TOR - The Onion Router

Dozent: Prof. Dr. Michael Eichberg
Kontakt: michael.eichberg@dhbw.de

Version: 1.3.0.1

Folien: [HTML] https://delors.github.io/sec-tor/folien.de.md.html
[PDF] https://delors.github.io/sec-tor/folien.de.md.html.pdf

Fehler melden: https://github.com/Delors/delors.github.io/issues

KI Verwendung: Bei der Erstellung der Unterlagen wurden KI Assistenten (insbesondere Claude aber ggf. auch ChatGPT, Ollama mit Gemma/LLama/Qwen oder OpenCode mit Kimi/Qwen/Deepseek...) unterstützend eingesetzt. Dies erfolgte insbesondere zur Unterstützung bei der Generierung von Grafiken (d. h. SVG Dateien), oder um sich Übersichtstabellen generieren zu lassen. Weiterhin wurde KI zur allgemeinen Qualitätssicherung eingesetzt. Inhalte, die ggf. von der KI vorgeschlagen wurden, wurden im Falle der Übernahme explizit validiert und angepasst.

1. Grundlagen

Tor (The Onion Router)

Anwendungsunabhängiger **low-latency Anonymisierungsdienst für TCP-Verbindungen**, der den Standort und die IP des Nutzers verschleiert.

- Typische Anwendung: anonymes Surfen im Internet und Instant Messaging (z. B. Briar)
- Frei und Open Source
- gegründet 2002, öffentlich nutzbar seit 2003, Code seit 2004 frei verfügbar
- Baut ein *Overlay-Netzwerk* auf
- Grundlegendes Prinzip: Onion Routing

---

low-latency: Die Verzögerung durch die Anonymisierung ist so gering, dass Tor für Instant Messaging und das Surfen im Internet verwendet werden kann.

Overlay-Netzwerk: Tor baut ein eigenes Netzwerk auf, welches auf dem Internet aufsetzt. Die Verbindungen zwischen den Tor-Knoten werden von Tor zusätzlich verschlüsselt.

Tor - Verwendung

- legale/intendierte Nutzungen: Nutzer mit allg. Datenschutzbedürfnissen, *Whistleblowers*, Dissidenten, Journalisten, ...
- illegale Zwecke (**Darknet**)

  Marktplatz für illegale Waren und Dienstleistungen, z. B. Drogen, Waffen, gefälschte Dokumente, ... (z. B. Silk Road, oder CyberBunker als Hosting-Provider für illegale Dienst)
  Es wird geschätzt, dass etwa 80% des Datenverkehrs im Zusammenhang mit dem Zugriff auf Kinderpornografie steht. Solche Schätzungen sind allerdings mit Vorsicht zu genießen!

Wir können unterscheiden zwischen:

***Surface Web*:**	Das frei-verfügbare Internet. (Auch als *Clear Web* bezeichnet.)
***Deep Web*:**	Das Internet, das durch Zugriffsbeschränkungen nicht öffentlich zugänglich ist.
***Dark Web*:**	Das Internet, das nicht indiziert ist und nur über spezielle Software (z. B. Tor) erreichbar ist. Häufig - aber nicht ausschließlich - wird das Dark Web für illegale Aktivitäten genutzt und ist meist zusätzlich durch Zugriffsbeschränkungen geschützt.

2. Sicherheit

Tor - potentielle Sicherheitslücken

- DoS Attacken
- Identifikation von *Onion Services* (aka *Hidden Services*)
- Deanonymisierungsattacken

Aufstellung von Angriffen auf Tor.

Tor - Deanonymisierungsattacke 2013

> Wenn ein einzelner Nutzer Tor über einen längeren Zeitraum [3 bis 6-Monate, abhängig von einigen Faktoren] regelmäßig nutzt, ist es fast sicher, dass er *de-anonymisiert* werden kann.
>
> [Übersetzt mit DeepL.]
>
> —2013 - Tor is not as safe as you may think

Tor - Deanonymisierungsattacke 2024

> Strafverfolgungsbehörden haben offenbar einen Weg gefunden, Nutzer des Tor-Netzwerkes gezielt zu deanonymisieren. Möglich sein soll dies [...] durch sogenannte Timing-Analysen, die bisher als praktisch nicht umsetzbar galten. Dadurch sei es aufgrund der verschlüsselten Datenpakete zwar nicht möglich zu sehen, was Nutzer verschickten, dass sie überhaupt kommunizierten, sei aber sichtbar.
>
> [...] ist dafür allerdings eine teils jahrelange Überwachung einzelner Tor-Nodes erforderlich. [...]
>
> —2024 - Behörden unterwandern Tor

