

基于 Web 应用的网络安全漏洞发现与研究

张晓双, 徐依凌, 刘渊

(江南大学数字媒体学院, 江苏 无锡, 214122)

摘要: Web 安全漏洞可分为 2 类, 即基于 Web 平台和基于 Web 自身应用的安全漏洞。通过分析 Web 应用网络安全漏洞的攻击原理和攻击过程, 研究了包括反射型、存储型和文档对象模型 (DOM, document object model) 的跨站脚本 (XSS, cross site scripting) 漏洞和结构化查询语言 (SQL, structured query language) 注入漏洞, 以及会话认证管理漏洞, 提出了基于 3 种不同漏洞的相应的防范措施。

关键词: Web 应用; 跨站脚本; SQL 注入; 会话认证管理

中图分类号: TP309

文献标识码: A

doi: 10.11959/j.issn.2096-109x.2016.00065

Discovery and research of network security vulnerabilities based on Web application

ZHANG Xiao-shuang, XU Yi-ling, LIU Yuan

(School of Digital Media, JiangNan University, Wuxi 214122, China)

Abstract: Web security vulnerabilities can be divided into two categories, including security vulnerabilities Web platform and Web their own application. By analyzing the attack principle and process of Web application network security vulnerabilities, XSS vulnerability including type of the reflective, stored, and DOM, SQL injection vulnerability and session authentication management vulnerability were studied. The corresponding preventive measures of the three kinds of vulnerabilities were put forward.

Key words: Web application, cross site scripting, SQL injection, session authentication management

1 引言

随着互联网全球化, 网络安全形势日益严峻, 各种安全漏洞层出叠见。CNNVD 自 2009 年 10 月 18 日正式上线收集漏洞以来, 每年新增漏洞数量呈明显的上升趋势。从 2010 年收录漏洞 4 636 个到 2015 年收录漏洞 7 754 个, 平均每年收录 6 781 个。截至 2015 年底, CNNVD 总共收录漏

洞高达 80 300 个。跨站脚本 (XSS) 攻击、SQL 注入攻击、失效的会话认证管理这 3 类漏洞占比高、破坏力强, 占网络安全漏洞的半壁江山。

目前, 网络安全领域受到广大学者的重视。Parvez 等^[1]研究了黑盒子 Web 应用扫描器对存储型 XSS 漏洞的检测作用。Shar 等^[2]针对 XSS 漏洞, 提出一种规范的审计模型, 该模型提取了所有目前已有的防御编码方法, 确保检测到每个潜在的

收稿日期: 2016-04-17; 修回日期: 2016-05-30。通信作者: 张晓双, 786002308@qq.com

基金项目: 江苏省自然科学基金资助项目 (No.BK20151131); 中央高校基本科研业务费专项基金资助项目 (No.JUSRP51614A)

Foundation Items: The Natural Science Foundation of Jiangsu Province (No.BK20151131), The Fundamental Research Funds for the Central Universities (No.JUSRP51614A)

脆弱的 HTML 输出。Sadeghian 等^[3]全面分析了不同类型 SQL 注入的检测方法和防范技术,并做了结构分类。Sathyanarayan 等^[4]根据 SQL 语法,对恶意的 SQL 查询语句进行枚举,并总结了恶意的查询模式。Yoon 等^[5]基于椭圆曲线离散对数问题,研究出一种新的针对会话初始协议(SIP)的认证机制。

本文基于 Linux 测试平台配合多种测试工具,结合具体网络漏洞,研究了3种主流网络安全漏洞的基本原理、攻击方式和防御措施。

2 渗透测试平台及工具

2.1 实验平台

本文工作主要在 Ubuntu 和 Kali 这2个操作系统上进行。Ubuntu 是一个基于 Linux 的开源系统。Kali 是一个基于 Debian 的 Linux 发行版,它自带很多用于渗透测试的工具,支持 ARM 架构,主要用于黑客攻击和渗透测试。

2.2 测试工具

在评估一个网络应用安全性能时,准确使用各种测试工具非常重要。本文主要使用漏洞检测框架 Metasploit、Web 应用扫描工具 W3AF、网络数据监控工具 FireBug。

