

硕士学位论文预答辩



基于多源知识的开源软件漏洞的补丁识别方法

Finding Patches for Open Source Software
Vulnerabilities from Multi-Source Knowledge

答辩人：许聪颖

导师：陈碧欢



復旦大學
FUDAN UNIVERSITY

目录

- 01 背景知识及相关工作
- 02 开源软件漏洞补丁的经验研究
- 03 开源软件漏洞的补丁识别方法
- 04 实验评估
- 05 总结与展望

背景知识及 相关工作

01 背景知识及相关工作

研究背景

- 在软件开发过程中，开发人员大量会使用开源软件中的功能，节省开发时间，加快开发速度。
- 伴随着开发效率的提高，开源软件中的安全漏洞也会被引入软件系统。
- 近些年所披露的开源软件安全漏洞越来越多，已知的漏洞数量已超过10070。
 - 据Black Duck公司发布的《开源安全和风险分析报告》显示，在分析的1,500 个应用程序中，98%的应用程序都使用了开源软件，且高达84%的应用程序包含至少一个已知的开源软件漏洞。

相关工作

- 研究如何降低开源软件漏洞带来的安全风险（漏洞检测、漏洞修复）
- 评估数据库中漏洞知识的质量（漏洞重现描述、软件、版本信息）
- 漏洞补丁知识采集

研究问题

- 漏洞数据库中补丁知识的质量情况如何？
- 能否实现自动化地查找漏洞补丁？

01 背景知识及相关工作 > 1.1 CVE及NVD

通用漏洞披露 (Common Vulnerabilities and Exposures, CVE), 是一个与网络安全有关的漏洞字典, 收集各种信息安全漏洞并分配唯一编号以便公众查阅及引用。

CVE-ID	
CVE-2021-44228	Learn more at National Vulnerability Database (NVD) • CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information
Description	
Apache Log4j2 2.0-beta9 through 2.12.1 and 2.13.0 through 2.15.0 JNDI features used in configuration, log messages, and parameters do not protect against attacker controlled LDAP and other JNDI related endpoints. An attacker who can control log messages or log message parameters can execute arbitrary code loaded from LDAP servers when message lookup substitution is enabled. From log4j 2.15.0, this behavior has been disabled by default. From version 2.16.0, this functionality has been completely removed. Note that this vulnerability is specific to log4j-core and does not affect log4net, log4cxx, or other Apache Logging Services projects.	
References	
Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.	
<ul style="list-style-type: none">• CERT-VN:VU#930724• URL:https://www.kb.cert.org/vuls/id/930724• CISCO:20211210 A Vulnerability in Apache Log4j Library Affecting Cisco Products: December 2021• URL:https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-apache-log4j-qRuKNEbd• CISCO:20211210 Vulnerabilities in Apache Log4j Library Affecting Cisco Products: December 2021• URL:https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-apache-log4j-qRuKNEbd• CISCO:20211210 Vulnerability in Apache Log4j Library Affecting Cisco Products: December 2021• URL:https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-apache-log4j-qRuKNEbd	

每一个 CVE 条目都有唯一通用标识符(CVE ID)、一段漏洞描述 (Description)以及至少一个参考链接(Reference)。

01 背景知识及相关工作 > 1.1 CVE及NVD

美国国家漏洞数据库(NVD)，与CVE平台数据完全同步，并为每个漏洞条目 (CVE Entry)提供更丰富的信息，如:影响的软件名及版本、修复信息、严重性评分、影响评级等。

🚩 CVE-2021-44228 Detail

Current Description

Apache Log4j2 2.0-beta9 through 2.12.1 and 2.13.0 through 2.15.0 JNDI features used in configuration, log messages, and parameters do not protect against attacker controlled LDAP and other JNDI related endpoints. An attacker who can control log messages or log message parameters can execute arbitrary code loaded from LDAP servers when message lookup substitution is enabled. From log4j 2.15.0, this behavior has been disabled by default. From version 2.16.0, this functionality has been completely removed. Note that this vulnerability is specific to log4j-core and does not affect log4net, log4cxx, or other Apache Logging Services projects.

[+View Analysis Description](#)

Severity

CVSS Version 3.xCVSS Version 2.0

CVSS 3.x Severity and Metrics:

 NIST: NVD





Base Score: 10.0 CRITICAL

Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H

NVD Analysts use publicly available information to associate vector strings and CVSS scores. We also display any CVSS information provided within the CVE List from the CNA.

Note: NVD Analysts have published a CVSS score for this CVE based on publicly available information at the time of analysis. The CNA has not provided a score within the CVE List.

Weakness Enumeration

CWE-ID	CWE Name	Source
CWE-502	Deserialization of Untrusted Data	 NIST  Apache Software Foundation
CWE-400	Uncontrolled Resource Consumption	 Apache Software Foundation
CWE-20	Improper Input Validation	 Apache Software Foundation

Known Affected Software Configurations [Switch to CPE 2.2](#)

Configuration 1 [\(hide\)](#)

🚩 cpe:2.3:a:apache:log4j:2.0:-:*:*:*:*		
Show Matching CPE(s)▼		
🚩 cpe:2.3:a:apache:log4j:2.0:beta9:*:*:*:*		
Show Matching CPE(s)▼		
🚩 cpe:2.3:a:apache:log4j:2.0:rc1:*:*:*:*		
Show Matching CPE(s)▼		
🚩 cpe:2.3:a:apache:log4j:2.0:rc2:*:*:*:*		
Show Matching CPE(s)▼		
🚩 cpe:2.3:a:apache:log4j:*:*:*:*:*	From (including) 2.0.1	Up to (excluding) 2.12.2
Show Matching CPE(s)▼		
🚩 cpe:2.3:a:apache:log4j:*:*:*:*:*	From (including) 2.13.0	Up to (excluding) 2.15.0
Show Matching CPE(s)▼		

