Vendorneutrales NAC mit 802.1X und FreeRADIUS: Konfiguration und Erfahrungsbericht

18. März 2026

Simon Ruderich

Regionales Rechenzentrum Erlangen (RRZE)
Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU)

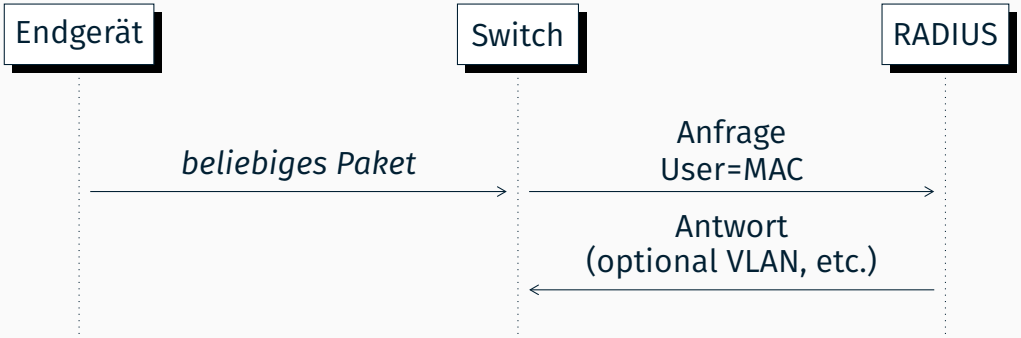
Inhalt

Konzepte

Network Access Control (NAC)

- Fokus heute: Netzwerkzugang im LAN
- Ziele
  - Herstellerunabhängigkeit $\rightarrow$ digitale Souveränität
  - Automatisierung: keine Ports „von Hand“ schalten
  - Erhöhte Sicherheit: nur authentifizierte Geräte im LAN

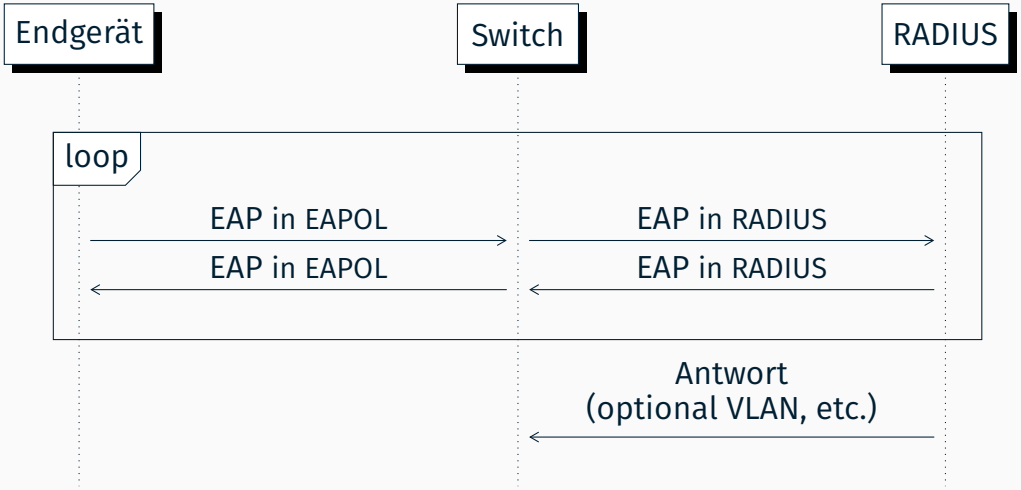
NAC per MAC



Vorteile braucht keine Konfiguration auf dem Endgerät

Nachteile keine wirkliche Authentifizierung; „unsicher“

NAC per 802.1X



Vorteile „sicher“; verschiedene Verfahren (EAP-TLS, etc.)

Nachteile Konfiguration auf Endgerät nötig

Implementierung

Ausgangslage

- Viele Geräte im DHCP erfasst
- Großer Aufwand durch manuelles Schalten von Ports
- (Neue) Standorte haben alle Ports gepatcht

- Zentrale Verwaltung der Endgeräte
- Mehr Sicherheit für manche Netze gewünscht

- „OpenSpace“ Arbeitsplätze
- Nicht autorisierte Geräte in „Gast-VLAN“

Ausgangslage

- Viele Geräte im DHCP erfasst
- Großer Aufwand durch manuelles Schalten von Ports
- (Neue) Standorte haben alle Ports gepatcht

  → NAC per MAC

- Zentrale Verwaltung der Endgeräte
- Mehr Sicherheit für manche Netze gewünscht

  → NAC per 802.1X (EAP-TLS, Zertifikat)

- „OpenSpace“ Arbeitsplätze
- Nicht autorisierte Geräte in „Gast-VLAN“

  → Kombination aus NAC und Private-VLANs

Ansatz: NAC per MAC

- Datenbank auf RADIUS-Server
- Tabelle mit (MAC, VLAN, Client, Switch, Port)
  - Statische Konfiguration: Netz $\mapsto$ VLAN und Client
  - MACs aus DHCP
- RADIUS-Server prüft Tabelle bei Anfrage
  - Liefert VLAN zurück
  - Switch/Port können weiter einschränken;
    angedacht für mehr Sicherheit bei Geräten die stationär sind

Ansatz: NAC per 802.1X (EAP-TLS, Zertifikat)

