业务逻辑漏洞的利用机理与检测方法研究

王 超,任天宇,李 群,王小虎,师恩洁,李 新
(国网北京市电力公司电力科学研究院,北京 100075)

摘 要:业务逻辑漏洞是近年来Web应用关注的热点问题,因为逻辑漏洞自身的特殊性,很难实现高效精准的自动化发现工具。随着业界对该类漏洞认识的逐步深入,将逐渐实现对其自动检测。本文从业务逻辑漏洞的基本概念入手,分析了逻辑漏洞生成原因,并结合实例介绍了漏洞利用方式。最后介绍了三类逻辑漏洞检测的方法。

关键词:业务逻辑;漏洞;检测方法
中图分类号:TN914 **文献标识码**:A DOI:10.16157/j.issn.0258-7998.2016.S1.019

中文引用格式:王超,任天宇,李群,等.业务逻辑漏洞的利用机理与检测方法研究[J].电子技术应用,2016,42(S1):56-59,63.

0 引言

信息系统的主流架构通常都是由Web应用和数据库组成,在Web应用层中实现的业务规则和工作流、数据流构成了应用系统的业务逻辑。业务逻辑是支撑信息系统实现其预期功能的核心,一旦存在缺陷将导致信息系统无法实现预期功能甚至是重大财务损失。业务逻辑漏洞是仅次于SQL注入和跨站脚本等技术型漏洞的另一类常见漏洞,针对逻辑漏洞的利用通常会造成很严重的损失。随着技术型漏洞问题可通过自动化工具得到较好的解决,逻辑漏洞正在逐步引起企业信息系统管理人员的重视。

本文将从业务逻辑漏洞的定义分类、可能的危害以及发生机理等方面全面描述业务逻辑漏洞,并给出了业务逻辑漏洞检测的方法框架。

1 业务逻辑漏洞简介

业务逻辑用来描述那些处理数据库、Web应用程序和用户界面之间的信息交换的功能算法(业务规则、业务策略和工作流)。业务逻辑由业务规则、工作流组成,它是流程、公式、算法、决策树、方法论、查询或任何其他用于执行应用程序的方法,是对现实生活的业务对象建模。例如账号、贷款、行程和存货。它还规定业务对象如何与其他对象交互。业务逻辑通过访问或更新业务对象来保证执行路径和方法。

业务逻辑漏洞就是允许攻击者操纵应用的工作流或数据流,绕过业务规则,从而实现恶意目的。业务逻辑漏洞通过合法的处理流程来实现对企业信息系统的入侵和攻击。

业务逻辑漏洞具有如下特点:(1)独特性:与SQL注入等这类跨应用仍保持相同特性的漏洞不同,业务逻辑漏洞与应用密切相关。(2)无规律性:业务逻辑在不同的Web应用中表现出各种各样的形式,几乎有无限多的方法来辨认并利用业务逻辑漏洞。(3)人工依赖性:对业务漏洞的检测和利用,都需要人的灵感和超常规的思维。(4)难以防护:很难用附加的防护设备等通用方法去防护并杜绝业务逻辑漏洞。(5)后果严重:业务逻辑与企业的业务活动关系密切,漏洞往往会导致严重的损失。

攻击者利用逻辑漏洞实现欺诈交易,给受害人造成经济上的损失是这类漏洞最典型的危害。利用逻辑漏洞,通常还可以修改任意账户的密码或利用受害用户的Cookie和SessionID仿冒身份,造成用户隐私和身份数据的泄露。结合利用逻辑漏洞还可通过重放攻击用来构造短信炸弹、批量注册用户、批量发送业务数据等,干扰应用的正常运行。

2 业务逻辑漏洞产生原因与利用机理

信息系统的工作过程中包含无数个客户端与服务端之间的数据交互和数据处理环节,如果对客户端提交的数据不做合法性校验和适当限制(次数、频率、执行顺序等),就会存在关键数据被篡改、控制策略被绕过、数据泄露、功能被滥用的情况。大多数业务逻辑漏洞的利用发生在与身份认证(密码找回、身份冒用)或支付(包括网络支付、积分兑换等)相关的环节。

