公 开

2016 年第一季度开源软件源代码安全漏洞分析报告

国家互联网应急中心

实验室

2016 年 4 月

目录

1 概述

近年来，随着软件技术飞速发展，开源软件已在全球范围内得到了广泛应用。数据显示，从 2012 年起，已有超过 80%的商业软件使用开源软件。开源软件的代码一旦存在安全问题，必将造成广泛、严重的影响。为了解当前开源软件的安全情况，实验室本季度对 30 款广泛使用的知名开源软件进行了源代码安全测试。结合漏洞扫描工具和人工审计的结果，形成了本漏洞分析报告。本次测试在代码层面共发现高危安全漏洞 2647 个。与上季度的结果相比，这些开源软件存在的安全问题依然严重。

2 被测开源软件

表 1 列出了本次被测的 30 个开源软件项目的概况，涵盖了 PHP，JAVA， C，JavaScript 四种编程语言。这些开源软件项目都是国际、国内知名的，拥有广泛用户的软件项目，其中不乏由知名软件公司开发的软件。本文列出了被测项目在 Github 上的 star 数量[1]（表 1 最右列）——通常认为，项目的 star 越高，则项目被越多的开发者关注，其影响力也越大。由于这些软件大多具有巨大的用户群体，软件中的安全漏洞很可能会造成严重的后果。

本次检测的所有软件均为当前最新版本，其中 12 款软件为上季度报告中被测软件的升级版本（表 1 第二列给出了版本号升级详情），本期报告新增了对这几款软件升级前后安全情况的对比分析（见第 5 章）。

[1]一般来说，star 数高于 1000 的项目被认为是流行项目，star 数高于 10000 的，被认为是非常著名的“明星”项目。

表 1 被测开源软件项目概览

项目名称	版本号(括号中为上季度版本)	主要编程语言	功能说明	功能分类	代码行数(L)	Github star 数
async-http-client	1.9.34 （1.9.32）	JAVA	异步 HTTP 请求处理框架	开发框架	36278	2747
Cachet	v2.1.2	PHP	开源的状态页面系统	测试及部署工具	6911	3991
Cakephp	3.2.3 （3.2）	PHP	Web 应用程序开发框架	开发框架	26325	6357
composer	1.0.0-beta1	PHP	php 依赖管理工具	测试及部署工具	35475	7241
core	v9.0.0	PHP	ownCloud 的核心框架	开发框架	233157	4265
daux.io	0.1.0	PHP	项目文档生成器	测试及部署工具	2669	3889
druid	1.0.18 （1.0.16）	JAVA	数据库访问组件	开发框架	240579	3000
Faker	v1.5.0	PHP	生成数据 PHP 库	测试及部署工具	39974	8022
fish-shell	2.2.0	C	面向 OSX、linux 的智能和用户友好的命令行界面	办公及其他 WEB 应用	37075	5661
google-api-php-client	v2.0.0-RC6	PHP	一个PHP访问谷歌的 API 客户端库	开发框架	173714	3375
grav	1.0.10	PHP	平面文件系统	开发框架	9054	3868
greenDAO	2.1.0 （1.3.7）	JAVA	移动开发的 ORM 框架	开发框架	18417	4236
guzzle	6.1.1	PHP	PHP HTTP 客户端	测试及部署工具	3035	6540
jq	1.5	C	灵活的轻量级命令行 JSON 处理器	开发框架	15934	6335
Jquery	2.2.1 （2.2.0）	JavaScript	轻量级的 javascript 开发库	开发框架	34326	38747
junit5	ALPHA	JAVA	Java 语言的单元测	测试及部署工具	19086	330

			试框架			
Koel	2.0 （1.1）	PHP	基于 Web 的个人音乐流媒体应用	办公及其他 WEB 应用	2228	5718
laravel	5.2.15（5.2.0）	PHP	PHP Web 开发框架	开发框架	368	22395
monolog	1.18.1	PHP	记录日志组件	开发框架	9925	4482
okhttp	3.2.0 （3.0.1）	JAVA	HTTP+SPDY 客户端开发包	开发框架	47235	9874
Pdf.js	1.4.11（1.3.88）	JavaScript	在线 PDF 阅读框架	办公及其他 WEB 应用	25620	13983
phalcon	v2.0.10	PHP	php 框架	开发框架	51932	6500
Phpbb	3.2.0-a1（3.1.x）	PHP	网络论坛框架	开发框架	157384	853
Phpmyadmin	4.5.5（4.5.3.1）	PHP	MySQL 数据库管理工具	测试及部署工具	236343	1602
phpunit	5.2.12	PHP	PHP 的单元测试框架	测试及部署工具	19210	5056
piwik	3.x-dev	PHP	是一个 PHP 和 MySQL 的开放源代码的 Web 统计软件	测试及部署工具	185009	5217
scribejava	2.3.0	JAVA	简单的 JAVA 版 OAUTH 库	开发框架	8938	3199
Slim	3.3.0	PHP	PHP 框架	开发框架	2474	6509
symfony	3.0.2 （3）	PHP	基于 PHP 的 Web 开发框架	开发框架	172258	12011
yii2	2.0.7	PHP	PHP 框架	开发框架	32177	7696