- Datenbank auf RADIUS-Server
- Tabelle mit (Subject, Issuer, Serial, MAC, VLAN, Client, Switch, Port)
  - Zertifikatsdetails aus lokaler Liste (Linux), Windows-AD und Jamf (Mac)
  - MACs aus DHCP, um Weitergabe der Zertifikate zu erschweren
- RADIUS-Server prüft Zertifikat und Tabelle bei Anfrage
  - Liefert VLAN zurück
  - Switch/Port analog

FreeRADIUS

- https://www.freeradius.org/
- Erstes Release: August 1999; aktuell ist Version 3.2
- Lizenz: GPLv2

- Details: Vortrag „FreeRADIUS 3: Konzepte und Konfiguration“ https://www.dfn.de/wp-content/uploads/2024/10/BT81_Forum_Mobile-IT_freeradius.pdf

FreeRADIUS: PostgreSQL

```
sql sql_nac {
    dialect = "postgresql"
    driver = "rlm_sql_${dialect}"
    radius_db = "dbname=nocnac"

    # Sonst werden Werte wie "=" in Queries escaped "=3D"
    auto_escape = yes

    pool {
        # Standardwerte aus mods-available/sql
    }
}
```

FreeRADIUS: PostgreSQL II

```
sql_map sql_map_example {
    sql_module_instance = "sql_nac"

    query = "\
SELECT spalte1 \
FROM beispiel \
WHERE spalte2 = '%{Attribut}' \
LIMIT 1"
    update {
        # ":= 0" ist die erste Spalte im Ergebnis
        control:Tmp-String-0 := 0
    }
}
```

FreeRADIUS: NAC per MAC (SQL)

```
sql_map sql_map_nac_macbased {
    sql_module_instance = "sql_nac"
    query = "SELECT vlan FROM mappings_macbased \
  WHERE mac    = '%{Calling-Station-Id}' \
    AND client = '%{Client-Shortname}' \
    AND switch = '%{NAS-Identifier}' \
    AND port   = '%{NAS-Port-Id}' \




LIMIT 1"
    update { control:Tmp-String-0 := 0 }
}
```

FreeRADIUS: NAC per MAC (SQL)

```
sql_map sql_map_nac_macbased {
    sql_module_instance = "sql_nac"
    query = "SELECT vlan FROM mappings_macbased \
  WHERE mac    = '%{Calling-Station-Id}' \
    AND client = '%{Client-Shortname}' \
    AND switch = '%{NAS-Identifier}' \
    AND port   = '%{NAS-Port-Id}' \
UNION SELECT vlan FROM mappings_macbased \
  WHERE mac    = '%{Calling-Station-Id}' \
    AND client = '%{Client-Shortname}' \
    AND switch = '' and port = '' \
LIMIT 1"
    update { control:Tmp-String-0 := 0 }
}
```

FreeRADIUS: NAC per MAC (authorize) (1)

```
# Mit Aruba OS (ProCurve), Aruba CX und Cisco IOS getestet
if (&Service-Type == Call-Check) {
    sql_map_nac_macbased
    if (!updated) {
        update {
            &request:Module-Failure-Message += \
                "MAC-based: Not Found: \
                MAC %{Calling-Station-Id} \
                for %{Client-Shortname}"
        }
        reject
    }
```

FreeRADIUS: NAC per MAC (authorize) (2)

```
    update {
        &session-state:Tunnel-Type := VLAN
        &session-state:Tunnel-Medium-Type := IEEE-802
        # VLAN zuweisen
        &session-state:Tunnel-Private-Group-Id \
            := "%{control:Tmp-String-0}"

        # Zugriff erlauben
        &control:Auth-Type := Accept
    }
}
```

FreeRADIUS: NAC per MAC (authorize) (3)

```
post-auth {
    update {
        &reply:Tunnel-Type \
            := &session-state:Tunnel-Type
        &reply:Tunnel-Medium-Type \
            := &session-state:Tunnel-Medium-Type
        &reply:Tunnel-Private-Group-Id \
            := &session-state:Tunnel-Private-Group-Id
    }
    [...]
}
```

FreeRADIUS: NAC per 802.1X (SQL)

```
sql_map sql_map_nac_8021x {
    sql_module_instance = "sql_nac"
    query = "SELECT vlan FROM mappings_8021x \
  WHERE subject = '%{TLS-Client-Cert-Subject}' \
    AND issuer  = '%{TLS-Client-Cert-Issuer}' \
    AND serial  = '%{TLS-Client-Cert-Serial}' \
    AND mac     = '%{Calling-Station-Id}' \
    AND client  = '%{Client-Shortname}' \
    AND switch  = '%{NAS-Identifier}' \
    AND port    = '%{NAS-Port-Id}' \
UNION SELECT vlan FROM mappings_8021x [...]  # analog \
LIMIT 1"
    update { control:Tmp-String-0 := 0 }
}
```

FreeRADIUS: NAC per 802.1X (EAP)

```
eap eap_switch_nac {
    default_eap_type = tls

    tls-config tls-common {
        private_key_file = ${certdir}/nac.key
        certificate_file = ${certdir}/nac.crt
        ca_file = ${certdir}/nac-cas.crt
    }
    # EAP-TLS
    tls {
        tls = tls-common
        virtual_server = "switch-nac-inner"
    }
}
```