2.1 缺少有效性验证

对客户端提交信息中的关键数据缺乏必要的有效性和一致性验证,是业务逻辑漏洞产生的最主要原因。这些关键数据包括用户身份(UserID)、会话标识(SessionID)、Cookies、交易价格、价格币种、商品ID、订单ID、商家ID等。

2.1.1 篡改关键数据

在电子商务的应用场景中,完成一笔交易,除了买卖双方以外,还涉及电子支付的第三方,整个交易过程中包含多个关键数据,每个关键数据都可以被当作攻击向量,攻击思路都是对关键数据偷梁换柱。这里仅以篡改价格为例说明,如图1,恶意客户向电子支付方发送了篡改价格后的支付请求,第三方支付公司没有进行严

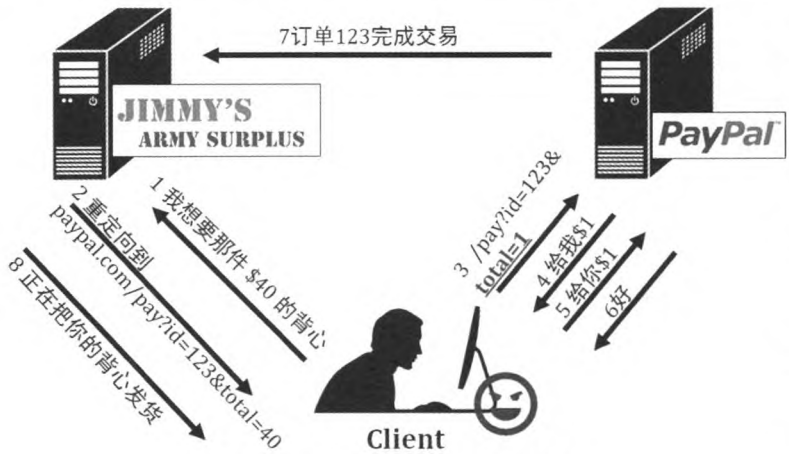

图 1 电商应用中的业务逻辑漏洞利用

格的校验就轻信了攻击者提供的信息。

2.1.2 获取认证信息仿冒登录

Cookie 和 SessionID 是浏览器访问服务器时创建的用于标识用户身份的特殊字符，攻击者设法获得他人的 Cookie 和 SessionID 就有可能冒充受害人登录服务器。有的应用系统中，直接修改 Cookie 中的某个参数可以冒用其他用户身份登录。一种被称为会话固定攻击的方法，利用的就是 SessionID。首先攻击者设法让受害人使用攻击者预先设定的 Session ID 来访问目标应用程序，一旦用户的会话 ID 被成功固定，攻击者就可以通过此 SessionID 来冒充用户访问应用程序。具体过程如下：

(1)首先，Bob 打开一个网站 http://unsafe，然后服务器会回复他一个 SessionID。比如 SID=mjg4qid0wioq。Bob 记下这个 ID。

(2)Bob 给 Alice 发送一个电子邮件，谎称是银行的促销活动链接，引诱 Alice 点击邮件中的链接：http://unsafe/?SID=mjg4qid0wioq。SID 后面是 Bob 自己的 SessionID。

(3)Alice 点击了链接，输入了自己的帐号和口令从而登录到银行网站。

(4)因服务器的 SessionID 不改变，现在 Bob 点击该链接后，他就拥有了 Alice 的身份，可以冒充身份进行恶意操作。

2.1.3 本地验证过程被绕过

将验证过程放在客户端进行，也是一种常见的无效验证情况。例如在某个网站密码重置的过程中，首先需要输入网站帐号捆绑的手机号，点击发送验证短信后，系统向该手机号发出短信。与此同时进行网络抓包或是使用 Firefox 浏览器的 firebug 查看请求链接，如图 2 所示，将发现验证码以明文出现在 URL 中，这样直接输入验证码就可以任意重置密码了。

