

DFNRoaming / eduroam Workshop

Einsatz von X.509 SSL Zertifikaten im FreeRADIUS Server

26.04.2007
Berlin

Reimer Karlsen-Masur, DFN-PCA

Technischer Kontakt DFN-PCA:
dfnpca@dfn-cert.de

Administrativer Kontakt DFN-Verein:
pki@dfn.de

Gliederung

- X.509 / TLS / SSL Zertifikate?!?
- DFN-PKI
- Zertifikate in einer WLAN / 802.1x / RADIUS Architektur
- 802.1x EAP Varianten mit Zertifikaten
- Zertifikatsprofile
- Konfigurationsbeispiele
 - FreeRADIUS PEAP / EAP-TTLS / EAP-TLS
 - OpenLDAP als Passwort-Datenbank für FreeRADIUS
 - Windows XP PEAP
 - Windows XP EAP-TLS
 - Windows XP EAP-TTLS (SecureW2)

Was sind Zertifikate?

- Zertifikate nach X.509v3 Standard
- Werden von Zertifizierungsstellen (CAs) ausgestellt
- Beglaubigen eine digitale Identität z.B. von End-Entitys (EEs) wie Servern (Server-Zertifikat z.B. „radius.dfn.de“) und Nutzern (Client / Nutzer-Zertifikat z.B. „Ralf Paffrath vom DFN-Verein“) oder von CAs (CA-Zertifikat z.B. „DFN Global CA - G01“)

Was sind Zertifikate? (2)

- Kryptographie, geheimer Schlüssel, öffentlicher Schlüssel, Metadaten
- Hierarchische Verkettung der Zertifikate von der Wurzel-CA über zwischen CAs zum EE-Zertifikat
- Sperrlisten (CRLs) mit ungültigen Zertifikaten durch die CA erstellt

Server-Zertifikat

Serial Number: 123456789 (0x1a2b3c4)
Issuer: C=DE, O=DFN-CERT Services GmbH,
CN=DFN-CERT Services GmbH CA - G02
Validity
Not Before: Apr 3 15:20:57 2007 GMT
Not After : Apr 1 15:20:57 2012 GMT
Subject: C=DE, O=DFN-CERT Services GmbH, CN=radius-test.dfn-cert.de
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
RSA Public Key: (2048 bit)
X509v3 extensions:
X509v3 Basic Constraints:
CA:FALSE
X509v3 Key Usage:
Digital Signature, Non Repudiation, Key Encipherment
X509v3 Extended Key Usage:
TLS Web Client Authentication, TLS Web Server Authentication
X509v3 Subject Alternative Name:
email:dfnpca@dfn-cert.de
X509v3 CRL Distribution Points:
URI:http://cdpl.pca.dfn.de/dfn-cert-services-gmbh-ca/
pub/crl/g_cacrl.crl
Authority Information Access:
CA Issuers - URI:http://cdpl.pca.dfn.de/dfn-cert-services-gmbh-ca/
pub/cacert/g_cacert.crt

Nutzer-Zertifikat

Serial Number: 123456789 (0x1a2b3c4)
Issuer: CN=DFN-Verein-GS-CA - G02,OU=Geschaeftsstelle,O=DFN-Verein,C=DE
Validity
Not Before: Jan 18 15:01:01 2007 GMT
Not After : Jan 17 15:01:01 2010 GMT
Subject: CN=Ralf Paffrath,OU=Geschaeftsstelle,O=DFN-Verein,C=DE
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
RSA Public Key: (2048 bit)
X509v3 extensions:
X509v3 Basic Constraints:
CA:FALSE
X509v3 Key Usage:
Digital Signature, Non Repudiation, Key Encipherment
X509v3 Extended Key Usage:
TLS Web Client Authentication, E-mail Protection
X509v3 Subject Alternative Name:
email:paffrath@dfn.de
X509v3 CRL Distribution Points:
URI:http://cdpl.pca.dfn.de/dfn-verein-gs-ca/pub/crl/g_cacrl.crl
Authority Information Access:
CA Issuers - URI:http://cdpl.pca.dfn.de/dfn-verein-gs-ca/
pub/cacert/g_cacert.crt

