Struts2漏洞利用原理及OGNL机制研究

cryin / 2017-03-20 03:19:00 / 浏览数 6480 技术文章 技术文章 顶(0) 踩(0)

概述

在MVC开发框架中，数据会在MVC各个模块中进行流转。而这种流转，也就会面临一些困境，就是由于数据在不同MVC层次中表现出不同的形式和状态而造成的：

View层—表现为字符串展示

数据在页面上是一个扁平的、不带数据类型的字符串，无论数据结构有多复杂，数据类型有多丰富，到了展示的时候，全都一视同仁的成为字符串在页面上展现出来。数据在

Controller层—表现为java对象

在控制层，数据模型遵循java的语法和数据结构，所有的数据载体在Java世界中可以表现为丰富的数据结构和数据类型，你可以自行定义你喜欢的类，在类与类之间进行继承

可以看到，数据在不同的MVC层次上，扮演的角色和表现形式不同，这是由于HTTP协议与java的面向对象性之间的不匹配造成的。如果数据在页面和Java世界中互相传递，

1．当数据从View层传递到Controller层时，我们应该保证一个扁平而分散在各处的数据集合能以一定的规则设置到Java世界中的对象树中去。同时，能够灵活的进行由字符

2．当数据从Controller层传递到View层时，我们应该保证在View层能够以某些简易的规则对对象树进行访问。同时，在一定程度上控制对象树中的数据的显示格式。

我们稍微深入思考这个问题就会发现，解决数据由于表现形式的不同而发生流转不匹配的问题对我们来说其实并不陌生。同样的问题会发生在Java世界与数据库世界中，面对

现在在Web层同样也发生了不匹配，所以我们也需要使用一些工具来帮助我们解决问题。为了解决数据从View层传递到Controller层时的不匹配性，Struts2采纳了XWork的
技术内幕)。

关于OGNL

OGNL（Object Graph Navigation
Language）即对象图形导航语言，是一个开源的表达式引擎。使用OGNL，你可以通过某种表达式语法，存取Java对象树中的任意属性、调用Java对象树的方法、同时能够
OGNL进行对象存取操作的API在Ognl.java文件中，分别是getValue、setValue两个方法。getValue通过传入的OGNL表达式，在给定的上下文环境中，从root对象里取值

setValue通过传入的OGNL表达式，在给定的上下文环境中，往root对象里写值:

OGNL同时编写了许多其它的方法来实现相同的功能，详细可参考Ognl.java代码。OGNL的API很简单，无论何种复杂的功能，OGNL会将其最终映射到OGNL的三要素中通

OGNL三要素

Expression(表达式)：

Expression规定OGNL要做什么，其本质是一个带有语法含义的字符串,这个字符串将规定操作的类型和操作的内容。OGNL支持的语法非常强大，从对象属性、方法的访问到

Root(根对象)：

OGNL的root对象可以理解为OGNL要操作的对象，表达式规定OGNL要干什么，root则指定对谁进行操作。OGNL的root对象实际上是一个java对象，是所有OGNL操作的

Context(上下文):

有了表达式和根对象，已经可以使用OGNL的基本功能了。例如，根据表达式对root对象进行getvalue、setvalue操作。
不过事实上在OGNL内部，所有的操作都会在一个特定的数据环境中运行，这个数据环境就是OGNL的上下文。简单说就是上下文将规定OGNL的操作在哪里进行。
OGNL的上下文环境是一个MAP结构，定义为OgnlContext，root对象也会被添加到上下文环境中，作为一个特殊的变量进行处理。

OGNL基本操作

对root对象的访问：

针对OGNL的root对象的对象树的访问是通过使用'点号'将对象的引用串联起来实现的。通过这种方式，OGNL实际上将一个树形的对象结构转化为一个链式结构的字符串来

```
//■■root■■■■name■■■
name
//■■root■■department■■■■name■■■
department.name
//■■root■■department■■■■manager■■■■name■■■
department.manager.name
```