3 测试内容

3.1 安全漏洞种类

本次测试涵盖各类常见安全漏洞。根据安全漏洞形成的原因、被利用的可能性、造成的危害程度和解决的难度等因素进行综合考虑，可以将常见的安全漏洞分为八类：

1) 输入验证与表示（Input Validation and Representation）

输入验证与表示问题通常是由特殊字符、编码和数字表示所引起的，这类问题的发生是由于对输入的信任所造成的。这些问题包括 :缓冲区溢出、跨站脚本、SQL 注入、命令注入等。

2) API 滥用（API Abuse)

API 是调用者与被调用者之间的一个约定，大多数的 API 滥用是由于调用者没有理解约定的目的所造成的。当使用 API 不当时，也会引发安全问题。

3) 安全特性（Security Features）

该类别主要包含认证、访问控制、机密性、密码使用和特权管理等方面的漏洞。

4) 时间和状态（Time and State）

分布式计算与时间和状态有关。线程和进程之间的交互及执行任务的时间顺序往往由共享的状态决定，如信号量、变量、文件系统等。与分布式计算相关的漏洞包括竞态条件、阻塞误用等。

5) 错误和异常处理缺陷（Errors）

这类漏洞与错误和异常处理有关，最常见的一种漏洞是没有恰当的处理错误（或者没有处理错误）从而导致程序运行意外终止，另一种漏洞是产生的错误给潜在的攻击者提供了过多信息。

6) 代码质量问题（Code Quality）

低劣的代码质量会导致不可预测的行为。对于攻击者而言，低劣的代码使他们可以以意想不到的方式威胁系统。常见的该类别漏洞包括死代码、空指针解引用、资源泄漏等。

7) 封装和隐藏缺陷（Encapsulation）

合理的封装意味着区分校验过和未经检验的数据，区分不同用户的数据，或区分用户能看到和不能看到的数据等。常见的漏洞包括隐藏域、信息泄漏、跨站请求伪造等。

8) 代码运行环境的缺陷（Environment）

该类漏洞是源代码之外的问题，例如运行环境配置问题、敏感信息管理问题等，它们对产品的安全仍然是至关重要的。

前七类漏洞与源代码中的安全缺陷相关，它们可以成为恶意攻击的目标，一旦被利用会造成信息泄露、权限提升等严重后果。最后一类漏洞描述实际代码之外的安全问题，它们容易造成软件的运行异常、数据丢失等严重问题。

3.2 安全漏洞级别

我们将源代码的安全问题分为四种级别：极高危（Critical）、高危（High）、中等（Medium）和低（Low）。衡量级别的标准包括两个维度，可信程度（confidence）和严重程度（severity）。可信程度是指发现的问题是否准确的可能性，比如将每个strcpy()调用都标记成缓冲区溢出漏洞的可信程度很低。严重程度是指假设测试技术真实可信的情况下检出问题的严重性，比如缓冲区溢出（buffer overflow）通常是比空

