



TU Clausthal

Einrichtung von radsecproxy

Dipl.-Math. Christian Strauf
Rechenzentrum TU Clausthal

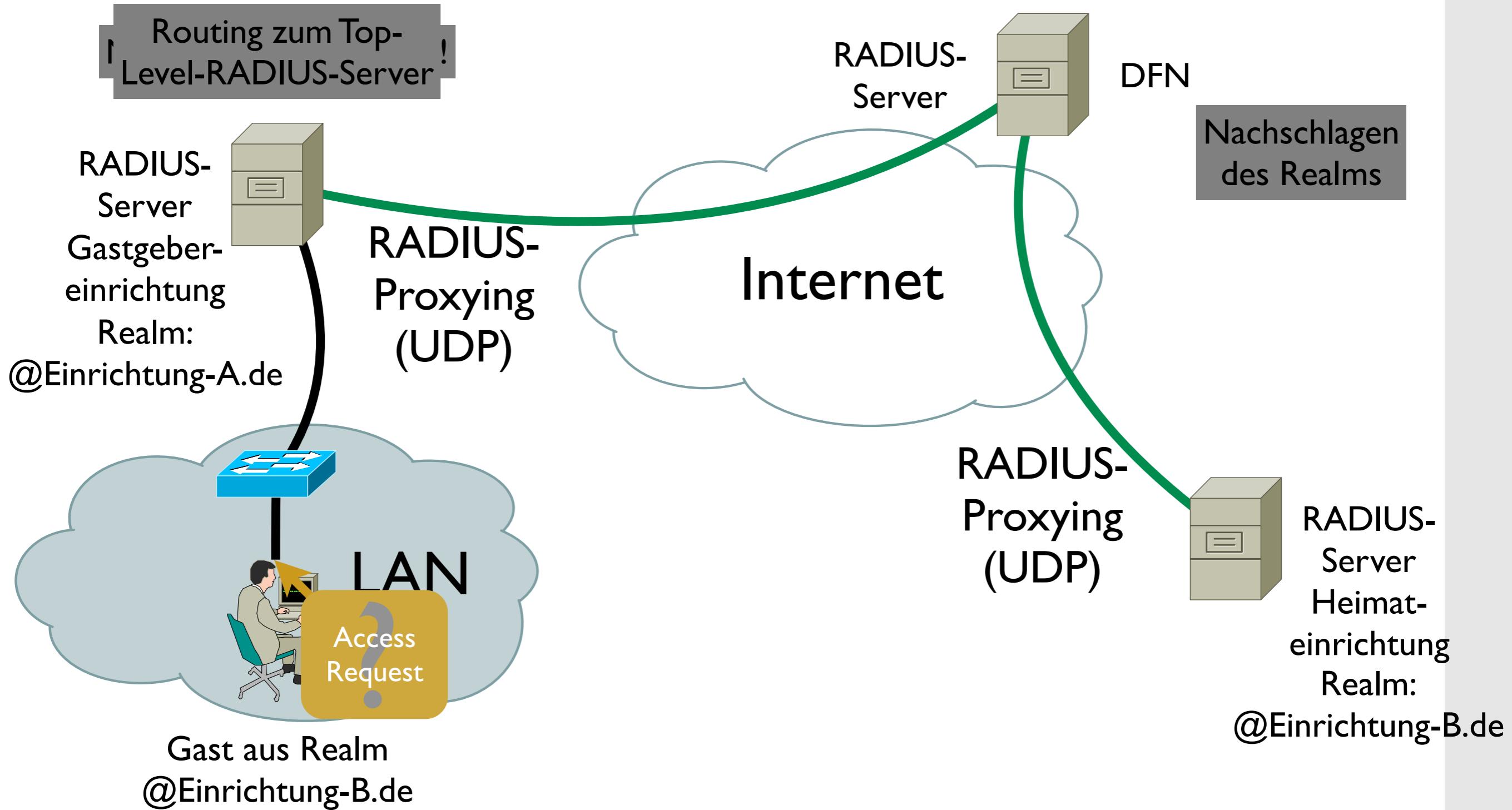


Agenda

- Erinnerung: Funktionsweise von RADIUS
- RadSec - eine Übersicht
- Systemvoraussetzungen
- Installation von radsecproxy
- Konfiguration von radsecproxy
- Debugging



