Computer immer erreichbar dank Tor Hidden Service

Axel Beckert <abe@debian.org>

https://noone.org/talks/tor/

Computer immer erreichbar dank Tor Hidden Service

Folien

sind online unter https://noone.org/talks/tor/.

Agenda

- Kurz: Was ist Tor?
- Trockendurchlauf mit Beispielen
- Demo: Hidden Services auf einem Raspberry Pi einrichten.

Was ist Tor?

Tor ist ein Anonymisierungsnetzwerk, das jedes Paket über mindestens drei verschiedene, immer jeweils nur den direkten Nachbarn sehende Hops schickt. Dabei gleichen die ineinander geschachtelten Tunnel Zwiebelschalen. Daher u.a. der Name.

Es kann für alle Verbindungen genutzt werden, die auf TCP/IP aufbauen sowie eingeschränkt für DNS.

Eine Verbindung zum Tor-Netzwerk wird von einem lokalen Programm ("Daemon") zu einer der vielen Tor-Nodes aufgemacht, im Notfall auch durch NAT oder Proxies hindurch.

Entsprechend wird Tor besonders gerne von Leuten verwendet, die anonym bleiben, ihre Herkunft oder ihr Ziel verschleiern wollen.

Tor als öffentliches VPN

Tor ist aber nicht nur was für investigative Journalisten, Paranoiker oder böse Jungs. Man kann es auch als leicht einzurichtendes, öffentliches VPN nutzen um durch ein NAT-Router hindurch auf eigene Dienste zuzugreifen:

- ohne eigenen VPN-Server
- unabhängig vom Router-Modell
- ohne Anfassen der Router-Konfiguration
- durch Carrier-Grade-NAT hindurch
- auch nach Zwangstrennung weiterhin unter dem selben Namen erreichbar
- ohne Nutzung eines DynDNS-Services

Installation notwendiger Pakete

Das einzige zusätzliche Paket, das gebraucht wird, ist `tor`:

```
# apt install --no-install-recommends tor
```

Natürlich sollte auch noch die Server-Software, deren Dienst als "Hidden Service" angeboten werden soll, installiert sein, z.B. eines der Pakete `apache2`, `nginx` oder `openssh-server`.

Webserver als "Hidden Service" einrichten

- `/etc/tor/torrc` editieren und dann bei diesem Abschnitt die Kommentarzeichen entfernen: `#HiddenServiceDir /var/lib/tor/hidden_service/ #HiddenServicePort 80 127.0.0.1:80` sodass er so aussieht: `HiddenServiceDir /var/lib/tor/hidden_service/ HiddenServicePort 80 127.0.0.1:80`

- Danach Tor die Konfiguration neu einlesen lassen: `service tor reload`.

- Dadurch wurde ein neues Verzeichnis `/var/lib/tor/hidden_service/` angelegt. Darin befindet sich zwei Dateien: `# ls -l /var/lib/tor/hidden_service/ total 8 -rw------- 1 debian-tor debian-tor 23 Mar 31 22:55 hostname -rw------- 1 debian-tor debian-tor 887 Mar 31 22:55 private_key` Der generierte Hostname im Tor-Netzwerk findet sich in der Datei `/var/lib/tor/hidden_service/hostname`: `# cat /var/lib/tor/hidden_service/hostname kumd43gasfh6ywxt.onion`

Auf den neuen Tor Hidden Service zugreifen

Webzugriff ohne speziell installierte lokale Software

Wenn Privatsphäre und Vertaulichkeit kein Problem sind, kann man fremde Gateway-Server (Tor2Web) nutzen, die Webseiten im Tor-Netzwerk auch übers normale WWW erreichbar macht, z.B. als `.onion.to`, `.onion.cab` und diverse andere:

- https://kumd43gasfh6ywxt.onion.to/
- https://kumd43gasfh6ywxt.onion.cab/

Mit lokal installierter Software direkt ins Tor-Netzwerk

Am einfachsten geht das mit dem auf Firefox ESR basierenden Tor Browser.

Unter Debian und Derivaten (Raspbian, Ubuntu, Linux Mint, etc.) geht das z.B. mit dem Paket torbrowser-launcher which is in the *contrib* area of the archive. Partially, e.g. for Debian 9 Stretch, it's only available via backports:

```
# apt install torbrowser-launcher
```

Danach muss man als Benutzer `torbrowser-launcher` aufrufen. Beim ersten Aufruf wird der Tor-Browser heruntergeladen und im Home-Verzeichnis des aufrufenden Benutzers unter `~/.local/share/torbrowser/` installiert.