指针引用（null pointer dereference）更严重的安全问题。将这两个因素综合起来可以准确的为安全问题划分级别，如图 1 所示。

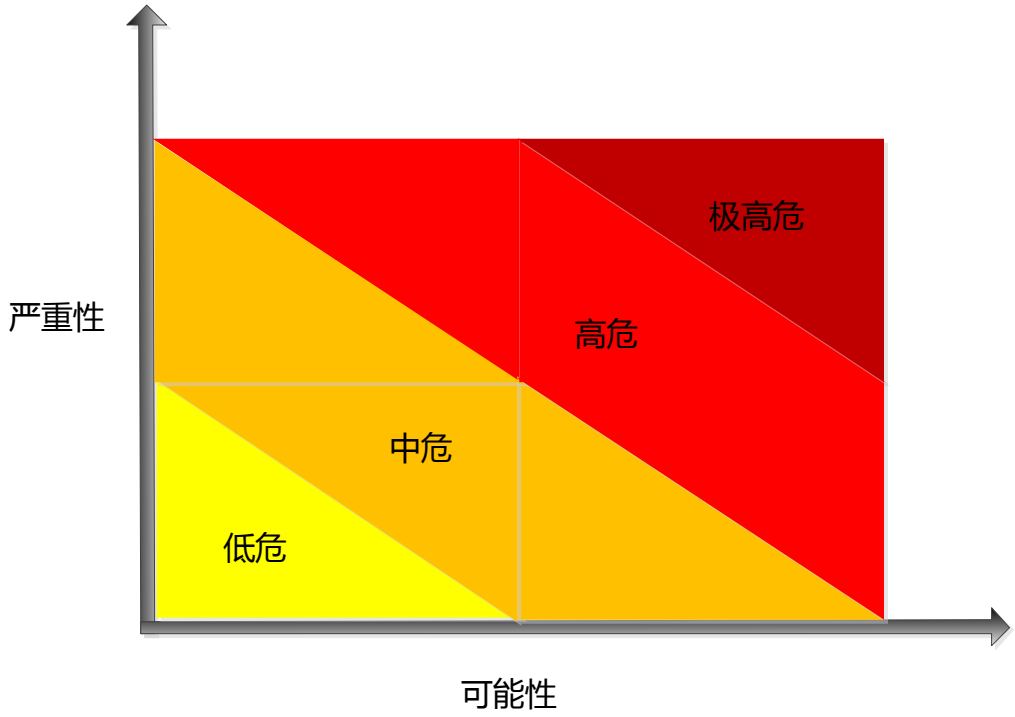

图 1 漏洞级别与严重程度、可信程度的关系

4 开源软件项目的安全漏洞情况

本部分首先展示从被测项目中检出安全漏洞的数量，由此对被测项目的安全性进行大致的评估。然后进一步讨论被测项目中安全漏洞的分布情况，了解项目中出现较多的、容易被忽略的安全问题。

4.1 安全漏洞情况概览

由于极高危、高危级别的安全漏洞危害程度较高，更能反映出软件中迫切需要解决的安全问题，本部分展示被测项目中这两种级别漏洞的检出情况，由此对被测项目的安全性进行大致的评估。图 2 展示了被测项目中检出的高危以上安全漏洞情况，并对项目

