基于 Web 的 SQL 注入解析及防范

邱俊峰　李　林
四川大学锦城学院,计算机与软件学院　四川　成都　611731)

【摘要】随着现在网络的不断发展,个人或者集体的数据在不断的产生,并且被存储在电脑当中。如果有着比较更加私人的信息被泄露或者是盗取,就会给个人,家庭,甚至是社会带来严重的危害。如何来保护个人的隐私,或者是系统应用的安全就成为了当务之急。网上的黑客利用不同手段的攻击原理来破解或者“偷盗”等恶意行为。SQL 注入就是现在随处可见,而且也是比较常见的一种攻击方式,熟知其原理能够让我们有效的防范这个部分的恶意攻击。

【关键词】SQL 注入,Web 安全,防御措施。

随着互联网的爆发式的发展,Web 的相关运用在生活中随处可见。人们在日常生活中都在无时不刻的产生数据,而这些数据也在不停的被相关的 Web 数据系统所收纳。比如最常见的网上购物,有些人可以通过盗取客户购买的物品信息来进行分析。比如:买了一个奶瓶,别人可能就会得出客户的家里有小孩或者是马上会有小孩等。由于写代码的开发人员的经验和水平参差不齐,而且有很大一部分人在写代码的时候,并没有考虑到安全这方面的问题,大多开发人员的想法基本都是能跑,能跑得快,能满足客户的要求就行。正式由于这样的开发人员不在少数,所以就给了有些心术不正的人可乘之机。

1　什么是 SQL 注入

SQL 注入这个名字,在搞 IT 的行业内基本都知道。它是现在 Web 程序面临的一种恶意攻击,它主要是执行恶意的 SQL 语句。通过一些 SQL 语句来对数据库进行一些操作。用简单易懂的一句话来说就是:程序执行的 SQL 语句和开发人员想让程序执行的 SQL 语句不一样,导致操作结果完全不同,而且一般结果都是严重的。

1.1　SQL 注入的分类(按照输入的类型分类)

数字型注入点:当用户输入的参数为数字类型的时候,比如页码之类的,当输入页码的值为 1 and 1=2 时,会被注入。但是这种一般是出现在弱类型语言之中比如:PHP,ASP。但在强类型语言中则会报错,导致注入失败。

字符串注入:常见的 SQL 都存在利用 where 条件来作为条件约束来对数据库进行增删改查等操作,举个简单的例子:select * from 表名 where name='(这里是条件,一般是用户来输入)',如果用户输入的是' or 1=1 #,那么原来的语句就变成了 select * from 表名 where name='' or 1=1 #'.那么注入后的 SQL 语句的意思就是查询所有的记录,而不是原来的意思:查询 name 为用户输入的名字的信息了。

当然,按照分类的不同还可以用其他标准来分类,我这里只是简单的举例,不做细致的多种分类,主旨是解释一下 SQL 注入的一些形式。

1.2　SQL 注入攻击的主要特点

攻击种类繁多:比较有经验的老手,通常会手动调整攻击参数,致使攻击的数据不能枚举,这样就导致一些传统的特征识别匹配方法不能正确的识别攻击,难以防范。

攻击操作简单:网上不管是付费的还是免费的 SQL 注入工具一大堆,而且能够快速上手,攻击者能够借助这些工具有效的,快速的,简单的对目标 Web 进行攻击或者是破坏。

攻击产生的危害很大:由于 Web 本身的一些缺陷不论是语言,还是本身的配置,而且具有安全意识

的开发人员较少,大多数 Web 系统均有被注入的可能。而攻击者一旦成功注入,就可以对整个 Web 的数据进行盗取,修改甚至是删除,导致的后果也会严重。

1.3 SQL 注入的危害

最严重的就是数据库信息或者是数据泄露:数据库存储的各种用户相关的信息,数据库本身的信息,获取到这些数据的黑客会让数据库更容易被攻击。

网页被恶意修改:黑客可能通过修改数据库的某些信息对特定的网页进行篡改。

网站被挂木马,传播恶意软件:黑客可能会修改数据库的信息,并嵌入带极具威胁的网站链接。

攻击者恶意操作数据库:修改数据库的配置或者是数据,数据库的管理员账号或者是权限被修改,导致数据库服务器无法正常工作。

服务器被黑客远程控制:通常黑客操作完之后一般会留下后门,以便下次“光临”。

黑客得到数据库的权限之后,先导出全表,再破坏数据库,甚至是破坏服务器系统,导致网站崩溃等。

2 SQL 注入的主要防范方法

由于 Web 占据 Internet 的绝大部分,对于这样非常严重的危害我列举出了一些比较常用的抵御手段:

2.1 黑、白名单防护

第一种比较常用的防 SQL 注入方式是利用黑名单和白名单对传入数据库的字段进行过滤。白名单:这是开发人员自定义的,相当于是制定的一个标准:程序只接受用户输入的达到开发人员的标准的数据。当然也有可能当用户输入一个非常复杂的信息,但确实不是恶意的数据,而开发人员定义的标准却判定为恶意,导致用户无法输入正确的信息,所以这种方法比较适合在用户输入较少的情况下,对于输入多的可以采用其他的防范方式来抵御攻击;黑名单也是开发人员制定的标准,他是用来拒绝用户输入的,但是这种标准的效率不高,因为开法人员想到的很有限,但是存在威胁的字符还多的多,不然也不会有那么多 Bug 的存在了。利用数据库原有的这种方法确实有一些效果,但是攻击者也是非常聪明的。他们也会想出更多的手法去绕过这样的一道防线。

