F5远程代码执行漏洞分析

原创 队员编号045 酒仙桥六号部队 4天前

这是 **酒仙桥六号部队** 的第 **45** 篇文章。
全文共计1162个字，预计阅读时长5分钟。

概述

在 F5 BIG-IP 产品的流量管理用户页面（TMUI)/配置实用程序的特定页面中存在一处远程代码执行漏洞。

未授权的远程攻击者通过向该页面发送特制的请求包，可以造成任意Java 代码执行。进而控制 F5 BIG-IP 的全部功能，包括但不限于：执行任意系统命令、开启/禁用服务、创建/删除服务器端文件等。

影响范围

BIG-IP 15.x：15.1.0/15.0.0

BIG-IP 14.x：14.1.0 ~ 14.1.2

BIG-IP 13.x：13.1.0 ~ 13.1.3

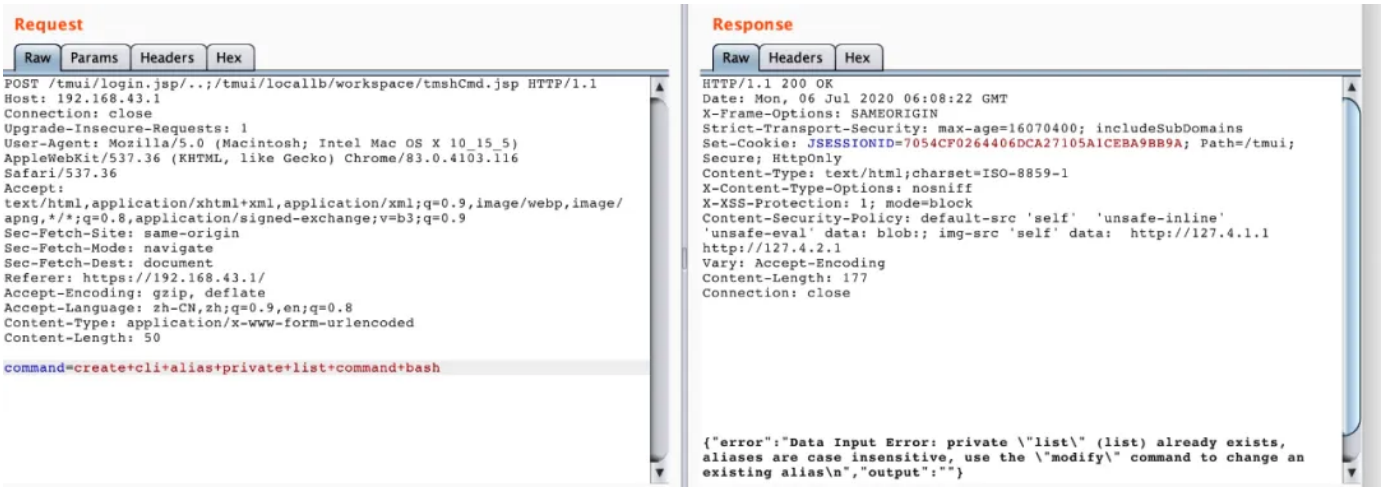
