

On the Relation between Outdated Docker Containers, Severity Vulnerabilities, and Bugs

Ahmed Zerouali, Tom Mens, Gregorio Robles,
Jesus M. Gonzalez-Barahona

古い Docker コンテナ, 重大な脆弱性
バグの関係について

発表者: 辻 裕真

IEEE SANER 2019

背景

- ・コンテナイメージにはソフトウェアパッケージのコレクションが含まれており、そこにはコンテナに影響するバグやセキュリティに関する脆弱性が含まれている場合がある。
- ・この研究の目的は、古くなったコンテナと、そこにインストールされた脆弱でバグのあるパッケージとの関係进行分析することでコンテナのデプロイをサポートすることである。

Research Questions

RQ0:Dockerイメージはどのくらいの頻度で更新されるか.

RQ1:コンテナのパッケージはどれほど古くなっているか.

RQ2:コンテナのパッケージはどの程度脆弱であるか.

RQ3:コンテナはどの程度パッケージによるバグに苦しんでいるか.

RQ4:バグとセキュリティに関する脆弱性はどのくらいの期間修正されないままであるか.

手法とデータセット

- ・ Docker HubからDockerイメージを取得し(2018年10月), インストールされているパッケージを特定, それらのパッケージの技術的遅れ(the technical lag)を調べることで, イメージの技術的遅れを計算する.
- ・ サンプルはDebianベースのDocker Hubの全ての公式イメージと, プル数が上位のイメージ.

Debianをベースとするイメージ

- ・Debianプロジェクトは複数のパッケージが同時リリースされており、重要であるのはテスト版(Testing), 安定版(Stable), 旧安定版(Oldstable)である.

TABLE I GENERAL INFORMATION ABOUT THE ANALYZED <i>Debian</i> VERSIONS.			
Version name	Version	Distribution	Release date as stable
<i>Buster</i>	<i>Debian 10</i>	Testing	-
<i>Stretch</i>	<i>Debian 9</i>	Stable	2017-06-17
<i>Jessie</i>	<i>Debian 8</i>	Oldstable	2013-04-25

分析するイメージの特定

- ・イメージはレイヤーと呼ばれる1つまたは複数の中間イメージで構成されており、各レイヤーは一意のハッシュ署名を持っている。
- ・DebianをベースとするイメージにはDebianのレイヤーが含まれているので、それを利用してDebianベースイメージから派生したイメージを識別する。

分析するイメージの特定

- Docker Hub APIを利用して、124の公式リポジトリから全ての利用可能なイメージ名を抽出、上位30,000のコミュニティリポジトリから500以上のプルを持つものを抽出した.
- skopeoツールを用いて、それらすべてのイメージを検査し一意のイメージを特定し、Debianベースイメージのレイヤーを含むイメージを特定.

分析するイメージの特定

・公式リポジトリの14,653のイメージから、Debianベースの2,453のイメージが、30,000のコミュニティリポジトリからDebianベースの4,927のイメージが見つかった。

TABLE II
NUMBER OF *Docker* IMAGES PER *Debian* DISTRIBUTION.

Containers	<i>Buster</i> / Testing	<i>Stretch</i> / Stable	<i>Jessie</i> / Oldstable
Official	150	620	1,683
Community	86	1,248	3,593
Total	236	1,868	5,276

脆弱性, バグのレポート

- ・データセット内の既知のセキュリティ脆弱性を見つけるため, Debian Security Bug Trackerを利用(2018年3月18日時点).
- ・バグレポートにはUltimate Debian Databaseを使用して既知のバグを見つけた.

実証分析の結果:RQ0

RQ0:Dockerイメージはどのくらいの頻度で更新されるか.

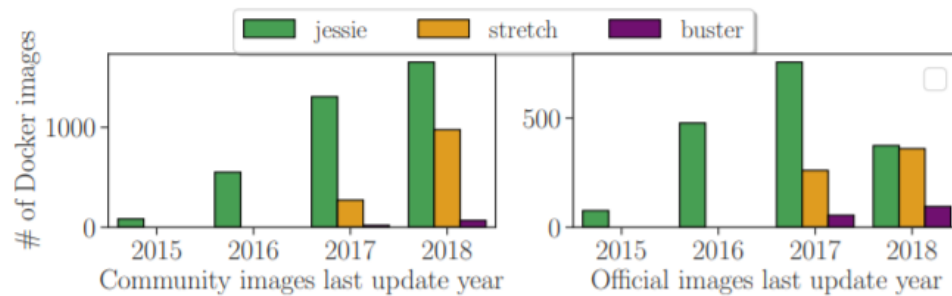


Fig. 2. Year of *Docker* images last update, grouped by *Debian* distribution and community type (community or official)