Tor - Hintergrund

- Die grundlegende Idee ist es eine Trennung zwischen der Quelle und dem Ziel des Datenverkehrs zu schaffen.
- Der Datenverkehr wird über *mehrere Knoten* (*Relays*) umgeleitet, die jeweils nur den vorherigen und den nächsten Knoten kennen. Der Weg den ein Datenpaket nimmt, wird als *Circuit* oder *Path* bezeichnet.
- Der Pfad wird dazu vorher ausgewählt und der gesamte Datenverkehr entsprechend des Pfades verschlüsselt.
- Tor bietet Anonymität auch für die Serverseite durch *Onion Services* (auch *Hidden Services*), die nur über eine von Tor vergebene Onion-Adresse erreicht werden können.

Tor - Bedrohungsmodel

Tor bietet Schutz vor folgenden Angreifern: Einem Angreifer dem es gelingt ...

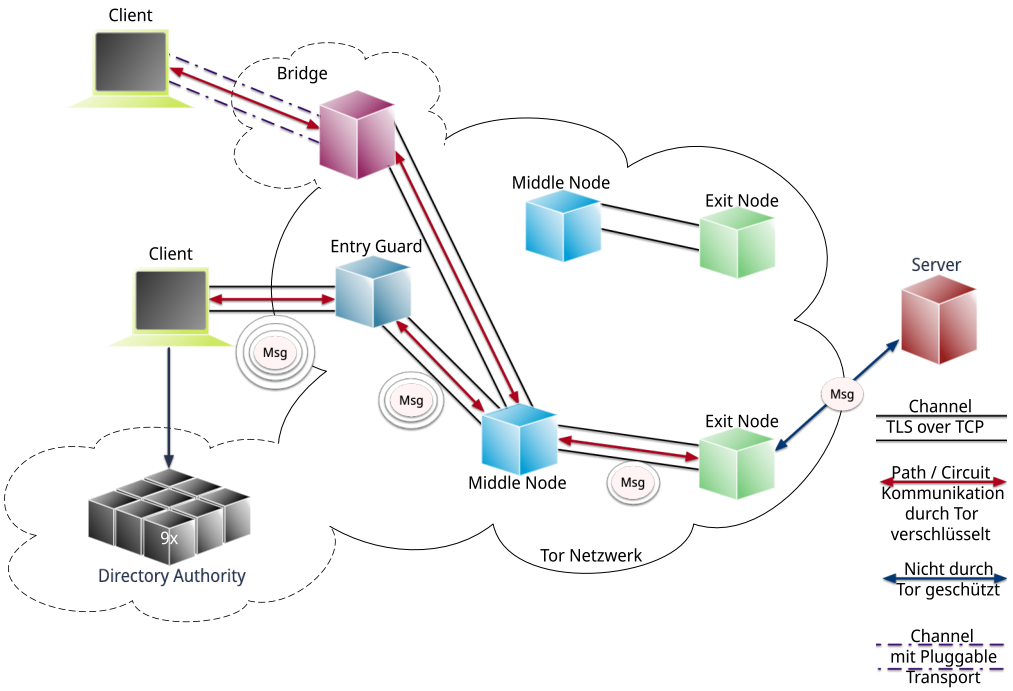
- einen Teil der Kommunikation zu beobachten und
- nur einen Teil der Tor-Knoten zu kontrollieren, indem er entweder einen eigenen Tor-Knoten (*Relay*; früher *Onion-Router*) betreibt oder einen bereits laufenden Knoten kompromittiert.

⚠ Warnung

Folgendes Szenario ist nicht abgedeckt: Ein Angreifer, der beide Enden der Kommunikation, den `Entry Guard` und den `Exit Node` überwachen kann.

Gegen solche Angreifer bietet Tor keine Anonymität.

