文章编号:1674 - 2869(2018)05 - 0565 - 04

基于AOP的Web应用程序的安全会话管理

叶志鹏,何成万*,张峥峰
武汉工程大学计算机科学与工程学院,湖北 武汉 430205

摘 要:为降低Web应用程序中合法用户身份被非法窃取的风险,同时提高应用系统机密性及完整性,提出了一种基于AOP的安全会话管理方法。该方法关联远程IP地址与会话标识符(SessionID),通过对访问请求的关联性分析验证合法用户身份,从而应对Web应用程序中常见的失效身份验证与会话管理问题。通过方面(Aspect)封装的应用程序接口(API)具有较好的可扩展性,经编织后的Web应用程序无需修改原业务逻辑代码,就能有效提升自身会话管理机制的安全性及可靠性,保障用户数据不遭受未授权访问。

关键词:应用程序接口;面向方面编程;会话固定;失效的身份验证与会话管理
中图分类号:TP393.08 **文献标识码**:A **doi**:10.3969/j.issn.1674-2869.2018.05.017

Secure Session Management of Web-Based Application Using Aspect-Oriented Programming

YE Zhipeng, HE Chenwan, ZHANG Zhengfeng*
School of Computer Science and Engineering, Wuhan Institute of Technology, Wuhan 430205, China

Abstract: To reduce the risk of illegally spoofing legitimate users in web applications and improve the confidentiality and integrity of the application, a method of secure session management using Aspect-oriented programming (AOP) was proposed. By associating the remote IP address with the session identifier (SessionID) and analyzing the relevance of access requests, the legitimate user's identity was authenticated, so problems of the common broken authentication and session management in Web applications were solved. Application programming interface (API) encapsulated by Aspect has good scalability. The woven web application without modifying the original business logic code can effectively improve the security and reliability of its own session management mechanism, and protect the user data from unauthorized access.

Keywords: application programming interface; aspect-oriented programming; session fixation; broken authentication and session management

伴随“互联网+”时代的到来,互联网、网站与网络应用程序受欢迎程度日渐增高,在人们的日常生活中扮演了越来越重要的角色。Web应用程序[1]涉猎领域广泛,如商业合同、财务通知、医疗信息、教育或个人目的。身份验证是任何应用程序中关键及重要组成部分,但即使是可靠的身份验证机制也会被错误的管理功能所破坏,包括密码重置、忘记密码、记住密码、帐户更新和其他相关功能。Web应用程序中失效的身份验证和会话管理包括处理用户认证和管理活动会话的所有特性。

模块化编程[2]和面向方面编程是软件工程中用于帮助应用程序设计的两种方法[3],其中面向方面编程(Aspect Oriented Programming, AOP)[4]作为

收稿日期:2018-05-27
作者简介:叶志鹏,硕士研究生。E-mail:412134033@qq.com
*通讯作者:何成万,博士,教授。 E-mail:hechengwan@hotmail.com
引文格式:叶志鹏,何成万,张峥峰. 基于AOP的Web应用程序的安全会话管理[J]. 武汉工程大学学报,2018,40(5):565-568.

一种改进软件系统中关注点分离问题的方法在1997年首次提出,AOP提供了特定语言机制,能够在不改变业务逻辑的情况下,通过在编译或运行期间向复杂软件系统增加横切关注点(crosscutting)功能,使得以模块化方式解决诸如代码分散与缠绕之类的问题成为可能。安全性问题通常分散于各业务逻辑模块,是Web应用程序中重要的横切关注点。面向方面编程语言如AspectJ,提供了新的机制和可能性,将系统分解为模块并将模块重新组合至系统,从而充分利用了新型软件开发方法的优势,使得软件系统中安全问题得以解决。文献[5]通过实验验证了运行时改变安全策略的优点。本文基于AOP实现了Web应用程序安全会话机制的应用程序接口(API),并采用AspectJ作为安全方面的实现语言用于开发原型系统以验证实验结果。在无须修改程序业务逻辑代码前提下,通过提供安全方面API指导Web应用程序开发人员保护其新的或遗产应用系统,使其免受来自针对失效身份验证与会话管理[6]的攻击。