按照漏洞数量进行了排序，图中还用蓝色折线图展示了每千行包含漏洞数[2]，用红色折线图标出了项目的 Github 上 star 的数量。

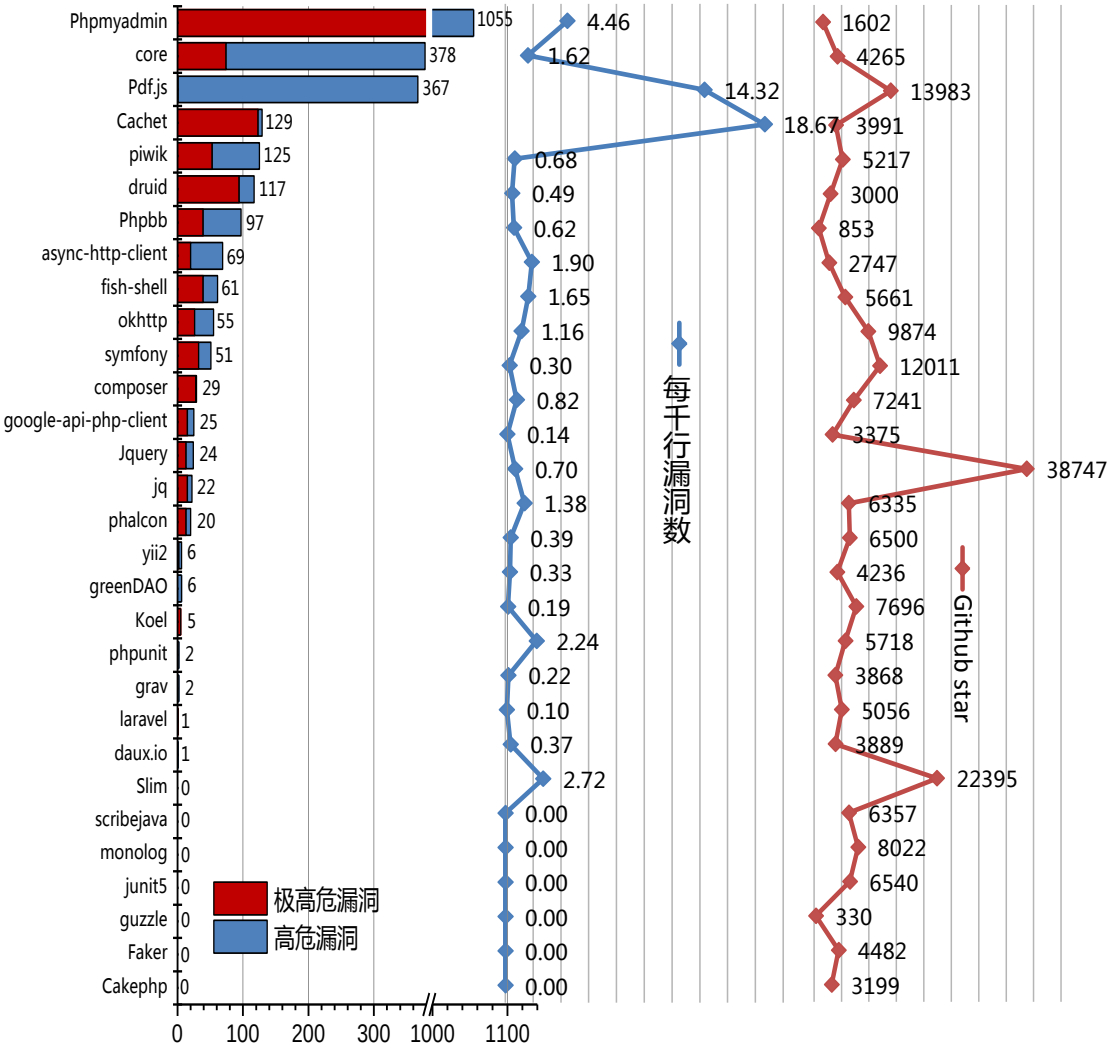

图 2 开源软件项目中高危以上漏洞分析图

从中可以看出，大多数软件项目都存在不同程度的安全问题。本次检测从这些项目中总计发现极高危漏洞 1334 个，高危漏洞 1313 个。漏洞数量排名靠前的项目处于极易被攻击者利用的状态，实际使用者急需通过安装补丁或者更新版本的方式进行修复和升级。

[2]每千行漏洞数计算方式：漏洞总数/代码行数*1000，精确到小数点后两位

结合 Github 上 star 数量来看，本次被检测软件中共有 6 款软件的 star 高于 8000，其中 4 款软件被检出 20 个以上的高危及以上漏洞 漏洞数量位于前 50% ,分别是 Pdf.js（高危以上漏洞数 367，star 数为 13983）、okhttp（高危以上漏洞数 55，star 数为 9874）、symfony（高危以上漏洞数 51，star 数为 12011）及 Jquery（高危以上漏洞数 24，star 数为 38747）。除 Pdf.js 为办公类应用外，另外三款开源软件都是开发框架类软件。由于这些软件使用极其广泛，其存在的安全风险可能造成极大危害，使用这些开源软件的用户及开发者，应警惕由于使用开源软件而可能造成的安全隐患。

在所有被测软件中，漏洞总数最多的是 MySQL 数据库管理工具 Phpmyadmin，包含漏洞高达 1055 个，远远高于漏洞总数排名第二的软件。其中极高危漏洞 741 个，高危漏洞 314 个，其每千行漏洞数 4.46，在全部被测软件中排名第三。由于极高危漏洞数量巨大，使用该软件进行开发可能存在较高安全风险。漏洞总数排名第二的是 PHP 开发框架 core，包含漏洞 378 个，其中极高危漏洞 74 个，高危漏洞 304 个，每千行漏洞数 1.62，也属于较高水平。漏洞总数排名第三的软件是 JavaScript 语言开发的 pdf 解析工具 Pdf.js，包含漏洞 367 个，其中极高危漏洞 1 个，高危漏洞 366 个，每千行漏洞数 14.32，在所有被测软件中排名第二，由于其 Github 热度极高，使用极其广泛，因此该软件漏洞影响范围较大。

每千行漏洞数最高的被测软件是开源的状态页面系统 Cachet，其每千行漏洞数高达 18.67。全部被测软件中，只有 Cachet 与 Pdf.js 这两款每千行漏洞数高于 10，远远高于排名第三的 Phpmyadmin（每千行漏洞数 4.46）。

本次被测软件中，有 18 款软件为开发框架，如 Jquery、laravel、symfony 等，共检出高危以上漏洞 873 个，其每千行漏洞数平均为 0.69，远低于全部 30 款被测软件

