

The Extent of Orphan Vulnerabilities from Code Reuse in Open Source Software

David Reid, Mahmoud Jahanshahi, Audris Mockus,

ICSE 2022 Technical Track

“The Extent of Orphan Vulnerabilities from Code Reuse in Open Source Softwareudy”

発表者: 田中 智達

用語：オープン脆弱性(1/2)

- オープンソースコードの特徴として「新プロジェクトにコピーできる」ことが挙げられる。
- しかしこのようなコピーは脆弱性や他の問題をもたらす可能性がある^[15].
- コピーされたコードの含まれる脆弱性が、他のプロジェクトで発見・修正された場合でも、自身のプロジェクトに存在する脆弱性のことを「オープン脆弱性」と定義。

[15] Antonios Gkortzis, Daniel Feitosa, and Diomidis Spinellis. 2021.

Software reuse cuts both ways: An empirical analysis of its relationship with security vulnerabilities.

Journal of Systems and Software 172 (2021), 110653. <https://doi.org/10.1016/j.jss.2020.110653>

用語：オーファン脆弱性(2/2)

- 「オーファン脆弱性」による問題
 - 脆弱性の悪用が広く知られ、既知の脆弱性を持つソフトウェアを攻撃することが可能になる[6].
 - 脆弱性があるにも関わらず、脆弱性を知らない他のプロジェクトにコピーされる可能性がある.
 - 脆弱性があるにも関わらずアプリケーションに組み込まれ、疑いを持たないユーザーによって使用される可能性がある.
 - 仮に相応数のOSSプロジェクトが既知の未修正の脆弱性を含んでいた場合、OSS全体に対する風評被害が起こりうる.

[6] Mircea Cadariu, Eric Bouwers, Joost Visser, and Arie van Deursen. 2015.

Tracking known security vulnerabilities in proprietary software systems. In 2015 IEEE 22nd International Conference on Software Analysis, Evolution, and Reengineering (SANER). 516–519. <https://doi.org/10.1109/SANER.2015.7081868>

本研究の目的・アプローチ

- OSSのコードをコピーできるようになったことで、実際にオーファン脆弱性が蔓延しているかどうかを確認し、ソフトウェアのセキュリティ脆弱性を低減すること.
- オーファン脆弱性を持つプロジェクトを特定するために構築されたツールVDiOSを使用し、libpng^[35], OpenSSL, xz圧縮の中から4件の脆弱性を調査.

[35] Greg Roelofs. 2006. libpng.org. Retrieved August 4, 2020 from <http://www.libpng.org>

背景：ソフトウェアの再利用

- ソフトウェアの再利用には2つある.
 - ブラックボックス再利用
 - 自身では変更出来ない外部コードを利用
 - Pythonのライブラリなどが該当する.
 - ホワイトボックス再利用
 - 元のコードをコピーしたものを自身のリポジトリ内にて利用
 - 問題の例
 - コピー元が脆弱性を発見し修正した場合でも、その修正が全てにプロジェクトに適応されるわけではない.
 - 本研究ではこちらに注目している.

背景：World of Code (WoC)

- 主要なソースコード・リポジトリ・ホスティング・プラットフォームのすべてにおいて、再利用されたソースコード、ファイルのすべてのバージョンを効率的に見つけるインフラストラクチャを提供.
- これによりあらゆる言語のファイルレベルのコード重複を見つけることが可能.

研究方法

- 既知のソフトウェアセキュリティ脆弱性のうち、元のプロジェクトのリポジトリで修正された4つの具体的な事例を詳細に調査.
- 結果の信頼性を高め、一般性の向上のため複数の脆弱性のケースを使用.
 - ユニークなケースやエッジケースは除く.
 - ケースのスクリーニングにはVDiOSを使用.

