Jenkins 任意文件读取漏洞复现与分析 - 【CVE-2018-1999002】

chybeta / 2018-07-26 00:44:15 / 浏览数 7962 安全技术 漏洞分析 顶(1) 踩(0)

SECURITY-914 / CVE-2018-1999002

```
An arbitrary file read vulnerability in the Stapler web framework used by Jenkins allowed unauthenticated users to send crafte

Input validation in Stapler has been improved to prevent this.
```

漏洞影响版本：

```
Jenkins weekly up to and including 2.132
Jenkins LTS up to and including 2.121.1
```

漏洞复现

测试环境： win平台

通过查找commit记录可知需要将其检出至 29ca81dd59c255ad633f1bd86cf1be40a5f02c64之前

```
> git clone https://github.com/jenkinsci/jenkins.git
> git checkout 40250f08aca7f3f8816f21870ee23463a52ef2f2
```

检查`core/pom.xml`的第41行，确保版本为`1.250`

```
<staplerFork>true</staplerFork>
<stapler.version>1.250</stapler.version>
```

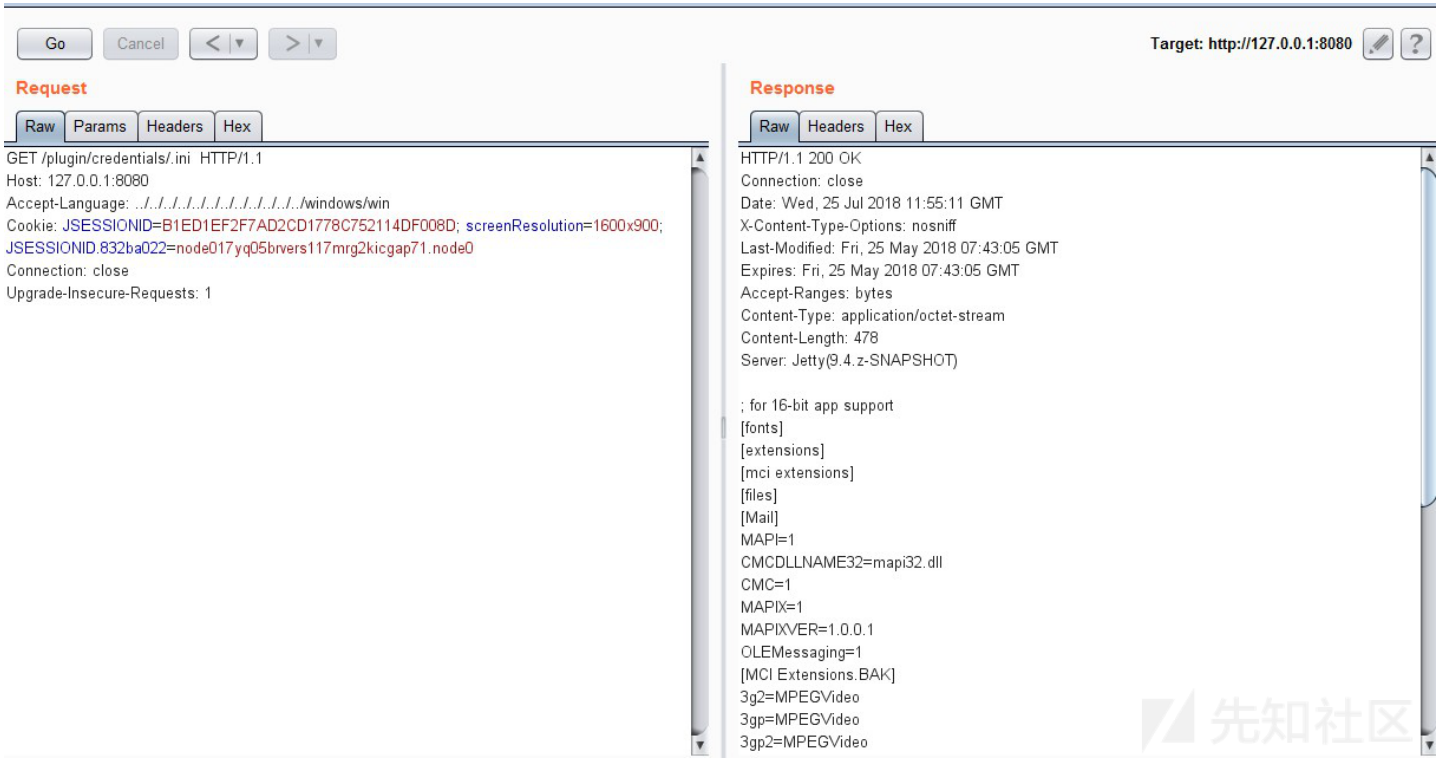
然后命令行下编译war包

```
mvn clean install -pl war -am -DskipTests
```

