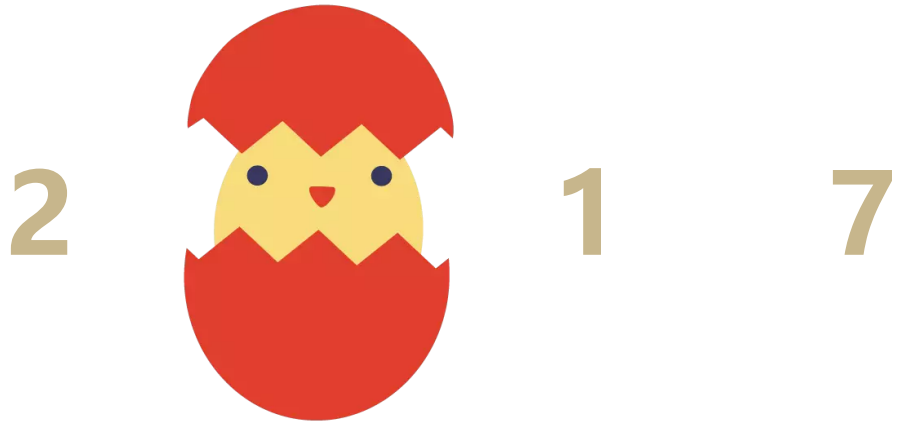
# 详解XPath注入—安全小课堂第四十二期

京东安全应急响应中心 2017-01-13




**安全小课堂第四十二期**

XPath注入攻击主要是通过构建特殊的输入，这些输入往往是XPath语法中的一些组合，这些输入将作为参数传入Web 应用程序，通过执行XPath查询而执行入侵者想要的操作。本期邀请到JSRC白帽子恋锋为大家分享交流~

**恋锋：JSRC白帽子，具备多年web及app安全测试经验，在代码安全审计领域有较深入的研究。**

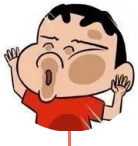
文丨豌豆妹　图丨源自网络

> > > 1 < < <

豌豆妹

咱们先聊聊XPath注入漏洞存在的原因呗？

恋峰

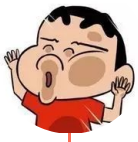


其实XPath注入漏洞的成因与其他注入类漏洞有相似之处，主要由于系统未对输入内容进行严格校验和检查而产生，导致黑客通过提交符合XPath语法的恶意代码，即可实现攻击。

豌豆妹

除了“主要是因为系统未对输入内容进行严格校验和检查产生”这一原因，还有其他原因么？

恋峰

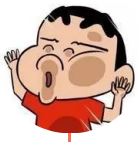


还有就是：XPath解析器的松散输入和容错特性。

豌豆妹

能解释下XPath解析器的松散输入和容错特性么？

恋峰



就是XPath解析器本身对URL、表单中提交的代码内容未做严格限制，导致恶意代码可以直接解析执行。

> > > 2 < < <

豌豆妹

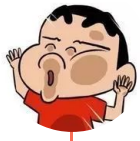
XPath注入存在的危害呢？

恋峰

一是，在URL及表单中提交恶意XPath代码，可获取到权限限制数据的访问权，并可修改这些数据；二是，可通过此类漏洞查询获取到系统内部完整的XML文档内容。

豌豆妹

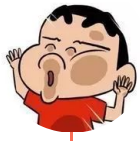
恶意的XPath代码可以提供一些具体举例么？比如要获取数据的语法是什么样，修改数据的语法是什么样的。

恋峰 

举个例子：
//users/user[loginID/text()='abc' and password/text()='test123'] 。
这是一个XPath查询语句，获取loginID为abc的所有user数据，用户需要提交正确的loginID和password才能返回结果。如果黑客在 loginID 字段中输入：' or 1=1 并在 password 中输入：' or 1=1　就能绕过校验，成功获取所有user数据。

豌豆妹

那比如我想要获取所有用户的ID和密码语法应该如何操作呢？XPath的语法和SQL的语法应该有差异吧？

恋峰 

这样即可：//users/user[LoginID/text()=''or 1=1 and password/text()=''or 1=1]。

> > > 3 < < <

豌豆妹

XPath注入与一般注入的区别有哪些呢？

恋峰

与其他注入相比有两点区别：

1、广泛性：XPath注入攻击利用的是XPath语法，由于XPath是一种标准语言，因此只要是利用XPath语法的Web 应用程序，如果未对输入的XPath查询做严格的处理，都会存在XPath注入漏洞，所以可能在所有的XPath实现中都包含有该弱点，这和SQL注入攻击有很大区别。在SQL注入攻击过程中根据数据库支持的SQL语言不同，注入攻击的实现可能不同。

2、危害性：XPath语言几乎可以引用XML文档的所有部分，而这样的引用一般没有访问控制限制。但在SQL注入攻击中，一个“用户”的权限可能被限制到某一特定的表、列或者查询，而XPath注入攻击可以保证得到完整的XML文档，即完整的数据库。只要Web服务应用具有基本的安全漏洞，即可构造针对 XPath应用的自动攻击。

> > > 4 < < <

豌豆妹

XPath注入攻击原理，能否以实例进行说明？

恋峰

XPath注入攻击主要是通过构建特殊的输入，这些输入往往是XPath语法中的一些组

合，这些输入将作为参数传入Web 应用程序，通过执行XPath查询而执行入侵者想要的操作。

攻击案例：

以下为一个XPath查询语句，获取loginID为abc的所有user数据，用户需要提交正确的loginID和password才能返回结果。

//users/user[loginID/text()='abc' and password/text()='test123']。

如果黑客在loginID字段中输入：'or 1=1并在password中输入：'or 1=1就能绕过校验，成功获取所有user数据。

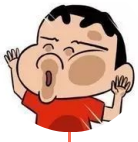
//users/user[LoginID/text()=''or 1=1 and password/text()=''or 1=1]。

> > > 5 < < <

豌豆妹

介绍下Xpath注入的防御方法吧。

恋峰



说到防御，我们结合漏洞的成因给出，主要有以下4个方面：

1、服务端对输入内容的合法性进行验证，检查提交的数据是否包含特殊字符，对特殊字符进行编码转换。

2、对于系统出现的错误信息，使用统一的错误页面，屏蔽系统本身的出错信息。

3、参数化XPath查询，将需要构建的XPath查询表达式，以变量的形式表示，变量不是可以执行的脚本。

4、通过MD5、SSL等加密算法，对于数据敏感信息和在数据传输过程中加密，即使某些非法用户通过非法手法获取数据包，看到的也是加密后的信息。

总结下就是：限制提交非法字符，对输入内容严格检查过滤，参数化XPath查询的变量。

长 按 关 注

官方微信号 : jsrc_team

新浪微博：京东安全应急响应中心