Engineer

onionサービス

Tor経由でのみ使えるアドレス

どのようなものか

onionサービスは、IPアドレスを指すDNS名の代わりに、サーバー自身の公開鍵から作った名前を使い、Torネットワーク内からアクセスするサーバーです。単にonionとも呼ばれ、一つのtorデーモンで、複数のonionサービスを並行して動かせます。仕様には、以前のhidden service、隠しサービスという名前も残っています。

名前は、base32の56文字に `.onion` を付けたものです。デコードすると35バイトで、サービスの32バイトのed25519識別鍵、その鍵のSHA3-256ハッシュから取る二バイトのチェックサム、3に設定したバージョンの一バイトからなります。最後がバージョンなので、現在のアドレスはすべてdで終わります。ラベルを誰かが選ぶのでなく、鍵を書き表したものです。以前のバージョン2の名前は、RSA-1024鍵のSHA-1ハッシュから取った16文字で、torは2021年の0.4.6.1-alphaで対応を終了しました。

サービスに接続するには、二種類の合流場所を使います。サービスは、いくつかのリレーを紹介ポイントとして選び、そこへのサーキットを開いておき、その一覧を載せた署名付きディスクリプターを、ディレクトリーリレーへ送ります。ディスクリプターは、期間ごとに変わるブラインド化した鍵に対応付けて保存し、本文も暗号化します。保存するリレーがアドレスを知らなければ、一覧を読めません。アドレスを持つクライアントは、ディスクリプターを取得し、名前に含まれる鍵で署名を検査し、一つのリレーをランデブーポイントに選びます。紹介ポイントを通して、そこで合流するようサービスに求めます。サービスは、自分のサーキットをそのリレーへ作ってハンドシェイクを完了し、リレーが二本のサーキットをつなぎます。経路には、両側が三つずつ選んだ、計六つのリレーが入り、互いのIPアドレスは分かりません。以前はクライアント認可、現在は制限付き探索と呼ぶ任意のモードでは、許可したクライアントごとの鍵に向けて、ディスクリプターをさらに暗号化します。紹介ポイントの場所が分かるのは、そのクライアントだけになります。

概念を判断に生かす

「onionサービスとは？」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

名前自体が、通常のサイトでDNSと認証局が分担する仕事を果たし、どちらも必要としません。ディスクリプターを検証し、ハンドシェイクを完了すると、アドレスの背後の秘密鍵を持つ相手と通信していると分かります。 `.onion` を特別用途のドメイン名に予約するRFC 7686は、この名前を、場所と認証を組み合わせるものと説明します。その帰結として、onion名は、登録、割り当て、移転、取り消しができません。所有するとは、鍵ペアを管理することだけです。

暗号化も、同じ設計から生まれます。onionサービスへの通信は、Torネットワークの外へ出ないため、相手側の出口リレーも、最後の平文区間もありません。onionの単なる `http://` でも、入力した名前と照合してクライアントが身元を確認したホストまで、ページをエンドツーエンドで暗号化して運びます。

サービスは、紹介ポイントと、各ランデブーポイントへ向けて、外向きの接続だけを作ります。外から入るポートを開ける必要がなく、NATの内側でも動きます。onionを追加しても、ホストで開いているポートの一覧は変わりません。

読者には、アドレスを知る手段が必要です。サイトが応答ヘッダー `Onion-Location` で知らせると、Tor Browserは、ページのonion版へ進める、紫のカプセル形のボタンをアドレスバーに出します。Tor Projectは、三つの条件を示します。値は、onionホストを持つ `http:` か `https:` のURLであること、ヘッダーを送るページはHTTPSで届くこと、そのページ自体はonion上にないことです。ヘッダーを設定できなければ、metaタグでも知らせられますが、パスは引き継がず、すべてのページから一つの固定URLへ向かいます。

実際の活用例

自社のonionミラーを、読みやすいアドレスで公開した手順を、段階ごとに記録しています。デーモンは、ディストリビューション自身のパッケージから入れ、外向きプロキシのポートを無効にしています。アドレスに対応する秘密鍵は、ホ

スト外で生成し、リポジトリーにもノートパソコンにも、一度も置いていません。ロールは、自分が生成しなかった識別情報を上書きしないため、手で置いた鍵をそのまま使います。アドレスはインベントリーで宣言し、毎回の収束処理で、ホスト上の識別ファイルと一致するかを検査します。鍵を差し替えて宣言を古いままにすると、実行を止めます。二つの秘密情報検査は、どちらもこの鍵ファイルを検出します。分かりやすい鍵ファイルのパターンでは、デプロイ鍵には一致しても、この鍵を見落とすと、測定で分かったためです。

