GitHub

NEWS RELEASE

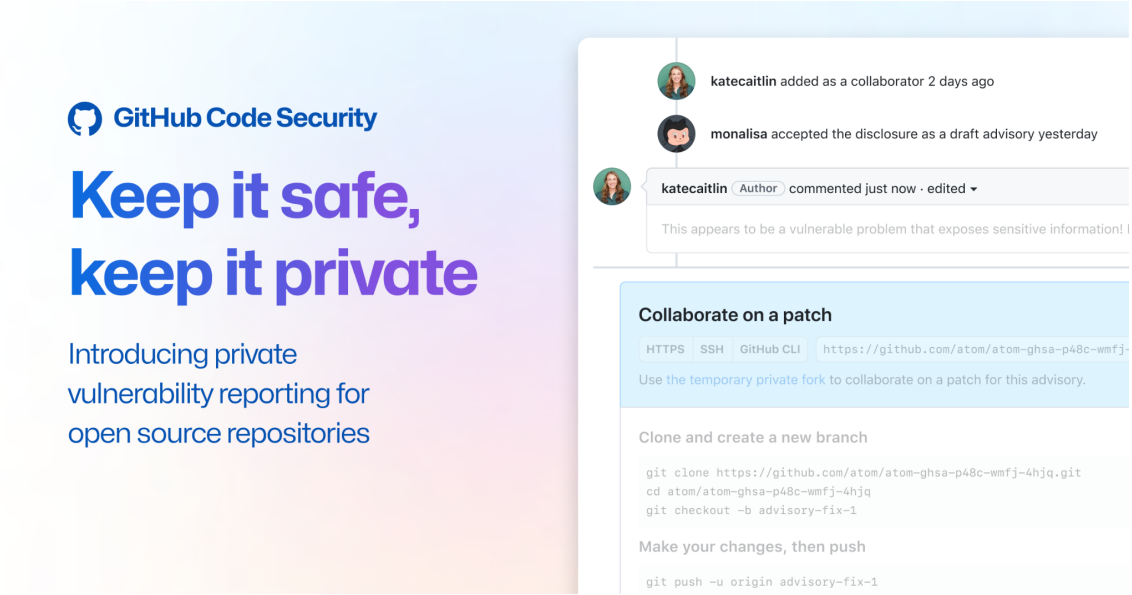
2023年4月26日
ギットハブ・ジャパン合同会社

GitHub、セキュリティ関連のアップデートを発表

～プライベート脆弱性レポート機能の一般提供（GA）開始／npmパッケージプロベナンスを導入～

オープンソースプロジェクトおよびビジネスユースを含む、ソフトウェアの開発プラットフォームを提供するGitHub, Inc.（本社：米国サンフランシスコ）は、2023年4月19日(米国時間)にセキュリティ関連のアップデートとして、プライベート脆弱性レポート機能の一般提供（GA）および、npmパッケージプロベナンスの導入を発表しました。

プライベート脆弱性レポート機能の一般提供（GA）を開始



4月19日にGitHubは、プライベートコラボレーションチャネルであるPrivate vulnerability reporting機能の一般提供を開始することを発表しました。本機能により、セキュリティの研究者やオープンソースのメンテナーが、パブリックリポジトリ上の脆弱性をより簡単に報告・修正できるようになりました。GitHubスターであり、GitHubセキュリティアンバサダーかつOpen Source Security Foundation（OpenSSF）Project Alpha-OmegaのシニアオープンソースセキュリティリサーチャーであるJonathan Leitschuh氏は、「リサーチャーとして最大の苦労の1つは、脆弱性を開示するための最初の連絡をメンテナーに取ることでした。Private vulnerability reportingは大きな前進です」と述べています。

GitHub Universe 2022にて、このような問題に対する解決策をテストし、メンテナーやセキュリティ研究者からのフィードバックを得るために、Private vulnerability reportingのパブリックベータ版を発表しました。以来、30,000社以上の企業のメンテナーが180,000以上のリポジトリにおいて本機能を有効化し、セキュリティ研究者から1,000件以上のフィードバックを得ています。