Metasploit 根据明确的攻击目标,创建或从库文件中选择相应的漏洞代码并执行攻击,其体系结构如图1所示。

W3AF 主要用于 Web 应用攻击和检查,目前已拥有130多个插件。按其功能,插件可分为9个模块,分别是审计模块(audit)、认证模块(auth)、破解模块(bruteforce)、爬虫模块(crawl)、逃逸模块(evasion)、搜索模块(grep)、基础模块(infrastructure)、修改模块(mangle)、输出模块(output)。各个模块的运作关系如图2所示。

3 XSS 漏洞分析与检测防御

XSS 攻击是对动态网站常见的一种攻击方式,由于对用户提交的输入数据过滤不严格,导致攻击者可以将恶意脚本注入到网页上。由于 XSS 攻击脚本注入方式的不同,产生了反射型、存储型、DOM 型3类漏洞^[6],本文以存储型 XSS 漏洞为例进行分析。

3.1 存储型 XSS 漏洞测试

攻击者提交的恶意代码没有被过滤或净化而直接保存在 Web 服务器,一旦有用户访问被攻击的页面,恶意代码就会执行并实现攻击,这就是存储型 XSS 攻击。WordPress 是以 PHP 作为开发语言的开源博客平台,其 WordPress Googmonify Plugin 0.8.1 版本存在一个存储型漏洞。

