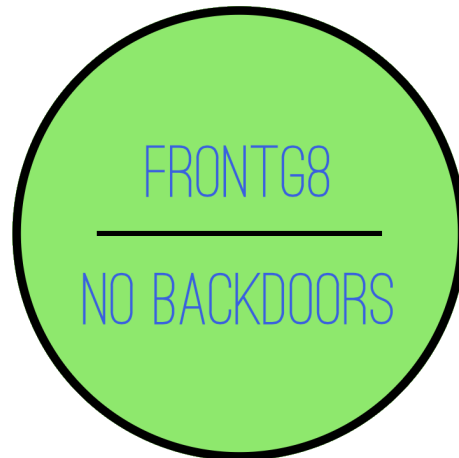




Projekt: frontg8

Handbuch pyg8

Felix Morgner - fmorgner@hsr.ch
Tobias Stauber - tstauber@hsr.ch
Ueli Bosshard - ubosshar@hsr.ch



Änderungsgeschichte

Datum	Version	Änderung	Autor
28.05.15	1.0		Ueli Bosshard



Inhalt

Änderungsgeschichte	2
Inhalt.....	3
1. Installation	4
1.1 Requirements.....	4
1.1.1 Patch für protobuf	4
1.2 Sourcecode holen.....	4
2. Konfiguration	5
2.1 Konfigurationsdatei.....	5
2.2 Zertifikate für TLS-Verbindung	5
2.3 Benutzer-Schlüsselpare.....	5
3. Benutzung	6
3.1 CLI-Modus	6
3.2 Ncurses-basiertes Userinterface	6

1. Installation

1.1 Requirements

Benötigt wird eine aktuelle Python3-Installation mit folgenden Modulen:

- protobuf3
- pycrypto
- pyelliptic
- npyscreen
- pyOpenSSL
- xmlrunner

Für die Installation der Abhängigkeiten können folgende Kommandos ausgeführt werden:

```
apt get update
apt get install git python3-pip python3-openssl
apt get install libprotobuf-dev protobuf-compiler python-protobuf
pip3 install protobuf3
pip3 install xmlrunner
pip3 install pycrypto
pip3 install pyelliptic
pip3 install pyOpenSSL
pip3 install npyscreen
```

1.1.1 Patch für protobuf

In der Python-Version von protobuf existiert ein Fehler, der uns direkt betrifft. Folgende Datei muss angepasst werden, damit pyg8 funktioniert:

/usr/local/lib/python3.4/dist-packages/protobuf3/message.py

Vor Zeile 88 einfügen:

```
if number == 0:
    result.append(number)
```

1.2 Sourcecode holen

Der Sourcecode ist frei erhältlich, ein git-Clone kann mit dem folgenden Kommando erstellt werden:

```
git clone https://source.arknet.ch/frontg8/client.git
```

2. Konfiguration

2.1 Konfigurationsdatei

In der Konfigurationsdatei „config“ sollte zumindest der „username“ und der „host“ auf die verwendete Umgebung angepasst werden.

2.2 Zertifikate für TLS-Verbindung

Damit die Kommunikation mit dem Server funktioniert ist es erforderlich, die passenden Zertifikate bereit zu stellen. Zu diesem Zweck wird auf einem Client das Hilfsscript „gen_certs.sh“ ausgeführt, welches ein CA-Zertifikat und jeweils ein Key-Pair für Server und Client im Verzeichnis „data/certs“ bereit stellt.

Die generierten Dateien werden wie folgt verwendet:

Dateiname	Verwendung
client.crt	Der Client Public-Key ist für jeden Client derselbe und muss deshalb auf jeden Client unter data/certs/client.crt bereitgestellt werden.
client.key	Der Client Private-Key ist für jeden Client derselbe und muss deshalb auf jeden Client unter data/certs/client.key bereitgestellt werden.
root.crt	Der Root-CA Public-Key muss auf den Server und jeden Client kopiert werden. Auf den Clients muss er unter data/certs/root.crt und auf dem Server unter data/dummy-ca.pem bereitgestellt werden.
server.crt	Der Server Public-Key muss auf dem Server unter data/dummy-cert.pem bereitgestellt werden.
server.key	Der Server Private-Key muss auf dem Server unter data/dummy-key.pem bereitgestellt werden.

