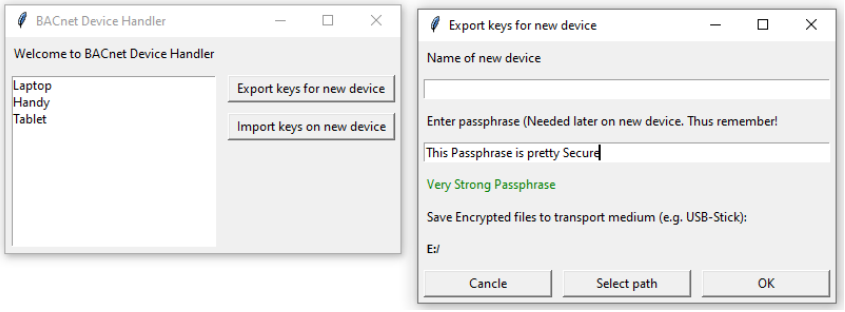
Kurs: Introduction to Internet and Security (IaS), Frühjahrsemester 2021
Dozenten: Prof. Dr. Christian Tschudin
Datum: 18. Juli 2021

Projekt Report

BACnet:
Multi Device Feed



Gruppe 10
Matthias Müller matthias01.mueller@stud.unibas.ch
Patrick Steiner pa.steiner@stud.unibas.ch
Reto Krummenacher reto.krummenacher@unibas.ch

1 Einleitung

Das BACnet ist eine dezentralisiertes Netzwerk, welche ohne das Internet und dessen gängige Protokolle (IP, TCP) die Kommunikation zwischen Teilnehmern ermöglicht. Die einfachste Art der Informationsübertagung ist die Übergabe von USB-Sticks mit den gewünschten Daten der Applikationen. Jede Applikation erstellt ihre eigenen Feeds mit einem privaten und öffentlichen Schlüssel. Eine Feed kann als erweiterbare aber nicht veränderbare verkettete Liste betrachtet werden. Anhand des öffentlichen Schlüssels sind die einzelnen Feeds unterscheidbar. Ein Nutzer im BACnet wird anhand eines sogenannten Master-Feeds[1] eindeutig identifiziert.

Bisher war es einem Teilnehmer im BACnet nicht möglich, die gleiche Applikation von verschiedenen Geräten zu bedienen. Dazu müssen einerseits die Privaten Schlüssel der Feeds geteilt werden, damit der gleiche Benutzer auf einem anderen Gerät die selben Feeds erweitern kann. Damit besteht die Gefahr von Kollisionen. Ein Nutzer schreibt von Gerät A und B gleichzeitig in einen Chat. Das dezentrale BACnet hat keine automatische Synchronisation, sprich die Daten von Gerät A werden nicht sofort auf das Gerät B übertragen. Für den Chatpartner müssen die Einträge von Gerät A und B zusammengefügt werden. Mit der BACnet Prämisse von nicht veränderbaren Feeds ist dies nicht möglich. Eine Lösung sind virtuelle Feeds.

Ziel dieses Begleitprojekts im Rahmen der Vorlesung ‘Internet and Security’ ist das Entwickeln eines Systems virtueller Feeds für das BACnet sowie einer Möglichkeit der Geräteverwaltung für den Benutzer. Der vorliegende Bericht erläutert die erarbeiteten Lösungen und beschreibt die erstellten Python-Module [2] konzeptionell. In einem ersten Teil werden die virtuellen Feeds behandelt während die Geräteverwaltung in einem zweiten Teil besprochen wird. Abschluss bildet die kritische Würdigung der Arbeiten. Im Anhang finden sich sämtliche Abbildungen sowie die Liste mit den benützten Python-libraries.

2 Virtuelle Feeds

3 Geräteverwaltung

Die Hauptaufgaben der Geräteverwaltung ist das verteilen von privaten Feed-Schlüssel auf weitere Geräte eines Nutzers. Die Klassen und Methoden finden sich im Modul ‘deviceHandler’ und stellen einerseits die Funktionalität als auch eine GUI bereit.

3.1 Funktionalität

Die gesamte Funktionalität ist in `uiFunctions.py` implementiert.

Schlüsselverteilung

Das verteilen von privaten Schlüssel ist unter dem Begriff ‘Key exchange problem’ bekannt. Dies kann mittels Verwendung eines asymmetrischen Schlüsselpaars umgangen werden. Zu sendende Nachrichten werden mit dem öffentlichen Schlüssel des Empfängers verschlüsselt. Nur mit dem privaten Schlüssel des Empfängers ist eine Entschlüsseln möglich. Im vorliegenden Fall ist dies nicht praktikabel, da das Erstellen eines asymmetrischen Schlüsselpaars den Austausch der dazu notwendigen Informationen benötigt. Im dezentralen BACnet erfordert dies ein mehrmaliges hin und her reichen eines USB-Sticks. Aus diesem Grund haben wir uns für eine symmetrischen Verschlüsselungsverfahren mit einem gemeinsamen privaten Schlüssel entschieden.