メンテナーにとってのメリット
毎週ダウンロード数が6,000万を超えるJSON5は、npmで最も依存されているパッケージの上位0.1%に入っており、Chromium、Next.js、Babel、Retool、WebStorm、およびその他の主要プロジェクトで採用されています。JSONファイル形式はマシン間通信でよく使われていますが、拡張されたJSON5を使うことで、開発者の手動によるコーディングとメンテナンスが容易になるため、JSON5がこれほど普及していると考えられます。

侵入テストのエキスパートであるJonathan Gregson氏がJSON5の脆弱性を発見した際、発見当初はGitHubのIssueを介してJSON5のメンテナーであるJordan Tucker氏と連絡を取り、サブミッションを調整しようとしましたが、そこから事態が複雑化しました。Tucker氏は、扱いにくいメールスレッドを使わず、パブリックな場での議論を避けたいと考えました。「最初は他のベンダーにも脆弱性を提出してもらおうとしたのですが、返事が来ることはありませんでした」そこで、代替手段を探したTucker氏はGitHubのPrivate vulnerability reporting機能のパブリックベータ版にたどり着きました。「自分のリポジトリでこの機能を有効化し、Gregson氏にGitHubで報告するように依頼しました。そこからは、すべてが迅速に問題なく進みました」

その結果、作成された修正プログラム（CVE）により、1,100万件以上のアラートがトリガーされました。これは、JSON5の人気と、メンテナーやセキュリティ研究者がオープンソースプロジェクトの健全性と安全性を保つベストプラクティスとして、Private vulnerability reportingの価値を証明するものです。

> Private vulnerability reportingは、オープンソースコミュニティが脆弱性を報告・修正することを非常に容易にします。すべてのメンテナーに対して、公開リポジトリで本機能を有効化することをお勧めします。
> - Jordan Tucker/ JSON5のメンテナー

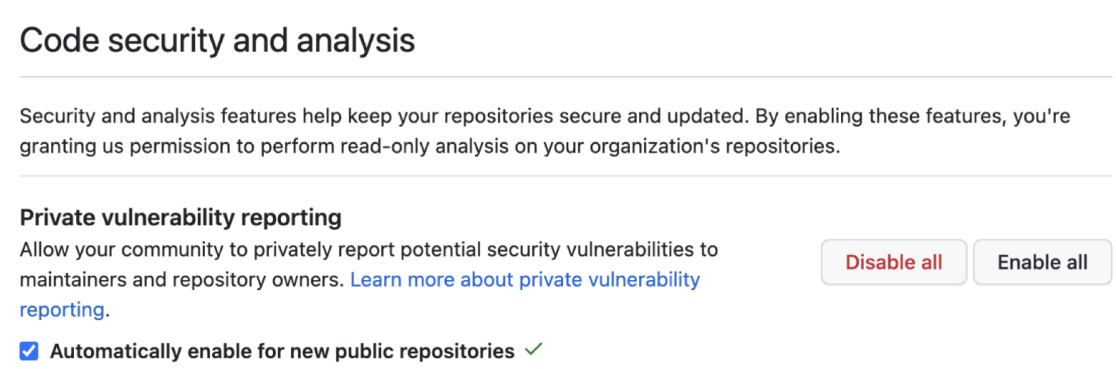
セキュリティ研究者にとってのメリット
TU WienのResearch Unit Security and Privacyで博士研究者をしているMarco Squarcina氏は、学術研究の過程で、Cookieのセキュリティをバイパスする脆弱性を発見しました。「GitHubのPrivate vulnerability reporting機能で報告できることを知り、何人かのメンテナーに連絡して機能を有効化してもらったことで、メールのやり取りを介さずにIssueを報告できるようになりました」

Squarcina氏は、本機能の本当のメリットはプロジェクトオーナーにあると語っています。「脆弱性に関するメールは、フィッシングが疑われたり、気付かれなかったりすることがあります。Private vulnerability reportingでは、プルリクエストのドラフトを使ったコラボレーションチャネルがオープンされるため、メンテナーは必要なものをすべてGitHub上で手に入れることができます」Squarcina氏はGitHub Security Advisoryの報告者に認定されているほか、CVEも作成されています。

