

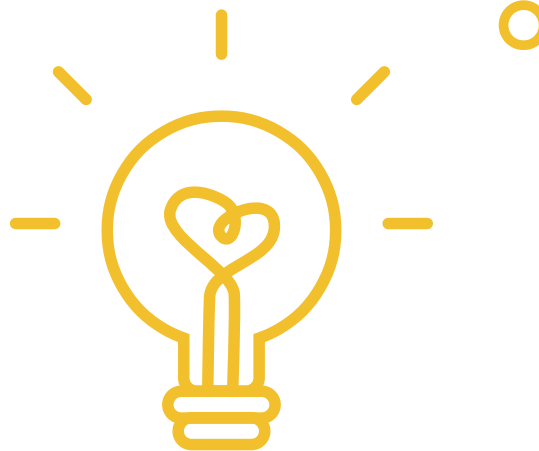
Banduras Lan Messenger (BLM)

Datentransfer in internen Netzwerken und sich daraus ergebende Probleme und Lösungen

Wer kennt es nicht: Man möchte eine Datei von Gerät A nach Gerät B versenden, aber es gibt weder ein Programm, welches das andere Gerät findet, noch die Übertragung verschlüsselt?



Wer kennt es nicht: Man möchte eine Datei von Gerät A nach Gerät B versenden, aber es gibt weder ein Programm, welches das andere Gerät findet, noch die Übertragung verschlüsselt?



01 Kriterien

- freie Software
- Integrität und Verschlüsselung
- Plattformunabhängigkeit
- mehrere Dateien senden





01

freie Software

"Frei wie in Redefreiheit, nicht wie in Freibier"

- gibt dem Benutzer die Kontrolle über das Programm zurück
- Verschlüsselung verifizierbar

02

Integrität und Verschlüsselung

- Daten nicht mit Dritten teilen
- Absender und Empfänger verifizieren
- BLM verwendet OpenSSL und eine manuelle Verifizierung der Prüfsummen von den Zertifikaten, welche zur Verschlüsselung verwendet werden



03

mehrere Dateien senden

- praktisch, besonders bei Urlaubsfotos
- bei BLM aufgrund der Architektur gegeben

04

Plattformunabhängigkeit

- in Ruby geschrieben
- verwendet als GUI-Bibliothek das Fox-Toolkit (fxruby)
- möglich, ein zusätzliches Headless-Programm zu schreiben (siehe Punkt 1)

02 Andere Lösungen

- Airdrop
- Nearby Share
- Snapdrop
- Sharik





- Andere Lösungen



Airdrop

- nicht frei, daher erfüllt es nicht meine Kriterien

Nearby Share

- nicht plattformunabhängig: Probleme mit i-Produkten, Linux und Windows
- teilweise sogar inkompatibel mit älteren Androidversionen oder älteren Smartphones

Snapdrop

- Open-Source
- Sicher
- sehr plattformunabhängig, da im Browser ausführbar
- Schwierigkeiten beim Finden von anderen Geräten, insbesondere wenn ein Gerät mit mehreren Netzwerken verbunden ist oder durch Firewall oder VPN gestört wird



Sharik

- teilweise unsicher, da offen im Netz
- keine Integritätsprüfung (verwendet eine Auslieferung über HTTP)



03 Umsetzung

- Ruby
- fxruby
- Architektur
- OpenSSL





01

Ruby

- einfache Programiersprache
- viele Bibliotheken (RubyGems)
- viele Funktionen bereits Inbuild vorhanden

02

FOX-Toolkit (fxruby)

- plattformunabhängig
- long-live project
- einfach
- Windows 95-Style





03

Architektur

- "Auftragsschlange": einzelnes Abarbeiten von Aufgaben
- Server/Client, aber Peer-to-Peer, da gleichberechtigt
- Server und Client haben nach Verbindung gleiche Aufgaben

04

OpenSSL

- man muss das Rad nicht neu erfinden
 - ausgereift
 - sicher
 - ähnlich wie bei HTTPS
 - teilweise asymmetrische Verschlüsselung
- 
- 

