本节讲述一个比较简单鸡肋的漏洞 - URL 跳转漏洞。算是多掌握一个小小技巧吧。

一、URL 跳转漏洞

URL重定向漏洞是指当应用程序使用 Java 的内置函数或类来处理 URL 重定向时，如果没有正确地验证输入，则可能允许攻击者构造恶意的 URL 来重定向到恶意网站。这样，攻击者就可以让用户访问恶意网站。

举个简单例子。

腾讯网是`www.qq.com`，里面有很多其他网站，比如：腾讯新闻，腾讯视频，直接点击就能跳转访问。

假如他的跳转链接是`https://www.qq.com/redirect?url=https://news.qq.com/mobile/`。如果后端代码没有进行校验，那么攻击者可以将后面的 URL 地址，改为恶意网站或钓鱼网站的地址，比如：`https://www.qq.com/redirect?url=https://hacker.com`。然后批量发给用户，说是点击即可让 QQ 等级增加一个太阳，来诱导大家点击。（当然了后面的 URL 太明显，有心的人一眼就能出来，此时可以对 URL 进行短连接处理。）

然后大家第一反应看到是 QQ 域名，还说点击就能领取一个太阳等级，没多想就点击了，最终导致了各种信息泄露，后果也是不堪设想……

在对网站进行黑盒测试时，可以观察一下 URL 中是否携带跳转参数，从而进一步进行测试是否存在 URL 跳转漏洞。如下所示：

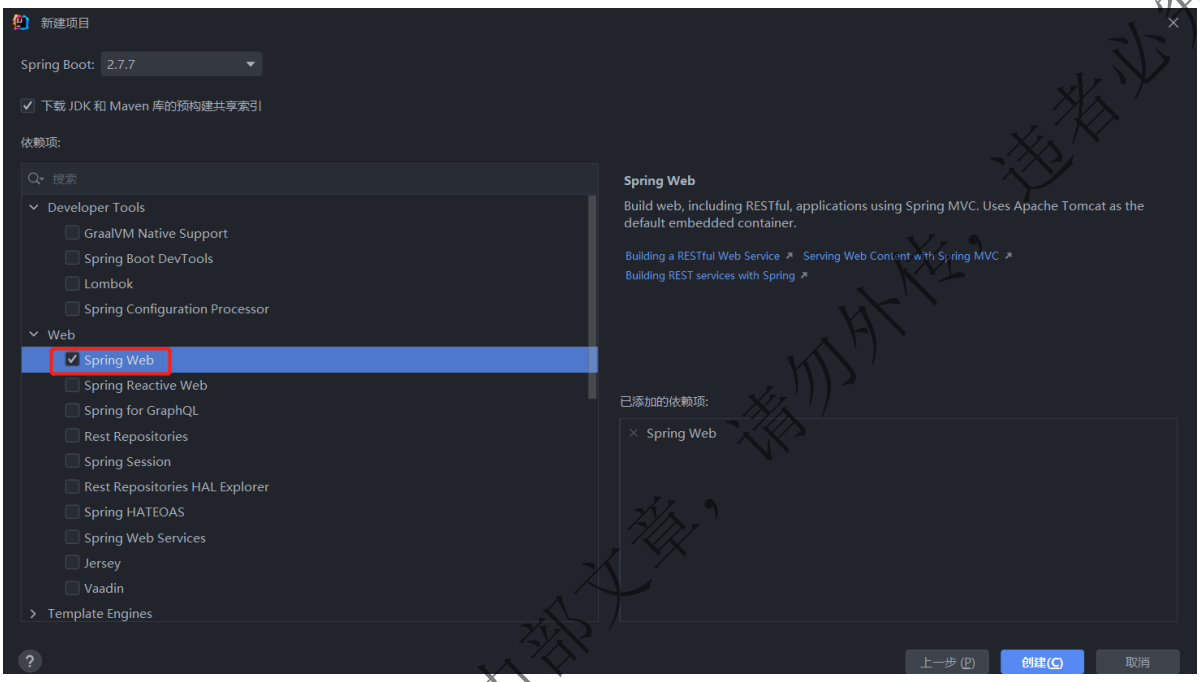
```
redirect
url
redirectUrl
callback
return_url
toUrl
ReturnUrl
fromUrl
redUrl
request
redirect_to
redirect_url
jump
jump_to
target
to
goto
link
linkto
domain
oauth_callback
```