BIG-IP 12.x：12.1.0 ~ 12.1.5

BIG-IP 11.x：11.6.1 ~ 11.6.5

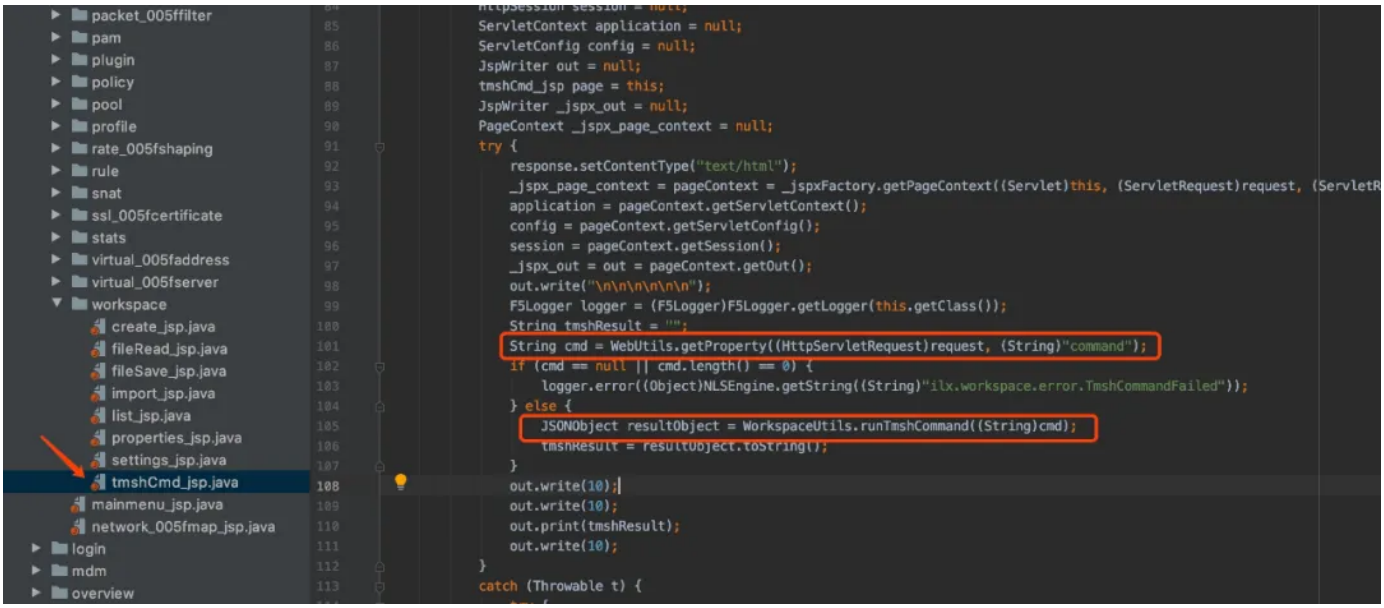
漏洞分析

F5 Tmsh命令执行

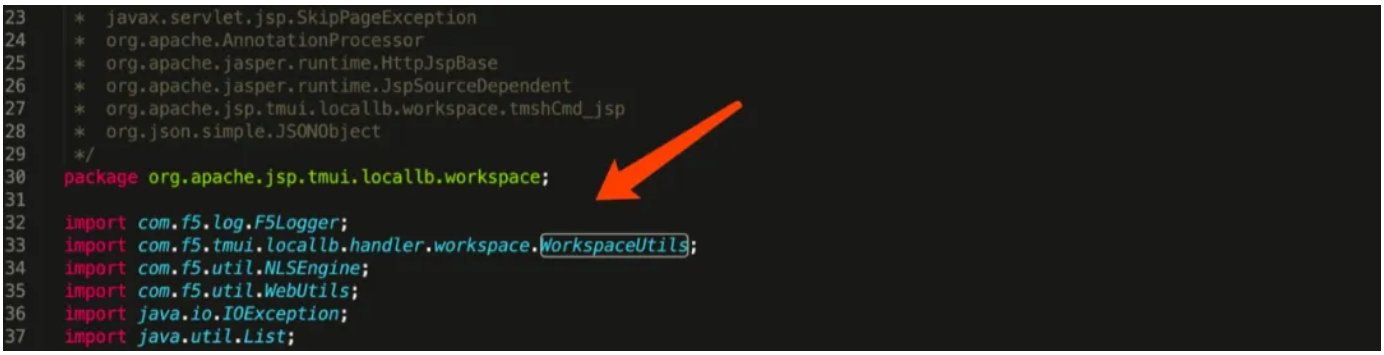
创建bash脚本:对应的F5的命令为command后面内容 请求路径为下图:



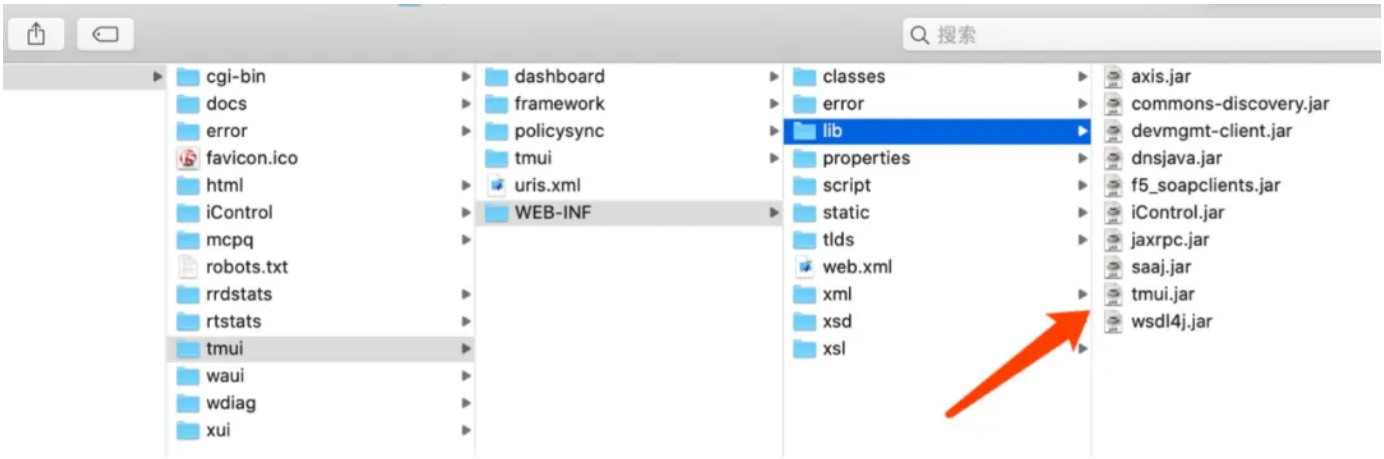
URL中存在../;绕过登录验证，这个是属于Tomcat对URI解析差异导致绕过了原有的权限校验，导致可以直 接访问到tmshCmd.jsp，对应代码:tmshCmd_jsp.java 文件cmd参数直接从请求中获取。