Anschließend kann man im Tor-Browser die in o.g. Datei angegebene Adresse per HTTP aufrufen:

- https://kumd43gasfh6ywxt.onion/

Hidden Services aus der Sicht des Servers

Den Zugriffen zuschauen

```
$ tail -F /var/log/apache2/access.log
```

Potentielle Stolperfallen

Per Tor eingehende Verbindungen kommen von `localhost` (127.0.0.1). Damit ist ein Schutz für "interne" Webseiten durch Einschränken auf `127.0.0.1` kein Schutz mehr. Für den Moment reicht wahrscheinlich noch Einschränken auf `::1` (IPv6 Localhost) aus, aber sicher nicht für immer.

SSH als Tor Hidden Service einrichten

Es wird davon ausgegangen, dass bereits ein Webserver als Hidden Service läuft, ansonsten muss man wie in o.g. Beispiel eine Zeile mit `HiddenServiceDir` ebenfalls einkommentieren und den neuen `.onion`-Hostnamen auslesen.

- `/etc/tor/torrc` editieren und dann bei diesem Abschnitt die Kommentarzeichen entfernen: `#HiddenServicePort 22 127.0.0.1:22` sodass er so aussieht: `HiddenServicePort 22 127.0.0.1:22`
- Danach Tor die Konfiguration neu einlesen lassen: `service tor reload`.

Per SSH via Tor Hidden Service auf den Rechner zugreifen

Braucht entweder das Programm `nc` (netcat; aus dem Paket `netcat-openbsd` oder das Programm `connect` aus dem Paket `connect-proxy`:

```
ssh -o 'ProxyCommand=nc -X 5 -x localhost:9050 %h %p' user@kumd43gasfh6ywxt.onion
```

oder `ssh -o 'ProxyCommand=env SOCKS5_PASSWORD="" connect -R remote -5 -S localhost:9050 %h %p' user@kumd43gasfh6ywxt.onion`

Damit man das nicht jedes Mal tippen muss, macht man sich passende Einträge in `~/.ssh/config`:

```
Host meinrechner_via_tor Hostname kumd43gasfh6ywxt.onion ProxyCommand /bin/nc -X 5 -x localhost:9050 %h %p
```

oder `Host meinrechner_via_tor Hostname kumd43gasfh6ywxt.onion ProxyCommand env SOCKS5_PASSWORD="" connect -R remote -5 -S 127.0.0.1:9050 %h %p`

Danach reicht ein `$ ssh user@meinrechner_via_tor`

SSH Hostkey verifizieren

Bei ersten Verbinden mit SSH per Tor will SSH wissen, ob man dem Hostkey traut:

```
$ ssh meinrechner_via_tor
The authenticity of host 'meinrechner_via_tor' (<no hostip for proxy command>)' can't be established.
ECDSA key fingerprint is SHA256:UJcXI5/GBJUenScbY59O5TxHpwL59UrU5SS3Iffdur4.
Are you sure you want to continue connecting (yes/no)?
```

Um diese Frage sicher beanworten zu können, führt man auf dem Rechner mit dem Hidden Service folgenden Befehl aus, um den Fingerabdruck des passenden Hostkeys (hier: ECDSA) anzuzeigen:

`$ ssh-keygen -l -f /etc/ssh/ssh_host_ecdsa_key.pub 256 SHA256:UJcXI5/GBJUenScbY59O5TxHpwL59UrU5SS3Iffdur4 root@meinrechner (ECDSA)` (Als normaler Benutzer muss man den lesbaren Public-Key (Endung `.pub`) verwenden, als Root kann man stattdessen auch den Private-Key (ohne spezielle Dateiendung) verwenden.

Sind die beiden Fingerabdrücke identisch, kann man sicher sein, dass man mit dem erwarteten Rechner verbunden ist.

Häufig gestellte Fragen

Geht das auch mit HTTPS?

Prinzipiell ja, aber demnächst hoffentlich auch mit einer der in den Browsern enthaltenen Certificate Authorities (CA), auch nicht mit *Let's Encrypt*.

Es gibt aber bereits entsprechende Diskussionen im dafür zuständigen CA/Browser-Forum, d.h. das könnte in Zukunft möglich sein.

Mit einer eigenen CA und deren Root-Zertifikat im Webbrowser ist das aber bereits jetzt möglich. (Aber alles andere als trivial.)

Geht auch UDP bzw. Mosh?

Vorerst leider nicht solange man nicht einen weiteren, TCP-basierten Tunnel (OpenVPN, sshuttle, etc.) über Tor laufen lässt, was meistens die Vorteile von UDP wieder zunichte macht.

Wie sorge ich dafür, dass niemand rauskriegen kann, wo der Server steht?