CA-Zertifikat

```
Serial Number: 123456789 (0x1a2b3c4)
Issuer: C=DE, O=DFN-Verein, OU=DFN-PKI, CN=DFN-Verein PCA Global - G01
Validity
  Not Before: Feb 14 11:50:09 2007 GMT
  Not After : Feb 13 00:00:00 2019 GMT
Subject: C=DE, O=DFN-CERT Services GmbH,
         CN=DFN-CERT Services GmbH CA - G02
Subject Public Key Info:
  Public Key Algorithm: rsaEncryption
  RSA Public Key: (2048 bit)
X509v3 extensions:
  X509v3 Basic Constraints: critical
    CA:TRUE
  X509v3 Key Usage:
    Certificate Sign, CRL Sign
  X509v3 CRL Distribution Points:
    URI:http://cdpl.pca.dfn.de/global-root-ca/pub/crl/cacrl.crl
  Authority Information Access:
    CA Issuers - URI:http://cdpl.pca.dfn.de/global-root-ca/
    pub/cacert/cacert.crt
```

Zertifikatsketten

The image shows two overlapping windows from a certificate management tool. The left window, titled 'Zertifikat-Ansicht: "Ralf Paffrath"', displays a certificate hierarchy starting with 'Deutsche Telekom Root CA 2', followed by 'DFN-Verein PCA Global - G01', 'DFN-Verein-GS-CA - G02', and finally 'Ralf Paffrath'. It also shows the 'Zertifikats-Layout' (Certificate Structure) with fields like 'Inhaber', 'Public-Key-Algorithmus des Inhabers', and 'Erweiterungen'. The 'Feld-Wert' (Field-Value) section lists 'Nicht kritisch', 'TLS Web Client Authentication (1.3.6.1.5.5.7.3.2)', and 'E-mail protection (1.3.6.1.5.5.7.3.4)'. The right window, titled 'Zertifikat-Ansicht: "radius-test.dfn-cert.de"', shows a similar hierarchy ending with 'radius-test.dfn-cert.de'. Its 'Zertifikats-Layout' includes 'Erweiterungen' such as 'Zertifikats-Basisbedingungen', 'Verwendung eines Zertifikatsschlüssels', and 'Erweiterter Schlüsselgebrauch'. The 'Feld-Wert' section lists 'Nicht kritisch', 'TLS Web Client Authentication (1.3.6.1.5.5.7.3.2)', and 'TLS Web Server Authentication (1.3.6.1.5.5.7.3.1)'. A 'Schließen' button is visible at the bottom right of the right window.

Zertifikat-Sperrliste (CRL)

Issuer: /C=DE/O=DFN-CERT Services GmbH
/CN=DFN-CERT Services GmbH CA - G02

Last Update: Apr 17 14:50:39 2007 GMT

Next Update: May 17 14:50:39 2007 GMT

CRL extensions:

X509v3 CRL Number:

16

Revoked Certificates:

Serial Number: 0A444028

Revocation Date: Apr 17 13:40:38 2007 GMT

Serial Number: 0A445799

Revocation Date: Apr 17 14:50:37 2007 GMT

Warum Zertifikate?

- Absicherung der elektronischen Kommunikation
- Nutzer wollen sicherstellen, dass sie mit dem „richtigen“ Server sprechen (Server-Authentisierung)
- Server-Betreiber wollen wissen, mit welchem Nutzer ihre Server sprechen, damit Zugriffsautorisierungen auf Basis der authentisierten Nutzeridentität gefällt werden können (Client-Authentisierung)
- Der übermittelte Inhalt (z.B. Passwörter, Daten) soll vertraulich & integer bleiben (Verschlüsselung & Unversehrtheit der Daten)

- Server- und Nutzerzertifikate für die Verwendung im DFNRoaming gibt es von der DFN-PKI (www.pki.dfn.de)
- Teilnehmende Einrichtungen an der DFN-PKI <https://www.pki.dfn.de/teilnehmer>
- Wenn Ihre Einrichtung eine ausgelagerte CA beim DFN-Verein betreiben lässt, dann können Zertifikatanträge unter <https://pki.pca.dfn.de/<Kürzel>-ca/pub> gestellt werden
- DFN-Test-PKI CA steht zur Verfügung

Certificate Signing Request (CSR)

