AAA – Authentication, Authorization, Accounting

AAA (Authentication, Authorization, Accounting) oder **TripleA** ist ein "architectural framework", daß folgende 3 Sicherheitsfunktionalitäten in Netzwerken implementiert:

Authentication (Authentifizierung)
"Wer ist die Person?"

- Identität einer bestimmten Person mit bestimmten Methoden überprüfen
- Stellt unterschiedliche Methoden zur Verfügung um Benutzer zu identifizieren (einschließlich login und password dialog, challenge und response, messaging support und mgl. encryption).
- Schlagworte: Benutzername & PW, Biometrie, Karte und PIN, ...
- Authentication wird bei unterschiedlichen Zugriffen auf das Netzwerk bzw. auf Netzwerkdienste verwendet und muß vor der Authorization erfolgen.

Authorization (Autorisierung)
"Zu was ist die Person berechtigt?"

- Zuweisung und Überprüfung von Zugriffsrechten
- Stellt die Methode für remote access control (einschließlich one-time authorization oder authorization für jeden Dienst) zu Verfügung.
- Schlagworte: Listen (ACLs), Gruppenmitgliedschaft, Rollen, ...
- Steuerung via lokaler Informationen oder via RADIUS oder TACACS+ Server.
- Erfolgt nach der Authenticaiton.

Accounting (Abrechung)
"Was hat der User im Netz gemacht?"

- Sammeln und Senden von User-Informationen für unterschiedlichste Zwecke und Weiterverarbeitung (z.B. Abrechnungssysteme).

Vorteile von AAA gegenüber NON-AAA

- **Flexibilität und Kontrolle für Zugriffe**,
  Passwort-Konfigurationen sind nicht mehr physikalisch auf den Geräten und müssen dadurch auch nicht mehr einzeln gepflegt werden

- **Skalierbarkeit**
  Beim editieren von Usern werden sämtliche Änderungen nur einer Datenbank realisiert. Dadurch ist die Übersicht besser gewährleistet.

- **Standarisierte Authentication Methods (RADIUS, TACACS+, Kerberos)**
  Standarisierte Serverdienste bzw. Protokolle nutzbar

- **Backup-Systeme**
  Durch zusätzliche Server

Authentication

AAA Authentication läßt sich für verschiedenste Zugriffe bzw. **services** konfigurieren, wie z.B.:

- **login** Login Zugriff
- **enable** Zugriff auf den PrivilegeEXEC Mode (CLI)
- **ppp** Netzwerkauthentifizierugung via PPP
- **dot1x** Geräteauthentifizierung via dot1x (via RADIUS)

Für diese Zugriffsarten gibt es wiederum jeweils unterschiedliche **authenticaton methods**. Nachfolgend einige **Methods für den Login Zugriff**:

- **enable** enable PW verwenden
- **line** line PW verwenden
- **local** lokale Username-Daten verwenden (default Einstellung)
- **local-case** lokale Username-Daten case-sensitive verwenden
- **group radius** default Radius Gruppe: Liste aller Radius-Server
- **group tacacs+** default TACACS+ Gruppe: Liste aller TACACS+ Server
- **group** *WORD* benannte Gruppe: administrativ erstellte Server-Liste
- **none** keine Authentication

Authorization

Switches und Router verwenden AAA Authorisation um verfügbare Dienste/Funktionen für einen User bereitzustellen bzw. einzuschränken.

Der AAA Authorization process greift dabei auf Informationen (innerhalb einer Datenbank) zu, die entweder lokal auf dem Switch oder Router gehalten oder von einem remote Radius oder TACACS+ Server bereitgestellt werden.

Server stellen die Information in Form eines **AVP** (**attribute-value pair**) zur Verfügung

Der Zugriff auf einen Dienst/Funktion wird nur gestattet, wenn das Benutzerprofil entsprechnede Erlaubnis aufweist.

AAA Authorisation lässt sich für unterschiedliche **services** konfigurieren, wie z.B.

- **exec** Starten einer CLI
- **commands** *level* Ausführen von CLI Kommandos im Priv. Level (via TACACS+)
- **network** Für Netzwerkzugriffe bzw. -dienste (dyn. VLAN ID, PPP, ..)

Für diese Dienste/Funktionen (services) gibt es wiederum jeweils unterschiedliche **authorization methods**:

- **if-authenticated** nach erfolgreicher AAA Authentication
- **local** lokale Username-Daten verwenden
- **group radius** default Radius Gruppe: Liste aller Radius-Server
- **group tacacs+** default TACACS+ Gruppe: Liste aller TACACS+ Server
- **group** *WORD* benannte Gruppe: administrativ erstellte Server-Gruppe
- **none** keine Authentication

AAA Konfiguration 1

ZUERST: Aktivierung der AAA Funktion

Erst nach der Aktivierung können weitere Einstellungen vorgenommen werden

(config)# **aaa new-model**

AAA verwendet **method-lists** für die Steuerung von AAA Funktionen, über die administrativ Methoden für unterschiedliche Funktionen festgelegt werden können:

- Eine Standard-Liste mit dem Namen **default** wird bei der Aktivierung von AAA automatisch angelegt, kann jedoch auch administrativ angepasst werden
- neue Listen können mit einem frei vergebenem **list-name** können administrativ erstellt werden

In der **default method-list** wird die lokale Benutzerdatenbank verwendet für

- die **Authentifizierung** für den **Login** Zugang und
- die **Authorisierung** für CLI Zugriff (exec)

ACHTUNG: die **default method-list wird automatisch auf den VTY Lines aktiviert**.
ACHTUNG: falls kein lokaler User konfiguriert wurde, ist ein Zugang über VTY nach Aktivierung von AAA nicht mehr möglich .. nur noch über die Console - hier wird keine Änderung an der bestehenden Konfiguration vorgenommen.

Nachfolgend die entsprechenden default-Einstellungen (nicht in der running-config lesbar), die nach Aktivierung von AAA verwendet werden.

```
!
aaa new-model
!
aaa authentication login default local
aaa authorization exec default local
!
line vty 0 last-vty-nr
 login authentication default
 authorization exec default
!
```

TIPP: Nach Aktivierung von AAA (aaa new-model) Debugging anschalten und Zugriff über VTY durchführen.

debug aaa authentication
debug aaa authorization

Authentication/Authorization

Übersicht: services und methods

	service	method
authentication	login enable ppp dot1x	local, local-case line enable group {radius | tacacs+ | word} none
authorization	exec commands *level* network	if-authenticated local group {radius | tacacs+ | word} none

Durch die Angabe von **default** kann die default method-list admistrativ verändert werden. Es können jedoch auch eigene method-lists angelegt werden.

(config)# **aaa authentication** *service* {**default** | *list-name*} *method1* [*method2* [..]]
(config)# **aaa authorization** *service* {**default** | *list-name*} *method1* [*method2* [..]]

