**Apache Spark Command Injection Vulnerability**

CVE-2022-33891

|  |  |
| --- | --- |
| Name\_zh | Apache Spark命令注入漏洞 |
| Name\_en | Apache Spark Command Injection Vulnerability |
| CVE | CVE-2022-33891 |
| CVSS评分 | 10 |
| 威胁等级 | High |
| CNNVD | CNVD-2022-52835 |
| 其他id | ---- |
| 受影响软件 | Apache Spark |

简介

Apache Spark 是专为大规模数据处理而设计的快速通用的计算引擎。Spark是UC Berkeley AMP lab (加州大学伯克利分校的AMP实验室)所开源的类Hadoop MapReduce的通用并行框架，Spark，拥有Hadoop MapReduce所具有的优点；但不同于MapReduce的是——Job中间输出结果可以保存在内存中，从而不再需要读写HDFS，因此Spark能更好地适用于数据挖掘与机器学习等需要迭代的MapReduce的算法。

Apache Spark UI 可以设置选项 spark.acls.enable 启用 ACL ，使用身份验证过滤器。用以检查用户是否具有查看或修改应用程序的访问权限。如果启用了 ACL则 HttpSecurityFilter 中的代码路径可以允许用户通过提供任意用户名来执行命令。该功能最终将根据用户输入构建一个 Unix shell 命令并执行它，最终导致任意命令执行。

Apache spark is a fast and universal computing engine designed for large-scale data processing. Spark is a general parallel framework similar to Hadoop MapReduce, which is open-source by UC Berkeley amp Lab (AMP lab at the University of California, Berkeley). Spark has the advantages of Hadoop MapReduce; But what is different from MapReduce is that the intermediate output results of job can be saved in memory, so there is no need to read and write HDFS. Therefore, spark can be better applied to MapReduce algorithms that need iteration, such as data mining and machine learning.

Apache spark UI can set the option spark.acls Enable enable ACL and use authentication filter. It is used to check whether the user has access to view or modify the application. If ACL is enabled, the code path in httpsecurityfilter allows users to execute commands by providing any user name. This function will eventually build a UNIX shell command based on user input and execute it, eventually leading to arbitrary command execution.