1 会话管理脆弱性

Web应用程序体系结构[7]中主要包含三大部分:运行Web应用程序的服务器、后端数据库服务器及客户端应用程序。用户通常使用浏览器在Web应用服务器中获取信息,Web应用程序服务器充当中介负责客户端与数据库服务器间交互。HTTP[5]作为一种无状态协议,这意味着其没有为Web服务器提供一种维护用户后续请求状态的完整方式。为解决该问题,Web服务器或应用程序自身通过实现各类会话管理技术,从而为用户提供友好的环境。其基本思想是:服务器在终端用户使用浏览器第一次发起访问请求(或认证)时由伪随机数函数生成唯一的会话标识符(session ID),并在服务器端使用包括本地内存、平面文件(flat files)和数据库等多项技术存储会话ID和与其关联的会话数据(如用户名、账户ID等)。通过Cookie和Session机制,允许客户端与服务器间通过超文本传输协议(Hyper Text Transfer Protocol,HTTP)报文来交换状态信息,浏览器将确保后续访问请求携带特定会话ID字段。因此,会话ID成为了服务器标识用户会话信息并维护会话数据的用户身份标记。

目前Web应用程序中有三种广泛采用的维持会话方式:URL参数,隐藏表单字段和Cookie[6]。其中Cookie包含五个可选属性字段:domain和path指定Cookie范围;expires指定过期时间;secure指定只能在HTTPS加密信道上传输;HTTPonly防止客户端脚本读取Cookie值。Cookie自身有两种非常不寻常的行为:其一,Cookie的存储和读取间存在严重不对称性,Cookie被设置和存储为name/domain/path与属性值间的映射,但是仅有name-value键值对被传递给JavaScript和Web服务器,这种不对称性允许具有相同名称但不同域和路径范围的Cookie被写入浏览器,而随后的读取器可以一并读取所有名称相同的Cookie,却无法做出有效区分,因为在读取过程中不存在诸如路径之类的其他属性;其二,开发者可以为路径属性指定任意值,并且不受浏览器URL上下文的任何限制。虽然上述三种维持会话机制都有其缺陷,但Cookie已被证明为是其中最方便且安全性较高的一种方式。从安全角度出发,大多数针对Cookie会话管理机制的已知攻击,也可以用于针对URL或隐藏表单字段的方案。而反之不行,这使得Web应用程序广泛采用Cookie作为浏览器端协助状态管理的机制,并成为安全的最佳选择。但在目前常见的Web应用系统中会话ID不仅作为身份标识,同时也是身份验证器。这意味着登录时,用户将根据其凭据(例如,用户名/密码或数字证书)进行身份验证过程,服务器端分配的会话ID将作为有效访问会话的临时静态密码,这对于攻击者而言是极具吸引力的目标。

在复杂网络空间环境下,任何类似于HTTP这类通过在通信双方间传递密钥方式来维护状态的协议[8]通常都存在脆弱性,很容易遭受中间人攻击(Man-In-The-Middle,MITM)[9]。不怀好意的第三方攻击者通常在不引起用户任何怀疑的前提下拦截明文传输信道,并模拟所在信道中任意节点向其余节点注入伪造信息。成功窃取有效用户会话ID的攻击者无需得知合法用户名和密码,即可冒充任意用户身份发送伪造请求,进而直接进入该用户会话中以获取特权信息或增删改查数据,若窃取的是持久型会话,那么这类伪装则会持续相当长一段时间。这类通过窃取有效计算机会话(有时也称为会话密钥)以获得对计算机系统中数据或服务的未授权访问方式称为会话劫持(Session hijacking)[7,10]。目前针对Web会话安全性问题,主要集中在阻止攻击者获取(拦截、预测和暴力破解)由目标服务器发送至用户浏览器的会话ID。然而上述三种方法忽略了一种可能性:即攻击者可能向用户浏览器"发布"会话ID,从而迫使

浏览器登录至攻击者会话中。文献[11]将这种类型称为“会话固定”攻击，因为用户的会话ID在登录前就已被攻击者预先固定，而不是由服务器端伪随机数函数根据访问请求动态生成。综上所述，这类针对终端用户凭据的安全问题统称为失效的身份验证与会话管理[12-15]。

2　系统实现

目前，绝大部分Web应用程序系统通常使用凭据对用户进行身份验证，并授权合法用户通过Web页面访问符合其身份的数据。当用户登录过程中所提交的用户名和密码信息与后端数据库相匹配时，则成功完成身份验证过程，Web服务器将通过分配会话ID方式以标识不同用户身份。在会话生命周期过程中，Web服务器将通过检查HTTP请求头字段中携带的会话ID信息，来辨识发送连接请求的用户，并利用可用的请求数据进行适当的响应。但这种弱类型验证方式无法有效应对当前复杂网络环境中各类攻击行为，终端用户的信息安全无法得到有效保障。