某知名网站的邮箱密码可通过登录页中的找回密码功能重置，点击下面的“网上申诉”按钮，在申诉页面的源代码里，如图 3 所示，不但有密码提示问题，Hide 表单里竟然泄露问题答案，这样可获得任意用户修改密码的问题和答案，从而轻松修改任意用户邮箱密码。

与上面案例类似，另一个知名网站的密码重置功能也同样泄露了用户信息从而造成任意重置密码的攻击。首先在主站登录处点击登录，随后点击忘记密码，在 http://broker.xxx.com.cn/login/findpassword 界面输入用户名时抓包，可以看到返回数据中含有该用户的手机号，通过解密，得到此用户 shenzhen 的手机号为 182****7672。在密码找回处输入解密后的手机号，并点发送验证码，再次抓包，可发现短信验证码加密后返回给了浏览器，解密后得出验证码为 234589。输入验证码成功重置密码。

2.1.4 修改参数绕过中间步骤

有些系统的业务逻辑可能是按顺序完成 ABCD 四个过程，但是用户可以控制它们给应用程序发送的每一个请求，因此能够通过修改本地参数控制过程的访问顺序。于是，用户就可能从 B 直接进入了 D 过程，而绕过了 C。如果 C 是支付过程，那么用户就绕过了支付过程而买到了一件商品。如果 C 是验证过程，就会绕过验证直接进入网站程序了。