01 背景知识及相关工作 > 1.2 漏洞公告

漏洞公告(Advisory)，也被称为漏洞通告，一般是由受漏洞影响的软件的厂商(Vendor)对外发布的安全漏洞警报，通常包含: 漏洞触发描述、漏洞影响结果、漏洞软件名、软件版本等描述信息，有时也会包含漏洞发现者、漏洞问题报告(Issue Report)、漏洞补丁等知识。

Fixed in Log4j 2.15.0 (Java 8)

CVE-2021-44228 🚩: Apache Log4j2 JNDI features do not protect against attacker controlled LDAP and other JNDI related endpoints.

CVE-2021-44228 🚩	Remote Code Execution
Severity	Critical
Base CVSS Score	10.0 CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H
Versions Affected	All versions from 2.0-beta9 to 2.14.1

Description

In Apache Log4j2 versions up to and including 2.14.1 (excluding security releases 2.3.1, 2.12.2 and 2.12.3), the JNDI features used in configurations, log messages, and parameters do not protect against attacker-controlled LDAP and other JNDI related endpoints. An attacker who can control log messages or log message parameters can execute arbitrary code loaded from LDAP servers when message lookup substitution is enabled.

Mitigation

Log4j 1.x mitigation

Log4j 1.x does not have Lookups so the risk is lower. Applications using Log4j 1.x are only vulnerable to this attack when they use JNDI in their configuration. A separate CVE (CVE-2021-4104) has been filed for this vulnerability. To mitigate: Audit your logging configuration to ensure it has no JMSAppender configured. Log4j 1.x configurations without JMSAppender are not impacted by this vulnerability.

Log4j 2.x mitigation

Implement one of the following mitigation techniques:

- Upgrade to Log4j 2.3.1 (for Java 6), 2.12.3 (for Java 7), or 2.17.0 (for Java 8 and later).
- Otherwise, in any release other than 2.16.0, you may remove the JndiLookup class from the classpath: `zip -q -d log4j-core-*.jar org/apache/logging/log4j/core/lookup/JndiLookup.class`

Note that only the log4j-core JAR file is impacted by this vulnerability. Applications using only the log4j-api JAR file without the log4j-core JAR file are not impacted by this vulnerability.

Also note that Apache Log4j is the only Logging Services subproject affected by this vulnerability. Other projects like Log4net and Log4cxx are not impacted by this.

Work in progress

The Log4j team will continue to actively update this page as more information becomes known.

Credit

This issue was discovered by Chen Zhaojun of Alibaba Cloud Security Team.

References

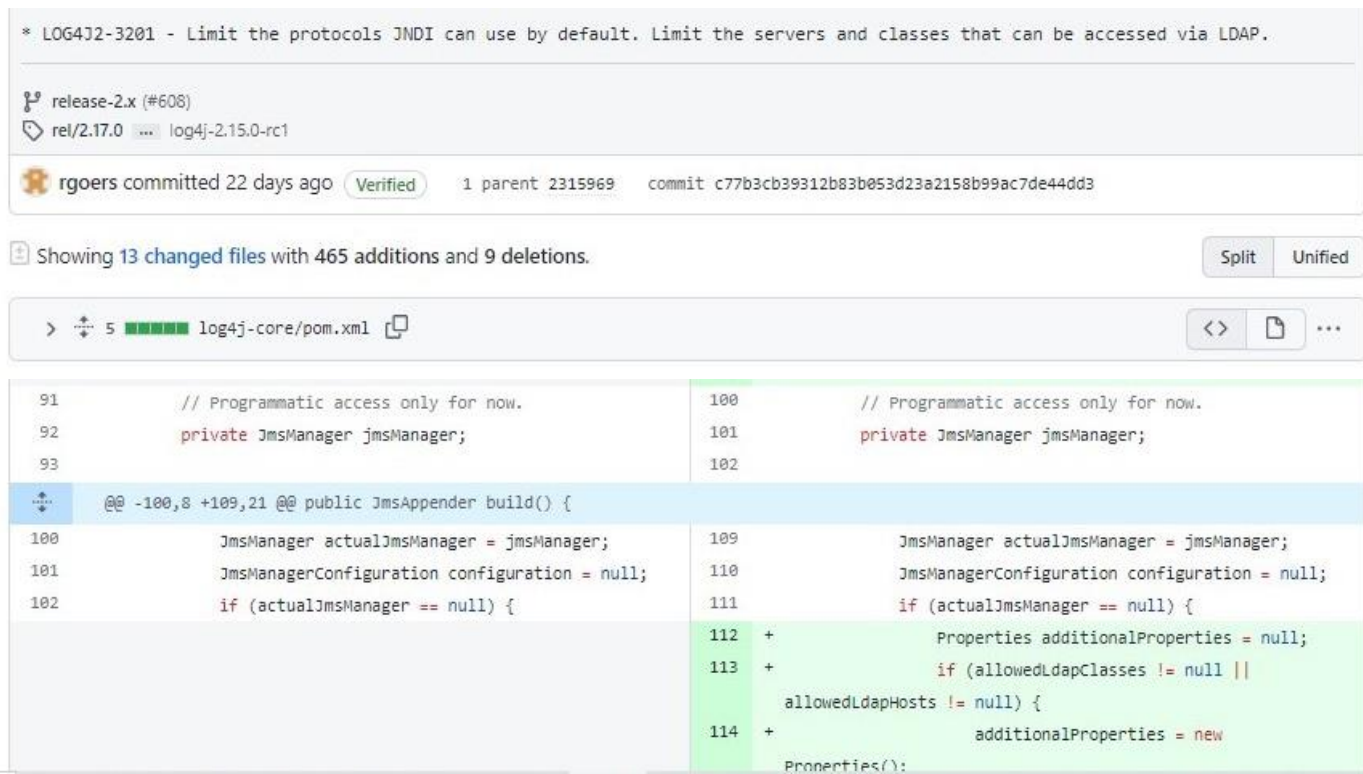
- <https://issues.apache.org/jira/browse/LOG4J2-3201> 🚩
- <https://issues.apache.org/jira/browse/LOG4J2-3198> 🚩