1) 漏洞攻击。配置本地服务器,按需求配置 WordPress 使其可正常使用,并安装有漏洞的插件版本 Googmonify 0.8.1。最后,编写攻击代码并

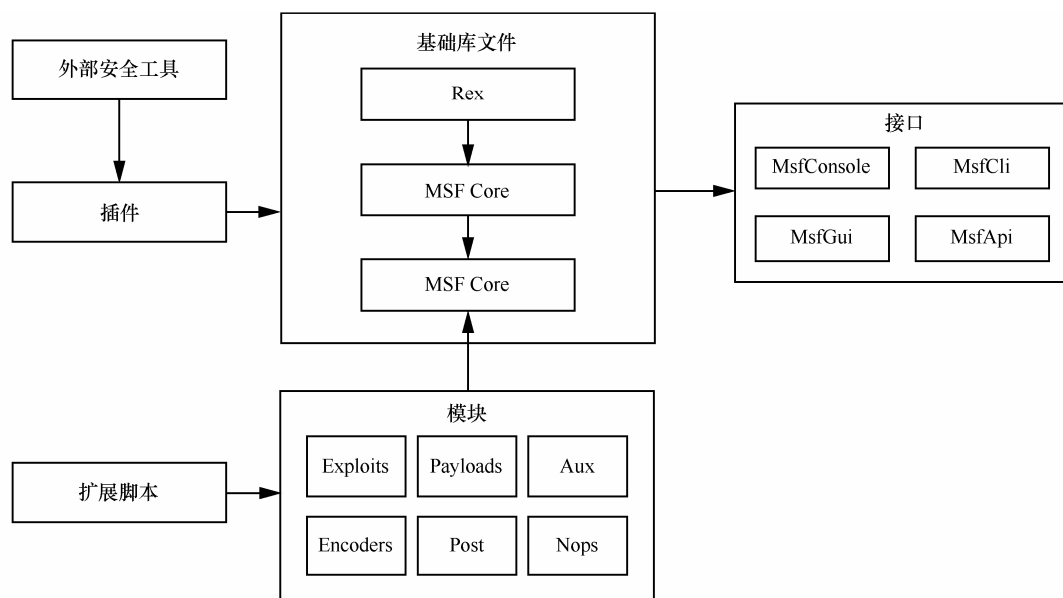


图1 Metasploit 体系结构

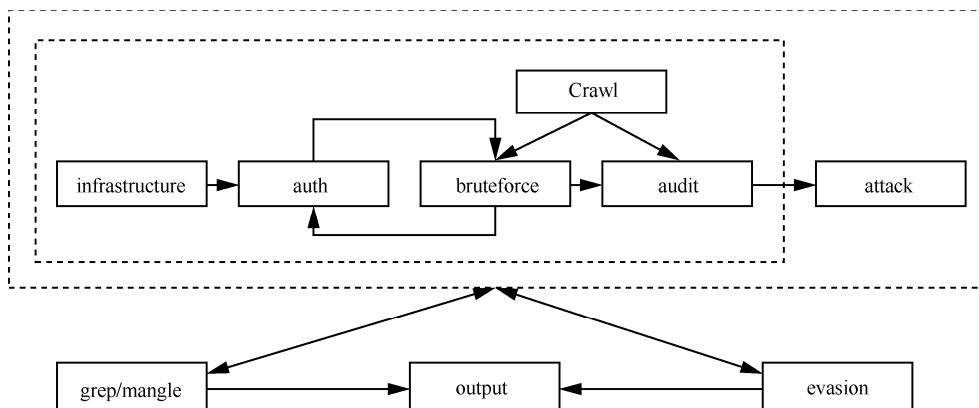


图2 W3AF 运作关系

在浏览器上执行，当执行代码后跳转到 WordPress 的登录界面，一旦用户输入账号、密码登录，就实现了 XSS 攻击。测试结果如图 3 所示。

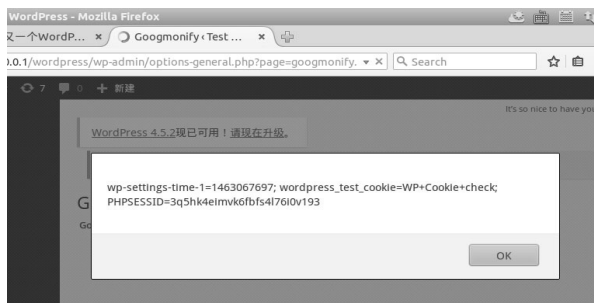


图3 存储型 XSS 漏洞攻击效果

2) 漏洞特征源码。部分漏洞源码如下。

```
<!--/wordpress/wp-content/plugins/googmonify
y/googmonify.php -->
<td><input id="PID" name="PID" type="text"
value="<?php echo $pid;?>"></td>
<td><input id="Limit" name="Limit" type="
text" value="<?php echo $limit;?>" size="5"></td>
```

这个漏洞产生的原因是 googmonify.php 页面面对输入参数 pid、limit 和 aid 没有进行严格过滤，直接向服务器提交并返回给响应，导致攻击者可以给它们传递恶意脚本。除了仿冒用户与服务器通信并获取敏感信息，攻击者还可以直接在恶意脚本中嵌入一个 URL，将用户界面跳转到攻击者设计的恶意网址，进行进一步的攻击。

3) 漏洞修复。类似反射型 XSS 漏洞，可以用 htmlspecialchars() 函数把输入参数 pid、limit 等中含有的一些特殊字符转化为 HTML 实体，把

输入的参数当作 HTML 内容来处理，而不是当作结构来处理。

4) 漏洞攻击分析。通过对 Googmonify 0.8.1 漏洞的剖析可知，入侵者可以在远程的 Web 页面中提交恶意脚本，一旦这些攻击后的 Web 网页被访问，入侵者在其中嵌入的恶意脚本将会被执行，达到入侵目的。其攻击实施步骤如图 4 所示。

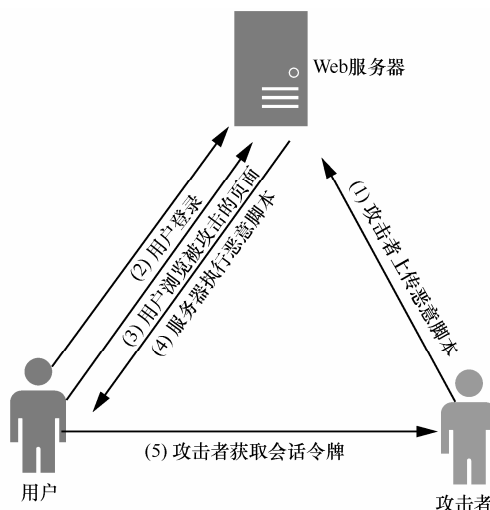


图4 存储型 XSS 漏洞攻击实施步骤

就危害程度而言，存储型 XSS 漏洞比反射型 XSS 漏洞的威胁更大。利用反射型 XSS 漏洞，攻击者需要借助某种媒介，如钓鱼邮件，将设计好的 URL 发送给用户，诱骗用户点击恶意 URL，在这个过程中，就会降低攻击的成功率。而对于存储型 XSS 漏洞，恶意脚本已经被保存在服务器上，一旦被攻击后的 Web 网页被访问，入侵者在其中嵌入的恶意脚本将会被执行。