Tor - Aufbau



Spezifikation

Tor-Knoten: Rechner, die das Tor-Netzwerk bilden. Es gibt drei Arten von Tor-Knoten:

- *Entry Nodes* (auch *Guard Nodes*): Diese Knoten sind die ersten Knoten in der Kette. Sie kennen die IP-Adresse des Clients. Sie können den Datenverkehr nicht entschlüsseln. Sie können aber sehen, dass der Datenverkehr von einem bestimmten Client kommt.
- *Middle Nodes*: Diese Knoten sind die mittleren Knoten in der Kette. Sie kennen weder die IP-Adresse des Clients noch die IP-Adresse des Ziels. Sie können den Datenverkehr nicht entschlüsseln. Sie können aber sehen, dass der Datenverkehr von einem bestimmten Entry Node kommt und an einen bestimmten Exit Node geht.
- *Exit Nodes*: Diese Knoten sind die letzten Knoten in der Kette. Sie kennen die IP-Adresse des Ziels. Sie können den Datenverkehr entschlüsseln. Sie können aber nicht sehen, von welchem Entry Node der Datenverkehr kommt.
- *Bridge Nodes*: Diese Knoten sind *Entry Nodes*, die nicht bzw. nicht vollständig öffentlich bekannt. Diese dienen ggf. dazu in Ländern, in denen Tor blockiert wird, den Zugang zu Tor zu ermöglichen. Sollte eine Verbindung zu einer Bridge nicht hergestellt werden können, aufgrund der Struktur der Nachrichten - zum Beispiel aufgrund der Verwendung von *Deep Packet Inspection* - dann ist es möglich diese mit Hilfe von *Pluggable Transports* zu verschleiern.

Tor-Netzwerk: besteht aus mehreren tausend Tor-Knoten. Viele Knoten sind freiwillig betriebene Knoten.

Circuit/Path:

Ein Circuit besteht typischerweise aus drei Knoten: *Entry Node*, *Middle Node* und *Exit Node*. Mehr Knoten sind möglich, haben jedoch nur einen geringen Einfluss auf die Sicherheit. Die Übertragung der Daten zwischen diesen Knoten erfolgt verschlüsselt. In welcher Form die Daten vom *Exit Node* zum Ziel übertragen werden, ist nicht Teil von Tor. Hat der Client eine verschlüsselte Verbindung initiiert (HTTPS), dann ist auch der Datenverkehr zwischen dem Exit Node und dem Ziel (noch) verschlüsselt ansonsten nicht und der Exit Node kann den Datenverkehr lesen.

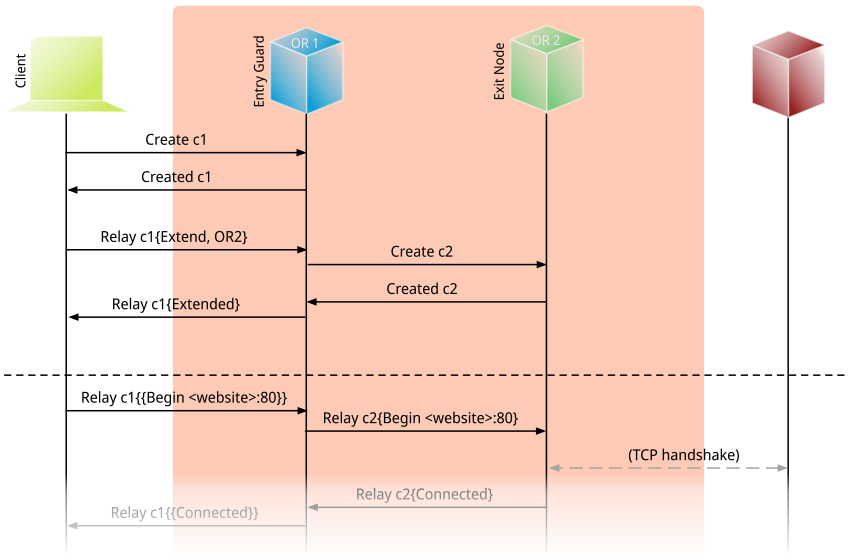
Directory Authority:
Knoten, die die Liste der aktiven Tor-Knoten verwalten. Diese Liste wird von allen Tor-Knoten regelmäßig in Hinblick auf das *Consensus Document* bzgl. der Knoten und deren Eigenschaften sowie Zustand abgefragt. Das *Consensus Document* wird von den *Directory Authorities* einmal pro Stunde gemeinsam erstellt und beschreibt die relevanten Eigenschaften jedes Tor-Knotens. Die Authentizität des *Consensus Document* wird durch die Signaturen der *Directory Authorities* nachgewiesen.
Es gibt (Stand 2023) 9 *Directory Authorities*.

