Verschlüsseln und Datenschutz

- GnuPG
- GPG-Signaturen überprüfen
- Ordner verschlüsseln
- Tor Systemweit
- Verschlüsseln von Dateien und E-Mails
- VPN über Tor

GnuPG

Schnellübersicht

Befehl	Kommentar
`gpg -e -r <Empfänger-Schlüssel-ID> <Datei>`	Datei verschlüsseln
`gpg -d <verschlüsselte-Datei.gpg> > entschlüsselte-Datei`	Datei entschlüsseln
`gpg --import private-key.asc`	Schlüssel importieren
`gpg --delete-key`	Gefolgt vom Namen oder Fingerabdruck

Komplettübersicht

Befehl	Kommentar
gpg -k	Liste der Schlüssel

Verschlüsseln

Um eine Datei mit GnuPG auf einem Debian Linux zu verschlüsseln, können Sie die folgenden Schritte ausführen:

GnuPG installieren

GnuPG installieren Falls GnuPG noch nicht auf Ihrem Debian Linux-System installiert ist, können Sie es mit folgendem Befehl installieren:

```
sudo apt-get update sudo apt-get install gnupg
```

Schlüsselpaar generieren

GnuPG-Schlüsselpaar generieren Falls Sie noch keinen GnuPG-Schlüssel haben, müssen Sie zuerst ein Schlüsselpaar generieren. Hierzu können Sie den folgenden Befehl ausführen:

```
gpg --gen-key
```

Dieser Befehl startet den Assistenten zur Schlüsselerstellung, der Sie durch die Schritte zur Erstellung eines Schlüsselpaars führt. Sie werden unter anderem nach Ihrem Namen, Ihrer E-Mail-Adresse und einem Passwort gefragt. Nachdem der Assistent abgeschlossen ist, wird Ihr Schlüsselpaar generiert und in Ihrem GnuPG-Schlüsselring gespeichert.

Datei verschlüsseln

Um eine Datei zu verschlüsseln, verwenden Sie den folgenden Befehl:

```
gpg -e -r <Empfänger-Schlüssel-ID> <Datei>
```

Ersetzen Sie `<Empfänger-Schlüssel-ID>` durch die Schlüssel-ID des Empfängers, an den Sie die verschlüsselte Datei senden möchten. Sie können die Schlüssel-ID des Empfängers entweder in dessen öffentlichem Schlüssel oder in einem Keyserver finden.

Beispiel:

```
gpg -e -r john@example.com secret.txt
```

Dieser Befehl verschlüsselt die Datei `secret.txt` mit dem öffentlichen Schlüssel, der mit der E-Mail-Adresse `john@example.com` verknüpft ist.

Schritt 4: Verschlüsselte Datei speichern Nachdem die Datei erfolgreich verschlüsselt wurde, wird eine neue Datei mit der Erweiterung `.gpg` erstellt, die die verschlüsselten Daten enthält. Sie können diese Datei nun an den Empfänger senden oder anderweitig speichern.

Das war's! Sie haben erfolgreich eine Datei mit GnuPG auf Debian Linux verschlüsselt. Der Empfänger kann die Datei mit seinem privaten Schlüssel entschlüsseln, um auf den Inhalt zuzugreifen

Entschlüsseln am Mac

Verschlüsseln mit GnuPG

Um Ihren privaten Schlüssel von einem Debian Linux-System auf einen Mac zu übertragen und eine mit GnuPG verschlüsselte Datei zu entschlüsseln, können Sie die folgenden Schritte ausführen:

Privaten Schlüssel exportieren

Auf Ihrem Debian Linux-System können Sie Ihren privaten Schlüssel mit dem folgenden Befehl exportieren:

```
gpg --export-secret-keys -o private-key.asc
```

Dieser Befehl exportiert Ihren privaten Schlüssel in eine Datei mit dem Namen `private-key.asc`. Sie sollten diese Datei sicher speichern, da sie Ihren privaten Schlüssel enthält und Zugriff darauf Dritten ermöglichen würde.

Privaten Schlüssel auf den Mac übertragen

Sie können den privaten Schlüssel `private-key.asc` auf den Mac übertragen, z.B. per E-Mail, USB-Stick oder über das Netzwerk.

