# sql注入

[编辑](https://baike.baidu.com/item/sql%E6%B3%A8%E5%85%A5/javascript:;) [锁定](https://baike.baidu.com/item/sql%E6%B3%A8%E5%85%A5/javascript:;" \o "锁定" \t "https://baike.baidu.com/item/sql%E6%B3%A8%E5%85%A5/_blank) [讨论2](https://baike.baidu.com/planet/talk?lemmaId=150289" \t "https://baike.baidu.com/item/sql%E6%B3%A8%E5%85%A5/_blank)

本词条由[“科普中国”科学百科词条编写与应用工作项目](https://baike.baidu.com/science" \t "https://baike.baidu.com/item/sql%E6%B3%A8%E5%85%A5/_blank) 审核 。

SQL注入即是指web应用程序对用户输入数据的合法性没有判断或过滤不严，攻击者可以在web应用程序中事先定义好的查询语句的结尾上添加额外的SQL语句，在管理员不知情的情况下实现非法操作，以此来实现欺骗数据库服务器执行非授权的任意查询，从而进一步得到相应的数据信息。 [1]

中文名

sql注入

外文名

SqlInject

语    言

计算机语言

类    型

参数

类    别

平台层注入、代码层注入

## 目录

1. 1 [定义](https://baike.baidu.com/item/sql%E6%B3%A8%E5%85%A5/150289?fr=aladdin" \l "1)
2. 2 [特点](https://baike.baidu.com/item/sql%E6%B3%A8%E5%85%A5/150289?fr=aladdin" \l "2)
3. 3 [原理](https://baike.baidu.com/item/sql%E6%B3%A8%E5%85%A5/150289?fr=aladdin" \l "3)
4. 4 [危害](https://baike.baidu.com/item/sql%E6%B3%A8%E5%85%A5/150289?fr=aladdin" \l "4)
5. 5 [注入过程](https://baike.baidu.com/item/sql%E6%B3%A8%E5%85%A5/150289?fr=aladdin" \l "5)
6. 6 [注入方法](https://baike.baidu.com/item/sql%E6%B3%A8%E5%85%A5/150289?fr=aladdin" \l "6)
7. 7 [攻击特点](https://baike.baidu.com/item/sql%E6%B3%A8%E5%85%A5/150289?fr=aladdin" \l "7)
8. 8 [攻击手法](https://baike.baidu.com/item/sql%E6%B3%A8%E5%85%A5/150289?fr=aladdin" \l "8)
9. 9 [检测技术](https://baike.baidu.com/item/sql%E6%B3%A8%E5%85%A5/150289?fr=aladdin" \l "9)
10. 10 [注入防范措施](https://baike.baidu.com/item/sql%E6%B3%A8%E5%85%A5/150289?fr=aladdin" \l "10)

## sql注入定义

[编辑](https://baike.baidu.com/item/sql%E6%B3%A8%E5%85%A5/javascript:;)

SQL是操作[数据库](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93/103728" \t "https://baike.baidu.com/item/sql%E6%B3%A8%E5%85%A5/_blank)数据的结构化查询语言，网页的应用数据和后台数据库中的数据进行交互时会采用SQL。而SQL注入是将Web页面的原[URL](https://baike.baidu.com/item/URL/110640" \t "https://baike.baidu.com/item/sql%E6%B3%A8%E5%85%A5/_blank)、表单域或数据包输入的参数，修改拼接成SQL语句，传递给Web服务器，进而传给数据库服务器以执行数据库命令。如Web应用程序的开发人员对用户所输入的数据或cookie等内容不进行过滤或验证(即存在注入点)就直接传输给数据库，就可能导致拼接的SQL被执行，获取对数据库的信息以及提权，发生SQL注入攻击。 [2]

## sql注入特点

[编辑](https://baike.baidu.com/item/sql%E6%B3%A8%E5%85%A5/javascript:;)

1、广泛性

任何一个基于[SQL](https://baike.baidu.com/item/SQL/86007" \t "https://baike.baidu.com/item/sql%E6%B3%A8%E5%85%A5/_blank)语言的[数据库](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93/103728" \t "https://baike.baidu.com/item/sql%E6%B3%A8%E5%85%A5/_blank)都可能被攻击，很多开发人员在编写Web应用程序时未对从输入参数、Web表单、cookie等接受到的值进行规范性验证和检测，通常会出现SQL注入漏洞。 [3]