🇺🇸 ***Onion Routing***: bedeutet, dass die Datenpakete mehrfach verschlüsselt werden. Jeder Tor-Knoten kann nur die Verschlüsselungsschicht entfernen, für die er den Schlüssel hat. Die Schlüssel werden mit dem Client während des Aufbaus des Circuits ausgehandelt. Es gibt für jeden Tor-Knoten einen eigenen Schlüssel und die Nachrichten werden in umgekehrter Reihenfolge der Tor-Knoten entlang des Pfades verschlüsselt. d. h. die Verschlüsselung für den Entry Node wird als letztes angewendet, da diese als erstes entfernt wird.

Cells: sind die Datenpakete, die zwischen den Tor-Knoten ausgetauscht werden. Cells sind immer 512Byte groß, um es unmöglich zu machen anhand der Größe der Datenpakete Rückschlüsse auf die Daten zu ziehen.

> Hinweis
> In älteren Dokumenten wird der *Client* auch als *Onion Proxy (OP)* bezeichnet und die Tor-Knoten als *Onion Router (OR)*. Die Tor-Knoten (🇺🇸 *Nodes*) werden auch als *Onion Relay* bezeichnet.

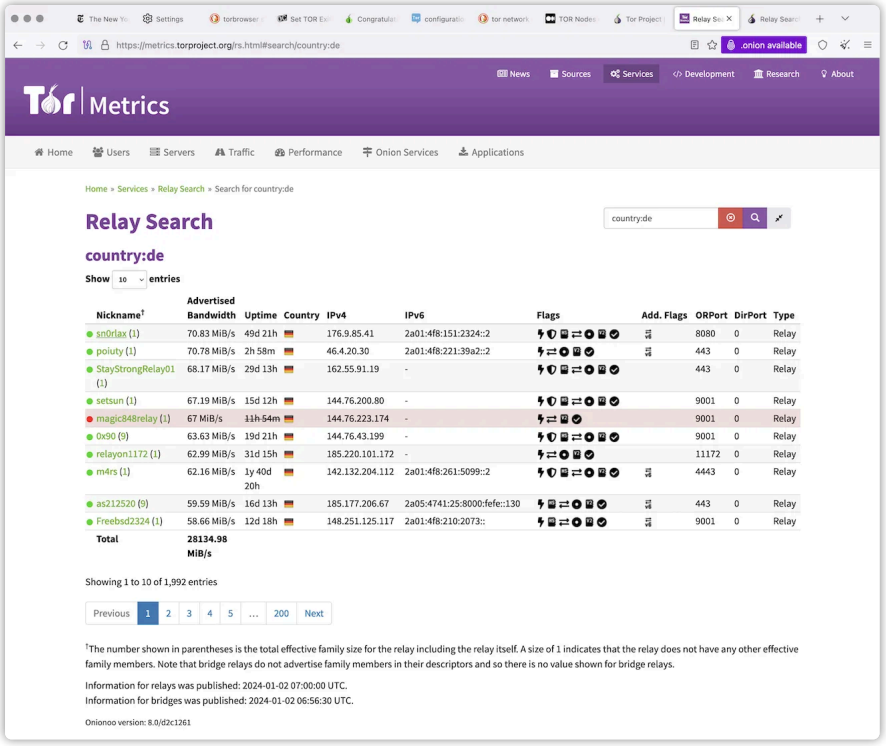
Initiierung eines Circuits (konzeptionell)



Jeder Tor-Knoten verfügt über mehrere Keys. Für den Aufbau der Verbindung werden die *Onion Keys* verwendet. Mit Hilfe dieser werden die initialen Datenpakete mittels Public-Key Kryptografie verschlüsselt. Dies wird benötigt, um den AES Key - einer pro Knoten - der für den eigentlichen Versand benötigt wird, auszuhandeln und sicher zu übertragen.

In der Grafik wird der Aufbau eines Circuits mit zwei Tor-Knoten dargestellt. Der Client kennt die Onion Keys der Tor-Knoten (`OR1` und `OR2`). Die Onion Keys werden verwendet, um die *Create* Zelle zu verschlüsseln. Der Entry Node verwendet diese Onion Keys um die *Create* Zelle zu entschlüsseln und den gemeinsamen Schlüssel zu erzeugen. Um einen längeren Pfad aufzubauen, muss der Client ggf. einfach mehrere `Extend` Nachrichten versenden. Erhält ein Knoten eine Relay Nachricht, dann kann der Knoten diese mit dem mit ihm ausgehandelten AES Key entschlüsseln und die Nachricht weiterleiten. Er kann den Inhalt (zum Beispiel eine weitere Relay Nachricht oder eine Extend Nachricht) nicht lesen.

