为id=1’ and 1=1%23和id=1’ and 1=2%23来进行测试，如果1=1页面显示正常和原页面一样，并且1=2时页面报错或者页面部分数据显示不正常，那么可以确定此处为字符型注入。

* 判断数据库类型：

使用exists函数判断查询对应数据库的系统表是否存在，根据页面是否正常来判断对应数据库类型。

MySQL：http://127.0.0.1:8090/JavaSQLInjection/mysql.jsp?id=1’ and exists(select 1 from information\_schema.tables) and ‘a’=’a

SQLServer:http://127.0.0.1:8090/JavaSQLInjection/mysql.jsp?id=1’ and exists(select 1 from sysobjects) and ‘a’=’a

Access:http://127.0.0.1:8090/JavaSQLInjection/mysql.jsp?id=1’ and exists(select 1 from MSysAccessObjects) and ‘a’=’a

Oracle:http://127.0.0.1:8090/JavaSQLInjection/mysql.jsp?id=1’ and exists(select 1 from user\_tables) and ‘a’=’a

#### 通用搜索型

* URL地址：

http://127.0.0.1:8090/JavaSQLInjection/mysql.jsp?id=1

* 判断注入：

改变参数id=1%’ and 1=1 and ‘%’=’ 和 id=1%’ and 1=2 and ‘%’=’进行测试，或者利用注释符号“#”或者“— ”，当然特殊字符记得进行URL编码，“#”URL编码后是“%23”，“— ”编码后是“—%20”，利用注释符，那么可以改变参数为id=1%’ and 1=1%23和id=1%’ and 1=2%23来进行测试，如果1=1页面显示正常和原页面一样，并且1=2时页面报错或者页面部分数据显示不正常，那么可以确定此处为搜索型注入。

* 判断数据库类型：

使用exists函数判断查询对应数据库的系统表是否存在，根据页面是否正常来判断对应数据库类型。

MySQL：http://127.0.0.1:8090/JavaSQLInjection/mysql.jsp?id=1%’ and exists(select 1 from information\_schema.tables) and ‘%’=’

SQLServer:http://127.0.0.1:8090/JavaSQLInjection/mysql.jsp?id=1%’ and exists(select 1 from sysobjects) and ‘%’=’

Access:http://127.0.0.1:8090/JavaSQLInjection/mysql.jsp?id=1%’ and exists(select 1 from MSysAccessObjects) and ‘%’=’

Oracle:http://127.0.0.1:8090/JavaSQLInjection/mysql.jsp?id=1%’ and exists(select 1 from user\_tables) and ‘%’=’

#### OR类型

* URL地址:

http://127.0.0.1:8090/JavaSQLInjection/login.jsp?username=1&pass=1

* 判断注入：

改变参数username=1’ or(1=1 and 1=1) and ‘1’=’1 和 username=1’ or(1=1 and 1=2) and ‘1’=’1进行测试，或者利用注释符号“#”或者“— ”，当然特殊字符记得进行URL编码，“#”URL编码后是“%23”，“— ”编码后是“—%20”，利用注释符，那么可以改变参数为1’ or(1=1 and 1=1)%23和1’ or(1=1 and 1=2)%23来进行测试，如果1=1页面显示正常和原页面一样，并且1=2时页面报错或者页面部分数据显示不正常，那么可以确定此处为字符型注入。

* 判断数据库类型：

使用exists函数判断查询对应数据库的系统表是否存在，根据页面是否正常来判断对应数据库类型。

MySQL:http://127.0.0.1:8090/JavaSQLInjection/mysql.jsp?username=1’ or(1=1 and exists(select 1 from information\_schema.tables)) and ‘a’=’a&password=1

SQLServer:http://127.0.0.1:8090/JavaSQLInjection/mysql.jsp?id=1’ or(1=1 and exists(select 1 from sysobjects)) and ‘a’=’a

Access:http://127.0.0.1:8090/JavaSQLInjection/mysql.jsp?id=1’ or(1=1 and exists(select 1 from MSysAccessObjects)) and ‘a’=’a

Oracle:http://127.0.0.1:8090/JavaSQLInjection/mysql.jsp?id=1’ or(1=1 and exists(select 1 from user\_tables)) and ‘a’=’a

### 注入标记

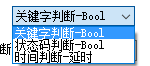
很简单，首先选中需要进行URL编码的参数值，选择需要进行URL编码的参数后再选中之前判断注入类型中的“and 1=1”，然后点击右边的标记注入按钮，或者直接删除“and 1=1”后，在对应的位置插入注入标记。

### 获取数据配置

#### Bool盲注取数据配置

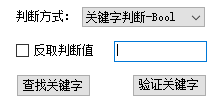
##### 基本配置

Bool盲注配置依靠关键字、状态码、响应时间等因素进行逻辑上的真假判断。



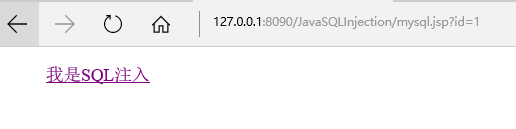
* + **关键字判断**

关键字是正常逻辑上为真，页面上存在的字符，而逻辑上为假，页面上没有出现的字符。即“and 1=1”出现的字符，而“and 1=2”时未出现的字符。



示例：

正常的数字型注入：







盲注关键字可选择and 1=1时，页面上出现的关键字，and 1=2时没有出现的关键字，比如“SQL” 、“我是SQL”、“SQL注入”。一个原则，只要确保这个词是1=1条件是页面出现这个关键词，而1=2时不出现。

* + 状态码判断

状态码判断原理同上，只是条件变成了and 1=1时出现这个状态码，二and 1=2时不出现这个状态码。

选择状态码判断后，程序盲注的关键字将依靠状态码判断，即“and 1=1”时HTTP请求的状态码为此关键字（200、403、302、500等），而“and 1=2”时为其他状态码。

* + 时间判断：

需人工判断网页平均响应时间，设置一个时间阀值，当页面响应时间超过这个阀值，证明当前判断是正确的，否则不正确。

##### 关键字取反

关键字取反选择后，程序将反过来取注入的判断依据，比如即“and 1=2”出现的字符，而“and 1=1”时未出现的字符。如果选择了状态码，即“and 1=2”出现的状态码，而“and 1=1”时未为其他状态码。

##### 验证关键字是否正确

点击关键字可以验证关键字是否配置正确。

#### 错误显示方式取数配置

无需配置，程序将自动获取错误信息，需要注意的是程序可以显示错误信息。

#### Union注入取数据配置

因为Union联合查询前后的列数需要保持一致，程序只能利用显示列显示注入获取的数据，所以Union注入需要配置SQL注入的列数、和数据显示列。人工判断可通过order by方式判断或Union select 1,2,3,…………来判断列数和显示列。

**注意事项：**由于Access数据库没有系统表，所以只能盲猜表和列，所以需要配置盲注取数据配置。配置参考Bool盲注取数据配置。

### 获取数据

切换到数据中心，尝试获取数据，如获取失败，可尝试切换数据获取方式，或者人工测试是否有相关防护规则，检查网页编码是否正常，对HTTP发包日志进行查看分析失败原因，在调整配置。

# Bug反馈

邮箱反馈：1341413415@qq.com

QQ群反馈：84978967