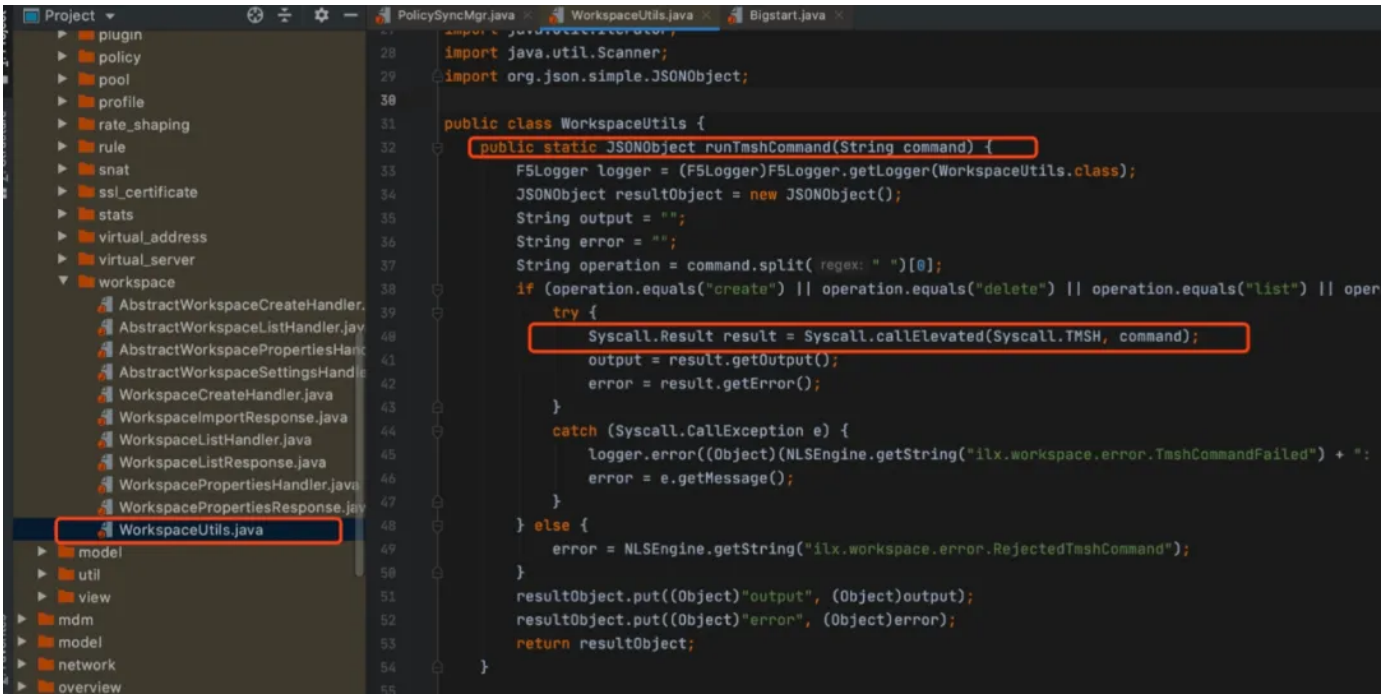
跟进WorkspaceUtils类中runTmshCommand方法，从导入包中寻找com.f5.tmui.locallb.handler.workspace.WorkspaceUtils对应的文件。