01 背景知识及相关工作 > 1.3 漏洞补丁

补丁(Patch), 也称: 补丁程序, 是指对计算机程序进行的一组更改, 旨在更新其功能或修复其缺陷。

漏洞补丁(Vulnerability Patch)则指为修复程序中的安全漏洞所开发的补丁, 补丁的形式通常是Git和SVN中的代码提交(Commit), 或是文本文件(.patch)。



```
* LOG4J2-3201 - Limit the protocols JNDI can use by default. Limit the servers and classes that can be accessed via LDAP.

release-2.x (#608)
rel/2.17.0 ... log4j-2.15.0-rc1

rgoers committed 22 days ago Verified 1 parent 2315969 commit c77b3cb39312b83b053d23a2158b99ac7de44dd3

Showing 13 changed files with 465 additions and 9 deletions.

> 5 log4j-core/pom.xml

91 // Programmatic access only for now.
92 private JmsManager jmsManager;
93
@@ -100,8 +109,21 @@ public JmsAppender build() {
100 JmsManager actualJmsManager = jmsManager;
101 JmsManagerConfiguration configuration = null;
102 if (actualJmsManager == null) {
109 JmsManager actualJmsManager = jmsManager;
110 JmsManagerConfiguration configuration = null;
111 if (actualJmsManager == null) {
112 + Properties additionalProperties = null;
113 + if (allowedLdapClasses != null ||
    allowedLdapHosts != null) {
114 + additionalProperties = new
    Properties();
```


开源软件漏洞补丁 的经验研究



02 开源软件漏洞补丁的经验研究 > 2.1 研究设计及数据准备

研究目的

- 探究当前商业漏洞数据库中开源软件漏洞补丁的质量和特征，包括补丁覆盖度、补丁一致性、补丁类型、补丁映射以及补丁准确性5个方面。

研究问题

- RQ1 补丁覆盖率分析: 当前商业漏洞数据库中，漏洞补丁的覆盖度如何?即，有多少漏洞含有补丁知识?
- RQ2 补丁一致性分析: 不同漏洞库间，漏洞补丁的一致性如何?即，有多少漏洞在不同漏洞数据库中有相同的补丁?
- RQ3 补丁类型分析: 开源软件漏洞补丁有哪些类型?
- RQ4 补丁映射分析: 开源软件漏洞与其补丁在数量上有怎样的映射关系?
- RQ5 补丁准确性分析: 当前商业漏洞数据库中，漏洞补丁的准确性如何?

数据准备

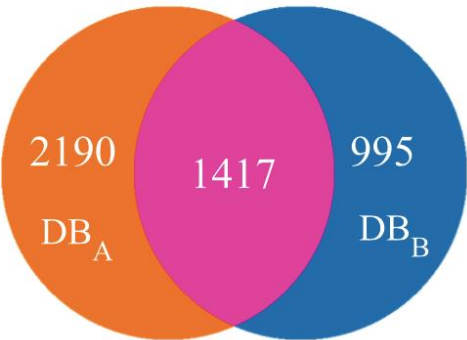
- 从五家知名度较高的商业公司Black Duck、Sonatype、WhiteSource、Veracode 和 Snyk中，选择两个漏洞数据库作为研究对象，简称为: *DBA* 和 *DBB*，共获取10070个开源漏洞，用以研究RQ1和RQ2。
- 基于10070个开源漏洞，人工查找1295个安全漏洞的补丁，用以准确地研究RQ3、4、5。

02 开源软件漏洞补丁的经验研究 > 2.2 研究结果

RQ1: 补丁覆盖率分析



(a) 开源软件漏洞



(b) 含补丁知识的开源软件漏洞

- 10,070 个开源软件漏洞中，仅有4,602个漏洞含有补丁，补丁覆盖率为45.7%。
- DB_A 中，3,607(41.8%)的漏洞含有补丁；DB_B 中，2,412(41.2%)的漏洞含有补丁。
- DB_A 和 DB_B 数据库共有的4,418个开源软件漏洞中，仅有1,417(32.0%)的漏洞含有补丁。

RQ2: 补丁一致性分析

表 3-1 DB_A 与 DB_B 补丁一致性分析结果

补丁一致	存在性不一致			内容不一致		
	总数	某一数据库中无漏洞	某一数据库中无补丁	总数	补丁为包含关系	补丁非包含关系
907 (19.7%)	3,185 (69.2%)	1,392 (30.2%)	1,793 (39.0%)	510 (11.1%)	176 (3.8%)	334 (7.3%)

- 4,602个含有补丁漏洞中，只有907(19.7%)的漏洞在 DB_A 和 DB_B 中有一致的补丁。
- 超过三分之二(即3,185 69.2%)的漏洞在数据库 DB_A 和 DB_B 中补丁存在性不一致。
- 510(11.1%)的漏洞补丁都存在于 DB_A 和 DB_B 中，但补丁集不一致。