FreeRADIUS: NAC per 802.1X (authorize)

```
# Mit Aruba OS (ProCurve), Aruba CX und Cisco IOS getestet
    } elsif (&Service-Type == Framed-User) {
        eap_switch_nac
    }
    [...]
}

authenticate {
    Auth-Type eap_switch_nac {
        eap_switch_nac
    }
}
```

FreeRADIUS: NAC per 802.1X (authorize inner) (1)

```
eap_switch_nac {
    ok = return
}
sql_map_nac_8021x
if (!updated) {
    update {
        &request:Module-Failure-Message \
            += "802.1X: Not Found: \
            Subject %{TLS-Client-Cert-Subject}, [...] \
            for %{Client-Shortname}"
    }
    reject
}
```

FreeRADIUS: NAC per 802.1X (authorize inner) (2)

```
update {
    &outer.session-state:Tunnel-Type := VLAN
    &outer.session-state:Tunnel-Medium-Type := IEEE-802
    &outer.session-state:Tunnel-Private-Group-Id \
        := "%{control:Tmp-String-0}"

    # EAP-TLS prüft Gültigkeit des Zertifikats
    &control:Auth-Type := Accept
}
```

Switches: Anforderungen

- Unterstützung für NAC per MAC und 802.1X
- Alle VLANs auf allen Switches (bei dynamischer Zuweisung)
  - RPVST problematisch für „schwächere“ Switches $\rightarrow$ MST
  - Aruba CX 6100 (max. 32 VLANs), 6200 (max. 128), 6300 (max. 512)

Switches: Aruba „ProCurve“

```
radius-server host 192.0.2.3    key "[...]"
radius-server host 198.51.100.5 key "[...]"
aaa server-group radius "nc" host 192.0.2.3
aaa server-group radius "nc" host 198.51.100.5
aaa authentication port-access eap-radius server-group "nc"
aaa authentication mac-based pap-radius server-group "nc"
aaa port-access authenticator active

aaa port-access authenticator 1-48
aaa port-access authenticator 1-48 client-limit 1
aaa port-access mac-based 1-48
```

Switches: Aruba „ProCurve“ (Debugging)

- `show port-access clients`
- `show port-access authenticator [config/clients]`
- `show port-access mac-based [config/clients]`

- `debug destination logging`
- `debug security port-access`
- `debug security radius-server`

Switches: Aruba CX (1)

```
radius-server host 192.0.2.3 key [...]
radius-server host 198.51.100.5 key [...]
aaa group server radius nc
    server 192.0.2.3
    server 198.51.100.5
aaa authentication port-access dot1x authenticator
    radius server-group nc
    enable
aaa authentication port-access mac-auth
    radius server-group nc
    auth-method pap
    enable
```

Switches: Aruba CX (2)

```
interface 1/1/1-1/1/48
    port-access onboarding-method concurrent enable
    aaa authentication port-access dot1x authenticator
        enable
    aaa authentication port-access mac-auth
        enable
```

Switches: Aruba CX (Debugging)

- `show port-access clients`
- `show aaa authentication port-access dot1x authenticator interface all client-status`
- `show aaa authentication port-access mac-auth interface all client-status`

Switches: Cisco IOS (nur im Labor getestet) (1)

```
aaa new-model
dot1x system-auth-control
! "group radius" = konfigurierte RADIUS-Server verwenden
aaa authentication dot1x default group radius
! Für VLAN-Zuweisungen
aaa authorization network default group radius

radius-server host 192.0.2.3 \
    auth-port 1812 acct-port 1813 key SECRET
radius-server host 198.51.100.5 \
    auth-port 1812 acct-port 1813 key SECRET
```

Switches: Cisco IOS (nur im Labor getestet) (2)

```
interface Gi1/0/1-48
    switchport mode access
    authentication port-control auto
    authentication order mab dot1x
    authentication priority dot1x mab
    ! NAC per 802.1X
    dot1x pae authenticator
    ! NAC per MAC (mac authentication bypass)
    mab
```

Switches: Cisco IOS (Debugging)

- `show dot1x all`
- `show authentication sessions`

Clients

- NAC per MAC: keine Konfiguration nötig
- NAC per 802.1X EAP-TLS: Zertifikat pro Endgerät
- Separate Zugangsdaten für LAN und WLAN verwenden (!)
- Zertifikate wenn möglich automatisiert verteilen

Clients: Linux

- Zertifikate werden manuell verteilt (wenige Endgeräte)
- Verwenden separate CA
- Laufzeit CA: 100 Jahre, Zertifikat: 10 Jahre (nur Rechenzentrum)

- `wpa_supplicant -c nac.conf -i eth0 -Dwired`

```
network={
    key_mgmt=IEEE8021X
    eap=TLS
    identity="anonymous"
    client_cert="nac.crt"
    private_key="nac.key"
}
```

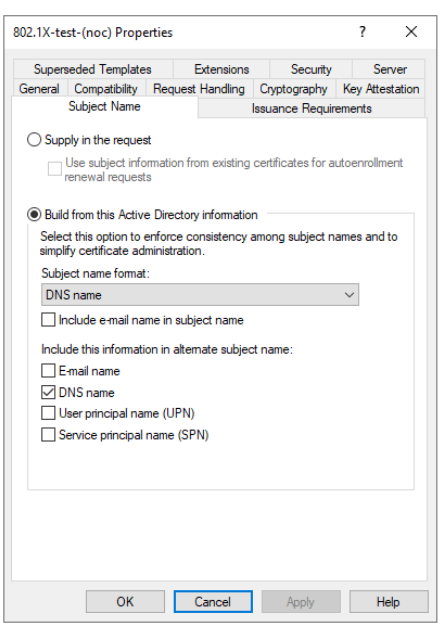
Clients: Windows

- Umsetzung durch Windows-Kollegen; Fragen an `rrze-windows@fau.de`

- Zertifikate per Active Directory Certificate Services (AD CS)
- Verteilung auf Clients per Group Policy
- MAC-Adressen und Export der Zertifikate per Microsoft Endpoint Configuration Manager (MECM), ehemals SCCM

- Hinweise
  - Änderungen an Zertifikatsvorlage erzeugt neue Zertifikate
  - Zertifikatsvorlage kann nicht aus Backup wiederhergestellt werden (!)