2.3 Benutzer-Schlüsselpare

Jeder Benutzer muss ein eigenes Key-Pair generieren und den öffentlichen Teil seinen Kommunikationspartnern übergeben.

Zu Demonstrations- und Test-Zwecken werden folgende Scripts zur Verfügung gestellt: „gen_keys.py“ und „gen_contacts.py“. Diese zwei Scripts legen Zertifikate bzw. die Testuser an, welche dann anschliessend sofort im Client verfügbar sind.

Für den produktiven Gebrauch kann das Script „make_me.py“ benutzt werden:

```
./make_me.py vorname nachname
```

Der öffentliche Schlüssel ist anschliessend in der folgenden Datei zu finden:
data/keys/username.pub

3. Benutzung

3.1 CLI-Modus

Der CLI-Modus ist eher für Testzwecke gedacht und unterstützt folgende Kommandos:

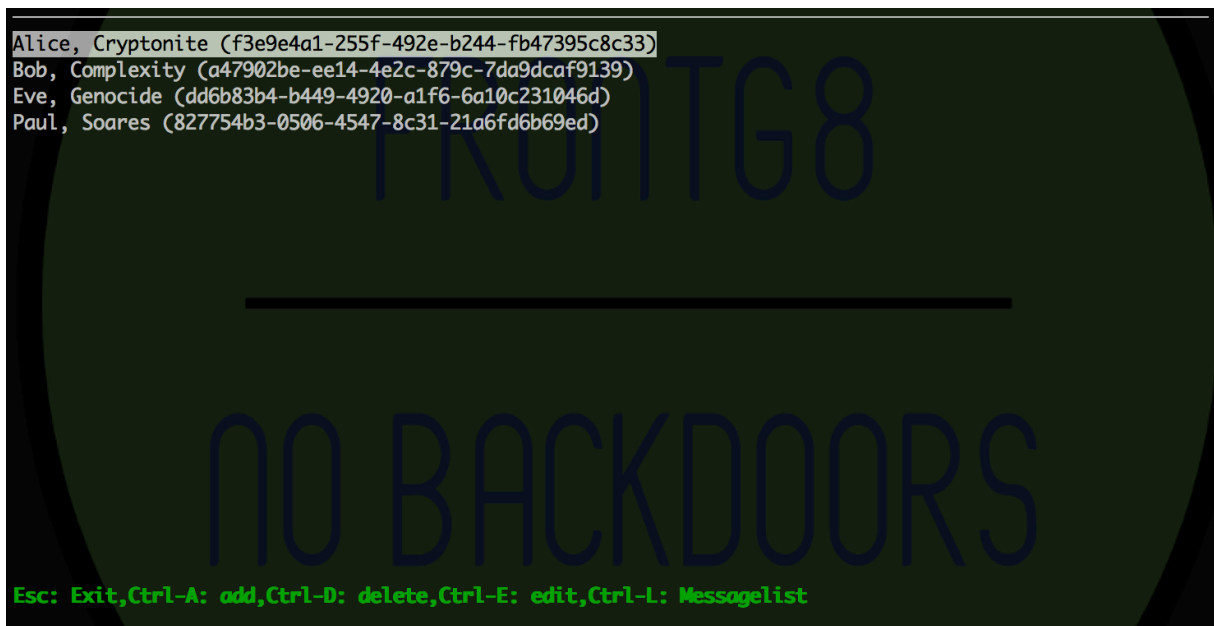
Aufruf	Verwendung
<code>./pyg8 -R</code>	Listet empfangende Meldungen auf (one shot)
<code>./pyg8 -T Nachricht</code>	Sendet eine Nachricht (one shot)
<code>./pyg8 -k -R</code>	Listet fortlaufend empfangene Meldungen auf

3.2 Ncurses-basiertes Userinterface

Wird der Client ohne Parameter gestartet, so wird zuerst nach dem Benutzerpasswort gefragt und anschliessend das Ncurses-basierte Userinterface gestartet:

```
demol@ubuntu:~/frontg8$ ./pyg8
Password for demol:
```

