公 开

2015 年第四季度开源软件源代码
安全漏洞分析报告

国家计算机网络与信息安全管理中心

实验室

2016 年 1 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 4348 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 JAVA，PHP，C++，JavaScript 四种编程语言。这些软件项目都是国际、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。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。

表 1 被测开源软件项目概览

项目名称	版本号	主要编程语言	功能说明	代码行数(L)
async-http-client	1.9.32	JAVA	异步 HTTP 请求处理框架	46366
bitcoin	0.11.2	C++	比特币在线交易系统	37813
Cakephp	3.2	PHP	Web 应用程序开发框架	26288
clojure	1.8.0	JAVA	运行在 Java 平台上的动态函数式编程语言	39993
CodeIgniter	3.1.6	PHP	Web 应用程序开发框架	28877
druid	1.0.16	JAVA	数据库访问组件	297400

EventBus	2.4.0	JAVA	Android 组件间通信库	3828
Graylog2	1.3.3	JAVA	日志管理平台	115652
greenDAO	1.3.7	JAVA	移动开发的 ORM 框架	14988
jedis	2.8.0	JAVA	Redis 的 Java 客户端开发包	30701
Jquery	2.2.0	JavaScript	轻量级的 javascript 开发库	4028
jsoup	1.8.3	JAVA	HTML 解析器	14453
junit	4.12	JAVA	Java 语言的单元测试框架	25813
Koel	1.1	PHP	基于 Web 的个人音乐流媒体应用	1962
laravel	5.2.0	PHP	PHP Web 开发框架	364
Okhttp	3.0.1	JAVA	HTTP+SPDY 客户端开发包	48069
Paperwork	3751d936	PHP	开源网络云笔记系统	14565
Pdf.js	1.3.88	JavaScript	在线 PDF 阅读框架	13770
Phabricator	1	PHP	可视化代码审查工具	351536
Phpbb	3.1.x	PHP	网络论坛框架	151065
PHPMailer	5.2.14	PHP	电子邮件框架	6320
Phpmyadmin	4.5.3.1	PHP	MySQL 数据库管理工具	237401
Picasso	2.5.2	JAVA	Android 图形缓存库	8674
redis	3.2	C++	数据结构存储系统	38691
retrofit	1.9.0	JAVA	Android 平台下的 REST 客户端	10297
Storm	0.10.0	JAVA	流式大数据处理框架	95558
symfony	3	PHP	基于 PHP 的 Web 开发框架	172536
TextSecure-Server	0.54	JAVA	Android 下的加密聊天工具服务器端	13278
vert	2.1.6	JAVA	高性能 JVM 应用平台	43730
Wordpress	4.5	PHP	内容管理系统	169529