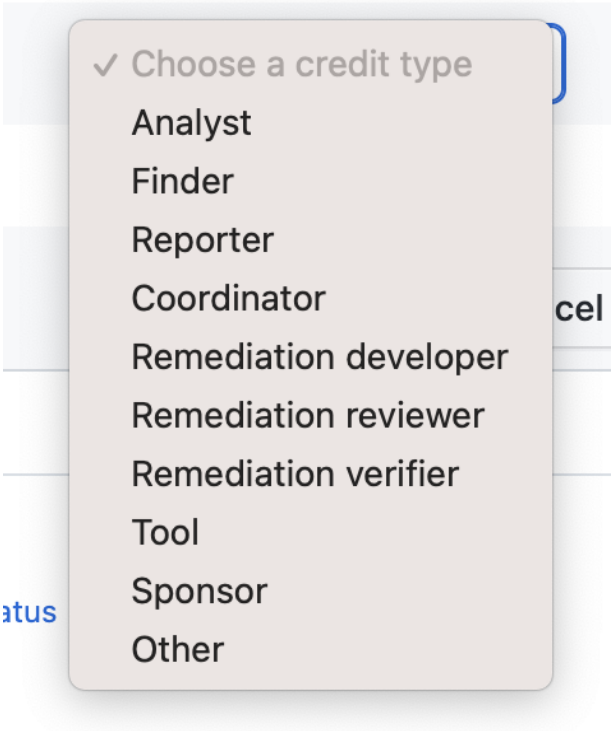
新機能と自動化

オープンソースコミュニティからのフィードバックを受け、GitHubはPrivate vulnerability reportingの一般提供に向けて、多くの改善を実施してきました。

- 大規模な有効化：パブリックベータ版では、Private vulnerability reportingは個々のリポジトリでのみ有効化が可能でした。現在、メンテナーは企業内のすべてのリポジトリでPrivate vulnerability reportingを有効化することができます。