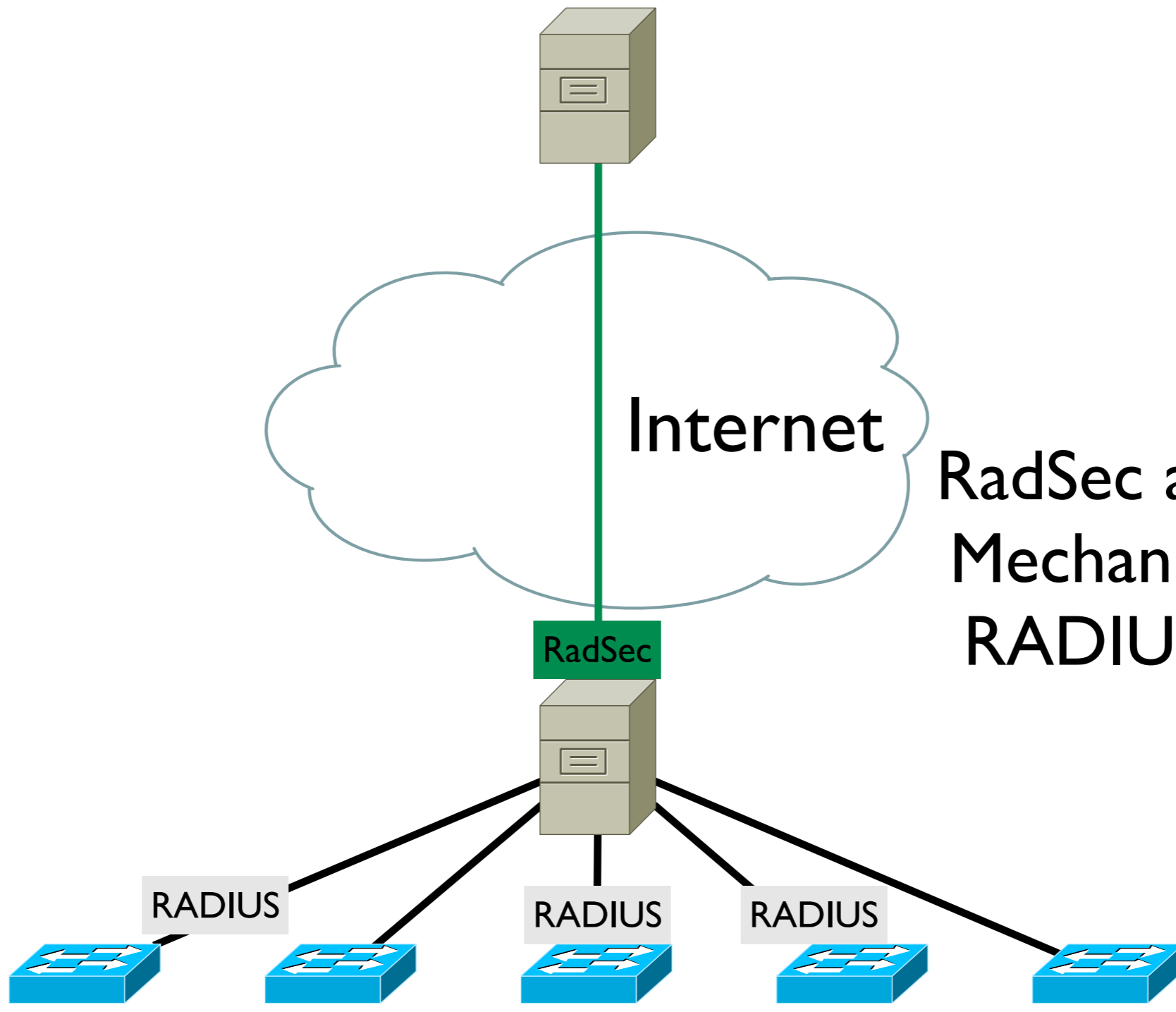
Erinnerung: Funktionsweise eduroam auf Basis von RADIUS