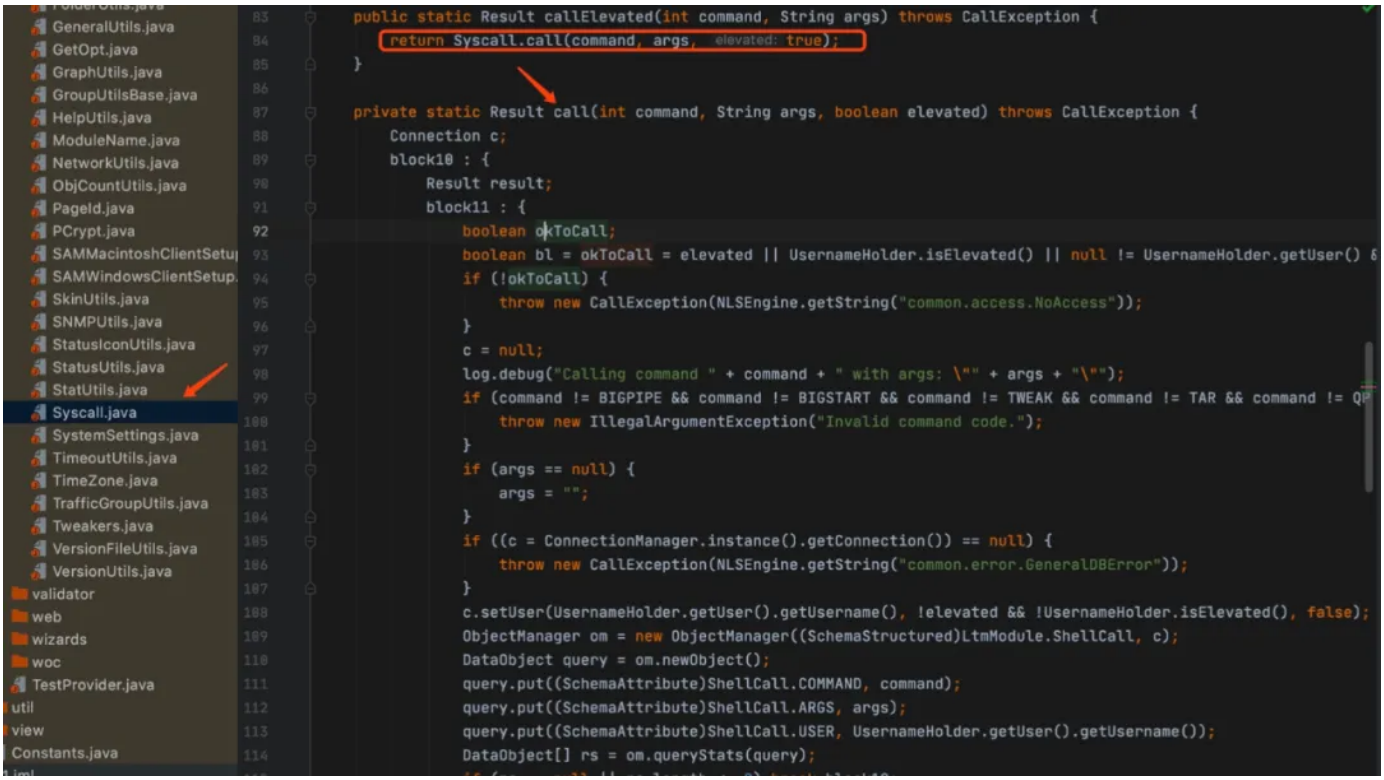
在lib中找到对应jar包，反编译:



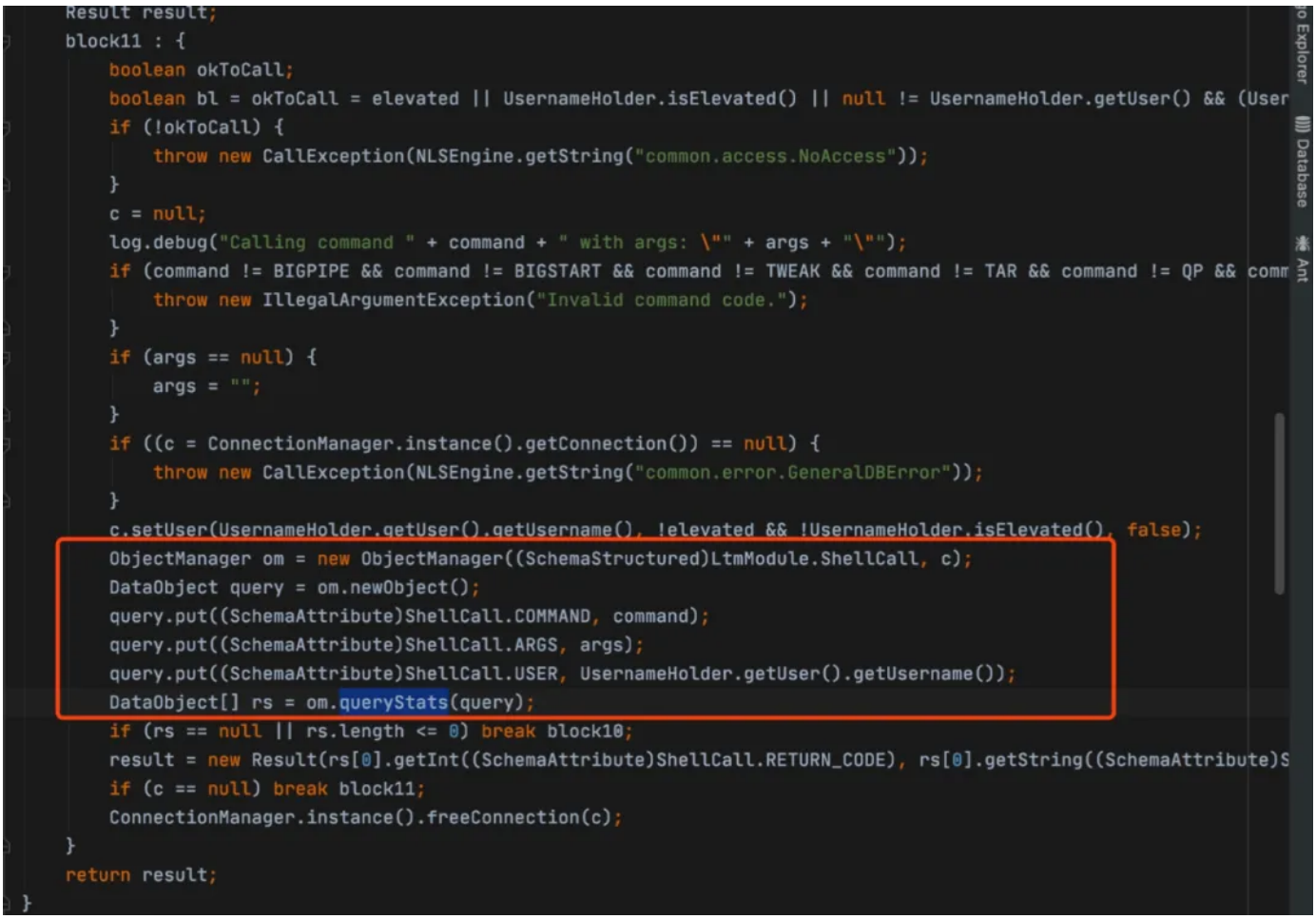
紧接着上文runTmshCommand方法，可以看到在38行处，做了命令判断，命令被分割后，仅允许 create，delete，list，modify等开头的命令。



