公 开

开源软件源代码安全缺陷分析报告

——大数据专题

国家互联网应急中心

实验室

2017 年 3 月

目录

1 概述

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。为了解开源软件的安全情况，实验室持续对广泛使用的知名开源软件进行源代码安全缺陷分析，并发布季度安全缺陷分析报告。

在当今这个信息爆炸的时代，大数据已经逐渐从前沿技术转变为众多企业的必备解决方案。随着大数据和预测分析技术的成熟，从初创企业到行业巨头，各种规模的供应商都在借助开源软件处理大数据和运行预测分析。实验室本季度聚焦 20 款大数据领域的知名开源软件。结合缺陷扫描工具和人工审计的结果，形成了本缺陷分析报告。本次检测在代码层面发现高危安全隐患共 133 个。从结果来看，开源大数据软件存在着不容忽视的安全风险。

2 被测开源软件

综合考虑用户数量、受关注程度以及更新频率等情况，实验室选取了 20 款具有代表性的大数据软件。表 1 列出了本次被测的开源大数据软件项目的概况。本次检测的软件涵盖了 C，Java，Python，PHP 等编程语言。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。由于这些软件大多具有巨大的用户群体，软件中的安全缺陷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
Drill	1.8.0	Java	低延迟的分布式海量数据交互式查询引擎	380,645

KosmosFS	0.3	Java	分布式文件存储系统	596,551
Cascading	3.1.2	Java	Hadoop 集群数据处理 API	2,699,212
Lucene	6.3.0	Java	全文检索引擎工具包	768,443
HaLoop	3.0.0-alpha1	Java	支持高效迭代、递归分析任务的 MapReduce 框架	2,375,727
Hama	0.7.1	Java	基于 BSP 分析模式的大数据分析框架	67,582
Splunk	6.5.1	Python	机器数据的智能分析、管理平台	247,843
Storm	1.1.0	Java	分布式流数据实时计算系统	246,191
Tokyo Cabinet	1.4.48	C	一个键值对的数据库管理系统	90,460
Alluxio	1.3.0	Java	一个基于内存的虚拟分布式存储系统	224,059
Apache Hive	2.1.0	Java	基于分布式存储管理大数据集的数据仓库软件	1,460,161
Solr	6.3.0	Java	基于 Lucene 的企业级数据搜索平台	1,399,459
HyperTable	0.9.8.11	Java+C	一个高性能、可伸缩的数据库，采用与 Google 的 Bigtable 相似的模型	408,498
Apache Tajo	0.11.3	Java	基于 Hadoop 的分布式数据仓库系统	296,998
Apache Spark	2.0.2	Java	分布式大规模数据处理引擎	122,582
ZooKeeper	3.5.2-alph	Java	一个分布式应用程序协调服务，是 Google 的 Chubby 的一个开源实现，是 Hadoop 和 Hbase 的重要组件	339,759
MariaDB	0.2	C+PHP	完全兼容 MySQL 的数据库管理系统	904,259
Openstack Swift	1.5.0	Python	提供弹性可伸缩、高可用的分布式对象存储服务	337,712
Mahout	0.12.2	Java	分布式机器学习算法开发框架	151,073
LevelDB	1.19	C	一款高效的键值对数据库	23,364

3 测试内容

3.1 安全缺陷种类

本次测试涵盖各类常见安全缺陷。根据缺陷形成的原因、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，可以将常见的安全缺陷分为八类：

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 误用（API Abuse)

API 是调用者与被调用者之间的一个约定，大多数的 API 误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缺陷。

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缺陷包括竞态条件、阻塞误用等。

5) 错误和异常处理缺陷（Errors）

这类缺陷与错误和异常处理有关，最常见的一种缺陷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，另一种缺陷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。常见的该类别缺陷包括死代码、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。常见的缺陷包括隐藏域、信息泄漏、跨站请求伪造等。

8) 代码运行环境的缺陷（Environment）

该类缺陷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。

前七类缺陷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、命令执行等严重后果。最后一类缺陷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 安全缺陷级别

我们将源代码的安全问题分为三种级别：高危（High）、中等（Medium）和低（Low）。衡量级别的标准包括两个维度，置信程度（confidence）和严重程度（severity）。置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缺陷的可信程度很低。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。