- Für Zertifikatanträge für Server wird ein CSR mit 2048 Bit RSA Schlüssellänge benötigt
- entweder direkt aus der Applikation heraus (z.B. IIS) erzeugt
- oder per `openssl` auf der Kommandozeile z.B.

```
openssl req -newkey rsa:2048 -nodes  
-out request-csr.pem -keyout pub-sec-  
key.pem -subj '/C=DE/O=Universitaet  
ABC/CN=radius.uni-  
abc.de/emailAddress=radiusmaster@uni-  
abc.de'
```

- Um bei der Authentifizierung von (roamenden) Nutzern am AccessPoint (AP) der (Gast-) Einrichtung Server- oder Nutzer-Zertifikate verwenden zu können, müssen sowohl der Supplikant beim Nutzer als auch der AP das IEEE 802.1x Authentisierungsprotokoll unterstützen

- Dabei kann die WLAN Verbindung (Supplikant <-> AP) durch folgende Verfahren geschützt werden:
 - WEP (mit einem mehr oder weniger „geheimen“ Passwort)
 - WPA & WPA2 (Enterprise / EAP Variante)
- Login-Credentials werden innerhalb des 802.1x Protokolls über das Extensible Authentication Protocol (EAP) übertragen

- RADIUS stellt etliche Verfahren zur Nutzerauthentifizierung zur Verfügung
- Bei 802.1x läuft die Authentisierung zwischen Supplikanten und RADIUS-Server der *Heim*einrichtung via EAP, welches vom AP sowie „auf dem Weg liegenden“ RADIUS-Proxy-Servern entsprechend weitergeleitet wird
- EAP wird durch entsprechende Module erweitert, die die das eigentliche Authentisierungsverfahren implementieren
- Heimat-RADIUS-Server und Supplikant müssen dabei ein gemeinsames Verfahren unterstützen, dabei dürfen beide Seiten durchaus mehrere EAP Verfahren unterstützen

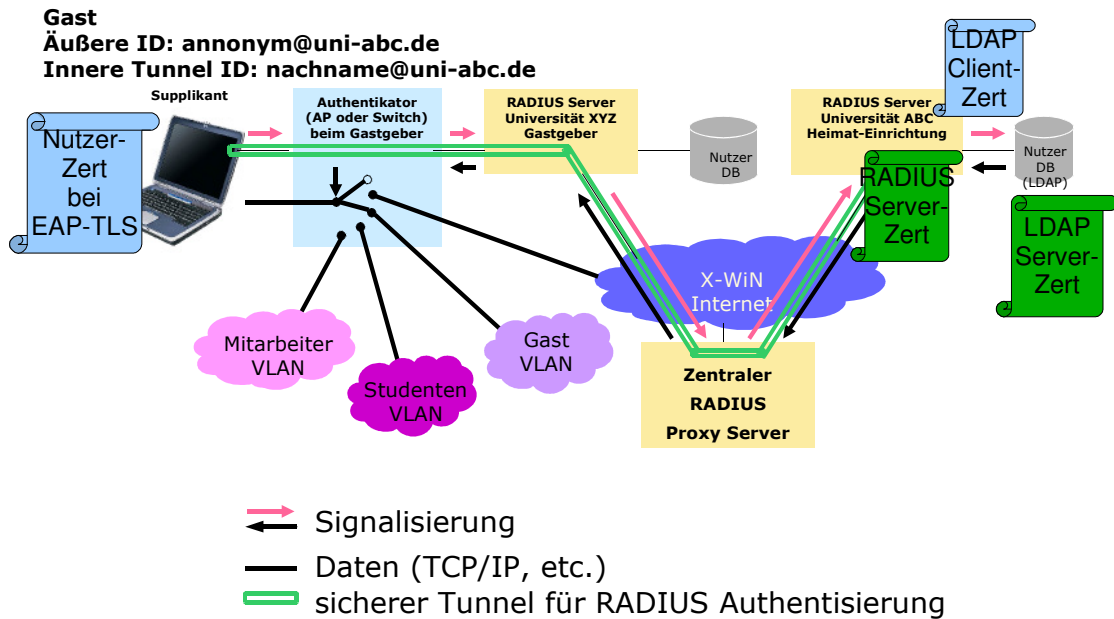
- EAP-TTLS (Tunneled Transport Layer Security)
 - RADIUS-Server Zertifikat in der Heimateinrichtung
 - beim roamenden Nutzer: Username / Passwort
- PEAP (Protected Extensible Authentication Protocol)
 - RADIUS-Server Zertifikat in der Heimateinrichtung
 - beim roamenden Nutzer: Username / Passwort
- EAP-TLS (Transport Layer Security)
 - RADIUS-Server Zertifikat in der Heimateinrichtung
 - beim roamenden Nutzer: Client-Zertifikat