02 开源软件漏洞补丁的经验研究 > 2.2 研究结果

RQ3: 补丁类型分析

表 3-2 补丁类型分析结果

补丁总数	GitHub 代码提交	SVN 代码提交	其他 Git 平台代码提交
3,043	2,852 (93.7%)	136 (4.5%)	55 (1.8%)
漏洞总数	仅 GitHub 代码提交	仅 SVN 代码提交	仅其他 Git 平台代码提交
1,295	1,202 (92.8%)	4 (0.3%)	30 (2.3%)

RQ4: 补丁映射分析

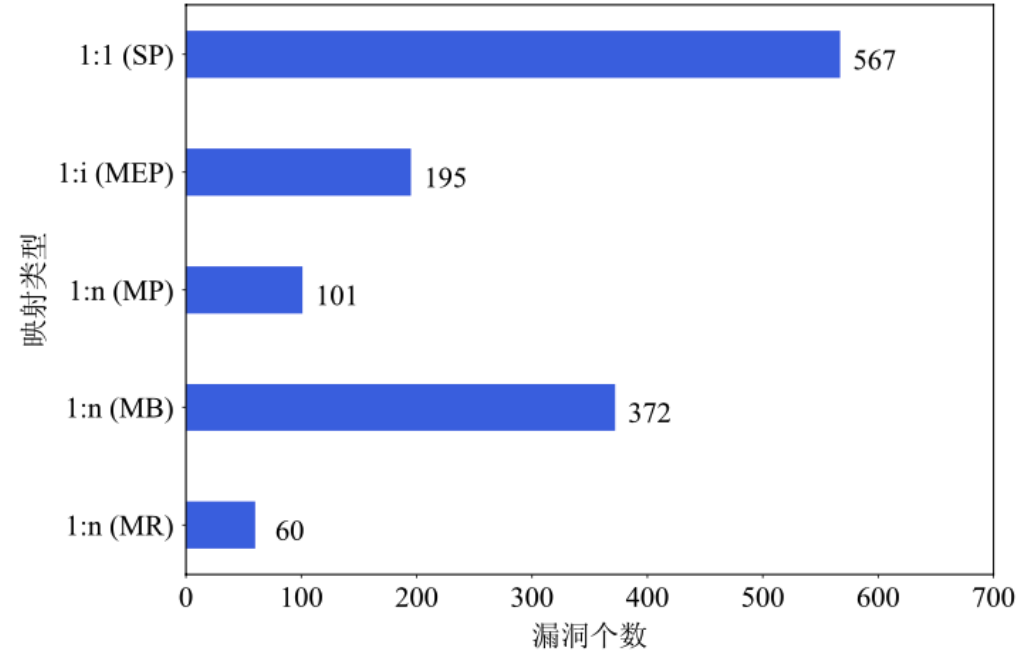


图 3-4 漏洞及其补丁映射类型统计

- 一对一，漏洞与其补丁在数量上为一对一的关系，SP(Single Patch)。
- 一对一组，漏洞与其补丁在数量上非一对一关系。然而，这些补丁又都是等效的，MEP(Multiple Equivalent Patch)。
- 一对多，漏洞与其补丁在数量上为一对多的关系，即一个漏洞需多个非等效的补丁来修复，MP (Multiple Patch)、MB(Multiple Branches)、MR(Multiple Repositories)。

02 开源软件漏洞补丁的经验研究 > 2.2 研究结果

RQ5: 补丁准确性分析

表 3-3 DB_A 和 DB_B 补丁准确性评估结果

映射类型	数量	DB_A			DB_B		
		Pre.	Rec.	F1	Pre.	Rec.	F1
1:1 (SP)	567	0.908	0.915	0.910	0.900	0.921	0.906
1:i (MEP)	195	0.935	0.898	0.902	0.924	0.909	0.906
1:n (MP)	101	0.923	0.483	0.616	0.911	0.520	0.638
1:n (MB)	372	0.941	0.510	0.620	0.932	0.436	0.555
1:n (MR)	60	0.913	0.610	0.695	0.964	0.526	0.636
总计	1,295	0.923	0.748	0.793	0.917	0.730	0.771

- DB_A 和 DB_B 具有较高的精确率，但经常会遗漏一些漏洞的补丁，尤其是对于具有多个补丁的漏洞。

02 开源软件漏洞补丁的经验研究 > 2.3 研究发现

商业漏洞数据库中漏洞补丁质量并不理想

- 开源软件漏洞补丁缺失情况较为普遍，商业数据库 *DBA* 和 *DBB* 中开源软件漏洞的补丁覆盖率仅为 41.8% 和 41.2%。
- 商业漏洞数据库 *DBA* 和 *DBB* 具有较高的精确率，但经常会遗漏一些漏洞的补丁，尤其是对于具有多个补丁的漏洞。