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。

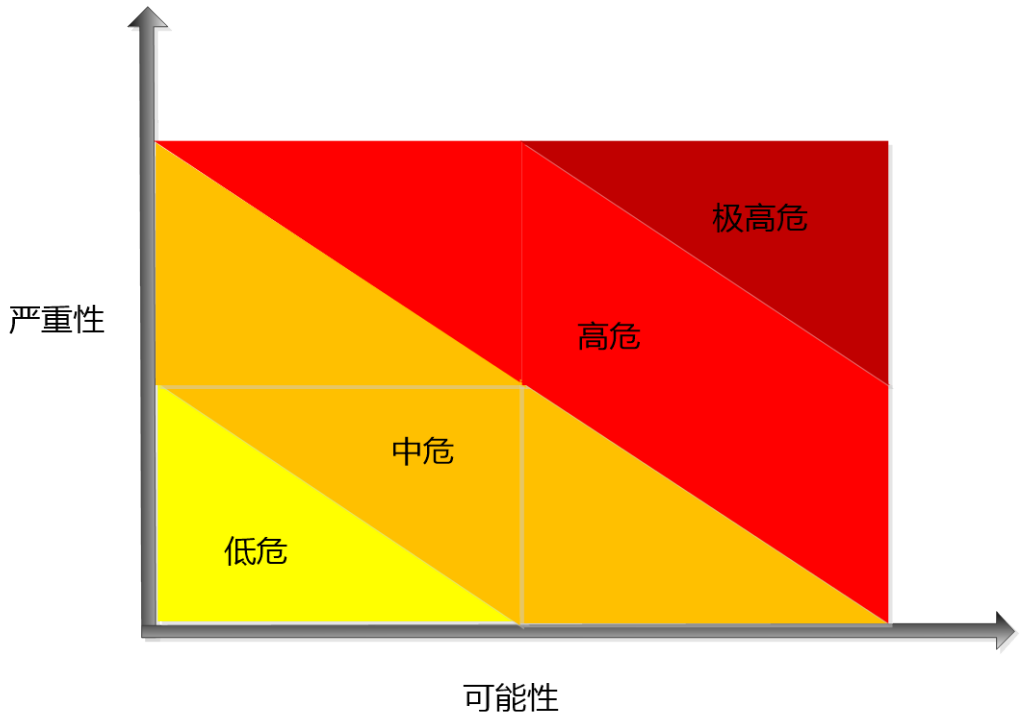

图 1 漏洞级别与严重程度、可信程度的关系

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。表 2 展示了被测项目中检出的极高危以及高危安全漏洞的情况。

图 2 用条状图进行了直观的展示，并按照漏洞数量对项目进行了排序。图 3 展示了各款软件每千行代码包含漏洞数。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。这些项目中总计发现极高危漏洞 2821 个，高危漏洞 1527 个。

依据此次代码测试结果，PHP 语言开发的软件安全问题最为严重。安全问题总数排名前五名的软件中有四款为 PHP 语言开发，每千行漏洞数前五名的软件中也有四款为 PHP 语言开发，可见相对于其他语言，PHP 语言开发者安全意识相对较弱，而该语言使用极其普遍，更应引起高度重视。

漏洞总数排名第一的是著名的 PHP 内容管理系统 Wordpress（4.5 版本），包含漏洞高达 1889 个，其中极高危漏洞 1459 个，高危漏洞 430 个，同时它也是本次被测项目中漏洞密度最高的软件，平均每一千行代码就有高达 11.14 个漏洞，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。而另一款 PHP 开源软件 Phpmyadmin（4.5.3.1 版本）以总漏洞数 1065 个在漏洞总数上排名第二，其中极高危漏洞 746 个，高危漏洞 319 个，其每千行漏洞数 4.49，在全部软件中排名第五，存在较高的安全风险。漏洞总数排名第三的是 C++数据结构存储系统 redis（3.2 版本），包含漏洞 396 个，其中极高危漏洞 208 个，高危漏洞 188 个，其漏洞密度仅低于 Wordpress，平均每一千行代码包含漏洞 10.23 个。

表 2. 开源软件项目中安全漏洞概览

项目名称	**极高危(C)**	**高危(H)**	**漏洞总数(H+C)**	**每千行漏洞数（漏洞总数/代码行数*1000，精确到小数点后两位）**
async-http-client	20	49	69	1.49
Bitcoin	1	0	1	0.03
Cakephp	0	0	0	0.00

Clojure	0	7	7	0.18
CodeIgniter	31	22	53	1.84
Druid	94	22	116	0.39
EventBus	0	6	6	1.57
Graylog2	19	87	106	0.92
greenDAO	0	3	3	0.20
Jedis	2	1	3	0.10
Jquery	0	0	0	0.00
Jsoup	2	1	3	0.21
Junit	0	34	34	1.32
Koel	4	1	5	2.55
Laravel	2	0	2	5.49
Okhttp	27	29	56	1.16
Paperwork	49	89	138	9.47
Pdf.js	0	2	2	0.15
Phabricator	12	121	133	0.38
Phpbb	46	61	107	0.71
PHPMailer	22	4	26	4.11
Phpmyadmin	746	319	1065	4.49
Picasso	2	4	6	0.69
Redis	208	188	396	10.23
Retrofit	0	1	1	0.10
Storm	2	18	20	0.21
symphony	31	19	50	0.29
TextSecure-Server	41	8	49	3.69

