

Deckblatt: GUI-Projekt für wstunnel

Studiengang: Ingenieurinformatik

Modul: Objekt Orientierte Programmierung Vertiefung

Dozent: Pr. Memari Agater

Datum der Abgabe: 10/10/2025

Projekttitle: Entwicklung einer grafischen Benutzeroberfläche für **wstunnel**

Mitglieder der Gruppe:

- Fernandes Moreira Gustavo
- Ngami Marc Bill
- Azangue Shouna Billy Ivan
- Shaikh Abdul Kadir
- Nasrollahi Rasoul

1. Projektbeschreibung

Das Kommandozeilen-Tool **wstunnel** ermöglicht das Tunneln von TCP-Verbindungen durch einen WebSocket. Dies ist besonders nützlich, um Netzwerkbeschränkungen zu umgehen. Die Bedienung über die Kommandozeile kann jedoch für unerfahrene Benutzer umständlich und fehleranfällig sein.

Ziel dieses Projekts war es, eine intuitive grafische Benutzeroberfläche (GUI) mit Python und PyQt zu entwickeln, die die Nutzung von **wstunnel** erheblich vereinfacht. Die GUI führt den Benutzer durch die Konfiguration der notwendigen Parameter für den Server- und Client-Modus, generiert den entsprechenden Kommandozeilenbefehl und startet den Prozess. Dies senkt die Einstiegshürde und verbessert die Benutzerfreundlichkeit des Tools.

2. Installationsanleitung

Voraussetzungen:

1. **Python 3:** Muss auf dem System installiert sein.
2. **wstunnel:** Das `wstunnel` -Executable muss entweder im Systempfad (`PATH`) hinterlegt oder im selben Ordner wie die GUI-Anwendung abgelegt sein. Das Tool kann von der [offiziellen GitHub-Seite](#) heruntergeladen werden.

Installation der GUI:

1. Laden Sie den Projekt-Code aus unserem [GitHub-Repository](#) herunter.

Installieren Sie die benötigte `PyQt6`-Bibliothek mit dem folgenden Befehl im Terminal:
`pip install PyQt6`

- 2.

Starten Sie die Anwendung mit:
`python main.py`

- 3.

(Der Dateiname `main.py` ist ggf. anzupassen)

3. Beschreibung der GUI-Funktionen

Die Anwendung bietet eine klare und strukturierte Oberfläche zur Steuerung von `wstunnel` .

- **Modus-Auswahl:** Der Benutzer kann einfach zwischen dem **Server-** und **Client-Modus** wechseln.
- **Parameter-Eingabe:** Für jeden Modus gibt es dedizierte Eingabefelder und Checkboxes für alle wichtigen Parameter (z.B. Adressen, Ports, TLS-Optionen, Proxy-Einstellungen).
- **Befehls-Generierung:** Die GUI generiert in Echtzeit den vollständigen `wstunnel` -Befehl, der auf den Eingaben des Benutzers basiert. Dies dient der Transparenz und als Lernhilfe.
- **Prozess-Steuerung:** Mit "Starten"- und "Stoppen"-Buttons kann der `wstunnel` -Prozess direkt aus der Anwendung heraus gestartet und beendet werden.
- **Live-Log-Anzeige:** Eine integrierte Konsole zeigt die Live-Ausgaben des `wstunnel` -Prozesses an, was die Fehlersuche und Überwachung erleichtert.

4. Kurze Reflektion: Wer hat was gemacht?

Die Aufgabenverteilung innerhalb des Teams wurde wie folgt vorgenommen, um eine effiziente Zusammenarbeit zu gewährleisten:

- **Fernandes Moreira Gustav****Integration GUI & CLI:** Verantwortlich für die Implementierung der Logik, die die Benutzereingaben aus der GUI in gültige **wstunnel** -Kommandozeilenbefehle umwandelt und die Subprozesse startet und verwaltet.
- **Ngami Marc Bil****Dokumentation & README:** Erstellung der Projektdokumentation, einschließlich dieser Abgabe und der **README.md** Datei für das GitHub-Repository.
- **Azangue Shouna Billy Iva****Präsentation:** Ausarbeitung und Durchführung der Abschlusspräsentation des Projekts.
- **Shaikh Abdul Kadi****Client-Parameter & UI:** Entwurf und Implementierung des GUI-Teils für die Konfiguration des **wstunnel** -Clients.
- **Nasrollahi Rasou****Server-Parameter & UI:** Entwurf und Implementierung des GUI-Teils für die Konfiguration des **wstunnel** -Servers.

Die Zusammenarbeit im Team verlief koordiniert über regelmäßige Absprachen und die Nutzung des gemeinsamen GitHub-Repositorys.