GnuPG auf dem Mac installieren

Falls GnuPG noch nicht auf Ihrem Mac installiert ist, können Sie es mit folgendem Befehl installieren:

```
brew install gnupg
```

Hinweis: Sie müssen Homebrew auf Ihrem Mac installiert haben, um den obigen Befehl auszuführen. Informationen zur Installation von Homebrew finden Sie unter https://brew.sh/.

Privaten Schlüssel auf dem Mac importieren

Nachdem Sie den privaten Schlüssel `private-key.asc` auf den Mac übertragen haben, können Sie ihn mit dem folgenden Befehl importieren:

```
gpg --import private-key.asc
```

GnuPG importiert den privaten Schlüssel in Ihren Schlüsselring auf dem Mac.

Datei entschlüsseln

Nachdem Sie Ihren privaten Schlüssel erfolgreich auf dem Mac importiert haben, können Sie die mit GnuPG verschlüsselte Datei entschlüsseln. Verwenden Sie dazu den folgenden Befehl:

```
gpg -d <verschlüsselte-Datei.gpg> > entschlüsselte-Datei
```

Ersetzen Sie `<verschlüsselte-Datei.gpg>` durch den Dateinamen der verschlüsselten Datei und `<entschlüsselte-Datei>` durch den gewünschten Dateinamen für die entschlüsselte Datei.

Beispiel:

```
gpg -d secret.txt.gpg > secret-decrypted.txt
```

Dieser Befehl entschlüsselt die Datei `secret.txt.gpg` und speichert den entschlüsselten Inhalt in der Datei `secret-decrypted.txt`.

Das war's! Sie haben erfolgreich Ihren privaten Schlüssel von Debian Linux auf einen Mac übertragen und eine mit GnuPG verschlüsselte Datei entschlüsselt. Beachten Sie bitte, dass der private Schlüssel sensitiv ist und mit Vorsicht behandelt werden sollte, um unbefugten Zugriff zu vermeiden

Öffentlichen Schlüssel exportieren

Um einen öffentlichen Schlüssel zu exportieren, müssen Sie den öffentlichen Schlüssel in einer geeigneten Datei speichern. Hier ist ein Beispiel, wie Sie dies mit dem GnuPG (GNU Privacy Guard) Tool tun können, das oft für die Verwaltung von Schlüsselpaaren verwendet wird:

Öffnen Sie ein Terminal auf Ihrem Linux Debian-System.
Überprüfen Sie zunächst, ob GnuPG installiert ist. Wenn es nicht installiert ist, können Sie es mit dem folgenden Befehl installieren:

```
sudo apt-get update
sudo apt-get install gnupg
```

Exportieren Sie den öffentlichen Schlüssel mit dem gpg Befehl und speichern Sie ihn in einer Datei. Hier ist ein Beispielbefehl:

```
gpg --output <Dateiname>.asc --armor --export <Key-ID>
```

Ersetzen Sie `<Dateiname>` durch den gewünschten Dateinamen, unter dem der öffentliche Schlüssel gespeichert werden soll, und `<Key-ID>` durch die ID des öffentlichen Schlüssels, den Sie exportieren möchten. Sie können die ID Ihres öffentlichen Schlüssels mit dem Befehl `gpg --list-keys` anzeigen lassen.

Beachten Sie, dass der --armor-Schalter verwendet wird, um den exportierten Schlüssel im ASCII-Rüstungformat zu speichern, das häufig für die Weitergabe von Schlüsseln über Text basierte Kanäle wie E-Mail verwendet wird.