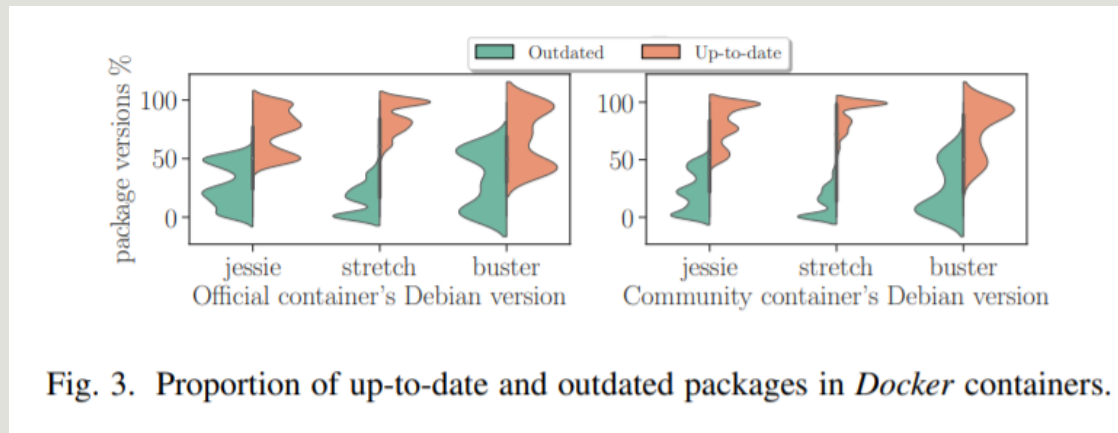
公式イメージの旧安定版が2018年に更新されたイメージ数は2017年に更新されたものより少ないが、コミュニティイメージの場合は逆である。

また、重要な結果としてコミュニティイメージの48%, 公式イメージの66%が最後に更新されたのが2017年以前であった。

実証分析の結果:RQ1

RQ1:コンテナのパッケージはどれほど古くなっているか.

- ・コンテナ内の最新, 古いパッケージの数を調べる.



図からバージョンに関係なく、ほとんどのパッケージが最新であることが分かる.コンテナごとの最新パッケージの割合の中央値は82%である.

実証分析の結果:RQ1

- ・分析を最新 (2018年に更新された)のイメージに限定すると、公式コンテナ内のパッケージは、コミュニティのものよりわずかに最新であった.
- ・コミュニティイメージは、公式イメージをベースにしているため、公式イメージからデプロイヤーが最新の更新イメージを使用する傾向があることを意味している.

実証分析の結果:RQ1

- ・古くなったパッケージを定量化するため, それらの技術的遅れを計算する.
- ・コンテナにインストールされている古いパッケージが使用しているバージョンを調べ, 最新バージョンから遅れている分が技術的遅れとなる.

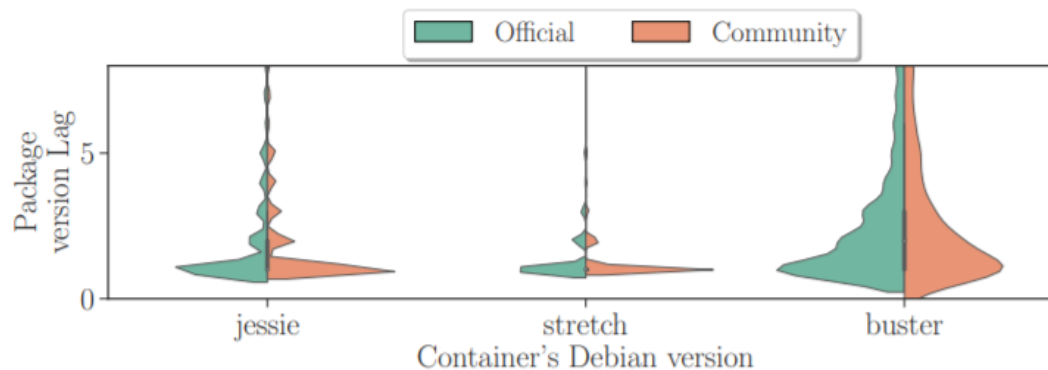


Fig. 4. Version lag induced by outdated packages in *Docker* containers.

一般的にコンテナ内のパッケージは最新であるか, 中央値を見ると1から2バージョン遅れていると結論付けられる.