3.2 XSS 漏洞防御措施

反射型 XSS 漏洞和存储型漏洞存在的根本原因均是用户提交的数据未经过滤直接被复制到输出响应中。DOM 型 XSS 漏洞的根本原因是用户提交的数据未经过滤直接由客户端浏览器处理。为了避免这 3 种类型的 XSS 漏洞,开发人员可以在开发过程中可以从以下 3 个方面入手。

1) 控制输入。应用程序在收到用户在页面上输入的内容后,需要对这些数据进行严格的验证和转义。首先,确保数据的长度在一定范围内,因为嵌入了恶意脚本的代码长度往往会超出正常长度;其次,对特殊字符进行严格过滤,如<、>、/等常用的脚本字符;最后,对不同的输入制定不同的类型要求。

2) 控制输出。开发人员应该尽量避免将用户的输入不经任何处理直接复制到响应中,应对这些输入进行 HTML 编码,把输入的参数当作 HTML 内容来处理,而不是当作结构来处理。

3) 避免敏感插入点。对于反射型和存储型 XSS 漏洞,应用程序中存在一些位置,如果这些位置存在外部接口允许用户输入内容,就容易产生 XSS 漏洞。如在 JavaScript 脚本中,标签的 href 属性中。应尽量寻找其他方法替代相应的输入功能。对于 DOM 型 XSS 漏洞,尽可能避免直接将 DOM 数据插入到页面中,并使用客户端的脚本来处理。

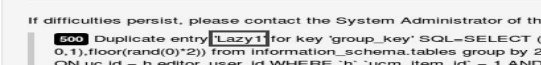
4 SQL 注入漏洞分析与检测防御

SQL 注入的基本原理是攻击者利用数据库的一些外部函数接口,将 SQL 查询命令传输到数据库服务器,然后在数据库内执行这个查询,导致数据丢失或劫持用户服务器。

4.1 SQL 注入漏洞测试

Joomla 是一套基于 PHP 平台、在全球认可度很高的内容管理系统,其 Joomla 3.2~3.4.4 版本存在 SQL 注入漏洞。

1) 漏洞攻击。配置服务器并安装 Joomla 实例范本。在浏览器提交含有 SQL 查询语句的 URL,分别获取用户的账号和密码的散列值。攻击成功后可以获取管理员的用户名和管理员密码的散列值,如图 5 和图 6 所示。

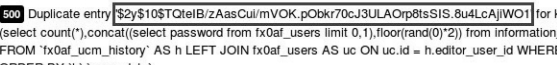


If difficulties persist, please contact the System Administrator of the site.

500 Duplicate entry for key 'group_key' SQL=SELECT (0,1),floor(rand(0)*2)) from information_schema.tables group by 2 ON uc.id = h.editor_user_id WHERE 'h'.ucm_item_id' = 1 AND

图5 管理员用户名获取结果

If difficulties persist, please contact the System Administrator of this site and report the error below.



500 Duplicate entry for key 'group_key' SQL=SELECT (select count(*),concat((select password from fx0af_users limit 0,1),floor(rand(0)*2)) from information_schema.tables group by 2 ON uc.id = h.editor_user_id WHERE 'h'.ucm_item_id' = 1 AND ORDER BY 'h'.save_date

图6 管理员密码散列值获取结果

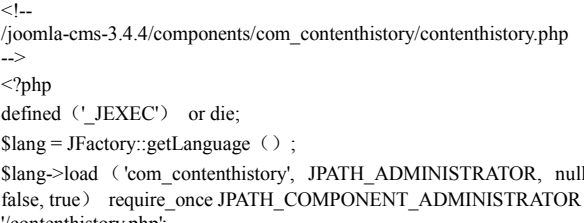
对比网站后台数据库,如图 7 所示。可以证明 SQL 注入获取结果的正确性。



Options	id	name	username	email	password	block	sendEn
☐ Edit ☐ Copy ☐ Delete	730	Super User	Lary	linda5252@qq.com	\$2y\$10\$TqteIB/zAasCui/mVOK.pObkr70cJ3ULAOrp8tsSIS.8u4LcAjlWO1	0	