Nachdem der Befehl ausgeführt wurde, wird der öffentliche Schlüssel in der angegebenen Datei mit der Erweiterung .asc gespeichert. Sie können die Datei mit einem Texteditor öffnen, um den exportierten öffentlichen Schlüssel anzuzeigen oder ihn auf andere Weise weiterzugeben, z. B. per

E-Mail oder auf einer Webseite.
Bitte beachten Sie, dass der öffentliche Schlüssel allgemein zugänglich ist und von jedermann verwendet werden kann, um Ihnen verschlüsselte Nachrichten zu senden oder Ihre Signaturen zu überprüfen. Schützen Sie daher den exportierten öffentlichen Schlüssel sorgfältig und übertragen Sie ihn sicher an die beabsichtigten Empfänger

Öffentlichen Schlüssel importieren

Um einen öffentlichen Schlüssel zu importieren und damit Dateien zu verschlüsseln, können Sie das GnuPG (GNU Privacy Guard) Tool verwenden. Hier ist ein Beispiel, wie Sie dies auf einem Linux Debian-System tun können:

Öffnen Sie ein Terminal auf Ihrem Linux Debian-System.
Überprüfen Sie zunächst, ob GnuPG installiert ist. Wenn es nicht installiert ist, können Sie es mit dem folgenden Befehl installieren:

```
sudo apt-get update
sudo apt-get install gnupg
```

Laden Sie den öffentlichen Schlüssel herunter oder erhalten Sie ihn von der Person, die Ihnen die Dateien verschlüsselt hat. Der öffentliche Schlüssel wird normalerweise in einer Datei mit der Erweiterung .asc oder .gpg bereitgestellt.
Importieren Sie den öffentlichen Schlüssel mit dem gpg Befehl. Hier ist ein Beispielbefehl:

```
gpg --import <Dateiname>.asc
```

Ersetzen Sie `<Dateiname>` durch den tatsächlichen Namen der Datei, die den öffentlichen Schlüssel enthält.

GnuPG wird den öffentlichen Schlüssel importieren und in Ihrem GnuPG-Schlüsselbund speichern. Sie können nun Dateien mit dem importierten öffentlichen Schlüssel verschlüsseln. Verwenden Sie dazu den gpg Befehl und geben Sie den Dateinamen als Argument an. Hier ist ein Beispielbefehl:

```
gpg --encrypt --recipient <Empfänger> <Dateiname>
```

Ersetzen Sie <Empfänger> durch den Namen oder die E-Mail-Adresse des Empfängers, für den Sie die Dateien verschlüsseln möchten, und <Dateiname> durch den Namen der Datei, die Sie

verschlüsseln möchten.

GnuPG erstellt eine verschlüsselte Version der Datei mit der Erweiterung .gpg. Diese verschlüsselte Datei kann an den Empfänger gesendet werden. Der Empfänger kann die Datei dann mit seinem privaten Schlüssel entschlüsseln.
Bitte beachten Sie, dass der private Schlüssel, der zum Entschlüsseln der Dateien benötigt wird, geheim und sicher aufbewahrt werden muss und nicht an unbefugte Personen weitergegeben werden darf

Keyserver

Unter keyserver2.gnupg.org kann man seinen Key Hochladen und auch andere finden.

Schlüssel vertrauen

1. Schlüssel überprüfen: Überprüfen Sie Ihren eigenen Schlüssel, um sicherzustellen, dass er korrekt erstellt wurde. Sie können dies tun, indem Sie den Fingerabdruck Ihres Schlüssels überprüfen und ihn mit anderen vertrauenswürdigen Quellen, z.B. Ihrer eigenen Notiz oder einem vertrauenswürdigen Freund, abgleichen.
2. Vertrauenseinstellungen ändern: Ändern Sie die Vertrauenseinstellungen für Ihren eigenen Schlüssel. Verwenden Sie dazu den Befehl `gpg --edit-key` gefolgt von Ihrem Schlüsselnamen oder Fingerabdruck. Dies öffnet den GnuPG-Editor für Ihren Schlüssel.
3. Vertrauensstellung festlegen: Im GnuPG-Editor können Sie die Vertrauensstellung für Ihren eigenen Schlüssel festlegen. Verwenden Sie den Befehl `trust`, gefolgt von der gewünschten Vertrauensstufe, die Sie für Ihren eigenen Schlüssel festlegen möchten. Zum Beispiel können Sie die Vertrauensstufe "Ultimate" (höchstes Vertrauen) festlegen, indem Sie `5` eingeben und mit Enter bestätigen.
4. Änderungen speichern: Speichern Sie die vorgenommenen Änderungen im GnuPG-Editor mit dem Befehl `save` und bestätigen Sie die Speicherung mit Enter.
5. Editor verlassen: Verlassen Sie den GnuPG-Editor mit dem Befehl `quit`, um zum Hauptterminal zurückzukehren.