EAP-Varianten mit Zertifikatnutzung (2)

- bei EAP-TTLS und PEAP werden die Credentials als Username / Passwort übertragen
 - abgesichert durch einen TLS Tunnel vom Supplikanten bis zum RADIUS-Server der Heimateinrichtung
 - die eigentliche Passwortübermittlung geschieht dabei meist mittels eines Challenge-Response-Verfahrens – z.B. Microsofts Challenge Handshake Authentication Protocol (MS-CHAPv2 oder MS-CHAP)

EAP-Varianten mit Zertifikatnutzung (3)

- bei EAP-TLS wird die Client-Authentisierung mittels des Client-Zertifikats innerhalb des TLS Protokolls durchgeführt. Keine Übertragung von Username/Passwort nötig



Wo werden EE-Zertifikate in einer RADIUS Architektur eingesetzt?

- Als Server-Zertifikat auf dem RADIUS-Server Heimateinrichtung zur Authentisierung des RADIUS-Servers gegenüber den Supplikanten (EAP-TTLS, PEAP, EAP-TLS)
- Als Server-Zertifikat auf dem LDAP-Server zur Authentisierung des LDAP-Servers gegenüber dem integrierten LDAP Client des RADIUS-Servers (LDAPS oder STARTTLS)
- Als Client-Zertifikat auf dem RADIUS-Server zur Authentisierung des in den RADIUS-Server integrierten LDAP Client gegenüber dem LDAP-Server (LDAPS oder STARTTLS mit Client-AuthN)
- Als Client-Zertifikat im Supplikanten zur Authentisierung des Nutzers gegenüber dem RADIUS-Server (EAP-TLS)

Wo werden CA-Zertifikate in einer RADIUS Architektur eingesetzt?

- Überall, wo Zertifikatsketten zur Validierung eines EE-Zertifikats benötigt werden:
 - RADIUS-Server in der Heimateinrichtung
 - LDAP-Server in der Heimateinrichtung
 - Supplikant beim roamenden Nutzer

Zertifikatsprofile (1)

- Server-Zertifikat zur kombinierten Nutzung als RADIUS-Server- und ggf. als LDAP-Client-Zertifikat (im RADIUS-Server)
 - Extended Key Usage
 - PKIX serverAuth (OID: 1.3.6.1.5.5.7.3.1)
 - PKIX clientAuth (OID: 1.3.6.1.5.5.7.3.2)
 - passendes Zertifikatsprofil bei ausgelagerter CA in der DFN-PKI (Global & Classic): RADIUS-Server

Zertifikatsprofile (2)

- Server-Zertifikat zur kombinierten Nutzung als LDAP-Server- (und ggf. als LDAP-Client-Zertifikat für Replikationen etc)
 - Extended Key Usage
 - PKIX serverAuth (OID: 1.3.6.1.5.5.7.3.1)
 - PKIX clientAuth (OID: 1.3.6.1.5.5.7.3.2)
 - passendes Zertifikatsprofil bei ausgelagerter CA in der DFN-PKI (Global & Classic): LDAP-Server

Zertifikatsprofile (3)

- Client-Zertifikat auf dem Supplikanten des roamenden Nutzers (nur bei EAP-TLS nötig)
 - Extended Key Usage
 - PKIX clientAuth (OID: 1.3.6.1.5.5.7.3.2)
 - PKIX emailProtection (OID: 1.3.6.1.5.5.7.3.4) (eigentlich für EAP-TLS optional)
 - **NICHT** Microsoft SmartCard Logon (OID: 1.3.6.1.4.1.311.20.2.2)
 - Macht im Windows-eigenen Supplikanten Probleme
 - passendes Zertifikatsprofil bei ausgelagerter CA in der DFN-PKI (Global & Classic): 802.1x User