Tor Relays in Deutschland



Flags

Beschreibung jedes Tor-Knotens in Hinblick auf die Rolle des Knotens im Tor-Netzwerk. Zum Beispiel: kann der Knoten als Entry Node verwendet werden? Ist der Knoten schnell genug um als Exit Node verwendet zu werden?

Auszug wichtiger *Flags*:

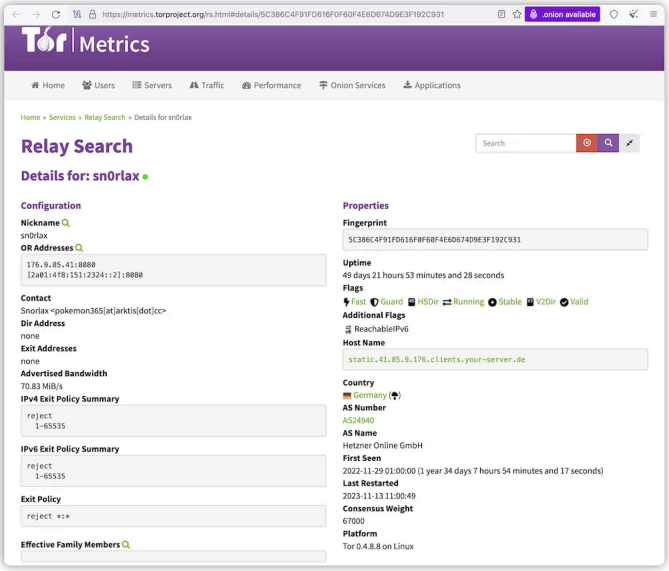
HSDir: Ein Router ist ein *v2 Hidden Service Directory*

Running: Eine Authority konnte sich innerhalb der letzten 45 Minuten mit dem Router verbinden.

Stable: die gewichtet Zeit zwischen zwei Fehlern (*weighted MTBF*) ist größer als 7 Tage oder größer als der Median aller aktiven Router.

Valid: eine Version von Tor wird ausgeführt, die von den Authorities als aktuell angesehen wird und keine bekannten Schwachstellen aufweist.

Informationen über Tor Relays



- Viele Tor Relays werden von Freiwilligen betrieben
- In Deutschland gibt es viele Relays
- Hetzner ist diesbezüglich beliebt...

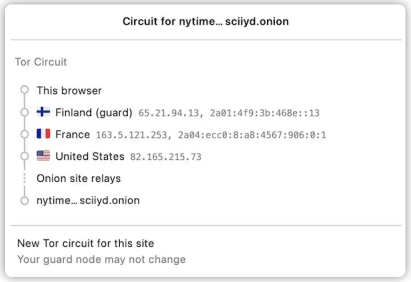
... und deswegen steht Hetzner auf der Liste der zu vermeidenden Hoster (Stand Jan. 2024).

Ein Tor-Knoten wird as `schnell` (*fast*) eingestuft, wenn er aktiv ist und eine Bandbreite von mindestens 100KB/s hat oder unter den Top 7/8tel aller bekannten aktiven Router ist.

Zum Vergleich: Die durchschnittliche Bandbreite in Deutschland ist 80Mbit/s (cf. Statista).

(Stand Jan. 2024)

Pfade, die über die ganze Welt gehen, verhindern, dass der `Entry-` und `Exit-node` beim gleichen Anbieter liegen.