2、隐蔽性

SQL注入语句一般都嵌入在普通的HTTP请求中，很难与正常语句区分开，所以当前许多防火墙都无法识别予以警告，而且SQL注入变种极多，攻击者可以调整攻击的参数，所以使用传统的方法防御SQL注入效果非常不理想。 [3]

3、危害大

攻击者通过SQL注入获取到服务器的库名、表名、字段名，从而获取到整个服务器中的数据，对网站用户的数据安全有极大的威胁。攻击者也可以通过获取到的数据，得到后台管理员的密码，然后对网页页面进行恶意篡改。这样不仅对数据库信息安全造成严重威胁，对整个数据库系统安全也影响重大。 [3]

4、操作方便

互联网上有很多SQL注入工具，简单易学，攻击过程简单，不需要专业知识也能自如运用。 [3]

## sql注入原理

[编辑](https://baike.baidu.com/item/sql%E6%B3%A8%E5%85%A5/javascript:;)

SQL注入攻击是通过操作输入来修改SQL语句，用以达到执行代码对WEB服务器进行攻击的方法。简单的说就是在post/getweb表单、输入域名或页面请求的查询字符串中插入SQL命令，最终使web服务器执行恶意命令的过程。可以通过一个例子简单说明SQL注入攻击。假设某网站页面显示时URL为http://www.example.com?test=123，此时URL实际向服务器传递了值为123的变量test，这表明当前页面是对数据库进行动态查询的结果。由此，我们可以在URL中插入恶意的SQL语句并进行执行。另外，在网站开发过程中，开发人员使用动态字符串构造SQL语句，用来创建所需的应用，这种情况下SQL语句在程序的执行过程中被动态的构造使用，可以根据不同的条件产生不同的SQL语句，比如需要根据不同的要求来查询数据库中的字段。这样的开发过程其实为SQL注入攻击留下了很多的可乘之机。 [4]

## sql注入危害

[编辑](https://baike.baidu.com/item/sql%E6%B3%A8%E5%85%A5/javascript:;)

但凡使用数据库开发的应用系统，就可能存在SQL注入攻击的媒介。自1999年起，SQL注入漏洞就成了常见安全漏洞之一。至今SQL注入[漏洞](https://baike.baidu.com/item/%E6%BC%8F%E6%B4%9E/1688129" \t "https://baike.baidu.com/item/sql%E6%B3%A8%E5%85%A5/_blank)仍然在CVE列表中排前10。 [5]

2011年美国国土安全局，Mitre和SANA研究所将SQL注入作为第一危险的安全漏洞。至今，SQL注入仍然是首要的难以修复的安全威胁漏洞（数据库生产厂商难以通过维护数据库自身功能或提高数据库安全策略来防范SQL注入）。 [5]

2012年，Barclaycard的一个代表声称97%的数据泄露都是由SQL注入引起的。2011年年末和2012年年初，在不到一个月的时间里，超过百万的网页遭受到SQL注入攻击。2008年见证了由于SQL注入引起的经济失调。甚至在2010年秋季，联合国官方网站也遭受SQL注入攻击。 [5]

2014年一个叫“TeamDigi7al”的黑客组织攻击了美国海军的一个名为“Smart Web Move”的web应用。此次事件直接造成美国海军数据库超过22万服役人员的个人信息被泄露。而事后，美国海军动用了超过50万美元来弥补此次的数据泄密事故。 [5]

在日本电话电报公司集团（NTT）发布的2014全球威胁情报的报告中提出了一个惊人的数字——“企业对一次小规模SQL注入攻击的平均善后开支，通常超过19.6万美元。” [5]

随着Web安全事件的不断频发，我们不得不思考SQL注入攻击的代价。显然，SQL注入不是一个过期的安全问题，恰恰相反，它是一种非常容易被使用的攻击方式，SQL注入并不需要高深的攻击手段便可以轻易使敏感的数据库信息被非法浏览或删除。事实上，由于SQL注入攻击简单而又非常高效，高级黑客们已开始采用某些软件自动搜索web应用程序的SQL漏洞，并利用SQL注入自动化工具来制造僵尸，并建立可自动攻击的僵尸网络。 [5]

