Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

RECHENZENTRUM

78. DFN-Betriebstagung – Forum „Mobile IT“

Konsolidierung von IoT-Inseln im WLAN

Markus Krieger
Rechenzentrum Uni Würzburg
markus.krieger@uni-wuerzburg.de

Das WLAN der Uni

- Betrieb des Uni-WLANs durch das RZ
- Ca. 1000 APs (920 Cisco, 80 Aruba), Ziel Flächendeckung
- 88 Gebäude
- 5 Campusbereiche mit jeweils eigenen Controllern
- Zentrales WLAN-Management
- „Eigentlich“ nur APs der zentralen WLAN-Lösung
- „Eigentlich“ nur 2 SSIDs auf Uni-APs
  - Eduroam
  - BayernWLAN

Realität

- 10 historische SSIDs mit PSK in der zentralen Lösung
- Aktuell 3 „abgestimmte“ dezentrale APs/SSIDs
- Allein am größten Campus > 150 „Rogue APs“
  - Aus umliegenden Wohngebäuden?
  - WLAN-fähige Drucker?
  - Bildschirme mit drahtlosem Präsentieren?
  - „Ich will ohne Kabel in mein lokales Netz“?
  - IoT?

IoT-Devices sind HW-technisch oft limitiert

- Nur 2.4GHz
- Nur PSK (und Problem diesen auf Dauer geheim zu halten)
- Feature first, Security second („Das S in IoT steht für Sicherheit“)
  - Wenig Firmware-Updates => wer spielt diese konsequent ein?
  - Schnell End of Life
- Nicht für offene Uni-Umgebung geeignet => Eigene Inseln / VLANs notwendig
- Je nach Szenario ein-/ausgehende Kommunikation mit Uni / Internet nötig (Management, Cloudanbindung, ...)

=> Das Uninetz muss vor IoT geschützt werden (siehe Mirai Botnet)

=> Firewalls zwischen IoT-Inseln und Uninetz notwendig

Anfragen der vergangenen Jahre zeigen

- (Berechtigter) Bedarf an individuellen WLAN-Inseln / SSIDs steigt
  - IoT Sensoren / Aktoren (z.B. Gewächshausüberwachung)
  - Projekt-Komponenten (z.B. Roboter, Anzüge mit Sensoren, VR)
  - Projekte zur Erforschung von Technikfolgen auf menschliches Verhalten (z.B. Alexa + Sensoren + Aktoren)
  - Drahtloses Präsentieren
  - Laptops / Tablets ohne TP-Anschluss
- Teils reicht lokale Insel, teils Uplink zu Uni / Internet nötig
- Ehr High-Level Know-How bei Bedarfsträgern
- Individuelle Beratung (incl Einrichtung) zeitaufwändig
- **Skalierbare Lösung für WLAN, Netz und Sicherheit nötig**

Resultierende Probleme WLAN

- AirTime (Beacons, mehrfache Kanalbelegungen)
- Herausforderungen bei „klassischem“ PSK
- Sichere Konfiguration dezentraler „APs“?
  - Evtl. unzureichend / ungesicherter Zugang zum Uni-Netz
  - Verwaisen der Projekte
  - Dezentrales Know-How & Personalwechsel
  - Automatisches Resource Management und dezentrale APs

Resultierende Probleme Netzabsicherung / Firewalls

- Knowhow im dezentralen Bereich?
- Hoher Beratungsaufwand bei dezentral divergenten Lösungen

=> Insgesamt hoher Beratungsaufwand

=> Individuelle Lösungen skalieren nicht mehr

Lösungsansatz WLAN

- „Embrace“ durch RZ um Notwendigkeit dezentraler APs loszuwerden
- **Eine generisch konfigurierte** Zusatz-SSID für alle IoT-Inseln
- Statt „klassischem“ PSK authentifizieren die WLAN-Controller ein Gerät gegen den RADIUS-Server und werten RADIUS-Antworten aus
  - Herstellerabhängige Bezeichnung des Verfahrens
    - iPSK („identity“, z.B. Cisco)
    - mPSK („multiple“, z.B. Aruba)
    - pPSK („private“, z.B. Mikrotik)
- Im RADIUS
  - Jedes Endgerät bekommt **individuellen** PSK zugeordnet
  - Jedes Endgerät bekommt VLAN-Information hinterlegt
  - „User“-Einträge basieren auf Client MAC