Im letzten Schritt werden die method-lists für die Dienste an eine Funktion gebunden .. z.B. an die Line. HINWEIS: Falls die default method-list verwendet wird, müssen die folgenden Kommandos nicht eingegeben werden.

(config)# **line vty 0 4**
(config-line)# **login authentication** { **default** | *list-name* }
(config-line)# **authorization** *service* { **default** | *list-name* }

Beispielkonfiguration

- Benutzerdatenbank: Benutzer "admin" mit dem Passwort "geheim"
- Authentication: Method-Lists "myLOGIN" für Login (login)
  → über lokale Benutzerdatenbank, Groß-/Kleinschreibung ist relevant
- Authorization: Method-List "myCLI" zum Ausführen der CLI (exec)
  → wird gewährt, falls die Authentifizierung erfolgreich war
- Anbindung an die VTY Lines 0 bis 4.

```
!
username admin secret geheim
!
aaa authentication login myLOGIN local-case
aaa authorization exec myCLI if-authenticated
!
line vty 0 4
  login authentication myLOGIN
  authorization exec myCLI
!
```

Optionale Kommandos

Anpassung der Prompts und Authentication Banner

(config)# **aaa authentication password-prompt** *word*
(config)# **aaa authentication username-prompt** *word*

Anpassung des Authentication Banner

(config)# **aaa authentication banner** *trennzeichen*
text,text
text,text
..
trennzeichen

Festlegung einer Nachricht, die bei einem "Login Fail" generiert wird

(config)# **aaa authentication fail-message** *trennzeichen*
text,text
text,text
..
trennzeichen

Beispielkonfiguration

```
!
aaa authentication username-prompt Name->
aaa authentication password-prompt Pass->
!
aaa authentication banner *

------------------------------------------------------------
 ACHTUNG: Zugriff nur für authorisierte Personen!
Zuwiederhandlungen werden strafrechtlich verfolgt!
------------------------------------------------------------

*
!
aaa authentication fail-message *

------------------------------------------------------------
 ACHTUNG: Zugriff nur für authorisierte Personen!
 - IHRE AKTIVITAETEN WERDEN PROTOKOLLIERT -
Zuwiederhandlungen werden strafrechtlich verfolgt!
------------------------------------------------------------

*
!
```

Accounting (optional)

AAA Accounting sammelt Informationen bei Services unterschiedlicher Art:

- **auth-proxy** For authentication proxy events.
- **commands** For exec (shell) commands.
- **connection** For outbound connections. (telnet, rlogin)
- **exec** For starting an exec (shell).
- **network** For network services. (PPP, SLIP, ARAP)
- **resource** For resource events.
- **system** For system events.

.. und tut das, bei einem bestimmten Aufttreten der Aktivität:

- **start-stop** Record start and stop without waiting
- **stop-only** Record stop when service terminates.

.. mit Hilfe unterschiedlichen Methoden (remote Server only)

- **group radius** Liste aller Radius-Server verwenden
- **group tacacs+** Liste aller TACACS+ Server verwenden

Konfigurationskommando:

(config)# **aaa accounting** *service* { **start-stop** | **stop-only** }
{ **default** | *list-name* } *method1* [*method2* ..]

RADIUS/TACACS+ Überblick

	RADIUS Remote Authentication Dial-In User Service	**TACACS+** Terminal Access Controller Access Control System
Protokoll	offener Standard, RFCs 2865-2869	Cisco proprietär
Multiprotocol Support	Unterstützt nur IP: IPv4, IPv6	Unterstützt: IPv4, IPv6, IPX, Apple Talk, NetBIOS
AAA Support	Kombiniert: → Authentication und Authorisation	Separiert: → Authentication und Authorization
Verschlüsselung	gesamte Kommunikation	nur Passwort in einer Anfrage
Kommunikation	UDP → Port 1812 (neu), 1645: Auth → Port 1813 (neu), 1646: Acct	TCP → Port 49
	differenzierte Anfragen und Antworten: Anfragen werden einzeln gestellt und beantwortet	einzelne Anfrage und Antwort: alle Anfragen (Authentication und Authorization) werden gesammelt gestellt und beantwortet
Funktionen	unterstützt im Gegensatz zu TACACS+: 802.1x Authentication mit dynamischer VLAN ID Vergabe	unterstützt im Gegensatz zu RADIUS: Anpassung der CMD Level

Anfragen werden in Form sogenannter **AV-Pairs** (Attribute-Value Pair) gestellt. Nachfolgend eine Auswahl der wichtigsten, von Cisco Geräten unterstützten **Attribute zur Authorization**:

Attribute	Bedeutung
TACACS+ # **show aaa attributes**	
Priv-lvl	authorization exec → Zuweisung eines Privilege Levels
CMD	authorization exec → Erlauben/Verbieten von Kommandos
Autocmd	authorization exec → automatischer Start eines Kommandos
ACL	authorization exec → automatische Anbindung einer line ACL
Route	authorization network (SLIP, PPP) → dyn. Konfiguration einer Route
ADDR, Addr-Pool	authorization network (SLIP, PPP) → dyn. Konfiguration von IPs
RADIUS # **show radius table attributes**	
cisco-avpair	"Priv-lvl": authorization exec → Zuweisung eines Privilege Levels
Service-Type	"NAS-Prompt-User": authorization exec "Framed-User": authorization network (SLIP, PPP)
Framed-Protocol	authorization network (SLIP, PPP) → SLIP oder PPP
Framed-IP-Address	authorization network (SLIP, PPP) → dyn. Konfiguration von IPs
Tunnel-Type Tunnel-Medium-Type Tunnel-Private-Group-ID	authorization network (dot1x) → dyn. Zuweisung einer VLAN ID

Kommunikation

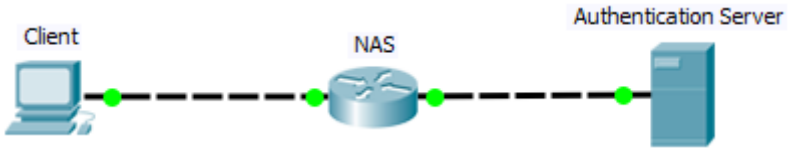
RADIUS Message Types

Access-Request	Beinhaltet Username/Passwort und weitere Infos in Form von AV-Pairs
Access-Challenge	Wird für Challenge-base Authentication verwendet: → CHAP, MS-CHAP, EAP-MD5
Access-Accept	Authentication/Authorisation ok
Access-Reject	Authentication/Authorization failure

TACACS+ Response Types

ACCEPT	Authentication oder Authorization Request ok
REJECT	Authentication oder Authorization Request failure
ERROR	Fehlerhafte Kommunikation (Misconfiguration, Connection problem)
CONTINUE	Weitere Informationen notwendig

Kommunikation



	RADIUS Kommunikation **# debug radius**	
	← Username Prompt	
Username →		
	← Password Prompt	
Password →		
	Access-Request (incl. AV-Pairs zur Authorization) →	
		← Access-Accept/Reject
	TACACS+ Kommunikation **# debug tacacs**	
connect →	username prompt ? →	
	← username prompt	← username prompt
username →	username →	
		← ACCEPT/REJECT
	password prompt ? →	
	← password prompt	← password prompt
password →	password →	
		← ACCEPT/REJECT
	network authorization →	
		← ACCEPT/REJECT
Kommando Eingabe →	command authorization →	
		← ACCEPT/REJECT

AAA Konfiguration 2 - Zugriff auf Server

RADIUS

alte Ports: auth 1645, acct 1646 (default)
neue Ports: auth 1812, acct 1813 → Portangaben sind in allen Kommandos optional

altes Kommando – Hinweismeldung IOS 15:
→ "Warning: This CLI will be deprecated soon. Please move to radius server <name> CLI."