[1] Dies basiert auf Arbeiten aus dem Frühjahressemester 2020:
https://github.com/cn-uofbasel/BACnet/tree/master/20-fs-ias-lec/groups/14-feedCtrl
(letzter Aufruf 11.07.2021)

[2] Sämtlicher Code findet sich unter:
... link zum code auf GitHub

Unser Lösungsansatz ist, dass der private Schlüssel gar nicht geteilt, sonder auf jedem Gerät individuell erstellt wird. Dazu verwenden wird den gleichen Algorithmus auf beiden Geräten zusammen mit einem vom Benutzer bereitgestellten Password. Dieses Verfahren ist als 'password based key derivation' bekannten und von der IETF[3] empfohlen. Die Daten, hier die privaten Schlüssel aller Feeds eines Benutzers, werden mit dem passwortbasierten Schlüssel auf Gerät A verschlüsselt und auf einem Transportmedium gespeichert. Nur mit dem Passwort ist das Entschlüsseln auf Gerät B möglich.

Um die Stärke des eingegebenen Passworts zu bewerten, wird dessen Entropie berechnet. Die Entropie wird durch die Länge und die möglichen Anzahl der Zeichen bestimmt.[4] Je höher der Wert, desto mehr Möglichkeiten ($2^{Entropie}$) gibt es. Entsprechenden länger dauert ein Brut-Force erraten des Passworts.

Geräte hinzufügen und löschen

3.2 GUI

Die graphische Benutzeroberfläche wurde mit `tkinter` erstellt und ist in `ui.py` implementiert. Das Hauptfenster ermöglicht dem Benutzer die Funktionalitäten der Geräteverwaltung per Klick auszuführen. Darunter das Exportieren und Importieren der Feed-Schlüssel. Beide Buttons öffnen ein Dialog-Box, worin der Nutzer die benötigten Informationen (Pfad, Passwort) eingeben kann. Abbildung **??** zeigt das Hauptfenster zusammen mit dem geöffneten Exportdialog. Letzterer enthält einen Hinweis zur Stärke des eingegebenen Passworts.

Bild vom fertigen Ui

4 Fazit

[3] Internet Engineering Task Force, RFC 8018, Password-Based Cryptography Specification Version 2.1
https://datatracker.ietf.org/doc/html/rfc8018#appendix-A.2
(letzter Aufruf 11.07.2021)

[4] Berechnung der Entropy im Detail:
https://www.omnicalculator.com/other/password-entropy
(letzer Aufruf 9.7.2021)

Referenzen

- Fitzgerald, Scott und Shiloh, Michael (Hrg.), The Arduino Projects Book, 2015. (dem Arduino Starterkit beiliegend).

Anhang

Anhang 1 Python libraries

- `cryptography`
  https://cryptography.io/en/latest/ (letzter Aufruf 11.07.2021)

  `.hazmat.primitives.kdf.pbkdf2.PBKDF2HMAC`
  Erstellen eines Key auf Basis eines Passworts

  `.hazmat.primitives.ciphers.aead.AESGCM`
  Verschlüsselung mittels AES und GCM mittels Key

  `.hazmat.primitives.hashes`
  Enthält den Algorithmus SHA256

  `.exceptions.InvalidTag`
  Exception beim Entschlüssel mit falschem Passwort

- `getpass`
  Bestimmung des Benutzernamens auf dem Betriebssystem

- `json`
  Library für den Umgang mit JSON-Dateien

- `math`
  Mathematische Operationen

- `os`
  Pfad und Datei Methoden

- `re`
  Verwendung von regulären Ausdrücken

- `secrets`
  Erstellen von Test-Keys für das Testing

- `shutil`
  Kopieren von Dateien

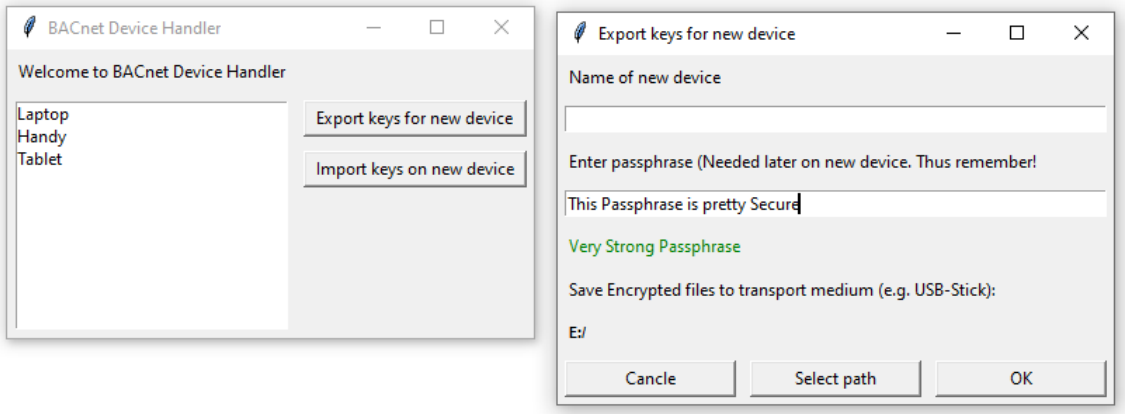
- `sys`
  Hinzufügen von Modulen zum Python runtime environment

- `tkinter`
  Erstellen einer GUI

- `unittest`
  Wird in der Klasse `TestMethods` verwendet. Diese enthält Funktionstest des Device Handlers.

Anhang 2 Abbildungen

2.1 GUI



Das Hauptfenster und der geöffnete Exportdialog, welcher unter anderem die Passwortstärke angibt.

2.2