

An Exploratory Study on Self-Fixed Software Vulnerabilities in OSS Projects

Sara Forootani*, Andrea Di Sorbo*, Corrado A. Visaggio*

*University of Sannio, Italy

{forootani, disorbo, visaggio}@unisannio.it

発表者：坂本

背景 & 目的

▶ 背景

- ▶ 脆弱性の是正管理は、開発者の観点から重要
- ▶ 脆弱性の修正には特定のスキルや知識が必要[24], [5]
- ▶ これまでの不具合修正割り当てが脆弱性の修正に適用できるかは明らかではない



自己修正が効率がいいのでは

▶ 目的

- ▶ 脆弱性の修正において、自己修正は有効か

[24] P. J. Morrison, R. Pandita, X. Xiao, R. Chillarege, and L. Williams. Are vulnerabilities discovered and resolved like other defects? *Empirical Software Engineering*, 23(3):1383–1421, 2018.

[5] G. Canfora, A. Di Sorbo, S. Forootani, A. Pirozzi, and C. A. Visaggio. Investigating the vulnerability fixing process in oss projects: Peculiarities and challenges. *Computers & Security*, 99:102067, 2020.

RQ

- ▶ RQ1：OSSプロジェクトにおいて、脆弱性の自己修正はどの程度行われているのか
 - ▶ 開発者が自分の書いたコードを一番理解しており、改善策を効果的に特定できる可能性
- ▶ RQ2：脆弱性の種類によって自己修正率は異なるか
 - ▶ 脆弱性の種類によって修正に専門的な知識やスキルを必要とする可能性
- ▶ RQ3：自己修正された脆弱性は、より早く修正可能か

データセット

- ▶ NVD[1] (National Vulnerability Database) からC言語とPHPのプロジェクトを選定
- ▶ NVDデータベースから2002年～2020年のCVE (共通脆弱性識別子) を収集
 - CVEに対応するCWE (共通脆弱性タイプ一覧) を使用
- ▶ Cプロジェクト, PHPプロジェクト合わせて65種類のCWE
 - 表 1 の12種類のみを考慮

TABLE I DISTRIBUTION OF THE SELECTED CWES OVER PROGRAMMING LANGUAGES			
Software Weakness	C project	PHP project	Definition
cwe-200	147	128	Exposure of Sensitive Information
cwe-264*	186	163	Permissions, Privileges, and Access Control
cwe-284*	173	202	Improper Access Control
cwe-20	325	70	Improper Input Validation
cwe-89	8	616	Sql injection
cwe-22	3	147	Improper Limitation of Pathname to a Restricted Directory
cwe-787	159	1	Out-of-bounds Write
cwe-476	148	-	NULL Pointer Dereference
cwe-119	724	-	improper restriction of operations within the bounds of a memory buffer
cwe-125	468	-	Out-of-bounds Read
cwe-189*	117	-	Numeric Errors
cwe-399*	171	-	Resource Management Errors
cwe-190	154	-	Integer Overflow or Wraparound
cwe-416	209	-	Use After Free
cwe-362*	93	-	Race Condition
cwe-79	-	583	Improper Neutralization of Input During Web Page Generation
cwe-352	-	136	Cross-Site Request Forgery (CSRF)
Total-Hunk	3085	2046	

[1]https://nvd.nist.gov/vuln/data-feeds#JSON_FEED

RQ1 (自己修正はどの程度行われているのか)

▶ CとPHPそれぞれで自己修正率を計算

$$\text{Self-fixing rate (pl)} = \frac{\text{num_of_self_fixed}_{pl}}{\text{num_of_fixed}_{pl}}$$

分子：自己修正ハンク数 分母：修正ハンク数

▶ 結果

- ▶ 自己修正率はC言語のプロジェクトでは20.55%, PHPのプロジェクトでは36.46%
- ▶ 開発者の知識・経験不足が原因と考えられる

TABLE II
DETAILS OF THE SELECTED CWES OVER C AND PHP PROGRAMMING
LANGUAGES

	C	PHP
Commits	1395	357
Files	1666	1058
Hunks	3085	2046
SLOC	18074	10591
Self-fixing Rate	20.55%	36.46%