データセット

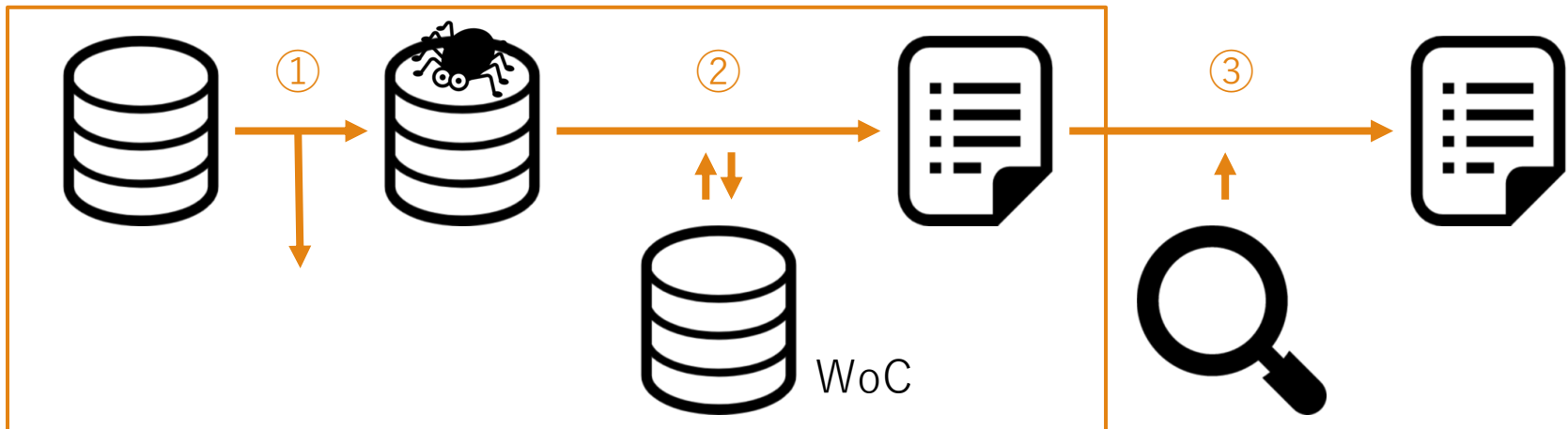
- libpng
 - 長期間にわたってコード内に存在し，その間に多くのコピーが可能であったものを選択.
- OpenSSL3
 - 脆弱性の発生時期の違いを明確にするため，新しい脆弱性と古い脆弱性を選択.
- xz圧縮
 - C言語ではない言語(Go言語)で書かれているため.
 - OSSにおける既知の脆弱性を代表しているものを選択.

VDiOSツール

- コードの再利用を発見し、すべてのリポジトリとバージョン履歴にわたって、単一のファイルのバージョンを追跡するツール.
- WoCのインフラストラクチャをベースに、膨大な量の重複ファイルの発見を実現.
- 脆弱性を含むリビジョンと含まないリビジョンとで分別する機能がある.
 - 脆弱性があるプロジェクト
 - 以前は存在したが修正されたプロジェクト
 - 以前は存在したが変更され、脆弱性が修正されたか不明なプロジェクト

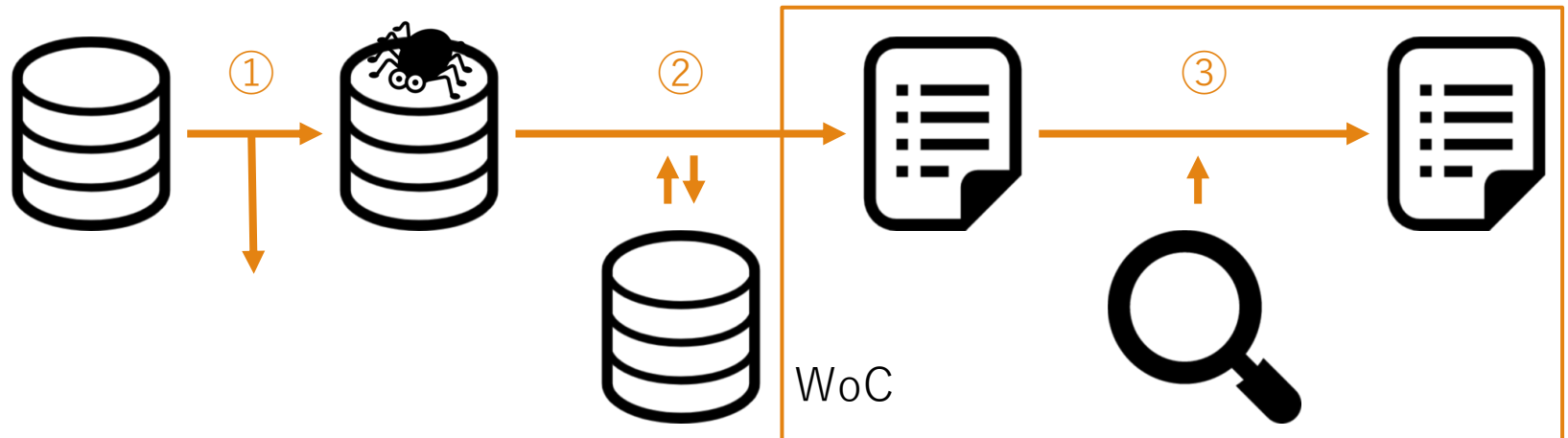
研究手順(1/3)