的每千行漏洞数平均值（1.41）。需要注意的是，开发工具存在的安全漏洞，将使在此基础上开发出软件的安全性和质量大打折扣。9 款软件为测试及部署类工具，其中 3 款软件高危以上漏洞数排在前五位，分别是 Phpmyadmin、Cachet 及 piwik，应引起高度重视。测试及部署类工具共检出高危以上漏洞 1719 个，其每千行漏洞数平均为 2.20，远高于全部被测软件的每千行漏洞数平均值（1.41）。3 款软件为办公及其他 WEB 应用类软件，分别是 Pdf.js、fish-shell 及 Koel，其中 Pdf.js 漏洞总数及每千行漏洞数都很高，安全问题比较严重，另外两款软件安全性在全部被测软件中处于中等水平。

本次测试的全部 30 款软件中，有 7 款没有包含任何高危以上漏洞，分别是 Web 应用程序开发框架 Cakephp、PHP 生成数据工具 Faker、PHP HTTP 客户端 guzzle、Java 语言的单元测试框架 junit5、日志记录组件 monolog、简单的 JAVA 版 OAUTH 库 scribejava 和 PHP 框架 Slim。此外，项目文档生成器 daux.io 及 PHP Web 开发框架 laravel 高危以上漏洞数也仅为 1，总体安全性也很好。

4.2 高危安全漏洞分布情况

此次测试中发现的高危漏洞不仅数量众多，覆盖的种类也较为繁杂。图 3 展示了被测项目中高危以上级别漏洞大类的分布情况。数据表明，大多数漏洞为“输入验证与表示”类漏洞，该类漏洞易被攻击者利用，通过绕过用户输入验证从而对 Web 应用系统进行攻击，产生严重危害。“安全特性”类漏洞也占据了较大比例，攻击者可利用该类漏洞破解加密算法，导致用户隐私信息泄露等严重安全问题。此外，“代码运行环境的缺陷”类漏洞也出现较多，产生的主要原因是系统运行环境配置及隐私数据的存储配置不当等，攻击者可以利用运行环境漏洞对系统发起攻击及窃取用户隐私。

