CVE-2012-0507

Update：本来想发一下CVE-2012-0507的分析，早晨起来发现instruder发了一篇简要分析，重复发一下应该问题不大吧。

漏洞成因

CVE-2012-0507漏洞主要是AtomicReferenceArray类的set方法存在安全问题导致的：

> AtomicReferenceArray.set
>
> |_____unsafe.putObjectVolatile

sun.misc.putObjectVolatile的具体使用可参见OpenJDK中的描述。或者可以参见Oracle JDK中的相关实现，不过Java的逻辑漏洞感觉没必要上升到代码层处理，因此大致看一下描述即可：

> public native void putObjectVolatile(Object o, long offset, Object x)

此函数功能未，将一个参考值存入指定变量中。

这里需要注意一个问题：由于Java是基于OOP思想建立的，所有的类型都是由基本Object扩展而来。为了增强代码的复用性，在实现一些功能时，会将传入的对象按照基本类型Object处理。但是按照类型安全的原则，不同类型的对象是无法直接进行互相操作的，如一个String类型是无法直接连接一个int型，除非对两者中一个对象进行转化。于是当出现直接对Object进行操作时，出于安全考虑，则需要对传入对象进行类型检查。

若未对该类型进行检查，则会导致可将不同类型A和B进行混淆的类型混淆漏洞。

漏洞利用技巧

比较通用的技巧是调用ClassLoader进行利用，这是由于ClassLoader拥有一个非常不错的方法：defineClass。

> protected final Class<?> defineClass(String name,byte[] b,int off,int len,ProtectionDomain protectionDomain)

其中ProtectionDomain的定义为：

> ProtectionDomain(CodeSource codesource, PermissionCollection permissions)

即，可通过defineClass对已有类或实例定义其权限，通过设定对应权限，即可绕过权限管理器，也就是达到了所谓的Bypass sandbox。