- ① 脆弱性を含むBLOBと、修正を含むBLOBを特定.
- ② 脆弱なBLOBのいずれかを重複して含むWoC内の全プロジェクトを検索し、過去に脆弱なバージョンのファイルを含んだことのあるプロジェクトをリスト化.



研究手順(2/3)

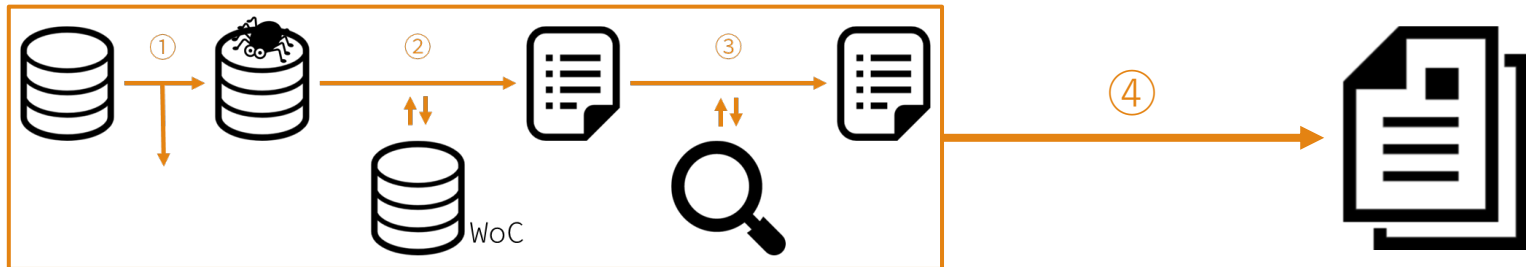
- ③ リスト化したBLOBそれぞれが最新のリビジョンか
チェック, カテゴリ分け.
- 一致した場合
 - 脆弱性のあるコードが含まれたプロジェクト. (脆弱性あり)
 - 一致していない場合
 - 既知の正常なBLOBと一致. (修正済み, 脆弱性なし)
 - 既知の正常なBLOBと不一致. (修正されたか不透明)



研究手順(3/3)

④ ブラウザで閲覧できるようHTML形式のレポートとして出力.

- 脆弱性を修正したコミット(該当する場合)
- 修正されたBLOB
- そのファイル名のリストへのリンク
- その祖先と子孫へのリンク
- 3カテゴリのプロジェクトリストへのリンク



結果(1/3)

- OpenSSLの脆弱性「CVE-2021-3449」^[43]
 - 2021年3月に修正され，56のリビジョンが存在.
- OpenSSLの脆弱性「CVE-2014-0160」^[39]
 - 2014年に修正された非常に深刻な脆弱性^[45]

	CVE-2021-3449	CVE-2014-0160
脆弱性あり	1614	121
修正済み	11	3156
不明	1079	211
削除済み	253	

結果(1/3)