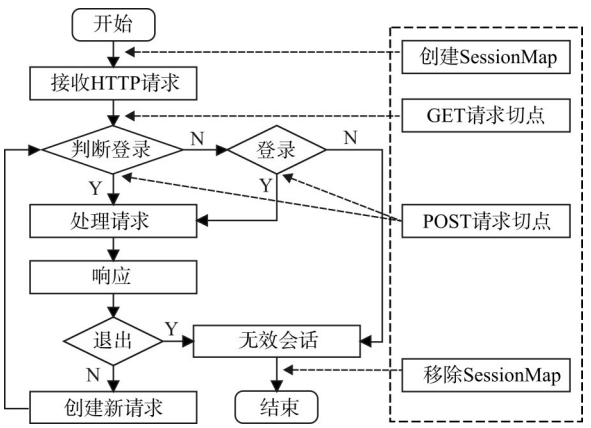

图1　安全方面编织图

Fig. 1　Security aspect weave diagram

针对上述问题本节提出一种应对失效身份验证与会话管理的通用模型如图1所示，其中安全会话管理主要由SessionMap类、切点(pointcut)及通知(advice)三部分构成，其具体实现过程依赖于应用程序编程接口的支持。SessionMap类采用哈希表数据结构，以键值对形式存储会话ID(键)与远程IP地址(值)间的映射关系，并定义了三类哈希表操作方法：

```
public class SessionMap {
    private static Map map = new HashMap();
    public static void put(String key, String value)
    public static String get(String key)
    public static void remove(String key)
    …
}
```

安全方面应用程序接口包括创建SessionMap、GET请求切点、POST请求切点及移除SessionMap四部分，安全方面在捕获Servlet应用程序成功执行init()方法调用后将首先创建SessionMap类的实例对象用于记录用户常用登陆IP信息。如下安全方面代码中，针对两类HTTP请求方式分别定义了Get及Post切点方法，通过捕获HttpServletRequest对象中getHeader()方法分别获取用于记录客户端首次连接请求及完成表单验证过程的远程用户IP地址。

```
public aspect ServletSecurityAspect{
    before (HttpServletRequest request, HttpServletResponse response):execution(* doGet (HttpServletRequest, HttpServletResponse)) && args(request, reponse);
    after(HttpServletRequest request, HttpServletResponse response):execution(* doPost (HttpServletRequest, HttpServletResponse)) && args(request, reponse);
}
```

针对不同Web应用程序的会话管理机制而言，需选取恰当的切点方法，并构建封装安全检测规范的通知，经AOP编织器在与切点相匹配的连接点处织入通知代码，引入会话映射(Session map)管理机制的Web应用程序认证模型如图2所示。

HttpSession对象在用户首次访问网站时自动创建，切点可通过匹配HttpServletRequest的getSession()方法调用获取该对象。原Web应用程序采用基于账户密码的用户凭据信息对用户身份进行验证：

若通过表单提交的用户名与密码和后端数据库中存储信息不一致，则表明登录失败是一次无效会话请求。

若用户成功登录，安全方面将通过上文切点中所捕获的request对象getHeader()方法获取远程IP地址，并调用SessionMap类中get方法检索已存储的用户有效IP。通过断言的方式判断IP地址一致性，若断言结果为真，则该请求来自真实客户端，应用程序将遵循正常业务逻辑流程；反之，安全方面将使用around通知调用HttpServletResponse类中setStatus()方法返回403状态码，并使用proceed()函数改变原应用程序响应流程，从而终止该恶意连接请求。