FreeRADIUS 1.1.6
Minimale Beispielkonfiguration
für
EAP-TTLS mit MS-CHAPv2
oder
PEAP mit MS-CHAPv2
oder
EAP-TLS

www.freeradius.org

FreeRADIUS Voraussetzungen

- Die Nutzer Passwörter bei EAP-TTLS und PEAP müssen für MS-CHAPv2 entweder im Klartext vorliegen (in users Datei oder LDAP-Verzeichnisdienst) oder auf einem Windows PDC via ntlm_auth
 - <http://deployingradius.com/documents/protocols/compatibility.html>
 - <http://deployingradius.com/documents/protocols/oracles.html>
- Für EAP-TLS muss entsprechend bei jedem Nutzer ein Client-Zertifikat vorliegen

- FreeRADIUS Konfigurationsverzeichnis in und unterhalb von `/usr/local/etc/raddb`
- Den RADIUS Daemon zum Debuggen starten mittels
`/usr/local/sbin/radiusd -X`

- leere Dateien
 - `/usr/local/etc/raddb/preproxy_users`
 - `/usr/local/etc/raddb/acct_users`
 - `/usr/local/etc/raddb/hints`
 - `/usr/local/etc/raddb/huntgroups`
- Verzeichnisse für vertrauenswürdige CAs und deren CRLs, bleiben je nach Fall ggf. leer(!)
 - `/usr/local/etc/raddb/certs/trustedCAsForRoamingClients/`
 - `/usr/local/etc/raddb/certs/trustedCAs-ldap-backend/`
 - `/usr/local/openldap/certs/trustedCAs-ldap-server/`

- Datei mit geheimen Schlüssel des Radius-Servers
/usr/local/etc/raddb/certs/radius-domain-de-key.pem
- Datei mit dem dazu passenden X.509 Zertifikat und der zugehörigen CA-Kette
/usr/local/etc/raddb/certs/radius-domain-de-cert-and-chain.pem
- Datei mit Diffie-Hellman Parametern
/usr/local/etc/raddb/certs/dh erzeugt mittels `openssl dhparam -out /usr/local/etc/raddb/certs/dh 1024`

```
# /usr/local/etc/raddb/radiusd.conf
prefix = /usr/local
exec_prefix = ${prefix}
sysconffdir = ${prefix}/etc
localstatedir = ${prefix}/var
sbindir = ${exec_prefix}/sbin
logdir = ${localstatedir}/log/radius
raddbdir = ${sysconffdir}/raddb
radacctdir = ${logdir}/radacct
confdir = ${raddbdir}
run_dir = ${localstatedir}/run/radiusd
log_file = ${logdir}/radius.log
libdir = ${exec_prefix}/lib
pidfile = ${run_dir}/radiusd.pid
```

Konfig-Datei - radiusd.conf (2)

```
max_request_time = 30
delete_blocked_requests = no
cleanup_delay = 5
max_requests = 1024
listen {
    ipaddr = radius.domain.de
    port = 1812
    type = auth
}
hostname_lookups = no
allow_core_dumps = no
regular_expressions = yes
extended_expressions = yes
log_stripped_names = no
log_auth = no
log_auth_badpass = no
log_auth_goodpass = no
```

Konfig-Datei - radiusd.conf (3)

```
usercollide = no
lower_user = no
lower_pass = no
nospace_user = no
nospace_pass = no

security {
    max_attributes = 200
    reject_delay = 1
    status_server = no
}

proxy_requests = yes
$INCLUDE ${confdir}/proxy.conf
$INCLUDE ${confdir}/clients.conf
snmp = no
```

Konfig-Datei - radiusd.conf (4)

```
thread pool {
    start_servers = 5
    max_servers = 32
    min_spare_servers = 3
    max_spare_servers = 10
    max_requests_per_server = 0
}
```

Konfig-Datei - radiusd.conf (5)

```
modules {
    $INCLUDE ${confdir}/eap.conf

    mschap {
        use_mppe = yes
        require_encryption = yes
        require_strong = yes
        with_ntdomain_hack = no
    }

    realm suffix {
        format = suffix
        delimiter = "@"
        ignore_default = no
        ignore_null = no
    }
}
```