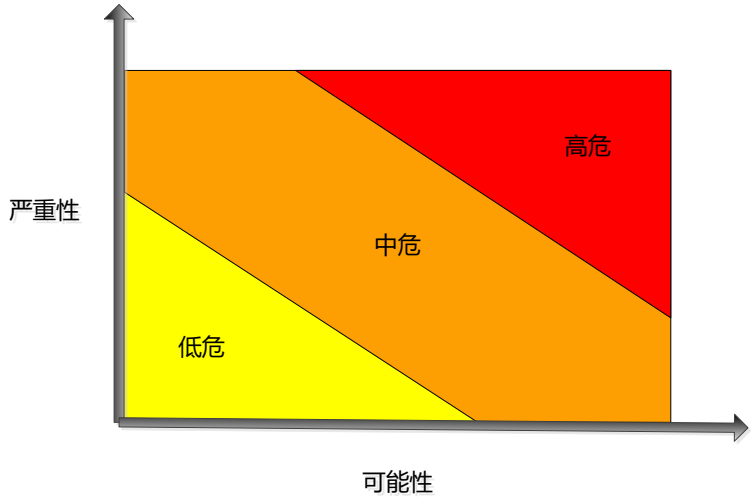

图 1 缺陷级别与严重程度、置信程度的关系

4 开源大数据软件项目的安全缺陷情况

本部分首先展示从被测项目中检出安全缺陷的数量，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缺陷的分布情况，了解项目中出现较多的、容易被忽略的安全问题。

4.1 安全缺陷情况概览

本部分展示被测项目查出缺陷的数量，由此对被测项目的安全性进行大致的评估。图 2 分别展示了项目不同级别缺陷的数量，并按照高危缺陷数量对项目进行了排序，图中用蓝色折线图展示了每千行包含缺陷数[1]。