漏洞影响

Apache Spark <= 3.0.3

3.1.1 <= Apache Spark <= 3.1.2

3.2.0 <= Apache Spark <= 3.2.1

漏洞复现

实验环境

准备两台虚拟机

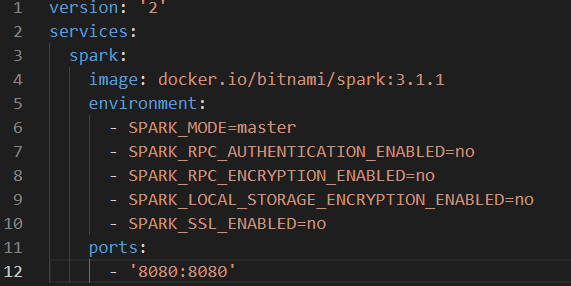
kali.2020 192.168.160.128

Windows 10 192.168.160.1

Apache Spark v 3.1.1

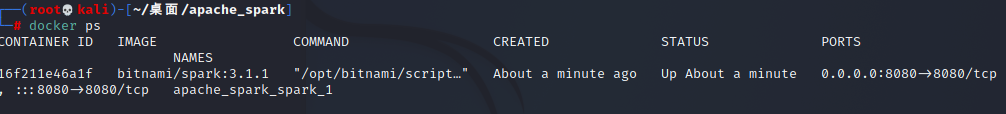
接下来利用这两台主机进行试验

使用docker搭建环境



使用docker-compose

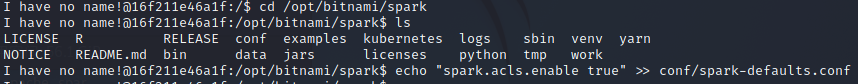
查看环境信息



漏洞需特定配置，默认不存在

sudo docker exec -it spark\_spark\_1 /bin/bash 进入docker

在conf/spark-defaults.conf中将spark.acls.enable设置为true



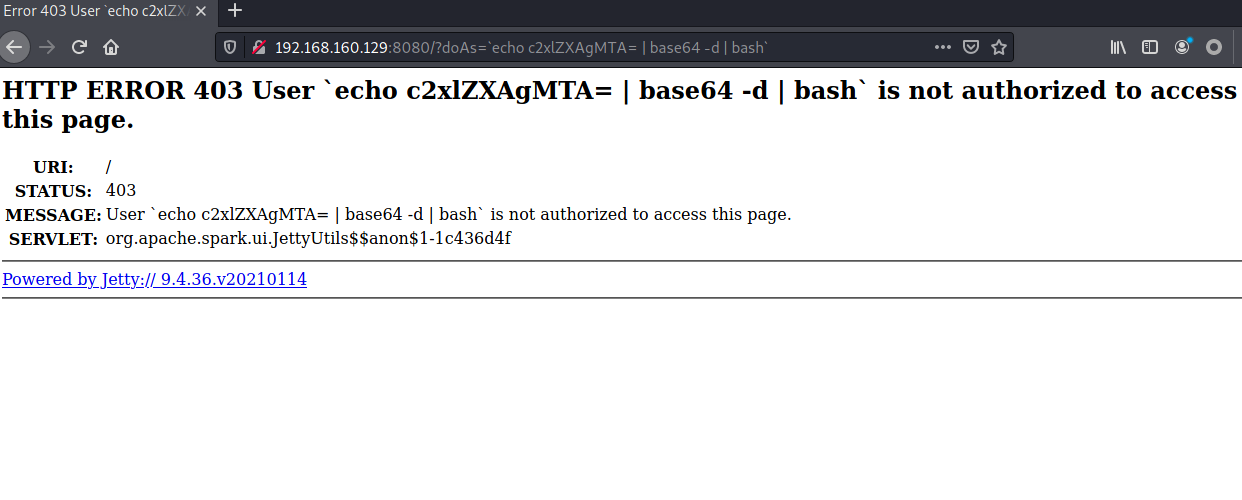
漏洞触发

http://localhost:8080/?doAs=`[command injection here]`

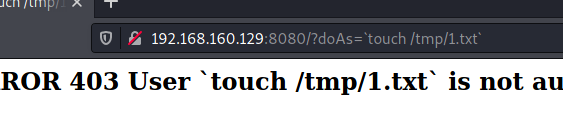
在测试页面中输入如下payload

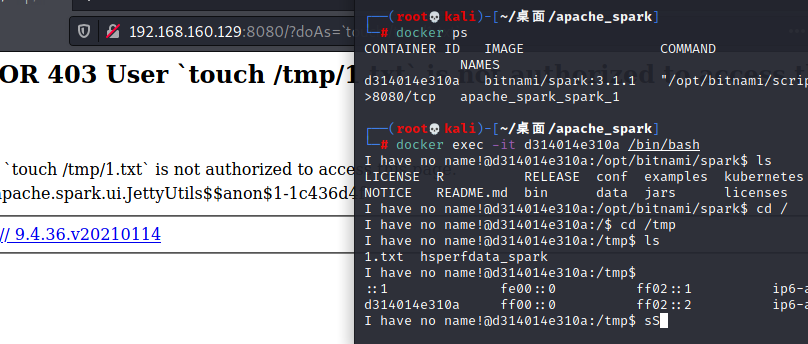
http://IP:8080/?doAs=`echo c2xlZXAgMTA= | base64 -d | bash`