Konfig-Datei - radiusd.conf (6)

```
ldap {
  ldap_debug = -1 # maximales Logging von LDAP Events

  server = "ldap.domain.de"
  #port = 636 # fuer ldaps

  timeout = 20
  timelimit = 20
  net_timeout = 10

  identity = "cn=ldapadmin,dc=domain,dc=de"
  password = "secret"

  basedn = "dc=domain,dc=de"

  #filter = "(uid=%{Stripped-User-Name:-%{User-Name}})"
  filter = "(uid=%{User-Name})"
```

Konfig-Datei - radiusd.conf (7)

```
# Entweder keines der 2 folgenden oder entweder nur
start_tls = yes
# oder
#tls_mode = yes
# setzen (nur fuer ldaps auf speziellem ldaps Port)

# CA Kette des LDAP Server-Zertifikats und CA CRLs
# c_rehash; bei CRL Update den Radius-Server neu starten.
tls_cacertdir = ${raddbdir}/certs/trustedCAs-ldap-backend/

# Client-Zertifikat fuer Authentisierung am LDAP Server
tls_certfile = \
  ${raddbdir}/certs/radius-domain-de-cert-and-chain.pem
tls_keyfile = ${raddbdir}/certs/radius-domain-de-key.pem

tls_randfile = /dev/urandom
tls_require_cert = demand
```

Konfig-Datei - radiusd.conf (8)

```
dictionary_mapping = ${raddbdir}/ldap.attrmap

ldap_connections_number = 5

password_header = "{clear}"
password_attribute = userPassword

set_auth_type = no
do_xlat = yes
compare_check_items = no
access_attr_used_for_allow = yes
}
```

Konfig-Datei - radiusd.conf (9)

```
files {
    usersfile = ${confdir}/users
    acctusersfile = ${confdir}/acct_users
    preproxy_usersfile = ${confdir}/preproxy_users
    compat = no
}

attr_filter {
    attrsfile = ${confdir}/attrs
}
}
```

Konfig-Datei - radiusd.conf (10)

```
authorize {
  attr_filter
  mschap
  suffix
  eap
  files
  ldap
}
authenticate {
  Auth-Type MS-CHAP {
    mschap
  }
  eap
}
post-proxy {
  attr_filter
  eap
}
```

Konfig-Datei - eap.conf (1)

```
# /usr/local/etc/raddb/eap.conf
eap {
  default_eap_type = ttls
  timer_expire      = 60
  ignore_unknown_eap_types = no
  cisco_accounting_username_bug = no
  tls {
    # Server-Zertifikat fuer den Radius-Server
    private_key_password = "password-for-radius-server-key"
    private_key_file = \
      ${raddbdir}/certs/radius-domain-de-key.pem
    certificate_file = \
      ${raddbdir}/certs/radius-domain-de-cert-and-chain.pem
    # erlaubte Krypto-Cipher, absteigend nach Staerke sortiert
    cipher_list = \
      "@STRENGTH:HIGH:MEDIUM:!LOW:!EXP:!aNULL:!eNULL:!MD5:!ADH"
```

Konfig-Datei - eap.conf (2)

```
# Fuer EAP-TLS:
# leeres Verzeichnis bei EAP-TTLS/PEAP oder bei EAP-TLS
# vertraute Client-CA Zertifikate, dann mit c_rehash
# entsprechende Symlinks anlegen
CA_path = ${raddbdir}/certs/trustedCAsForRoamingClients/

verify_depth = 1 # Prueftiefe 0 = beliebig lange CA-Kette

# Wenn zusaetzlich aktuelle CRLs in CA_path vorhanden,
# dann wieder c_rehash fuer Symlinks verwenden;
# bei CRL Update den RADIUS-Server neu starten.

check_crl = yes

#check_cert_issuer = \
# "/C=DE/O=Test-PKI/OU=ONLY FOR TESTING PURPOSES/CN=Test-PKI CA"
#check_cert_cn = %{Stripped-User-Name}
```

Konfig-Datei - eap.conf (3)

```
rsa_key_exchange = yes
dh_key_exchange = yes
rsa_key_length = 2048
dh_key_length = 1024

# Diffie-Hellmann Parameter-Datei fuer DH Krypto
dh_file = ${raddbdir}/certs/dh

random_file = /dev/urandom
fragment_size = 1024
include_length = yes
}
```