某票务网站就曾出现过这种情况，用户抓包修改某

图 2 抓包活动请求链接的内容

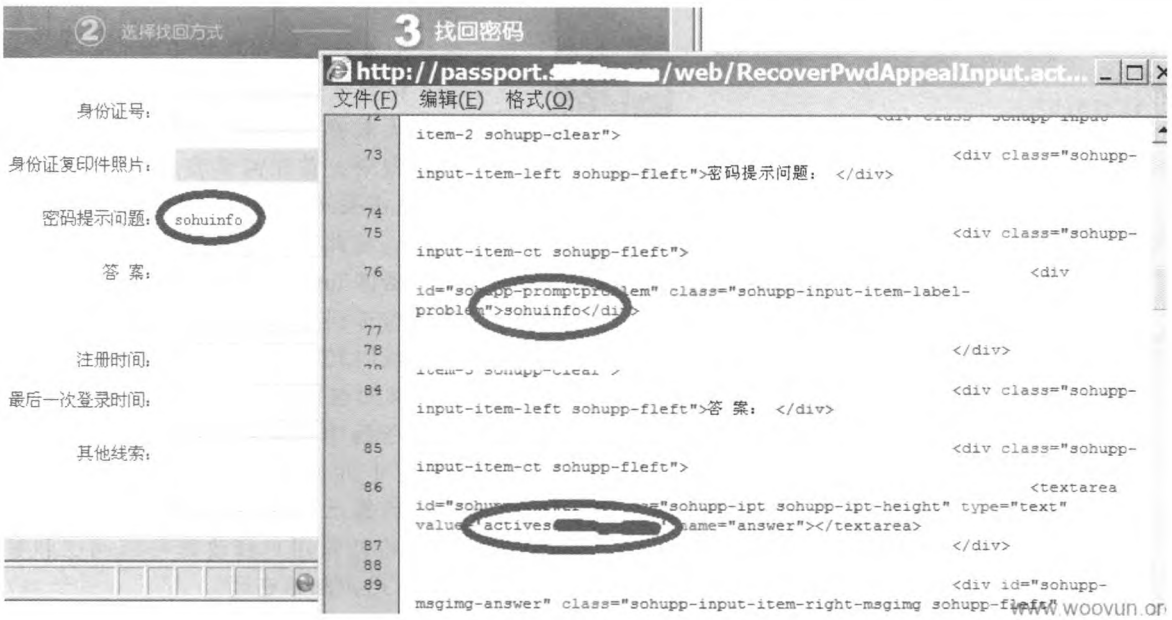

图 3 本地显示密码重置问题和答案

个参数，直接跳过了支付过程，而获得了领取门票的密码。这种漏洞本质上是没有对是否已经支付票款进行认证。

图 4 显示的是通过拦截本地 HTTP 流量，获取别人的 Cookies。当浏览器登录认证时，服务器在 Cookies 中保存各种数值、证书用于每个提交 HTTP 请求的身份认证。其中一个数值被识别出作为用户名，另一个是经过 MD5 哈希处理过的用户密码，通过重放这个请求，只使用 game hash 和 game user，用户就可以通过旁路身份认证过程，像其他用户一样访问游戏。

2.2 缺少必要的限制条件

还有一类业务逻辑漏洞是因为对用户提交的内容（例如在短信、邮件调用业务或生成业务数据环节中）缺少必要的次数和频度等限制，攻击者可以重复生成大量业务数据，形成所谓的“重放攻击”。攻击者可以利用重放攻击快速修改论坛跟贴的数量、批量注册用户、投放短信炸弹等，甚至可以和验证不足的缺陷配合在一起，发起拒绝服务攻击。例如攻击者可以伪造成被攻击者的身份，批量向自动翻译网站的页面上发送请求。有些网站为防止暴力破解，连续输入 5 次错误的密码就冻结账户一段时间。拍卖网站的出低价的用户可以用冒名登录输错密码的方法，将竞价对手排挤在外。

有些业务需要进行频度限制，但是这种限制如果是依赖本地设置的参数来完成，则攻击者可以通过修改本地参数来绕过相关的限制。

3 业务逻辑漏洞的检测方法

由于业务逻辑漏洞一直没有公认的定义，且不同的漏洞作用机理也有很大差异，同时，判断漏洞所必须依据的描述 Web 应用行为的规范也不存在，所以业务逻辑漏洞检测没有公认一致的方法。

目前业界提出的检测方法大致可以分为三类：手工测试、黑箱测试、白箱测试。倾向于使用手工测试方法的人认为，不同的信息系统，有自己特有的业务逻辑，因此业务逻辑漏洞不能像传统的应用漏洞通过扫描器来自动识别。但是由于效率的要求以及对这类漏洞的机理认识的逐渐深入，基于黑箱和白箱的自动测试方法被提出。业务逻辑漏洞的确与其他技术型漏洞不太一样，规律性较差。要想达到最好的检测效果，需要发挥人工和自动两类方法的各自特长，相互结合使用。

```
GET http://                                    &s=build_main HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-excel, application/vnd.ms-powerpoint,
application/msword, application/x-shockwave-flash, application/x-ms-application, application/x-ms-xbap, application/vnd.ms-
xpsdocument, application/xaml+xml, application/x-silverlight, */*
Referer: http://                               &s=build_main&p=2141&build
Accept-Language: el
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 2.0.50727; .NET CLR 3.0.4506.2152; .NET CLR
3.5.30729; FDM; .NET CLR 1.1.4322)
Host:
Cookie: game_hash=cce006dc722ff22ad8a8e5a13fd3c698; SD_FRAMEWORK_SESSION=0b1f74bebf7875e96338e9d4c6e37d4e;
game_user=some.user; game_pass=34718342761522lca90w24db1039a8cc
Proxy-Connection: Keep-Alive
```

图 4 使用哈希的密码旁路身份认证

3.1 手工测试方法

进行手工测试的前提是需要对 Web 服务的工作原理以及被测站点的系统架构非常了解，事先收集必要的识别信息，针对网站的工作流和数据流，以逆向思维方式建立测试思路，对认证授权方案、会话管理、Cookie 属性进行分析，尝试采用暴力破解、模糊测试、重放和竞争条件、权限提升等方法，实现突破网站访问限制规则、流程执行顺序的结果。