実証分析の結果:RQ1

調査結果:

- ・コンテナにインストールされているパッケージのうち約20%が古いものである.
- ・コミュニティコンテナのユーザは最近更新された公式イメージを使用する傾向がある.
- ・古いパッケージは1から2バージョン遅れている.

実証分析の結果:RQ2

RQ2:コンテナのパッケージはどの程度脆弱であるか.

インストールされた全てのパッケージのうち,
12.2%(3,975中488)がセキュリティの問題を抱えていた.

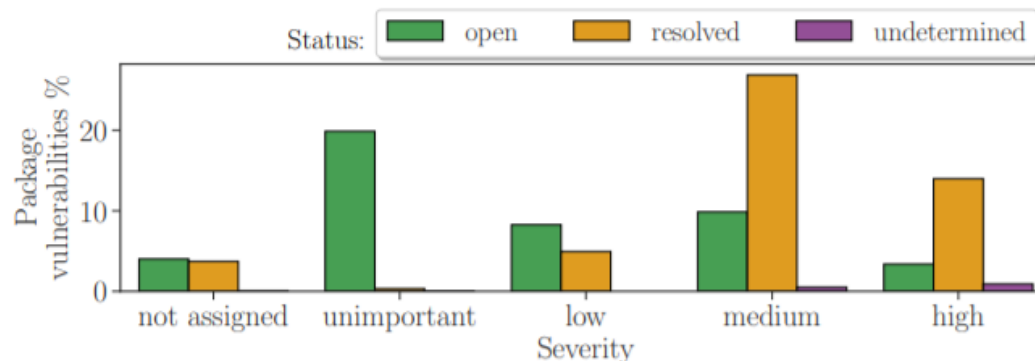


Fig. 5. Proportion of found vulnerabilities, grouped by severity and status.

図は脆弱性の分布を重大度および状態を示している.

全ての脆弱性の49.9%が解決されている.

全てのコンテナの96%が全ての種類の脆弱性を持っていることが分かった

実証分析の結果:RQ2

- ・次に, コンテナごとの脆弱性の数を調査した.

TABLE IV
MINIMUM, MEDIAN AND MAXIMUM NUMBER OF VULNERABILITIES PER
CONTAINER, GROUPED BY *Debian* VERSION AND CONTAINER TYPE.

Containers Debian	Official			Community		
	min	median	max	min	median	max
<i>Jessie</i>	155	658	7,106	155	916	7,338
<i>Stretch</i>	85	242	3,498	75	336	4,729
<i>Buster</i>	34	134	659	41	183	1,035

全体で, 平均
1,336, 中央値
601の脆弱性が
見つかった.
分布は大きく歪
んでおり, 7,338
もの脆弱性を持
つコンテナも見
つかった.

実証分析の結果:RQ2

- ・古いパッケージとコンテナの脆弱性の関係を調べるために、それらの数を比較した.

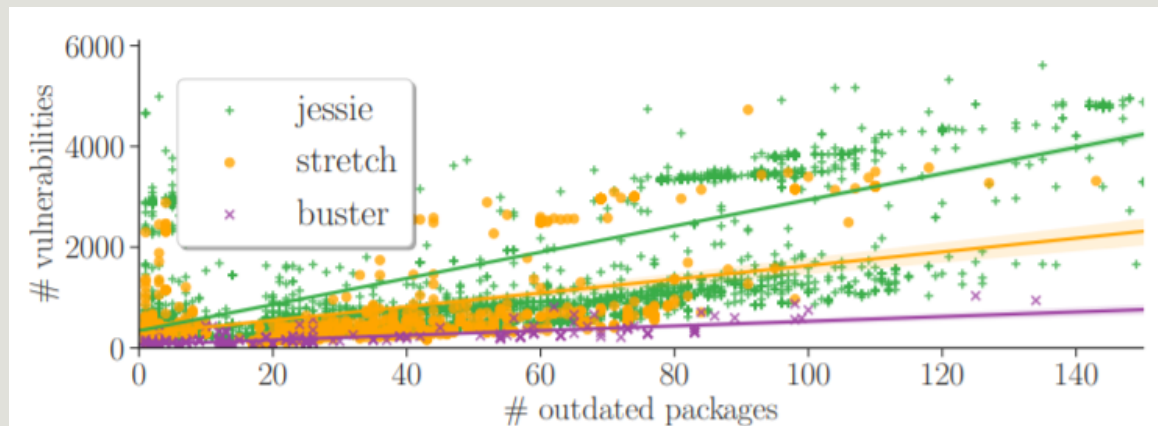


Fig. 6. Number of outdated packages and vulnerabilities per container.

重大度や状態で脆弱性を考慮せず、バージョンごとに脆弱性の数をプロットしている.