[1]每千行缺陷数计算方式：缺陷总数/代码行数*1000，精确到小数点后两位

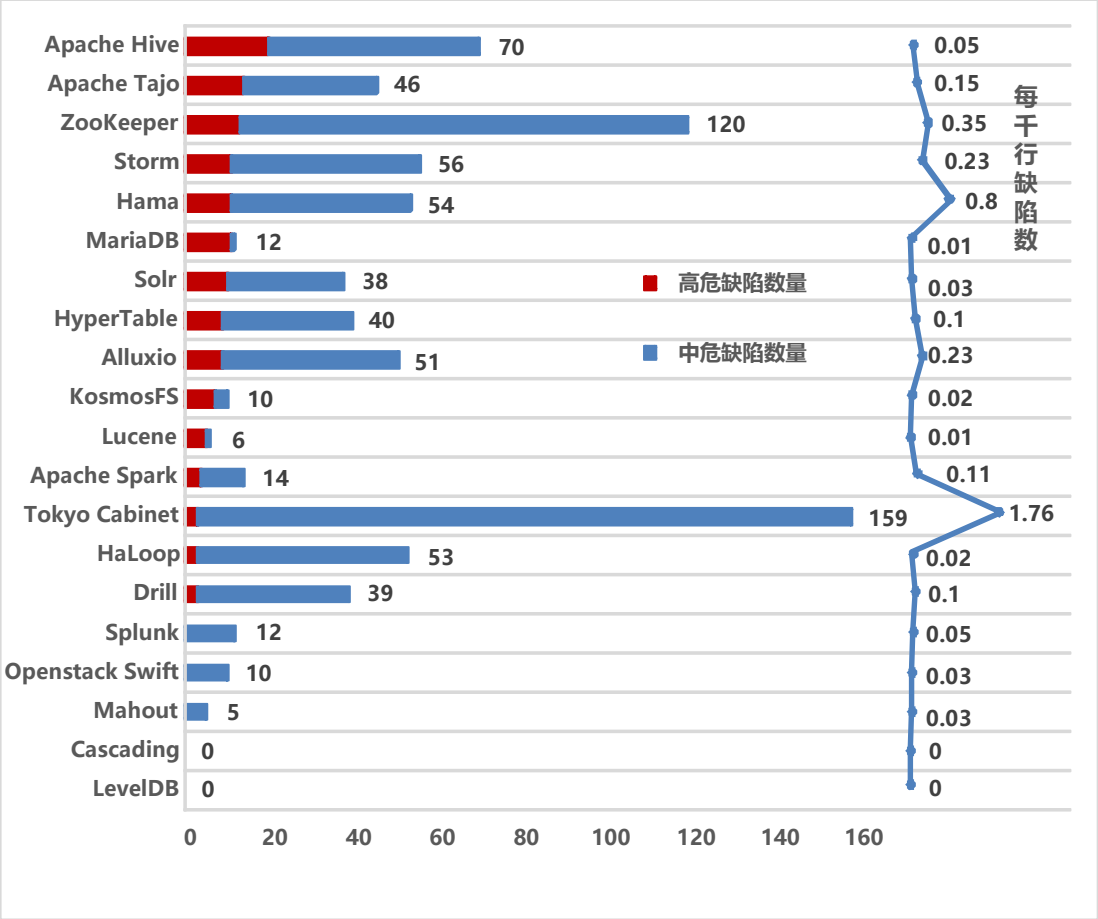

图 2 开源软件项目缺陷情况

从中可以看出，本次选取的大数据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高危缺陷 133 个，中危缺陷 662 个。缺陷数量排名靠前的项目处于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。

在所有被测软件中，中高危缺陷总数最多的是数据库管理系统 Tokyo Cabinet，包含 3 个高危缺陷和 156 个中危缺陷，其缺陷密度也是本次被测软件中最高的，平均每一千行代码就存在 1.76 个中高危缺陷。值得一提的是，这 159 个缺陷全部为缓冲区溢出问题，导致该类问题的主要原因是在数据拷贝时未进行有效的边界检查，一旦遇到恶意构造的数据，可能导致任意代码执行、提权等的严重问题。高危缺陷数量最多的是分布式数据仓库 Apache Hive，包含高危缺陷 20 个。此外，分布式应用程序协调服务 ZooKeeper 也总体风险较高，包含 13 个高危缺陷，107 个中危缺陷，中危缺陷中大多

数为资源泄露问题，这类问题将逐渐消耗系统资源，导致性能下降，可能被恶意攻击者利用以发动拒绝服务攻击。

全部被测软件中，有两款软件不存在中高危缺陷，分别是 Cascading（Hadoop 集群数据处理 API）和 LevelDB（键值对数据库）。此外，Mahout（分布式机器学习算法开发框架）、Openstack Swift（分布式对象存储服务）和 Splunk（机器数据的智能分析管理平台）均无高危缺陷，代码安全性较好。

4.2 高危安全缺陷分布情况

本部分对高危缺陷的分布情况进行分析说明。图 3 展示了被测项目中高危缺陷大类的分布情况。数据表明，大多数缺陷为“输入验证与表示”类缺陷，该类缺陷主要是由于对用户输入未做充分验证导致的，易造成缓冲区溢出、跨站脚本及各类注入问题，一旦攻击者构造恶意输入，可能造成任意命令执行、任意文件读取等严重安全问题。