(config)# **radius-server host** *IP* **auth-port** *port* **acct-port** *port* **key** *passwort*

```
!
radius-server host 192.168.1.100 auth-port 1812 acct-port 1813 key geheim
!
```

neues Kommando
→ kann IPv6

(config)# **radius server** *radius-srv-name*
(config-radius-server)# **address** {**ipv4** | **ipv6**} *IP* **auth-port** *port* **acct-port** *port*
(config-radius-server)# **key** *passwort*

```
!
radius server RADSRV
 address ipv4 192.168.1.100 auth-port 1812 acct-port 1813
 key geheim
!
```

Alternativ - Anlegen einer Server-Gruppe
→ kann multiple Server beinhalten

- server → IPv4 only, der Server kann auch in anderen Gruppen verwendet werden
- server-private → IPv4/IPv6, nur in dieser einen Gruppe verwendbar

→ es kann eine Absende IP (SRC IP) für die RADIUS Kommunikations festgelegt werden
→ der Server kann über VRF erreicht werden
ACHTUNG: als Name nicht radius, RAD, o.ä. verwenden, da die default-Radius Gruppe bereits unter dieser Bezeichnung angesprochen wird.

(config)# **aaa group server radius** *serv-gr-name*
(config-sg-radius)# **server-private** *ip* **auth-port** *port* **acct-port** *port* **key** *pass*
(config-sg-radius)# **server name** *radius-srv-name*
(config-sg-radius)# [**ip radius source-interface** *IF-ID*]
(config-sg-radius)# [[**ip**] **vrf forwarding** *vrf-name*]

```
!
aaa group server radius MYRAD
  server-private 192.168.1.100 auth-port 1812 acct-port 1813 key geheim
!
HINWEIS: es kann ebenfalls ein zuvor konfigurierter Server referenziert werden.
!
aaa group server radius MYRAD
  server name RADSRV
!
```

Login via RADIUS – neue Variante

Beispielkonfiguration .. alle AAA relevanten Kommandos

Aktivierung von AAA

```
!
aaa new-model
!
```

Server Zugriff – im Beispiel wird nach neuer Konfiguration eine Server mit dem Namen RADSRV angelegt, der in der RADIUS Server Gruppe MYRAD referenziert wird.

```
!
radius server RADSRV
  address ipv4 192.168.1.100 auth-port 1812 acct-port 1813
  key geheim
!
aaa group server radius MYRAD
  server name RADSRV
  ip radius source-interface loopback 0
!
```

AAA Method-Lists: für login Authentication (RADLOG) und exec Authorization (RADCLI) wird jeweils die RADIUS Server Gruppe MYRAD als erste (!) Methode verwendet.
Die Methoden local-case bzw. if-authenticated werden nur verwendet, falls der RADIUS Server nicht erreichbar ist (.. ein Hintertürchen – ein User sollte angelegt sein).

```
!
aaa authentication login RADLOGIN group MYRAD local-case
aaa authorization exec RADCLI group MYRAD if-authenticated
!
```

Verwendung der AAA Method-Lists an der Line vty.

HINWEIS:auf der line con wird die default method-list verwendet:
login und authentication via local user database (.. ein User sollte angelegt sein)

```
!
line vty 0 4
  login authentication RADLOGIN
  authorization exec RADCLI
!
```

Login via RADIUS – alte Variante

Beispielkonfiguration .. alle AAA relevanten Kommandos

Aktivierung von AAA

```
!
aaa new-model
!
```

Server Zugriff.

```
!
radius-server host 192.168.1.100 auth-port 1812 acct-port 1813 key geheim
ip radius source-interface loopback 0
!
```

AAA Method-Lists: default
.. mit Hintertürchen – lokaler Benutzer sollte angelegt sein.

```
!
aaa authentication login default group radius local-case
aaa authorization exec default group radius if-authenticated
!
```

Verwendung der AAA Method-Lists an der Line vty.
→ **unnötig**, da Verwendung der default method-lists bereits default-Einstellung ist.
ACHTUNG auf auf der line con 0.

```
!
line vty 0 4
  login authentication default
  authorization exec default
!
```

TACACS+

altes Kommando – Hinweismeldung IOS 15:
"This cli will be deprecated soon. Use new server cli"

(config)# **tacacs-server host** *IP* **key** *passwort*

```
!
tacacs-server host 192.168.1.100 key geheim
!
```

neues Kommando
→ kann IPv6

(config)# **tacacs server** *tacacs-srv-name*
(config-tacacs-server)# **address** {**ipv4** | **ipv6**} *IP*
(config-tacacs-server)# **key** *passwort*

```
!
tacacs server TACSRV
 address ipv4 192.168.1.100
 key geheim
!
```

Alternativ - Anlegen einer Server-Gruppe
→ kann multiple Server beinhalten

- server → IPv4 only, der Server kann auch in anderen Gruppen verwendet werden
- server-private → IPv4/IPv6, nur in dieser einen Gruppe verwendbar

→ es kann eine Absende IP (SRC IP) für die TACACS Kommunikation festgelegt werden
→ der Server kann über VRF erreicht werden
ACHTUNG: als Name nicht tacacs+, tacacs, TAC, o.ä. verwenden, da die default-TACACS+ Gruppe bereits unter dieser Bezeichnung angesprochen wird.

(config)# **aaa group server tacacs+** *serv-gr-name*
(config-sg-tacacs)# **server-private** *ip* **key** *pass*
(config-sg-tacacs)# **server name** *tacacs-srv-name*
(config-sg-tacacs)# [**ip tacacs source-interface** *IF-ID*]
(config-sg-tacacs)# [[**ip**] **vrf forwarding** *vrf-name*]

```
!
aaa group server tacacs+ MYTAC
  server-private 192.168.1.100 key geheim
!
```

HINWEIS: es kann ebenfalls ein zuvor konfigurierter Server referenziert werden.

```
!
aaa group server tacacs+ MYTAC
  server name TACSRV
!
```

ACHTUNG (Bug) – bei der Konfiguration einer eigenen tacacs+ Server-Gruppe:
.. fehlerhaftes Verhalten bei der CMD Authorization
–> Ausgabe von # **debug tacacs**: "no address for get_server"
TODO: Konfiguraton löschen – und neu konfigurieren, dabei den Server auf die alte oder neue Methode anlegen (keine eigene Server-Gruppe)

Login via TACACS+

Beispielkonfiguration 1 .. alle AAA relevanten Kommandos

Aktivierung von AAA

```
!
aaa new-model
!
```

Server Zugriff – im Beispiel wird nach neuer Konfiguration eine Server mit dem Namen TACSRV angelegt, der in der TACACS Server Gruppe MYRAD referenziert wird.

```
!
tacacs server TACSRV
  address ipv4 192.168.1.100
  key geheim
!
aaa group server tacacs MYTAC
  server name TACSRV
  ip tacacs source-interface loopback 0
!
```

AAA Method-Lists: es wird jeweils die TACACS Server Gruppe MYTAC als erste (!) Methode verwendet - die Methoden local-case bzw. if-authenticated werden nur verwendet, falls der RADIUS Server nicht erreichbar ist (.. ein Hintertürchen – ein lokaler User sollte präsent sein).

