stunnel

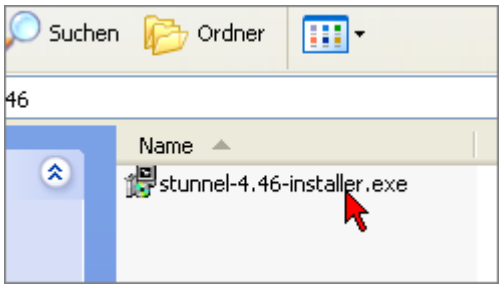
stunnel (sprich: S-Tunnel) arbeitet als universaler SSL-Tunnel zwischen dem Windows-Client und dem Server und ermöglicht somit auf einfache Weise verschlüsselte Sitzungen auch bei Programmen, die nativ keine Verschlüsselungsmethode beherrschen, der Server dies aber zulässt.

Für gesicherte Dialog- und FTP-Sitzungen empfehlen wir das freie Produkt **SSH - Secure Shell für Windows**.

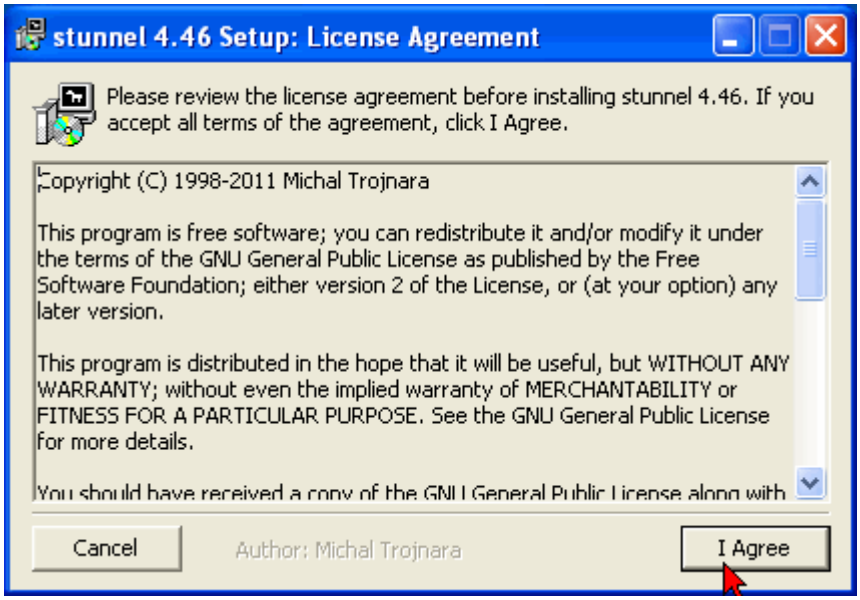
Download und Installation

Download der Installationsdatei **http://www.stunnel.org/?page=downloads** auf Ihren PC.

Speichern Sie die Datei und starten Sie die Installation durch Doppelklick auf die herunter geladene Datei.