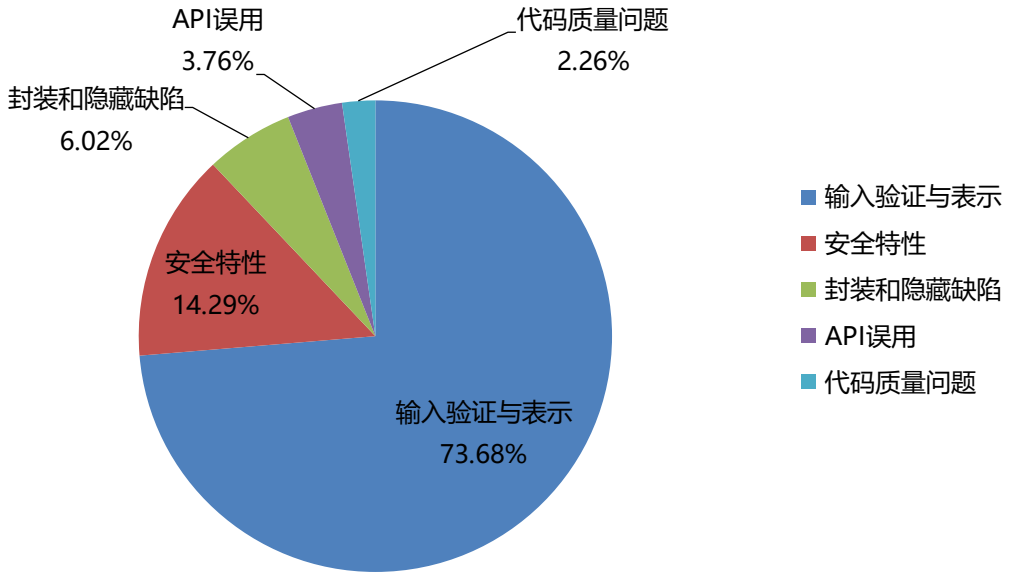

图 3 被测项目中高危安全缺陷的分布情况（按大类划分）

“安全特性”类缺陷也占据了一定份额，这类缺陷主要涵盖身份认证、数据加密等方面的问题，攻击者可利用该类缺陷实现越权访问，窃取隐私信息等。根据此次检测结果，多个软件存在“不安全的随机数”问题，这将严重降低软件抵御加密攻击的能力。

图4进一步展示了被测项目中的各种具体的高危安全缺陷的分布情况。为方便展示，将出现不超过 5 次的缺陷统一归入“其他”，主要包括空指针解引用、API 误用等问题。在被测的 25 个项目中，出现较多的几类缺陷是跨站脚本（29.32%，39 个）、路径遍历（12.03%，16 个）和不安全的随机数（10.53%，14 个）。下面对这三种缺陷进行简要说明，并给出防范建议。

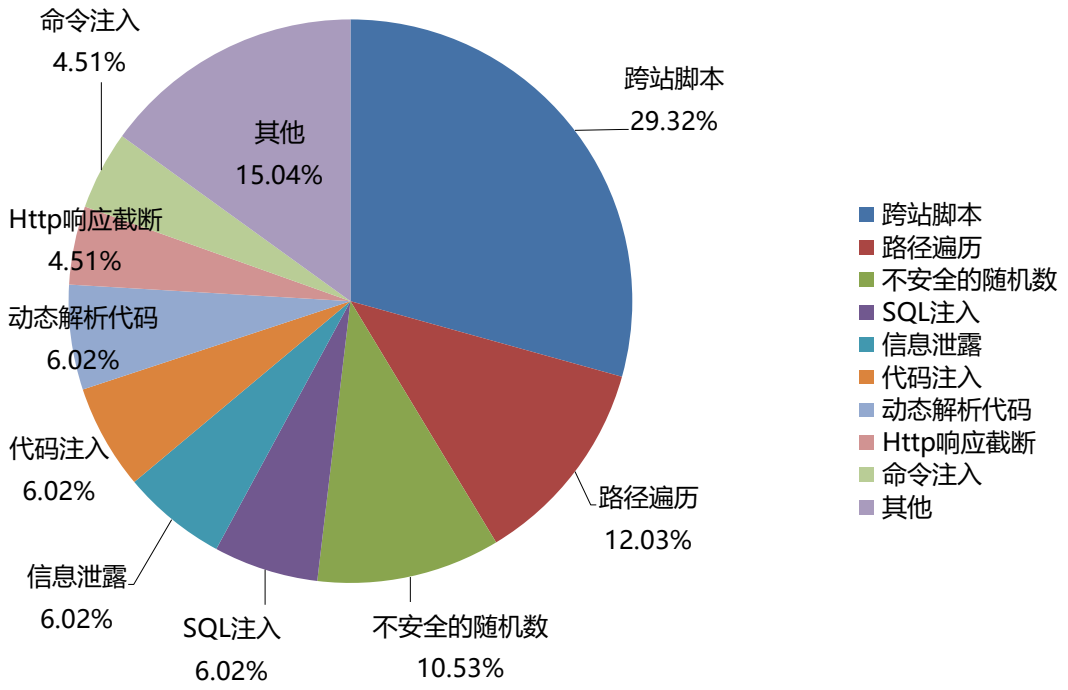

图 4 被测项目中高危安全缺陷的分布情况（按具体缺陷划分）

1）跨站脚本攻击（属于输入验证与表示类缺陷）

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）路径遍历（属于输入验证与表示类缺陷）

危害：应用程序对用户可控制的输入未经合理校验，就传送给文件访问 API。攻击者可能会使用一些特殊的字符（如“..”和“/”）摆脱受保护的限制，访问一些受保护的文件或目录。