- 複数のクレジットタイプ：メンテナーは、脆弱性の発見や修正に貢献した人に付けるクレジットを選択できます。

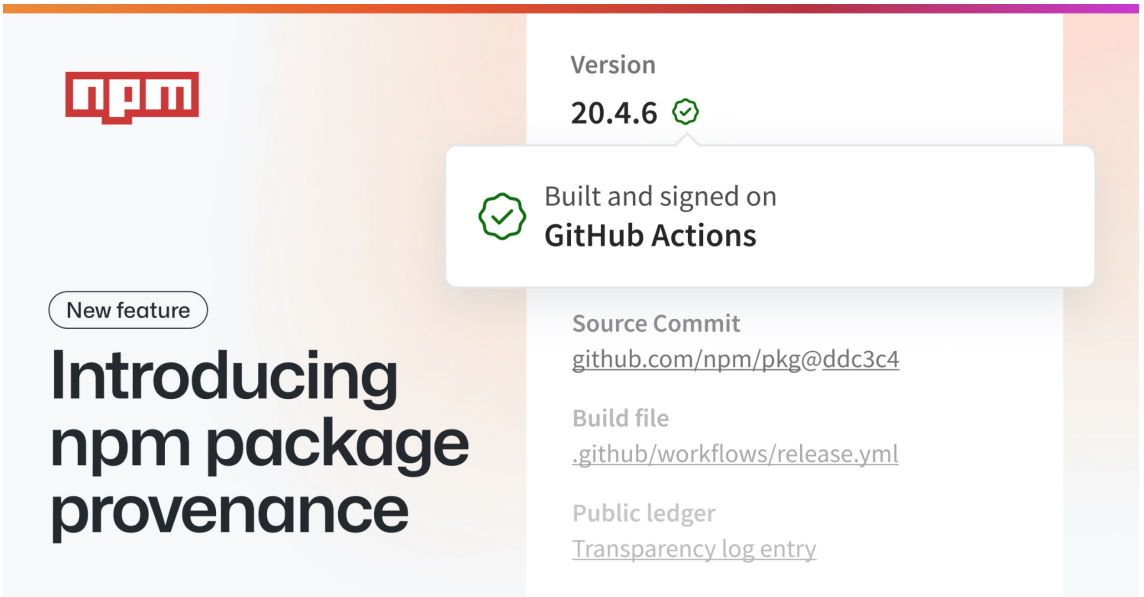


- インテグレーションと自動化：新たなリポジトリ セキュリティ アドバイザリAPIでは、複数の新しいインテグレーションおよび、自動化されたワークフローをサポートしています。
    - サードパーティシステムとのインテグレーション：メンテナーは、GitHubからサードパーティの脆弱性管理システムへ、プライベート脆弱性レポートをパイプライン処理することができます。
    - サブミッションの自動化：セキュリティ研究者は、APIを使用して、複数のリポジトリで確認されたプライベート脆弱性レポートをプログラムで開くこともできます。これは、パッケージが共通の脆弱性を共有している場合に、時間を節約できる便利な機能です。
    - 脆弱性アラート：自動Pingをスケジュールして新しい脆弱性報告通知を受け取れるようにすることで、誰でも重要なリポジトリを監視できます。

Private vulnerability reportingは、Dependabot、code scanning、secret scanningといったGitHubの他のセキュリティ機能と共に、パブリックリポジトリでは無料で利用できます。

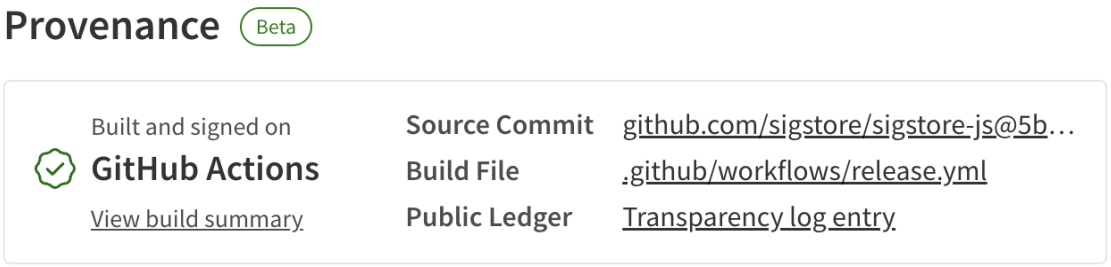
Private vulnerability reportingに関する詳細について
- プライベート脆弱性レポートについて
- リポジトリ セキュリティ アドバイザリ
- リポジトリ セキュリティ アドバイザリAPI

npmパッケージプロベナンスを導入



4月19日より、GitHub Actionsでnpmプロジェクトをビルドする際、"--provenance"フラグを記述することで、パッケージと合わせてプロベナンスの発行が可能になりました。このプロ

ベナンスデータは、ユーザーがパッケージを、そのソースリポジトリと、公開するために使用された特定のビルド手順に紐づけ検証することが可能です。
下記より、npmjs.comの例をご覧ください。
https://www.npmjs.com/package/sigstore#provenance



GitHubでは、ソースリポジトリと具体的なビルド手順について以下のようなメタデータを収集しています。

```
_type: https://in-toto.io/Statement/v0.1
subject:
- name: pkg:npm/sigstore@1.2.0
  digest:
    sha512: 16bf7e5b59e40522190a425047b8c39ffcc8d145cdb15a69fbb9834240a764e2311bda7ac8
predicateType: https://slsa.dev/provenance/v0.2
predicate:
  buildType: https://github.com/npm/cli/gha/v2
  builder:
    id: https://github.com/actions/runner
  invocation:
    configSource:
      uri: git+https://github.com/sigstore/sigstore-js@refs/heads/main
      digest:
        sha1: 5b8c0801d1f5d105351a403f58c38269de93f680
      entryPoint: ".github/workflows/release.yml"
    environment:
      GITHUB_EVENT_NAME: push
      GITHUB_REF: refs/heads/main
      GITHUB_REPOSITORY: sigstore/sigstore-js
      GITHUB_REPOSITORY_ID: '495574555'
      GITHUB_REPOSITORY_OWNER_ID: '71096353'
      GITHUB_RUN_ATTEMPT: '1'
      GITHUB_RUN_ID: '4503589496'
      GITHUB_SHA: 5b8c0801d1f5d105351a403f58c38269de93f680
      GITHUB_WORKFLOW_REF: sigstore/sigstore-js/.github/workflows/release.yml@refs/heac
      GITHUB_WORKFLOW_SHA: 5b8c0801d1f5d105351a403f58c38269de93f680
  materials:
  - uri: git+https://github.com/sigstore/sigstore-js@refs/heads/main
    digest:
      sha1: 5b8c0801d1f5d105351a403f58c38269de93f680
```

