Dark Pi - Den Raspberry Pi ins Tor-Netzwerk als "Hidden Node" einhängen, um weltweit drauf zugreifen zu können (nur bestimmte Ports/Dienste!)

Beispiel: Den NFC-Pi per SSH remote zugreifbar machen, um zu schauen, wann/wer die Tür geöffnet hat und ggf. die Tür auch remote zu öffnen.

Was braucht man?

```
sudo apt install tor
```

Hinweis: Nach der Installation von tor läuft ein lokaler tor-Proxy, der aber NICHT als Exit- oder Intermediate-Node, sondern nur als Client konfiguriert ist!

Konfiguration einer "Hidden Node" (einen Port im Toor-Netzwerk frei geben)

```
sudo nano /etc/tor/torrc
```

Dort nach "Hidden" suchen, und folgendes aktivieren:

```
# Hier geben wir den SSH-Port (22) auf dem eigenen Rechner im Tor-Netzwerk frei.

HiddenServiceDir /var/lib/tor/hidden_service/

HiddenServicePort 22 127.0.0.1:22
```

Speichern, dann das Verzeichnis anlegen und mit sehr restriktiven Rechten ausstatten:

```
sudo mkdir -p /var/lib/tor/hidden_service

sudo chown debian-tor:debian-tor /var/lib/tor/hidden_service

sudo chmod u=rwx,go= /var/lib/tor/hidden_service
```

Tor-Server neu starten:

```
sudo systemctl restart tor
```

Check, ob alles richtig ist: `sudo ls -la /var/lib/tor/hidden_service`

Wenn alles geklappt hat, befinden sich nun im hidden_service -Verzeichnis ein Schlüsselpaar für die Verschlüsselung im Tor-Netzwerk, sowie eine "hostname"-Datei, in der der neue Onion-Name für den Pi im Tor-Netzwerk steht!

```
sudo cat /var/lib/tor/hidden_service/hostname
```

```
fbysefsco2pft4vnunbostga6ifn6af7mivynrsdi7yleucem6lxzoid.onion
```

→Das ist der neue "Rechnername", der nur innerhalb des Tor-Netzwerks gilt.

Würde dort ein Webserver laufen (und in /etc/tor/torrc der Dienst

HiddenServicePort 80 127.0.0.1:80

freigegeben sein), könnte man nun mit dem Torbrowser direkt auf die Webseite http://fbysefsco2pft4vnunbostga6ifn6af7mivynrsdi7yleucem6lxzoid.onion gehen.

Um sich nun mit SSH über das Tor-Netzwerk mit dem Pi zu verbinden, sollte in `/etc/tor/torsocks.conf` (auf dem Client, nicht auf dem Pi!) folgende Direktive eingeschaltet werden:

```
AllowOutboundLocalhost 1
```

damit die "Namensauflösung" mit Onion-Adressen funktioniert.

Verbindung nun mit:

```
torify ssh pi@fbysefsco2pft4vnunbostga6ifn6af7mivynrsdi7yleucem6lxzoid.onion
```

Hinweis: `torify` oder `torsocks` verwenden Tor als Proxy für andere Verbindungen als http.

Tipp (optional): Um nicht immer dran denken zu müssen, "torify" zu schreiben, kann man in der .ssh/config auch einstellen, dass alle onion-Adressen generell über einen lokal laufenden Torproxy erreicht werden sollen:

```
Host *.onion
        ProxyCommand /bin/nc -xlocalhost:9050 -X5 %h %p
```

und für eine spezielle Adresse:

```
Host nfc-pi.onion
        Hostname fbysefsco2pft4vnunbostga6ifn6af7mivynrsdi7yleucem6lxzoid.onion
```

```
User pi

Port 22
```

was auch die Angabe von `pi@...` Adresse spart. :-)