Engineer

Onion-Dienst

eine Adresse, die nur im Tor-Netzwerk erreichbar ist

Was es ist

Ein Onion-Dienst ist ein Server, den Leser innerhalb des Tor-Netzwerks über einen aus seinem eigenen öffentlichen Schlüssel gebildeten Namen erreichen. An die Stelle eines DNS-Namens, der auf eine IP-Adresse verweist, tritt dieser Schlüsselname. Der Dienst wird häufig kurz Onion genannt. Ein tor-Daemon kann mehrere Onion-Dienste nebeneinander betreiben, und die Spezifikation verwendet noch die ältere Bezeichnung Hidden Service.

Der Name besteht aus 56 Base32-Zeichen, gefolgt von `.onion`. Dekodiert sind das 35 Bytes: der 32 Byte lange ed25519-Identitätsschlüssel des Dienstes, eine zwei Byte lange Prüfsumme aus einem SHA3-256-Hash dieses Schlüssels und ein auf 3 gesetztes Versionsbyte. Weil dieses Byte zuletzt kommt, endet jede heutige Adresse mit dem Buchstaben d. Niemand wählt die Zeichenfolge aus; sie ist der ausgeschriebene Schlüssel. Die früheren Namen der Version 2 bestanden aus 16 Zeichen eines SHA-1-Hashs eines RSA-1024-Schlüssels. tor entfernte ihre Unterstützung mit Version 0.4.6.1-alpha im Jahr 2021.

Um den Dienst zu erreichen, braucht es zwei Arten von Treffpunkten. Der Dienst wählt einige Relays als Introduction Points, hält Verbindungen in Form von Circuits zu ihnen offen und lädt einen signierten Deskriptor mit ihrer Liste auf Verzeichnis-Relays hoch. Der Deskriptor wird unter einem verblindeten Schlüssel abgelegt, der sich jede Periode ändert. Sein Inhalt ist verschlüsselt, sodass ein speicherndes Relay die Liste ohne Kenntnis der Adresse nicht lesen kann. Ein Client, der die Adresse kennt, ruft den Deskriptor ab, prüft die Signatur gegen den Schlüssel im Namen, wählt ein Relay als Rendezvous Point und bittet den Dienst über einen Introduction Point, sich dort zu treffen. Der Dienst baut einen eigenen Circuit zu diesem Relay auf und schließt den Handshake ab; das Relay verbindet die beiden Circuits. Der Weg durchläuft sechs Relays, drei von jeder Seite ausgewählt, und keine Seite erfährt die IP-Adresse der anderen. Ein optionaler Modus, heute Restricted Discovery und früher Client Authorization

genannt, verschlüsselt den Deskriptor zusätzlich für den Schlüssel jedes zugelassenen Clients. Nur diese Clients erfahren dadurch, wo die Introduction Points liegen.

Das Konzept für eine Entscheidung nutzen

Nutzen Sie „Was ist ein Onion-Dienst?“ für eine konkrete Entscheidung. Wir können helfen, das Konzept mit dem System in Verbindung zu bringen, das Sie entwickeln oder einkaufen. WLAN, Firewalls und Fernzugriff werden so eingerichtet, dass Mitarbeitende von überall arbeiten können und nur Berechtigte Zugang erhalten.

Unsere Leistungen entdecken Kontakt aufnehmen →

Die Annahmen hinter dem Begriff prüfen

Nutzen Sie die Erklärung und die verlinkten Arbeiten, um Fragen für Ihr Projekt zu formulieren. Wir können die Annahmen gemeinsam mit Ihnen prüfen, bevor ein Werkzeug ausgewählt wird. 30 Ubuntu-Linux-VPS-Instanzen verwaltet — 30 verwaltete Linux-Server mit geplantem Wiederherstellungsweg

30 Ubuntu-Linux-VPS-Instanzen verwaltet Kontakt aufnehmen →

Warum es wichtig ist

Der Name übernimmt die Aufgabe, die sich bei einer gewöhnlichen Website DNS und eine Zertifizierungsstelle teilen, und braucht keines von beiden. Die Prüfung des Deskriptors und der abgeschlossene Handshake zeigen dem Client, dass er mit dem Inhaber des privaten Schlüssels hinter der Adresse spricht. RFC 7686 reserviert `.onion` als Domainnamen für besondere Verwendungszwecke und beschreibt solche Namen als Verbindung von Ort und Authentifizierung. Daraus folgt: Ein Onion-Name kann weder registriert noch zugewiesen, übertragen oder widerrufen werden. Ihn zu besitzen bedeutet allein, das Schlüsselpaar zu kontrollieren.