图7 网站后台数据库

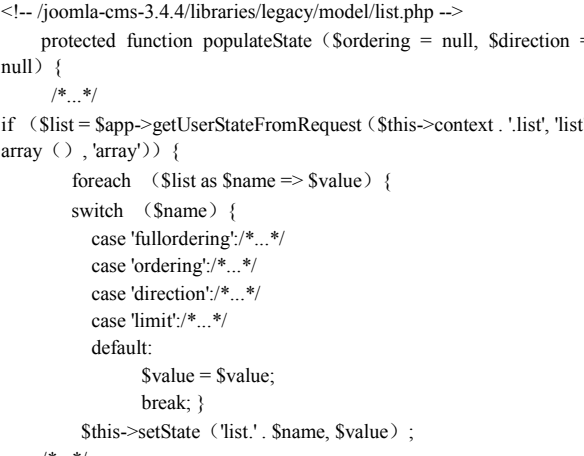
2) 漏洞特征源码。部分漏洞源码如下。
contentthistory.php 页面如图 8 所示。



```
<!-- /joomla-cms-3.4.4/components/com_contenthistory/contenthistory.php -->
<?php
defined ( '_JEXEC' ) or die;
$Jlang = JFactory::getLanguage ( );
$Jlang->load ( 'com_contenthistory', JPATH_ADMINISTRATOR, null, false, true );
require_once JPATH_COMPONENT_ADMINISTRATOR . '/contenthistory.php';
```

图8 contentthistory.php 页面

List.php 页面如图 9 所示。



```
<!-- /joomla-cms-3.4.4/libraries/legacy/model/list.php -->
protected function populateState ( $ordering = null, $direction = null ) {
/*...*/
if ( $list = $app->getUserStateFromRequest ( $this->context . '.list', 'list', array ( ), 'array' ) ) {
foreach ( $list as $name => $value ) {
switch ( $name ) {
case 'fullordering': /*...*/
case 'ordering': /*...*/
case 'direction': /*...*/
case 'limit': /*...*/
default:
$Svalue = $value;
break; }
$Sthis->setState ( 'list.' . $name, $value );
}
/*...*/
```

图9 List.php 页面

从图 8 和图 9 的特征源码以及图 10 的数据流可以看出,产生该漏洞的根本原因有 2 点。①对于 contentthistory.php 页面,未对组件访问权限进行严

格控制。历史编辑版本组件 `com_contenthistory` 应该只有管理员才有访问权,但实际加载该组件时,没有进行相关权限检测,而是直接被 `load`。②在 `list.php` 页面中,开发者对参数做了解析并设置了过滤,但却忽略了 `list[select]` 参数最后会被解析到 SQL 语句构建中的 `list.select`。攻击者就可以为此参数构造 SQL 查询语句,注入到执行流。

3) 漏洞修复。调用 `$inputFilter->clean` 对参数 `list[select]` 进行过滤,其中, `$inputFilter = JFilt-`

`terInput::getInstance()`, Joomla 中已经定义了 `JFilterInput` 类,用于对数据源中的输入进行过滤。

4) 漏洞攻击分析。从上述漏洞剖析可知,通过将 Web 客户端提交的恶意 SQL 查询命令传输到数据库服务器进行数据库检索,攻击者可以获得敏感信息,甚至是控制整个数据库服务器。其攻击实施过程如图 11 所示。