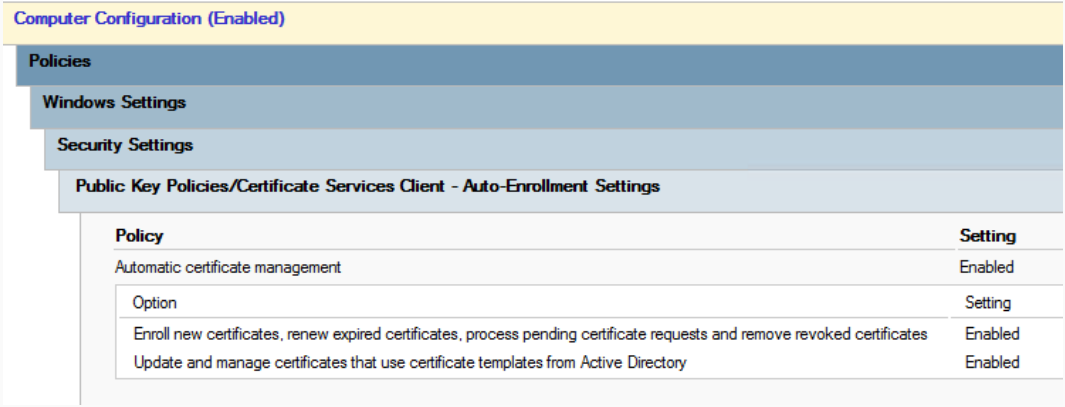
Clients: Windows (AD CS)



Clients: Windows (AD CS)

- Certificate Template in CA erstellen
  - Zertifikat für „Client/Server Authentication“
  - „General“: Name und Haken bei „Publish certificate in AD“
  - „Extensions“: Unter „Application Policies“ muss „Client Authentication“ und „Server Authentication“ vorhanden sein
  - „Security“: „Domain Computers“ oder Gruppe mit folgende Rechten: Read, Enroll, and Autoenroll
  - „Subject Name“: „Build from this Active Directory information“
- Template mit Rechtsklick unter „Certificate templates“ „veröffentlichen“

Clients: Windows (Group Policy: Zertifikat)



Clients: Windows (Group Policy: Zertifikat)

Public Key Policies/Intermediate Certification Authority Certificates

Issued To	Issued By	Expiration Date	Intended Purposes
fauad-NOC-PKI-CA	RRZE-Root-CA	09.11.2034 14:36:41	<All>

For additional information about individual settings, launch the Local Group Policy Object Editor.

Public Key Policies/Trusted People Certificates

Issued To	Issued By	Expiration Date	Intended Purposes
8021x.radius.rrze.uni-erlangen.de	8021x.radius.rrze.uni-erlangen.de	02.08.2124 18:08:48	Server Authentication, Client Authentication

For additional information about individual settings, launch the Local Group Policy Object Editor.

Public Key Policies/Trusted Publishers Certificates

Issued To	Issued By	Expiration Date	Intended Purposes
8021x.radius.rrze.uni-erlangen.de	8021x.radius.rrze.uni-erlangen.de	02.08.2124 18:08:48	Server Authentication, Client Authentication

For additional information about individual settings, launch the Local Group Policy Object Editor.

Clients: Windows (Group Policy: Netzwerk)

Windows Settings

Security Settings

System Services

Wired AutoConfig (Startup Mode: Automatic)

Permissions
No permissions specified

Auditing
No auditing specified

Wired Network (802.3) Policies

FAU-802.1x

Name	FAU-802.1x
Description	FAU-802.1x

Global Settings

Setting	Value
Use Windows wired LAN network services for clients	Enabled
Shared user credentials for network authentication	Enabled

Network Profile

Security Settings

Enable use of IEEE 802.1X authentication for network access	Enabled
Enforce use of IEEE 802.1X authentication for network access	Disabled

IEEE 802.1X Settings

Computer Authentication	Computer only
Maximum Authentication Failures	2
Maximum EAPOL-Start Messages Sent	
Held Period (seconds)	
Start Period (seconds)	
Authentication Period (seconds)	

Network Authentication Method Properties

Authentication method	Smart card or certificate
Validate server certificate	Disabled
Use a certificate on this computer	Enabled
Use simple certificate selection	Enabled
Use a different username for the connection	Disabled