RadSec - eine Übersicht

- RadSec ist ein im Rahmen der IETF entwickeltes Protokoll, das derzeit noch im Draft-Stadium ist:
<http://tools.ietf.org/html/draft-ietf-radext-radsec>
- RadSec verwendet TCP und TLS, um RADIUS-Nachrichten zu verschicken ⇒
Absicherung auf Transportebene



RadSec als Proxy-Mechanismus für RADIUS-Server



Vorteile von RadSec gegenüber RADIUS

- Vertrauensstellung wird nicht mit „Secrets“ wie bei RADIUS sondern über X.509-Zertifikate hergestellt:
- CA-Zertifikate werden verwendet, um Vertrauenswürdigkeit von RadSec-Gegenstellen zu prüfen.
- TLS-Verbindung zwischen RadSec-Client und-Server wird mit Hilfe von Zertifikaten aufgebaut.
- Verbindung per TCP/TLS sorgt für hohe Stabilität auf WAN-Strecken, da Status der Verbindung klar ist; unnötige Timeouts werden verhindert.



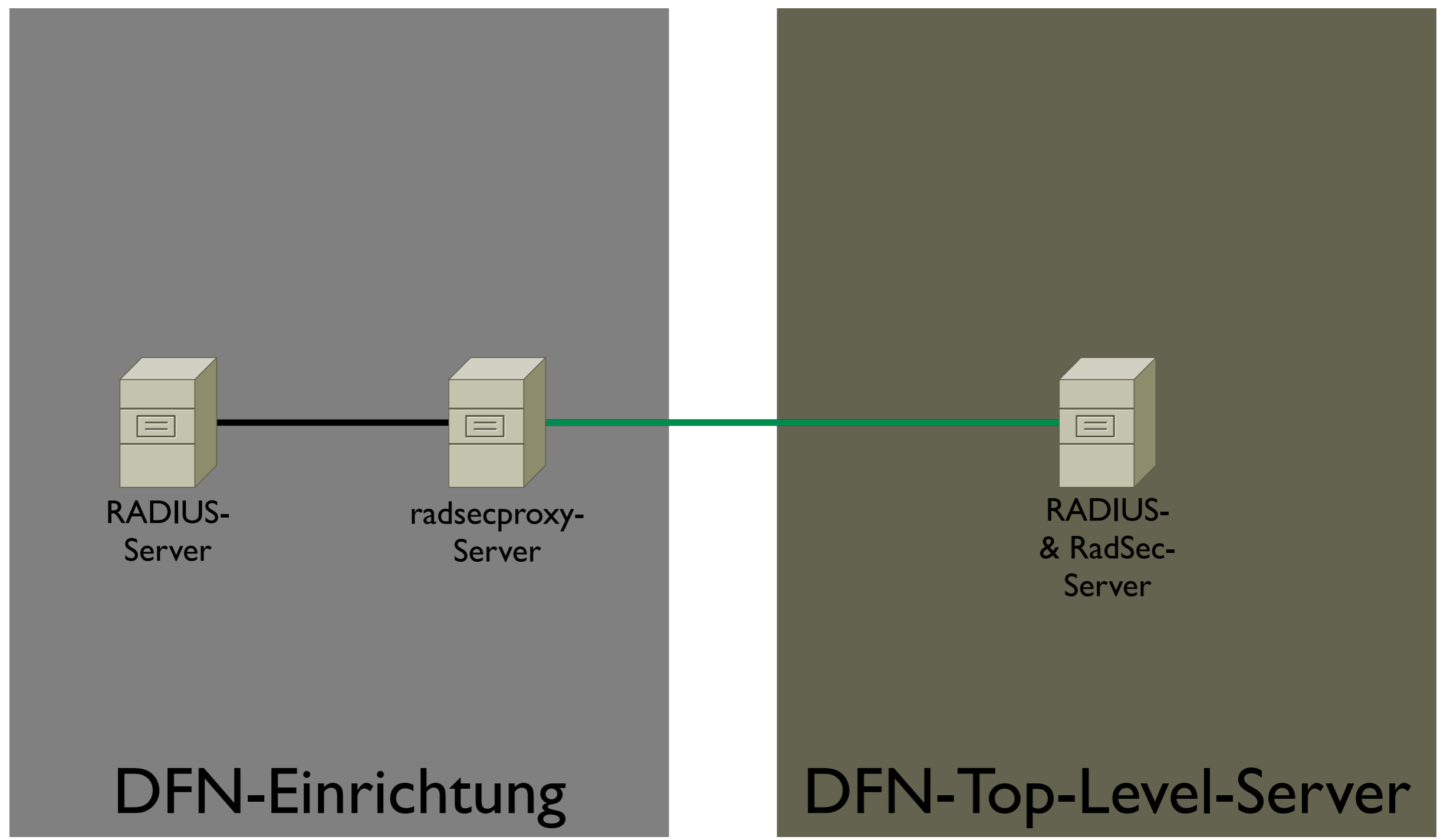
Neuerungen in RadSec gegenüber RADIUS

- Dynamisches RadSec-Server-Lookup
- Beim Roaming eines Users geschieht DNS-Lookup des Realms (SRV-Records).
- SRV-Record beinhaltet RadSec-Server-Namen der Heimateinrichtung.
- RadSec-Server der Gastgebereinrichtung kann RadSec-Server der Heimateinrichtung direkt ansprechen.