二、URL 跳转漏洞代码

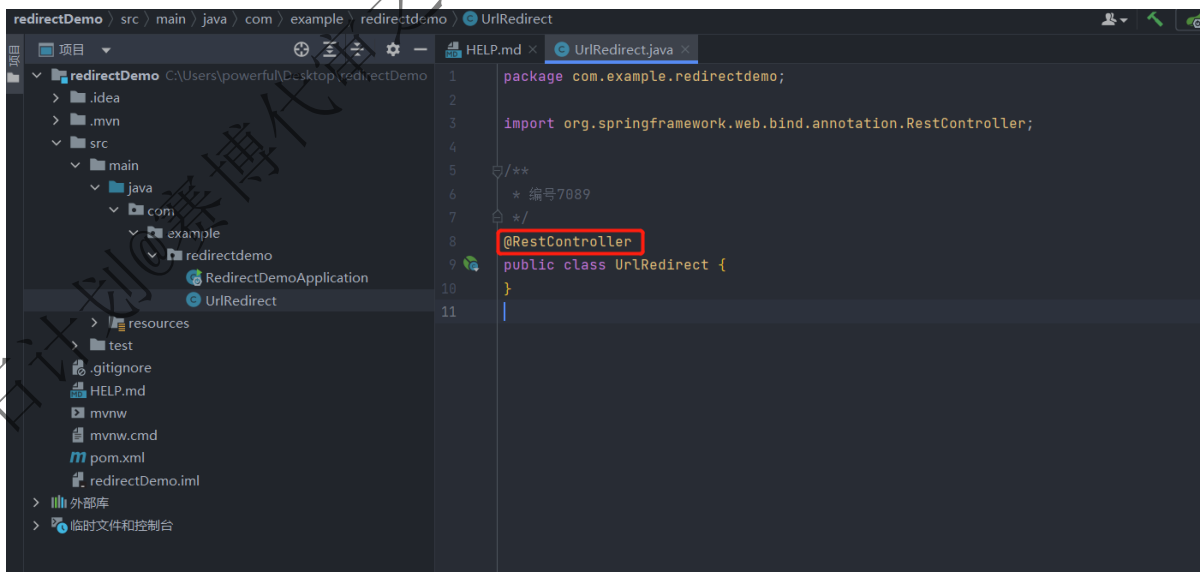
下面主要讲解两种跳转方式，302 重定向 - sendRedirect 和 301 重定向 - setHeader。

> 301跳转也叫301重定向，也叫301转向，也叫301永久重定向,是网站建设过程中的一个功能。一般用于2个域名指向同一个网站。 一般来说，利用跳转，对网站的排名不会有影响。但不会转移全部权重。只能说让损失降到最低。
>
> 302跳转就网址重定向的一种，它区别于301跳转，301是网址永久重定向，302则是网址的临时定向。302转向或者302重定向（302 redirect）指的是当浏览器要求一个网页的时候，主机所返回的状态码。302状态码的意义是暂时转向到另外一个网址。

①、老规矩，新建一个名为`redirectDemo`的项目，引入 Spring Web 依赖，最后点击创建。如下图所示：



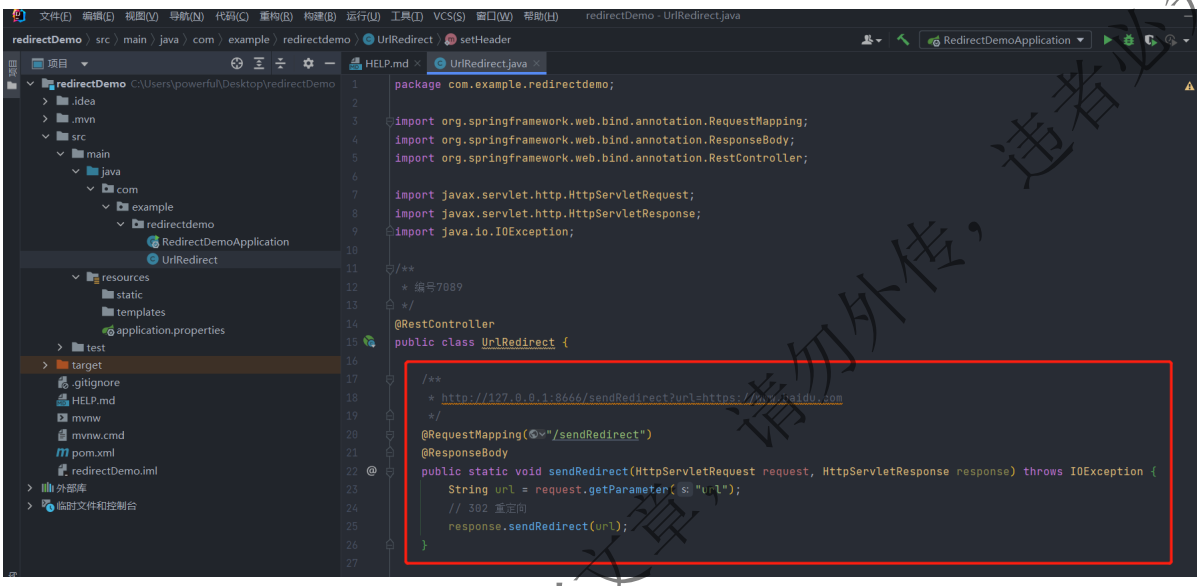
②、在`src/main/java/com/example/redirectdemo`下创建一个名为`UrlRedirect.java`的 Java Class，并键入`@RestController`注解，最终如下图所示：



