Oracle 10g 利用java命令执行的一些蛋疼的碎碎念

净身刀 / 2017-03-20 02:17:00 /

Oracle 10g 利用java命令执行的两个方法

这个网上太多了，不详细说。前提都是 10g，仅有create session或者其他普通权限（其实还是sys用户比较好使）。

1. 使用SYS.DBMS_EXPORT_EXTENSION函数。在oracle上创建Java包，里面创建执行命令函数。这个我将在后面的payload里详细说。
2. 使用 dbms_xmlquery.newcontext() public role

linux 上的 /bin/bash 于 windows 的 cmd.exe 对于 Runtime.getRuntime().exec() 函数的差异

Runtime.getRuntime().exec共有六个重载方法：

- public Process exec(String command)

在单独的进程中执行指定的字符串命令。

- public Process exec(String [] cmdArray)

在单独的进程中执行指定命令和变量

- public Process exec(String command, String [] envp)

在指定环境的独立进程中执行指定命令和变量

- public Process exec(String [] cmdArray, String [] envp)

在指定环境的独立进程中执行指定的命令和变量

- public Process exec(String command,String[] envp,File dir)

在有指定环境和工作目录的独立进程中执行指定的字符串命令

- public Process exec(String[] cmdarray,String[] envp,File dir)

在指定环境和工作目录的独立进程中执行指定的命令和变量

这里只讨论前两种：
在linux中推荐使用 public Process exec(String [] cmdArray) 这种重载方法，

```
String[] cmd=new String[3];
cmd[0]="/bin/bash";
cmd[1]="-c";
cmd[2]=args;
Runtime.getRuntime().exec(cmd);
```

这样可以获得一个非交互式的shell，否则如果只传入一个字符串的话，只能是执行那一条命令的作用，无法使用bash语法，而'-c'参数可以调用其他指令。

在windows中，还是推荐使用 public Process exec(String command) 语法，因为cmd和bash一个很大不同是 /c 参数的调用分为两种情况，一种是 dos 内部指令，另一种是执行其他可执行文件。但是按照日常对cmd的使用来说，就算是执行其他可执行文件这种情况，比如 systeminfo 或者 whoami 命令（在system32下的可执行文件），也不会出现 cmd /c 于 cmd /c start 这两种执行方式混为一谈的情况。但是就我实际测试发现，在某些环境下，比如 oracle 10.2.0.1.0 + windows 2003 ，在使用oracle内置java执行exec时，如果 cmd 调用了可执行文件，那么会另起一个进程，而此进程的输入输出流会莫名其妙的不知所踪，无法捕获到。所以这里应该尽量使用 String command 这个重载的方法，将这两种形式区分开来。

oracle 与 java 执行上的坑

一般来说在oracle中存储字符串类型是使用 varchar2 类型，并且这个类型可以与java的String类型隐式转化，兼容性比较好，所以一般payload都会使用这个。
如果对于一般的操作，比如可以反弹shell，或者可以web绝对路径写马，这种情况使用varchar2完全够用，但是只要是碰上oracle了，一般这个站不是这么好啃，站库分离 varchar2 就不够用了，因为varchar2最多只支持4000 bytes，如果你执行命令的输出超过了这个值，oracle执行会出错。4000看似很大，其实随便 ps 一下，netstat一下，或者读取个 .bash_history 就超过了。
这里有两种解决方案，要么分块读，要么换一个更大的数据类型，oracle中有存储大文本文件的类型 CLOB 和存储大二进制文件的类型 BLOB，本来我尝试使用 BLOB 类型，但是最后还是在java类型和 oracle blob 类型的转化上跳进去没有出来。
这是我研究到最后的payload，但是还是无法编译通过，时间太晚了，不想看也不想写了，就这样吧，谁如果弄出来的话求告诉我一声，雪靴您！！

```
select SYS.DBMS_EXPORT_EXTENSION.GET_DOMAIN_INDEX_TABLES('FOO','BAR','DBMS_OUTPUT".PUT(:P1);EXECUTE IMMEDIATE ''DECLARE PRAGMA
```

```
select SYS.DBMS_EXPORT_EXTENSION.GET_DOMAIN_INDEX_TABLES('FOO','BAR','DBMS_OUTPUT".PUT(:P1);EXECUTE IMMEDIATE ''DECLARE PRAGMA

select SYS.DBMS_EXPORT_EXTENSION.GET_DOMAIN_INDEX_TABLES('FOO','BAR','DBMS_OUTPUT".PUT(:P1);EXECUTE IMMEDIATE ''DECLARE PRAGMA

select UTL_RAW.cast_to_varchar2(sys.BBigRunCmd('cmd /c type c:\test\1.txt')) from dual;
```

最大的问题还是如何把string的byte数组存入 oracle.sql.BLOB
，根据手册中说的，可以使用setBytes，但是有个很操蛋的问题是，空的BLOB对象是不能操作的，编译可以通过，但是执行时，如果使用`BLOB.getEmptyBLOB()`这个静态
oracle 连接对象中获得。。。。这样太麻烦了，而且我并没有找到如何使一个`EmptyBLOB`变的可操作的方法。。。。希望有dalao可以解答。

最后的非交互式 shell 客户端

所以如果这个方法没解决的话，我们只能用那种比较弱智的方法了，我撸了一个 非交互式shell 的客户端，用该客户端结合oracle union
注入，基本可以得到一个较稳定的非交互式shell，并可以上传下载文件，如果目标上有相应依赖的话，还可以爆破ssh和端口扫描，暂时就这样吧，困的不行了，睡觉。
github地址： oracle_java_shell_client

上一篇：一个寄生虫一句话木马分析 下一篇：SDL软件安全设计初窥

1. 0 条回复
   - 动动手指，沙发就是你的了！