手工测试对测试人的个人经验依赖程度较高，所以不同的人，测试结果可能有较大差异。尽管如此，手工测试依然有一定的规律可循。例如，从前面的论述内容来看，业务逻辑漏洞绝大多数集中在与登录认证和网络支付相关的页面，这就为漏洞利用的突破点指出了方向。而具体技术也无外乎是先网络抓包，查看是否有可以篡改的关键数据或是采用重放攻击等方式，尝试提交篡改后的访问请求，观察结果。

根据系统的工作流，可以画出系统的正常业务流程，同时根据经验，可以直观地在图中标识出可能存在的业务逻辑漏洞点（如图 5 所示），然后进行漏洞利用的尝试。

3.2 黑箱测试

OWASP 测试指南 3.0 中，建议在黑箱装置中采用 4 步法来测试逻辑漏洞。(1)测试人通过实际操作和阅读任何有关的文档来研究并理解被测 Web 应用。(2)测试人应准备设计测试用例所需的信息，包括系统示意图实现的工作流和数据流。(3)设计测试用例。(4)准备测试所需环境，创建测试帐号，运行测试并检验结果。如图 6 所示。

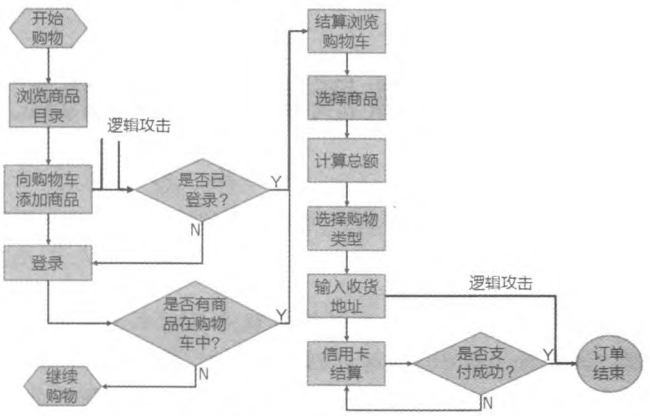

图 5 电子购物流程

黑箱测试的目标是将上述步骤自动化实现在一个黑箱工具中。首先，从一系列包含 HTTP 会话的网络追踪开始，系统推断应用模型并聚合与同一工作流步骤相关的资源；然后抽取一个行为模式集合对工作流和数据流建模；第三，应用攻击模式集合来自动生成测试用例；最后，在被测 Web 应用上执行测试用例，使用一个神秘的标记来检验应用的逻辑是否存在冲突。

3.3 白箱测试

白箱测试通过对源代码的自动化分析，发现程序存在的漏洞。Felmetsger 等人认为，源程序中通常都包括关于开发者意图执行的行为的“线索”，这些线索表现为对变量值和操作执行顺序的约束形式。有两种值得关注的约束，一是表现为程序检查的形式（例如 if 语句），这种约束在某种数据被访问之前往往提示一种线索，要么是允许输入的范围必须有限制，要么是访问某个项目必须