04 Präsentation

- Vorbereitung
- Video

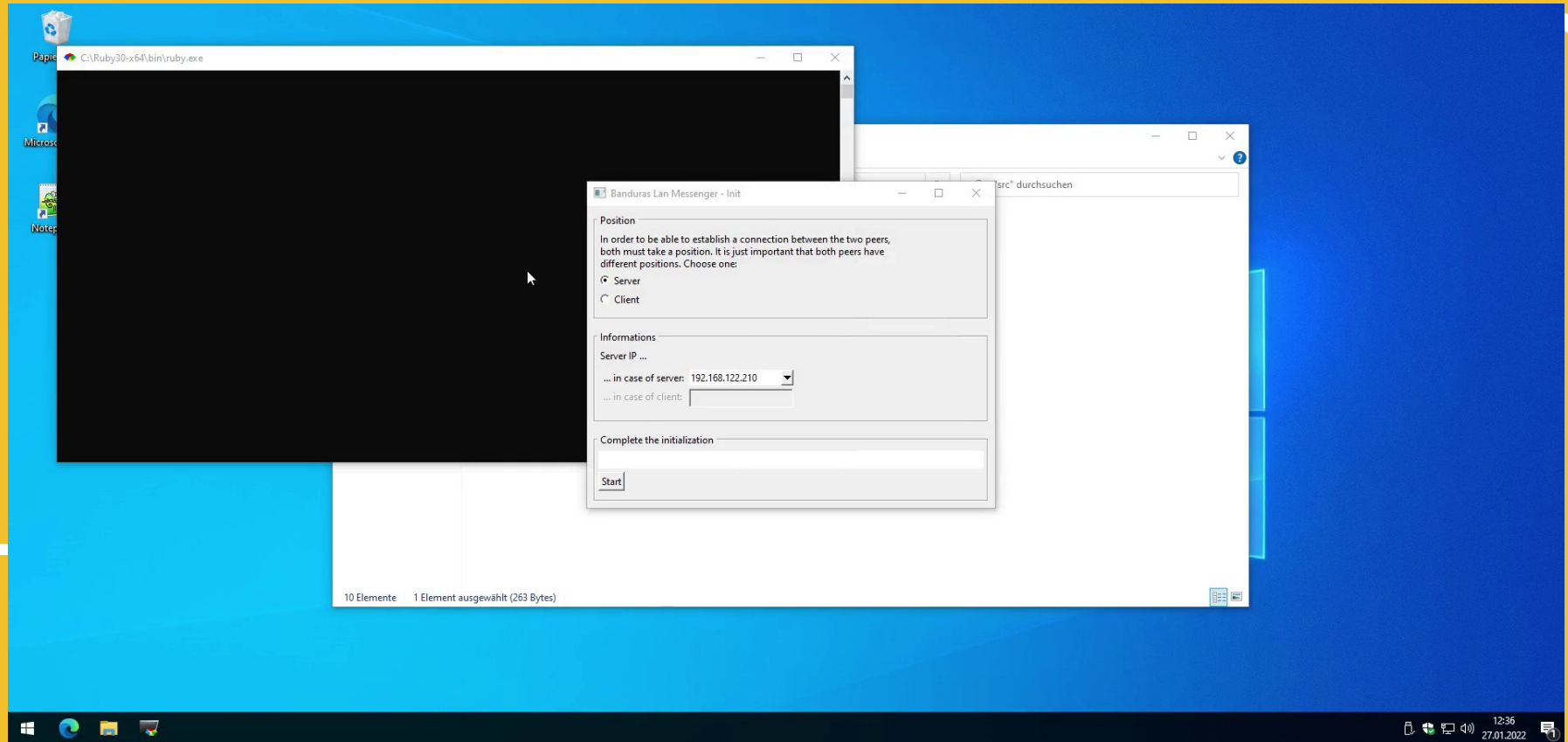


Vorbereitung

- Windows 10
 - Ruby mit dem RubyInstaller
- Debian 11
 - Ruby Version Manager (RVM)
 - Und fixruby Abhängigkeiten



Video



05 Ausblick

- Headless-Anwendung
- Stabilisierung der Anwendung
- Fehlerfindung und Behebung





Ausblick

Headless-Anwendung

- Funktionalität auf headless-Geräten
- beispielsweise Smartphone (da FOX damit nicht kompatibel ist)
- ggf. Server, allerdings unnötig, da SSH und FTP existiert

Stabilisierung der Anwendung

- mehr Möglichkeiten zur Fehlererkennung erarbeiten und implementieren
 - keine unerwarteten Abstürze oder Stehenbleiben des Programmes

Fehlerfindung und Behebung

- Versuch der automatischen Fehlerbehebung
 - Benutzer mit Entscheidungen für die Fehlerbehebung einbinden



Danke für die Aufmerksamkeit!

Banduras Lan Messenger von Marek
Küthe

Das Programm wird unter der
WTFPL veröffentlicht werden