Ablauf WPA2 iPSK (Cisco) / mPSK (Aruba) / pPSK (Mikrotik)

1) Client will sich mit „seinem“ PSK am WLAN anmelden

2) Controller schickt RADIUS-Anfrage mit

- User-Name = MAC
- User-Password = MAC
- Zusätzlichen AV-Paaren

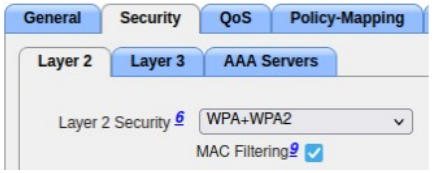
3) RADIUS: Gibt es passenden „User“ (incl evtl weiterer Check-Items)?

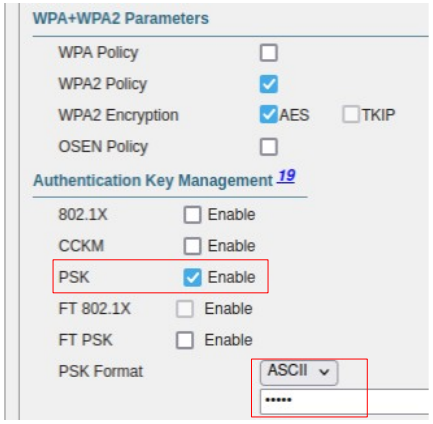
- Falls Ja „Accept“, Rückgabe des eigentlichen PSKs und evtl weiterer Reply-Items (z.b. VLAN-ID)

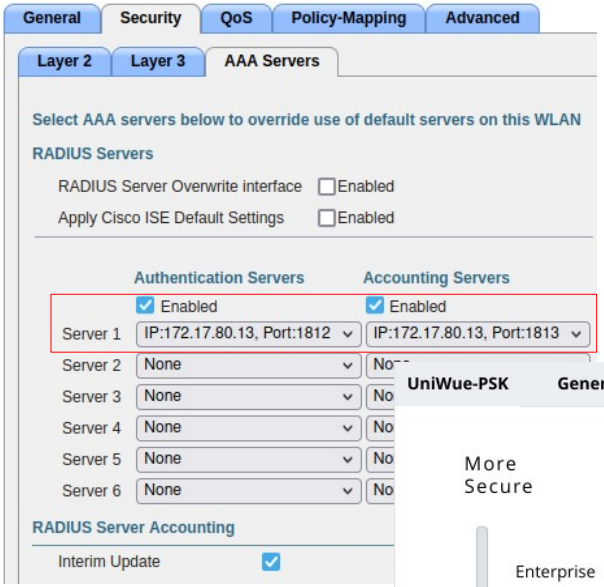
4) Client kann am Controller „joinen“, wenn der PSK des Clients zu dem vom RADIUS erhaltenen PSK passt.

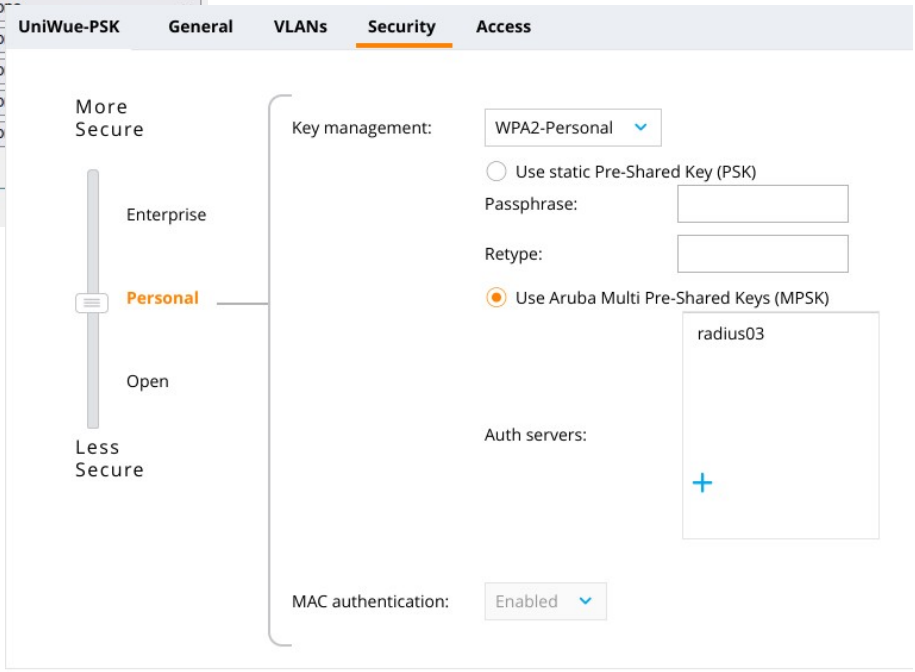
5) Evtl weitere Konfigurationsschritte am Controller (VLAN, ...)