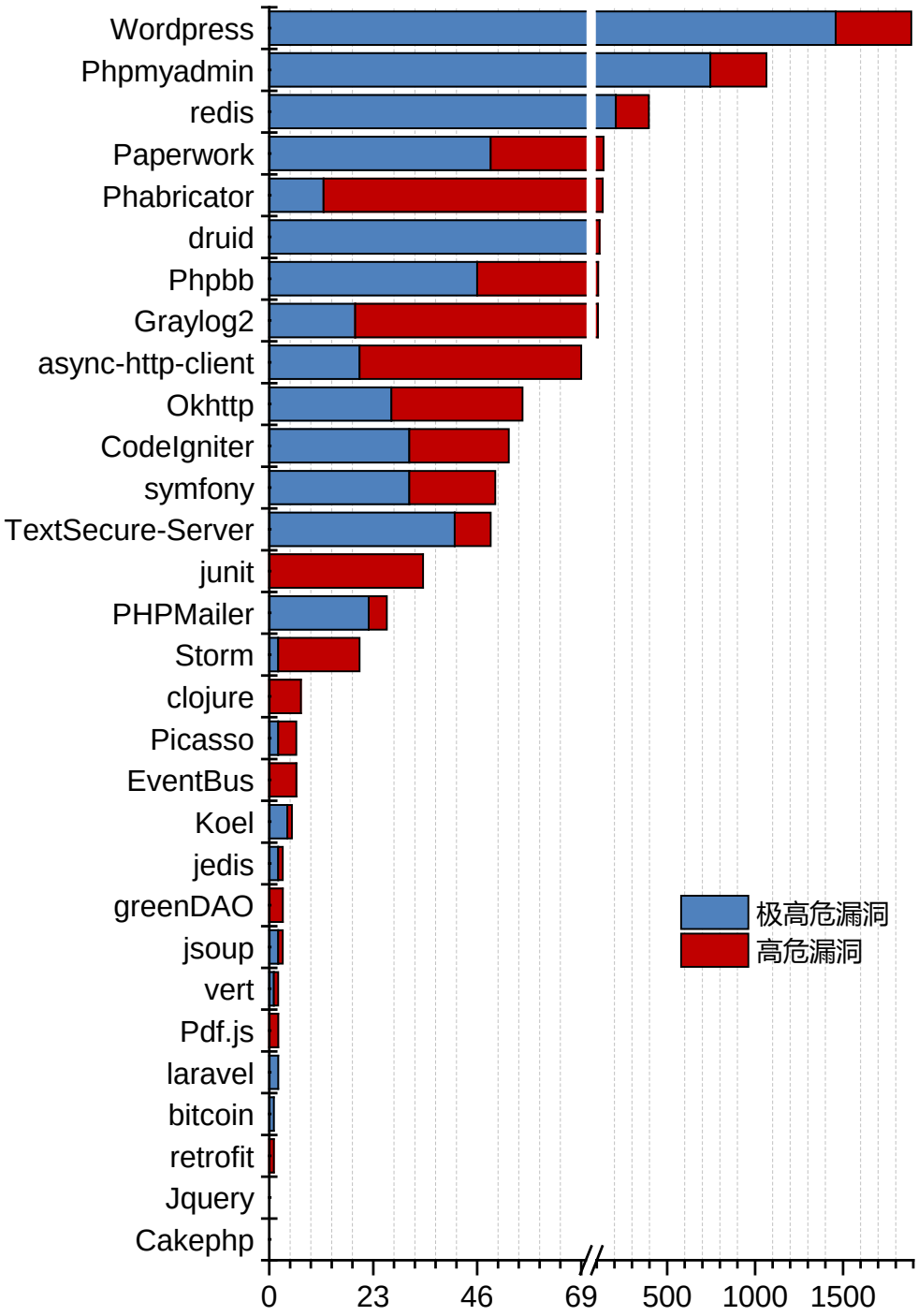

图 2 开源软件项目中高危以上漏洞分析图

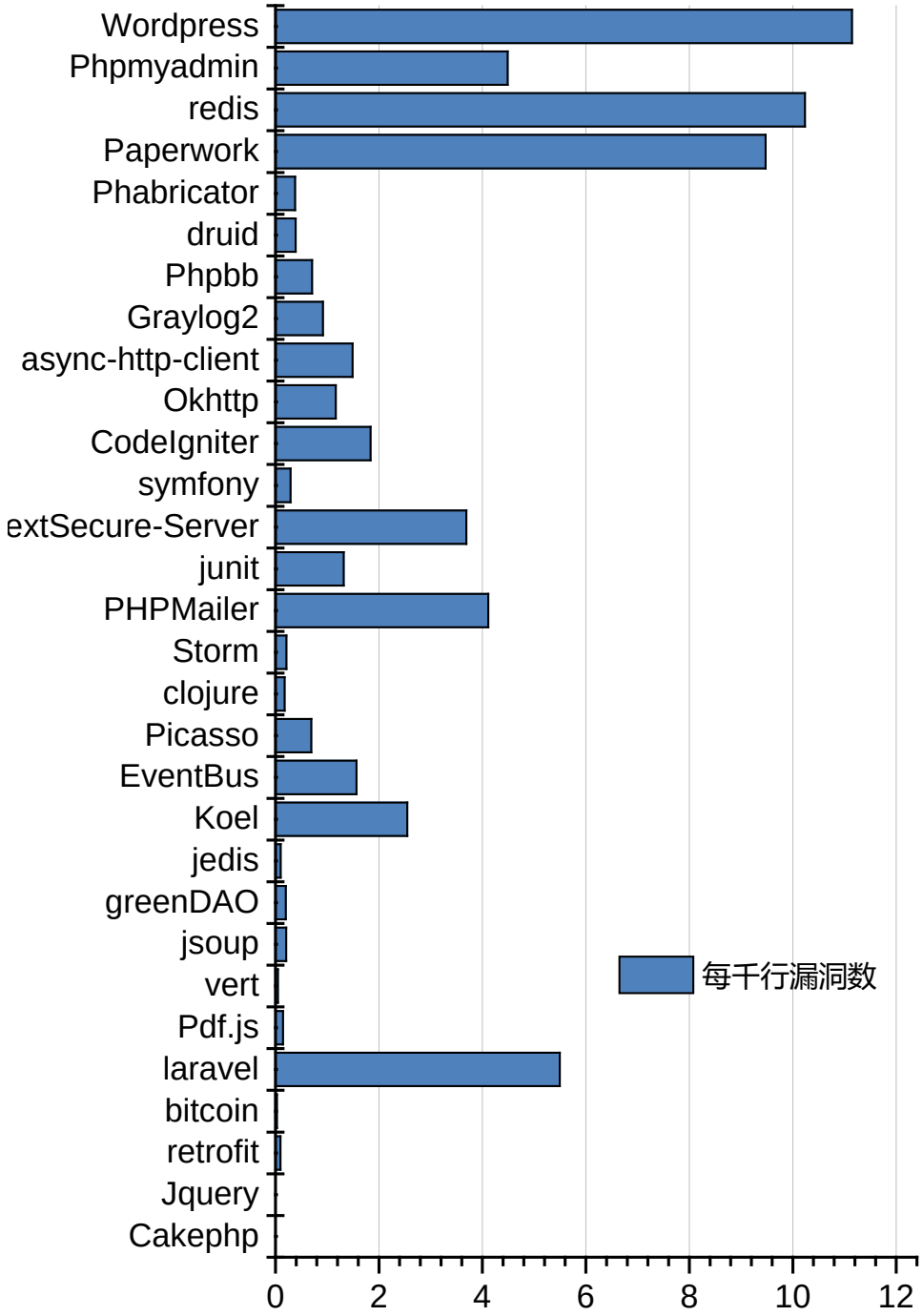

图 3 开源软件项目中每千行漏洞数分析图

JAVA 语言中，数据库访问组件 druid（1.0.16 版本）漏洞数为 116 个，极高危漏洞为 94 个，高危漏洞为 22 个，在本次测试的 JAVA 开源软件中，漏洞总数排名第一；但由于其代码量较大，每千行漏洞数仅为 0.39，在所有被测软件中属于较低水平。

在本次测试中,安全情况最好的软件是 PHP Web 应用程序开发框架 Cakephp (3.2 版本）及轻量级的 javascript 开发库 Jquery（2.2.0），这两款软件中不包含任何高危以上漏洞，安全性非常好。此外，比特币在线交易系统 bitcoin（0.11.2）及 Android 平台下的 REST 客户端 retrofit(1.9.0)高危以上漏洞数也仅为 1 ,总体安全性也很好。

这项统计充分说明了这批项目的安全性情况，漏洞数量排名靠前的项目处于极其容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。

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 4 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，“代码质量问题”类漏洞也出现较多，产生的主要原因是开发人员安全意识不足，此类漏洞会导致内存溢出、资源耗尽等安全隐患，严重情况下会导致系统运行异常、甚至系统崩溃。

图 5 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 200 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：跨站脚本攻击（41.81%，1818 个）、密码管理（10.92%，475 个）、缓冲区溢出（8.10%，352 个）、头文件操纵（6.90，300 个）、日志伪造（5.66%，246 个）。下面对这 5 种漏洞进行简要说明，并给出防范建议。