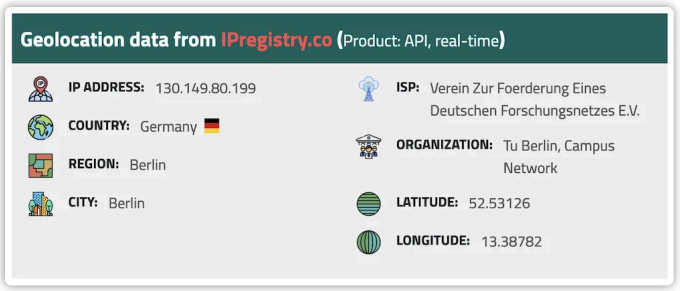
Jan. 2024 - zu vermeidende Hoster:

```
Frantech / Ponynet / BuyVM (AS53667)
OVH SAS / OVHcloud (AS16276)
Online S.A.S. / Scaleway (AS12876)
Hetzner Online GmbH (AS24940)
```

IONOS SE (AS8560)
netcup GmbH (AS197540)
Psychz Networks (AS40676)
1337 Services GmbH / RDP.sh (AS210558)

Tor Exit Nodes

Die Anzahl der *Exit Nodes* ist deutlich kleiner (2. Jan. 2024 - 1314 Einträge) als die Anzahl der Knoten. Dies liegt daran, dass die technischen Anforderungen höher sind (z. B. stabile IP Adressen) und insbesondere daran, dass die Betreiber der *Exit nodes* darauf vorbereitet sein müssen ggf. (zahlreiche) Anfragen von den Behörden zu bekommen. [1]



Reverse IP Lookup für 130.149.80.199 durchgeführt mit IP Location Service.

Die (aktuell) geringe Anzahl an Exit Nodes ist ein (steigendes?) Problem, da es dadurch ggf. einfach(er) ist den Datenverkehr zu überwachen.

---

[1] Tor Exit Node Guidelines

Tor Relays: *Exit Policy*

Jeder *Node* legt in seiner *Exit Policy* genau fest welchen Datenverkehr weiterleiten möchte:

- Es gibt offene Exit Nodes, die alle Anfragen weiterleiten.
- Es gibt Knoten, die die Daten nur an weitere Tor-Knoten weiterleiten.
- Es gibt Knoten, die nur bestimmte Dienste (z. B. HTTPs) weiterleiten.
- Es gibt „*private Exit Nodes*“, die nur zu einem bestimmten Netz Verbindungen aufbauen.

Bedeutung von Bridge Nodes (hier: WebTunnel bridges)

> The Tor Project has put out an urgent call to the privacy community asking volunteers to help deploy 200 new WebTunnel bridges by the end of the year to fight government censorship.
>
> Currently, the Tor Project operates 143 WebTunnel bridges, which help users in heavily censored regions bypass internet access restrictions and website blocks.
>
> This comes in response to increasing censorship in Russia, which Tor says currently impacts the browser's built-in censorship circumvention mechanisms, including obfs4 connections and Snowflake
>
> —28.11.2024 Tor needs 200 new WebTunnel bridges to fight censorship

> *How WebTunnels help bypass blocks*
>
> WebTunnels are a new type of bridge introduced by the Tor Project in March 2024, specifically designed to blend Tor traffic with regular web traffic, making it harder for censors to detect and block.
>
> The system achieves this by running over a web server with a valid SSL/TLS certificate, disguising Tor traffic as regular HTTPS traffic.
>
> Contrary to standard Tor bridges that use specific protocols, like obfs4, which makes their identification easier, WebTunnel bridges "hide in plain sight." This allows them to be resistant to aggressive censorship.
>
> —28.11.2024 Tor needs 200 new WebTunnel bridges to fight censorship

> Today, Tor launched a new campaign that runs until March 10, 2025, calling on volunteers to set up and maintain new WebTunnel bridges.
>
> The requirements for participation are the following:
>
> - One WebTunnel bridge per IPv4; subdomains or distinct domains allowed.
> - Provide a valid email for confirmation.
> - Keep bridges running for at least 1 year.
> - Ensure near 24/7 uptime; reboots for updates allowed.
> - **Avoid hosting with Hetzner.**
>
> —28.11.2024 Tor needs 200 new WebTunnel bridges to fight censorship

Onion Services/Hidden Services[2]

- Server, die Anfragen nur aus dem Tor-Netzwerk annehmen, werden als *Onion Services* (bzw. *Hidden Services*) bezeichnet.
- `.onion` ist eine *Pseudo*-Top-Level-Domain, die für Onion Services verwendet wird.
- Die Adresse eines Onion Services basiert auf dem öffentlichen Schlüssel des Servers.