跟进Syscall.callElevated方法，调用了call方法：



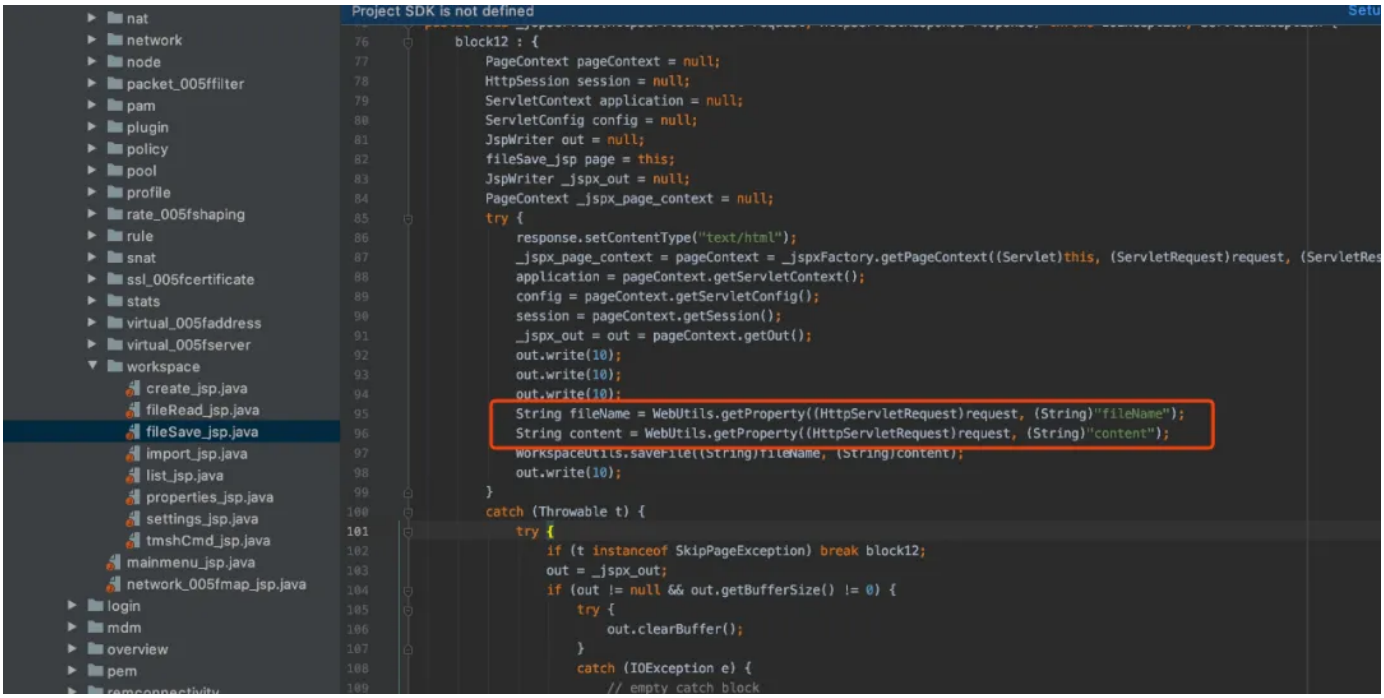
可以看到，args放到了ObjectManager里面，通过DataObject[] rs = om.queryStats(query);这行代码 把执行的命令的结果返回。



