11

権限プロンプトの先へ: Claude Code をより安全で自律的にする

— Beyond permission prompts: making Claude Code more secure and autonomous —

公開日	2025-10-20
原題	Beyond permission prompts: making Claude Code more secure and autonomous
著者	Anthropic Engineering Team
原文	https://www.anthropic.com/engineering/claude-code-sandboxing
翻訳	Claude（機械翻訳／Anthropic）
編集	2026-04-09

権限プロンプトの先へ: Claude Code をより安全で自律的にする

Claude Code では、Claude があなたと並んでコードを書き、テストし、デバッグします。コードベースをナビゲートし、複数のファイルを編集し、コマンドを実行して自分の作業を検証します。しかし、Claude にこれほどコードベースとファイルへのアクセスを与えることは、特にプロンプトインジェクションの場合にリスクをもたらします。

これに対処するため、Claude Code にサンドボックス化の上に構築された 2 つの新機能を導入しました。どちらも、開発者のより安全な作業場所を提供すると同時に、Claude がより少ない権限プロンプトでより自律的に動けるよう設計されています。社内での利用では、サンドボックス化が権限プロンプトを 84% 安全に削減することが分かっています。Claude が自由に作業できる境界を定めることで、セキュリティとエージェント性の両方が高まります。

Claude Code でユーザーを安全に保つ

Claude Code は権限ベースのモデルで動いています。デフォルトでは読み取り専用で、変更を加えたりコマンドを実行したりする前に権限を求めます。例外もいくつかあり、`echo` や `cat` のような安全なコマンドは自動許可されますが、ほとんどの操作はやはり明示的な承認が必要です。

「承認」を絶え間なくクリックするのは開発サイクルを遅くし、「承認疲れ」を招きます——承認している内容を注意深く見なくなり、結果として開発が安全でなくなるのです。

これに対処するために、私たちは Claude Code 用のサンドボックス化をリリースしました。

サンドボックス化: より安全で自律的なアプローチ

サンドボックス化は、Claude がアクションごとに権限を求める代わりに、より自由に作業できる事前定義された境界を作ります。サンドボックス化を有効にすると、権限プロンプトが劇的に減り、安全性が高まります。

私たちのサンドボックス化アプローチは、2 つの境界を可能にする OS レベルの機能の上に構築されています。

1. **ファイルシステム分離**: Claude が特定のディレクトリのみにアクセス・変更できるようにする。これは、プロンプトインジェクションされた Claude がセンシティブなシステムファイルを変更するのを防ぐのに特に重要です。

2. **ネットワーク分離**: Claude が承認されたサーバーにのみ接続できるようにする。これは、プロンプトインジェクションされた Claude がセンシティブな情報を漏洩したりマルウェアをダウンロードしたりするのを防ぎます。

効果的なサンドボックス化にはファイルシステム分離と **ネットワーク分離の両方** が必要であることは注目に値します。ネットワーク分離なしでは、侵害されたエージェントが SSH 鍵のようなセンシティブなファイルを外部に送出できてしまいます。ファイルシステム分離なしでは、侵害されたエージェントが簡単にサンドボックスを抜け出し、ネットワークアクセスを得られてしまいます。両方の技法を使うことで、Claude Code ユーザーにより安全で速いエージェント体験を提供できます。