HINWEIS: Cisco kennt default zwei Privilege Level (TIPP: **# show privilege**):

- Level 1 → UserEXEC Mode
- Level 15 → PrivilegeEXEC Mode
  Wenn Kommandos aus Level 15 über die TACACS Server Konfiguration für einen User (oder eine Gruppe) verboten werden sollen, dürfen diese nicht über die "default" Liste erlaubt sein .. denn das hat Priorität – daher ist auch eine Method-List für commands 15 anzulegen die auf die TACACS Gruppe zeigt (group tacacs+ oder group tacacs-grp-name).

```
!
aaa authentication login TACLOGIN group MYTAC local-case
aaa authorization exec TACEXEC group MYTAC if-authenticated
aaa authorization commands 15 TACCMD group MYTAC if-authenticated
!
```

Verwendung der AAA Method-Lists an der Line vty.

HINWEIS:auf der line con wird die default method-list verwendet:
login und authentication via local user database (.. ein User sollte angelegt sein)

```
!
line vty 0 4
  login authentication TACLOGIN
  authorization exec TACEXEC
  authorization commands 15 TACCMD
!
```

Login via TACACS+ - alte Methode

Beispielkonfiguration 1 .. alle AAA relevanten Kommandos

Aktivierung von AAA

```
!
aaa new-model
!
```

Server Zugriff.

```
!
tacacs-server host 192.168.1.100 key geheim
ip tacacs source-interface loopback 0
!
```

AAA Method-Lists: default
.. mit Hintertürchen – mindestens ein lokaler Benutzer sollte existieren.

```
!
aaa authentication login default group tacacs+ local-case
aaa authorization exec default group tacacs+ if-authenticated
aaa authorization commands 15 default group tacacs+ if-authenticated
!
```

Verwendung der AAA Method-Lists an der Line vty.
→ **unnötig**, da Verwendung der default method-lists bereits default-Einstellung ist.
ACHTUNG auf auf der line con 0.

```
!
line vty 0 4
  login authentication default
  authorization exec default
  authorization commands 15 default
!
```

802.1x User Authentication (Radius)

IEEE 802.1x ist eine offener Standard zur Layer 2 Zugangskontrolle (auch häufig in WLANs anzutreffen) und auf allen Cisco Switches verfügbar.

Bei einer dot1x Authentifizierung muss sich der User - i.d.R. interaktiv – durch Eingabe von Username und Passwort zur Teilname am Netzwerk authentifizieren.

Eine Zugangskontrolle via dot1x erfordert einen RADIUS Authentication Server (nicht TACACS), einen dot1x fähigen Client und einen Switch: das Gerät, das den Zugang zum Netzwerk realisiert und die Authentifizierung zwischen Client und Server vermittelt.

Übersicht Komponenten einer 802.1x Topologie.

- **Supplicant (Client)** – die Workstation, die eine Authentication via 802.1x anfordert.

- **Authenticator (Switch)** – der Proxy zwischen Client und Authentication Server. Leitet Requests vom Client an den Server weiter und ermöglicht (nach erfolgreicher Authentication) Zugang in das Netzwerk.

- **Authentication Server (RADIUS Server)** – validiert die Anfragen der Clients. 802.1x ist nur mit Radius möglich (.. und nicht mit TACACS oder TACACS+)

Die Kommunikation erfolt über **EAPOL** - Extensible Authentication Protocol over LAN (OSI 2).

Ablauf der Kommunikation
Anfänglich ist der Switchport ist im **unauthorized state**.
In diesem Status ist ausschließlich folgender Datenverkehr erlaubt: EAPOL, CDP, STP

1. **Supplicant** (Client)
   sendet (i.d.R. nach Aufforderung des Switches) eine EAPOL Frame mit Authentication Informationen (Username/Pass) an den Switch

2. **Authenticator** (Switch)
   leitet den EAPOL Datenverkehr vom Client an den Server weiter – führt also eine "proxy" Funktion aus. Der Client kommuniziert über den Autenticator mit dem Authentication Server.

3. **Authentication Server** (RADIUS)
   sendet nach Überprüfung entweder ein access-accept (und mgl. weitere Informationen) oder einer accept-reject.

4. **Authenticator** (Switch)
   falls die Authentication erfolgreich war, schaltet der Switch den Port zuletzt in den sogenannten **authorized state** und – in Abhängigkeit erhaltener Informationen vom Server - konfiguriert mgl. den Port dynamisch mit einer erhaltenen VLAN ID - ansonsten verbleibt der Port im **unauthorized** state.

   .. und sendet ACCEPT oder REJECT via EAPOL an den **Supplicant** weiter.

AAA Konfiguration 3 - dot1x

Authenticator

Aktivierung von AAA ..

(config)# **aaa new-model**

.. und Angabe des RADIUS Servers sind Voraussetzung für die dot1x Authentication:

(config)# **radius-server host** *IP* [**auth-port** *port*] [**acct-port** *port*] **key** *password*

.. ALTERNATIV kann natürlich auch
- die neue Methode zur Angabe des Servers verwendet werden (falls unterstützt)
  (config)# **radius server** *srv-name*
- oder es kann eine eigene radius server Gruppe erzeugt werden
  (config)# **aaa group server radius** *srv-group-name*

dot1x – globale Einstellungen

Die 802.1x Funktion muss global aktiviert werden.

(config)# **dot1x system-auth-control**

Die dot1x Authentifizierung muss den RADIUS Server als Methode festlegen.
ACHTUNG: Falls über den RADIUS Server auch die VLAN ID für den Switchport bezogen wird, muss die network Authorisierung ebenfalls über den RADIUS Server realisiert werden.

(config)# **aaa authentication dot1x** {**default** | *list-name*} **group** {**radius** | *srvgrp*}
(config)# **aaa authorization network** {**default** | *list-name*} **group** {**radius** | *srvgrp*}

dot1x – am Switchport

Dann wird 802.1x an dem entsprechenden **Switchport** aktiviert.
ACHTUNG: Der Port muss ein access link sein, PortFast muss aktiv sein.