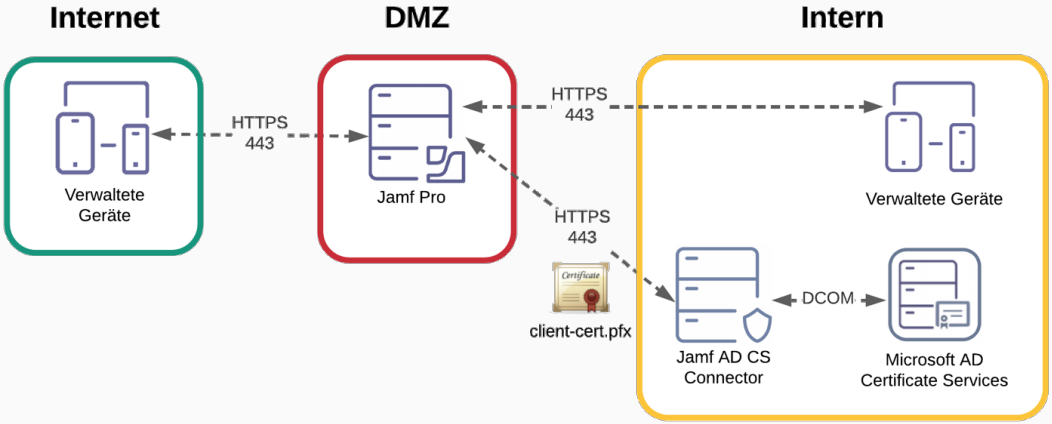
Clients: Mac

- Umsetzung durch Mac-Kollegen; Fragen an `rrze-mac@fau.de`

- Verwaltung der Geräte mit Jamf Pro
- Jamf bezieht Zertifikate über AD CS Connector von der Windows-AD
- Laufzeit: 1 Jahr, Erneuerung 30 Tage vor Ablauf
- Import der Zertifikatdetails über Windows-AD

- Hinweis: Änderungen am Konfigurationsprofil erzeugt neue Zertifikate für alle Clients

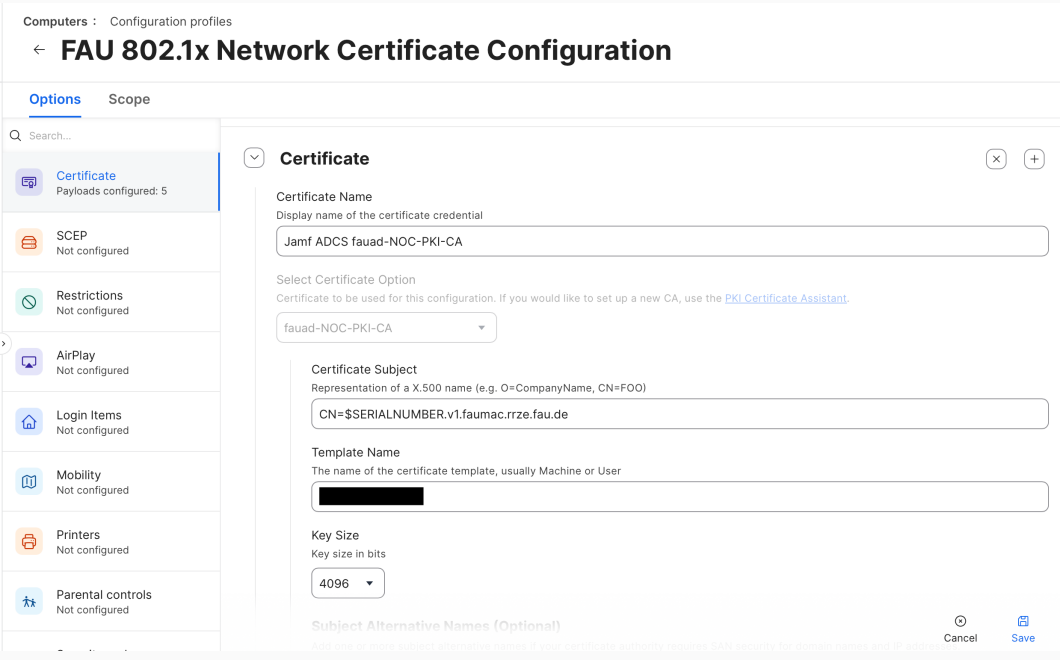
Clients: Mac (AD CS Connector)



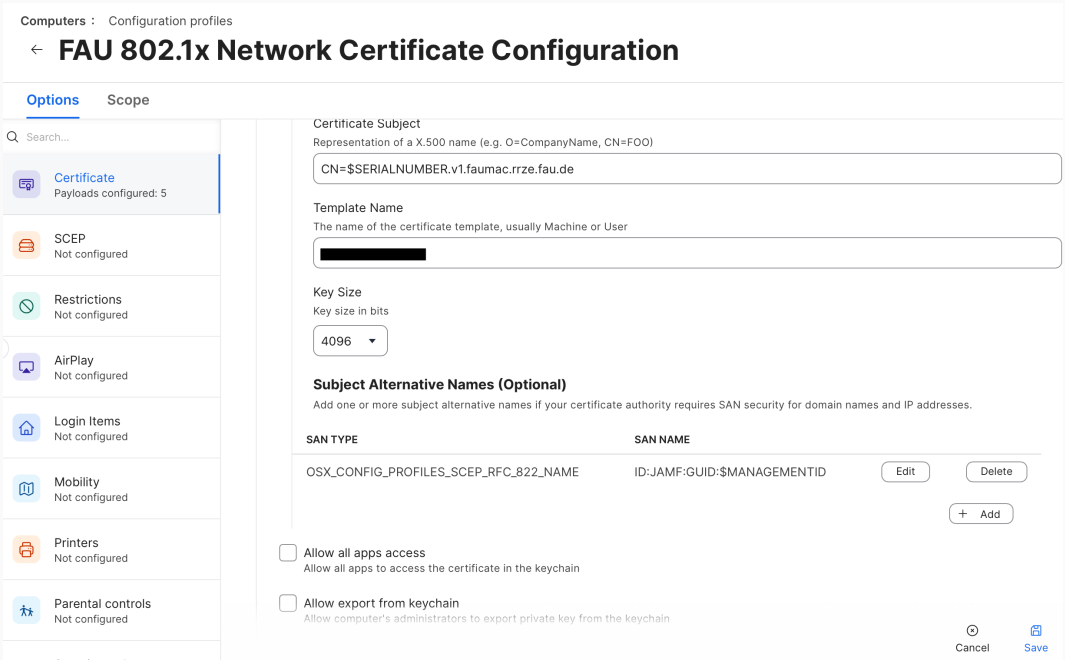
https://docs.jamf.com/de/technische-dokumente/jamf-pro/integrating-ad-cs/10.30.0/Configuring_a_Template_and_Permissions_in_ADCS.html