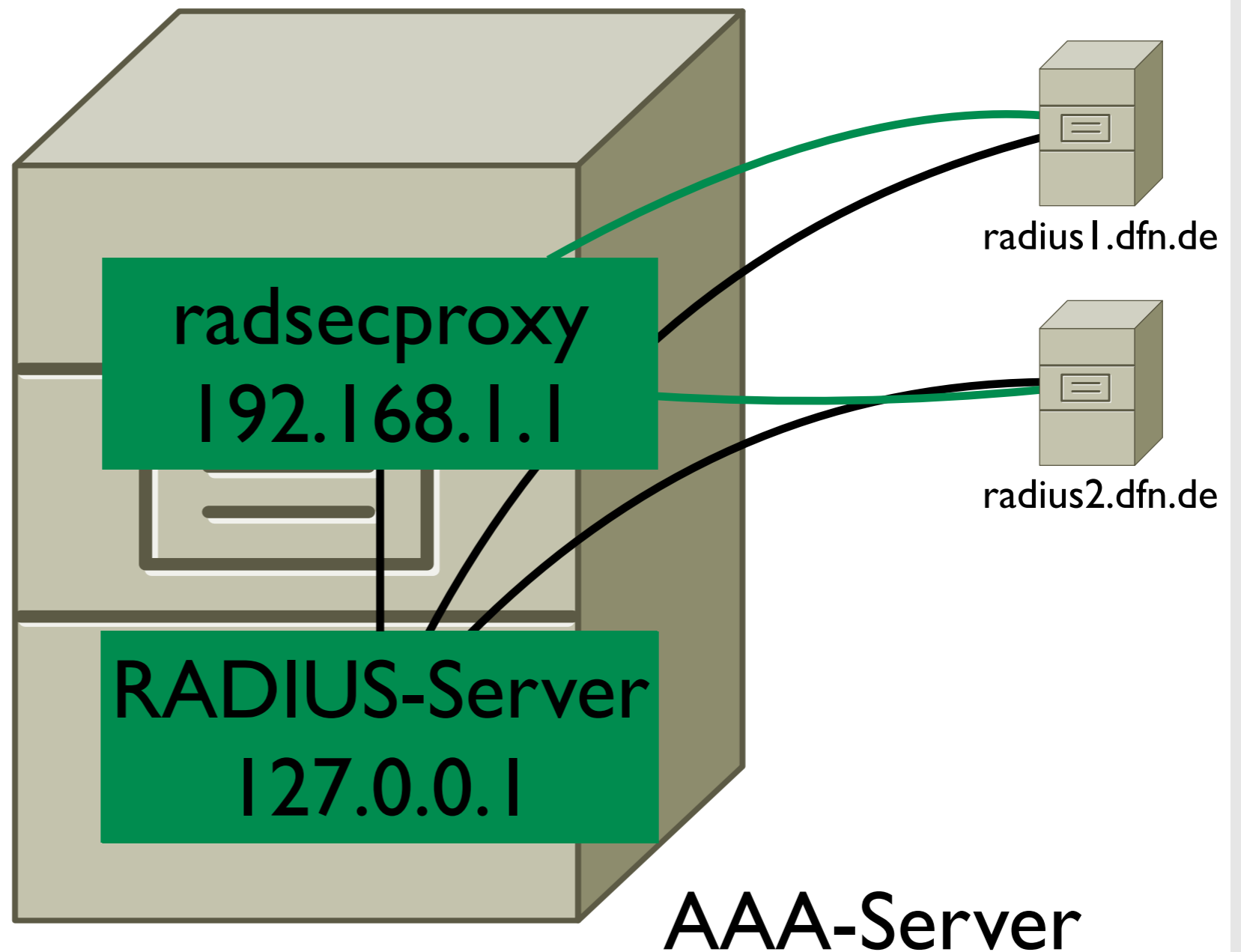


radsecproxy-Überblick

- Entwickelt von Stig Venås (damals UNINETT).
- Plattformübergreifende OpenSource-Implementierung von RadSec als Proxy für RADIUS.
- Idee: Vorhandene RADIUS-Server lokal weiter nutzen und Verbindungen über WAN mit RadSec über radsecproxy realisieren (z.B. für eduroam).



- RADIUS-Verbindung auflösen
- radsecproxy einrichten
- radsecproxy mit DFN-RadSec-Servern verbinden
- Lokale RADIUS-Server mit radsecproxy auf localhost verbinden





Systemvoraussetzungen

- radsecproxy wird in der Regel direkt aus den Quellen gebaut. Es muss daher eine (oft bereits installierte) Development-Umgebung vorhanden sein (GCC, libc-Dev-Pakete etc.).
- OpenSSL mit zugehörigen Headern muss vorhanden sein.
- radsecproxy läuft auf diversen Linux-, BSD-, Solaris- und MacOS X-Systemen.



Installation von radsecproxy

- Quellen kompilieren:
`./configure --prefix=/usr/local`
`make`
`make install`
- Anschließend die Konfigurationsdateien anpassen.

