绿盟科技网络安全攻防实验室安全研究员廖新喜：Java JSON反序列化之殇

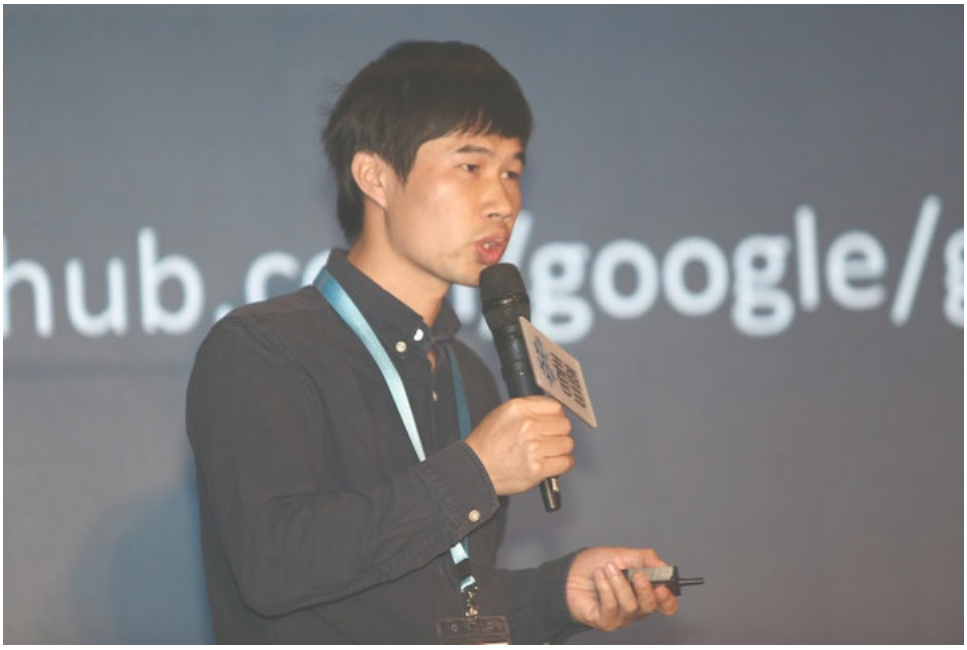
原创

csdn业界要闻 于 2017-12-01 16:45:09 发布 1748 收藏 1
文章标签： 廖新喜 绿盟科技 看雪安全开发者峰会 安全
版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。
本文链接：https://blog.csdn.net/csdn_bang/article/details/80133058
版权

11月18号，2017看雪安全开发者峰会在北京悠唐皇冠假日酒店举行。来自全国各地的开发人员、网络安全爱好者及相应领域顶尖专家，在2017看雪安全开发者峰会汇聚一堂，只为这场“安全与开发”的技术盛宴。

随着REST API的流行，JSON的使用也越来越多，但其中存在的安全问题却不容忽视，特别是由于反序列化导致的远程代码执行更是威力十足。虽然反序列化漏洞出来已有一段时间，前期的一些防御方案随着时间的推移不再有效，但是却传播广泛。

绿盟科技网络安全攻防实验室安全研究员廖新喜为大家阐述了由Java JSON库的反序列化特性导致的RCE，议题内容涉及Gson、Jackson和Fastjson这三个最常用的JSON序列化库的序列化和反序列的操作，并分析其安全机制，从安全机制上发现其潜在的安全漏洞。另外，他还公布了部分未公开的反序列化payload、0day。



绿盟科技网络安全攻防实验室安全研究员廖新喜

> 廖新喜（xxlegend），绿盟科技网络安全攻防实验室安全研究员，擅长代码审计，Web漏洞挖掘，拥有丰富的代码审计经验，曾在Pycon 2015 China大会上分享Python安全编码。安全行业从业六年，做过三年开发，先后担任绿盟科技极光扫描器的开发和开发代表。目前专注于Web漏洞挖掘、Java反序列化漏洞挖掘。曾向RedHat、Apache、Amazon和Oracle提交多份漏洞报告。2016年网络安全周接受央视专访。《谁动了我的VIP账号？》