Sprengt definitiv den Rahmen des Vortrag. Dazu sind viele Details zu beachten, gerade auch bei der Konfiguration der entsprechenden Server-Software (Apache HTTPd, OpenSSH Daemon, etc.).

Alternativen

- IPv6, falls vorhanden. Meist muss man noch den Port auf dem Router in der Firewall aufmachen. Und auf einem Client ohne IPv6 dann `apt install miredo`
- Yaler.net: Server und Client sind Open Source, wird ab CHF 10 pro Jahr auch als Hosted Service angeboten. Paketierung für Debian geplant.
- Pagekite: Open Source, Programm scheint Client wie auch Server sein zu können. Hosted Service ab $4 für einen Monat, für längere Perioden preiswerter. Paket pagekite auf Debian, Raspbian, Ubuntu und anderen Debian-Derivaten.
- Zerotier: Layer 2 "VPN", welches den zentralen Server auch als Hosted Service anbietet, sogar mit Einschränkungen kostenlos oder aber auch als fertige Appliance. Paketierung der Software für Debian geplant.

Demo: Hidden Services auf Raspberry Pi einrichten

Im folgenden wird gezeigt, was dafür alles auf einem Raspberry Pi mit Raspbian Stretch (Lite) zu tun ist.

Vorbereitung

- Passendes Image herunterladen von https://www.raspberrypi.org/downloads/raspbian/
- Optional: Von mir bevorzugte Tools für diese Aufgaben (als root) auf dem (Vortrags-) Computer installieren: `apt install atool pmount screen ccze`
- Image auf Micro-SD-Karte kopieren. Entpacken und auf SD-Karte kopieren ohne Zwischenspeicher mit `acat` aus dem Paket atool:

```
# acat 2018-03-13-raspbian-stretch-lite.zip \
       2018-03-13-raspbian-stretch-lite.img \
       | dd bs=1M of=/dev/sde status=progress
```

Optional: Serielle Konsole

```
$ pmount /dev/sde1
$ echo enable_uart=1 >> /media/sde1/config.txt
$ sed -e "s/console=tty1 root=/console=tty1 console=serial0,115200 root=/" -i /media/sde1/cmdline.txt
$ pumount /dev/sde1
```

USB-zu-TTL-Seriell-Kabel wie hier abgebildet anschliessen:

- Schwarz auf Ground (GND, Pin 6)
- Weiß auf RX (Pin 8)
- Grün auf TX (Pin 10)
- Rot *nicht* anschließen.

USB-Stecker in den (Vortrags-) Computer einstecken und dort dann in einer Shell sich mit der seriellen Verbindung verbinden:

```
screen /dev/ttyUSB0 115200
```

(Ausführender Benutzer muss ggf. in der passenden Gruppe, z.B. `dialout` sein. Sonst als `root` oder mit `sudo` aufrufen. Geht nicht mit `tmux`, weil `tmux` keine Unterstützung für serielle Schnittstellen hat. Alternativen sind `minicom`, `picocom` oder `cu`.)

Karte in den Raspberry Pi einlegen und Strom anstecken und damit booten. Die Bootmeldungen sollten dann auch auf dem (Vortrags-) Computer in der Screen-Session erscheinen.

Vorbereitungen auf dem Raspberry Pi

Netzwerk-Verbindung herstellen

`sudo raspi-config` → Network Options → Wi-fi

System auf aktuellen Stand bringen und Tor installieren

Da das frisch installierte Image immer schon ein paar Tage alt ist, muß man erst einmal die Paketlisten auf aktuellen Stand bringen und vielleicht auch grade alle Sicherheitsupdates einspielen

```
$ sudo apt update
$ sudo apt upgrade
$ sudo apt install --no-install-recommends tor
```

Schonmal einen Webserver installieren

Es ist egal, welcher Webserver installiert wird oder ist. Ich mag den Apache.

```
sudo apt install --no-install-recommends apache2
```

Default-Webseite individualisieren. :-)

```
sudo sed -e 's/2 Debian Default Page/ on the Tor-as-VPN-Demo-Raspberry-Pi/' -i /var/www/html/index.html
```

Webserver als "Hidden Service" einrichten

- `/etc/tor/torrc` editieren mit z.B. `sudo $EDITOR /etc/tor/torrc` und dann bei diesem Abschnitt die Kommentarzeichen entfernen: `#HiddenServiceDir /var/lib/tor/hidden_service/ #HiddenServicePort 80 127.0.0.1:80` sodass er so aussieht: `HiddenServiceDir /var/lib/tor/hidden_service/ HiddenServicePort 80 127.0.0.1:80`

- Danach Tor die Konfiguration neu einlesen lassen: `service tor reload`.