Schlüssel löschen

- Geben Sie den Befehl `gpg --list-keys` ein, um eine Liste der vorhandenen Schlüssel in Ihrem GnuPG-Schlüsselbund anzuzeigen. Finden Sie den Schlüssel, den Sie entfernen möchten, anhand seines Namens oder seines Fingerabdrucks.
- Geben Sie den Befehl `gpg --delete-key` gefolgt von dem Namen oder Fingerabdruck des Schlüssels ein, den Sie entfernen möchten. Zum Beispiel: `gpg --delete-key JohnDoe` oder `gpg --delete-key 0x12345678`.

GPG-Signaturen überprüfen

Eigentlich sollte man niemals Dateien aus dem Internet herunterladen und einfach auf dem eigenen Computer installieren. Linux-Nutzer*innen sollten immer die Paketverwaltung ihres Betriebssystems benutzen, die sicherstellt dass die installierten Pakete vertrauenswürdig sind.

Manchmal ist das aber nicht möglich, zum Beispiel wenn man Windows verwendet. Besonders für sicherheitskritische Software, wie den Tor Browser oder die Passwortverwaltung KeePassXC ist es üblich, die Installations-Datei mit einer digitalen Signatur zu versehen. Dadurch können andere überprüfen, ob die Datei die sie heruntergeladen haben auch tatsächlich der*dem Entwickler*in der Software stammt und keien Schadsoftware beinhaltet.

Für die Überprüfung einer Signatur musst du GnuPG (Gnu Privacy Guard) installiert haben. Auf Linux-Systemen ist GnuPG vorinstalliert. Windows-Nutzer*innen müssen vorher GPG4Win installieren, für Mac gibt es die GPG-Suite

1. Schritt: Datei und Signatur besorgen

Lade zunächst Datei und Signatur herunter und **speichere sie im selben Verzeichnis ab**. Eine Signatur-Datei hat immer denselben Namen wie die ursprüngliche Datei mit dem Zusatz „.sig“ oder „.gpg“. Die Signatur-Datei zu einer Datei namens *Tor_Browser.exe* heißt dann etwa *Tor_Browser.exe.sig*

2. Schritt: Signatur überprüfen

2.1 Überprüfung mit grafischer Oberfläche

Es gibt für GnuPG verschiedene grafische Frontends, die es dir ersparen Befehle ins Terminal einzugeben.

Windows

Bei der Installation von GPG4Win wirst du gefragt, ob du Kleopatra oder GnuPrivacyAssistend als grafische Oberfläche mitinstallieren möchtest. *Kleopatra* ist wesentlich benutzer*innenfreundlicher und erweitert das Kontext-Menü (Rechtsklick-Menü) deines Dateibrowsers um die wichtigsten GPG-Funktionen.

Du kannst nun einfach das Verzeichnis öffnen, die Signatur-Datei mit Rechtsklick auswählen und „Entschlüsseln/Überprüfen“ auswählen.

Weiter unter Schritt 3.

Linux

Linux kannst du Kleopatra oder Seahorse verwenden.

Kleopatra

- integriert sich sehr gut in Kubuntu und andere KDE Desktops.
- installiere dafür das Paket `kleopatra`

Seahorse

- integriert sich sehr gut in die Desktop-Umgebungen GNOME (dem Standard-Desktop von Ubuntu), MATE und Cinnamon.
- Installiere dafür Paket `Seahorse` und die zugehörige Erweiterung für deinen Filebrowser, `seahorse-nautilus` (GNOME), `nemo-seahorse` (Cinnamon) `caja-seahorse` (MATE)

Nachdem du das passende Programm installiert hast, kannst du das Verzeichnis öffnen in dem die Dateien liegen und mit Rechtsklick auf die Signatur-Datei die Signatur überprüfen.

Weiter unter Schritt 3.

MacOs

Bei der Installation der GPG-Suite wird eine grafisches Oberfläche mitinstalliert. Du kannst im Filebrowser die Signatur-Datei im Kontext-Menü (Rechtsklick) auswählen und die Signatur überprüfen.