对上下文环境的访问:

Context上下文是一个map结构，访问上下文参数时需要通过#符号加上链式表达式来进行，从而表示与访问root对象的区别，如下：

```
//■■■■■■■■■■introduction■■■■■
introduction
//■■■■■■■■■■parameters■■■■■user■■■■■name■■■■
\#parameters.user.name
```

对静态变量、方法的访问:

在OGNL中，对于静态变量、方法的访问通过@[class]@[field/method]访问，这里的类名要带着包名。如下

```
//■■com.example.core.Resource■■■■img■■■■
@com.example.core.Resource@img
//■■com.example.core.Resource■■■■get■■■
@com.example.core.Resource@get()
```

访问调用:

```
//■■root■■■■group■■■users■size()■■
group.users.size()
//■■root■■■group■■containsUser■■■■■■■■■■■■■■user■■■■■
group.containsUser(#user)
```

构造对象

OGNL支持直接通过表达式来构造对象，构造的方式主要包括三种：

```
1■■■List■■■{}■■■■■■■■■■■■■■■■■■
2■■■map■■■#{},■■■■■■■■■■■■■■■■■■■■key■value
3■■■■■■■■■■■■■■■■■■■■■■■■■■
```

可以看到OGNL在语法层面上所表现出来的强大之处，不仅能够直接对容器对象构造提供语法层面支持，还能够对任意java对象提供支持。然而正因为OGNL过于强大，它也

深入OGNL实现类

在OGNL三要素中，context上下文环境为OGNL提供计算运行的环境，这个运行环境在OGNL内部由OgnlContext类实现，它是一个map结构。在OGNL.java中可以看到Og

这里看到MemberAccess属性，在struts2漏洞利用中经常看到。转到OgnlContext类中查看。

MemberAccess指定OGNL的对象属性、方法访问策略，这里初始默认情况下是flase，即默认不允许访问。在struts2中SecurityMemberAccess类封装了DefaultMeberAc

另外在OGNL实现了MethodAccessor及PropertyAccessor类规定OGNL在访问方法和属性时的实现方式，详细可参考代码文件。

在struts2中，XWorkMethodAccessor类对MethodAccessor进行了封装，其中在callStaticMethod方法中可以看到，程序首先从上下文获取到DENY_METHOD_EXECUT

Struts2漏洞利用原理

上文已经详细介绍了OGNL引擎，因为OGNL过于强大，它也造成了诸多安全问题。恶意攻击者通过构造特定数据带入OGNL表达式即可能被解析并执行，而OGNL可以用来

虽然Struts2出于安全考虑，在SecurityMemberAccess类中通过设置禁止静态方法访问及默认禁止执行静态方法来阻止代码执行。即上面提及的denyMethodExecution为t
POC如下：

```
http://mydomain/MyStruts.action?('\u0023_memberAccess[\'allowStaticMethodAccess\']')(meh)=true&(aaa)(('\u0023context[\'xwork.M
```

在POC中\u0023是因为S2-003对#号进行了过滤，但没有考虑到unicode编码情况，而通过#_memberAccess[\'allowStaticMethodAccess\']')(meh)=true，可以设置允许

参考

[1] Struts2 技术内幕-深入解析Struts2架构设计与实现原理

[2] http://blog.csdn.net/u011721501/article/details/41610157

[3] http://www.freebuf.com/articles/web/33232.html

[4] http://blog.csdn.net/three_feng/article/details/60869254

[5] https://github.com/apache/struts/

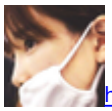
[6] http://www.secbox.cn/hacker/program/205.html

[7] http://struts.apache.org/docs/s2-005.html

[8] http://archive.apache.org/dist/struts/

上一篇：SDL软件安全设计初窥 下一篇：Php写入配置文件的经典漏洞

1. 1 条回复


hades 2017-03-20 09:14:44

幸苦了哈

0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录