RQ2 (脆弱性の種類によって自己修正率は異なるか)

▶ 総自己修正率と平均自己修正率を計算 (CWE毎)

▶ プロジェクトの言語によって修正率が異なるかの検証

$$\text{Total Self-fixing rate } (w) = \frac{\sum_{pl=1}^n \text{num_of_self_fixed}_{pl,w}}{\sum_{pl=1}^n \text{num_of_fixed}_{pl,w}}$$

分子 : 自己修正ハンク数の全てのプログラミング言語における合計
分母 : 修正ハンク数の全てのプログラミング言語における合計
n : プログラミング言語の数

$$\text{Average self-fixing rate}(w) = \frac{1}{n} \sum_{pl=1}^n \frac{\text{num_of_self_fixed}_{pl,w}}{\text{num_of_fixed}_{pl,w}}$$

▶ 結果

- ▶ 脆弱性の種類によって自己修正率が50%以上のものもあれば、10%前後のものもある
- ▶ 平均自己修正率と総自己修正率が大きく離れている脆弱性はない
→ プロジェクトの言語は自己修正率と大きく関係がない

RQ2

- ▶ 表 3 : CプロジェクトのCWE毎の自己修正率
- ▶ 表 4 : PHPプロジェクトのCWE毎の自己修正率
- ▶ 表 5 : 平均自己修正率と総自己修正率

TABLE III

THE NUMBER OF *self-fixed-hunk (SF-h)*, *non-self-fixed-hunk (NSF-h)*, AND OF *Self-fixing rates (SFR)* FOR INDIVIDUAL CWEs AMONG C PROJECTS

CWE	#SF-h	#NSF-h	SFR
cwe-190	84	70	0.55
cwe-89	3	5	0.37
cwe-22	1	2	0.33
cwe-476	46	102	0.31
cwe-20	96	229	0.30
cwe-787	45	114	0.28
cwe-119	156	568	0.21
cwe-189	20	97	0.17
cwe-284	27	146	0.15
cwe-264	26	160	0.14
cwe-200	21	126	0.14
cwe-125	58	410	0.12
cwe-399	21	150	0.12
cwe-362	10	83	0.10
cwe-416	20	189	0.09
Average			0.22
Median			0.17

TABLE IV

THE NUMBER OF *self-fixed-hunk (SF-h)*, *non-self-fixed-hunk (NSF-h)*, AND OF *self-fixing rates (SFR)* FOR INDIVIDUAL CWEs AMONG PHP PROJECTS

CWE	#SF-h	#NSF-h	SFR
cwe-89	357	259	0.57
cwe-352	75	61	0.55
cwe-200	62	66	0.48
cwe-264	74	89	0.45
cwe-79	123	460	0.21
cwe-20	11	59	0.15
cwe-284	29	173	0.14
cwe-22	15	132	0.10
Average			0.335
Median			0.332

TABLE V

Average *self-fixed rates (ASFR)* AND Total *Self-fixing rates (TSFR)* FOR COMMON CWEs BETWEEN C AND PHP PROJECTS

CWE	ASFR	TSFR
cwe-89	0.47	0.57
cwe-200	0.31	0.30
cwe-264	0.29	0.28
cwe-20	0.22	0.27
cwe-284	0.14	0.14
cwe-22	0.21	0.10

RQ3（自己修正された脆弱性は、より早く修正可能か）

▶ 自己修正/非自己修正された脆弱性についてCWEと言語ごとに生存時間の平均を計算

▶ 結果

TABLE X AVERAGE SURVIVAL TIME OF SELF-FIXED (AT-SF) AND OF NON-SELF-FIXED (AT-NSF) FOR BOTH C AND PHP LANGUAGES			
Language	AT-SF (days)	AT-NSF (days)	AT-NSF/AT-SF
C	941.56	1831.47	1.94
PHP	814.86	1429.02	1.75