在`jenkins\war\target`目录下获得编译好的`jenkins.war`，同目录下启动：

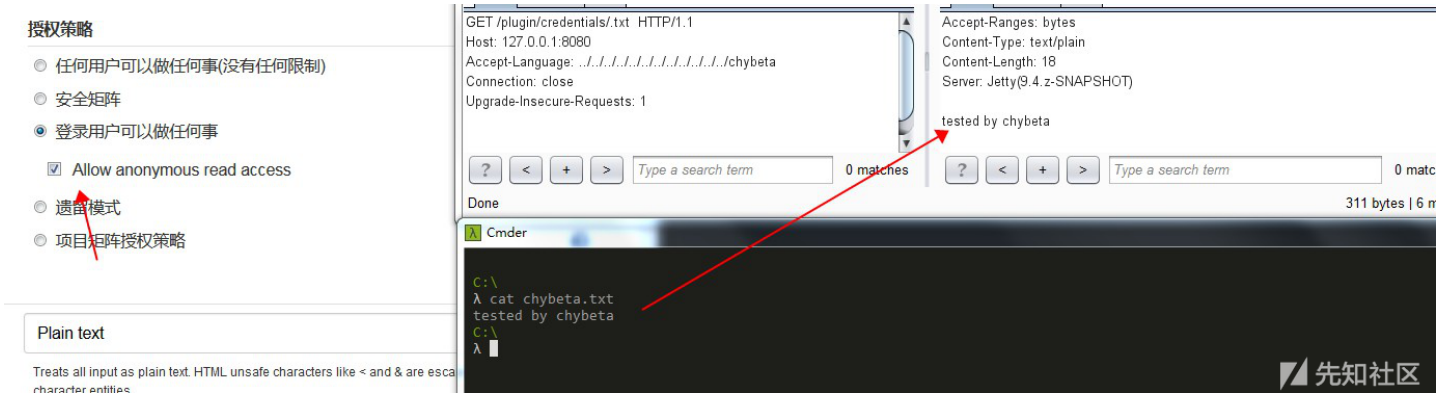
```
java -jar jenkins.war
```

在管理员登陆（有cookie）的情况下

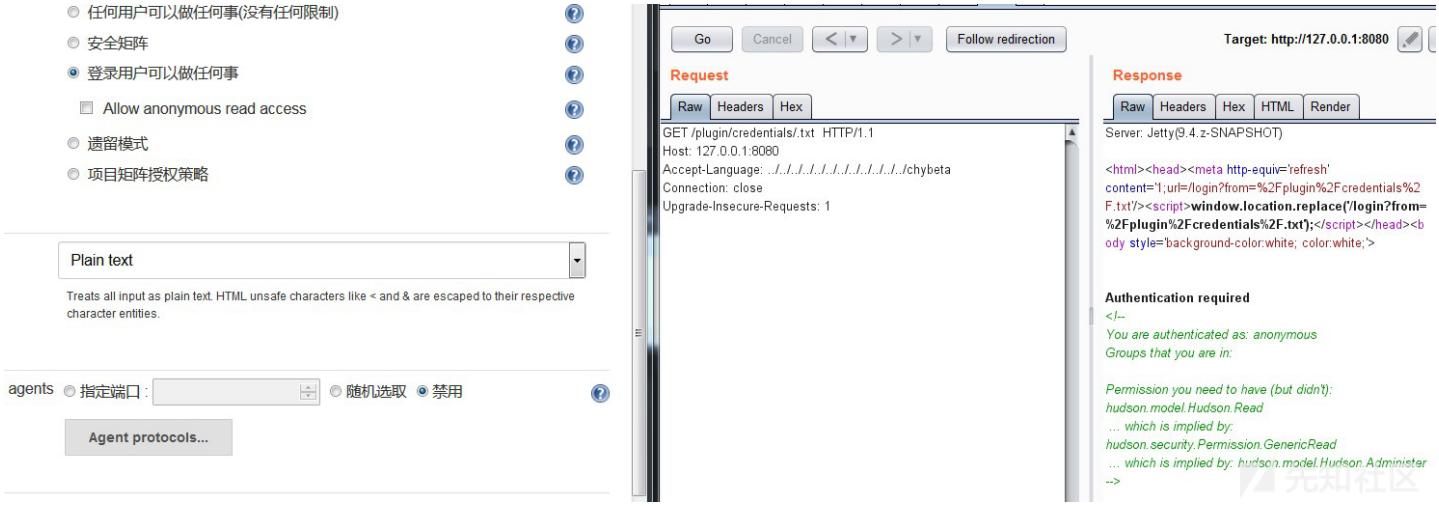


在没有登陆（未授权，cookie清空）的情况下，只有当管理员开启了`allow anonymous read access`的时候，才能实现任意文件读取，否则仍需登陆。