Weiter unter Schritt 3.

2.2 Überprüfung mit Terminal

Öffne das Terminal und bewege dich mit **cd** in das Verzeichnis, in dem die Datei lieg die du überprüfen willst.

```
cd /Pfad/zur/Datei/
```

Mit ls kannst du überprüfen, welche Dateien im Verzeichnis liegen (und wie genau sie heißen)

```
ls
```

Der Output des Terminals könnte nun sein: **Dateiname.exe Dateiname.exe.sig**

Jetzt kannst du die Dateien mit GnuPG (gpg) überprüfen. Der kurze Befehl dafür ist `gpg –verify`. Weil gpg aber den öffentlichen Schlüssel der Unterzeichnerin benötigt, würde es eine Fehlermeldung herausgeben dass es den öffentlichen Schlüssel nicht auf deinem Computer gefunden hat. Als zusätzliche Option kannst du deswegen –auto-key-retrieve angeben. Dadurch wird gpg die Identität der Unterzeichnerin aus der Signatur ablesen, deren Öffentlichen Schlüssel herunterladen und die Dateien überprüfen. Wichtig ist nur, dass im Anschluss daran **erst** die Signatur, und **danach** die Datei selber angegeben werden:

```
gpg  --verify  --auto-key-retrieve  Dateiname.exe.sig  Dateiname.exe
```

3. Schritt: Fingerprint überprüfen

Die Antwort von GPG sollte etwa so aussehen:

```
gpg: Good signature from "Irgendeine Identität <user@mail.org>"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: AAAA BBBB CCCC DDDD EEEE  FFFF GGGG HHHH IIII
```

Wichtig ist der Teil **Good signature from…**

Die Warnung kannst du ignorieren, das bedeuet nur das der Öffentliche Key bisher nicht von dir als *vertrauenswürdig* signiert wurde.

Aber ob diese Person auch wirklich die richtige ist, musst du selber herausfinden.Dazu schaust du auf der Webseite dieser Person nach, ob der Fingerprint `AAAA BBBB CCCC DDDD EEEE FFFF GGGG HHHH IIII` auch tatsächlich ihr gehört. Wenn es sich um Software handelt Du kannst den Fingerprint auch in verschiedene Suchmaschinen tippen, und gucken ob es vertrauenswürdige Seiten gibt die diesen Fingerprint mit derjenigen Person assoziieren.

Ordner verschlüsseln

Um einen Ordner in Linux zu verschlüsseln, kannst du das Programm "encfs" verwenden. Hier ist eine Schritt-für-Schritt-Anleitung, wie du vorgehen kannst:

1. Installiere "encfs" auf deinem Linux-System, falls es noch nicht installiert ist. Dies kannst du in der Regel über den Paketmanager deiner Distribution tun. Zum Beispiel kannst du unter Ubuntu den folgenden Befehl ausführen:

```
sudo apt-get install encfs
```

2. Erstelle einen leeren Ordner, der als "verschlüsselter" Ordner dienen wird. Du kannst dies an einem beliebigen Ort tun. Zum Beispiel:
   ```
   mkdir ~/encrypted_folder
   ```

3. Führe den folgenden Befehl aus, um den verschlüsselten Ordner einzurichten:
   ```
   encfs ~/encrypted_folder ~/decrypted_folder
   ```

   Dabei ist "~/encrypted_folder" der Pfad zum verschlüsselten Ordner und "~/decrypted_folder" der Pfad zum entschlüsselten Ordner.

4. Während der Einrichtung wirst du nach der Art der Verschlüsselung gefragt. Du kannst die Standardeinstellungen verwenden, indem du einfach Enter drückst.

5. Anschließend wirst du aufgefordert, ein Passwort für die Verschlüsselung festzulegen. Gib ein starkes Passwort ein und bestätige es.

6. Sobald die Einrichtung abgeschlossen ist, kannst du den verschlüsselten Ordner verwenden, indem du Dateien in den entschlüsselten Ordner kopierst. Die Dateien werden automatisch verschlüsselt und im verschlüsselten Ordner gespeichert.