▶ おおよそ自己修正された脆弱性の方が早期に解決される

▶ C, PHP共に自己修正された脆弱性は非自己修正されたものの半分程度の時間

RQ3

▶ 自己修正/非自己修正の脆弱性の解決時間に統計的な有意差があるか検証

▶ 仮設検定：U検定（p値は0.05）

▶ 効果量：Cliff's d （ $|d| < 0.33$ で小、 $0.33 \leq |d| < 0.474$ で中、 $|d| \geq 0.474$ で大）

▶ 結果

▶ 表VIIIとIXでは $p < 0.05$ のCWEのみ（簡略化のため）

▶ Cプロジェクトでは3つのCWEで効果量が多い

▶ PHPプロジェクトは1つ効果量が多い

TABLE VIII
THE RESULTS OF P-VALUE TOGETHER WITH CLIFF'S $|d|$ EFFECT SIZE ON
SURVIVAL TIME FOR SELF VS NON-SELF-FIXED VULNERABILITIES OF
CWEs IN C LANGUAGE

CWE	p-value	effect-size
CWE-190	3.476e-13	0.615 (large)
CWE-20	$< 2.2\text{e-}16$	0.675 (large)
CWE-264	7.639e-09	0.506 (large)
CWE-399	0.003	0.301 (small)
CWE-416	0.001	0.232 (small)
CWE-476	0.018	0.204 (small)

TABLE IX
THE RESULTS OF P-VALUE TOGETHER WITH CLIFF'S $|d|$ EFFECT SIZE ON
SURVIVAL TIME FOR SELF VS NON-SELF-FIXED VULNERABILITIES OF
CWEs IN PHP LANGUAGE

CWE	p-value	effect-size
CWE-352	4.651e-13	0.661 (large)
CWE-79	9.484e-15	0.378 (medium)
CWE-264	0.002	0.206 (small)
CWE-284	0.033	0.195 (small)
CWE-200	0.046	0.136 (negligible)
CWE-89	0.032	0.077 (negligible)

まとめ & 所感

▶ まとめ

- ▶ 自己修正された脆弱性はそうでないものよりも早く解決される可能性が高い
- ▶ 脆弱性の種類によっても修正速度は変化する
 - ・ 自己修正されていない脆弱性が、専門的な知識が必要な修正かどうかの議論はなされていない

▶ 所感

- ▶ 専門的な知識・スキルが必要か否かの議論がなかったので読んでいてスッキリしなかった
- ▶ データ収集の方法が複雑だった

補足

► CWEの内容

TABLE I
DISTRIBUTION OF THE SELECTED CWES OVER PROGRAMMING LANGUAGES

Software Weakness	C project	PHP project	Definition
cwe-200	147	128	Exposure of Sensitive Information
cwe-264*	186	163	Permissions, Privileges, and Access Control
cwe-284*	173	202	Improper Access Control
cwe-20	325	70	Improper Input Validation
cwe-89	8	616	Sql injection
cwe-22	3	147	Improper Limitation of Pathname to a Restricted Directory
cwe-787	159	1	Out-of-bounds Write
cwe-476	148	-	NULL Pointer Dereference
cwe-119	724	-	improper restriction of operations within the bounds of a memory buffer
cwe-125	468	-	Out-of-bounds Read
cwe-189*	117	-	Numeric Errors
cwe-399*	171	-	Resource Management Errors
cwe-190	154	-	Integer Overflow or Wraparound
cwe-416	209	-	Use After Free
cwe-362*	93	-	Race Condition
cwe-79	-	583	Improper Neutralization of Input During Web Page Generation
cwe-352	-	136	Cross-Site Request Forgery (CSRF)
Total-Hunk	3085	2046	