F5 任意文件写入

请求路径：

对应的jsp代码文件：



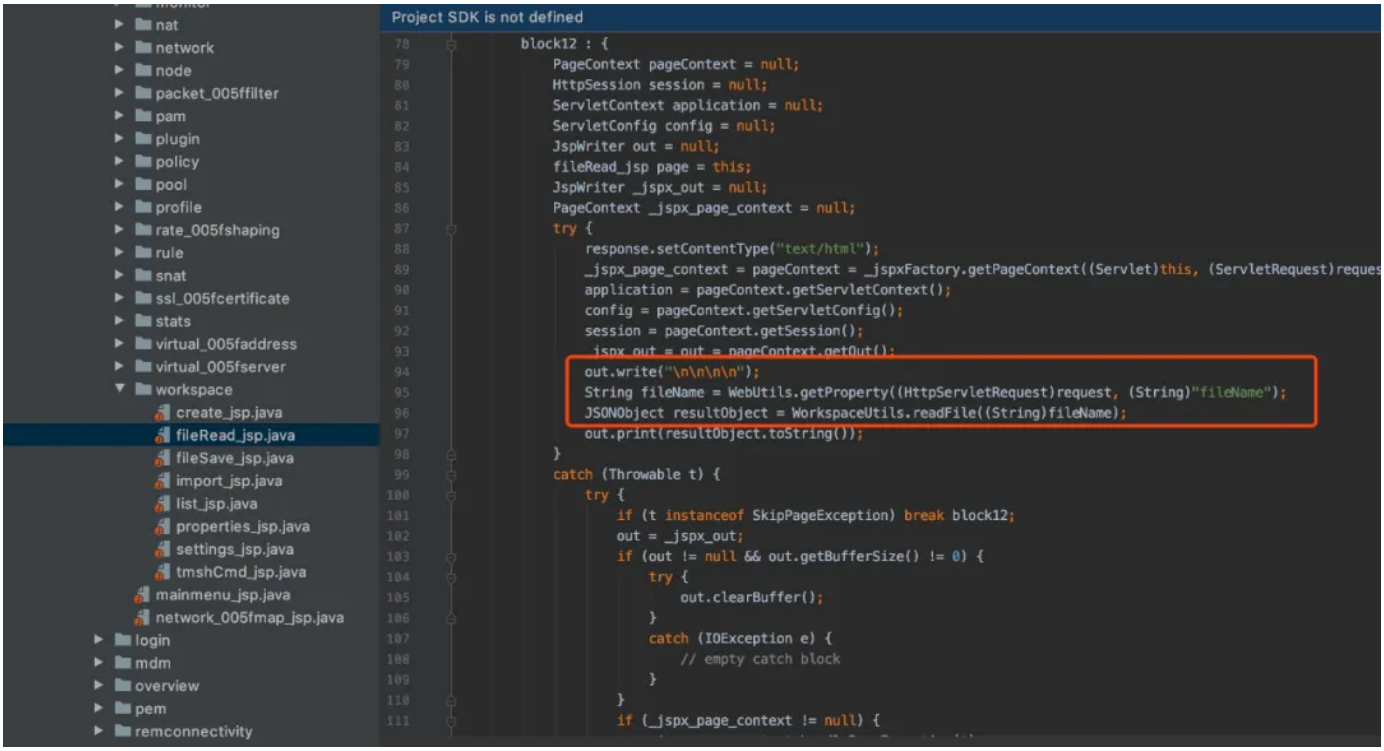
跟进对应的save方法，可以看到参数一路传递，fileName为路径，content为内容，最终通过 writer.println写入。

```java
public static void saveFile(String fileName, String content) throws IOException {
    F5Logger logger = (F5Logger)F5Logger.getLogger(WorkspaceUtils.class);
    PrintWriter writer = null;
    try {
        writer = new PrintWriter(new FileWriter(fileName));
        writer.println(content);
    }
    catch (IOException e) {
        String error = NLSEngine.getString("ilx.workspace.error.ErrorWritingFile") + ": " + e.getMessage();
        logger.error((Object)error);
        throw e;
    }
    finally {
        if (writer != null) {
            writer.close();
        }
    }
}
```