Konfiguration von radsecproxy



```
##
##
## Sektion mit globaler Konfiguration
##
##

#
# Lausche auf 127.0.0.1:2084 (UDP), um RADIUS-Anfragen des lokal auf der
# Maschine laufenden RADIUS-Servers zu empfangen.
#
listenUDP          127.0.0.1:2084

#
# Lausche auf 127.0.0.1:2085 (UDP), um RADIUS-Accounting-Pakete des lokal
# auf der Maschine laufenden RADIUS-Servers zu empfangen.
#
listenUDP          127.0.0.1:2085

#
# Lausche auf der IP-Adresse 192.168.1.1, um RadSec-Anfragen über TCP mit
# TLS verschlüsselt zu empfangen (dies ist die IP und der Port, über den
# der DFN-Top-Level-Server per RadSec mit dem radsecproxy spricht).
#
listenTLS          192.168.1.1:2083
```



```
#
# Einstellen des gewünschten Log-Levels und des Ziels der Log-Nachrichten
#
LogLevel                1
LogDestination file:///var/log/radsecproxy.log

#
# Für alle per TLS verschlüsselten Verbindungen sollen die folgenden
# CA- und Zertifikatseinstellungen verwendet werden:
#
tls default {
    # Dies ist die Kette mit den Zertifikaten Ihrer PKI (z.B. das Ihrer
    # RA, das der DFN-CA (Global) und der Telekom G02, wenn Sie
    # eine Einrichtung mit ausgelagerter CA sind). Die Zertifikatskette
    # können Sie einfach auf der PKI-Seite Ihrer RA herunter laden.
    CACertificateFile    dfn-pki-global-chain.txt

    # Hier werden die X.509-Zertifikate Ihres RADIUS-Servers angegeben.
    # Beachten Sie, dass der private Schlüssel wie bei einem Web-Server
    # ohne Passphrase vorliegen muss.
    CertificateFile      cert.pem
    CertificateKeyFile   key.pem
}
```



```
##
##
## Client-Sektion von radsecproxy
## Hier werden alle Clients angegeben, die den radsecproxy nutzen sollen:
## - Ihr Lokaler RADIUS-Server,
## - Die beiden redundant ausgelegten DFN-Top-Level-RADIUS-Server mit RadSec.
##
##

#
# Zugriff über klassisches RADIUS-Protokoll per UDP für Ihren lokal auf der
# Maschine laufenden RADIUS-Server konfigurieren.
#
client 127.0.0.1 {
    type udp
    secret strenggeheim
}

#
# Zugriff per TLS-verschlüsseltem RadSec von den beiden DFN-TOP-Level-
# RADIUS-Servern, die RadSec sprechen, konfigurieren.
#
client radius1.dfn.de {
    type    tls
}

client radius2.dfn.de {
    type    tls
}
```



```
##
##
## Server-Sektion von radsecproxy
## Hier werden alle Server angegeben, die der radsecproxy nutzen soll:
## - Ihr Lokaler RADIUS-Server,
## - Die beiden redundant ausgelegten DFN-Top-Level-RADIUS-Server mit RadSec.
##
##

#
# radsecproxy soll den lokalen RADIUS-Server über klassisches RADIUS-Protokoll
# per UDP nutzen. Es muss dabei einen Eintrag für die Authentisierung
# und einen für das Accounting geben.
#
server freeradius {