以下为演讲速记：

廖新喜：反序列化漏洞在这两年影响非常广泛，主持人也介绍了，今天我主要讲Java json 反序列化，也会讲到Java反序列化防御。我叫廖新喜，来自绿盟科技。首先介绍一下json，再介绍json安全特性，接着会介绍Fastjson 反序列的PoC。最后讲解Java防序列化防御。json是什么，就是一个大的数据结构，有一些键值对。Gson，是谷歌公司发布的一个开源代码的Java库，用于序列化和反序列化，序列化即将Java对象转换成JSON结构，而反序列刚好相反，则是将JSON字符串结构转换成Java对象。这是GSON应用的的实例，Gson提供了toJson与fromJson两个转换函数,实现JSON字符串和Java对象的转换。

Fastjson是由阿里巴巴开发，号称速度非常快。提供两个主要接口toJsonString和parseObject来分别实现序列化和反序列化。使用方法介绍完了，进入今天的正题，JSON的安全特性。首先我们看一下GSON，会用到默认的构造函数，如果没有找到的话会调用sun.misc.Unsafe生成一个实例，Gson默认只会反序列化那些基本类型，比如String，Date，URL等，如果反序列化的类型不在这个之内就需要程序员自己实现其反序列化机制。对于基本类型都是通过反射直接调用Field.set方法来实现，这一块应该不存在安全问题，也是推荐使用的。

Jackson的安全特性，无参默认构造方法，不会序列化非Public属性。这一块其实还有很多绕过，我们看一下Jackson和Fastjson，序列化和反序列化非常相似，也需要一个无参的默认方法，也有Jackson特有的属性，需要通过enableDefaultTyping方法来打开支持多态的特性。Fastjson也有一个@type来指定具体的反序列化类。

分析完JSON安全特性之后，我们可以看一下Fastjson的结构，主要是涉及反序列化这一块。最外层有一个门面类JSON提供了一些静态方法，其具体功能都是由DefaultJSONParser实现的，在DefaultJSONParser中引用了ParserConfig，主要保存反序列的一些配置。还引用了JSONLexer，处理JSON字符流。在DefaultJSONParser中还会调用一些Deserializer类，完成JSON字符串到Java对象的转换。反序列化过程中的主要工作都是在JavaBeanDeserializer完成。

前面是安全特性，下面介绍Fastjson PoC。可以分为两大类，一个是基于TemplateImpl，还有就是基于JNDI，基于JDNI的又可以分为两个小类，分别是Bean Property类型和Field类型，这两个小子类的区别就在于是否是通过setter方法触发。我在https://github.com/shengqi158/fastjson-remote-code-execute-poc上面放了一个Demo，大家可以下载研究。首先看基于TemplateImpl的利用链，先构造一个类Test.java继承自AbstractTranslet，在其构造方法中添加恶意代码，这里是弹出计算器。然后重写写几个transform方法。这一页ppt是完整的利用代码，首先将@type设置为”com.sun.org.apache.xalan.internal.xsltc.trax.TemplatesImpl”，再设置_typecodes为恶意代码，即Test.java编译后的文件Test.class。_name和_tfactory为辅助代码，_outputProperties则是触发代码，在反序列化过程中Fastjson会调用TemplateImpl.getProperties()方法。红框中的Feature.SupportNonPublicField则是触发条件。这一页就是展示反序列化过程的调用栈，最外层是Java应用，会调用到Fastjson的parse方法，如果传入的是刚才讨论的PoC的化，执行流程就是先实例化一个TemplateImpl类，设置_name,设置_tfactory，调用getOutputProperties(),最后执行恶意代码。