4.2 SQL 注入漏洞防御措施

为了避免 SQL 注入漏洞,开发人员在开发过

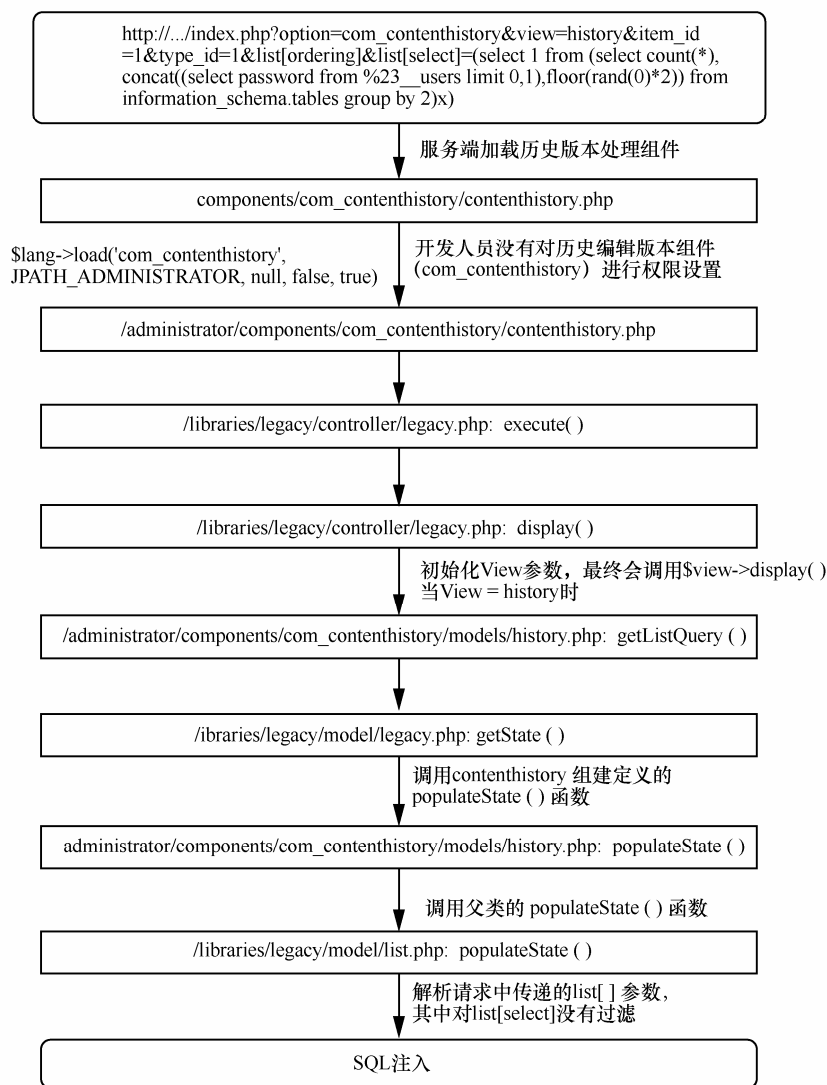


图 10 SQL 注入的数据流

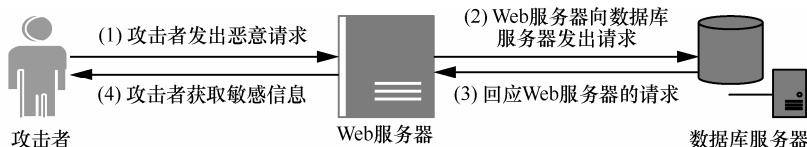


图 11 SQL 注入漏洞攻击实施方式

程中可以从以下 3 个方面入手。

1) 输入控制。很多 SQL 注入漏洞都是由单引号造成的。攻击者往往在插入的数据中先用一个单引号匹配原查询语句中已存在的一个单引号, 然后再在后面输入其他的查询语句, 因此, 在将用户的输入插入数据库服务器前, 应先匹配单引号; 其次, 需要对输入进行预处理, 过滤输入中包含的注释字符、转义序列等一些特殊的字符; 最后, 控制输入数据的长度在一定范围内, 将不同的输入强制转换为相应的类型。

2) 使用参数化查询。参数化查询指的是应用程序已经确定查询结构, 在需要用户输入的地方预先设置占位符。随后将用户的输入数据填充到预留的占位符。这样做的好处是, 即使攻击者精心设计数据, 也无法破坏查询结构, 因为查询结构已经被确定。

3) 控制访问权限。应用程序应该根据用户的身份和需求设置相应的访问权限。比如, 一般用户只需要读取和查询自己的数据库。但对于管理员而言, 则需要拥有更高的权限来管理好整个系统的数据库。

5 会话认证管理漏洞分析与检测防御

会话管理攻击是攻击者通过某种方式劫持用户的会话, 在获取会话令牌后, 伪装成这位用户获取该用户的敏感信息和相应权限。如果会话令牌是可预测的, 攻击者甚至可以猜测其他用户的会话令牌, 扩大攻击范围。