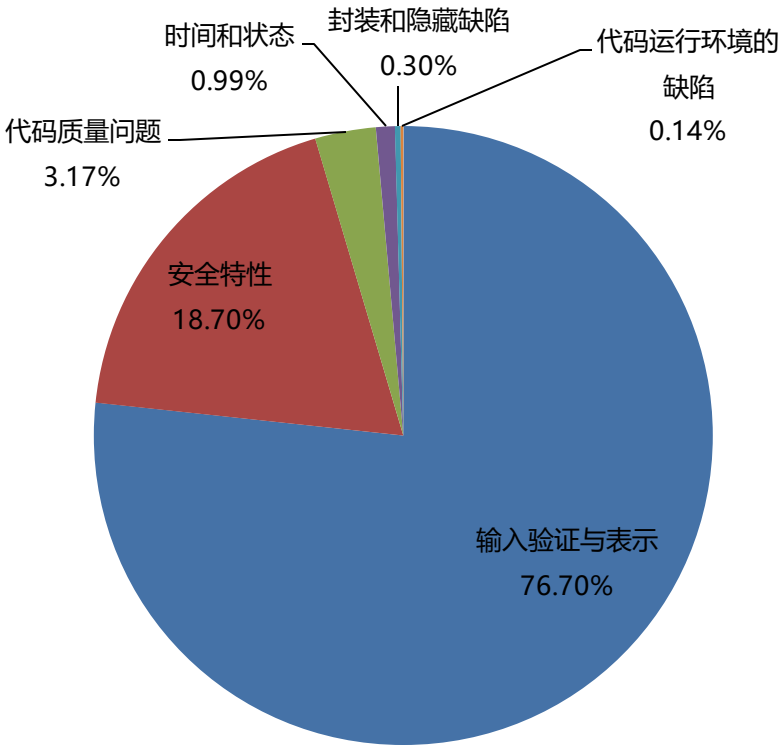

图 4 被测项目中高危以上安全漏洞的分布情况（按大类划分）

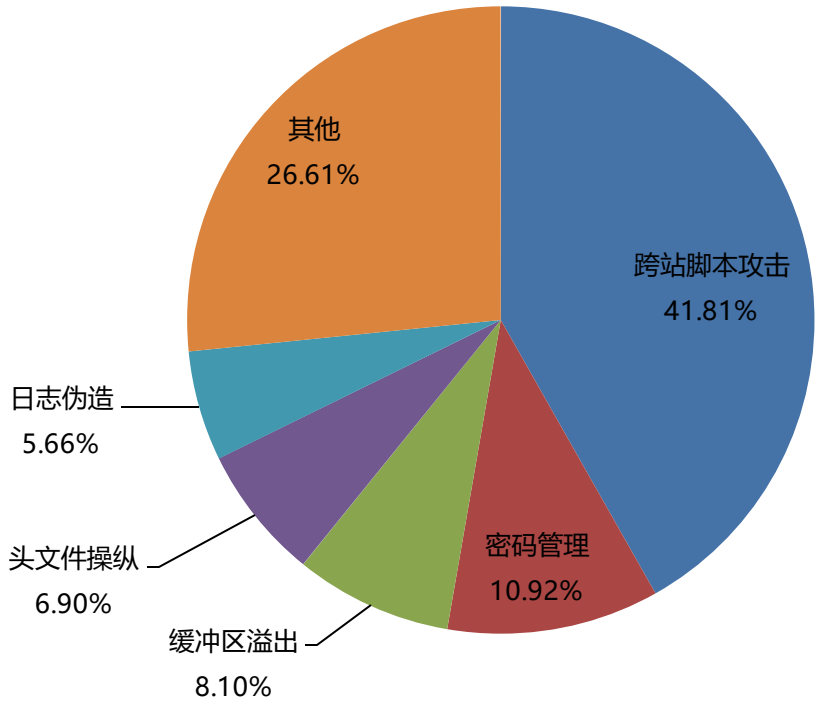

图 5 被测项目中高危以上安全漏洞的分布情况（按具体漏洞划分）

1) 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。

2) 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。

3) 缓冲区溢出（属于输入验证与表示类漏洞）

危害：在一块分配的内存边界之外写入数据可能会破坏数据、造成程序崩溃或导致恶意代码的执行。

防范：针对内存处理函数执行边界检查，并尽量避免使用依靠外部的数据来控制行为的代码。

4) 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。

5) 日志伪造（属于输入验证与表示类漏洞）

危害：将未经验证的用户输入写入日志文件可致使攻击者伪造日志条目或将恶意信息内容注入日志。

防范：禁止用户可以控制写入日志的信息。

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 6 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”和“代码质量问题”比重明显增加。这三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 7 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 6 种。出现最多的 5 种漏洞依次是：不当的错误处理（43.15%，9211 个），跨站脚本攻击（14.39%，3073 个），不当的日志操作（14.09%，3007 个），不良的编码方式（7.08%，1511 个），代码正确性（3%，641 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。