显然，SQL注入攻击并不会在短时间内消失，而其所造成影响更是一个刻不容缓、代价不菲的重大威胁，处理一次web应用安全事故几乎要花掉20万美元。网络安全工程师们务必要意识到，研究与防范SQL注入攻击是必要的，也是首要的安全任务。 [5]

## sql注入注入过程

[编辑](https://baike.baidu.com/item/sql%E6%B3%A8%E5%85%A5/javascript:;)

第一步：SQL注入点探测。探测SQL注入点是关键的一步，通过适当的分析应用程序，可以判断什么地方存在SQL注入点。通常只要带有输入提交的动态网页，并且动态网页访问数据库，就可能存在SQL注入漏洞。如果程序员信息安全意识不强，采用动态构造SQL语句访问数据库，并且对用户的输入未进行有效性验证，则存在SQL注入漏洞的可能性很大。一般通过页面的报错信息来确定是否存在SQL注入漏洞。 [6]

第二步：收集后台数据库信息。不同数据库的注入方法、函数都不尽相同，因此在注入之前，我们先要判断一下数据库的类型。判断数据库类型的方法很多，可以输入特殊字符，如单引号，让程序返回错误信息，我们根据错误信息提示进行判断；还可以使用特定函数来判断，比如输入“1 and version（）>0”，程序返回正常，说明version（）函数被数据库识别并执行，而version（）函数是MySQL特有的函数，因此可以推断后台数据库为MySQL。 [6]

第三步：猜解用户名和密码。数据库中的表和字段命名一般都是有规律的。通过构造特殊SQL语句在数据库中依次猜解出表名、字段名、字段数、用户名和密码。 [6]

第四步：查找Web后台管理入口。WEB后台管理通常不对普通用户开放，要找到后台管理的登录网址，可以利用Web目录扫描工具（如：wwwscan、AWVS）快速搜索到可能的登录地址，然后逐一尝试，便可以找到后台管理平台的登录网址。 [6]

第五步：入侵和破坏。一般后台管理具有较高权限和较多的功能，使用前面已破译的用户名、密码成功登录后台管理平台后，就可以任意进行破坏，比如上传[木马](https://baike.baidu.com/item/%E6%9C%A8%E9%A9%AC/530" \t "https://baike.baidu.com/item/sql%E6%B3%A8%E5%85%A5/_blank)、篡改网页、修改和窃取信息等，还可以进一步提权，入侵Web服务器和数据库服务器。 [6]

## sql注入注入方法

[编辑](https://baike.baidu.com/item/sql%E6%B3%A8%E5%85%A5/javascript:;)

由于编写程序时未对用户输入数据的合理性进行判断，导致攻击者能在SQL Injection的注入点中夹杂代码进行执行，并通过页面返回的提示，获取进行下一步攻击所需的信息。根据输入的参数，可将SQL注入方式大致分为两类：数字型注入、字符型注入。 [7]

1、数字型注入

当输入的参数为整型时，如ID、年龄、页码等，如果存在注入漏洞，则可以认为是数字型注入。这种数字型注入最多出现在ASP、PHP等弱类型语言中，弱类型语言会自动推导变量类型，例如，参数id=8，PHP会自动推导变量id的数据类型为int类型，那么id=8 and 1=1，则会推导为string类型，这是弱类型语言的特性。而对于Java、C#这类强类型语言，如果试图把一个字符串转换为int类型，则会抛出异常，无法继续执行。所以，强类型的语言很少存在数字型注入漏洞。 [7]

2、字符型注入

当输入参数为字符串时，称为字符型。数字型与字符型注入最大的区别在于：数字型不需要单引号闭合，而字符串类型一般要使用单引号来闭合。 [7]

## sql注入攻击特点

[编辑](https://baike.baidu.com/item/sql%E6%B3%A8%E5%85%A5/javascript:;)

SQL注人攻击是目前web应用网络攻击中最常见的手段之一，安全风险较高，在一定程度上超过缓冲区溢出

[](https://baike.baidu.com/pic/sql%E6%B3%A8%E5%85%A5/150289/0/a71ea8d3fd1f4134c1207f0f2a1f95cad1c85eaf?fr=lemma%26ct=single)SQL注入