5.1 会话认证管理漏洞测试

Yxcms 是基于 PHP 平台的开源企业建站系统, Yxcms 1.1.3 版本存在会话认证管理漏洞。

1) 漏洞攻击。配置服务器并安装漏洞版本 Yxcms 1.1.3。构造一个恶意 URL, 将管理员会话的 SESSIONID 设置为本文构造的值 1234qwer。利用社会工程学方法, 诱骗管理员点击这个恶意 URL。管理员点击后, 由于重置了会话, 会重新跳到登录界面。当管理员再次登录后, 相当于对这个 SESSION 做了授权处理, 被攻击者的 SESSIONID 将变成本文构造的 1234qwer。利用这个 SESSIONID, 可以获取管理员的身份。攻击结果如图 12~图 14 所示。



图 12 正常捕获的管理员 SESSIONID



图 13 会话重置后捕获的管理员 SESSIONID



图 14 利用 SESSIONID 后获得的管理员界面

2) 漏洞特征源码。部分漏洞源码如图 15 所示。

```
<!--
/YxcmsApp-1.1.3-BETA/protected/apps/admin/controller/common
Controller.php -->
<?php
class commonController extends BaseController{
    public function __construct()
    {
        parent::__construct();
        if(!empty($_GET['phpsessid']))session_id($_GET['phpsessid']);
        @session_start();
        $config['AUTH_LOGIN_URL']=url('admin/index/login');
        $config['AUTH_LOGIN_NO']=array('index'=>
        array('login','logout','verify'),'common'=>'*');
        $config['AUTH_POWER_CACHE']=false;
        Auth::check($config);
    }
}
/*...*/
```

图 15 部分漏洞源码

由特征源码可知,这个漏洞存在的原因是 Yxcms 允许用户通过 GET 方法自定义会话,这样,攻击者就可以直接为管理员设计一个会话,而不需费劲心思劫持用户会话,猜测它的会话令牌。然后,攻击者将包含会话令牌的 URL 发送给管理员,诱骗其点击。一旦管理员点击这个 URL 并重新登录,他的会话令牌就变成攻击者设计的值。这样攻击者可以利用自己设计的会话令牌获取管理员身份,控制后台。

3) 漏洞修复。从代码分析中可以看出,GET 方法除了从外部接收一个会话令牌,没有其他作用,且该功能实际上是不需要的,所以,可以直接删除从外部获取会话令牌的接口。

4) 漏洞攻击分析。会话认证漏洞产生的根本原因是攻击者获得了用户的会话令牌,且该令牌在一段时间内是有效的。会话令牌常见的获取方式是劫持正在会话的用户。会话劫持是攻击者窃取 Web 浏览器和服务器连接的会话,具体流程如图 16 所示。

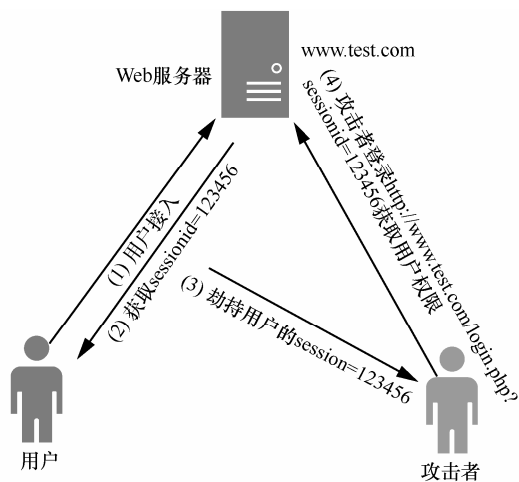


图 16 会话劫持过程

上述漏洞就是一个变相的会话固定,只不过攻击者发送给管理员的会话是直接构造的,而不是攻击者自己先建立的。会话固定是会话劫持的一种方式,会话固定是攻击者自己先建立一个会话,然后诱使其他用户登录此会话,帮助自己获得认证,最后冒充该用户。上述漏洞具体实施过程如图 17 所示。