(config-if)# **switchport host**

Aktivierung der dot1x Portkontrolle:
ACHTUNG: unterschiedliches Kommando je nach Hardware-Platform und IOS (siehe prompt).
Für die korrekte Funktion der dot1x Portkontrolle muss immer der Parameter **auto** verwendet erden (default: force-unauthorized – nie authorisiert; alternativ: force-authorized – immer authorisiert).

2950,3550(config-if)# **dot1x port-control auto**
.. oder:
3560,3750(config-if)# **authentication port-control auto**

ACHTUNG: bei Multilayer Switches ab 3560 muss zusätzlich das folgenden Kommando verwendet werden, dass den PAE Type (Port Access Entity) des Ports – gemäß der Rolle des Switches im Authentifikationsprozess – als supplicant definiert.

3560,3750(config-if)# **dot1x pae authenticator**

dot1x - optional

Multiple Clients

Falls mehere Clients mit 801.1x Authentication an einem Switchport angeschlossen werden (z.B. ein PC und ein IP Phone) sollte das folgende Kommando am emsprechenden Switchport verwendet werden.

2950,3550(config-if)# **dot1x host-mode multi-host**
.. oder:
3560,3750(config-if)# **authentication host-mode** {**multi-domain** | **multi-auth**}

MAC-Address Bypass

3550(config-if)# **dot1x mac-auth-bypass**
.. oder:
3560,3750(config-if)# **mab**
3560,3750(config-if)# **authentication order dot1x mab**

Periodische Authentifizierung
Authentifizierung wird regelmäßig durchgeführt – default: alle 3600 Sekunden (1h).

3560,3750(config-if)# **authentication periodic** *sec*

Fehlerhafte Authentication
VLAN Zugehörigkeit definieren, wenn die Authentifikation fehlschlägt, der Server nicht erreichbar ist oder der Client nicht dot1x-fähig ist.

2950,3550(config-if)# **dot1x auth-fail vlan** *vlan-id*
.. oder
3560,3750(config-if)# **authentication event fail action**
authorize vlan *vlan-id*
3560,3750(config-if)# **authentication event no-response action**
authorize vlan *vlan-id*

Beispielkonfiguration dot1x – minimal Konfiguration

NOTWENDIGSTE minimal Konfiguration

Globale Einstellungen – auf allen Geräten identisch ..
.. Abweichungen bei der Angabe des RADIUS Servers – je nach IOS - möglich.

```
!
aaa new-model
!
radius server RADSRV
  address ipv4 192.168.1.100 auth-port 1812 acct-port 1813
  key geheim
!
dot1x system-auth-control
!
aaa authentication dot1x default group radius
aaa authorization network default group radius
!
```

Port-Konfiguration → **alt** (IOS < 15, L2 Switches)

```
!
interface IF-Typ IF-Nr
  switchport host

  dot1x port-control auto
!
```

Port-Konfiguration → **neu** (IOS 15.x, L3 Switches)

```
!
interface IF-Typ IF-Nr
  switchport host

  dot1x port-control auto
  dot1x pae authenticator
!
```

Beispielkonfiguration dot1x – für unterschiedliche Hardware Platforms

Globale Einstellungen – auf allen Geräten identisch
.. Abweichungen bei der Angabe des RADIUS Servers möglich.

```
!
aaa new-model
!
aaa group server radius MYRAD
 server-private 192.168.1.254 auth-port 1812 acct-port 1813 key geheim
!
dot1x system-auth-control
!
aaa authentication dot1x default group MYRAD
aaa authorization network default group MYRAD
!
```

Port-Konfiguration → **2950** (Aktivierung, multiple Hosts) .. notwendige Kommandos: fett

```
!
interface IF-Typ IF-Nr
 switchport host

 dot1x port-control auto

 switchport voice vlan vlan-ID
 dot1x host-mode multi-host
!
```

Port-Konfiguration → **2960**, **3550** (Aktivierung, multiple Hosts, MAC Bypass)

```
!
interface IF-Typ IF-Nr
 switchport host

 dot1x port-control auto
 dot1x pae authenticator

 switchport voice vlan vlan-ID
 dot1x host-mode multi-host

 dot1x mac-auth-bypass
!
```

Port-Konfiguration → **3650, 3750** (Aktivierung, multiple Hosts, MAC Bypass)

```
!
interface IF-Typ IF-Nr
 switchport host

 authentication port-control auto
 dot1x pae authenticator

 switchport voice vlan vlan-ID
 authentication host-mode { multi-domain | multi-auth }

 mab
 authentication order dot1x mab
!
```

Supplicant (PC Win7, Win10) – in Worten

dot1x Authentication für Windows 7 oder Windows 10 Client einstellen

1. Start
   → "Ausführen": **services.msc**

2. Dienst starten: Register Standard (unten)
   → **Automatische Konfiguration (verkabelt)** .. Starten mit Startknopf
   HINWEIS: der Dienst kann zukünftig auch immer automatisch gestartet werden

3. Netzwerkverbindung konfigurieren (jetzt gibt es einen Reiter Authentifizierung)
   Start
   → Systemsteuerung
   → Netzwerk und Internet
   → Netzwerk- und Freigabecenter
   → **Adaptereinstellungen** ändern

4. Mit rechter Maustaste auf den Adapter
   → Eigenschaften
   → Reiter **Authentifizierung**
   - IEEE 802.1X-Authentifizierung aktivieren (aktiviert)

   - "Microsoft Geschütztes EAP (PEAP)"
     → Einstellungen
     - Zertifikatsprüfung deaktivieren
     - Gesichertes Kennwort "EAP-MSCHAP v2"
       → Konfigurieren (Haken raus bei Frage und o.k.)

   - Für diese Verbindung … eigene Anmeldeinformationen ... speichern (aktiviert)