パブリックベータ版を利用する方法およびなぜ本機能を開発したのかについての詳細は下記よりご確認ください。
https://docs.npmjs.com/generating-provenance-statements

npmサプライチェーンの信頼性を高める
世界最大級のパッケージレジストリのホームであるGitHubでは、npmエコシステムの健全性を維持するため、セキュリティの改善に継続的に取り組んでいます。その責務の一環が、ビルドの基盤となるオープンソースプロジェクトへの信頼構築です。GitHubは、開発者がソフトウェアサプライチェーンの完全性を確保するために必要なツールを提供したいと考えています。

ソフトウェアサプライチェーンの完全性という問題を解決する答えはなく、ソフトウェア開発ライフサイクル全体に渡り、様々な解決策を必要とします。今までGitHubでは主に、偶発的な脆弱性の検出と修正に力を入れてきました。脆弱な依存関係に対処するツールが改善されるにつれ、アタッカーはますますソフトウェアサプライチェーンの他の弱点を狙うようになっています。彼らは悪用できる脆弱性が開示されるのを待たず、よく使われる依存関係を直接攻撃することで、プロジェクトに悪意のあるコードを挿入しようとします。

この種の攻撃は、意図的でない脆弱性の発生に比べると比較的まれですが、その数は増えています。標的を定めた意図的な攻撃であることから、その影響が大きくなることが多々あります。昨今、UAParser.js、Command-Option-Argument、rcといった主要なnpmパッケージへの攻撃が顕著になっています。

このような攻撃では、ソースコードを直接侵害して攻撃することはほぼなく、侵害した認証情報を悪意あるバージョンの発行に使用することが多くあります。

オープンソースモデルが本来持つ透明性は、ソースコード自体への高い信頼性を生み出します。誰もがソースを見て、あらゆる変更を監査できるという事実は、悪意あるコードが検出されずに残存する可能性を低下させます。しかし、ソースコードを信頼しているからといって、発行されたパッケージも同様に信頼するわけではありません。

レジストリからダウンロードしたnpmパッケージの信頼度を高めるには、どのソースが発行済みアーティファクトに変換されたのか、プロセスを可視化する必要があります。

npmエコシステムに対するGitHubの目標は、コードをビルドして発行するプロセスに対して、オープンソースコードそのものと同じ水準の透明性をもたらすことです。

パッケージをソースとビルドに結び付ける
npmレジストリのほとんどのパッケージページにはソースリポジトリへのリンクが記載されていますが、この情報は検証されていません。また、特定のコミットを指しているわけでもありません。コードエクスプローラーを使用すると、インストールする前にパッケージの内容

を見ることができますが、これはパッケージがどこから来たのかを把握する助けにはなりません。

必要なのは、npmパッケージから、そのパッケージが派生した正確なソースコードのコミットまでのつながりを直接把握する方法です。GitHubはパッケージのプロベナンスステートメントを必要とします。このステートメントでは、最終的なアーティファクトの組み立てに使われた元のソースとビルド手順を、検証可能な記録で確認できます。

SLSA（Supply-chain Levels for Software Artifacts）の仕様はまさにこのような目的のために作られたもので、npmのプロベナンスステートメントに使われています。SLSAのプロベナンススキーマでは、"subject"（発行されたnpmパッケージ）は入力の"materials"（ソースリポジトリとコミットSHA）から発生し、"buildConfig" （パッケージのビルド/発行のために実行したステップ）で処理されたものとして記述されます。これら3つの値は、発行されたパッケージがどのようにソースから派生したかを理解するために必要な情報を正確に教えてくれます。

コードを信頼できるようにする
プロベナンスステートメントを介して検証可能な署名を作成するにあたり、パッケージの署名には、一般的にメンテナーが直接管理するキーが使われます。こうすることで、パッケージが本当にそのキーの所有者によって作られたものであることを利用者が検証できます。ただし、この方法はキーのセキュリティ侵害に対して脆弱であり、ソースコードと発行されたパッケージを結び付ける検証可能な手段にはなりません。