2.2 正则表达式与业务逻辑相结合

由于没有对用户的输入进行严格的验证,新的防范技术使用正则表达式和业务逻辑是比较有效的验证方法[,但是正则表达式一般人看不懂,而且要求较高,匹配的字符串比较局限,导致有可能把没有问题的 SQL 语句一起过滤掉了。

2.3 蜜罐系统模型

一种以蜜罐系统为核心的防御模型。蜜罐是一种安全资源,它的使命就是成为未经授权访问或攻击的目标,以此获取攻击者的攻击行为和攻击策略[]。这个系统就相当于是给恶意的攻击者一个陷阱。做一个假的或者是没有意义的业务漏洞,让恶意的攻击者故意走进这个陷阱里面,然后利用记录下来的日志进行分析,然后再把这些恶意的 SQL 语句进行过滤,或者写入黑名单等方式,相当于是让系统自己不断的去学习,不断的进化的,从而屏蔽恶意的攻击。而且正常的用户一般是不会掉到这种虚设的陷阱里面,这也使得用户能够正常使用系统的业务。

2.4 参数化传值

将输入的查询条件参数化:这种方法也是目前用的比较广泛,而且是比较有效的防御方式;例如:

```
SqlConnection conn = new SqlConnection("...");
conn.Open();
SqlParameter a =new SqlParameter("@id", id);
SqlCommand cmd = new SqlCommand(sql, conn);
cmd.Parameters.Add(a);cmd.ExecuteScalar();
conn.Close();
```

上面这段代码的意思就是利用@id 先组成一个组成一个完整的 SQL 语句,你可以想象成是 SELECT * FROM 表名 WHERE ID=‘@id’。这样

一句话去数据库执行之后,然后把用户输入的 id 去和之前查询出来的数据作比对,就不用再去执行 SQL 语句了。这样就避免了把用户输入的信息拿去数据库当做 SQL 语句执行,这样可以有效的避免注入的可能,是目前最优的选择之一。但是什么情况都不是绝对的,它虽然可以抵挡大部分攻击,但是终究还是不能做到万无一失的防守。恶意的攻击者可以通过一些抓包工具来进行抓包,然后来修改包里面的参数。不仅如此,一些在与数据库交互的位置,恶意攻击者会把攻击代码插入到 HTML 里面,这样恶意代码就会被存进数据库,当用户访问 Web 的其他位置时,就产生了 SQL 注入。

3 二进制存储数据

把用户输入的数据采用二进制的形式存储:会存在 SQL 注入攻击成功的原因之一是:数据库中的取出来的数据与用户输入的信息都是字符型的数据,没有比较明确的界限。而且电脑始终只是机器,没办法来区分两者到底哪个才是用户输入的,用户输入的信息到底是恶意的还是正常的。由于电脑无法区分,它就遵循凡是只要是自己认识的语句就执行这样一个准则,从而改变了预期 SQL 语句的逻辑去执;如果利用二进制来存储用户输入的数据和数据库的数据,那么数据库在执行 SQL 语句的时候就能明确的区分谁是谁,SQL 注入的发生就会减少很多。开发语言之中也有相关的方法来把数据转化成二进制数据,而且也相对简单。但是在数据库中存储二进制数据会让数据库维护人员的工作变得困难了,毕竟人还是对中文或者是字符串熟悉一点。

3.1 使用存储过程

它是将要执行的语句提前写好,并保存进数据库里,需要使用的时候可以直接由程序调用,使用相对比较安全的存储过程可增加 SQL 注入的抵御能力,但是如果过滤的不够透彻,也可能会存在 SQL 注入的情况。

3.2 使用数据库的预编译原理

它主要有两个作用,一是提高数据的查询效率,比如某条 SQL 语句比较长,功能比较复杂,或者是需要反复被执行的时候只需要创建一个预编译的 SQL 语句给数据库解析,在程序需要调用的时候传入参数即可。二是提高了安全性。用户输入的数据是直接以纯文本的形式发送给数据库,就不用了在代码里面拼接 SQL,这样形成了代码与数据分离开来,从理论上是可以杜绝 SQL 注入的。

3.3 漏洞扫描

对于公司的网络管理员来说,利用一些漏洞扫描的工具(有付费的,也有免费的)定期的对系统扫描,这样就是及时的发现 Web 的安全漏洞,相关人员也就能够有针对性行的才去措施,从而有效避免被 SQL 注入的风险。比如:SQLIer、SQLMap、SQLNinja 等。我用过 SQLMap,不得不说它里面的东西真的非常的丰富。只是在扫描的时候要注意一下扫描的速度,不能过快,因为速度过快之后可能会导致服务器处理不过来,出现 time out。

4 结束语

由于现在的 Web 发展日新月异,各种 Web 服务器的漏洞与程序不够严密,恶意的攻击手段也是层出不穷,SQL 注入正在慢慢的成为主流的 Web 攻击方式。越来越多的 Web 遭受到攻击,致使个人或者团体的利益受到严重损坏。身为一个开发人员,需要更加深入的去了解 SQL 注入攻击或者漏洞产生的原因,并仔细研究 SQL 注入的检测和防范的方法,从根本上去保护数据不受恶意的侵害和偷盗。