5.2 会话认证管理漏洞防御措施

根据产生会话管理漏洞的 3 个原因,开发人员在开发过程中可以采取相应的防御措施,具体如下。

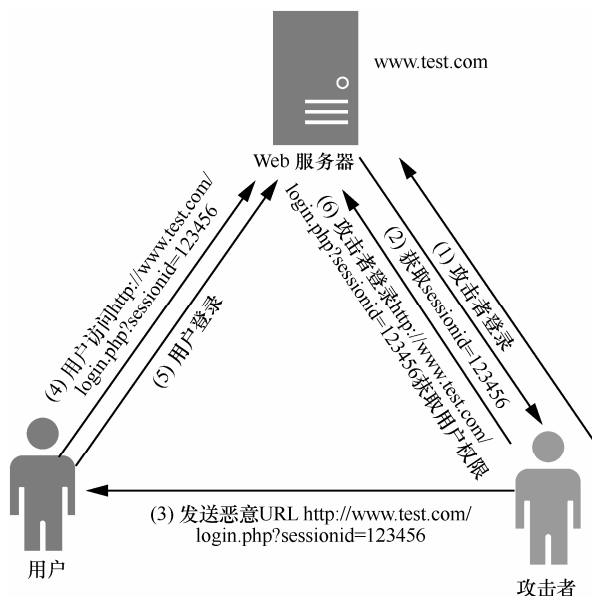


图 17 会话固定劫持过程

1) 确保会话令牌顽健性。Web 应用程序生成的会话令牌应使攻击者无法猜测或推算。在生成令牌时,可以使用一个强大的随机源,确保令牌无法预测。

2) 防止会话令牌泄露。在一个令牌产生后,必须保证它在整个生命周期的安全性。①尽可能使用 HTTPS 传送会话令牌,如果使用 HTTP Cookie 传送令牌,则必须标记 Cookie 安全,杜绝浏览器通过 HTTP 传送它们。②不要在 URL 中传送会话令牌,因为这样就相当于直接把令牌暴露给攻击者。③如果应用程序有管理会话令牌的功能,则应该对该功能设置严格的访问权限,对于包含令牌的日志,也需要有严格的访问权限。④确保客户端安全,对于依赖 HTTP Cookie 的会话管理机制,客户端的 XSS 漏洞、CSRF 漏洞等都会造成 Cookie 泄露,所以,必须防止在客户端产生漏洞。

3) 设置会话令牌的有效期。首先,应杜绝使用静态的会话令牌;其次,会话处于非活动状态一段时间内,应用程序应该要自动结束会话;最后,应用程序要确保能正确执行退出功能,一旦用户退出,应用程序要立刻删除服务器上所有关于本次会话的资源,并结束会话。

6 结束语

本文主要研究了 XSS 攻击、SQL 注入和失效

的会话认证管理这 3 类网络安全漏洞。通过分析 5 个实际的安全漏洞,研究了它们的攻击原理、攻击方式,并提出了针对不同类型漏洞的防范措施。

参考文献:

- [1] PARVEZ M, ZAVARSKY P, KHOURY N. Analysis of effectiveness of black-box web application scanners in detection of stored SQL injection and stored XSS vulnerabilities[C]//The 10th International Conference for Internet Technology and Secured Transactions (ICITST), IEEE. c2015: 186-191.
- [2] SHAR L K, TAN H B K. Auditing the XSS defence features implemented in Web application programs[J]. Software, IET, 2012, 6(4): 377-390.
- [3] SADEGHIAN A, ZAMANI M, MANAF A A . A taxonomy of SQL injection detection and prevention techniques[C]//2013 International Conference on Informatics and Creative Multimedia (ICICM), IEEE. c2013: 53-56.
- [4] SATHYANARAYAN S, QI D, LIANG Z, et al. SQLR: grammar-guided validation of SQL injection sanitizers[C]// 2014 19th International Conference on Engineering of Complex Computer Systems (ICECCS), IEEE. c2014: 154-157.
- [5] YOON E J, YOO K Y. A new authentication scheme for session initiation protocol[C]//International Conference on Complex, Intelligent and Software Intensive Systems, IEEE. c2009: 549-554.
- [6] 吴晓恒. 跨站脚本攻击的防御技术研究[D]. 上海: 上海交通大学, 2011.
WU X H. Defense technology research of cross site scripting at-

tack[D]. Shanghai: Shanghai Jiaotong University, 2011.

作者简介:



张晓双(1991-),女,湖北咸宁人,江南大学硕士生,主要研究方向为社交网络。



徐依凌(1993-),女,浙江绍兴人,江南大学本科生,主要研究方向为 IOT 安全。



刘渊(1967-),男,江苏无锡人,江南大学教授,主要研究方向为网络安全、计算机网络。