Als erstes erscheint die **Kontaktübersicht**:



```
Alice, Cryptonite (f3e9e4a1-255f-492e-b244-fb47395c8c33)
Bob, Complexity (a47902be-ee14-4e2c-879c-7da9dc9f9139)
Eve, Genocide (dd6b83b4-b449-4920-a1f6-6a10c231046d)
Paul, Soares (827754b3-0506-4547-8c31-21a6fd6b69ed)

Esc: Exit, Ctrl-A: add, Ctrl-D: delete, Ctrl-E: edit, Ctrl-L: Message list
```

Die verfügbaren Befehle werden unten angezeigt:

- Programm beenden mit Esc
- Ctrl-D um den ausgewählten Kontakt zu löschen
- Ctrl-E um den ausgewählten Kontakt zu editieren
- Und Ctrl-L um die Unterhaltung des ausgewählten Kontakts anzuzeigen

Falls es eine neue Nachricht vorliegt, so wird der Kontakt mit einem *-Zeichen gekennzeichnet.

In der Unterhaltung werden die bisher ausgetauschten **Nachrichten** angezeigt.



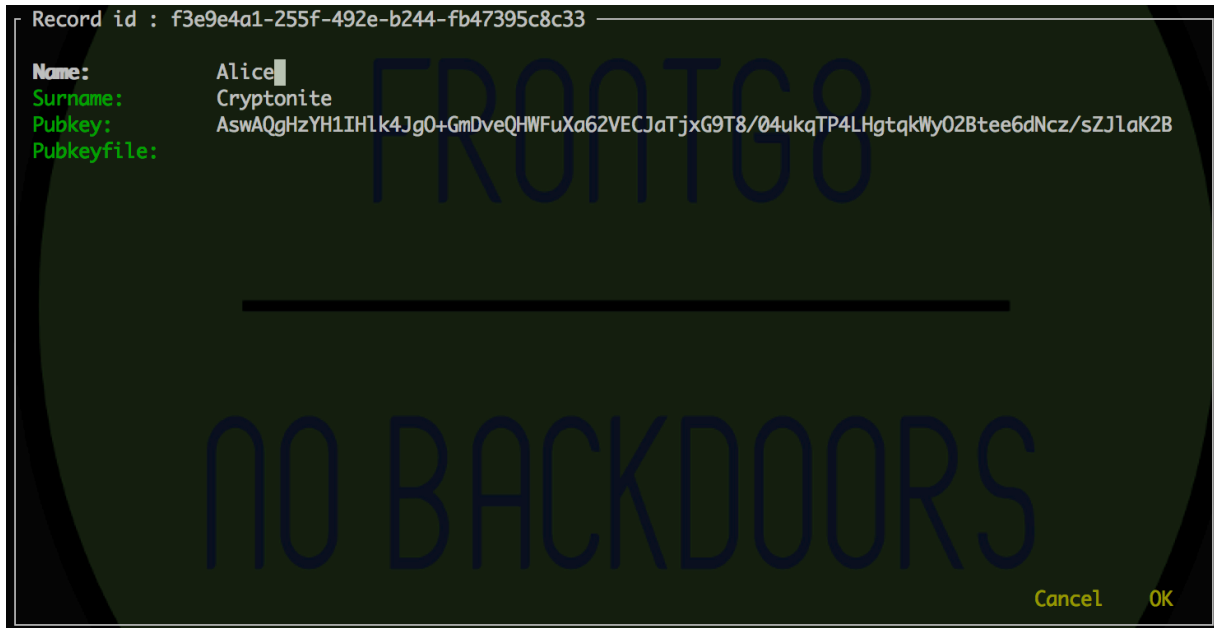
Die verfügbaren Befehle werden unten angezeigt:

- Zurück zur Kontaktübersicht mit Esc
- Ctrl-S um eine neue Nachricht zu senden

In der Kontaktübersicht kann mittels Ctrl-A ein **neuer Kontakt** erstellt werden:



In der Kontaktübersicht kann ein ausgewählter Kontakt mit Ctrl-E angepasst werden:



Record id : f3e9e4a1-255f-492e-b244-fb47395c8c33

Name: Alice

Surname: Cryptonite

Pubkey: AswAQgHzYH1IH1k4Jg0+GmDveQHWFuXa62VECJaTjxG9T8/04ukqTP4LHgtqkWy02Btee6dNcz/sZJ1aK2B

Pubkeyfile:

Cancel OK

Wird bei Pubkeyfile auf Enter gedrückt, so erscheint eine Dateiauswahl um ein Public-Key importieren zu können.