如果sleep 10秒 则存在漏洞



可以使用 touch /tmp/1.txt 创建文件

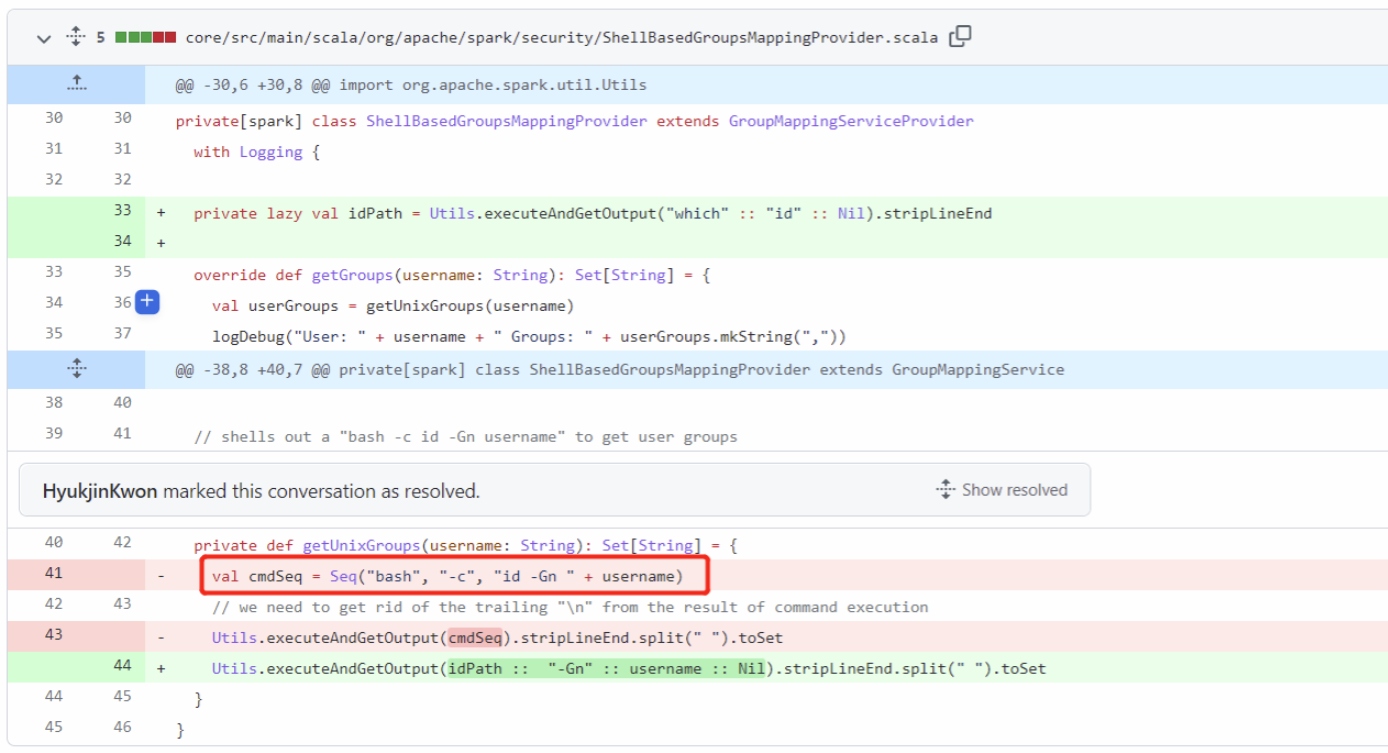




后台查看，1.txt已经被创建。

漏洞分析

根据diff（https://github.com/apache/spark/pull/36315/files）分析

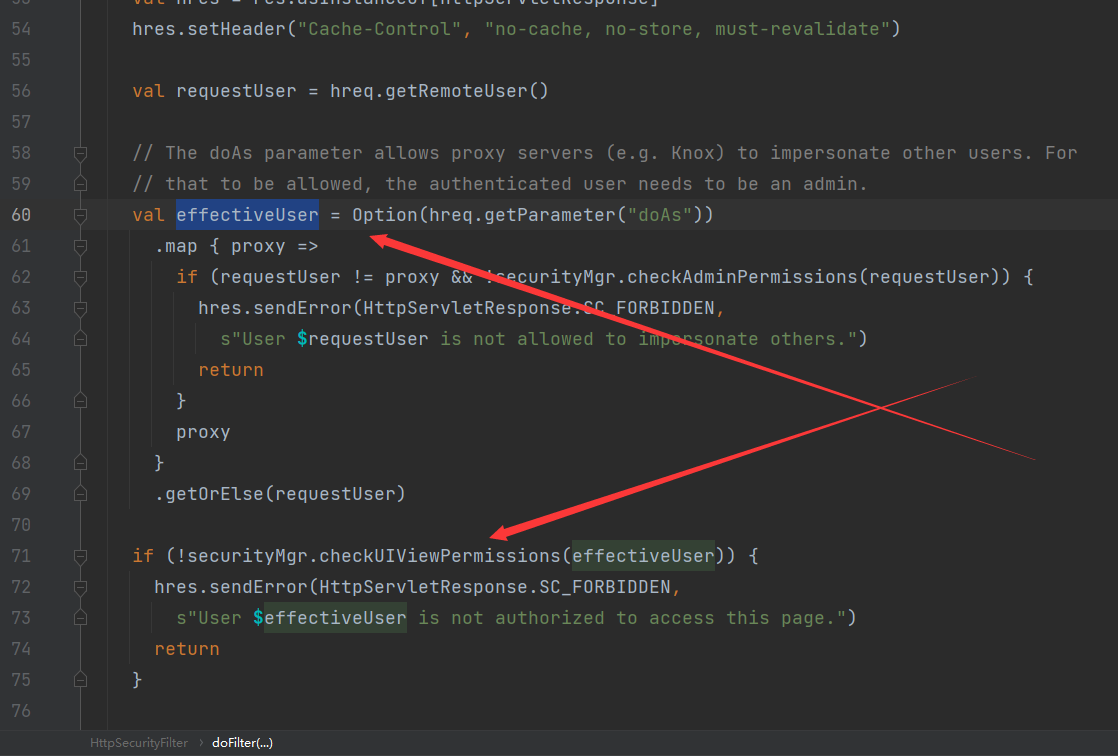