GitHubの目標は、個々のメンテナーに署名を依存するのではなく、ソースコードとビルドプロセスを直接信頼できるものにすることです。

この実現に向けては、信頼できるCI/CDプラットフォームでパッケージをビルドする必要があります。これにより、ビルドをトリガーした特定のコミットや、最終的なアーティファクトの発行に使用した手順を可視化できます。そして、この情報により、ビルドの監査可能性が高まり、コードを改ざんしようとする試みを大幅に把握しやすくなります。

さらに、CI環境とジョブのID（OpenID Connectトークン形式）を用いてプロベナンスステートメントに暗号署名を適用し、パッケージの利用者が検証できる方法でデータの有効性を証明することができます。

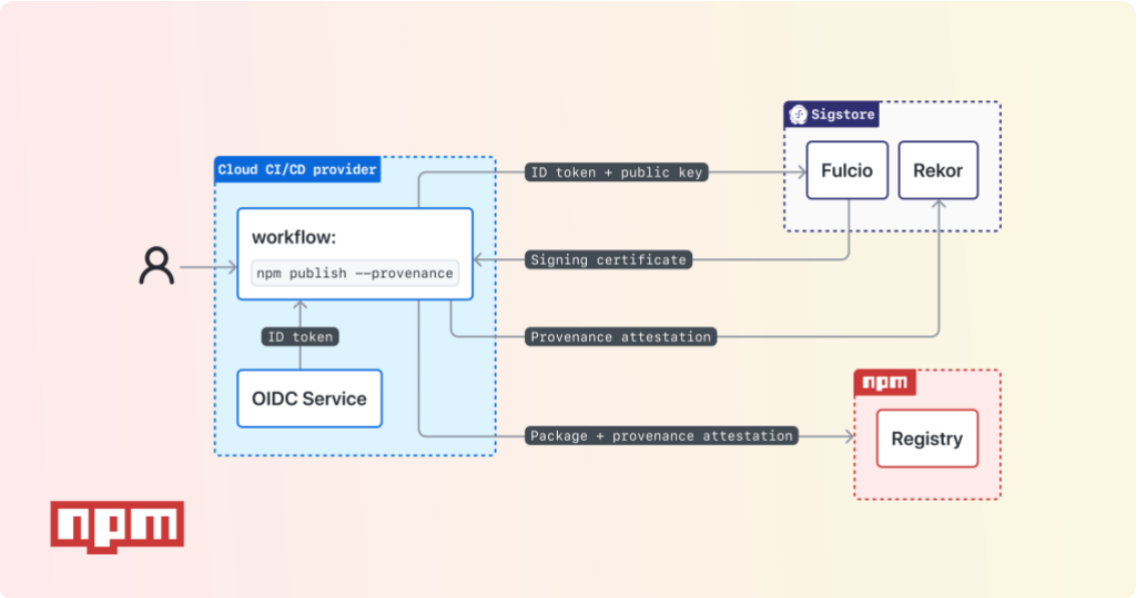
プロベナンスに署名するため、GitHubではSigstoreプロジェクトが提供するツールを活用しています。Sigstoreはパブリック認証局を運営しており、適合するCI/CDプロバイダーからOIDCトークンを受け取り、応答として短時間有効なX.509署名証明書を発行します。

パッケージのプロベナンス生成処理の一部として、GitHubではプロベナンスステートメントに署名するために使い捨てのキーペアを作成し、SigstoreのFulcio CAを呼び出して、そのキーとCIジョブのIDをバインドする署名証明書をリクエストします。キーを管理する必要

はありませんが（署名は生成されるとすぐに削除されます）、署名証明書を提示された人は署名を検証できるだけでなく、その署名を作成したCIジョブのIDも確認できます。

Sigstoreのパブリック認証局を利用するには、対応しているクラウドCI/CDプロバイダー上で実行する必要があります。現在、GitHub Actionsでは対応していますが、できるだけ多くのCI/CDプラットフォームでの対応の促進に取り組んでいます。