这体现出当前开源软件漏洞数据库的不足，以及利用自动化补丁识别方法完善漏洞数据的需求。

开源软件漏洞补丁在类型、映射关系方面有一定的特殊性

- 93.7% 的补丁都是GitHub代码提交的形式。
- 开源软件漏洞与其补丁之间映射关系具有多样性，超过 40% 的漏洞与其补丁具有一对多的映射关系。

设计自动化补丁识别方法时应充分考虑以上特征。

开源软件漏洞的 补丁识别方法

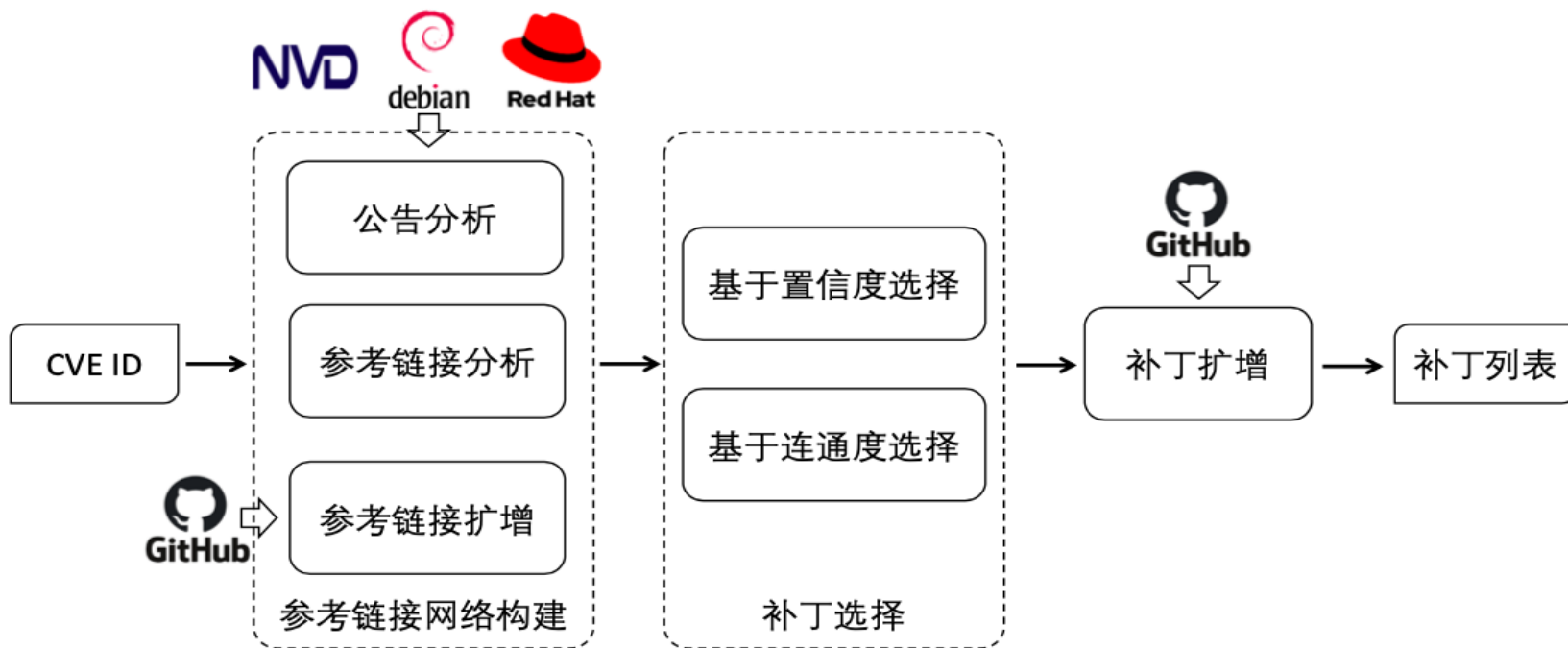


03 开源软件漏洞的补丁识别方法

TRACER

- 核心思想是: 漏洞补丁会在讨论和解决漏洞的、多种来源的漏洞公告、分析报告等参考链接中被频繁地提及和引用。
- 首先设计了一种基于多知识源的漏洞参考链接网络, 再从该网络中选出具有最高置信度和连通度的补丁节点作为结果, 并基于选定的补丁进行补丁扩展, 从而构建一对多的漏洞补丁映射关系。

方法概览



实验评估

04 实验评估 > 4.1 实验问题设计

实验目的

- 本文从准确性、削弱性、敏感度、通用性及实用性五个方面以尽可能全面地评估 TRACER的有效性。

实验问题

- RQ6 准确性评估: TRACER识别漏洞补丁的准确性如何?
- RQ7 削弱性分析: 如果去除 TRACER 中的某些环节, 对于最终结果会有怎样的影响?
- RQ8 敏感度分析: TRACER对设计参数的敏感性如何?
- RQ9 通用性分析: TRACER在更大范围的开源软件漏洞上表现如何?
- RQ10 实用性分析: TRACER在实际使用中表现如何?

评估指标

- Coverage(覆盖率)、Precision(精确率)、Recall(召回率)和 F1-Score(F1 值)