  Beispiel: Die Onion-Adresse `https://duckduckgogg42xjoc72x3sjasowoarfbgcmvfimaftt6twagswzczad.onion/` gehört zu DuckDuckGo. Der öffentliche Schlüssel ist somit: `duckduckgogg42xjoc72x3sjasowoarfbgcmvfimaftt6twagswzczad`.
- Onion Services können nur über das Tor-Netzwerk erreicht werden.
- Liste von öffentlichen Onion-Adressen: https://community.torproject.org/onion-services/#featured-onions

  ---

  Onion-Adresse der EFF im Tor Netzwerk (Tor-Browser notwendig): https://www.iykpqm7jiradoeezzkhj7c4b33g4hbgfwelht2evxxeicbpjy44c7ead.onion
- Sicher vor Person-in-the-Middle-Angriffen zwischen Client und Onion Service

  ---

  Normalerweise sind Person-in-the-Middle-Angriffe gegen HTTPS abgesicherte Verbindungen nicht möglich; so genannte State-sponsored Actors können jedoch in bestimmten Szenarien die Infrastruktur (*Certificate Authorities* (CAs)) zur Zertifikatsausstellung und Validierung kompromittieren und damit solche Angriffe durchführen. Bei Onion Services ist dies nicht in gleicher Weise möglich, da die Adresse des Onion Services auf dem öffentlichen Schlüssel basiert.

Funktionsweise von Onion Services (Registrierung von Onion Services)

1. Der Onion Service generiert ein Schlüsselpaar (privat/öffentlich) und erstellt daraus seine Onion-Adresse.
2. Der Onion Service veröffentlicht seinen öffentlichen Schlüssel und einige *Introduction Points* (3 Stück) im Tor-Netzwerk.

   Introduction Points sind Tor-Knoten, die als Vermittler für Verbindungsanfragen dienen. Die Verbindungen zwischen dem Onion Service und den *Introduction Points* sind langlebig und werden im Directory des Tor-Netzwerks veröffentlicht.

   Die Verbindungen zwischen Onion Service und *Introduction Point* gehen über das Tor-Netzwerk (drei Hops) und sind deshalb anonymisiert.

Funktionsweise von Onion Services (Verbindung zu Onion Services)

3. Ein Client, der eine Verbindung zum Onion Service herstellen möchte, nutzt den öffentlichen Schlüssel des Onion Services und schlägt im Directory die Introduction Points nach und wählt einen *Rendezvous Point* aus (einen Tor-Knoten).
4. Der Client sendet eine Nachricht an den Onion Service über einen *Introduction Point*, in der er den ausgewählten *Rendezvous Point* angibt.
5. Der Onion Service sendet eine Nachricht an den *Rendezvous Point*, um die Verbindung mit dem Client herzustellen.
6. Der Client und der Onion Service kommunizieren.

Sicherheit von Onion Services

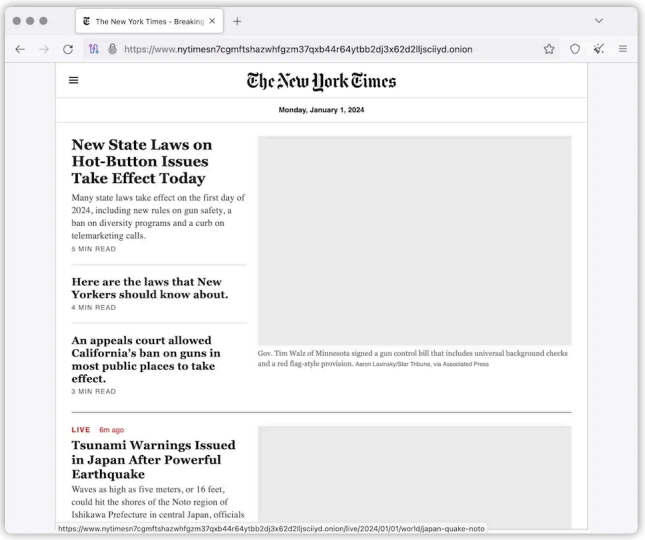
- Die Trennung von Introduction Points und Rendezvous Points sorgt dafür, dass der *Introduction Point* keine langlaufende Traffic-Analyse durchführen kann, um den Onion Service zu identifizieren.

---

[2] How Onion Services Work

Tor-Browser

Standardanwendung für den Zugriff auf das Tor-Netzwerk.



Sicherheitseinstellungen des Tor-Browsers

Standard: alle Browserfunktionen sind aktiviert.

Sicherer: JavaScript ist auf Nicht-HTTPS-Seiten deaktiviert. Wenn JavaScript aktiviert ist, dann sind die Leistungsoptimierungen deaktiviert. Audio und Video (HTML5-Medien) sowie WebGL werden nur nach Mausklick abgespielt.

