0.1 Design des Click-Systems

(Zeichnungen und Grafiken werden später hinzugefügt.)

Akronyme und Definitionen

KDP	Key-Distribution-Protokoll
MeshNode	MeshClient/MeshRouter

0.1.1 Vorüberlegungen

- Phase 1 des mobisec ist implizit dadurch realisiert, dass der MeshNode eine TLS-Session mit dem Authentisierungsserver aufbauen darf.
- NTP4 ist auf den Knoten installiert.

Planung der Entwicklungsphasen:

1. Es soll lediglich zwei Kommunikationsteilnehmer geben – MeshNode und Authentisierungsserver –, die jeweils ein einfaches KD-Protokoll ausführung. Dieses versucht lediglich eine TLS-Session aufzubauen.
2. Das KD-Protokoll wird um die eigentliche KD-Funktionalität erweitert werden.
3. Weitere MeshNodes sollen hinzugefügt werden.

0.1.2 Netzwerktopologie

- MeshNode: mesh_node.click
- Authentisierungsserver: server.click
- mesh_node.click, server.click: beide enthalten TLS- und KD-Protokoll
- bootstraping zwischen dem Server und der MeshNode;
  - beginn mit KDP
  - falls keine tls-session vorhanden, führe tls-proto aus

  - führe eigentlichen kpd aus
  - gelingt dies, wird der client router zum mesh router (rollen-upgrade) und muss dieses Zustand irgendwie propagieren oder abrufbereitshalten für höhere Netzwerkschichten;
  - beginn der MeshRouter-Dienste:
    * routing
    * forwarding (auth pkt, data pkt)
    * Verarbeitung eigener empfangener Pkt

- Laufzeit:
  - Aus- und Weiterführung der MeshRouter-Dienste
  - (parallel dazu kann ein simpleflow betrieben zum testen)

0.1.3 Architektur

- Elemente/Elementklassen in mesh_node.click:
  - wifidev, ethencap, classifier
  - timer (for kdp)
  - tls: tls_client.cc, tls_server.cc
  - kdp: kpd_client.cc, kpd_server.cc
  - simpleflow (zum testen)/receive
  - meshrouter_services: routing, forwarding, ...
- Elemente in server.click (ähnlich wie oben):
  - wifidev, ethencap, classifier
  - tls
  - kpd
- Struktur für kdp-pkt (nach spec)

- kdp führt die Kommunikation und entscheidet, wann tls-proto verwendet werden soll. Er kümmert sich auch darum, den timer zu setzen.
- openssl: verwende bio im memory modus, um die daten in click-pkt zu verpacken

0.1.4 Kommunikation

- MeshNode startet bei der Initialisierung durch den Beginn des kdp einen erste tls-Verbindungsversuche;
- kdp setzt nach erfolgreichen erhalt des Schlüsselmaterials und timestamp den Timer für den nächsten Pull des Schlüsselmaterials.
- MeshNode besitzt einen Timer, der den nächsten Verbindungsaufbau auslöst.
- Handler einrichten für testdaten-transfer (simpleflow)

0.1.5 Testing

- fromdump
- simple flow oder Ähnliches

0.1.6 Offene Fragen

- Wie lässt sich der Zustand einer TLS-Session speichern?
- Wie lässt sich ein Schalter für die MeshRouter-Dienste realisieren?
- Wie kann ich neue WEP-Schlüssel in einem laufenden Click-Router aktivieren?
- Welches wifi-Element entspricht am ehesten meinen Anforderungen: wifidev_ap.click, wifidev_client, wifidev_linkstat?
- Sollten die WEP-Schlüssel veraltet sein, muss eine neue TLS-Verbidung aufgebaut werden. doch woran merk dies der mRouter?

- Selbstorganisierte auswahl des Authentication-Servers. Hier ist höchst wahrscheinlich ein weiteres Protokoll erforderlich.