04 实验评估 > 4.2 实验结果

RQ6: 准确性评估

表 5-1 TRACER VS. 基于启发式规则的方法和商业数据库

映射类型	数量	TRACER				检索 NVD			
		Coverage	Pre.	Rec.	F1	Coverage	Pre.	Rec.	F1
1:1 (SP)	567	465 (82.0%)	0.860	0.951	0.881	282 (49.7%)	0.973	0.986	0.977
1:i (MEP)	195	189 (96.9%)	0.886	0.918	0.888	70 (35.9%)	0.932	0.925	0.921
1:n (MP)	101	81 (80.2%)	0.872	0.741	0.761	33 (32.7%)	0.980	0.552	0.683
1:n (MB)	372	349 (93.8%)	0.861	0.788	0.795	148 (39.8%)	0.979	0.416	0.546
1:n (MR)	60	56 (93.3%)	0.831	0.620	0.659	14 (23.3%)	1.000	0.708	0.794
总计	1,295	1,140 (88.0%)	0.864	0.864	0.837	527 (40.7%)	0.970	0.805	0.842
映射类型	数量	检索 GitHub				检索 NVD 以及 GitHub			
		Coverage	Pre.	Rec.	F1	Coverage	Pre.	Rec.	F1
1:1 (SP)	567	95 (16.8%)	0.416	0.642	0.471	345 (60.8%)	0.839	0.930	0.864
1:i (MEP)	195	33 (16.9%)	0.472	0.490	0.452	91 (46.7%)	0.821	0.867	0.820
1:n (MP)	101	28 (27.8%)	0.536	0.445	0.461	49 (48.5%)	0.779	0.605	0.647
1:n (MB)	372	126 (33.9%)	0.445	0.236	0.284	201 (54.0%)	0.704	0.393	0.465
1:n (MR)	60	23 (38.3%)	0.627	0.345	0.413	33 (55.0%)	0.801	0.539	0.604
总计	1,295	305 (23.6%)	0.461	0.417	0.386	719 (55.5%)	0.793	0.732	0.720
映射类型	数量	DB _A				DB _B			
		Coverage	Pre.	Rec.	F1	Coverage	Pre.	Rec.	F1
1:1 (SP)	567	100.0%	0.908	0.915	0.910	100.0%	0.900	0.921	0.906
1:i (MEP)	195	100.0%	0.935	0.898	0.902	100.0%	0.924	0.909	0.906
1:n (MP)	101	100.0%	0.923	0.483	0.616	100.0%	0.911	0.520	0.638
1:n (MB)	372	100.0%	0.941	0.510	0.620	100.0%	0.932	0.436	0.555
1:n (MR)	60	100.0%	0.913	0.610	0.695	100.0%	0.964	0.526	0.636
总计	1,295	100.0%	0.923	0.748	0.793	100.0%	0.917	0.730	0.771

- 与现有的基于启发式规则的方法相比，TRACER 能够显著的提高补丁覆盖率和 F1 值。TRACER 将补丁覆盖率提高58.6%到273.8%，同时，将 F1 值提高 116.8%。
- 与漏洞数据库 *DBA* 和 *DBB* 相比，TRACER 以略低的补丁精确率和覆盖率为代价，拥有更为显着的召回率。

这表明，TRACER 可用于补充现有漏洞数据库缺失的漏洞补丁数据。

04 实验评估 > 4.2 实验结果

RQ7: 削弱性分析

表 5-2 TRACER 削弱性分析结果（1）

映射类型	数量	TRACER				v_1^1 : TRACER w/o NVD			
		Coverage	Pre.	Rec.	F1	Coverage	Pre.	Rec.	F1
1:1 (SP)	567	465 (82.0%)	0.860	0.951	0.881	281 (49.6%)	0.820	0.936	0.846
1:i (MEP)	195	189 (96.9%)	0.886	0.918	0.888	116 (59.5%)	0.882	0.935	0.886
1:n (MP)	101	81 (80.2%)	0.872	0.741	0.761	60 (59.4%)	0.881	0.728	0.766
1:n (MB)	372	349 (93.8%)	0.861	0.788	0.795	288 (77.4%)	0.876	0.780	0.800
1:n (MR)	60	56 (93.3%)	0.831	0.620	0.659	52 (86.7%)	0.848	0.551	0.624
总计	1,295	1,140 (88.0%)	0.864	0.864	0.837	797 (61.5%)	0.856	0.839	0.815
映射类型	数量	v_1^2 : TRACER w/o Debian				v_1^3 : TRACER w/o Red Hat			
		Coverage	Pre.	Rec.	F1	Coverage	Pre.	Rec.	F1
1:1 (SP)	567	457 (80.6%)	0.847	0.943	0.869	454 (80.1%)	0.853	0.943	0.874
1:i (MEP)	195	187 (95.6%)	0.880	0.912	0.882	188 (96.4%)	0.883	0.918	0.886
1:n (MP)	101	79 (78.2%)	0.851	0.716	0.739	80 (79.2%)	0.880	0.736	0.760
1:n (MB)	372	344 (92.5%)	0.838	0.760	0.771	337 (90.6%)	0.844	0.761	0.767
1:n (MR)	60	55 (91.7%)	0.819	0.613	0.651	56 (93.3%)	0.738	0.640	0.618
总计	1,295	1,122 (86.6%)	0.848	0.849	0.821	1,115 (86.1%)	0.851	0.853	0.823
映射类型	数量	v_1^4 : TRACER w/o GitHub				v_1^5 : TRACER w/o Network			
		Coverage	Pre.	Rec.	F1	Coverage	Pre.	Rec.	F1
1:1 (SP)	567	418 (73.7%)	0.898	0.943	0.908	390 (68.8%)	0.910	0.972	0.925
1:i (MEP)	195	176 (90.3%)	0.887	0.921	0.892	117 (60.0%)	0.956	0.959	0.941
1:n (MP)	101	73 (72.3%)	0.873	0.690	0.726	61 (60.4%)	0.943	0.669	0.743
1:n (MB)	372	333 (89.5%)	0.874	0.752	0.773	263 (70.7%)	0.908	0.575	0.659
1:n (MR)	60	53 (88.3%)	0.816	0.545	0.604	50 (83.3%)	0.920	0.641	0.712
总计	1,295	1,053 (81.3%)	0.883	0.841	0.835	881 (68.0%)	0.918	0.812	0.823