特に, Jessie(旧安定版)の場合, 古いパッケージが増えると脆弱性も増える傾向にある.

実証分析の結果:RQ2

調査結果:

- ・ 脆弱性のほぼ半分は修正されていない.
- ・ 全てのコンテナには重大な脆弱性がある.
- ・ 脆弱性の数は使用されているバージョンによって異なる.
- ・ 最も脆弱なコンテナは2年以上更新されていない.

実証分析の結果:RQ3

RQ3:コンテナはどの程度パッケージによるバグに苦しんでいるか.

全てのパッケージのうち50.1%(3,975中1,994)にバグがあることが分かった.

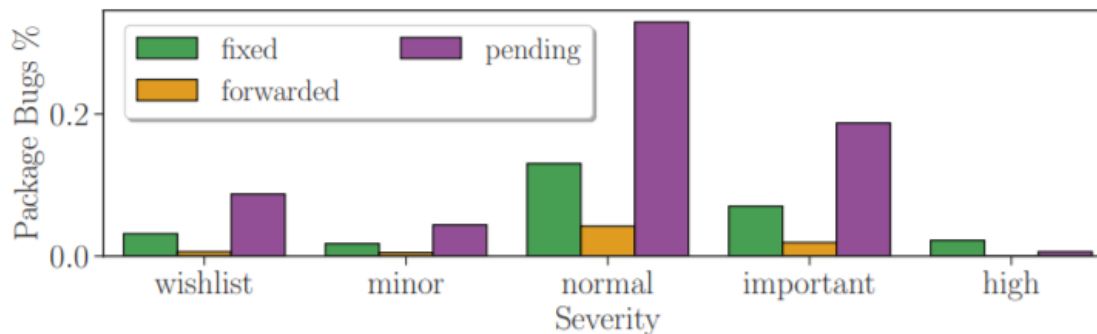


Fig. 7. Proportion of bugs grouped by severity and status.

図は、状態および重大度ごとにバグの分布を示している.

全てのバグの65.5%がまだ保留中である.

実証分析の結果:RQ3

- ・古いパッケージとコンテナ内のバグの関係を調査した.

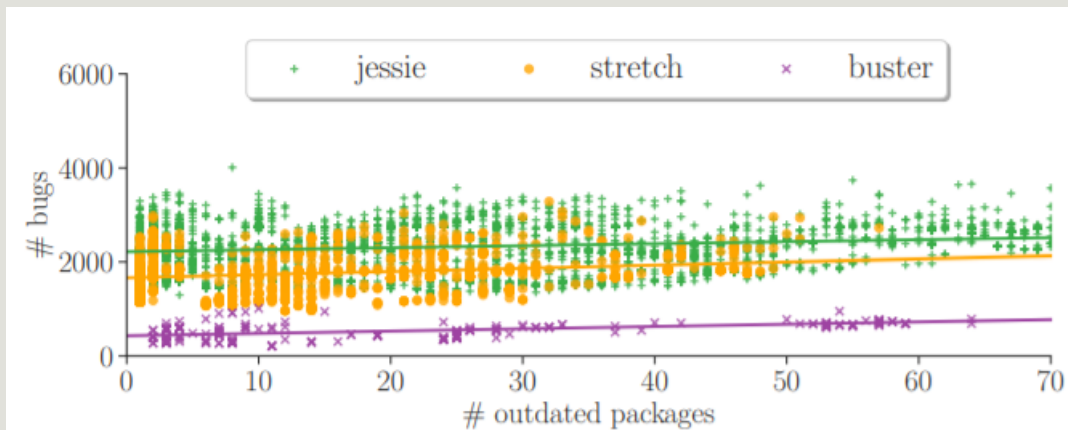


Fig. 8. Number of outdated packages and bugs per container.

図は, バージョンごとに古いパッケージの数と各コンテナ内のバグの数を示している.

Buster(テスト版)には中程度の相関がみられた.