漏洞，而市场上的防火墙又不能对SQL注入漏洞进行有效的检测和防范。防火墙为了使正常网络应用程序访问服务器端的数据，必须允许从互联网到[Web服务器](https://baike.baidu.com/item/Web%E6%9C%8D%E5%8A%A1%E5%99%A8/8390210" \t "https://baike.baidu.com/item/sql%E6%B3%A8%E5%85%A5/_blank)的正向连接，因此一旦web网络应用程序存在注入漏洞，攻击者就可以获取访问数据库的权利进而获得数据库所在服务器的访问权在某些情况下，SQL注入攻击的风险要高于缓冲区溢出漏洞等所有其他漏洞。SQL注入攻击普遍存在范围广、实现容易、破坏性大等特点。 [8]

SQL注入攻击者在H1TIP请求中输入含有恶意构造的SQL语法且是合法的，只要应用程序中没有做严格的处理那么就会出现SQL注入漏洞危险，目前以PHP、Perl、Cold Fusion Management等技术与Oracle、SQLServer、Sybase、DB2等数据管理系统相结合的Web应用程序都发现有SQL注入漏洞。 [8]

SQL注入技术公布后不久，互联网上出现了很多例如教主的HDSI、NBSI、明小子的Domain等SQL注人工具，对那些存在SQL注入的网站以及Web应用程序进行攻击，很容易就可以获取其服务器的控制权。 [8]

## sql注入攻击手法

[编辑](https://baike.baidu.com/item/sql%E6%B3%A8%E5%85%A5/javascript:;)

1、基于布尔的盲注

因为web的页面返回值都是True或者False，所以布尔盲注就是注入后根据页面返回值来得到数据库信息的一种办法。 [1]

2、基于时间的盲注

当布尔型注入没有结果（页面显示正常）的时候，我们很难判断注入的代码是否被执行，也可以说到底这个注入点存不存在？这个时候布尔型注入就无法发挥自己的作用了。基于时间的盲注便应运而生，所谓基于时间的盲注，就是我们根据web页面相应的时间差来判断该页面是否存在SQL注入点。 [1]

3、联合查询注入

使用联合查询进行注入的前提是我们要进行注入的页面必须有显示位。所谓联合查询注入即是使用union合并两个或多个SELECT语句的结果集，所以两个及以上的select必须有相同列、且各列的数据类型也都相同。联合查询注入可在链接最后添加order by 9基于随意数字的注入，根据页面的返回结果来判断站点中的字段数目。 [1]

4、基于错误信息的注入

此方法是在页面没有显示位，但是echo mysql\_error();函数输出了错误信息的时候方能使用。优点是注入速度快，缺点是语句较为复杂，而且只能用limit依次进行猜解。总体来说，报错注入其实是一种公式化的注入方法，主要用于在页面中没有显示位，但是用echo mysql\_error();输出了错误信息时使用。 [1]

## sql注入检测技术

[编辑](https://baike.baidu.com/item/sql%E6%B3%A8%E5%85%A5/javascript:;)

SQL注入的检测方式目前主要有两大类，第一：动态监测，即在系统运行时，通常在系统验收阶段或上线运行阶段使用该方法，使用动态监测攻击对其系统进行扫描，然后依据扫描结果判断是否存在SQL注入漏洞。第二：静态检测，又称静态代码扫描，对代码做深层次分析。 [5]

1、动态检测

动态监测分为两类：手工监测以及工具监测。相对于手动监测的高成本以及高漏检率，在实际生产过程中更偏向于工具监测，但工具监测同样存在较大的局限性。其原因在于工具是用报文来判断SQL注入是否生效，然而仅仅通过报文是很难精准地判断SQL注入是否存在，因此存在较高的误报率。 [5]

2、静态检测

静态检测的误报率相对较低，其主要原因在于SQL注入漏洞的代码特征较为明显。 [5]

（1）使用数据库交互代码； [5]

（2）使用字符串拼接方式构造动态SQL语句； [5]

（3）使用未过滤的不可信任数据。 [5]

在常规的排查应用系统中是否存在SQL注入漏洞时，由于静态扫描的代码特征明显，误报率低和直接阅读相关代码，工作总量减少的优势，通常使用静态扫描。 [5]

## sql注入注入防范措施

[编辑](https://baike.baidu.com/item/sql%E6%B3%A8%E5%85%A5/javascript:;)

SQL注入攻击的危害很大，而且防火墙很难对攻击行为进行拦截，主要的SQL注入攻击防范方法，具体有以下几个方面。 [9]