読みやすいプレフィックスは、探索して見つけられますが、選べません。一文字増えるたびに、探索量の期待値が32倍になるため、`engineer` の八文字には、平均で約1.1兆の鍵ペアが必要です。最初の公開時に使ったランダムなアドレスは、段階的に切り替えて廃止し、公開済みのアドレスは、その間も応答し続けました。終わりに見えて、そうではなかった段階もあります。鍵を配置し、収束処理は変更なしと報告したのに、そのディレクトリーを指定するものがなく、新しいアドレスは未公開のままでした。現在のロールは、どのサービスも指定しない識別情報がディスク上にあると、失敗します。

クレジットのページには、GeminiとGopherのアドレスと並べて、完全な `engineer2ezpeji52ipphljcflzyhetvy5r6tnqubtq45nzlxxv6rsqd.onion` を載せています。四つの入口は、同じ一つのフォルダーを読み、一回のデプロイですべてを更新するため、内容が互いにずれません。通常のサイトは `Onion-Location` を送り、読者が求めなくても、Tor Browserがonionを案内します。読者自身のネットワークに見えるものは、入口によって違います。Gopherなら読むページ、Geminiなら要求したアドレス、onionならどちらも見えません。読みやすいプレフィックスは、便利なだけで証明ではないので、比較するのはアドレス全体です。

アドレスの構造は、オフラインで検査できます。次の行は、デコードとチェックサムの再計算を行い、二つの比較がどちらも `True` を表示します。

```
import base64
import hashlib

label = "engineer2ezpeji52ipphljcflzyhetvy5r6tnqubtq45nzlxxv6rsqd"
raw = base64.b32decode(label.upper())
key, checksum, version = raw[:32], raw[32:34], raw[34:]
digest = hashlib.sha3_256(b".onion checksum" + key + version).digest()
print(version == b"\x03", digest[:2] == checksum)
```

正しいチェックサムが示すのは、形式が正しいことだけで、誰のアドレスかではありません。そこは、自社のHTTPSページに載ったアドレスを読むことで確かめます。

問題が起きるところ

鍵だけが、所有の証明です。 名前の背後に登録事業者がいないため、秘密鍵を失えばアドレスを失い、コピーされれば、他人がそこで応答できます。Tor Projectの設定ガイドは、鍵ファイルを秘密に保ち、長く続けるサービスなら、鍵をバックアップするよう求めます。設定でなく、署名鍵として扱います。

最初の文字だけ読みます。 RFC 7686は、狙う名前に似た名前の鍵を、攻撃者が見つけられると警告します。名前は人にとって意味を持たないため、リスクはさらに高くなります。相当の計算で得る読みやすい名前は対策の一つですが、八文字のプレフィックスがあっても、残る48文字は以前と同じく不透明です。アドレス全体を比較するか、ブックマークや、そのサイト自身のHTTPSページのリンクから進みます。

サーバーが自分の名前を明かします。 Torがサービスの稼働場所を隠しても、背後のソフトウェアが明かす場合があります。RFC 7686は、防御が失敗する例に、サーバーがアプリケーション層のメッセージで身元を漏らすことを挙げます。マシンの本当のホスト名を示すエラーページは、その一例です。設定ガイドは、サービスがローカルネットワークに露出しないよう、TCPでなくUnixソケットで、ウェブサーバーに通信を渡すことも勧めます。

```
HiddenServiceDir /var/lib/tor/my-website/
HiddenServicePort 80 unix:/var/run/tor/my-website.sock
```

特に重要なのは、場所そのものを秘密にする場合です。

名前がDNSへ送られます。 Torに対応しないソフトウェアは、通常のリゾルバーにonion名を問い合わせ、ローカルネットワークとDNS提供元に、どのonionを求めたかを知らせます。RFC 7686は、そのようなソフトウェアには、問い合わせずに失敗することを、DNSサーバーには、名前が存在しないと応答することを求めます。代わりに、名前をTorクライアントへ渡します。

`Onion-Location` **のボタンが出ません。** 平文HTTPのリダイレクトにしかヘッダーを設定しないと、HTTPSの条件を満たしません。アドレスを誤記しても、読者はどこにも着けません。最後のHTTPS応答で、ヘッダーを確認します。

参考資料

- Torランデブー仕様、バージョン3 — 識別鍵とブラインド化鍵、ディスクリプター、紹介、ランデブーを含む、プロトコル全体です。
- onionアドレスのエンコード — 公開鍵を56文字の名前に変える式と、アドレスの例です。
- onionサービスの仕組み — 紹介ポイント、ディスクリプター、ランデブーを、一歩ずつ説明します。
- onionサービスの設定 — torrcの行、torが作るファイル、鍵と場所の漏洩に関する助言です。
- Onion-Location — Apache、Nginx、Caddyのヘッダーとmetaタグの設定、およびヘッダーが満たすべき条件です。
- RFC 7686 — `.onion` を予約する理由と、リゾルバー、DNS運用者、登録事業者に必要な対応です。

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

「onionサービスとは？」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/onion/