Spring MVC Autobinding漏洞实例初窥

cryin / 2017-07-10 13:40:00 / 浏览数 5416 技术文章 技术文章 顶(0) 踩(0)

概述

在整理完这边文章时，nowill也在先知社区发了一篇关于自动绑定漏洞的文章《浅析自动绑定漏洞》，很详细～有兴趣的可以看下。所以本篇内容重点从实例介绍下Spring MVC Autobinding漏洞。

Autobinding-自动绑定漏洞，根据不同语言/框架，该漏洞有几个不同的叫法，如下：

- Mass Assignment: Ruby on Rails, NodeJS
- Autobinding: Spring MVC, ASP.NET MVC
- Object injection: PHP(对象注入、反序列化漏洞)

软件框架有时允许开发人员自动将HTTP请求参数绑定到程序代码变量或对象中，从而使开发人员更容易地使用该框架。这里攻击者就可以利用这种方法通过构造http请求，MVC框架，所以文中描述为自动绑定漏洞。这些信息都可以参考owasp上关于Mass Assignment的介绍。

文章代码实例以ZeroNights-HackQuest-2016的demo为例

@ModelAttribute注解

在Spring mvc中，注解@ModelAttribute是一个非常常用的注解，其功能主要在两方面：

- 运用在参数上，会将客户端传递过来的参数按名称注入到指定对象中，并且会将这个对象自动加入ModelMap中，便于View层使用；
- 运用在方法上，会在每一个@RequestMapping标注的方法前执行，如果有返回值，则自动将该返回值加入到ModelMap中；

@ModelAttribute注释一个方法的参数,从Form表单或URL参数中获取

```
@RequestMapping(value = "/home", method = RequestMethod.GET)
   public String home(@ModelAttribute User user, Model model) {
       if (showSecret){
           model.addAttribute("firstSecret", firstSecret);
       }
       return "home";
   }
```

view端通过${user.name}即可访问。注意这时候这个User类一定要有没有参数的构造函数，形如：

```
public class User {

   private String name;
   private String pass;
   private Integer weight;

   public User() {

   }

   public User(String name, String pass, Integer weight) {
       this.name=name;
       this.pass=pass;
       this.weight=weight;
   }
   ......
}
```

@ModelAttribute注释一个方法,该方法会在此controller每个@RequestMapping方法执行前被执行

```
@ModelAttribute("showSecret")
   public Boolean getShowSectet() {
       logger.debug("flag: " + showSecret);
       return showSecret;
   }
```

@SessionAttributes注解

在默认情况下，ModelMap 中的属性作用域是 request 级别，也就是说，当本次请求结束后，ModelMap 中的属性将销毁。如果希望在多个请求中共享 ModelMap 中的属性，必须将其属性转存到 session 中，这样 ModelMap 的属性才可以被跨请求访问。

Spring 允许我们有选择地指定 ModelMap 中的哪些属性需要转存到 session 中，以便下一个请求对应的 ModelMap 的属性列表中还能访问到这些属性。这一功能是通过类定义处标注 @SessionAttributes("user") 注解来实现的。SpringMVC 就会自动将 @SessionAttributes 定义的属性注入到 ModelMap 对象，在 setup action 的参数列表时，去 ModelMap 中取到这样的对象，再添加到参数列表。只要不去调用 SessionStatus 的 setComplete() 方法，这个对象就会一直保留在 Session 中，从而实现 Session 信息的共享

justiceleague 实例详解

把程序运行起来，可以看到这个应用菜单栏有about，reg，Sign up， Forgot password？这4个页面组成。我们关注的重点是密码找回功能，即怎么样绕过安全问题验证并找回密码。所以我们关注的重点是怎么样绕过密码找回功能。