Aus demselben Entwurf folgt die Verschlüsselung. Verkehr zu einem Onion-Dienst verlässt das Tor-Netzwerk nie. Am anderen Ende gibt es deshalb weder ein Exit-Relay noch einen letzten unverschlüsselten Abschnitt. Auch einfaches `http://` auf einem Onion-Dienst überträgt eine Seite somit Ende zu Ende verschlüsselt zu einem Host, dessen Identität der Client gegen den eingegebenen Namen geprüft hat.

Der Dienst baut ausschließlich ausgehende Verbindungen auf: zu seinen Introduction Points und zu jedem Rendezvous Point. Er braucht deshalb keinen offenen eingehenden Port und funktioniert hinter NAT. Das Hinzufügen eines Onion-Dienstes lässt die Liste offener Ports des Hosts unverändert.

Die Leser müssen die Adresse dennoch erfahren. Eine Website kann sie mit einem `Onion-Location` -Antwortheader bekannt geben. Tor Browser zeigt dann in der Adressleiste eine violette, pillenförmige Schaltfläche, die zur Onion-Kopie der Seite führt. Das Tor-Projekt nennt drei Bedingungen: Der Wert ist eine `http:` - oder `https:` -URL mit einem Onion-Host, die ankündigende Seite wurde über HTTPS abgerufen, und diese Seite liegt nicht selbst auf einem Onion-Dienst. Wo sich keine Header setzen lassen, kann ein Meta-Tag dieselbe Ankündigung übernehmen, allerdings ohne den Pfad: Jede Seite verweist dann auf eine feste URL.

Wie es in der Praxis aussieht

Die Veröffentlichung des Onion-Spiegels der Firma unter einer lesbaren Adresse ist Schritt für Schritt dokumentiert. Der Daemon stammt aus dem von der Distribution selbst bereitgestellten Paket; sein Port für ausgehende Proxy-Verbindungen ist abgeschaltet. Der private Schlüssel, der die Adresse bestimmt, wurde außerhalb des Hosts erzeugt und gelangte nie ins Repository oder auf einen Laptop. Die Rolle überschreibt keine Identität, die sie nicht selbst erzeugt hat. Ein von Hand abgelegter Schlüssel wird deshalb unverändert verwendet. Die Adresse ist als Wert im Inventar deklariert, und jeder Konvergenzlauf prüft, ob die Identitätsdatei auf dem Host dazu passt. Ein ausgetauschter Schlüssel bei veralteter Deklaration stoppt somit den Lauf. Beide Prüfungen auf Geheimnisse erkennen die Schlüsseldatei, nachdem eine Messung gezeigt hatte, dass das naheliegende Muster für Schlüsseldateien einen Deploy-Schlüssel fand, diese Datei aber übersah.

Ein lesbares Präfix lässt sich finden, aber nicht auswählen. Jedes zusätzliche Zeichen multipliziert den erwarteten Suchaufwand mit 32. Die acht Buchstaben von `engineer` erfordern deshalb im Durchschnitt etwa 1,1 Billionen Schlüsselpaare. Die Ablösung der zufälligen Adresse, mit der der Spiegel anfangs veröffentlicht wurde, erfolgte stufenweise, sodass die veröffentlichte Adresse durchgehend erreichbar blieb. Ein Schritt sah endgültig aus, war es aber nicht: Der Schlüssel war abgelegt, der Konvergenzlauf meldete keine Änderung, und die neue Adresse blieb unveröffentlicht, weil kein Dienst ihr Verzeichnis nannte. Die Rolle schlägt nun bei jeder Identität auf dem Datenträger fehl, die kein Dienst benennt.

Die Credits-Seite zeigt die vollständige Adresse, `engineer2ezpeji52ipphljcflzyhetvy5r6tnqubtq45nzlxxv6rsqd.onion` , neben denen für Gemini und Gopher. Die vier Zugänge teilen sich einen Dateiordner und eine Bereitstellung, sodass keiner von den anderen abweichen kann. Die gewöhnliche

Website sendet `Onion-Location`, damit Tor Browser den Onion-Zugang von sich aus anbietet. Was das eigene Netzwerk des Lesers sieht, hängt vom Zugang ab: bei Gopher die gelesene Seite, bei Gemini die angefragte Adresse, beim Onion-Zugang keines von beiden. Das lesbare Präfix dient der Bequemlichkeit, nicht als Nachweis. Verglichen werden muss deshalb die vollständige Adresse.

Der Aufbau der Adresse lässt sich offline prüfen. Diese Zeilen dekodieren sie und berechnen die Prüfsumme neu; beide Vergleiche geben `True` aus:

```
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)
```

Eine gültige Prüfsumme zeigt, dass die Adresse korrekt aufgebaut ist, nicht wem sie gehört. Diese Zuordnung ergibt sich daraus, dass sie auf den eigenen HTTPS-Seiten der Firma steht.