用户完成操作后在前端界面中点击退出按钮，安全方面使用后置通知捕获HttpSession类中invalidate()方法，该方法强制会话过期，并清空其

保存的对象。若该方法成功执行,则调用SessionMap类中remove()方法,移除此次连接点上绑定的会话信息。

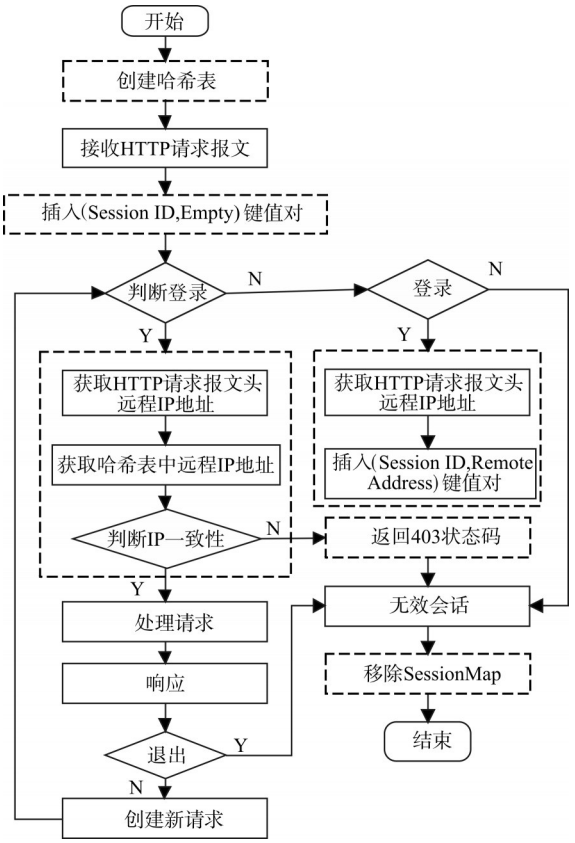

图2 编织后应用程序模型

Fig. 2 Application model after weaving

综上所述,本文完成了如下工作:通过关联远程IP地址与特定会话ID的方式对Web应用程序执行安全检查,防止破坏身份验证与会话管理机制;通过增加用户请求的信任条件以识别真实用户,从而确保响应数据不被非法窃取;当服务器接收来自被盗会话ID用户访问请求时,能有效禁止该请求检索数据或执行任意操作,并且能够不影响真正用户的访问请求;使用AOP方法只需在集中化的方面通知及切点上进行迭代过程,这有助于提升安全维护效率;安全方面API能够在不需修改当前应用程序业务逻辑代码前提下,提升遗产Web应用程序会话管理安全性。

3 结 语

通过安全方面API实现了IP地址与用户身份间的绑定功能,既满足了合法用户的正常访问请求,同时又防止攻击者通过会话劫持及会话固定两种方式对用户凭据的窃取,一定程度上解决了Web应用程序中失效的身份验证与会话管理问题。后续工作中需要研究高效的用户身份鉴别算法,进一步完善基于AOP的Web应用程序安全会话机制,进而提升依赖Cookie及Session的会话管理机制安全性。

参考文献:

[1] SALEH A Z M, ROZALI N A, BUJA A G, et al. A method for web application vulnerabilities detection by using boyer-moore string matching algorithm [J]. Procedia Computer Science, 2015, 72:112-121.

[2] BOUDREAU T, TULACH J, WIELENGA G. Rich client programming: plugging into the NetBeans (TM) Platform [M]. New York:Prentice Hall PTR, 2007.

[3] LIEBERHERR K, LORENZ D H, OVLINGER J. Aspectual collaborations: combining modules and aspects [J]. Computer Journal,2003,46(5): 542-565.

[4] BERGMANS L, LOPES C V. Aspect-Oriented Programming [C]// European Conference on Object-Oriented Programming.Berlin: Springer, 1999:288-313.

[5] KONG A, ZHANG D, KAMEL M. A study of brute-force break-ins of a palmprint verification system [J]. Lecture Notes in Computer Science, 2005, 57 (2): 447-454 .

[6] KRISTOL D, MONTULLI L. HTTP state management mechanism [J]. Rfc, 1997, 11(3):82-89.

[7] NIKIFORAKIS N, MEERT W, YOUNAN Y, et al. SessionShield: lightweight protection against session hijacking [C]// International Conference on Engineering Secure Software and Systems. Berlin: Springer-Verlag, 2011:87-100.

[8] HOWERTON J T. Service-oriented architecture and Web 2.0[J]. It Professional, 2007, 9(3):62-64.

[9] HERMOSILLO G, GOMEZ R, SEINTURIER L, et al. Using aspect programming to secure Web applications [J]. Journal of Software, 2007, 2(6):53-63.

[10] 汪定, 马春光, 翁臣,等. 强健安全网络中的中间人攻击研究 [J]. 计算机应用, 2012, 32(1):42-44.

[11] 王鹏, 季明, 梅强,等. 交换式网络下HTTP会话的劫持研究及其对策 [J]. 计算机工程, 2007, 33(5): 135-137.

[12] 徐兵, 谢仕义. Web应用程序会话安全模块的设计 [J]. 计算机工程, 2008, 34(19):176-178.

[13] 韩坤. Web服务安全会话管理的研究与实现 [D]. 北京:北京邮电大学, 2007.

[14] 刘新亮, 杜瑞颖, 陈晶,等. 针对SSL/TLS协议会话密钥的安全威胁与防御方法 [J]. 计算机工程, 2017, 43(3):147-153.

[15] JADHAV A P, LOHKARE V B. Session fixation vulnerability in web-based application [J]. International Journal of Computer Technology & Applications, 2012, 3(1):62-70.

本文编辑:陈小平