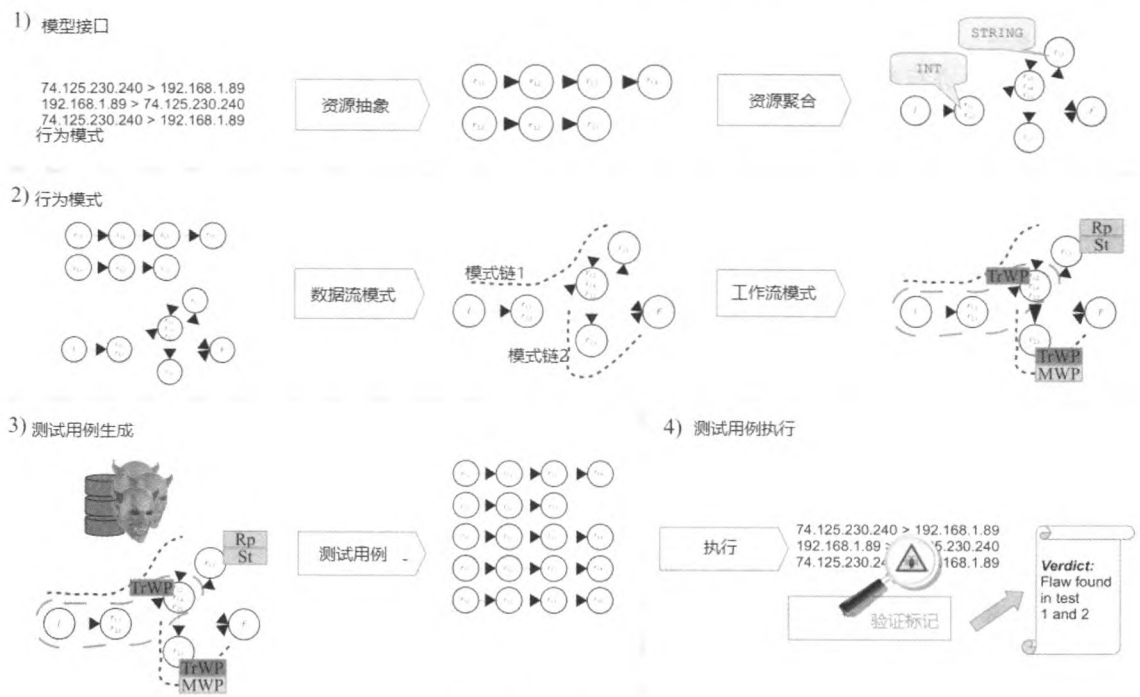

图 6 黑箱测试方法框架

(下转第 63 页)

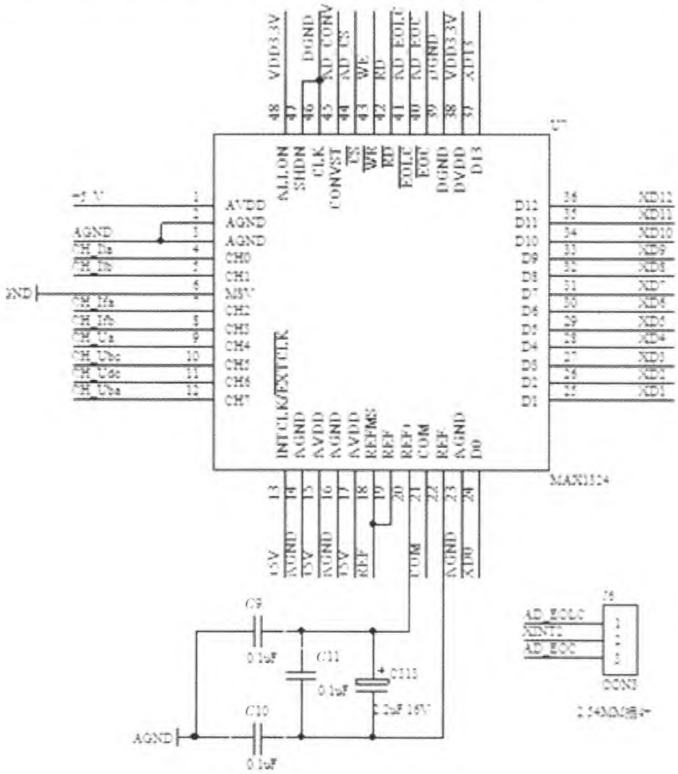

图 5 ADC 转换电路原理图

4 结束语

在电力系统中，随着非线性负荷的不断出现与应用，由此而产生的谐波分量对变压器产生了一定的影响，变压器运行过程中的损耗值也会越来越大。本文设计的兼顾谐波损耗变压器损耗计量仪能够考虑到因谐波造成的损耗，在进行适当的简化计算后，能够比较精确地计算出谐波损耗。由于并没有采用传统的双边测量方式，而是利用单边计量思想，只需在变压器某一侧安装设备进行计量，大大提高了仪器的适用范围，保证了计量精度，同时减小了操作风险，在我国节能降耗、提高能效水平的发展过程中具有良好的应用发展前景。

参考文献

[1] 吴喜红.配电变压器损耗和容量在线检测方法研究[D].重庆：重庆大学，2006.