表 5-3 TRACER 削弱性分析结果（2）

映射类型	数量	v_2^1 : TRACER w/o Selection				v_2^2 : TRACER w/o Connectivity			
		Coverage	Pre.	Rec.	F1	Coverage	Pre.	Rec.	F1
1:1 (SP)	567	465 (82.0%)	0.632	0.961	0.680	322 (56.8%)	0.892	0.978	0.913
1:i (MEP)	195	189 (96.9%)	0.622	0.976	0.682	84 (43.1%)	0.929	0.939	0.915
1:n (MP)	101	81 (80.2%)	0.615	0.933	0.656	45 (44.6%)	0.953	0.685	0.764
1:n (MB)	372	349 (93.8%)	0.616	0.903	0.657	181 (48.7%)	0.927	0.787	0.821
1:n (MR)	60	56 (93.3%)	0.368	0.891	0.394	33 (55.0%)	0.885	0.722	0.772
总计	1,295	1,140 (88.0%)	0.611	0.940	0.658	665 (51.4%)	0.910	0.889	0.871
映射类型	数量	v_2^3 : TRACER w/o Confidence				v_2^4 : TRACER with Path Length			
		Coverage	Pre.	Rec.	F1	Coverage	Pre.	Rec.	F1
1:1 (SP)	567	465 (82.0%)	0.860	0.942	0.879	465 (82.0%)	0.833	0.957	0.859
1:i (MEP)	195	189 (96.9%)	0.888	0.913	0.889	189 (96.9%)	0.848	0.945	0.867
1:n (MP)	101	81 (80.2%)	0.880	0.722	0.751	81 (80.2%)	0.849	0.760	0.742
1:n (MB)	372	349 (93.8%)	0.871	0.765	0.784	349 (93.8%)	0.830	0.798	0.770
1:n (MR)	60	56 (93.3%)	0.849	0.462	0.550	56 (93.3%)	0.652	0.747	0.590
总计	1,295	1,140 (88.0%)	0.869	0.844	0.826	1,140 (88.0%)	0.827	0.882	0.812
映射类型	数量	v_2^5 : TRACER with Path Number				v_2^6 : TRACER w/o Expansion			
		Coverage	Pre.	Rec.	F1	Coverage	Pre.	Rec.	F1
1:1 (SP)	567	465 (82.0%)	0.805	0.951	0.837	465 (82.0%)	0.871	0.948	0.889
1:i (MEP)	195	189 (96.9%)	0.849	0.920	0.858	189 (96.9%)	0.910	0.914	0.902
1:n (MP)	101	81 (80.2%)	0.801	0.756	0.726	81 (80.2%)	0.873	0.696	0.732
1:n (MB)	372	349 (93.8%)	0.833	0.811	0.791	349 (93.8%)	0.860	0.506	0.590
1:n (MR)	60	56 (93.3%)	0.789	0.630	0.644	56 (93.3%)	0.847	0.567	0.629
总计	1,295	1,140 (88.0%)	0.819	0.873	0.809	1,140 (88.0%)	0.873	0.771	0.776

- TRACER 中的知识源、网络构建、补丁选择和补丁扩增步骤的设计对最终结果都有一定的贡献度和必要性。

04 实验评估 > 4.2 实验结果

RQ8: 敏感度分析

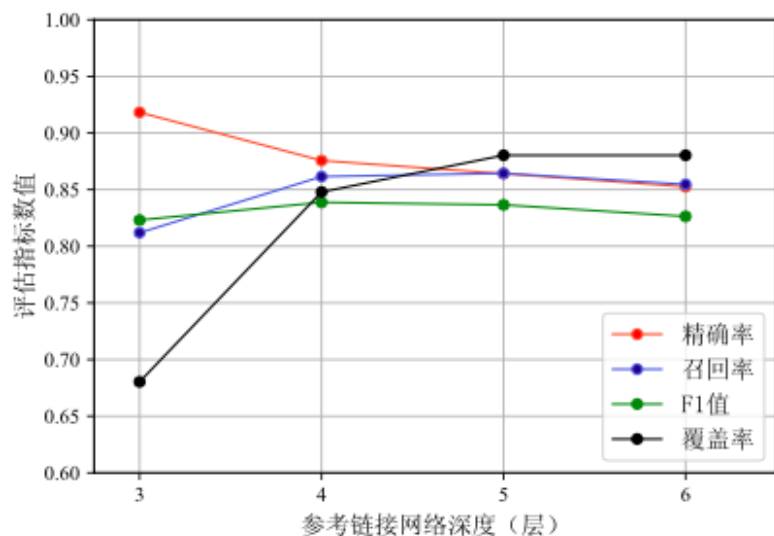


图 5-3 网络深度限制（层数）敏感性分析结果

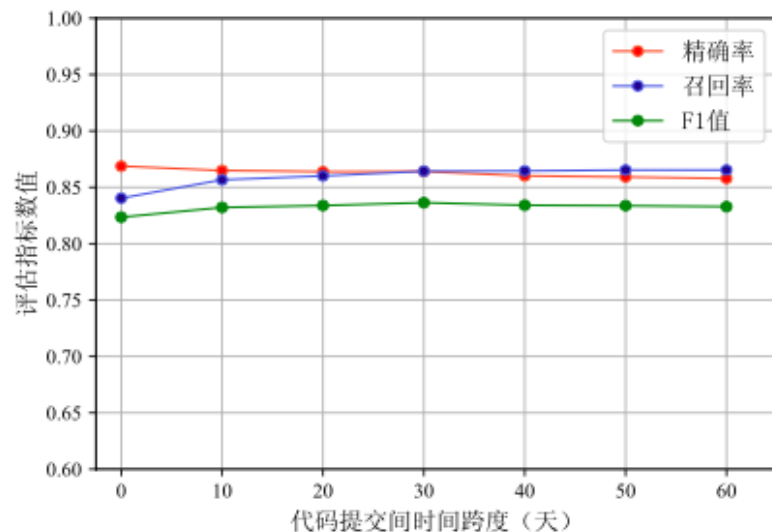


图 5-4 提交时间跨度的敏感性分析结果

- 随着网络层数的增加，网络中将包含更多补丁。
- 随着代码提交时间跨度的增加，TRACER 会搜索到更广的代码提交。

TRACER 的准确率对两个可配置参数的变化不是非常敏感，网络深度参数设置为 5 层、提交时间跨度设置为 30 天时效果相对最优。

04 实验评估 > 4.2 实验结果

RQ9: 通用性分析

- 数据集一共有3,185个漏洞，TRACER 找到了2,155(67.7%)漏洞的补丁。
- 数据集二共有5,468个漏洞，TRACER 找到了2,816(51.5%)漏洞的补丁。