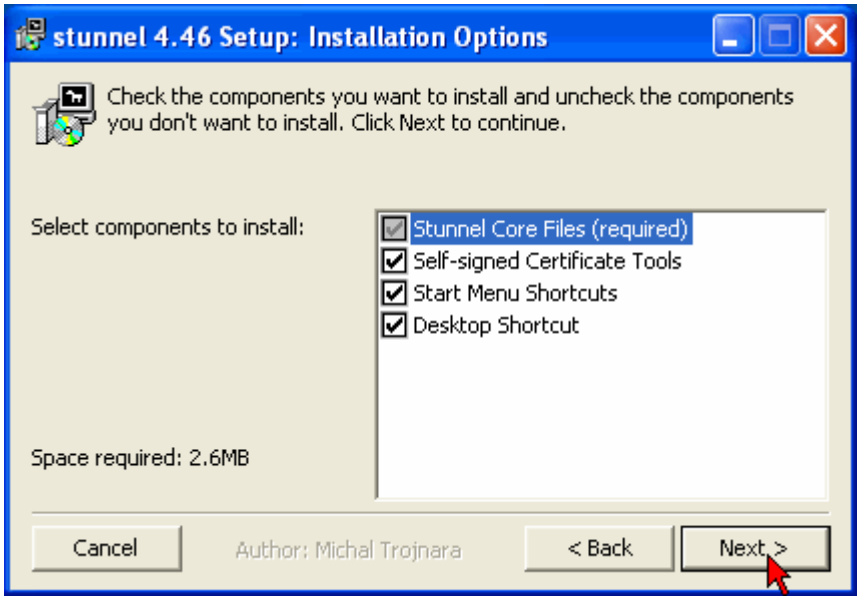


Zunächst werden Sie über die Lizenz-Situation aufgeklärt.
Es handelt sich um freie Software, die unter GNU General Public License publiziert wird.
Akzeptieren Sie die Lizenzbedinungen durch Klick auf „I Agree“.



Nun stellt das Programm die zu installierenden Komponenten vor, die Sie teilweise auch abwählen können.

Lassen Sie alle Häkchen stehen und klicken Sie auf „Next“.



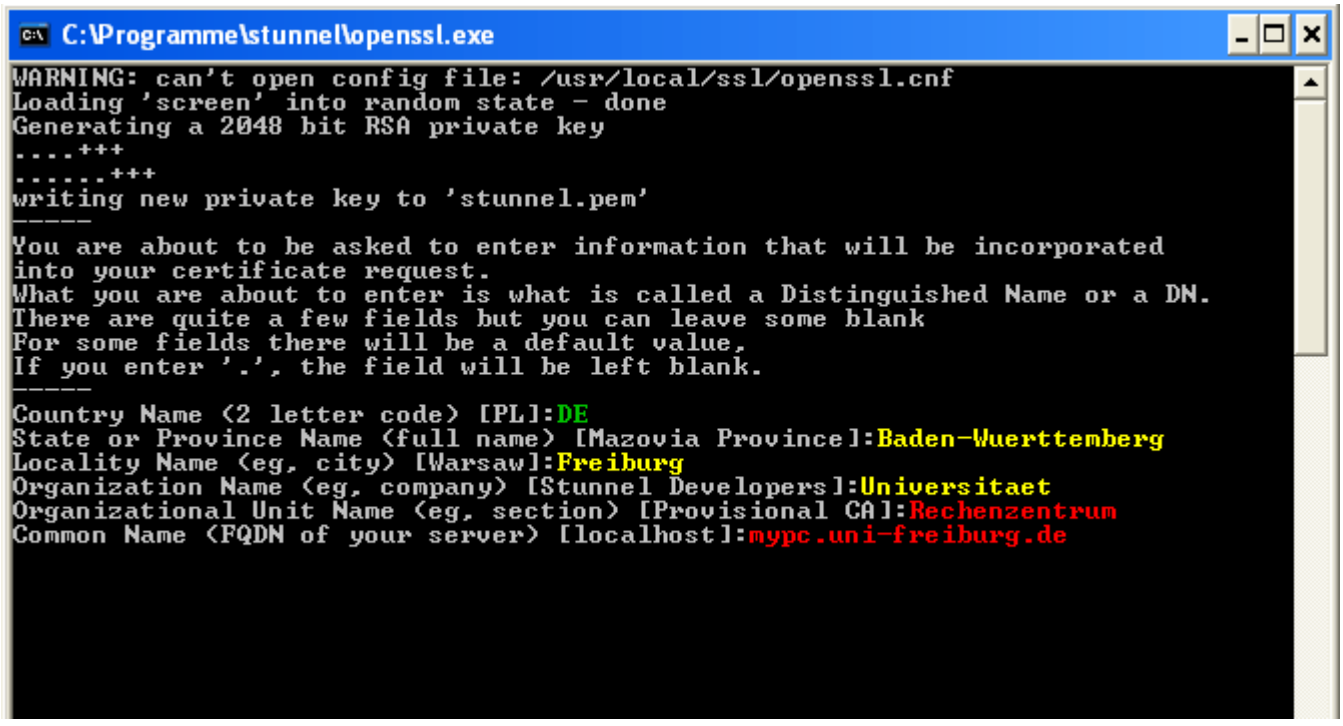
Als nächste Eingabe verlangt das Programm von Ihnen in einem Kommandozeilen-Fenster Angaben zum eigenen Zertifikat,
das dann zum Einsatz kommt, wenn diese stunnel-Installation auch als Server eingesetzt wird.