7. Wenn du auf die verschlüsselten Dateien zugreifen möchtest, musst du den verschlüsselten Ordner zunächst mounten. Führe dazu den folgenden Befehl aus:
   ```
   encfs ~/encrypted_folder ~/decrypted_folder
   ```

8. Du wirst erneut nach dem Passwort gefragt. Gib das Passwort ein, das du bei der Einrichtung festgelegt hast.

9. Sobald der Ordner gemountet ist, kannst du auf die Dateien im entschlüsselten Ordner zugreifen und sie verwenden.

10. Wenn du die verschlüsselten Dateien nicht mehr benötigst, kannst du den Ordner unmounten, indem du den folgenden Befehl ausführst:

```
fusermount -u ~/decrypted_folder
```

Das war's! Du hast jetzt erfolgreich einen Ordner in Linux verschlüsselt.

Tor Systemweit

Um eine systemweite Verbindung zu Tor auf einem Linux-Rechner einzurichten, kannst du den Tor-Dienst installieren und konfigurieren. Hier sind die Schritte dazu:

1. **Tor installieren**:
   - Öffne ein Terminal und füge das Tor-Repository hinzu:

```
sudo add-apt-repository universe
sudo add-apt-repository ppa:deadsnakes/ppa
sudo apt update
sudo apt install tor
```

2. **Tor konfigurieren**:
   - Bearbeite die Tor-Konfigurationsdatei:

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

   - Füge folgende Zeilen hinzu, um Tor als Proxy zu verwenden:

```
SocksPort 9050
```

   - Speichere die Datei und schließe den Editor.

3. **Tor-Dienst starten**:
   - Starte den Tor-Dienst:

```
sudo systemctl start tor
sudo systemctl enable tor
```

4. **Systemweite Proxy-Einstellungen**:
   - Konfiguriere deine Anwendungen, um den Tor-Proxy zu verwenden. Du kannst dies in den Netzwerkeinstellungen deiner Anwendungen tun, indem du den Proxy auf `localhost` und den Port auf `9050` setzt.

5. **Überprüfung**:
   - Überprüfe, ob der Tor-Dienst läuft:

```
sudo systemctl status tor
```

Diese Schritte sollten dir helfen, eine systemweite Verbindung zu Tor auf deinem Linux-Rechner einzurichten [1](https://de.linux-console.net/?p=4857) [2](https://de.wikihow.com/Unter-Linux-Tor-installieren). Wenn du weitere Fragen hast oder auf Probleme stößt, lass es mich wissen!

Um Tor systemweit auf einem Debian-Rechner zu installieren und es für spezielle Anwendungen zu verwenden, kannst du die folgenden Schritte befolgen:

1. Tor installieren

Zuerst musst du das Tor-Paket installieren. Öffne ein Terminal und führe die folgenden Befehle aus:

```bash
sudo apt update
sudo apt install tor
```

2. Tor konfigurieren

Die Konfigurationsdatei für Tor befindet sich normalerweise unter `/etc/tor/torrc`. Du kannst diese Datei mit einem Texteditor deiner Wahl bearbeiten:

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

Hier kannst du verschiedene Einstellungen vornehmen, z.B. den Port, den Tor verwenden soll, oder spezifische Anwendungsregeln.

3. Tor-Dienst starten

Starte den Tor-Dienst mit dem folgenden Befehl:

```bash
sudo systemctl start tor
```

Um sicherzustellen, dass Tor beim Booten automatisch gestartet wird, kannst du den folgenden Befehl verwenden:

```bash
sudo systemctl enable tor
```

```

### 4. Anwendungen für Tor konfigurieren

Um spezielle Anwendungen über Tor zu leiten, gibt es verschiedene Ansätze, je nach Anwendung:

#### a. Browser (z.B. Firefox)

Um Firefox über Tor zu verwenden, kannst du den Tor-Browser herunterladen und installieren. Alternativ kannst du Firefox so konfigurieren, dass es den Tor-Proxy verwendet:

1. Öffne Firefox und gehe zu den Einstellungen.
2. Wähle „Netzwerkeinstellungen“ und klicke auf „Einstellungen...“.
3. Wähle „Manuelle Proxy-Konfiguration“ und setze den SOCKS-Host auf `127.0.0.1` und den Port auf `9050`.
4. Aktiviere die Option „Proxy für DNS verwenden“ und speichere die Einstellungen.

#### b. Terminal-Anwendungen

Für Terminal-Anwendungen kannst du den Proxy-Umgebungsvariablen setzen. Zum Beispiel:

```bash
export http_proxy="socks5://127.0.0.1:9050"
export https_proxy="socks5://127.0.0.1:9050"
```

Füge diese Zeilen in deine `~/.bashrc` oder `~/.bash_profile` ein, um sie dauerhaft zu machen.

5. Überprüfen der Verbindung

Um zu überprüfen, ob deine Verbindung über Tor funktioniert, kannst du die folgende URL in deinem Browser aufrufen:

```
https://check.torproject.org/
```

Diese Seite zeigt dir an, ob du über das Tor-Netzwerk verbunden bist.

6. Sicherheitshinweise

- Verwende den Tor-Browser für das Surfen im Internet, um die beste Anonymität zu gewährleisten.
- Sei vorsichtig mit der Verwendung von Plugins oder Erweiterungen, da diese deine Anonymität gefährden können.

Mit diesen Schritten solltest du in der Lage sein, Tor systemweit auf deinem Debian-Rechner zu installieren und es für spezielle Anwendungen zu verwenden.
```

Verschlüsseln von Dateien und E-Mails

Verschlüsseln mit ccrypt

Das Programm ccrypt kann direkt aus den Paketquellen installiert werden.

- Verschlüsseln:

```
ccencrypt foobar
```

- Entschlüsseln:

```
ccdecrypt foobar
```

- Entschlüsseln – nur auf die Standardausgabe:

```
ccat foobar
```

Weitere Informationen bietet die Manpage zur Anwendung.

GPG

Um ein Dokument zu verschlüsseln, benutzt man die Option `--encrypt`. Dazu müssen Sie die öffentlichen Schlüssel der vorgesehenen Empfänger haben. Sollten Sie auf der Kommandozeile den Namen der zu verschlüsselnden Datei nicht angeben, werden die zu verschlüsselnden Daten von der Standard-Eingabe gelesen. Das verschlüsselte Resultat wird auf die Standard-Ausgabe oder in die Datei, die durch die Option `--output` spezifiziert ist, geschrieben. Das Dokument wird darüberhinaus auch noch komprimiert.

```
alice$ gpg --output doc.gpg --encrypt --recipient blake@cyb.org doc
```

Mit der Option `--recipient` wird der öffentliche Schlüssel spezifiziert, mit dem das Dokument verschlüsselt werden soll. Entschlüsseln läßt sich das so verschlüsselte Dokument jedoch nur von jemandem mit dem dazugehörigen geheimen Schlüssel. Das bedeutet konsequenterweise aber auch, daß Sie selbst ein so verschlüsseltes Dokument nur wieder entschlüsseln können, wenn Sie Ihren eigenen öffentlichen Schlüssel in die Empfängerliste aufgenommen haben.

Zum Entschlüsseln einer Nachricht wird die Option `--decrypt` benutzt. Sie benötigen dazu den geheimen Schlüssel, für den die Nachricht verschlüsselt wurde und das Mantra, mit dem der geheime Schlüssel geschützt ist.

```
blake$ gpg --output doc --decrypt doc.gpg
Sie benötigen ein Mantra, um den geheimen Schlüssel zu entsperren.
Benutzer: ``Blake (Staatsanwalt) <blake@cyb.org>''
1024-Bit ELG-E Schlüssel, ID F251B862, erzeugt 2000-06-06 (Hauptschlüssel-ID B2690E6F)
```

VPN über Tor

Um einen VPN-Server mit WireGuard einzurichten, der den ausgehenden Traffic über das Tor-Netzwerk leitet, sind mehrere Schritte erforderlich. Hier ist eine allgemeine Anleitung, wie Sie dies umsetzen können:

Voraussetzungen
- Ein Server (z. B. ein VPS) mit einem Linux-Betriebssystem (z. B. Ubuntu).
- Grundkenntnisse in der Verwendung der Kommandozeile.
- Root-Zugriff auf den Server.