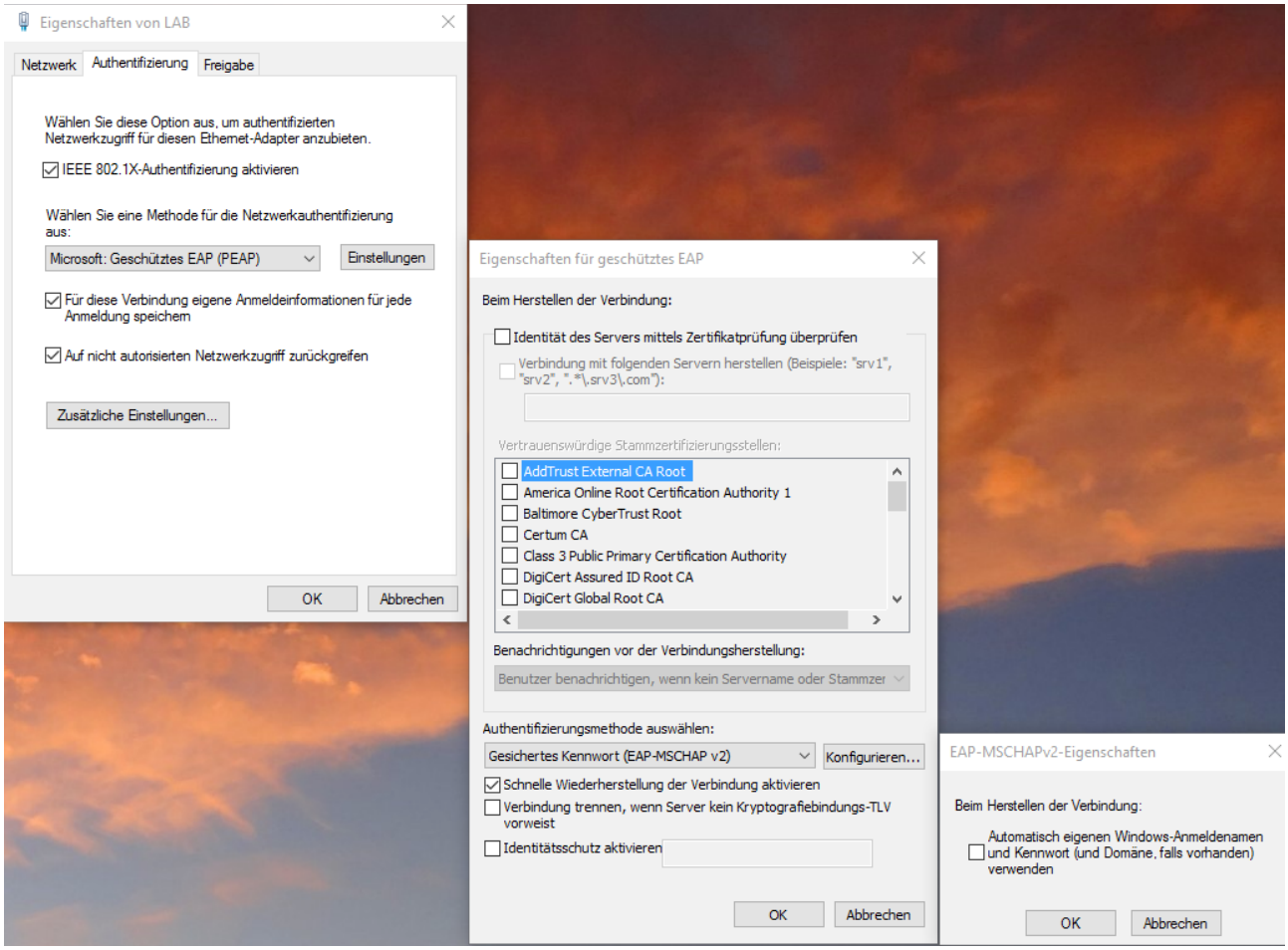
   - Auf nicht autorisierten Netzwerkzugriff zurückgreifen (aktiviert)

   - Zusätzliche Einstellungen
     - Authentifizierungsmodus angeben: "Benutzerauthentifizierung"
     - Einmaliges Anmelden für dieses Netzwerk aktivieren (aktiv)
     - "Unmittelbar vor .." (aktiv)

Supplicant (PC Win7, Win10) – in Bildern

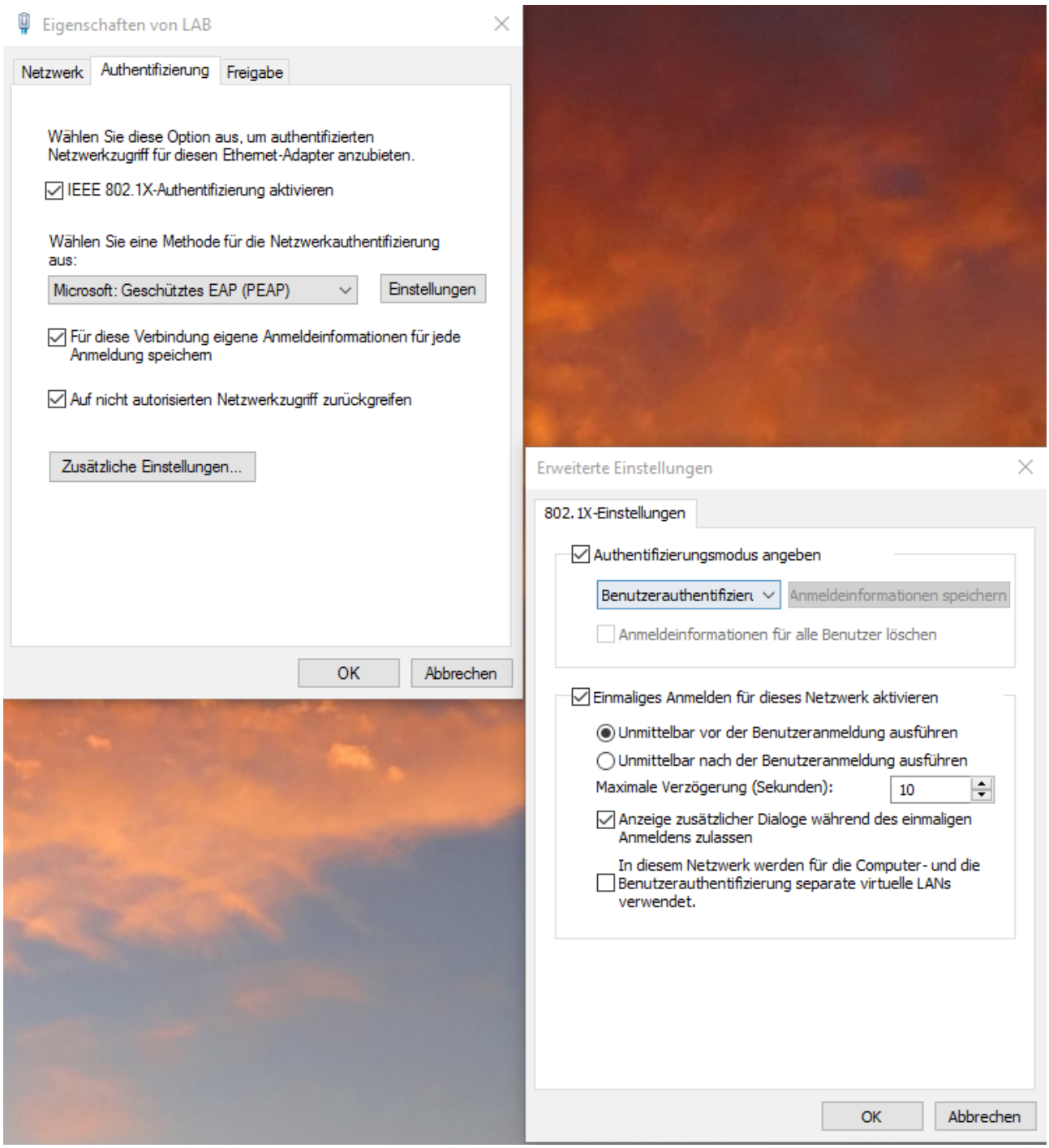
dot1x Authentication für Windows 7 oder Windows 10 Client einstellen