Die Beispiel-Angaben sind farbig ausgelegt und schränken den Geltungsbereich von

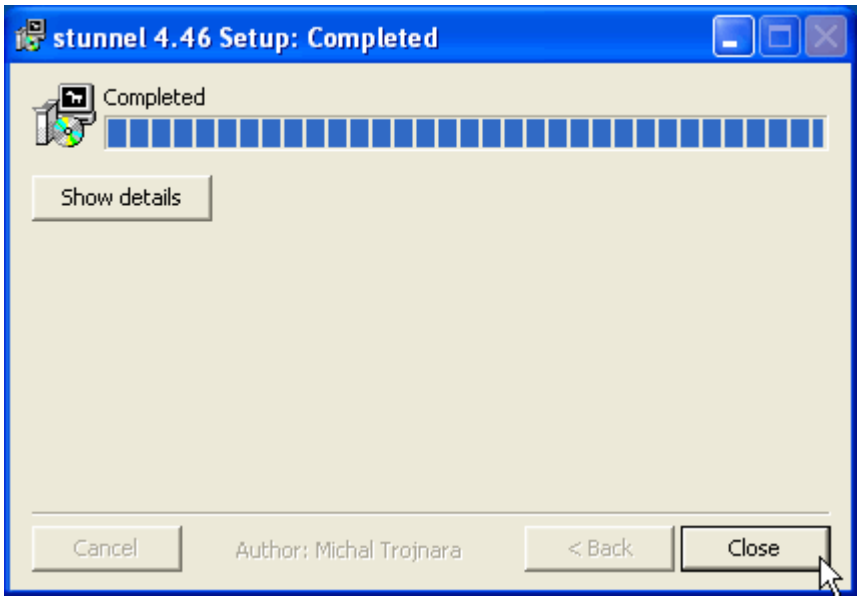
- grün (deutschlandweit) über
- gelb (Uni Freiburg) bis
- rot (der PC in Ihrer Einrichtung)

immer mehr ein. Tragen Sie hier die für Sie geltenden Werte sinngemäß ein.

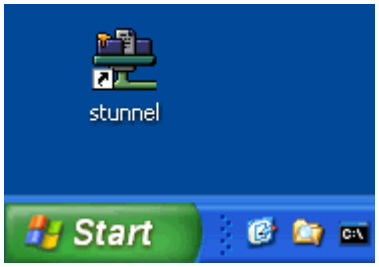
Vermeiden Sie auf jeden Fall nationale Sonderzeichen (z.B. Umlaute).
Beachten Sie in diesem Zusammenhang den letzten Abschnitt in dieser Dokumentation!



Nach der letzten Eingabe verschwindet das Kommandozeilen-Fenster wieder.
Die Installation wird fortgeführt und beendet. Klicken Sie auf „Close“.

