5K-01

ソフトウェアエコシステムにおける SemVer 準拠と脆弱性の関係

川口　友也†　　村上　晴美†

大阪公立大学大学院情報学研究科†

1. はじめに

ソフトウェアエコシステムは数多くのパッケージ依存関係によって構成され，これが「依存性地獄」と呼ばれる問題を生じさせている．この問題に対処するために，Semantic Versioning（SemVer）が提案されているが，SemVer への準拠が不完全であるためその効果は限られている．また，パッケージの脆弱性も問題であり，本研究ではソフトウェアエコシステム内のパッケージにおける SemVer の準拠実態と脆弱性の関係を明らかにすることを目的とする．

2. 関連研究

2.1 Semver の準拠実態

SemVer はソフトウェアエコシステムの安定性を維持するための重要なルールである．不適切な SemVer の適用は脆弱性やバグのリスクを高め，互換性の問題を引き起こす可能性がある．Decan ら[1]による調査では，npm, RubyGems, Cargo, Packagist の SemVer 準拠状況が分析されているが，パッケージレベルでの詳細な分析は不足している．本研究では，パッケージレベルでの SemVer の準拠実態を分析する．

2.2 SemVer と脆弱性の関係

脆弱性はプログラムの不具合や設計ミスに起因するセキュリティ上の欠陥で，依存関係を通じて広がることがあるため，迅速な情報取得と修正が重要である．Zerouali ら[2]の研究では，npm と RubyGems の依存関係ネットワークにおける脆弱性の影響が分析されているが，SemVer との関連については検討されていない．本研究では，SemVer の準拠と脆弱性の関係を分析する．

3. データセット

本研究では，npm と RubyGems を分析対象とし，Libraries.io の公開データセットと GitHub Advisory Database の脆弱性情報を使用する．データセットには，npm での依存関係数約 1 億 5,000 万，パッケージ数約 128 万，脆弱性レポート約 650 件，RubyGems での依存関係数約 570 万，パッケージ数約 16 万，脆弱性レポート約 300 件が含まれる．

4. SemVer 準拠カテゴリ

SemVer の準拠カテゴリとして，先行研究[1]に基づき，Compliant（SemVer に準拠），Restrictive（より厳格），Permissive（より寛容）を用いる．Major バージョンが 1 以上である本番開発リリースでは，「1.x.x」のように Minor バージョン及び Patch バージョンが最新であるものを SemVer 準拠，つまり Compliant とする．これより厳格なものは Restrictive，寛容なものは Permissive と呼ぶ．対して Major バージョンが 0 である初期開発リリースでは，「0.1.2」のように，Major，Minor，Patch の全バージョンを固定したものを Compliant（SemVer に準拠）とし，これより寛容なものを Permissive とする．固定的な制約より厳格なものは定義上存在しないため，初期開発リリースでは Restrictive カテゴリは定義されない．

5. 分析

本研究では，4 つの Research Questions（RQ）を設定し，4 節で定義した SemVer 準拠カテゴリに基づいて分析する．

5.1 パッケージが「宣言する」依存関係制約はどの程度寛容か？（RQ1）

パッケージが宣言する依存関係制約の寛容さについて，パッケージ内の依存関係宣言の各 SemVer 準拠カテゴリの割合を 2020 年 1 月の最新リリースパッケージで分析した．この結果，npm と RubyGems の両エコシステムで，パッケージ内の宣言が同じカテゴリに統一される傾向が見られた．これはパッケージ開発者が自身の考えに基づき寛容さを決定していることを示唆している．また本番開発リリースで RubyGems は npm よりも寛容な依存関係を宣言する傾向があることもわかった．

5.2 パッケージが「受ける」依存関係制約はどの程度寛容か？（RQ2）

パッケージが受ける依存関係制約の寛容さについて，その割合を分析した．この結果，本番開発リリースの Restrictive と初期開発リリースの Compliant，及び本番開発リリースの Compliant と初期開発リリースの Permissive の分布が似ていることがわかった．これは，パッケージ開発者が初期開発と本番開発の区別なく寛容さを決定し

Relationship Between Semver Compliance and Vulnerabilities in Software Ecosystems

†Tomoya Kawaguchi , Harumi Murakami，Graduate School of Informatics，Osaka Metropolitan University

ていることを示唆している．

5.3 依存関係制約の寛容さによって脆弱性の影響はどれくらい変わるか？（RQ3）

依存関係制約の寛容さが脆弱性の影響にどのように関係するかを，直接的依存関係による脆弱性の影響を受けたパッケージを対象に分析した．SemVer 準拠カテゴリごとに脆弱性の影響を受けた期間を集計し，生存分析（Kaplan-Meier 法）を実施した．結果を図 1, 2（X 軸が経過時間，Y 軸が脆弱性の生存確率）に示す．本番開発リリースでは Restrictive と Compliant の生存確率に大きな差があるが Compliant と Permissive の差は小さい．脆弱性を減じるために Restrictive から Compliant に変更することの有効性が示唆される．対して初期開発リリースでは，Compliant と Permissive の差が大きい．脆弱性を減じるために Compliant から Permissive に変更することが効果的であること，言い換えると SemVer への準拠が脆弱性を高めることが示される．