Wichtig: *alle* Parameter konfigurieren

Clients: Mac (Profil: Zertifikate)



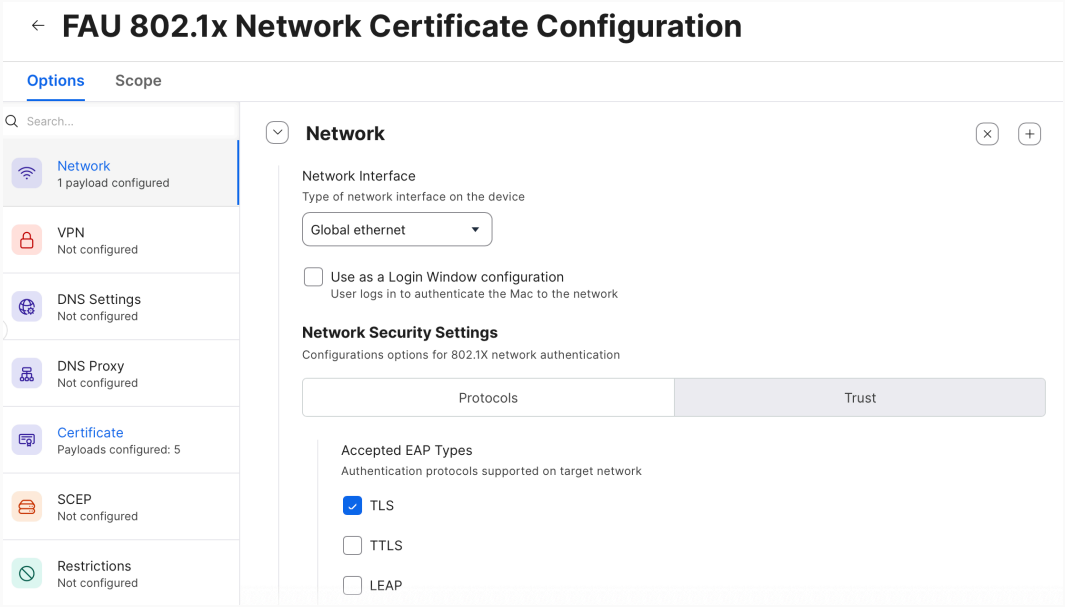
Clients: Mac (Profil: Zertifikate)



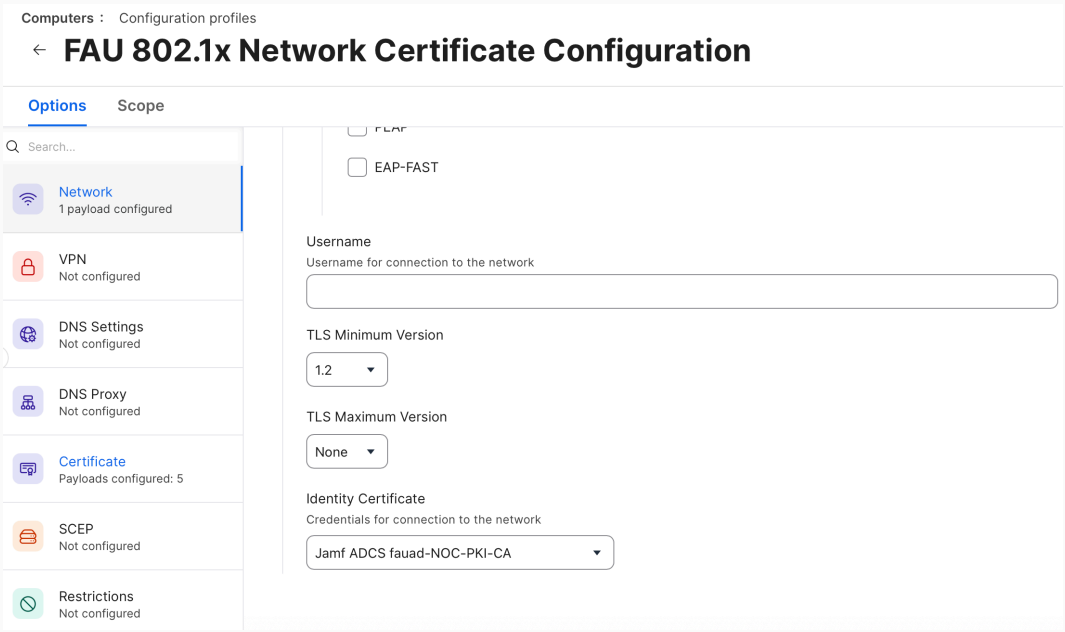
Clients: Mac (Profil: Zertifikate)