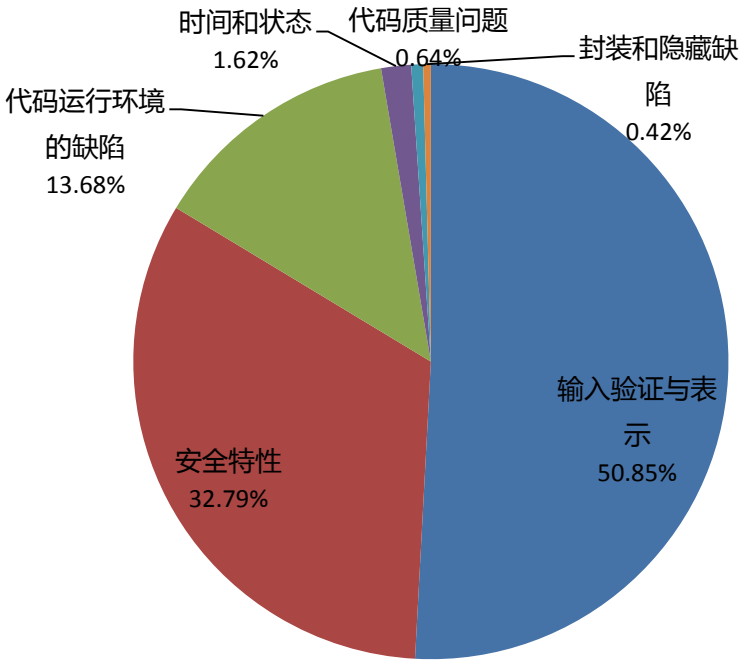

图 3 被测项目中高危以上安全漏洞的分布情况（按大类划分）

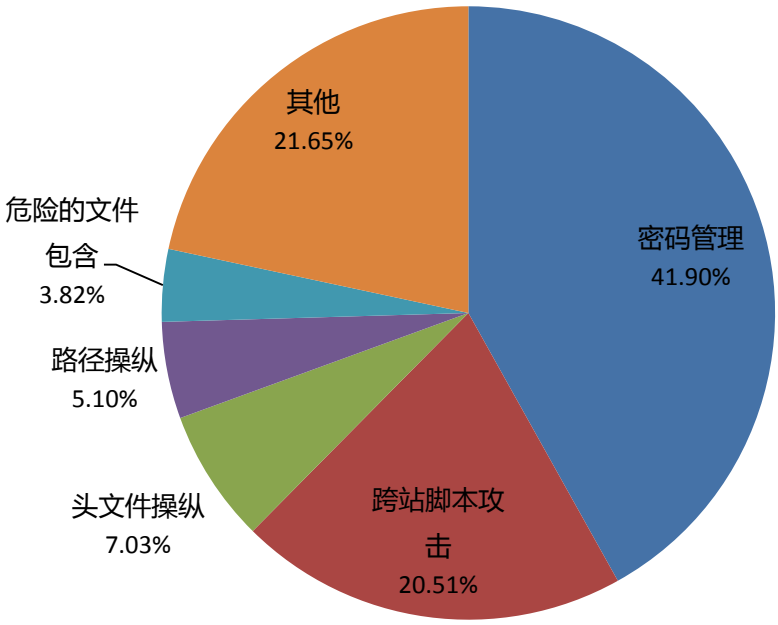

图 4 被测项目中高危以上安全漏洞的分布情况（按具体漏洞划分）

图 4 进一步展示了被测项目中的各种具体的高危以上级别安全漏洞的分布情况。在被测的 30 个项目中，出现次数超过 100 次的高危以上漏洞共有 5 种。可以看出，出现最多的前 5 种漏洞依次是：密码管理（41.90%，1109 个）、跨站脚本攻击（20.51%，

543 个）、头文件操纵（7.03%，186 个）、路径操纵（5.10%，135 个）、危险的文件包含（3.82%，101 个）。下面对这 5 种漏洞进行简要说明，并给出防范建议。

1) 密码管理（属于安全特性缺陷）

危害：程序中可能存在硬编码、弱加密等不安全的密码处理方式，容易导致密码泄露、口令被破解等安全问题。

防范：避免代码中存在硬编码，并使用安全的加密方式对密码进行加密处理，保存在数据库或配置文件中。

2) 跨站脚本攻击（属于输入验证与表示类漏洞）

危害：向一个 Web 浏览器发送未经验证的数据会导致该浏览器执行恶意代码。

防范：验证所有输入数据，有效检测攻击；对所有输出数据进行适当的编码，以防止任何已成功注入的脚本在浏览器端运行。

3) 头文件操纵（属于输入验证与表示类漏洞）

危害：HTTP 响应头文件中包含未验证的数据会引发跨站脚本攻击、页面劫持等问题。

防范：禁止 HTTP 相应头文件中包含换行符。

4) 路径操纵（属于输入验证与表示类漏洞）

危害：允许用户输入访问文件系统的路径，可以使攻击者访问或修改保护的系统资源。

防范：验证所有输入数据，避免用户能够通过操纵路径访问或修改系统资源。

5) 危险的文件包含（属于输入验证与表示类漏洞）

危害：许多现代网络编写语言都能够在一个封装的文件内包含附加的源文件，从而使代码可以重用和模块化。这种能力经常用于赋予应用程序标准外观（应用模板），因而，人们可以共享各种功能而不需要借助编译的代码，或将代码分解成较小的更好管理的文件。各个包含文件都会作为主文件的一部分进行解析，并采用相同的方式来执行。当未验证的用户输入控制了所包含文件的路径时，就会发生危险的文件包含漏洞。

防范：验证所有输入数据，避免用户控制被包含文件的路径。

4.3 安全漏洞总体分布情况

4.1 和 4.2 节针对被测项目中的高危以上漏洞的检出情况对项目的安全状况进行了分析。通常来说，与高危漏洞相比，中低危漏洞在实际运行环境中的危害相对较小，但仍能在一定程度上反映出项目的代码质量、开发人员对代码安全问题的重视程度等。为了更全面的了解被测项目的安全状况，本节进一步展示包括中低危漏洞在内的所有级别安全漏洞的总体分布情况。

图 5 展示了被测项目中安全漏洞大类的分布情况。与高危以上级别的漏洞分布情况相比，“错误和异常处理”、“封装和隐藏缺陷”比重明显增加。这两种类型的漏洞相对来说威胁较低，容易被开发人员忽视；这些漏洞虽然不易直接产生重大危害，但可能导致系统运行不稳定、系统重要信息泄露等安全隐患，一旦被攻击者利用也会造成严重后果。

