Engineer

Tor

ボランティアの中継ノード経由で通信を送るネットワーク

どのようなものか

Torは各接続を複数のマシンの連鎖へ通し、どの一台にも通信の出発点と到着点の両方を知らせないことで、回線を監視する人から、誰が誰と通信しているかを隠します。連鎖を選ぶのは、読者自身のコンピューターで動くtorというクライアントです。通常は三つのリレー（中継ノード）からなり、入口として繰り返し使うガード、中間ノード、目的地への最終接続を行う出口が並びます。クライアントは区間を一つずつ加えてサーキットを作り、各ノードとの間で別々の鍵を確立し、データの各部分を区間ごとに一層ずつ暗号化します。各ノードは自分の層だけを取り除き、直前と直後の隣接ノードだけを知ります。この入れ子の層が、設計論文The Second-Generation Onion Routerの題名にあるタマネギです。通常、ノード全体をTorネットワークと呼び、小文字のtorはプログラム、それを使うためにまとめられたブラウザーがTor Browserです。

データは固定長のセルで移動するため、中継ノードには元のパケットの形ではなく、均一なブロックの流れが見えます。隔離要件が両立する複数のTCPストリームは、一つのサーキットを共有できます。クライアントは事前にも必要時にもサーキットを作り、経過時間と使用状況に応じてその利用を終了します。長時間続くストリームは元のサーキットに残ることができ、隔離設定によっては、より長い再利用も認められます。アプリケーションはローカルのSOCKSプロキシを介してクライアントへ接続し、この経路で運べるのはTCPだけです。設計論文は、UDPのようなプロトコルはトンネル化しないと明記しています。中継ノードはボランティアが運用し、鍵、アドレス、帯域幅、出口ポリシーを含む記述子をそれぞれ公開します。少数のディレクトリオーソリティが掲載対象を投票で決め、結果に署名してコンセンサス文書を作ります。各クライアントは経路を選ぶ前に、その文書を取得します。

概念を判断に生かす

「Torとは？」を実際の判断に生かしましょう。その概念を、開発または購入するシステムに結び付けて考えるお手伝いをします。 Wi-Fi、ファイアウォール、リモートアクセスを整え、どこからでも働け、権限のある人だけが接続できるようにします。

サービスを見る お問い合わせ →

用語の背後にある前提を検証する

解説とリンク先の実績を参考に、プロジェクトについての質問を整理してください。ツールを選ぶ前に、前提を一緒に検討できます。 Ubuntu Linux VPSインスタンス30台を管理 — 30 復旧計画とともに管理したLinuxサーバー

Ubuntu Linux VPSインスタンス30台を管理 お問い合わせ →

なぜ重要なのか

公開Webサイトの運営者にとって、Torの利用者も読者に含まれるという事実が出発点です。Tor経由の読者は、自分のアドレスではなく出口ノードのアドレスから来たように見えます。ノード一覧は、各クライアントが経路選択に必要とするため、設計上公開されています。そのため、サイトはTorの通信を識別し、別の扱いを選べます。Tor Project自身も、そうしないことを望みつつ、運営者にはブロックする権利があるとしています。残る判断は、その読者をどう迎えるかです。ほかの読者と同じように応答する、速度を落とす、追加の確認を挟む、あるいはTorネットワークを一度も出ない入口を用意する方法があります。最後の方法がonionアドレスです。

Torの約束は正確に捉える必要があります。設計論文は対象外も列挙しています。接続の両端を観測してタイミングと量を比較する相手から通信を守れるとは主張していません。プロジェクトのドキュメントも、それを確実に防ぐ実用的な低遅延方式は研究者にも知られていないと補足しています。基本プロトコルはTorの使用自体を隠しません。プラガブルトランスポートを使うブリッジは、その通信を見分けにくくする方法を加えます。また、出口ノードから目的地までの最後の区間はTorが暗号化しないため、アプリケーション自身の暗号化が必要です。

VPNとは信頼の置き方が異なります。VPN事業者はユーザーと目的地の両方が見え、その情報を見ないよう求められます。Torでは一つのノードが両方を見ることはありません。大半の経路では、選択規則が同じファミリーとして申告されたノードを分け、既定で共通のIPv4 /16またはIPv6 /32サブネットを避けます。文書化された例外があり、サブネットの制限も設定可能です。代償は遅延です。通常の出口サーキットはボランティアの三台のマシンを通るため、軽く、往復の少ないページがTorの読者には適しています。onionアドレスの提供にはtorプロセスと保護すべき秘密鍵が必要ですが、外部からの着信用ポートを開く必要はなく、小さなインフラにも適しています。

実際の活用例

同社のサイトはTorでも読めます。コマンドラインのページには、onionアドレスと、ローカルのtorクライアント経由でMarkdownを取得する `curl` コマンドを掲載しています。アドレスは意図的に `http://` で始まります。バージョン3のonionアドレスは、サービスを識別する公開鍵と、チェックサム、版を示すバイトを符号化しています。TorがHTTPの通信に入る前にサービスを認証し、サーキットを暗号化するため、このonionアドレスによる認証に公開Web証明書は不要です。

onionサービスには、同社の2026年9月の公開サービス用ホストにtorプロセスを動かす余裕が必要でした。単一コア、使用可能なメモリー約464 MBで、すでに七つのドメインのTLSを終端し、Gitフォージも動かしていました。その余裕は議論ではなく、512 MBホストで記録した一日分のサンプルで確認しました。観測した利用可能メモリーの最小値は464 MB中の約310 MB、スワップ使用率は2.6%、CPUのアイドル時間は99.7%だったため、ほかのサービスと並べてTorを動かしました。