如上所示，使用命令拼接且没有做任何处理，而在修复的版本中直接删除了ShellBasedGroupsMappingProvider函数中对bash的调用。

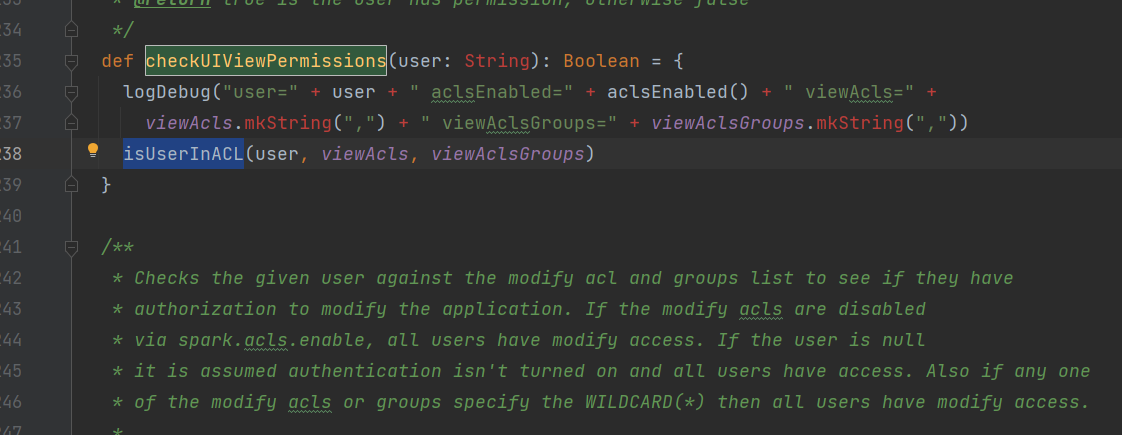
在之前说到漏洞需将spark.acls.enable配置为true，默认不存在

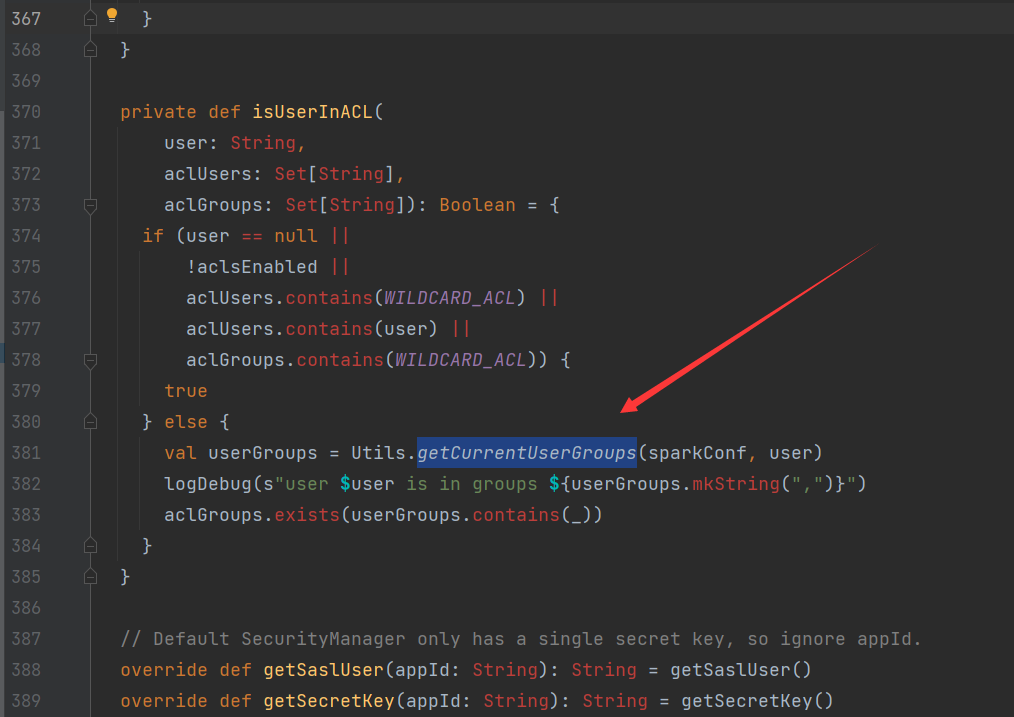
在Apache spark启用ACL后，会通过HttpSecurityFilter这个filter进行权限的校验。