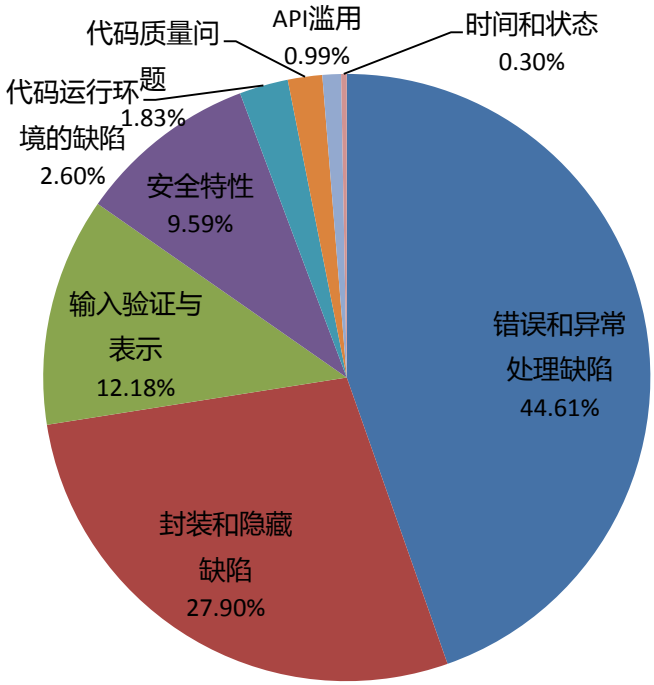

图 5 被测项目中的全部安全漏洞的分布情况（按大类划分）

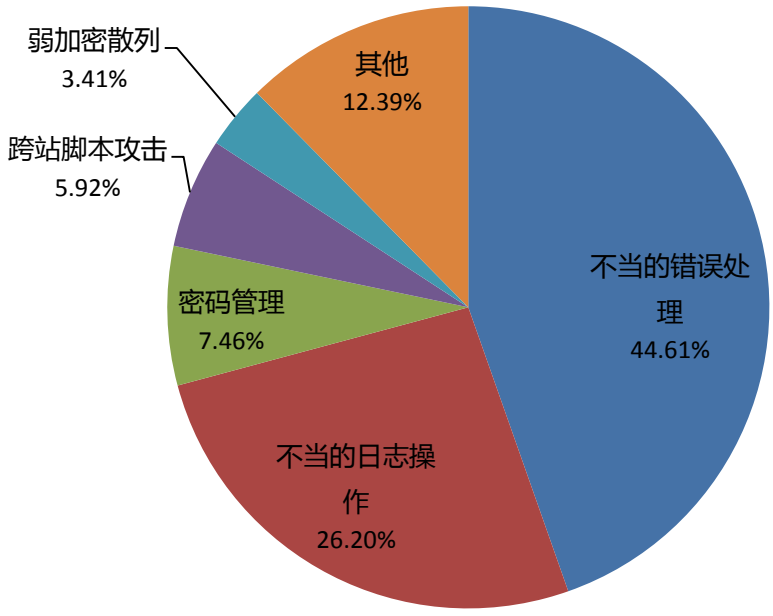

图 6 被测项目中的全部安全漏洞的分布情况（按具体安全漏洞划分）

图 6 进一步展示了被测项目中的各种具体的安全漏洞的分布情况。在被测的 30 个项目中，出现次数超过 500 次的漏洞共有 5 种。出现最多的 5 种漏洞依次是：不当的错误处理（44.61%，6767 个）、不当的日志操作（26.20%，3975 个）、密码管理（7.46%，

1132 个）、跨站脚本攻击（5.92%，898 个）、弱加密散列（3.41%，518 个）。下面对这排名前 5 种漏洞进行简要说明，并给出防范建议。

1）不当的错误处理（属于错误和异常处理缺陷）

危害：对于提供了错误或异常处理 API 的程序语言，如果对于错误不做处理或处理不当，会造成输出大量错误信息、程序崩溃等后果。

防范：遵守错误处理规范，错误处理要全面，尤其对于处理过程中出现的新错误也要进行嵌套式处理。

2）不当的日志操作（属于封装和隐藏缺陷）

危害：系统可能没有使用专门的日志记录工具进行日志处理，会导致难以监控程序的运行状况。

防范：使用专门的日志记录工具，方便随时监控程序的运行状况。

3）密码管理（属于安全特性缺陷）

危害：程序中可能存在硬编码、弱加密等不安全的密码处理方式，容易导致密码泄露、口令被破解等安全问题。

防范：避免代码中存在硬编码，并使用安全的加密方式对密码进行加密处理，保存在数据库或配置文件中。

4）跨站脚本攻击（属于输入验证与表示类漏洞）

危害：向一个 Web 浏览器发送未经验证的数据会导致该浏览器执行恶意代码。

防范：验证所有输入数据，有效检测攻击；对所有输出数据进行适当的编码，以防止任何已成功注入的脚本在浏览器端运行。

5）弱加密散列（属于安全特性缺陷）

危害：弱加密散列值无法保证数据完整性，且不能在安全性关键的上下文中使用。

防范：避免使用存在安全风险的散列算法，或只用加盐方式增加破解难度。

5 新旧版本软件安全性对比

上季度检测的开源软件中，有 12 款软件进行了版本升级。本部分对更新前后的软件安全性进行对比分析。图 7 展示了这 12 款软件新旧版本的每千行漏洞数情况。

从对比情况可以看出，版本更新不完全意味着安全性的提升。六款软件在更新后安全性有所降低，其中 Pdf.js 和 Jquery 这两款软件的安全性有明显下降。Pdf.js 在版本更新前每千行漏洞数仅为 0.15，版本更新后增加到高达 14.32，使用该版本软件可能带来较多安全风险。Jquery 在版本更新前未检测出任何漏洞，版本更新后，每千行漏洞数增加到 0.07，安全性能较之前有所下降。