[2] 韩绍宗.加速淘汰高损配电变压器对节能意义的探讨[J].城市建设理论研究(电子版)，2011，(17).

[3] 林怀德，张勇军，何通.更换高能耗配电变压器经济效益评估[J].电力需求侧管理，2010，12(2)：45-48.

[4] 汪彦良，岳智顺，王金全，等.谐波附加损耗及其降损节能分析[J].电气技术，2009(2)：15-19.

[5] 刘成君，杨仁刚.变压器谐波损耗的计算与分析[J].电力

(下转第 67 页)

(上接第 59 页)

有限制。在另外的程序路径上的类似检查如果缺失，则提示可能是一个漏洞。例如，攻击者可以在没有提供任何凭证的情况下参与特权操作，这种认证旁路的漏洞可以用这种方法检测。第二种约束是存储在后端数据库中的数据与存储在用户会话中的数据的隐含关联关系。更特别的是在 Web 应用中，数据库通常存储持久数据，用户会话存储最经常访问的部分数据，例如用户证书。这样当数据库查询被提交时，数据库数据和会话数据经常存在隐含约束。如果在一个路径上这种关系冲突被发现，某种应用逻辑漏洞(如非授权编辑属于其他人 POST 的数据)可以被检测到。在实现层面，用不变量代表前面所说的对变量取值的约束，如果在至少一个程序路径上的限制检查某个不变量的执行是正确的，则称这个不变量为支持不变量。如果在程序的其他路径上有与支持不变量的冲突，则提示可能存在漏洞。

4 结论

随着 SQL 注入等技术型漏洞问题的逐步解决，业务逻辑漏洞将成为 Web 安全领域下一步重点关注的问题。由于逻辑漏洞的特殊性，针对逻辑漏洞的检测完全依赖自动化方法还有一定难度，比较准确的检测方式还是手工。手工与自动化工具结合是兼顾效率和准确性的好方法。研究更准确的自动化检测逻辑漏洞的方法将是未来研究的课题。

参考文献

[1] PELLEGRINO G，BALZAROTTI D.Toward black-box detection of logic flaws in web applications[C].In Network and Distributed System Security Symposium，2014：23-26.

[2] SECPULSE.业务安全漏洞挖掘归纳总结[EB/OL].(2015-07-03).www.secpulse.com/archives/34540.html.

[3] Wang Rui，Chen Shuo，Wang Xiaofeng，et al.How to shop for free online security analysis of cashier-as-a-service based web stores[C].Proceedings in the 2011 IEEE Symposium on Security and Privacy，2011：465-480.

[4] Sun Fangqi，Xu Liang，Su Zhendong.Detecting logic vulnerabilities in e-commerce applications[C].Network and Distributed System Security Symposium，2014.

[5] STERGIOPOULOS G，TSOUMAS B，GRITZALIS D.Hunting application-level logical errors[J].Engineering Secure Software and Systems，2012，7159：135-142.

[6] FELMETSGER V，CAVEDON E，KRUEGEL C.Toward automated detection of logic vulnerabilities in web applications[C].Usenix Security Symposium，2010：143-160.

[7] MORANA M.How to prevent business flaws vulnerabilities in web application[Z].

[8] COVA M，BALZAROTTI D，FELMETSGER V，et al.Swaddler：an approach for the anomaly-based detection of state violations in web applications[J].The Series Lecture Notes in Computer Science，2007，4637：63-86.