- Dadurch wurde ein neues Verzeichnis `/var/lib/tor/hidden_service/` angelegt. Darin befindet sich zwei Dateien: `$ sudo ls -l /var/lib/tor/hidden_service/ total 8 -rw------- 1 debian-tor debian-tor 23 Mar 31 22:55 hostname -rw------- 1 debian-tor debian-tor 887 Mar 31 22:55 private_key` Den generierte Hostname im Tor-Netzwerk auslesen: `$ sudo cat /var/lib/tor/hidden_service/hostname kumd43gasfh6ywxt.onion`

Auf den neuen Tor Hidden Service zugreifen

Mit Tor unterstützender Software:

- https://kumd43gasfh6ywxt.onion/

Via fremde Gateway-Dienste:

- https://kumd43gasfh6ywxt.onion.to/
- https://kumd43gasfh6ywxt.onion.cab/

Den Zugriffen auf dem Raspberry Pi zuschauen

```
$ tail -F /var/log/apache2/access.log | ccze -A
```

(`ccze` ist ein Log-Coloriser, also ein Parser, der verschiedene Elemente von Log-Einträgen farblich hervorhebt.)

SSH als Tor Hidden Service einrichten

Als erstes SSH dauerhaft aktivieren auf dem Raspberry Pi (und ein neues Passwort setzen):

- `sudo raspi-config` → Change User Password
- `sudo raspi-config` → Interfacing Options → SSH

(Einloggen per SSH als Root mit Passwort ist in der Standard-Einstellung deaktiviert.)

Es wird davon ausgegangen, dass bereits ein Webserver als Hidden Service läuft, ansonsten muss man wie in o.g. Beispiel eine Zeile mit `HiddenServiceDir` ebenfalls einkommentieren und den neuen `.onion`-Hostnamen auslesen.

- `/etc/tor/torrc` editieren mit z.B. `sudo $EDITOR /etc/tor/torrc` und dann bei diesem Abschnitt die Kommentarzeichen entfernen: `#HiddenServicePort 22 127.0.0.1:22` sodass er so aussieht: `HiddenServicePort 22 127.0.0.1:22`
- Danach Tor die Konfiguration neu einlesen lassen: `service tor reload`.

Per SSH via Tor Hidden Service auf den Pi zugreifen

Beispiel mit netcat (`nc`):

```
ssh -o 'ProxyCommand=nc -X 5 -x localhost:9050 %h %p' pi@kumd43gasfh6ywxt.onion
```

Damit man das nicht jedes Mal tippen muss, macht man sich passende Einträge in `~/.ssh/config`:

```
Host meinpi_via_tor
        Hostname kumd43gasfh6ywxt.onion
        User pi
        ProxyCommand /bin/nc -X 5 -x localhost:9050 %h %p
```

Danach reicht ein `$ ssh meinpi_via_tor` und es wird aufgrund der Zeile `User pi` sogar automatisch der richtige Username verwendet.

SSH Hostkey verifizieren

Bei ersten Verbinden mit SSH per Tor will SSH wissen, ob man dem Hostkey traut:

```
$ ssh meinpi_via_tor
The authenticity of host 'meinpi_via_tor' (<no hostip for proxy command>)' can't be established.
ECDSA key fingerprint is SHA256:UJcXI5/GBJUenScbY59O5TxHpwL59UrU5SS3Iffdur4.
Are you sure you want to continue connecting (yes/no)?
```

Um diese Frage sicher beanworten zu können, führt man auf dem Raspberry Pi folgenden Befehl aus, um den Fingerabdruck des passenden Hostkeys (hier: ECDSA) anzuzeigen:

```
$ ssh-keygen -l -f /etc/ssh/ssh_host_ecdsa_key.pub
256 SHA256:UJcXI5/GBJUenScbY59O5TxHpwL59UrU5SS3Iffdur4 root@raspberrypi (ECDSA)
```

Sind die beiden Fingerabdrücke identisch, kann man sicher sein, dass man sich mit dem erwarteten Raspberry Pi verbunden ist.

Kontakt, Links und Ressourcen

Vortrag

- Kontakt: Axel Beckert `abe@debian.org`
- Folien: https://noone.org/talks/tor/

Links und Ressourcen

- Raspbian SD-Karten-Images: https://www.raspberrypi.org/downloads/raspbian/
- Tor: https://www.torproject.org/
- Tor Browser: https://www.torproject.org/projects/torbrowser.html
- Tor2Web: https://www.tor2web.org/
- Wikipedia-Artikel über Tor: [https://de.wikipedia.org/wiki/Tor*(Netzwerk)]*
  (https://de.wikipedia.org/wiki/Tor(Netzwerk))