Wo es schiefgeht

Der Schlüssel ist der einzige Eigentumsnachweis. Ohne Registrierungsstelle hinter dem Namen bedeutet ein verlorener privater Schlüssel eine verlorene Adresse. Wer ihn kopiert, kann selbst unter ihr antworten. Die Einrichtungsanleitung des Tor-Projekts fordert, die Schlüsseldateien privat zu halten und den Schlüssel zu sichern, wenn der Dienst dauerhaft bestehen soll. Behandeln Sie ihn als Signaturschlüssel, nicht als Konfiguration.

Nur die ersten Buchstaben lesen. RFC 7686 warnt, dass Angreifer Schlüssel finden können, deren Namen dem beabsichtigten ähneln. Das Risiko ist größer, weil die Namen für einen Menschen keine Bedeutung haben. Lesbare Namen, mit erheblichem Rechenaufwand erkauft, nennt der RFC als eine Abhilfe. Ein Präfix aus acht Zeichen lässt die übrigen 48 aber genauso undurchsichtig wie zuvor. Vergleichen Sie die ganze Adresse oder nutzen Sie ein Lesezeichen oder einen Link auf den eigenen HTTPS-Seiten der Website.

Der Server nennt sich selbst. Tor verbirgt, wo der Dienst läuft, doch die Software dahinter kann es verraten. RFC 7686 nennt als einen Fehlerfall einen Server, der seine Identität in einer Nachricht auf Anwendungsebene preisgibt. Eine Fehlerseite mit dem tatsächlichen Hostnamen der Maschine ist ein Beispiel dafür. Die Einrichtungsanleitung empfiehlt außerdem, den Verkehr über einen Unix-Socket statt über TCP an den Webserver weiterzugeben, damit der Dienst nicht im lokalen Netzwerk erreichbar wird:

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

Am wichtigsten ist das dort, wo der Standort selbst geheim bleiben soll.

Der Name landet im DNS. Software, die Tor nicht beherrscht, fragt den gewöhnlichen Resolver nach einem Onion-Namen. Damit erfahren das lokale Netzwerk und der DNS-Anbieter, welcher Onion-Dienst gesucht wurde. RFC 7686 weist solche Software an, ohne diese Abfrage zu scheitern, und DNS-Server, zu antworten, dass der Name nicht existiert. Übergeben Sie den Namen stattdessen dem Tor-Client.

Die `Onion-Location` -Schaltfläche erscheint nie. Ein Header, der nur auf der unverschlüsselten HTTP-Weiterleitung gesetzt ist, verletzt die HTTPS-Bedingung. Ein Header mit vertippter Adresse schickt Leser ins Leere. Prüfen Sie den Header in der endgültigen HTTPS-Antwort.

Weiterführende Lektüre

- Tor-Rendezvous-Spezifikation, Version 3 — das vollständige Protokoll: Identitäts- und verblindete Schlüssel, Deskriptoren, Introduction und Rendezvous.
- Onion-Adressen kodieren — die Formel, die aus einem öffentlichen Schlüssel einen Namen mit 56 Zeichen macht, mit Beispieladressen.
- Wie funktionieren Onion-Dienste? — Introduction Points, Deskriptoren und Rendezvous Schritt für Schritt erklärt.
- Einen Onion-Dienst einrichten — die torrc-Zeilen, die von tor erzeugten Dateien sowie Hinweise zu Schlüsseln und zum Verraten des Standorts.
- Onion-Location — Header- und Meta-Tag-Konfigurationen für Apache, Nginx und Caddy sowie die Bedingungen für den Header.

- RFC 7686 — warum `.onion` reserviert ist und was Resolver, DNS-Betreiber und Registrierungsstellen damit tun müssen.

Glossar

Für Ihre tatsächlichen Rahmenbedingungen auswählen

Beschreiben Sie den Bereitstellungsweg, die Betriebsrisiken und die Bereitschaftsverantwortlichen. Wir können eine Änderung mit Prüfungen, einem Wiederherstellungsweg und klaren Zuständigkeiten abgrenzen.

Preise ansehen Kontakt aufnehmen →

Die Begriffe praktisch nutzbar machen

Fragen Sie, wie „Was ist ein Onion-Dienst?“ auf Ihre Organisation zutrifft. Ein konkretes Beispiel aus Ihrem bestehenden System reicht aus, um das Gespräch zu beginnen.

Unsere Leistungen entdecken

welcome@engineer.company +45 50 10 61 84 Kopenhagen, Dänemark

Nehmen Sie diese Seite mit

Kontaktkarte

Mit Sorgfalt und Liebe gestaltet von D.

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

Copyright © 2025 – heute · Engineer ApS · Alle Rechte vorbehalten.

Gedruckt von https://engineer.company/de/glossary/onion/