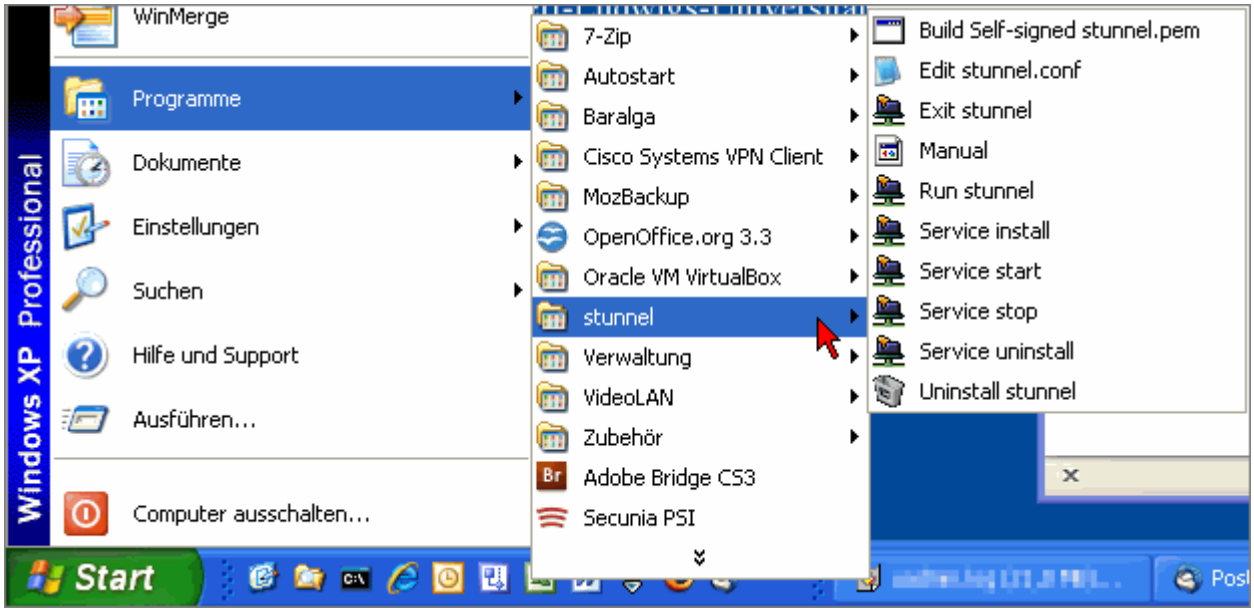


Anschließend finden Sie das Programmsymbol auf Ihrer Arbeitsoberfläche.



Programmkomponenten

Suchen Sie über das Startmenü das installierte Programm auf.
Sie finden dort eine Reihe von Einträgen vor, die im folgenden beschrieben werden.



- Mit dem ersten Eintrag **„Build Self-signed stunnel.pem“** erzeugen Sie in dem oben gezeigten Windows-Kommandozeilen-Fenster erneut ein Zertifikat für den Server-Betrieb von stunnel.
- Der Eintrag **„Edit stunnel.conf“** startet den Windows-Editor zum Bearbeiten der stunnel-Konfigurationsdatei. Diese Einstellungen sind auf jeden Fall zu überprüfen oder zu ändern, bevor Sie stunnel für Ihre Zwecke nutzen können. Im Abschnitt „Konfiguration“ (s. unten) wird näher darauf eingegangen.
- Den Punkt **„Manual“** sollten Sie zu Rate ziehen, wenn Sie das Programm konfigurieren wollen.
- Mit **„Run stunnel“** bzw. **„Exit stunnel“** starten bzw. beenden Sie das Programm manuell.
- Um stunnel als Service laufen zu lassen verwenden Sie die vier nächsten Einträge, die mit **„Service“** beginnen in folgender Weise:
  - **„Service install“** bzw. **„Service uninstall“** installiert bzw. deinstalliert das Programm als Windows-Service im Autostart-Modus; d.h. bei jedem Hochfahren den Systems wird der stunnel-Service automatisch gestartet. Nach dem Aufruf von „Service install“ wird der Service allerdings nicht sofort gestartet.
  - **„Service start“** bzw. **„Service stop“** startet bzw. stoppt den Service, belässt ihn aber im System. Nach einem manuellen Stop wird der Service auf Grund seiner Autostart-Eigenschaft beim nächsten Systemstart ebenfalls wieder gestartet.
- Mit dem letzten Eintrag „Uninstall stunnel“ entfernen Sie das Programm wieder aus dem System. Einen laufenden Service müssen Sie dazu nicht stoppen bzw. deinstallieren.

Konfiguration

Bei Internet-Verbindungen werden **Ports** zur Unterscheidung der verschiedenen Server-Dienste verwendet. Auf diese Weise kann über ein und dieselbe Verbindung z.B. Mail und WWW gleichzeitig betrieben werden. Die Portnummern von Quelle und Ziel sind Attribute der Datenpakete (Bestandteil des TCP header).

Portnummern können im Bereich von 0 bis 65535 liegen. Der Bereich unterhalb 1024 wird als „well known ports“ bezeichnet und bestimmten Diensten zugeordnet. Auf einigen Systemen sind diese Portnummern für die Benutzer-Programmierung nicht frei gegeben.