1、302 重定向之 sendRedirect

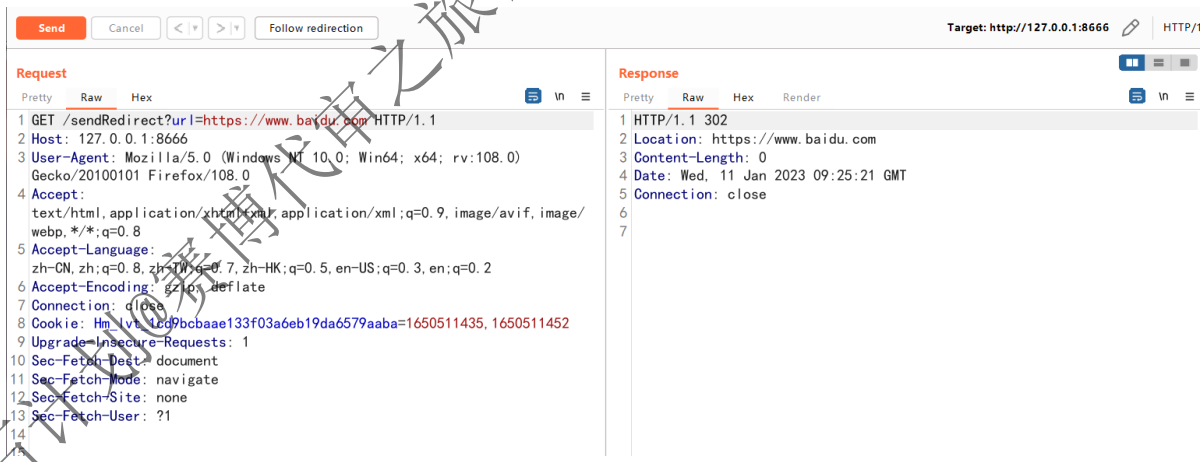
在`src/main/java/com/example/redirectdemo/UrlRedirect.java`中键入以下代码，最终如下图所示：

```
/**
     * http://127.0.0.1:8080/sendRedirect?url=https://www.baidu.com
     */
    @RequestMapping("/sendRedirect")
    @ResponseBody
    public static void sendRedirect(HttpServletRequest request,
HttpServletResponse response) throws IOException {
        String url = request.getParameter("url");
        // 302 跳转
        response.sendRedirect(url);
    }
```