    host 127.0.0.1
    port 1812
    type udp
    secret strenggeheim
}

server freeradius-accounting {
    host 127.0.0.1
    port 1813
    type udp
    secret strenggeheim
}

#
# Die DFN-Top-Level-RADIUS-Server sprechen RadSec und sollen deshalb per TLS
# angesprochen werden.
#
server radius1.dfn.de {
    type    tls
    StatusServer on
}
server radius2.dfn.de {
    type    tls
    StatusServer on
}
```



```
##
##
## Realm-Sektion
## Hier wird festgelegt, auf welche Server Anfragen zur Authentisierung für
## welche Realms ("@...") geschickt werden sollen.
##
##

#
# Anfragen Ihrer eigenen Nutzer beim Roaming außerhalb werden auf Ihren
# lokal auf der Maschine laufenden RADIUS-Server geschickt (dito für
# Accounting-Pakete von außerhalb).
#
realm /@einrichtung\.de$/ {
    server freeradius
    accountingserver freeradius-accounting
}

#
# Ansonsten werden alle Realms, die nicht konkret definiert werden, an
# die Top-Level-RADIUS-Server des DFN geleitet (dito für Accounting-
# Pakete).
#
realm * {
    server radius1.dfn.de
    server radius2.dfn.de
    accountingserver radius1.dfn.de
    accountingserver radius2.dfn.de
}
```



```
##
##
## Client-Sektion von radsecproxy
## Hier werden alle Clients angegeben, die den radsecproxy nutzen sollen:
## - Ihr Lokaler RADIUS-Server,
## - Die beiden redundant ausgelegten DFN-Top-Level-RADIUS-Server mit RadSec.
##
##

#
# Zugriff über klassisches RADIUS-Protokoll per UDP für Ihren lokal auf der
# Maschine laufenden RADIUS-Server konfigurieren.
#
client 127.0.0.1 {
    type udp
    secret strenggeheim
}

#
# Zugriff per TLS-verschlüsseltem RadSec von den beiden DFN-TOP-Level-
# RADIUS-Servern, die RadSec sprechen, konfigurieren.
#
client radius1.dfn.de {
    # Wende die Nutzernamen-Rewrite-Regel an.
    rewriteOut UserName
    type    tls
}

client radius2.dfn.de {
    # Wende die Nutzernamen-Rewrite-Regel an.
    rewriteOut UserName
    type    tls
}
```



Debugging

- Heraufsetzen des Log-Levels in der Konfigurationsdatei: „LogLevel <n>“; je größer „n“, desto mehr Informationen werden ausgegeben.
- Aufrufen von radsecproxy auf der Kommandozeile:
radsecproxy -f -c \
 <Konfigurationsdatei> \
 -d <Debug-Level>
(„-f“, um im Vordergrund zu starten).

Vielen Dank für Ihre
Aufmerksamkeit!