EINSTELLUNGEN 1
→ nach Starten des Dienstes **Automatische Konfiguration (verkabelt)** über
→ Ausführen "services.msc"



.. es folgen EINSTELLUNGEN 2 auf nächster Seite

EINSTELLUNGEN 2



AAA Troubleshooting

AAA Troubleshooting (mit oder ohne Server)

show aaa sessions

debug aaa authentication
debug aaa authorization
debug aaa accounting

Troubleshooting Server TACACS+/RADIUS

show tacacs
show radius server-group all

show aaa attributes
show radius table attributes

debug tacacs
debug radius

test aaa group {**radius** | **tacacs+** | *radius-group-name*}
username pass **new-code**

→ authenticate (sofort): keine Kommunikation mit Server (fehlerhafte Port Konfig.)
→ rejected (sofort): fehlerhafte User/Pass Übermittlung
→ rejected (nach einer Pause): keine Antwort vom Server

Troubleshooting dot1x

show authentication sessions method {**mab** | **dot1x** }

show authentication sessions [*IF-ID*]
show authentication interface *IF-ID*

show dot1x [**all**]

Mitteilungen von Servern

TACACS+: # **debug aaa authentication**

- "*Connection closed by foreign Host*"
  → TACACS+ Server ist offline.
- "*Invalid AUTHEN/START packet (check keys)*"
  → AAA Key auf TACACS+ Server und Client passen nicht
- "*Authentication failure*"
  → Falscher Username/Password vom AAA Client

Radius: # **debug ip radius**

- "*No response from server*"
  → Radius ist offline.
- "*Reply for id fails decrypt*"
  → AAA Key auf TACACS+ Server und Client passen nicht
- "*No appropriate authorization type for user*"
  → User will Dienst/Funktion verwenden, für die er nicht autorisiert ist.
- "*Received from id id IP-address:port-number, Access-Reject*"
  → Falscher Username/Password vom AAA Client

AAA Konfigurationsbeispiel – Login und dot1x

Hardware-Platform 3750, IOS 12.2

Login via TACACS

- mit Hintertürchen (local-case für Authentication bzw. if-authenticated für Authorization)
- inclusive CMD Authorization (authorization commands 15)
- Verwendung der default tacacs Gruppe, daher keine Konfiguration auf den lines notwendig. Ebenfalls Schutz gegen TACACS CMD Auth Bug.

dot1x via RADIUS

- inclusive dynamischer VLAN Konfiguration auf den dot1x enabled Ports

Einfachste "Minimal"-Konfiguration.

```
!
aaa new-model
!
radius-server host 192.168.1.100 auth-port 1812 acct-port 1813 key cisco
tacacs-server host 192.168.1.100 key cisco
!
aaa authentication login default group tacacs local-case
aaa authorization exec default group tacacs if-authenticated
aaa authorization commands 15 default group tacacs if-authenticated
!
dot1x system-auth-control
!
aaa authentication dot1x default group radius
aaa authorization network default group radius
!
interface gi 1/0/10
  switchport mode access
  spanning-tree portfast
  authentication port-control auto
  dot1x pae authenticator
!
```

AAA Konfigurationsbeispiel – PPP/CHAP Autentication

.. via TACACS+ Server – mit "lokalem" Hintertürchen

Router "A"

```
!
hostname A
!
username B password cisco
!
aaa new-model
aaa authentication ppp AUTHPPP group tacacs+ local
!
interface Serial0/1
 bandwidth 4000
 ip address 10.1.1.1 255.255.255.252
 encapsulation ppp
 ppp authentication chap authentication AUTHPPP
!
tacacs-server host 192.168.11.252 key cisco
!
```

Router "B"

```
!
hostname B
!
username A password cisco
!
aaa new-model
aaa authentication ppp PPP1 group tacacs+ local
!
interface Serial0/1
 bandwidth 4000
 ip address 10.1.1.2 255.255.255.252
 encapsulation ppp
 ppp authentication chap authentication PPP1
!
tacacs-server host 192.168.11.252 key cisco
!
```

Anhang A – RADIUS Server: freeradius 2.1.12 (Linux)

(Haupt)Quellen:

- http://www.administrator.de/wissen/netzwerk-zugangskontrolle-802-1x-freeradius-lan-switch-154402.html
- http://www.gnu.org/software/radius/manual/html_node/radius.html#Top
- http://wiki.freeradius.org/Home

FreeRADIUS INSTALLIEREN	
# **apt-get install freeradius**	
RADIUS DATEIEN ANPASSEN	
/etc/freeradius/**radiusd.conf** → Ports einstellen: "listen" port und "acct" port → NEUE Ports: 1812,1813 (alt: 1645,1646)	**port = 1812** **port = 1813** .. siehe auch folgende Seiten
/etc/freeradius/**eap.conf** → für dynamische VLAN Zuweisung → in den Sections: TTLS und PEAP .. von no auf yes	**use_tunneled_reply = yes** **use_tunneled_reply = yes**
/etc/freeradius/**modules/mschap** → WindowsKompatibilität: von no auf yes	**with_ntdomain_hack = yes**
/etc/freeradius/**proxy.conf** → WindowsKompatibilität: Kommentar weg.	**realm LOCAL {** **}**
/etc/freeradius/**sites-enabled/default** /etc/freeradius/**sites-enabled/inner-tunnel** → WindowsKompatibilität: in beiden Dateien Kommentierung ändern (muss aussehen wie rechts) .. default (an 2 Stellen): ca Zeile 120 und Zeile 310, .. inner-tunnel (nur einmal): ca Zeile 80	**#suffix** **ntdomain**
INDIVIDUELLE EINSTELLUNGEN (Authenticator und Supplicants/Non-Supplicants)	

/etc/freeradius/**clients.conf**
→ StandardPasswort für Zugriff auf den RADIUS (.. unter client localhost)
→ erlaubte Authenticators
	client {*ip* | *ip***/***mask*} {
	secret = *password*
	shortname = *name*
	nastype = cisco
	}
	.. siehe auch folgende Seiten

/etc/freeradius/**users** → Einstellungen für AAA → Login Benutzer, dot1x: User/Pass, VLANs, MAB	.. siehe folgende Seiten
TESTEN	
# **service freeradius stop** # **freeradius -X**	

/etc/freeradius/radiusd.conf (Auszug IP und Portkonfiguration)

```
# -----------
# Für Authentication/Authorization
# → es darf auf jede IP des Server zugegriffen werden
# → Zielport 1812
# Cisco: (config)# radius-server host IP auth-port 1812
#
listen {
        type = auth
        ipaddr = *
        port = 1812
}
# Für Accounting
# → es darf auf jede IP des Server zugegriffen werden
# → Zielport 1813
# Cisco: (config)# radius-server host IP acct-port 1813
#
listen {
        ipaddr = *
        port = 1813
        type = acct
}
```