版本更新后，有四款软件安全性得到提升，分别是 laravel，Phpbb，Koel 和 Phpmyadmin，特别是 laravel 的安全性有明显提升，因此建议使用这几款软件的用户尽快升级到最新版本。

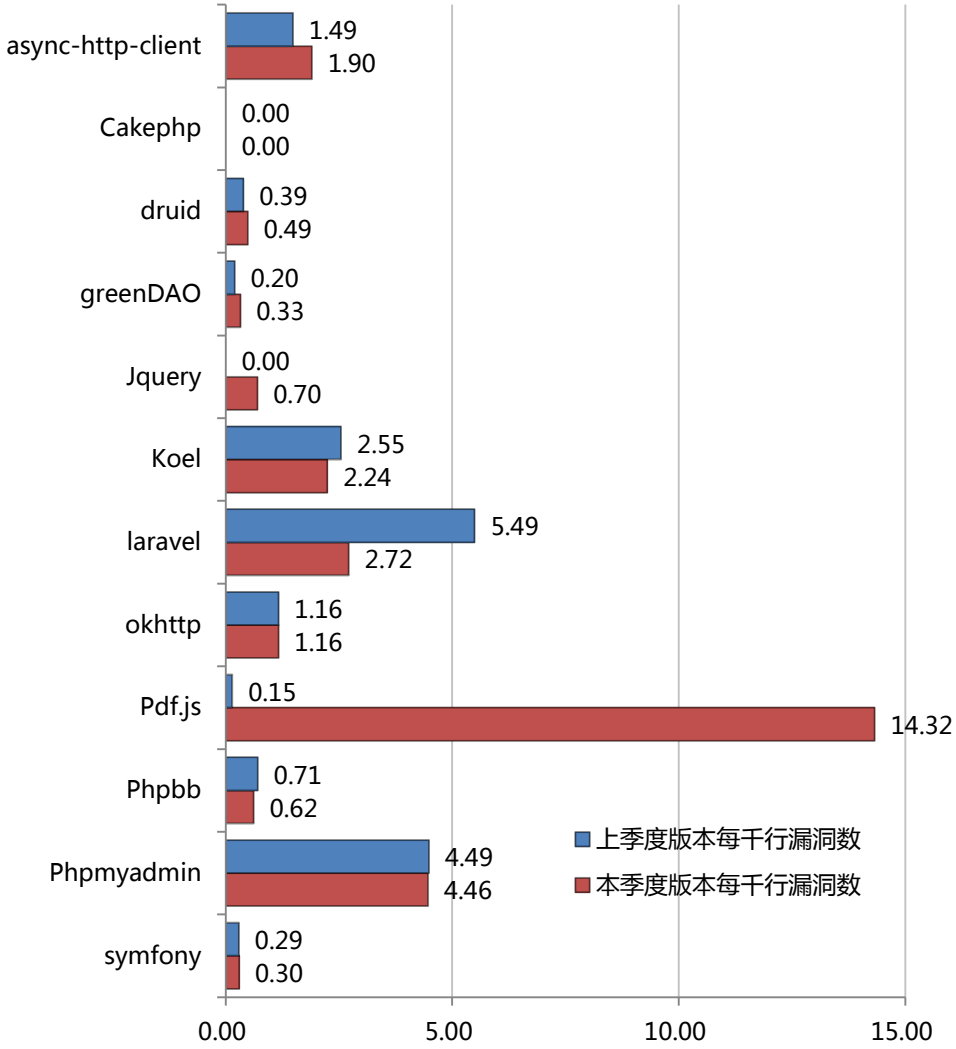

图 7 新旧版本漏洞密度对比分析

6 关于本报告的说明

一、本报告仅从代码角度进行漏洞分析。在实际系统中，由于软件实际部署环境、安全设备等的限制，部分漏洞可能无法通过渗透测试得到验证。

二、本报告中的漏洞仅适用于表 1 中列出的特定软件版本。当软件版本有任何更新、修改和优化时，本报告不再适用。