在进入doFilter函数之后，首先会提取参数“doAs”的值，然后赋值给effectiveUser，进入org.apache.spark.SecurityManager#checkUIViewPermissions函数

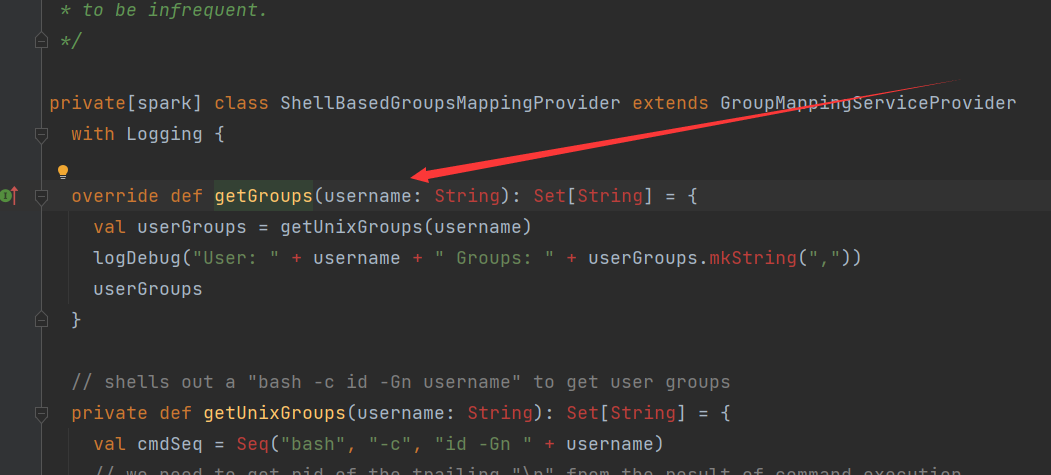


后续跟进一系列函数进行处理。

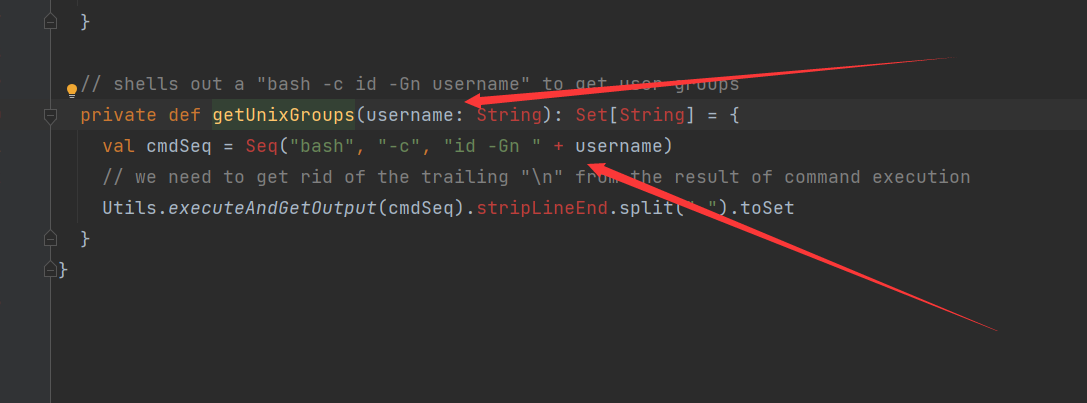




进入org.apache.spark.security.ShellBasedGroupsMappingProvider#getGroups函数时，username为传入参数。



随后进入org.apache.spark.security.ShellBasedGroupsMappingProvider#getUnixGroups函数。



在这里username进行了拼接处理，因为我们传入的username参数可控，便形成了命令注入。

后续将通过executeAndGetOutput函数直接触发传入的命令，造成命令执行。

修复建议

它现在在最新版本中得到缓解,建议升级到最新版修复此问题

https://archive.apache.org/dist/spark/