1、首先看reset方法，把不影响代码逻辑的删掉。这样更简洁易懂：

```
@Controller
@SessionAttributes("user")
public class ResetPasswordController {

private UserService userService;
...
@RequestMapping(value = "/reset", method = RequestMethod.POST)
public String resetHandler(@RequestParam String username, Model model) {
       User user = userService.findByName(username);
       if (user == null) {
           return "reset";
       }
       model.addAttribute("user", user);
       return "redirect: resetQuestion";
   }
```

这里从参数获取username并检查有没有这个用户，如果有则把这个user对象放到Model中。因为这个Controller使用了@SessionAttributes("user")，所以同时也会自动把

为什么这里会自动把user对象放到session中，具体原因见@SessionAttributes注解

2、resetQuestion密码找回安全问题校验页面有resetViewQuestionHandler这个方法展现

```
@RequestMapping(value = "/resetQuestion", method = RequestMethod.GET)
   public String resetViewQuestionHandler(@ModelAttribute User user) {
       logger.info("Welcome resetQuestion ! " + user);
       return "resetQuestion";
   }
```

这里使用了@ModelAttribute User
user，实际上这里是从session中获取user对象。但存在问题是如果在请求中添加user对象的成员变量时则会更改user对象对应成员的值。
所以当我们给resetQuestionHandler发送GET请求的时候可以添加"answer=hehe"参数，这样就可以给session中的对象赋值，将原本密码找回的安全问题答案修改成"heh

安全建议

Spring MVC中可以使用@InitBinder注解，通过WebDataBinder的方法setAllowedFields、setDisallowedFields设置允许或不允许绑定的参数。

参考

[1] https://www.owasp.org/index.php/Mass_Assignment_Cheat_Sheet#Spring_MVC [2] http://bobao.360.cn/learning/detail/3991.html [3] https://github.com/GrrrDog/ZeroNights-HackQuest-2016

上一篇：利用API NtQueryInfo... 下一篇：S2-048 漏洞调试及分析
1. 10 条回复


c0de 2017-07-11 02:44:51

学习了

0 回复Ta


svenll 2017-07-17 07:02:48

学习了

0 回复Ta


laocaogege 2017-07-19 10:48:06

学习了

0 回复Ta

 hades 2017-07-20 01:05:14

大伙有问题可以参与讨论~~

0 回复Ta

 laocaogege 2017-07-20 01:41:31

请教个基础问题，如果@SessionAttributes("user"）直接赋值给一个已存在的session属性。会刷新原值嘛？

0 回复Ta

 cryin 2017-07-20 02:31:38

我理解的如果user对象值不同，那赋值后是会刷新user对象的值的～

0 回复Ta


simeon 2017-07-20 03:11:52

学习了

0 回复Ta


laocaogege 2017-07-20 03:18:42

那我理解session这个注解在实际开发中，只能用户线程间的数据交互。但是这个数据结构的变量首先要严格控制，一定要避免讲上述数据用于任何操作的根据或者凭证。如果做不到上一点，那么这个注解本身就是个危险函数，如果做到这一点那么二次变量覆盖的危害其实也是可控的。 不知道理解是否正确

0 回复Ta


cryin 2017-07-20 04:01:25

引用第8楼laocaogege于2017-07-20 11:18发表的 回 6楼(cryin) 的帖子：
那我理解session这个注解在实际开发中，只能用户线程间的数据交互。但是这个数据结构的变量首先要严格控制，一定要避免讲上述数据用于任何操作的根据或者凭证。

如果做不到上一点，那么这个注解本身就是个危险函数，如果做到这一点那么二次变量覆盖的危害其实也是可控的。 不知道理解是否正确
[url=https://xianzhi.aliyun.com/forum/job.php?action=topost&tid=1843&pid=3457][/url]

嗯， 首先数据可控是一方面，就像php反序列化一样。。还需要具备一些magic函数，调用了注入的反序列化对象。而且是一些威胁的操作才会造成攻击。。

0 回复Ta


laocaogege 2017-07-20 05:58:13

嗯 明白了 以后审计中要注意下这个注解 谢谢撸主

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

RSS 关于社区 友情链接 社区小黑板