- Fünf Zertifikate werden verteilt
  - Private Root-CA und Zertifikat des RADIUS-Servers
  - Private Root-CA und Intermediate-CA der Windows-AD
  - Zertifikat des Rechners (CN=[...].v1.faumac.rrze.fau.de)
- „Template Name“
  - Windows-CA Templatename
- „Subject Alternative Names“
  - Optional, nur zur internen Erkennung

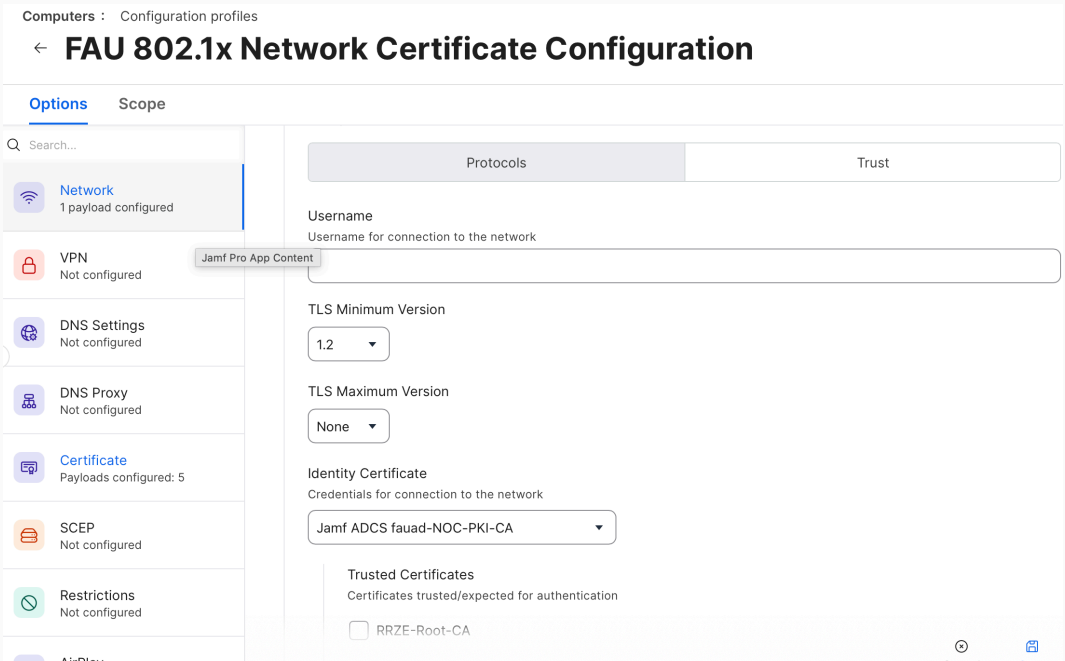
Clients: Mac (Profil: Netzwerk)



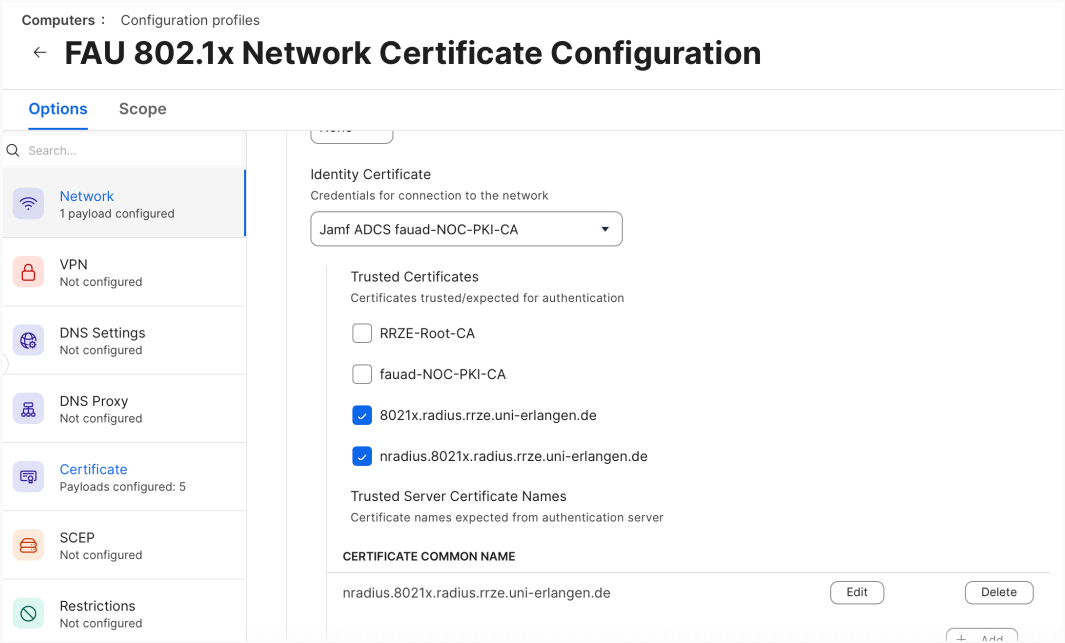
Clients: Mac (Profil: Netzwerk)