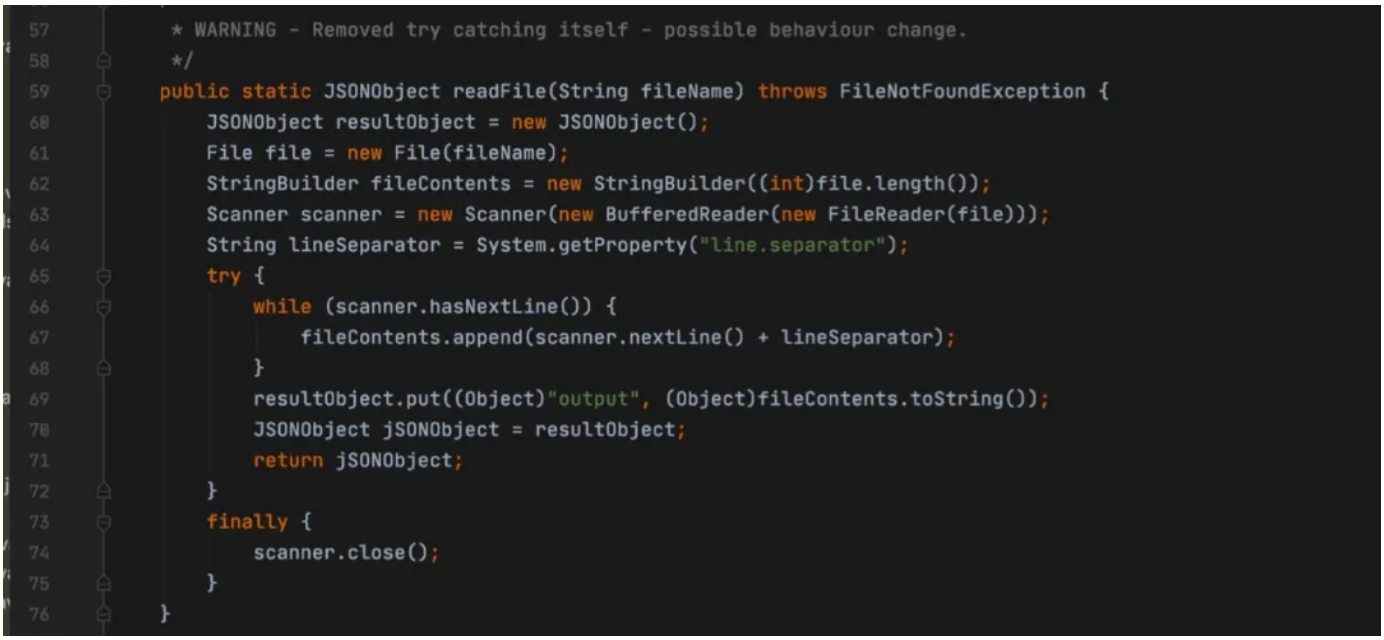
F5任意文件读取

请求路径：

对应的jsp代码文件：



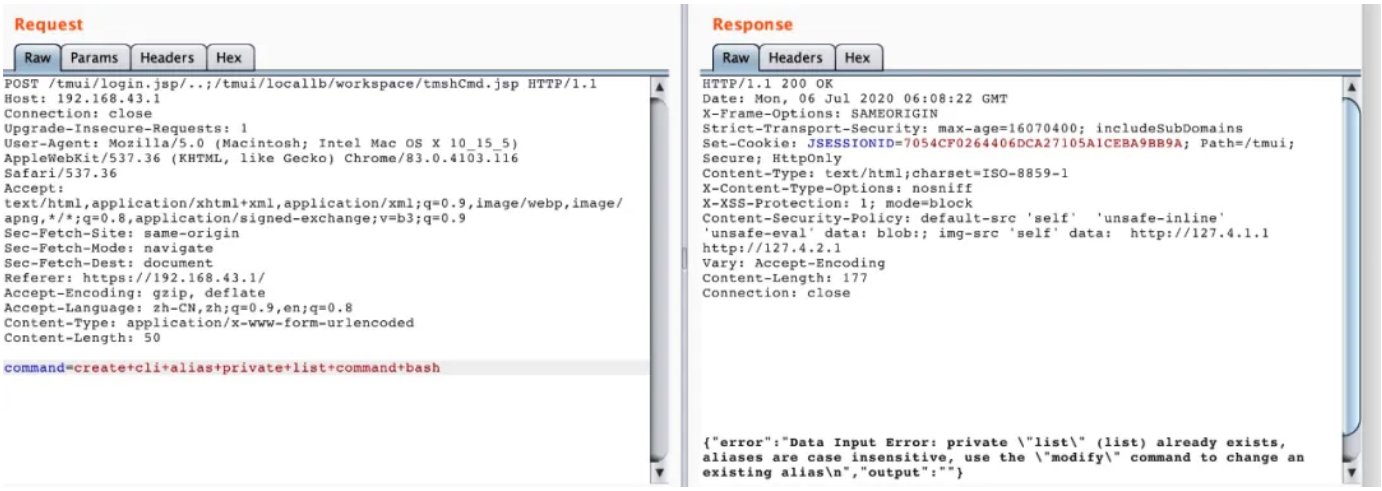
对应具体实现方法:



漏洞复现

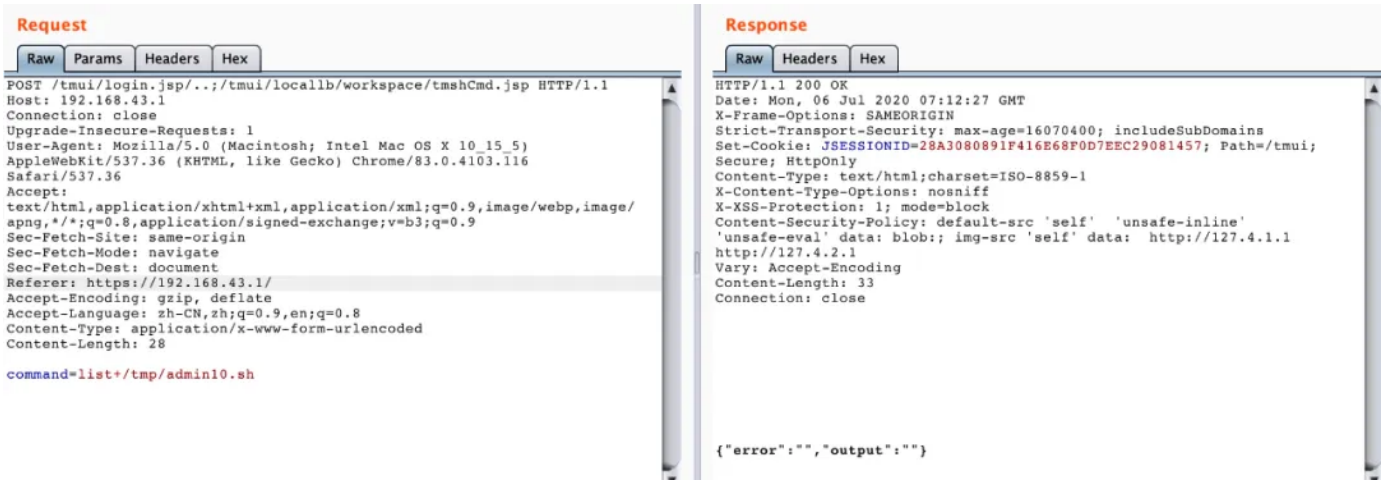
综合利用Getshell。

创建bash:

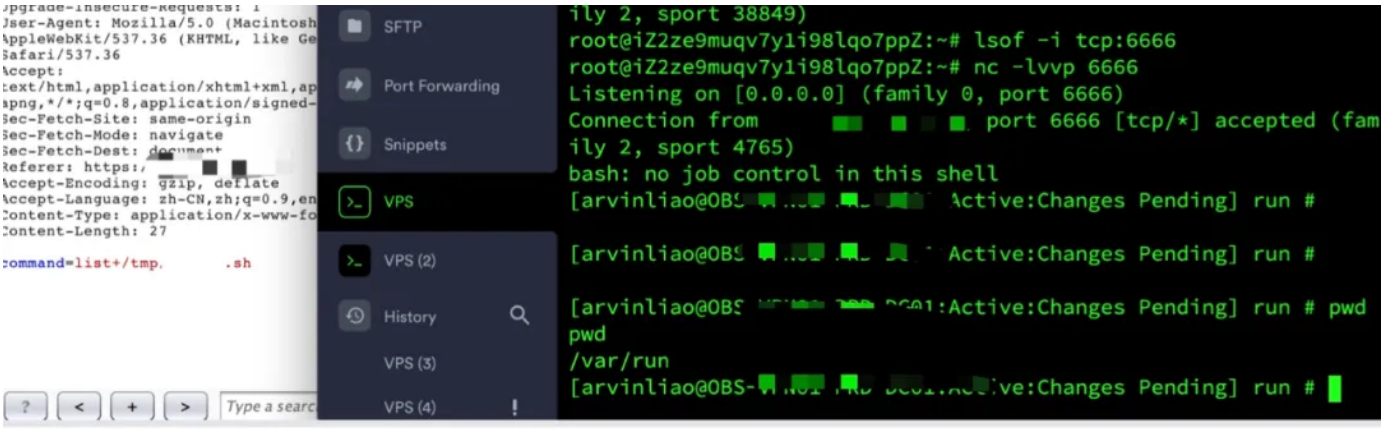


写入文件:

执行写入sh:



获取shell:



修复方案

7 月 7 日更新：

官方建议可以通过以下步骤暂时缓解影响（临时修复方案）

1）使用以下命令登录对应系统：tmsh

2）编辑 httpd 组件的配置文件；

edit /sys httpd all-properties

3）文件内容如下 include ' <LocationMatch "...;."> Redirect 404 / </LocationMatch> '

4）按照如下操作保存文件；

按下 ESC 并依次输入：wq

5）执行命令刷新配置文件；

save /sys config

6）重启 httpd 服务。

restart sys service httpd 并禁止外部IP对 TMUI 页面的访问。

7 月 7 日更新： 官方初版安全通告里给出的临时缓解方案是在 httpd 配置文件中加入如下部分，以禁止请求的 url 路径里出现 ..; 进行路径跳转：

```
include '
<LocationMatch ".*\.\.;.*">
Redirect 404 /
</LocationMatch>
```

```
5 '
```

然而却可以通过 /hsqldb; 无需跳转，去直接请求 org.hsqldb.Servlet，进一步执行 Java 代码。这种漏洞利用的方式，可以绕过上述配置规则。

7 月 9 日更新：

官方安全通告里给出的第二版临时缓解方案中在 httpd 配置文件中加入规则配置如下，以禁止请求的 url 路径里出现 ; 进行授权认证绕过：

```
include '
<LocationMatch ";">
Redirect 404 /
</LocationMatch>
'
```

然而却可以通过 /hsqldb%0a 的请求方式，再次绕过以往的漏洞缓解规则，去直接请求 org.hsqldb.Servlet，进一步执行 Java 代码。

Tomcat特性相关文章

https://xz.aliyun.com/t/7544#toc-9 https://i.blackhat.com/us-18/Wed-August-8/us-18-Orange-Tsai-Breaking-Parser-Logic-Take-Your-Path-Normalization-Off-And-Pop-0days-Out-2.pdf