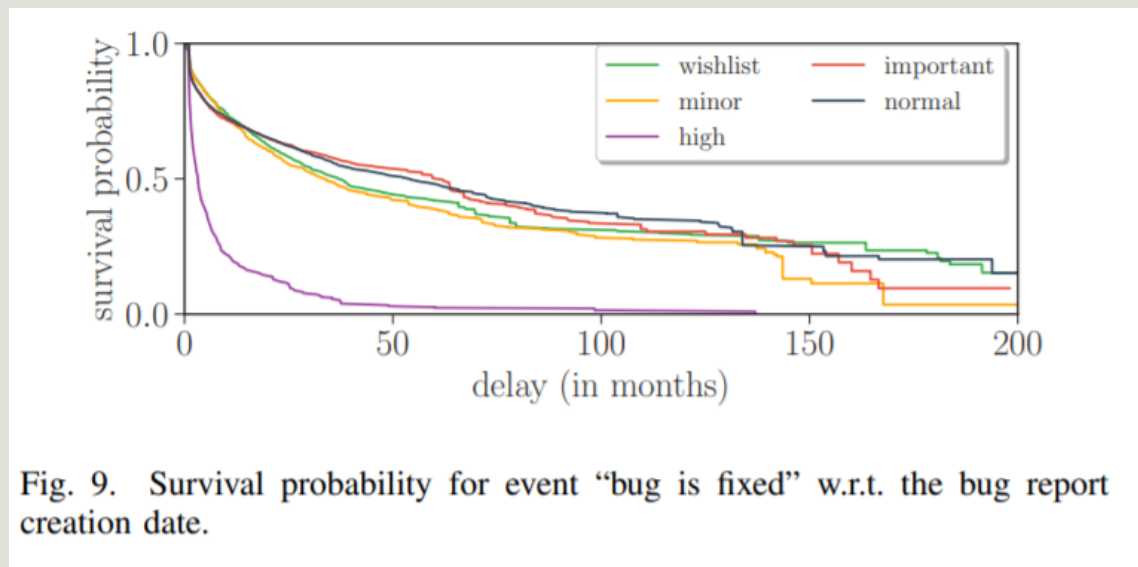
実証分析の結果:RQ3

調査結果:

- ・ 全てのコンテナにはバグのあるパッケージがある.
- ・ インストールされたパッケージのバグの65%は修正されていない.
- ・ バグの数は使用されているバージョンによって異なる.

実証分析の結果:RQ4

RQ4:バグとセキュリティに関する脆弱性はどのくらいの期間修正されないままであるか.



図はバグが修正されるまでの期間と生存確率を示している.

通常のバグ, マイナーなバグのそれぞれが50%が修正されるまで53.8ヶ月, と33.5ヶ月かかるが, 重大なバグでは3ヶ月しかかからない.

実証分析の結果:RQ4

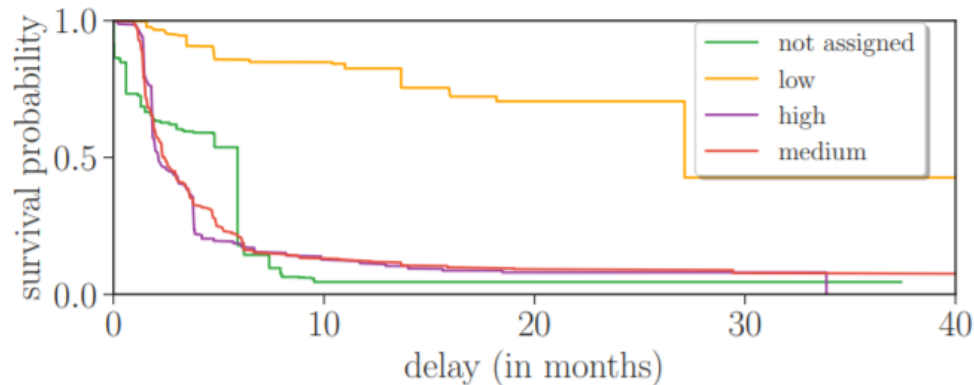


Fig. 10. Survival probability per severity level for event “security vulnerability is fixed” w.r.t. the date of the bug arrival.

図は、セキュリティに関する脆弱性が修正されるまでの期間と生存確率を示している。

脆弱性は他の種類のバグよりも早く修正されることが分かる。

修正期間は、中程度の重大度のものには2.4ヶ月、こう重大度のものには2.1ヶ月かかっている。

議論と実用的な結果

- ・バグを少なくすることに関心があれば、安定性が低下する代わりにテスト版を利用することが推奨される.
- ・メンテナーにとってセキュリティに関するバグを解決することは最優先事項であり、それらのバグを回避するために技術的遅れを最小限に抑える必要がある.

結論

- ・全てのコンテナに脆弱でバグのあるパッケージが存在していた.
- ・バグを処理するよりもセキュリティに関する脆弱性を処理することの方が重要視されているため、最新のパッケージをインストールしても未解決のバグの影響を受ける可能性がある.

所感

- ・それほど重大でないバグは解決されるまでに相当な期間がかかることが分かった.
- ・開発者は、最新の機能やバグが修正されたパッケージよりも、実際の環境で長くテストされて安定している古いパッケージを好むことが興味深かった.