Beispiele für reservierte Portnummern:

Port	Protokoll (Verwendungszweck)
21	FTP (Dateitransfer)
22	SSH (Verschlüsselte Terminalsitzung)
23	Telnet (Internet Terminal)
25	SMTP (Mail versenden)
80	HTTP (World Wide Web)
443	HTTPS: HTTP mit SSL
110	POP3 (Mail herunterladen)
119	NNTP (Network News)
143	IMAP (Mail lesen / Ordner verwalten)
993	IMAP mit SSL
995	POP3 mit SSL

Unter **port forwarding** versteht man das Weiterleiten („tunneln“) von Datenpaketen mit einer ausgewählten (Ziel-)Portnummer über die mit stunnel aufgebaute Verbindung. Dadurch wird der gesamte Datenverkehr des zur Portnummer gehörenden Internet-Dienstes verschlüsselt.

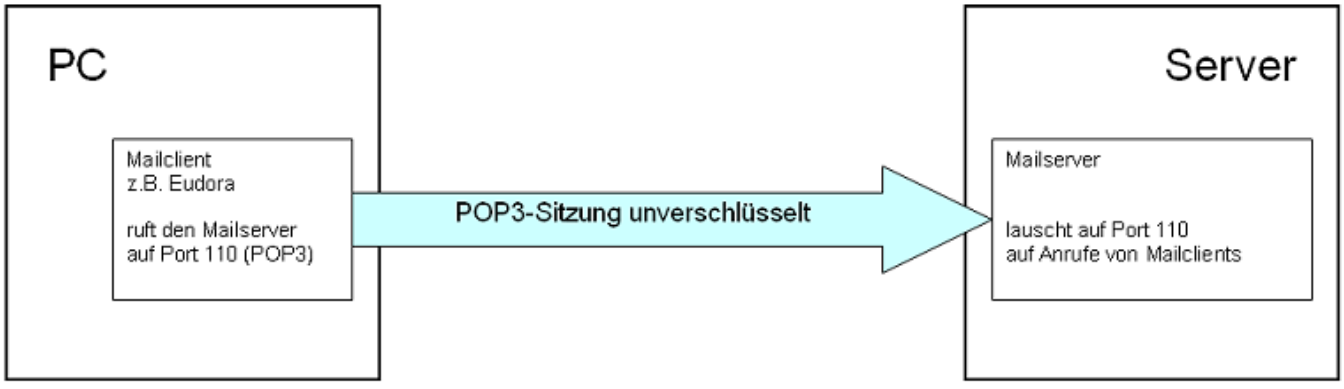
Zu diesem Zweck installiert man mit stunnel einen lokalen Transport Service Access Point (TSAP, mit zugehöriger Portnummer) der über eine verschlüsselte Verbindung mit einem entfernten (remote) TSAP verknüpft wird. Der remote Port wird vom gewünschten Dienst (z.B. POP3) festgelegt, der lokale Port kann prinzipiell frei gewählt werden, sollte aus Gründen der Übersichtlichkeit aber normalerweise identisch mit dem remote Port sein.

Dem Vorteil dieses Verfahrens (verschlüsselte Verbindung) stehen allerdings folgende Nachteile gegenüber:

- Bevor ein Dienst auf diese Weise verschlüsselt werden kann, muß zum Zielrechner eine stunnel-Verbindung aufgebaut werden.
- Der Zugang zu einem Dienst via stunnel muss mit dem Administrator des Servers vereinbart werden, da solche Zugänge nicht von vornherein eingerichtet sind bzw. nicht zum Standard gehören.