Sicher: (zusätzlich) JavaScript ist immer deaktiviert. Einige Schriftarten, Symbole, mathematische Symbole und Bilder sind deaktiviert.

Das Tor-Netzwerk erlaubt ggf. das Setzen des *Exit Nodes*, um zum Beispiel geografische Sperren zu umgehen. Entsprechende Dienstanbieter können dies jedoch leicht erkennen, da die Knoten des Tor Netzwerkes bekannt sind (https://check.torproject.org/torbulkexitlist) und verweigern dann den Zugriff.

Tor

★Schützt vor der Analyse des Datenverkehrs.

Von SecureDrop wird zum Beispiel für Whistleblower empfohlen sich mit dem SecureDrop Service über Tor zu verbinden und erst dann Dokumente hochzuladen.

★Der Tor-Browser schützt relativ effektiv vor Website-Fingerprinting.

✗Teilweise sehr langsam (insbesondere bei Onion Services).

✗Monitoring des Netzwerks ist an den Grenzen möglich.

✗Ende-zu-Ende Korrelation von Datenverkehr ist möglich.

✗Die Anonymität hängt auch von der Anzahl der Nutzer ab.

Website Fingerprinting

Website Fingerprinting ermöglicht es die besuchten Websites anhand des Datenverkehrs zu identifizieren. Dabei wird nicht der Inhalt der Datenpakete analysiert, sondern die statistischen Eigenschaften des Datenverkehrs. Wie groß sind die Datenpakete (d. h. die ausgelieferten Dateien)? Wie viele Datenpakete werden wann verschickt? Wie lange dauert es bis ein Datenpaket verschickt wird (d. h. Geschwindigkeit der Webseite)? Wie lange dauert es bis ein Datenpaket ankommt?

(Cross-)Browser Fingerprinting

Durch das Sammeln vieler (auch kleiner) Informationen über den/die Browser und das Betriebssystem kann ein für praktische Zwecke hinreichend eindeutiger Fingerabdruck erstellt werden. Dieser kann dann zur Identifikation des Nutzers verwendet werden.

Kleiner Auszug aus den möglichen Informationen:

- System Fonts
- Werden Cookies unterstützt?
- Betriebssystem
- Betriebssystem Sprache
- Keyboard layout
- Art/Version des Browsers
- verfügbare Sensoren: Beschleunigungssensor, Näherungssensor, Gyroskop
- verfügbare Browser Plugins
- HTTP-Header Eigenschaften
- CPU Klasse
- HTML 5 Canvas Fingerprinting
- Unterstützung von Multitouch

Monitoring des Netzwerks an den Grenzen

Hat in der Vergangenheit dazu geführt, dass Nutzer von Tor-Netzwerken identifiziert werden konnten.

Ende-zu-Ende Korrelation von Datenverkehr

Auch als *Traffic Confirmation* bekannt. Diese Art von Attacke ist möglich, wenn *Relays* am Anfang und am Ende der Verbindung kontrolliert werden. Die Angreifer können dann den Datenverkehr an bei-

den Enden beobachten und die Datenpakete korrelieren z. B. basierend auf statistischen Informationen über die Zeitpunkte und Volumen von Datenflüssen.

✎ Übung: Tor

2.1. Onion Services

Ist es für *Onion Services* (.onion) notwendig auf HTTPS zu setzen oder reicht HTTP für eine sichere Kommunikation? Ist die Verwendung von HTTPS ggf. sogar problematisch?

2.2. TOR und DNS Lookups

Warum führt der Tor Browser keine DNS Lookups durch? Warum ist dies wichtig und wer kann/muss es stattdessen machen?

2.3. TOR abschalten?

Warum hätte das Abschalten von TOR auf kriminelle Aktivitäten im Internet vermutlich nur einen geringeren Einfluss?

✎ Übung: Tor

2.4. Wie vergleichen sich Proxies und Tor-Knoten?

2.5. Wie unterscheidet sich Tor von einem VPN?

2.6. Tor über VPN oder VPN über TOR?

Macht es Sinn ein VPN über Tor oder anders herum zu betreiben?

2.7. Kontrolle über TOR Netzwerk?

Was passiert, wenn eine Angreifer in der Lage ist 50% + 1 der *Directory Authority Server* zu kontrollieren?