访问 `http://127.0.0.1:8080/sendRedirect?url=https://www.baidu.com` 观察效果。

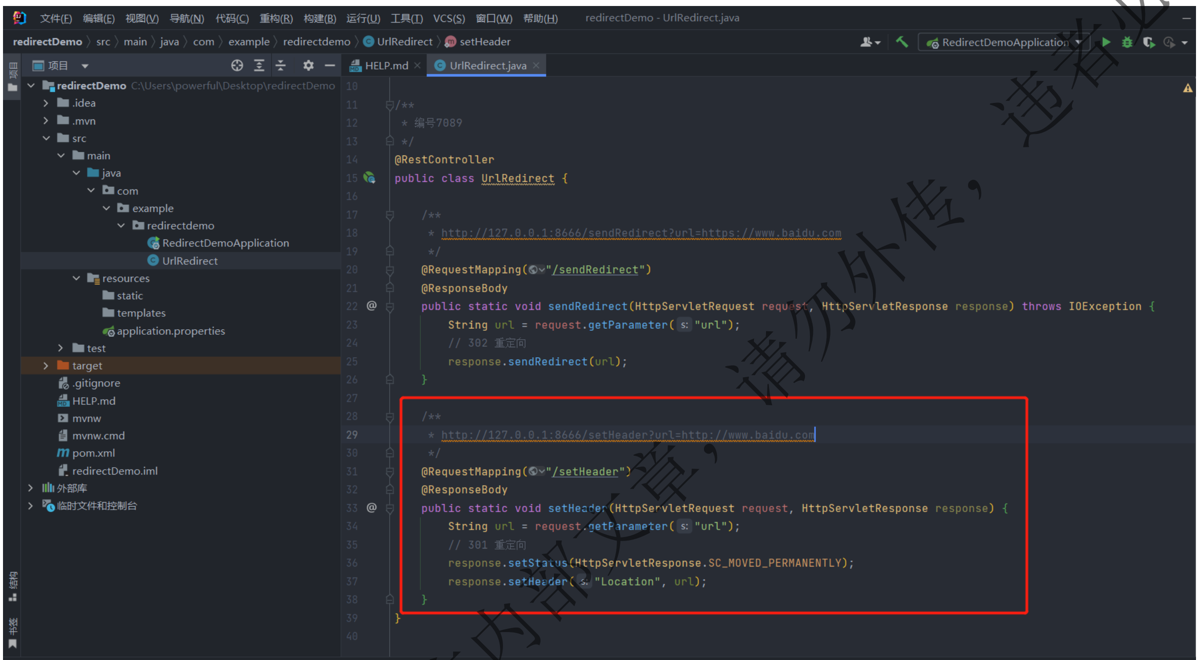
使用 BurpSuite 抓包，可以看到做了 302 跳转，Location 指向地址为 url 参数值，响应如下图所示：



2、301 重定向之 setHeader

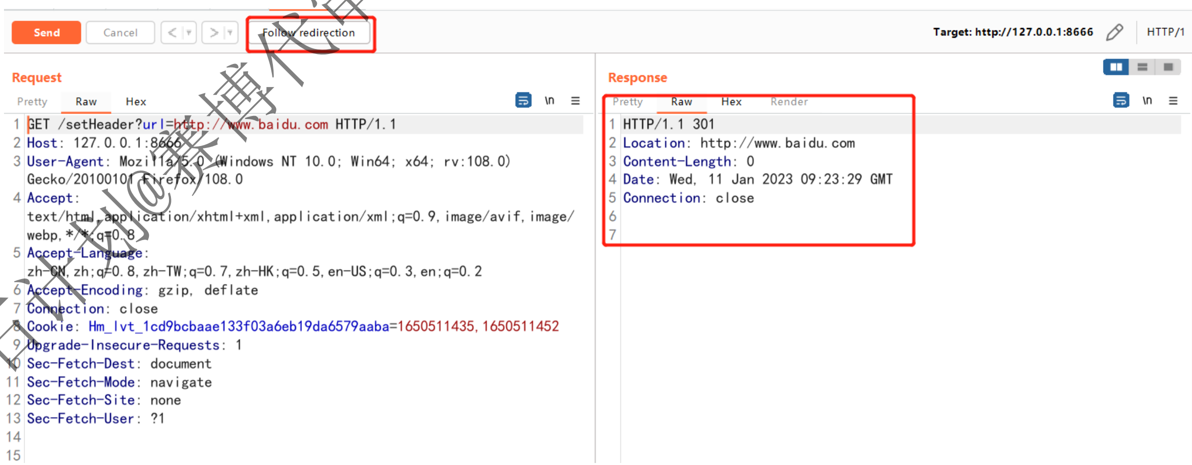
在 `src/main/java/com/example/redirectdemo/UrlRedirect.java` 中键入以下代码，最终如下图所示：

```java
    /**
     * http://localhost:8080/setHeader?url=http://www.baidu.com
     */
    @RequestMapping("/setHeader")
    @ResponseBody
    public static void setHeader(HttpServletRequest request, HttpServletResponse response) {
        String url = request.getParameter("url");
        // 301 重定向
        response.setStatus(HttpServletResponse.SC_MOVED_PERMANENTLY);
        response.setHeader("Location", url);
    }
```



访问 `http://127.0.0.1:8666/setHeader?url=http://www.baidu.com` 观察效果。

使用 BurpSuite 抓包，可以看到做了 301 跳转，Location 指向地址为 url 参数值，响应如下图所示：



三、URL 跳转和 SSRF 漏洞区别

依我理解，在调试代码时不难发现，URL 跳转动作主要发生在客户端处，也就是以你的浏览器进行访问和你直接使用浏览器访问意思相同。

而 SSRF 漏洞请求 URL 主要发生在服务端处，也就是服务器端请求了 URL，然后将他获取到的资源返回给了你。