开启：



未开启：



而在linux下利用条件会更加苛刻，见后文。

漏洞分析

以payload为例，请求的url为`/plugin/credentials/.ini`。而在`hudson/Plugin.java:227`

```
/**
    * This method serves static resources in the plugin under <tt>hudson/plugin/SHORTNAME</tt>.
**/
public void doDynamic(StaplerRequest req, StaplerResponse rsp) throws IOException, ServletException {
    String path = req.getRestOfPath();

    String pathUC = path.toUpperCase(Locale.ENGLISH);
    if (path.isEmpty() || path.contains("..") || path.startsWith(".") || path.contains("%") || pathUC.contains("META-INF") || p
        LOGGER.warning("rejecting possibly malicious " + req.getRequestURIWithQueryString());
        rsp.sendError(HttpServletResponse.SC_BAD_REQUEST);
        return;
    }

    // Stapler routes requests like the "/static/.../foo/bar/zot" to be treated like "/foo/bar/zot"
    // and this is used to serve long expiration header, by using Jenkins.VERSION_HASH as "..."
    // to create unique URLs. Recognize that and set a long expiration header.
    String requestPath = req.getRequestURI().substring(req.getContextPath().length());
    boolean staticLink = requestPath.startsWith("/static/");

    long expires = staticLink ? TimeUnit2.DAYS.toMillis(365) : -1;

    // use serveLocalizedFile to support automatic locale selection
    rsp.serveLocalizedFile(req, new URL(wrapper.baseResourceURL, '.' + path), expires);
}
```

`doDynamic`函数用于处理类似`/plugin/xx`的请求，`serveLocalizedFile`在`stapler-1.250-sources.jar!/org/kohsuke/stapler/ResponseImpl.java`第20

```
public void serveLocalizedFile(StaplerRequest request, URL res, long expiration) throws ServletException, IOException {
    if(!stapler.serveStaticResource(request, this, stapler.selectResourceByLocale(res,request.getLocale()), expiration))
        sendError(SC_NOT_FOUND);
```

```
}
```

先看最里面的request.getLocale()，然后再来分析stapler.selectResourceByLocale()。

跟入request.getLocale()，至jetty-server-9.2.15.v20160210-sources.jar!/org/eclipse/jetty/server/Request.java:692:

```
@Override
public Locale getLocale()
{
   ...

   if (size > 0)
   {
       String language = (String)acceptLanguage.get(0);
       language = HttpFields.valueParameters(language,null);
       String country = "";
       int dash = language.indexOf('-');
       if (dash > -1)
       {
           country = language.substring(dash + 1).trim();
           language = language.substring(0,dash).trim();
       }
       return new Locale(language,country);
   }

   return Locale.getDefault();
}
```

这里用于处理HTTP请求中的Accept-Language头部。比如zh-cn，则会根据-的位置被分为两部分，language为zh，country为cn，然后返回Locale(language,co

返回后即进入selectResourceByLocale(URL url, Locale locale),这里的locale参数即上一步返回的locale对象。

```
OpenConnection selectResourceByLocale(URL url, Locale locale) throws IOException {
   // hopefully HotSpot would be able to inline all the virtual calls in here
   return urlLocaleSelector.open(url.toString(),locale,url);
}
```

urlLocaleSelector对象的声明见stapler-1.250-sources.jar!/org/kohsuke/stapler/Stapler.java:390:

```
private final LocaleDrivenResourceSelector urlLocaleSelector = new LocaleDrivenResourceSelector() {
   @Override
   URL map(String url) throws IOException {
       return new URL(url);
   }
};
```

在stapler-1.250-sources.jar!/org/kohsuke/stapler/Stapler.java:324实现了LocaleDrivenResourceSelector类的open方法：

```
private abstract class LocaleDrivenResourceSelector {
   /**
       * The 'path' is divided into the base part and the extension, and the locale-specific
       * suffix is inserted to the base portion. {@link #map(String)} is used to convert
       * the combined path into {@link URL}, until we find one that works.
       *
       * <p>
       * The syntax of the locale specific resource is the same as property file localization.
       * So Japanese resource for <tt>foo.html</tt> would be named <tt>foo_ja.html</tt>.
       *
       * @param path
       *      path/URL-like string that represents the path of the base resource,
       *      say "foo/bar/index.html" or "file:///a/b/c/d/efg.png"
       * @param locale
       *      The preferred locale
       * @param fallback
       *      The {@link URL} representation of the {@code path} parameter
       *      Used as a fallback.
       */
   OpenConnection open(String path, Locale locale, URL fallback) throws IOException {
       String s = path;
       int idx = s.lastIndexOf('.');
       if(idx<0)   // no file extension, so no locale switch available
```

```
            return openURL(fallback);
        String base = s.substring(0,idx);
        String ext = s.substring(idx);
        if(ext.indexOf('/')>=0) // the '.' we found was not an extension separator
            return openURL(fallback);

        OpenConnection con;

        // try locale specific resources first.
        con = openURL(map(base + '_' + locale.getLanguage() + '_' + locale.getCountry() + '_' + locale.getVariant() + ext));
        if(con!=null)   return con;
        con = openURL(map(base+'_'+ locale.getLanguage()+'_'+ locale.getCountry()+ext));
        if(con!=null)   return con;
        con = openURL(map(base+'_'+ locale.getLanguage()+ext));
        if(con!=null)   return con;
        // default
        return openURL(fallback);
    }

    /**
        * Maps the 'path' into {@link URL}.
        */
    abstract URL map(String path) throws IOException;
}
```