防范：严格验证用户的输入，建议创建合法资源名的列表，并规定用户只能访问其中的文件。

3）不安全的随机数（属于安全特性类缺陷）

危害：在对安全性要求较高的环境中，使用一个能产生可预测数值的函数作为随机数据源，将降低系统抵御加密攻击的能力，导致如易于猜测的密码、可预测的加密密钥、会话劫持攻击和 DNS 欺骗等严重缺陷。

防范：应使用密码学的伪随机数生成器，并使用信息熵最大的信息作为密码学伪随机数生成器的种子。如果信息熵不可用，可以在使用密码学伪随机数生成器的时候改变其种子来降低威胁。

4.3 安全缺陷总体分布情况

上文针对被测项目中的高危缺陷的检出情况对项目的安全状况进行了分析。通常来说，与高危缺陷相比，中危缺陷在实际运行环境中的危害相对较小，但仍不容忽视，且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危缺陷在内的安全缺陷的总体分布情况。

图 5 展示了被测项目中安全缺陷大类的分布情况。与高危级别的缺陷分布情况相比，代码质量类缺陷（193 个）所占比例大幅提升。其中，项目中共查出 128 处资源泄露缺

陷和 51 处空指针解引用缺陷，反映出代码编写的不规范。这类问题通常是由于开发者遗漏了对部分执行路径的处理导致的，例如，仅在常规路径进行了资源释放或空指针判断操作，而忽略了出现异常情况的路径。与输入验证类问题相比，这类问题被直接用于发动攻击的可能性较小，但仍然会造成性能降低、程序不稳定等风险，严重情况下也会导致系统运行异常、甚至系统崩溃。建议开发者辅助使用自动化检测工具针对所有路径进行扫描和验证，以减少此类问题的发生。