/etc/freeradius/clients.conf

```
# -----------
# für die Clients wird das Zugriffs-Passwort für den RADIUS eingestellt
# auf Cisco: (config)# radius-server host IP key cisco
#
client localhost {
        ipaddr = 127.0.0.1
        secret = cisco
        require_message_authenticator = no
}
# ----------
# 3 Clients (alte Konfiguration – in Abhängigkeit zur verwendetet freeradius Version 2.1.12)
# .. die jeweils für die privaten IPv4 Adressräume zuständig sind
# .. "faule" Konfig, damit TN grundsätzlich Zugriff haben
#
client 10.0.0.0/8 {
        secret = cisco
        shortname = CLASS_A
        nastype = cisco
}
client 172.16.0.0/12 {
        secret = cisco
        shortname = CLASS_B
        nastype = cisco
}
client 192.168.0.0/16 {
        secret = cisco
        shortname = CLASS_C
        nastype = cisco
}
```

/etc/freeradius/users

```
# -----------
# Login
# ---
# Login User (Privilege 1)
"hiwi"          Cleartext-Password := "cisco"
                Service-Type = NAS-Prompt-User,
                cisco-avpair = "shell:priv-lvl=1"
# ---
# Login User (normal)
"user"          Cleartext-Password := "cisco"
                Service-Type = NAS-Prompt-User
# ---
# Login User (Privilege 15)
"admin"         Cleartext-Password := "cisco"
                Service-Type = NAS-Prompt-User,
                cisco-avpair = "shell:priv-lvl=15" ,
                Reply-Message = "Moin, %{User-Name}"
# -----------
# dot1x mit automatischer VLAN Konfiguration
# ---
# gast/cisco → VLAN 666
"gast"          Cleartext-Password := "cisco"
                Tunnel-Type = 13,
                Tunnel-Medium-Type = 6,
                Tunnel-Private-Group-Id = 666
# ---
# user10/cisco → VLAN 10
"user10"        Cleartext-Password := "cisco"
                Tunnel-Type = 13,
                Tunnel-Medium-Type = 6,
                Tunnel-Private-Group-Id = 10
# ---
# adm/cisco → VLAN 51 "Mgmt"
"adm"           Cleartext-Password := "cisco"
                Tunnel-Type = 13,
                Tunnel-Medium-Type = 6,
                Tunnel-Private-Group-Id = 51
# -----------
# advanced dot1x EXAMLES
# ---
# MAC ByPass → VLAN 7
"0090cce50986"      Cleartext-Password := "0090cce50986"
                Tunnel-Type = 13,
                Tunnel-Medium-Type = 6,
                Tunnel-Private-Group-Id = 7
# ---
# Time-based User: nachtschicht/geheim → VLAN 7
# alle Tage (any) zwischen 18:00 und 06:00 (wk = wochentags, mo = monday, ..)
#
"nachtschicht"          Cleartext-Password := "geheim",
                Login-Time := "Any1800-0600"
                Tunnel-Type = 13,
                Tunnel-Medium-Type = 6,
                Tunnel-Private-Group-Id = 7
```

Anhang B – RADIUS Server: FreeRADIUS 2.2.3 (Windows 7)

Installation von FreeRADIUS Server auf Windows 7

1. Freeradius.net RADIUSd Server **besorgen**

   http://sourceforge.net/projects/freeradius/files/latest/download
   Datei: FreeRADIUS-server-2.2.3-x86.exe (Stand Januar 2014)

2. .. **auspacken und installieren** nach C:\FreeRADIUS

3. **Einige Dateien anpassen ..**

   - C:\FreeRADIUS\etc\raddb\radiusd.conf
     Port des Servers manuell festlegen (default-Einstellung 0)
     .. nach neuer RFC: 1812 (auth), 1813 (account)
     .. nach alter RFC: 1645 (auth), 1646 (account)
     port = 1645

   - C:\FreeRADIUS\etc\raddb\clients.conf
     Client-Einstellungen (IP Adresse, Key, Art des Clients)
     client 192.168.100.6 {
     secret = geheim
     shortname = switch1
     nastype = cisco
     }

   - C:\FreeRADIUS\etc\raddb\users
     User definieren und Passwort festlegen
     admin Cleartext-Password := "cisco"
     Reply-Message = "Moin, %{User-Name}"

4. .. dann **starten** → Mit der Verknüpfung wird folgende Datei ausgeführt:
   C:\FreeRADIUS\sbin**StartServer.cmd**

Anhang C – TACACS+ Server: tac_plus F4.0.4.26 (Linux)

TACACS+ Funktion steht als OpenSource Server "tacacs+" für Linux/Unix Betriebssysteme zur Verfügung

Installieren

1. Backports in /etc/apt/sources.list aktivierenF4.0.4.26

2. # apt-get install tacacs+

Konfigurieren/Editieren

1. # cd /etc/tacacs+

2. # cp tac_plus.conf tac_plus.conf.org

3. **tac_plus.conf** editieren
   → siehe folgende Seite

4. # vi /etc/default/tacacs+
   DAEMON_OPTS="-C /etc/tacacs+/tac_plus.conf -d 16"

Konfiguration Testen (Syntax)

1. # tac_plus -P -C /etc/tacacs+/tac_plus.conf

Loggen (anderes Terminal) und Starten

1. # tail -f /var/log/tacacs+/tac_plus.log

2. # service tacacs_plus start

/etc/tacacs+/tac_plus.conf

```
# -----------
# Log Einstellungen für Accounting
#
accounting syslog;
accounting file = /var/log/tac_plus.acct
# ---
# das Zugriffs-Passwort für den TACACS+ Server
# auf Cisco: (config)# tacacs-server host IP key cisco
#
key = cisco
#
# ---
# Priv-Level 15 User (Gruppe ADMIN)
# jupp → Authentication über PAM (lokale Linux User/Password Datenbank)
#
user = meinereiner {
        login = PAM
        member = ADMIN
}
# admin → Authentication über Passwort cisco
#
user = admin {
        login = cleartext cisco
        member = ADMIN
}
# Gruppe ADMIN mit Priv-Level 15 Zuordnung
#
group = ADMIN {
        default service = permit
        service = exec { priv-lvl = 15 }
        member = ALL
}
# Gruppe ALL: übergeordnete Gruppe (keine Einstellungen – momentan)
#
group = ALL {
}
# ---
# User hiwi ohne Gruppenmitgliedschaft
# .. mit Kommandobeschränkung → keine Konfiguration, nur Troubleshooting
# auf Cisco: (config)# aaa authorization commands 15 default group tacacs
#
user = hiwi {
        login = cleartext cisco
        cmd = enable { permit .* }
        cmd = configure { deny .* }
        cmd = show { permit .* }
        cmd = debug { permit .* }
        cmd = terminal { permit monitor }
        cmd = ping { permit .* }
        cmd = traceroute { permit .* }
        cmd = exit { permit .* }
}
```