先看看开头的注释，这段代码本意是想根据对应的语言（Accept-Language）来返回不同的文件，比如在ja的条件下请求`foo.html`，则相当于去请求`foo_ja.html`，这个

结合payload来看，我们请求的url为`/plugin/credentials/.ini`，则base为空，扩展名（ext变量）即为`.ini`，然后通过一系列的尝试openURL，在此例中即最后一个`= openURL(map(base+'_'+ locale.getLanguage()+ext));`，会去请求`_../../../../../../../../../../../../../windows/win.ini`，尽管目录`_..`并不存在，但在win下可以直接通过路径穿越来绕过。但在linux，则需要一个带有_的目录来想办法绕过。

补丁分析

Jenkins官方修改了pom.xml，同时增加一个测试用例文件。真正的补丁在`stapler`这个web框架中，见commit记录：
https://github.com/stapler/stapler/commit/8e9679b08c36a2f0cf2a81855d5e04e2ed2ac2b3 ：

```
353 +           // RegExps found in Locale JavaDoc
354 +           String language = locale.getLanguage();
355 +           boolean languageOk = language.matches("^[a-zA-Z]{2,8}$");
356 +           String country = locale.getCountry();
357 +           boolean countryOk = country.matches("^[a-zA-Z]{2}|[0-9]{3}$");
358 +           String variant = locale.getVariant();
359 +
360 +           String SUBTAG = "(?:[0-9][0-9a-zA-Z]{3}|[0-9a-zA-Z]{5,8})";
361 +           boolean variantOk = variant.matches("^" + SUBTAG + "(?:[_\\-]" + SUBTAG + ")*$");
362 +
363             OpenConnection con;
364
365             // try locale specific resources first.
    -           con = openURL(map(base + '_' + locale.getLanguage() + '_' + locale.getCountry() + '_' + locale.getVariant() + ext));
    -           if(con!=null)   return con;
    -           con = openURL(map(base+'_'+ locale.getLanguage()+'_'+ locale.getCountry()+ext));
    -           if(con!=null)   return con;
    -           con = openURL(map(base+'_'+ locale.getLanguage()+ext));
    -           if(con!=null)   return con;
366 +           if(languageOk && countryOk && variantOk){
367 +               con = openURL(map(base + '_' + language + '_' + country + '_' + variant + ext));
368 +               if(con!=null)
369 +                   return con;
370 +           }
371 +           if(languageOk && countryOk){
372 +               con = openURL(map(base + '_'+ language + '_' + country + ext));
373 +               if(con!=null)
374 +                   return con;
375 +           }
376 +           if(languageOk){
377 +               con = openURL(map(base + '_' + language + ext));
378 +               if(con!=null)
379 +                   return con;
380 +           }
```

对从locale取出的language,country,variant均做了正则的校验，只允许字母数字以及特定格式的出现。在接下来的openUrl中，根据三种变量的不同检查情况来调用不

Reference

- https://jenkins.io/security/advisory/2018-07-18/
- https://github.com/jenkinsci/jenkins/blob/d71ac6ffe98ee62e0353af7a948a4ae1a69b67e9/test/src/test/java/jenkins/security/stapler/Security914Test.java
- https://github.com/stapler/stapler/commit/8e9679b08c36a2f0cf2a81855d5e04e2ed2ac2b3

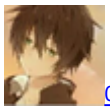
上一篇：高级USB key钓鱼（含poc） 下一篇：x86_64逆向工程简介

1. 3 条回复


z1nc 2018-08-02 09:43:55

大佬有分析出linux下如何利用了吗？Thx~

0 回复Ta


0r3ak 2018-08-08 11:46:43

想问一下，request.getLocale() 是怎么跟进去的？动态调试的？

0 回复Ta


57470****@qq.com 2018-10-23 23:37:00

@0r3ak 老哥找到方法了吗？我调试也进不去

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