说到JNDI，2016年的blackhat大会上讲到了，但是对于json这一块没有涉及。JNDI就是Java命令和目录接口，提供了很多实现方式，主要有RMI，LDAP，CORBA等。我们可以看一下它的架构图，JNDI提供了一个统一的外部接口，底层SPI则是多样的。在使用JNDIReferences的时候可以远程加载外部的对象。如果说其lookup方法的参数是我们可以控制的，可以将其参数指向我们控制的RMI服务，切换到我们控制的RMI/LDAP服务等等。这段代码主要讲到了在1099端口上创建一个RMI服务，RMI的内容则是通过外部的http服务地址获取。在客户端则是将lookup的地址指向刚才我们创建的RMI服务，即能达到远程代码执行的目的。

攻击流程：首先攻击者准备Rmi服务和web服务，攻击者将Rmi绝对路径注入Lookup方法中，受害者JNDI接口会指向攻击者控制的RMI服务器，JNDI接口将执行外部加载类的构造函数，把那个加载进来进行初始化，初始化的过程中就会执行我们的恶意代码。另外一种类型，基于Field的类型，不需要setter方法，利用HashSet触发，Fastjson默认处理Set类型都是通过HashSet来实现，通过epuals方法触发。一般通过Field类型都是利用Collection的equals(),toString(),hashCode()来触发的。

这个是调用栈，一个类的加入到会调用HashSet.add()，然后跑到putval，再到这个加载的equals方法，最后执行到lookup方法。而基于BeanPorperty类型的则是通过setter或者getter方法触发，最后调用到lookup方法。

今天我们主要讲这个json反序列化防御和Java反序列化防御，这一块对于开发者来说是最需要的。GSON基本无安全风险。Jackson如果不打开enableDefaultTyping也没有风险。如果一定要实现多态的特性，可以在具体类上加上jsonTypeInfo注解，这样子类也生效，在反序列化的时候，就不存在RCE的问题。Fastjson不起用Autotype就没问题。其他的json库，因为是非主流，还是不建议使用，里面的问题更多。

这是对于json的想法，因为现在都是堵RCE的漏洞，堵远程代码执行的漏洞，但是对于DOS或者一些其他漏洞可能不是太关注，像json或者Fastjson里面有很多的漏洞，但是没有精力关注。

Java反序列化防御：主要从三个方面来讲：1，过时的建议。2，错误的建议。3，正确的建议。这是反序列化利用时序图，整个反序列化利用的过程，首先我们解释一下，Java应用调用readObject，这一块就会实现类的实现化。像我们现在的主流机制，阻止这种方案都是在这一块实行的，其实他有很多这样的接口，这个接口的话还包括resolveproxyclas。最后返回，返回之后再强制转换，这个时候其实相对来说已经晚了。因为前面已经去完成了反序列化了。最后，会有一个垃圾回收机制，我们也可以看到Gadget还是非常多的，这个图来说我们可以看到如果要防御的话怎么防御呢？首先是在实例化之前阻拦，另外实例化之后，如果说这个Gadget是第三方库要以来依赖，把这个一些方法之前加上白名单或者黑名单的控制，这样可以实现第三方库以及自己一些黑名单的控制，这是整个反序列化利用过程主流方案，目前也是如此。

过时的建议：都是我从网上抄下来的，这个过时的建议当时是2015年，现在有两年多的时间了，大家当时都是按照这样方法做的防御，但是这种方法现在来看是过时了。第一种方法，使用Serialkiller替代进行序列化操作的objectinputstream类。另外，建议临时删除项目库，这是我们常常用的commmons-collections，这是所有反序列化出现最多的，这种建议还是被大家广泛传播，也没有变弱的趋势，因为搜索引擎都收录了。还有使用grep的方法把那个commons方法删除了，我们看一下这个方法为什么是无效的呢？因为POC出现多达29种，实在太多了。我也跟了很多种，一起POC仅仅依赖于JDK，跟第三方没有关系。还有一些问题，第三方库Gadget和应用方在打架，这不是我的问题，应用方也说这不是我的问题。到底是谁的问题呢？自从这个漏洞爆发影响面非常广了之后，大家说这是我们的问题，我们都来解决这个问题。