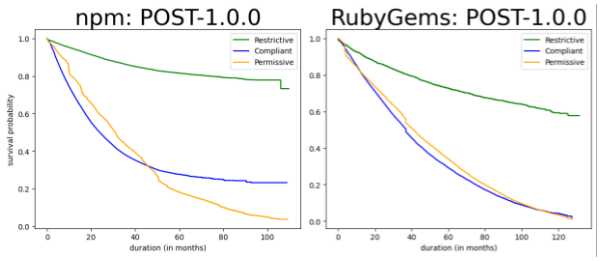

図 1　本番開発リリースの生存分析

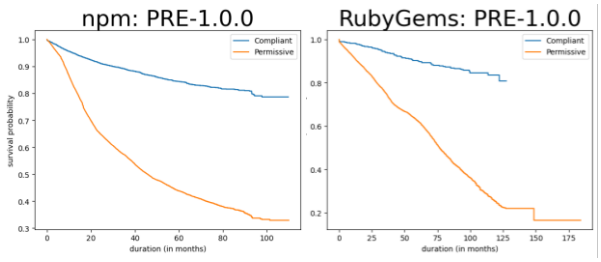

図 2　初期開発リリースの生存分析

5.4 依存関係制約の寛容さによって脆弱性はどの程度「自動解消」されるか？（Q4）

データセット内の全依存関係において，準拠カテゴリごとに脆弱性修正を自動で取り込めた割合を集計した．結果を図 3, 4 に示す．本番開発リリースでは Restrictive と Compliant の自動解消率に大きな差があり，Compliant と Permissive の差が小さいことを確認した．初期開発リリースでは Compliant と Permissive に大きな差が見られた．これらは 5.3 節の結果と一致する．

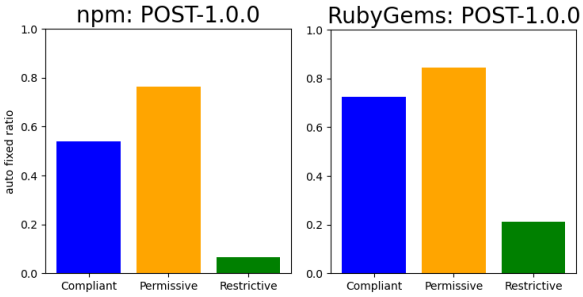

図 3　本番開発リリースの脆弱性自動解消率

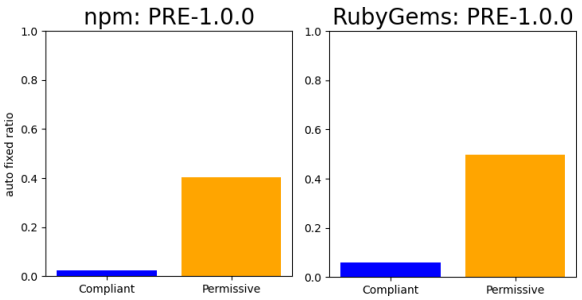

図 4　初期開発リリースの脆弱性自動解消率

6. おわりに

npm と RubyGems を対象として，SemVer の準拠実態と脆弱性の関係の分析を行った．パッケージ開発者が寛容さを統一して依存関係を宣言する傾向があることを明らかにした．また，SemVer への準拠が本番開発リリースにおいては脆弱性の軽減に効果があるが，初期開発リリースでは効果がないことを確認した．これらの結果から，実態と照らして初期開発リリースの SemVer 規約が厳しすぎる可能性があることが示唆された．

また，依存関係の管理において，寛容さには変更の自動取り込みと互換性維持がトレードオフの関係になっているが，本研究では互換性維持の観点からの分析を行っていないため，この点について，例えばパッケージのソースコードレベルでの詳細な解析を行う必要がある．

本研究では npm と RubyGems の分析を行ったが，今後は Cargo, Packagist など，異なるエコシステムを対象に分析を行う必要がある．

参考文献

[1] Alexandre Decan, Tom Mens: What Do Package Dependencies Tell Us About Semantic Versioning?, IEEE Transactions on Software Engineering, Vol. 47, Issue 6, pp. 1226-1240 (2021)

[2] Ahmed Zerouali, Tom Mens, Alexandre Decan, Coen De Roover: On the impact of security vulnerabilities in the npm and RubyGems dependency networks, Empirical Software Engineering, Article number: 107 (2022)