表 5-4 TRACER 通用性分析结果

评估对象	数据集一（91 个漏洞）				数据集二（89 个漏洞）			
	Coverage	Pre.	Rec.	F1	Coverage	Pre.	Rec.	F1
TRACER	100.0%	0.823	0.845	0.784	100.0%	0.888	0.899	0.867
DB_A	62 (68.1%)	0.935	0.827	0.858	0.0%	—	—	—
DB_B	29 (31.8%)	0.885	0.664	0.725	0.0%	—	—	—

这些结果表明，即使对于更大范围的开源软件漏洞，TRACER 的准确率也基本稳定。在开源软件漏洞的补丁定位方面，TRACER 具有较好的通用性。

RQ10: 实用性分析

- 从国内外多所大学和科技公司的安全实验室中共招募了 10 名实验人员，他们中有软件安全方向的博士后、博士生、硕士生以及工程师。
- 随机选择了 10 个漏洞作为实验任务，对比分析用户在有无TRACER 的辅助下找到漏洞补丁的用时和准确性。

表 5-5 用户研究中任务的用时和准确率

任务	w/o TRACER				with TRACER			
	用时	Pre.	Rec.	F1	用时	Pre.	Rec.	F1
全部 10 个任务	5.66 mins	0.880	0.677	0.765	4.66 mins	0.983	0.920	0.951
5 单补丁任务	5.60 mins	0.960	0.960	0.960	3.84 mins	1.000	1.000	1.000
5 多补丁任务	5.72 mins	0.800	0.393	0.527	5.48 mins	0.967	0.840	0.899

结果表明，在实际使用中，TRACER 有助于用户更准确、更快速地识别到补丁。

04 实验评估 > 4.3 实验结论

本文从准确性、削弱性、敏感度、通用性及实用性五个方面对TRACER 进行了评估。结果表明:

- 准确性: TRACER 可以达到 88.0% 的补丁覆盖率、86.4% 的精确率和 86.4% 的召回率。
 - 与现有的基于启发式规则的方法相比, TRACER 能够显著地提高补丁覆盖率和 F1 值, 将补丁覆盖率提高58.6%到273.8%, 将 F1 值提高116.8%。
 - 与商业漏洞数据库 *DBA* 和 *DBB* 相比, TRACER 以略低的补丁精确率和覆盖率为代价, 拥有更为显著的召回率。
 - 这表明, TRACER 可用于补充现有漏洞数据库缺失的漏洞补丁数据。
- 削弱性: TRACER 中的知识源、网络构建、补丁选择和补丁扩增步骤的设计 对最终结果都有一定的贡献度和必要性。
- 敏感度: TRACER 的准确率对两个可配置参数的变化不是非常敏感, 且网络深度参数设置为 5 层、提交时间跨度设置为 30 天时效果相对最优。
- 通用性: 在更大范围的开源软件漏洞上, TRACER 具有较好的通用性, 覆盖率和准确率比较稳定。
 - 在商业漏洞数据库 *DBA* 和 *DBB* 都没有补丁时, TRACER 仍可以找到大量漏洞的补丁。这表明, TRACER 可以极大地补充或 增强现有的商业漏洞数据库。
- 实用性:在实际工作场景下, TRACER 有助于用户更准确、更快速地识别到 补丁。



总结与展望

05 总结与展望

本文总结

- 开展了一项针对开源软件漏洞补丁的经验研究，以了解当前商业漏洞数据库中开源软件漏洞补丁的质量和特征，涵盖补丁覆盖度、补丁一致性、补丁类型、补丁映射关系以及补丁准确性五个方面。
 - 研究发现: 商业漏洞数据库中开源软件漏洞补丁的质量并不理想，开源软件漏洞补丁在类型、映射关系方面有一定的特殊性。
- 基于经验研究的发现，本文提出了一种名为 TRACER 的基于多源知识的开源软件漏洞的补丁识别方法。该方法用于识别代码提交类型的补丁，并构建一对多的漏洞补丁映射关系。
- 本文还进行了大量实验，通过五个研究问题，从准确性、通用性、实用性等 多个方面对TRACER进行了评估。

未来展望

- TRACER不适用于没有 CVE ID 的开源漏洞，未来可以考虑以 Advisory ID、Issue ID 作为 TRACER 的输入，进一步挺高补丁覆盖率。
- TRACER 中仅仅包含了四个知识源，未来可以扩增更多的知识源，从而构建更完整的漏洞参考链接网络。
- TRACER 基于置信度和连通度选择补丁， 未来可以尝试基于语义的补丁识别方法。

硕士学位论文预答辩



谢谢聆听!

基于多源知识的开源软件漏洞的补丁识别方法

Finding Patches for Open Source Software
Vulnerabilities from Multi-Source Knowledge

答辩人：许聪颖

导师：陈碧欢



復旦大學
FUDAN UNIVERSITY