典型的错误方案：大厂的安全编码规范，为什么拿出来讲，因为这个厂非常大，但是安全编码规范居然是错的，可想而知，很多人到现在没有理解这个反序列化漏洞。为什么是错的？我也是推荐通过加密实现这个反序列化的机制，但是这存在一个问题主要问题，首先是反序列化，之后再解密。解密之后再得到Gadget，也是一个反序列化的过程。在第一次反序列化的时候可能存在着问题，因为这个接口的服务不光光是开在内部，可能说我是一个服务的方式来展现了出来，这种就会存在着问题。

错就错在反序列在前，解密在后。首先的话要把数据解密，解密完了之后再反序列化，不然那个数据还是被控的。我也希望这个大厂看到之后把他们的代码规范给改了。一些其他的建议：不要反序列化不可信的数据，反序列建议加密、签名，我们建议的话，加密签名是在之前，反序列化之后，这样反序列化的东西才不会被用户控制。比如说缓解这个漏洞的方式，给反序列化接口添加一些认证授权，还有反序列化接口只允许舰艇在本地，没有对外开通服务或者开通相应防火墙加控制。当然了，这些都是缓解。目前业内主流方案，采用LOOK-Ahead Object，两种解决方法，一种是serialkiller还有一种是contrast Security Contrast-RO0。基本上每出现一个新的bypass它的规则都会进行更新，如果没有跟上它的脚步，就存在着问题。2015-2017年的时间，反序列化POC太多了，加注了很多的黑名单进来。所以说一定要紧急更新，跟上bypass的步伐，重新提一下SerialKiller这种方式，前面也讲过是错误的，现在又为什么推荐了，这里强调的核心就是紧跟更新，随时保持更新。

实现方式，其实很简单。前面已经介绍了，反序列化就是在resolveclass之前进行控制，在判断它是不是在我的黑名单里面。

还有一些其他的建议，我们升级第三方库。刚开始Gadget或者其他应用方都不承认是自己的问题，自从这个漏洞大了之后大家都说是自己的问题，然后推出相应更新机制。一个是Apache Commons Collections，要设置一个系统参数才可以应用反序列化，默认已经不存在这个漏洞。像apache Commons Fileupload，就是默认不打开，这些漏洞可以得到缓解，还有一些第三方的库不承认是自己的问题，也没有修改，就不提了。还有升级JDK，它是加的一个过滤机制，在JEP290实现方案中会加入一些黑名单，把一些常用的黑名单放在里面。你也可以添加自己的白名单。

谢谢大家。

> 注：本文根据大会主办方提供的速记整理而成，不代表CSDN观点。

2017看雪安全开发者峰会更多精彩内容：

- 2017看雪安全开发者峰会在京召开 共商网络安全保障之策
- 中国信息安全测评中心总工程师王军：用技术实现国家的网络强国梦
- 兴华永恒公司CSO仙果：Flash之殇—漏洞之王Flash Player的末路
- 中国婚博会PHP高级工程师、安全顾问汤青松：浅析Web安全编程
- 威胁猎人产品总监彭巍：业务安全发展趋势及对安全研发的挑战
- 启明星辰ADLab西南团队负责人王东：智能化的安全——设备&应用&ICS
- 自由Android安全研究员陈愉鑫：移动App灰色产业案例分析与防范
- 腾讯反病毒实验室安全研究员杨经宇：开启IoT设备的上帝模式
- 绿盟科技应急响应中心安全研究员邓永凯：那些年，你怎么写总会出现的漏洞
- 腾讯游戏安全高级工程师胡和君：定制化对抗——游戏反外挂的安全实践
- 阿里安全IoT安全研究团队Leader谢君：如何黑掉无人机