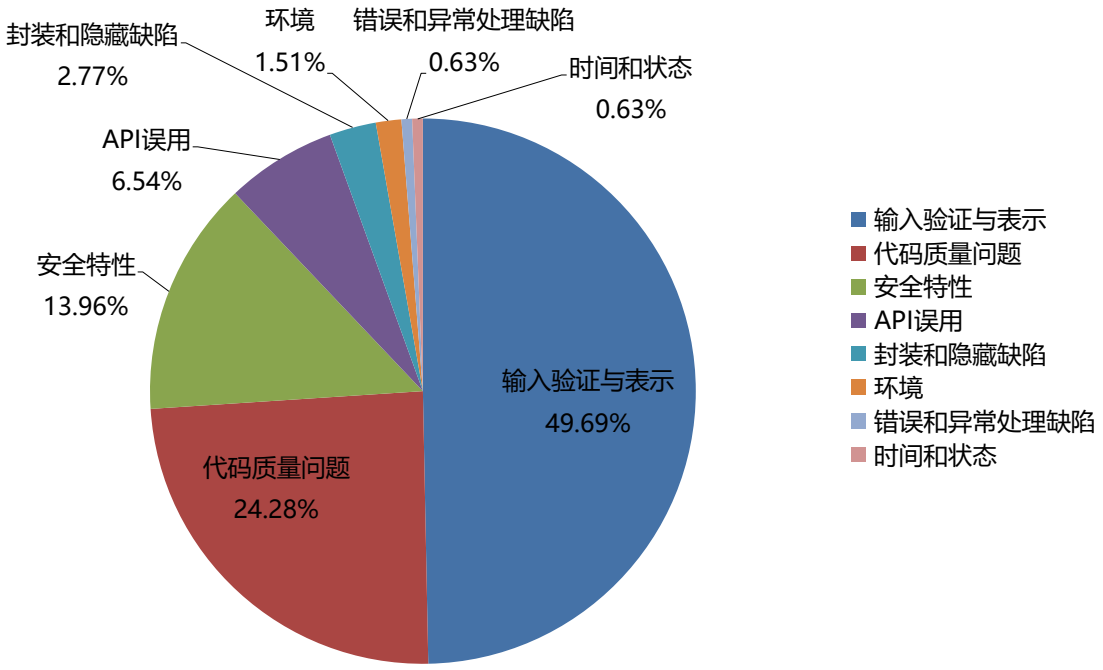

图 5 被测项目中的中高危安全缺陷的分布情况（按大类划分）

图 6 进一步展示了被测项目中的各种具体的中高危安全缺陷的分布情况。本次检测结果中有 24 种出现不超过 10 次的缺陷，如“命令注入”、“SQL 注入”、“弱密码”等，为方便展示，将其统一归入“其他”。在被测的 20 个项目中，出现较多的缺陷是是缓冲区溢出（20.25%，161 个）、跨站脚本攻击（17.74%，141 个）和资源泄露（16.10%，128 个）。下面对这几种缺陷进行简要说明，并给出防范建议。

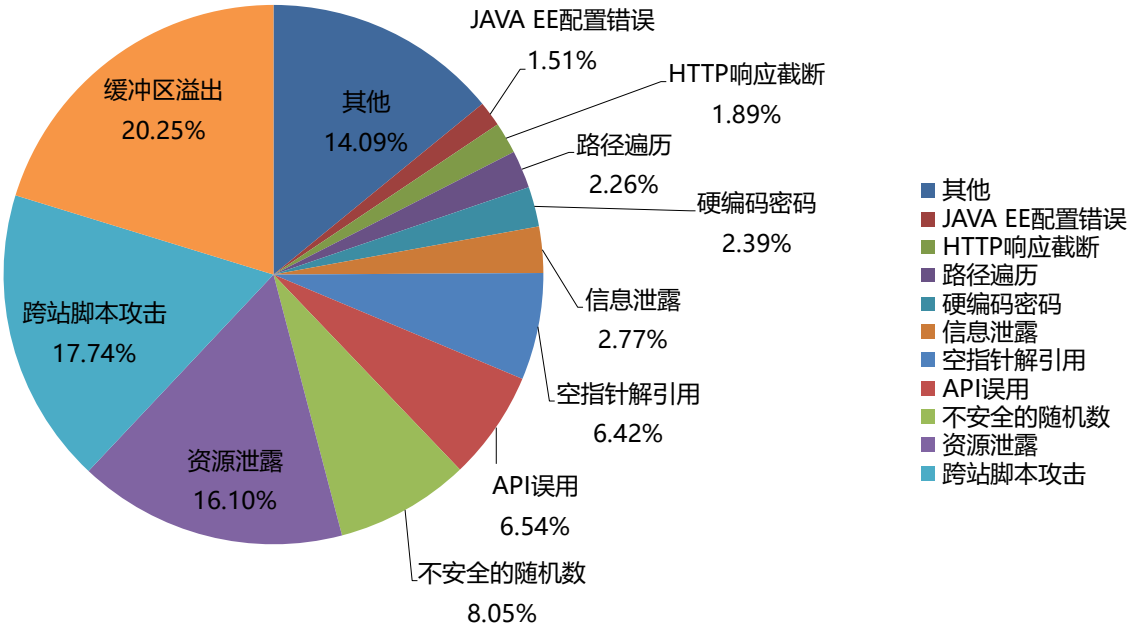

图 6 被测项目中的中高危安全缺陷的分布情况（按具体安全缺陷种类划分）

1) 缓冲区溢出（属于输入验证与表示类缺陷）

危害：在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。

2) 跨站脚本攻击（属于输入验证与表示类缺陷）

危害：向 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) 资源泄露（属于代码质量类缺陷）

危害：当程序未及时释放资源时，会降低系统性能。攻击者可能会通过耗尽系统资源的方式发起拒绝服务攻击。

防范：确保在所有程序路径上及时释放资源。

5 关于本报告的说明

一、本报告仅从代码角度进行缺陷分析。本报告中统计的缺陷是指由于代码编写不规范导致的有可能被攻击者利用的安全隐患。在实际系统中，由于软件实际部署环境、安全设备等的限制，部分缺陷可能无法通过渗透测试得到验证。

二、本报告中的缺陷仅适用于表 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。