1、分级管理

对用户进行分级管理，严格控制用户的权限，对于普通用户，禁止给予数据库建立、删除、修改等相关权限，只有系统管理员才具有增、删、改、查的权限。例如上述实例中用户在查询语句中加入了drop table。肯定是不能让其执行的，否则系统的数据库安全性就无法保障。故而通过权限的设计限制。使得即使恶意攻击者在数据提交时嵌入了相关攻击代码。但因为设置了权限，从而使得代码不能执行。从而减少SQL注入对数据库的安全威胁。 [9]

2、参数传值

程序员在书写SQL语言时，禁止将变量直接写入到SQL语句，必须通过设置相应的参数来传递相关的变量。从而抑制SQL注入。数据输入不能直接嵌入到查询语句中。同时要过滤输入的内容，过滤掉不安全的输入数据。或者采用参数传值的方式传递输入变量。这样可以最大程度防范SQL注入攻击。 [9]

3、基础过滤与二次过滤

SQL注入攻击前，入侵者通过修改参数提交“and”等特殊字符，判断是否存在漏洞，然后通过select、update等各种字符编写SQL注入语句。因此防范SQL注入要对用户输入进行检查，确保数据输入的安全性，在具体检查输入或提交的变量时，对于单引号、双引号、冒号等字符进行转换或者过滤，从而有效防止SQL注入。当然危险字符有很多，在获取用户输入提交的参数时，首先要进行基础过滤，然后根据程序的功能及用户输入的可能性进行二次过滤，以确保系统的安全性。 [9]

4、使用安全参数

SQL数据库为了有效抑制SQL注入攻击的影响。在进行SQLServer数据库设计时设置了专门的SQL安全参数。在程序编写时应尽量使用安全参数来杜绝注入式攻击。从而确保系统的安全性。 [9]

SQLServer数据库提供了Parameters集合，它在数据库中的功能是对数据进行类型检查和长度验证，当程序员在程序设计时加入了Parameters集合，系统会自动过滤掉用户输入中的执行代码，识别其为字符值。如果用户输入中含有恶意的代码，数据库在进行检查时也能够将其过滤掉。同时Parameters集合还能进行强制执行检查。一旦检查值超出范围。系统就会出现异常报错，同时将信息发送系统管理员，方便管理员做出相应的防范措施。 [9]

5、漏洞扫描

为了更有效地防范SQL注入攻击，作为系统管理除了设置有效的防范措施，更应该及时发现系统存在SQL攻击安全漏洞。系统管理员可以通过采购一些专门系统的SQL漏洞扫描工具，通过专业的扫描工具，可以及时的扫描到系统存在的相应漏洞。虽然漏洞扫描工具只能扫描到SQL注入漏洞，不能防范SQL注入攻击。但系统管理员可以通过扫描到的安全漏洞，根据不同的情况采取相应的防范措施封堵相应的漏洞，从而把SQL注入攻击的门给关上，从而确保系统的安全。 [9]

6、多层验证

现在的网站系统功能越来越庞大复杂。为确保系统的安全，访问者的数据输入必须经过严格的验证才能进入系统，验证没通过的输入直接被拒绝访问数据库，并且向上层系统发出错误提示信息。同时在客户端访问程序中验证访问者的相关输入信息，从而更有效的防止简单的SQL注入。但是如果多层验证中的下层如果验证数据通过，那么绕过客户端的攻击者就能够随意访问系统。因此在进行多层验证时，要每个层次相互配合，只有在客户端和系统端都进行有效的验证防护，才能更好地防范SQL注入攻击。 [9]

7、数据库信息加密

传统的加解密的方法大致可以分为三种： [9]

(1)对称加密：即加密方和解密方都使用相同的加密算法和密钥，这种方案的密钥的保存非常关键，因为算法是公开的，而密钥是保密的，一旦密匙泄露，黑客仍然可以轻易解密。常见的对称加密算法有：AES、DES等。 [9]

(2)非对称加密：即使用不同的密钥来进行加解密，密钥被分为公钥和私钥，用私钥加密的数据必须使用公钥来解密，同样用公钥加密的数据必须用对应的私钥来解密，常见的非对称加密算法有：RSA等。 [9]

(3)不可逆加密：利用哈希算法使数据加密之后无法解密回原数据，这样的哈希算法常用的有：md5、SHA-1等。