Schritt 1: WireGuard installieren

1. **Server aktualisieren**:
   ```bash
   sudo apt update
   sudo apt upgrade
   ```

2. **WireGuard installieren**:
   ```bash
   sudo apt install wireguard
   ```

Schritt 2: WireGuard konfigurieren

1. **Schlüssel generieren**:
   ```bash
   umask 077
   wg genkey | tee privatekey | wg pubkey > publickey
   ```

2. **WireGuard-Konfigurationsdatei erstellen**:
   Erstellen Sie eine Datei `/etc/wireguard/wg0.conf` und fügen Sie Folgendes hinzu:
   ```ini
   [Interface]
   PrivateKey = <Ihr privater Schlüssel>
   Address = 10.0.0.1/24  # VPN-Subnetz
   ListenPort = 51820

   [Peer]
   PublicKey = <Öffentlicher Schlüssel des Clients>
   AllowedIPs = 10.0.0.2/32  # IP des Clients
   ```

   ```

3. **WireGuard aktivieren**:
   ```bash
   sudo wg-quick up wg0
   ```

Schritt 3: Tor installieren

1. **Tor installieren**:
   ```bash
   sudo apt install tor
   ```

2. **Tor konfigurieren**:
   Bearbeiten Sie die Tor-Konfigurationsdatei `/etc/tor/torrc` und fügen Sie Folgendes hinzu:
   ```ini
   SocksPort 9050
   ```

3. **Tor-Dienst starten**:
   ```bash
   sudo systemctl start tor
   sudo systemctl enable tor
   ```

Schritt 4: Traffic über Tor leiten

1. **IP-Forwarding aktivieren**:
   Bearbeiten Sie die Datei `/etc/sysctl.conf` und stellen Sie sicher, dass die folgende Zeile nicht auskommentiert ist:
   ```ini
   net.ipv4.ip_forward=1
   ```
   Wenden Sie die Änderungen an:
   ```bash
   sudo sysctl -p
   ```

2. **iptables-Regeln hinzufügen**:
   Fügen Sie die folgenden iptables-Regeln hinzu, um den Traffic über Tor zu leiten:
   ```bash
   sudo iptables -t nat -A POSTROUTING -o tor0 -j MASQUERADE
   sudo iptables -A FORWARD -i wg0 -o tor0 -j ACCEPT
   sudo iptables -A FORWARD -i tor0 -o wg0 -m state --state ESTABLISHED,RELATED -j ACCEPT
   ```
   ```

3. **Routing für WireGuard konfigurieren**:
   Bearbeiten Sie die WireGuard-Konfigurationsdatei `/etc/wireguard/wg0.conf` und fügen Sie die folgende Zeile hinzu:
   ```ini
   PostUp = iptables -t nat -A POSTROUTING -o tor0 -j MASQUERADE
   PostDown = iptables -t nat -D POSTROUTING -o tor0 -j MASQUERADE
   ```

Schritt 5: Client konfigurieren

1. **Client-Schlüssel generieren** (auf dem Client):
   ```bash
   umask 077
   wg genkey | tee privatekey | wg pubkey > publickey
   ```

2. **Client-Konfigurationsdatei erstellen**:
   Erstellen Sie eine Datei (z. B. `wg0-client.conf`) und fügen Sie Folgendes hinzu:
   ```ini
   [Interface]
   PrivateKey = <Ihr privater Schlüssel des Clients>
   Address = 10.0.0.2/24  # IP des Clients

   [Peer]
   PublicKey = <Öffentlicher Schlüssel des Servers>
   Endpoint = <Server-IP>:51820
   AllowedIPs = 0.0.0.0/0  # Leitet gesamten Traffic über den VPN
   ```

3. **Client aktivieren**:
   ```bash
   sudo wg-quick up wg0-client
   ```

Schritt 6: Testen

- Überprüfen Sie, ob der Client erfolgreich mit dem Server verbunden ist.
- Testen Sie, ob der Traffic über das Tor-Netzwerk geleitet wird, indem Sie eine Website wie `check.torproject.org` besuchen.

###