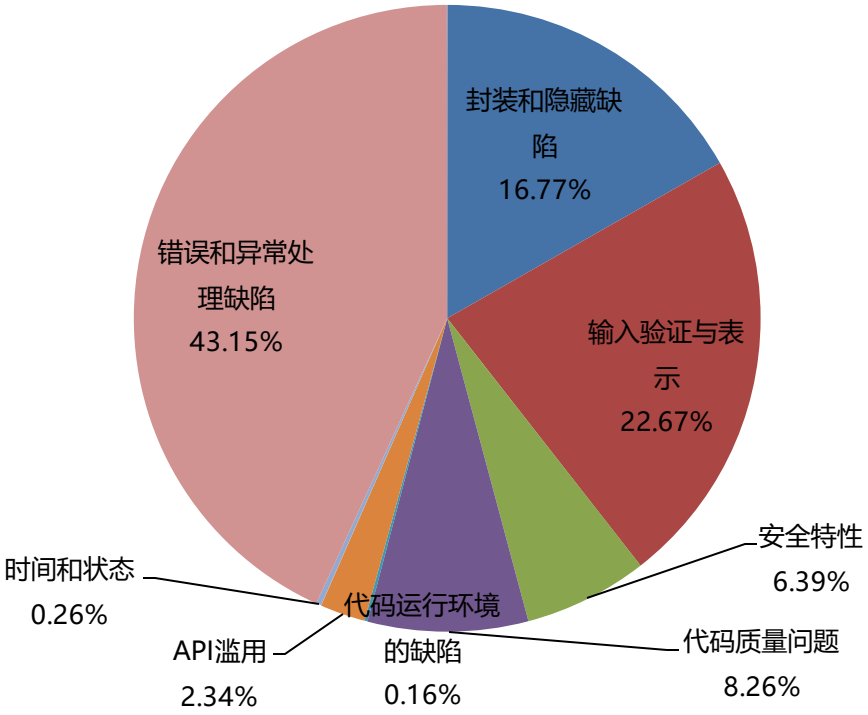

图 6 被测项目中的全部安全漏洞的分布情况（按大类划分）

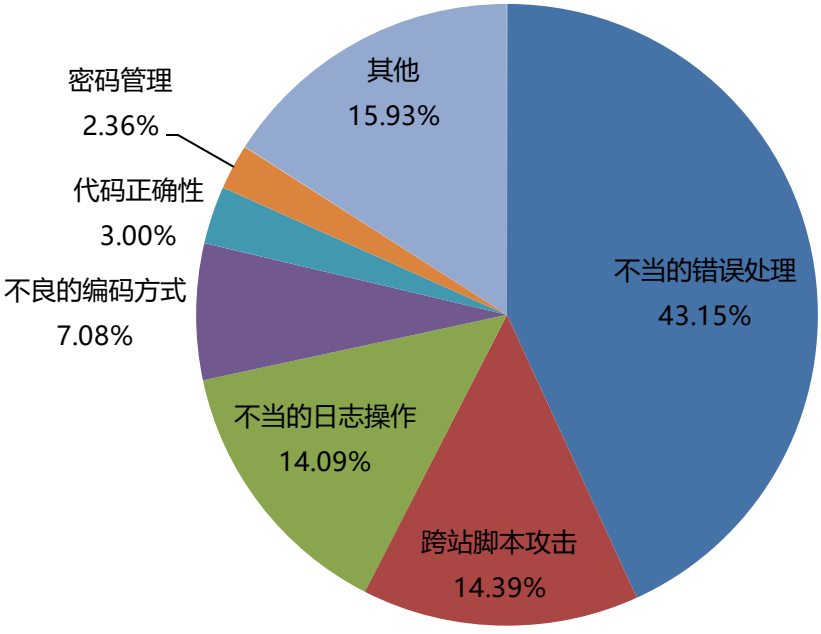

图 7 被测项目中的全部安全漏洞的分布情况（按具体安全漏洞划分）

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）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）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。

4）不良的编码方式（属于代码质量缺陷）

危害：代码中可能存在不好的编码方式（如变量赋值后并不使用，而变成死存储；一个代码块中不包含任何指令等），除了造成存储浪费以外，很可能是代码逻辑出现错误。

防范：检查代码编写是否存在错误；如果逻辑上不存在错误，建议删除冗余代码并修改代码样式，提高可读性。

5）代码正确性（属于代码质量缺陷）

危害：错误代码会导致不可预测的行为。对于攻击者而言，错误代码使他们可以通过意想不到的方式威胁系统。

防范：提高代码质量，降低代码问题对系统的影响。

5 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。