Claude Code の 2 つの新しいサンドボックス機能

サンドボックス化された bash ツール: 権限プロンプトなしの安全な bash 実行

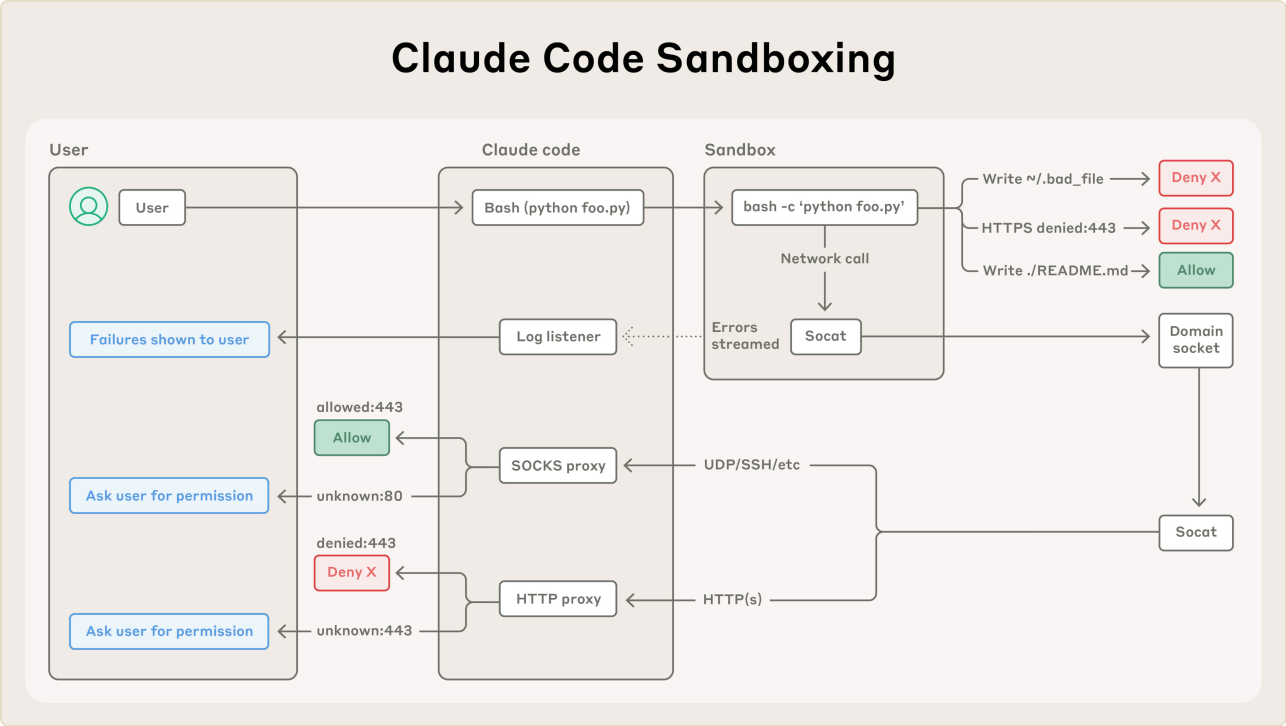
コンテナをスピンアップ／管理するオーバーヘッドなしに、エージェントがアクセスできるディレクトリとネットワークホストを正確に定義できる新しいサンドボックスランタイムをベータとしてリサーチプレビューで導入します。これは任意のプロセス、エージェント、MCP サーバーのサンドボックス化に使えます。また、オープンソースのリサーチプレビュー としても利用可能です。

Claude Code では、このランタイムを bash ツールのサンドボックス化に使い、Claude が設定した定義された制限の中でコマンドを実行できるようにします。安全なサンドボックス内では、Claude はより自律的に動き、権限プロンプトなしでコマンドを安全に実行できます。もし Claude がサンドボックスの *外* の何かにアクセスしようとしたら、即座に通知され、許可するかどうかを選べます。

OS レベルプリミティブ（Linux の bubblewrap や macOS の seatbelt）の上に構築しており、これらの制限を OS レベルで強制します。Claude Code の直接の相互作用だけでなく、コマンドから生成されるあらゆるスクリプト、プログラム、サブプロセスもカバーします。上述のとおり、このサンドボックスは以下を両方強制します。

1. **ファイルシステム分離**: 現在の作業ディレクトリへの読み書きを許可しつつ、その外のファイルの変更をブロック。
2. **ネットワーク分離**: サンドボックスの外で動くプロキシサーバーに接続された Unix ドメインソケット経由でのみインターネットアクセスを許可。このプロキシサーバーは、プロセスが接続できるドメインに関する制限を強制し、新しく要求されたドメインに対するユーザー確認を扱います。さらにセキュリティを高めたい場合は、送信トラフィックに任意のルールを強制するようプロキシをカスタマイズできます。

両方のコンポーネントは設定可能で、特定のファイルパスやドメインを簡単に許可／不許可にできます。



Claude Code のサンドボックスアーキテクチャは、ファイルシステムとネットワーク制御でコード実行を隔離し、安全な操作は自動許可、悪意ある操作はブロック、必要なときだけ権限を求める。

サンドボックス化は、プロンプトインジェクションが成功したとしても完全に隔離され、ユーザー全体のセキュリティに影響を与えないことを保証します。こうして侵害された Claude Code は、あなたの SSH 鍵を盗んだり、攻撃者のサーバーへ通信したりできません。