対応するCI/CDプロバイダーからプロベナンス付きで発行される場合、以下の手順が実行されます。

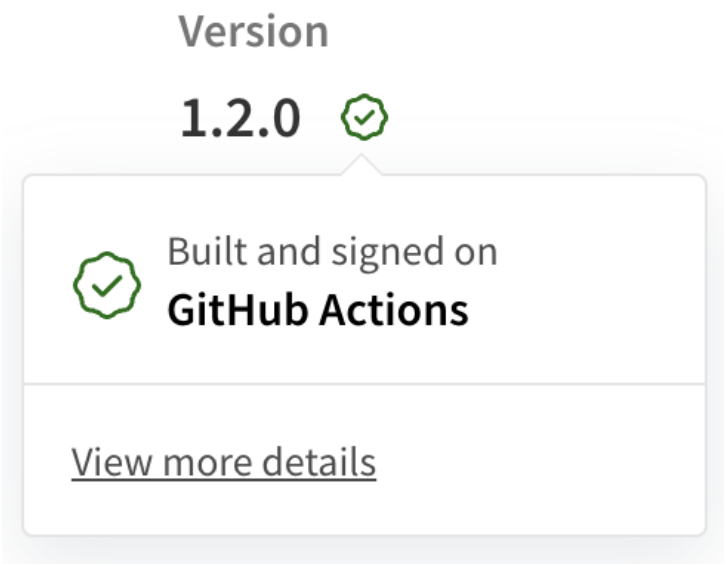


検証

信頼できるソースにより証明されたプロベナンスであることを利用者が検証できるツールも必要です。署名プロセスの一部として、プロベナンス証明はSigstoreのRekorサービスにアップロードされます。公開された、この改ざん不可能な透明性ログにより、後で誰かがプロベナンスや発行済みのパッケージの内容を変更しようとしても、その行為を検出できます。

プロベナンス証明がRekorに投稿されると、発行されるパッケージと合わせてnpmレジストリに送信されます。レジストリは、署名と署名証明書に添付されたIDをチェックし、誰もプロベナンスを偽装しようとしていないことを確認した後、発行されたバージョンを受け入れます。

プロベナンスと一緒に発行されたパッケージは、バージョン番号の横に新しいバッジが付いた状態でnpmjs.comの画面上に表示されます。



開発者はnpm CLI(”npm” 9.5.0以降で利用可能)を使用して、インストールされた依存関係のプロベナンスの完全性も検証できます。

```
npm audit signatures
```

今後の展望
npmパッケージのプロベナンスを一般提供できるようにするため、GitHubはさらに多くの改良に取り組んでいます。

- SLSAプロベナンス仕様のバージョン1.0を採用
- 他のクラウドCI/CDプロバイダーと協力して、プロベナンス署名のサポートを追加
- 想定されるソースリポジトリとコミットが存在することを検証
- CI/CD環境とnpmレジストリ間のアクセスを管理する新しいツールを開発

意図的なサプライチェーン攻撃を防ぐことは、GitHubだけでは不可能です。GitHubはOpenSFFの創設メンバーであり、ソフトウェアリポジトリを保護するワーキンググループに積極的に参加し、同様の機能を他のプラットフォームやパッケージエコシステムに導入することを目指しています。業界全体で一丸となり、オープンソースサプライチェーンを保護するこれらの取り組みに尽力しています。

GitHub Blog
英語:
https://github.blog/2023-04-19-private-vulnerability-reporting-now-generally-available/

https://github.blog/2023-04-19-introducing-npm-package-provenance/#anchor-trust-in-code

GitHub NEWS RELEASE

日本語:
https://github.blog/jp/2023-04-26-private-vulnerability-reporting-now-generally-available/
https://github.blog/jp/2023-04-26-introducing-npm-package-provenance/

GitHubに関する情報は、こちらからもご覧いただけます。
Blog: (英語) https://github.blog (日本語) https://github.blog/jp
Twitter: (英語) @github(https://twitter.com/github)
(日本語) @GitHubJapan(https://twitter.com/githubjapan)

【GitHub について**】https://github.co.jp**
GitHubは、すべての開発者のためのグローバルホームとして、安全なソフトウェアを構築、拡張、提供するための統合された開発者プラットフォームです。フォーチュン100社に採択された企業のうち90社に所属する開発者を含む1億人以上がGitHubを利用し、3億3千万以上のリポジトリで素晴らしいものを共に創造しています。GitHubのすべてのコラボレーション機能によって、個人やチームがより迅速に、より高品質なコードを書くことがかつてないほど容易になっています。

【製品／サービスに関するお問い合わせ先】
ギットハブ・ジャパン営業およびサポート窓口
Email: jp-sales@github.com