プライバシーの通知には、読者を数えるかどうかについて同社が選んだ方針が記載されています。GeminiとGopherのアプリケーションはアクセスログを保持せず、日々のカウンターには要求の合計数だけを記録します。Gopherには接続元アドレスを含むOSの接続ログもあり、その保存期間は30日です。Torでは何も数えず、有効にする設定も提供しないと説明しています。これは運営者のプライバシー上の選択です。Torは読者のネットワークアドレスを隠しますが、サービスによる要求の集計を妨げるわけではありません。通知では、知る仕組みを持つより、知らないままでいる方を選ぶとしています。

問題が起きるところ

名前をTorの外で引いてしまう。 SOCKSプロキシへ接続を送るプログラムでも、先に通常のリゾルバーでホスト名を自分で解決することがあります。その問い合わせは、アクセスするサイトをローカルネットワークとDNS事業者へ知らせます。接続自体がTorを通っても、目的地はすでに漏れています。 `curl` ではオプション一つの違いです。 `--socks5` は名前をローカルで解決し、 `--socks5-hostname` は名前をプロキシへ渡して、出口ノードが解決します。

```
# Resolves the name locally, then connects through the proxy
curl -s --socks5 127.0.0.1:9050 https://example.org/
# Hands the name to tor; the exit relay looks it up
curl -s --socks5-hostname 127.0.0.1:9050 https://example.org/
```

出口の先が平文になる。 Torが暗号化するのはノード間で、最後の区間ではありません。出口は目的地へ通常のTCP接続を開き、アプリケーションが渡した内容を送ります。そのため、平文HTTPで送ったログインフォームは、出口の運営者や、その先のネットワークにいる人が読めます。Torネットワークの外へ出るものには、すべてHTTPSを使います。onionサービスは通信が外へ出ないため、この問題を避けられます。

一つのアドレスが多数の読者を代表する。 クライアントアドレス単位のレート制限や遮断は、同時に多くの人が使う出口ノードを、一人の訪問者と見なします。一人向けに調整した制限は、その出口に到達する全員を締め出し、一人の不正利用で受けた遮断が全員に及びます。汎用のアドレス単位ルールに任せず、出口アドレスをどう扱うかを意識的に決めます。

サーキットの中を本人の情報が通る。 Torが隠すのはネットワークアドレスで、アプリケーションが送る内容には手を加えません。個人アカウントへのログイン、特徴的なブラウザー設定、開くと外部データを取得する文書は、間のノードがいくつあっても、セッションを個人と結び付けます。公式ドキュメントも、アプリケーションの通信から識別情報を除く処理は、利用者側で依然必要だと説明しています。Tor Browserはその仕事を担いますが、通常のブラウザーをプロキシへ向けるだけでは担えません。

家庭の回線で出口ノードを動かす。 ガードや中間ノードに不正利用の苦情が来ることはまれです。目的地から見た通信元は出口なので、苦情や、ときには法的な照会も、その運営者に届きます。Tor Projectの中継ガイドは、家庭から出口を動かさないよう案内しています。自宅のアドレスを一つだけ持つボランティアなら、公開一覧に載らないノードであるブリッジを動かせます。同じガイドが、その状況で勧める選択です。

UDPも使えると期待する。 音声通話、多くのゲーム、DNS問い合わせを自分で送るツールはUDPに依存しますが、Torが運ぶのはTCPストリームだけです。これらはTor経由では動かず、プロキシの使用を強制していなければ、警告なしにTorの外で直接つながることもあります。

参考資料

- Tor仕様 — プロトコル、経路選択、ディレクトリ、onionサービスの仕様をまとめた資料です。
- Tor: The Second-Generation Onion Router — 設計論文。サーキット、セル、出口ポリシーと、設計が目指さないことを明記しています。

- Tor経路仕様 — ガード、中間、出口の選び方と、同じ経路に含められないノードを説明しています。
- Torネットワークの中継ノードの種類 — ガード、中間、出口、ブリッジと、それぞれの運営者が負う法的なリスクです。
- HTTPSの暗号化とTor — Torのみ、HTTPSのみ、両方、どちらもなしの場合に、盗聴者から何が見えるかを対話的に示しています。

用語集

実際の制約に合わせて選ぶ

デプロイの流れ、運用上のリスク、緊急対応の担当者をお知らせください。検査、復旧手段、明確な責任の所在を伴う変更の範囲を定められます。

料金を見る お問い合わせ →

用語を実務に結び付ける

「Torとは？」がお客様の組織にどう当てはまるかをご相談ください。現在のシステムの具体例が一つあれば、話を始められます。

サービスを見る

welcome@engineer.company +45 50 10 61 84 デンマーク、コペンハーゲン

このページを持ち歩く

連絡先カード

心を込めて丁寧に制作： D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

著作権 © 2025 – 現在 · Engineer ApS · すべての権利を留保します。

印刷元： https://engineer.company/ja/glossary/tor/