Konfiguration auf unseren Cisco / Aruba Controllern









Beispiel aus Users-File

```
123456789abc    Cleartext-Password := "123456789abc", Called-Station-Id =~ '.*:UniWue-PSK$'
        Tunnel-Type = VLAN,
        Tunnel-Medium-Type = IEEE-802,
        Tunnel-Private-Group-Id = "1234",
        Cisco-AVPair = "psk-mode=ascii",
        Cisco-AVPair += "psk=pass1",
        Aruba-MPSK-Passphrase = "pass1"

23456789abcd    Cleartext-Password := "23456789abcd", Called-Station-Id =~ 'testap.*:UniWue-PSK$'
        Tunnel-Type = VLAN,
        Tunnel-Medium-Type = IEEE-802,
        Tunnel-Private-Group-Id = "1002",
        Cisco-AVPair = "psk-mode=ascii",
        Cisco-AVPair += "psk=pass2",
        Aruba-MPSK-Passphrase = "pass2"
```

RADIUS-Antwort zu „123456789abc“

```
User-Name = "123456789abc"
Tunnel-Type:0 = VLAN
Tunnel-Medium-Type:0 = IEEE-802
Tunnel-Private-Group-Id:0 = "1001"
Cisco-AVPair = "psk-mode=ascii"
Cisco-AVPair = "psk=pass1"
Aruba-MPSK-Passphrase = 0x7061737331
```

RADIUS-Antwort zu Client „23456789abcd“

```
User-Name = "23456789abcd"
Tunnel-Type:0 = VLAN
Tunnel-Medium-Type:0 = IEEE-802
Tunnel-Private-Group-Id:0 = "1002"
Cisco-AVPair = "psk-mode=ascii"
Cisco-AVPair = "psk=pass2"
Aruba-MPSK-Passphrase = 0x7061737332
```

Testumgebung: Hinterlegen von Endgeräten im RADIUS

- Täglich per Script neu generierte Datei „users_psk“
  - Input: CSV-Dateien der dezentralen IT-Betreuer (MAC, PSK, VLAN-ID, AP-Name)
  - Hinterlegte Liste von möglichen AP-Namen / VLAN-Kombinationen

Ziel: Selfservice-Anwendung mit interaktiver Pflege der Endgeräte

Lösungsansatz Netzanbindung / Firewall

- Aufbau eines zentralen IoT-Firewalling
  - **Eine** vom RZ supportete „einfache“ Firewall-Lösung (OPNsense?)
  - „Projekt“-Firewalls als VMs
  - Zentrale Firmwarepflege durch RZ
  - Regelpflege durch dezentrale IT-Betreuer
- Tunneln der IoT-Inseln per VXLAN an das zentrale IoT-Firewalling
  - Pro Campusbereich VXLAN Tunnel Endpunkte (VTEPs) auf Linux, um Setup zeitlich vom laufenden Backbone-Umbau zu entkoppeln
  - VM-Host(s) des zentralen IoT-Firewalings sind ebenfalls VTEPs
  - Dynamische VTEP-Kommunikation via BGP/Route Reflektoren

Ziel: Ablösung der Linux-VTEPs durch Netzkomponenten

VXLAN-Setup statisch (zu tunnelndes VLAN 100)

Endpunkt (VTEP) 1 (Host-IP 10.1.1.1)

ip link add vxlan2492 type vxlan id 2492 dev vlan2499 local 10.1.1.1 dstport 0
brctl addif br_2492 vxlan2492
brctl addiff br_2492 vlan100
bridge fdb append to 00:00:00:00:00:00 dst 10.2.2.2 dev vxlan2492