- OpenSSLの脆弱性「CVE-2021-3449」^[43]
 - 1614プロジェクトが最新リリースに脆弱性のあるリリースを含んでいる。(脆弱性あり)
 - そのうち4件はスター数が10000個以上、10件は1000個以上あり、OSSコミュニティ内で大きな影響力を持っているといえるものであった。
- OpenSSLの脆弱性「CVE-2014-0160」^[39]
 - 121プロジェクトが最新リリースに脆弱性のあるリリースを含んでいる。(脆弱性あり)
 - そのうち110件は2014年時には活動が停止していた。
 - その他11件も詳細に調査した結果を含め、事実上脆弱性は排除されたといえる。

結果(2/3)

●xz圧縮の脆弱性「CVE-2021-29482」 [42]

	脆弱性を含む			脆弱性を含まない		
総計	7105			185		
フォークされていない	758			82		
最後のコミット<6ヶ月	271	35.75%		68	82.93%	
最後のコミット<18ヶ月	472	62.27%		82	100.00%	
スター数>1	443	58.44%		71	86.59%	
スター数>10	273	36.02%		44	53.66%	
スター数>100	101	13.32%		23	28.05%	
スター数>1000	31	4.09%		10	12.20%	
スター数>10000	10	1.32%		4	4.88%	
貢献者数>1	531	70.05%		77	93.90%	
貢献者数>10	254	33.51%		48	58.54%	
貢献者数>100	55	7.26%		17	20.73%	
貢献者数>1000	10	1.32%		6	7.32%	

表：プロジェクト数とフォークされていないプロジェクトからの割合

結果(2/3)

- 脆弱なプロジェクトのアクティブなプロジェクトのは、脆弱でないプロジェクトに比べて著しく低い。
 - しかしある期間にコミットがないことは他のプロジェクトが使われていないことを意味しないので、やはり脆弱性の問題への対処は重要である。
 - これはすでにこの脆弱性が広く普及していることを示している。
- 脆弱性を修正したプロジェクトの星の数は脆弱性のあるプロジェクトよりも相対的に多い。
- また脆弱なプロジェクトには比較的少数の開発者が関与している。

結果(3/3)

●libpngの脆弱性「CVE-2017-12652」 [40]

	脆弱性を含む			脆弱性を含まない		
総計	63441			458		
フォークされていない	9680			51		
最後のコミット<6ヶ月	660	6.82%		20	39.22%	
最後のコミット<18ヶ月	2095	21.64%		29	56.86%	
スター数>1	2646	27.33%		34	66.67%	
スター数>10	1089	11.25%		24	47.06%	
スター数>100	395	4.08%		15	29.41%	
スター数>1000	131	1.35%		10	19.61%	
スター数>10000	25	0.26%		2	3.92%	
貢献者数>1	4317	44.60%		45	88.24%	
貢献者数>10	632	6.53%		27	52.94%	
貢献者数>100	173	1.79%		11	21.57%	
貢献者数>1000	44	0.45%		5	9.80%	

表：プロジェクト数とフォークされていないプロジェクトからの割合

結果(3/3)

- 63441プロジェクトが最新リビジョンに脆弱性のあるリビジョンを含んでいる。(脆弱性あり)
 - そのうち特定条件下で選出した82件について、修正パッチを作成し、プルリクエスト・電子メール・Issueを通じて報告し、2週間ほど待機。
 - プルリクエストを受け入れ：7件
 - libpngの新しいバージョンにアップデート：1件
 - 脆弱性のある既存のコードを使い続けると報告：2件
 - 反応なし：72件

結論

- 全体として、修正に数年かかる様な脆弱なコードは広範囲に拡散し、活動休止中のプロジェクトだけでなく、非常に人気で活発なプロジェクトにも影響を与えている。
- 多くのプロジェクトは、修正プログラムを提供されていてもパッチを適応しようとししない。
- 以下の3種のサポートがOSSにおける未修正の脆弱性への対処に有効。
 - 修正パッチの提供
 - 潜在的な利用者への情報提供
 - 開発者への情報提供、修正パッチの提案

所感

- 論文内の図表が少なく(3こ), 情報の整理に非常に時間がかかった.
- その分かなり情報量が多く, まとめるのにも非常に時間がかかった.