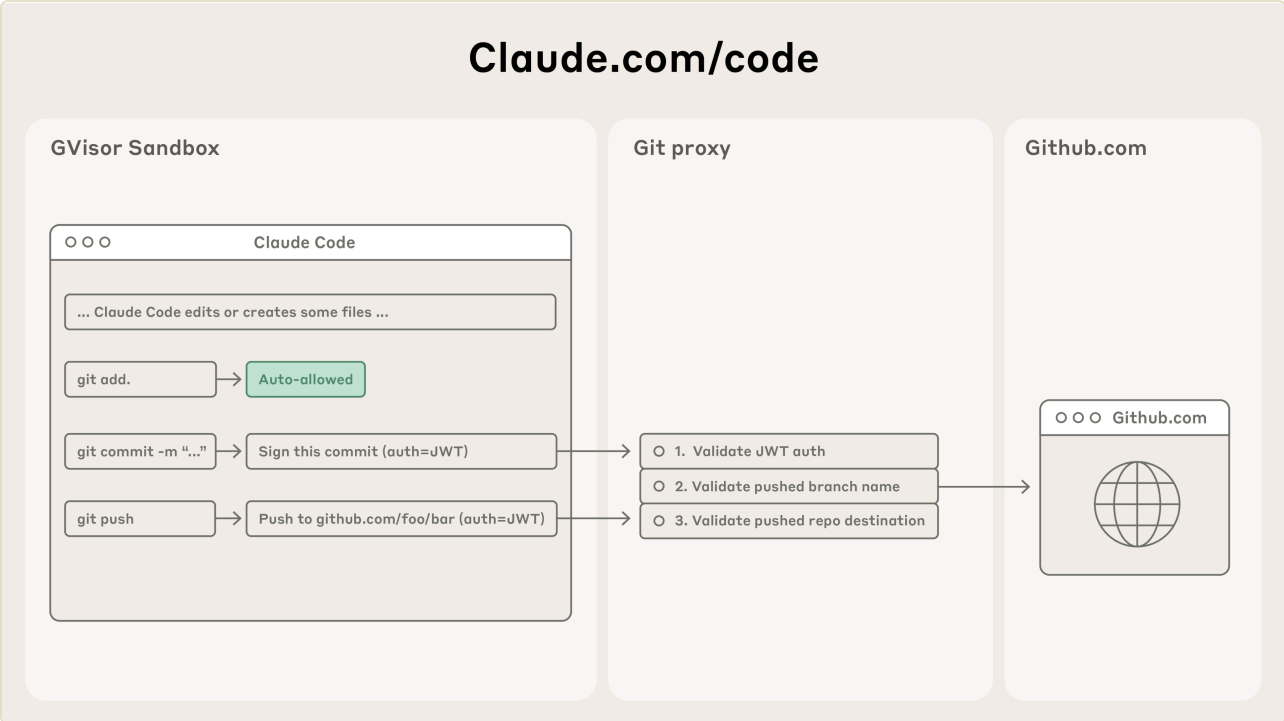
この機能を試すには、Claude Code で `/sandbox` を実行し、私たちのセキュリティモデルに関する技術詳細を確認してください。

他のチームがより安全なエージェントを作りやすいよう、この機能はオープンソース化しました。エージェントのセキュリティ姿勢を高めるために、自分のエージェントでもこの技術の採用を検討してもらえればと思います。

Web 上の Claude Code: クラウドで安全に Claude Code を動かす

同時に今日、Web 上の Claude Code をリリースします。ユーザーがクラウド上の隔離されたサンドボックスで Claude Code を動かせるようにするものです。Web 上の Claude Code は、各 Claude Code セッションを隔離されたサンドボックスで実行し、そこではサーバーへの完全なアクセスを安全かつセキュアに持ちます。このサンドボックスは、git クレデンシャルや署名鍵のようなセンシティブな資格情報が Claude Code のサンドボックス内に入らないよう設計しました。こうしてサンドボックス内で動くコードが侵害されても、ユーザーはさらなる被害から守られます。

Web 上の Claude Code は、git の相互作用をすべて透過的に扱うカスタムプロキシサービスを使います。サンドボックス内では、git クライアントはカスタムビルドのスコープ付き資格情報でこのサービスに認証します。プロキシはこの資格情報と git 相互作用の内容（たとえば設定されたブランチにのみプッシュしていること）を検証し、GitHub にリクエストを送る前に適切な認証トークンを付与します。



Claude Code の Git 連携は、認証トークン、ブランチ名、リポジトリの宛先を検証するセキュアプロキシ経由でコマンドをルーティングする。安全なバージョン管理ワークフローを可能にしつつ、許可されていないプッシュを防ぐ。

始め方

新しくサンドボックス化された bash ツールと Web 上の Claude Code は、Claude をエンジニアリング作業に使う開発者に対して、セキュリティと生産性の両面で大きな改善をもたらします。

これらのツールを試すには:

1. Claude 内で `/sandbox` を実行し、サンドボックスの設定方法についてドキュメントを確認してください。
2. claude.com/code にアクセスして Web 上の Claude Code を試してください。

自分のエージェントを作っているなら、オープンソース化されたサンドボックスコードを確認し、自分の作業に統合することを検討してください。皆さんが何を作るのか楽しみにしています。

Web 上の Claude Code についてもっと知りたい方は、ローンチブログ記事 もご覧ください。

謝辞

記事執筆は David Dworken と Oliver Weller-Davies、貢献者は Meaghan Choi、Catherine Wu、Molly Vorwerck、Alex Isken、Kier Bradwell、Kevin Garcia。