Endpunkt (VTEP) 2 (Host-IP 10.2.2.2)

ip link add vxlan2492 type vxlan id 2492 dev vlan2301 local 10.2.2.2 dstport 0
brctl addif br_2492 vxlan2492
brctl addiff br_2492 vlan100
bridge fdb append to 00:00:00:00:00:00 dst 10.1.1.1 dev vxlan2492

Auf VTEP1

bridge fdb show | grep -i 52:54:24:92:00:01
52:54:24:92:00:01 dev vxlan2492 master br_2492
52:54:24:92:00:01 dev vxlan2492 dst 10.2.2.2 self

VXLAN-Setup mit Route Reflector (RR)

Endpunkt (VTEP 1) (Host 10.1.1.1)

ip link add vxlan2492 type vxlan id 2492 dev vlan2499 local 10.1.1.1 dstport 0 nolearning
brctl addif br_2492 vxlan2492
brctl addiff br_2492 vlan100
 # Kein „bridge fdb append“ → FDB Einträge werden über RR gelernt

Endpunkt (VTEP) 2 (Host-IP 10.2.2.2)

ip link add vxlan2492 type vxlan id 2492 dev vlan2301 local 10.2.2.2 dstport 0 nolearning
brctl addif br_2492 vxlan2492
brctl addiff br_2492 vlan100
 # Kein „bridge fdb append“ → FDB Einträge werden über RR gelernt

Auf VTEP1

bridge fdb show | grep 52:54:24:92:00:01
52:54:24:92:00:01 dev vxlan2492 extern_learn master br_2492
52:54:24:92:00:01 dev vxlan2492 dst 10.2.2.2 self **extern_learn**

BGP Konfiguration mit FRR-Daemon

VXLAN Host 1

```
router bgp 65000
 bgp router-id 10.81.255.201
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor fabric peer-group
 neighbor fabric remote-as 65000
 neighbor fabric update-source 10.1.1.1
 neighbor fabric capability extended-nexthop
 neighbor 10.10.10.10 peer-group fabric
 !
 address-family l2vpn evpn
  neighbor fabric activate
  advertise-all-vni
 exit-address-family
```

VXLAN Host 2

```
router bgp 65000
 bgp router-id 10.88.1.221
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor fabric peer-group
 neighbor fabric remote-as 65000
 neighbor fabric update-source 10.2.2.2
 neighbor fabric capability extended-nexthop
 neighbor 10.10.10.10 peer-group fabric
 !
 address-family l2vpn evpn
  neighbor fabric activate
  advertise-all-vni
 exit-address-family
```

BGP Konfiguration mit FRR-Daemon

Route Reflector

```
router bgp 65000
 bgp router-id 10.81.255.202
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 bgp cluster-id 10.81.255.202
 neighbor fabric peer-group
 neighbor fabric remote-as 65000
 neighbor fabric update-source 10.10.10.10
 neighbor fabric capability extended-nexthop
 bgp listen range 10.1.1.0/24 peer-group fabric
 bgp listen range 10.2.2.0/24 peer-group fabric
 !
 address-family l2vpn evpn
  neighbor fabric activate
  neighbor fabric route-reflector-client
 exit-address-family
```

Zusammenfassend erhoffen wir uns im IoT-Szenario durch

- Eine SSID, WPA-iPSK / mPSK, VLAN-Zuordnung per RADIUS
  - Bessere Nutzung der Airtime, weniger Kanalwiederverwendung
  - Einheitliche, zentral administrierte Sicherheitseinstellungen im WLAN
  - Skalierende & übersichtlichere Konfiguration der WLAN-Controller
  - Skalierende „Inventarisierung“ der IoT-Devices und individuelle PSKs

- Zentralen Firewall-Host und VXLAN
  - Einheitlich supportbares Firewallsetup (incl Beratungsaufwand)
  - Vom allgemeinen Backbone-Umbau entkoppelte Anbindung des FW-Hosts
  - Mit VXLAN leichte Erweiterung der zu tunnelnden VLANs

Zusammenfassend (Fortsetzung)

- Generell
  - Niedrigeren Beratungsaufwand zu den Fachbereichen
  - Keine / weniger Notwendigkeit für dezentrale APs / Firewalls
  - Fachbereiche können sich auf ihre Anwendungen konzentrieren
  - **Höhere Zufriedenheit aller Beteiligten (Fachbereiche und RZ)**