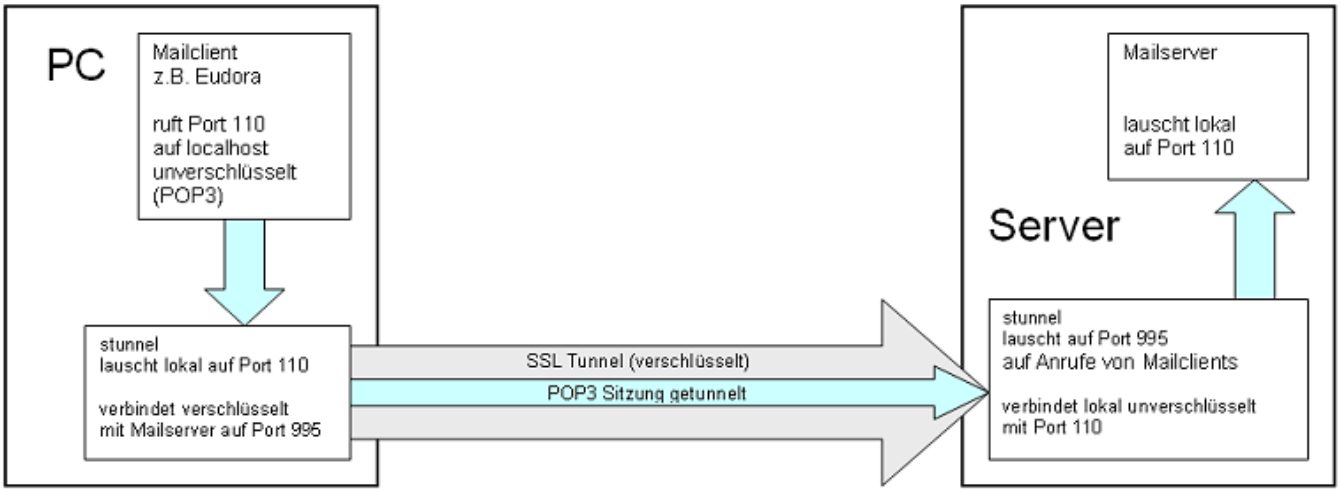
In den folgenden Abbildungen sehen Sie Beispiele für …

…eine unverschlüsselten POP3-Sitzung



…eine Verbindung zwischen stunnel im client mode und einem SSL-fähigen Server

…eine Verbindung zwischen stunnel im client mode und stunnel im server mode mit einem nicht-SSL-fähigen Server



Die bei der Installation mitgelieferte Beispiel-Konfiguration („Edit stunnel.conf“ bei den Programmkomponenten) führt Mustereinträge auf sowohl für den Client- als auch für den Server-Betrieb. Der Abschnitt für die Zertifikate ist in der Voreinstellung deaktiviert. Das bedeutet, dass im Client-Mode jedes vom Server ausgegebene Zertifikat akzeptiert wird und im Server-Mode das bei der Installation selbst erstellte Zertifikat an die Clients verschickt wird.

Eine komplette Dokumentation findet man wie schon erwähnt bei den Programmkomponenten under „Manual“.

Sicherheit

Wenn der Schutz durch Zertifikate zuverlässiger gestaltet werden soll, muss man wie folgt vorgehen:

- **Server-Mode**
  Es wird ein Zertifikat verschickt, das von einer allgemein anerkannten Zertifizierungsstelle beglaubigt wird. An der Uni Freiburg kann man sich diesbezüglich an die **Zertifizierungsstelle**

der Universität (Uni FR CA) wenden.
Mit dem Konfigurationsparameter **cert** wird stunnel der Name einer Datei mitgeteilt, in der sich, angefangen vom Wurzelzertifikat, die gesamte Zertifikatskette bis herunter zum eigenen Server-Zertifikat befindet.

- **Client-Mode**
  Zur Überprüfung der von einem Server ausgelieferten Zertifikate muss mit dem Parameter **CAfile** die Datei festgelegt werden, die die gesamte Zertifikatskette bis herunter zur ausstellenden Instanz (in Freiburg die Uni FR CA) enthält.

- **Rückruflisten**
  Um ein übriges zu tun, kann man mit einer zeitgesteuerten Task regelmäßig (und pünktlich!) die sog. Revocation Lists herunterladen und den Pfad dazu im Parameter **CRLfile** angeben. Die Rückruflisten enthalten die Zertifikate, die bis zum aktuellen Zeitpunkt für ungültig erklärt wurden.

Mit dem Parameter **verify** legen Sie fest, wie stringent stunnel mit den Zertifikaten umzugehen hat.

Wiki-Seiten zu den Zertifikaten finden Sie hier: **Zertifikate - Artikelübersicht**

Sicherheit - Artikelübersicht, Software - Artikelübersicht, Netz - Artikelübersicht