Clients: Mac (Profil: Netzwerk)



Clients: Mac (Profil: Netzwerk)



Clients: Mac (Profil: AD CS)

Settings : Global > PKI Certificates > AD CS

← **fauad-NOC-PKI-CA**

AD CS Server Integration

ⓘ Settings to integrate Jamf Pro with Active Directory Certificate Services

Display Name for Integration
User-defined name to identify this integration across Jamf Pro

fauad-NOC-PKI-CA

Required

CA Name from AD CS Server
The common name of the issuing certification authority

fauad-NOC-PKI-CA

Required

Fully Qualified Domain Name
Fully qualified domain name of the certificate authority server

.rrze.uni-erlangen.de

Required

Jamf AD CS Connector

Erfahrungen

Erfahrungen: NAC per MAC

- Plug&Play: Durchweg positive Erfahrungen
- Quasi kein Support-Aufwand: Eintrag im DHCP $\rightarrow$ funktioniert
- Nutzung
  - Aktuell 5 Standorte mit ca. 330 Switches im Produktivbetrieb
  - MACs für ca. 3000 Geräte konfiguriert
  - Alle Ports am Standort konfiguriert
  - Ausnahme: kritische Systeme wie Schließsystem/Gebäudeleittechnik

Erfahrungen: NAC per 802.1X

- Funktioniert (eigentlich) ohne Probleme
- Plug&Play wenn die Zertifikate korrekt installiert sind
- Nutzung
  - Aktuell 1 Standort im Testbetrieb
  - Zertifikate für 650 Geräte konfiguriert

Erfahrungen: NAC per 802.1X

- Funktioniert (eigentlich) ohne Probleme
- Plug&Play wenn die Zertifikate korrekt installiert sind
- Nutzung
  - Aktuell 1 Standort im Testbetrieb
  - Zertifikate für 650 Geräte konfiguriert
- Probleme mit Windows
  - Neue Zertifikate brauchten mehrere Client-Reboots
  - Falsch verteilte Zertifikate schwer ersetzbar
  - Konfiguration fehleranfällig

Erfahrungen: Probleme

- „stille“ Geräte: (3D-)Drucker
  → aktuell „manueller“ Ping auf problematische IPs (6 Geräte)
- Wake-on-LAN (WOL): Lösung noch offen

- Aruba APs flappen wenn beide Ports angeschlossen werden
  → zweiten Port ausmachen
- Multimedia-Geräte mit mehren MAC-Adressen pro Port
  → `aaa authentication port-access client-limit 4`
  (Aruba CX)

Erfahrungen: NAC und „OpenSpace“

- NAC und PVLAN (Konfiguration im Anhang)
  - Aruba CX 6200 an 1 Standort im Produktivbetrieb
  - Aruba „ProCurve“ 2930F getestet
- Funktioniert bisher ohne Probleme
  - PVLAN nur in „teureren“ Modellreihen unterstützt
  - Braucht bei Aruba MST

Ressourcen

- Ansprechpartner
  - Netzwerk: noc@fau.de
  - Mac: rrze-mac@fau.de
  - Windows: rrze-windows@fau.de

- Beispiele als Textdatei
  https://download.rrze.fau.de/noc/nac/radius.txt
  https://download.rrze.fau.de/noc/nac/procurve.txt
  https://download.rrze.fau.de/noc/nac/arubacx.txt
  https://download.rrze.fau.de/noc/nac/cisco.txt

Anhang

NAC und PVLAN: Aruba CX (Switch)

```
vlan 123
    private-vlan primary
vlan 124
    private-vlan isolated primary-vlan 123

! Aruba CX kann PVLAN nur per Rolle zuweisen
port-access role noc-openspace
    private-vlan port-type secondary
    vlan access 124

! Keine Rolle per "unauth VLAN" weil sonst lange Timeouts
```

NAC und PVLAN: Aruba CX (FreeRADIUS)

```
if (&Service-Type == Call-Check) { # NAC per MAC
    sql_map_nac_macbased
    if (!updated) {
        # Standort mit Gäste-VLAN (inkl. PVLAN)
        if (&Client-Shortname == "[...]") {
            update {
                # Rolle zuweisen
                &reply:Aruba-CPPM-Role := "noc-openspace"
                &control:Auth-Type := Accept
            }
            return
        }
        [...] # sonst wie "NAC per MAC"
```

NAC und PVLAN: Aruba „ProCurve“ (Switch)

```
! Uplink
interface 52
   no private-vlan promiscuous

vlan 123
   private-vlan primary
   private-vlan isolated 124
   tagged 52
   exit
vlan 124
   exit
```

NAC und PVLAN: Aruba „ProCurve“ (FreeRADIUS)

```
if (&Service-Type == Call-Check) { # NAC per MAC
    sql_map_nac_macbased
    if (!updated) {
        # Standort mit Gäste-VLAN (inkl. PVLAN)
        if (&Client-Shortname == "[...]") {
            update {
                &reply:Tunnel-Type := VLAN
                &reply:Tunnel-Medium-Type := IEEE-802
                &reply:Tunnel-Private-Group-Id := 124
                &control:Auth-Type := Accept
            }
            return
        }
        [...] # sonst wie "NAC per MAC"
```