Konfig-Datei - eap.conf (4)

```
ttls {
    default_eap_type = mschapv2
    copy_request_to_tunnel = no
    use_tunneled_reply = no
}
peap {
    default_eap_type = mschapv2
    copy_request_to_tunnel = no
    use_tunneled_reply = no
    proxy_tunneled_request_as_eap = no
}
mschapv2 {
}
}
```

Konfig-Datei - users

```
# /usr/local/etc/raddb/users

"sample-user" Cleartext-Password := "password for this user"

"One Rejected User"
    Auth-Type := Reject

DEFAULT User-Name == "[Aa][Nn][Oo][Nn][Yy][Mm][Oo][Uu][Ss]$"
    Auth-Type := Reject

DEFAULT User-Name == "[Aa][Nn][Oo][Nn][Yy][Mm][Oo][Uu][Ss]@.*$"
    Auth-Type := EAP

DEFAULT Realm == NULL
    Auth-Type := Reject
```

Konfig-Datei - dictionary

```
# /usr/local/etc/raddb/dictionary

$INCLUDE /usr/local/share/freeradius/dictionary
```

Konfig-Datei - clients.conf

```
# /usr/local/etc/raddb/clients.conf
client local-wlan-ap.domain.de {
    secret = "local wlan ap shared secret"
    shortname = local-wlan-ap
}
client radius1.dfn.de {
    secret = "shared secret1"
    shortname = dfnroaming-top-level-radius1
    nastype = other
}

client radius2.dfn.de {
    secret = "shared secret2"
    shortname = dfn-roaming-top-level-radius2
    nastype = other
}
```

Konfig-Datei - proxy.conf (1)

```
# /usr/local/etc/raddb/proxy.conf
proxy server {
    synchronous = no
    retry_delay = 5
    retry_count = 3
    dead_time = 120
    default_fallback = yes
    post_proxy_authorize = no
}

realm domain.de {
    type      = radius
    authhost  = LOCAL
}

realm NULL {
    type      = radius
    authhost  = LOCAL
}
```

Konfig-Datei - proxy.conf (2)

```
realm DEFAULT {
    type      = radius
    authhost  = radius1.dfn.de:1812
    accthost  = radius1.dfn.de:1813
    secret    = "shared secret1"
    ld_flag   = fail_over
    nostrip
}

realm DEFAULT {
    type      = radius
    authhost  = radius2.dfn.de:1812
    accthost  = radius2.dfn.de:1813
    secret    = "shared secret2"
    ld_flag   = fail_over
    nostrip
}
```

Konfig-Datei - attrs

```
# /usr/local/etc/raddb/attrs

DEFAULT
    EAP-Message =* ANY,
    Message-Authenticator =* ANY,
    User-Name =~ ".+",
    Reply-Message =* ANY,
    MS-MPPE-Recv-Key =* ANY,
    MS-MPPE-Send-Key =* ANY,
    State =* ANY
```

Konfig-Datei - ldap.attrmap

```
# /usr/local/etc/raddb/ldap.attrmap

checkItem    $GENERIC$        radiusCheckItem
replyItem    $GENERIC$        radiusReplyItem
```

Erweiterte Konfiguration des integrierten LDAP Klienten von FreeRADIUS (1)

```
# /usr/local/etc/openldap/ldap.conf
loglevel -1 # alle LDAP Client-Events loggen
TLS_RANDFILE /dev/urandom
# erlaubte Krypto-Cipher, absteigend nach Staerke sortiert
TLS_CIPHER_SUITE \
  @STRENGTH:HIGH:MEDIUM:!LOW:!EXP:!aNULL:!eNULL:!MD5:!ADH
# LDAP Server muss ein gueltiges Server-Zertifikat haben
TLS_REQCERT demand
# vertraenswuerdige CAs fuer LDAP-Server-Zertifikate und deren
# CRLs (fetch-crl, c_rehash wie im FreeRADIUS Beispiel)
TLS_CACERTDIR \
  /usr/local/etc/openldap/certs/trustedCAs-ldap-server/
# LDAP Server-Zertifikate in CRLs pruefen
TLS_CRLCHECK all
```

Erweiterte Konfiguration des integrierten LDAP Klienten von FreeRADIUS (2)

```
# LDAP Client Zertifikat-Konfiguration fuer ldap(search|add|modify)
#TLS_CERT \
# /usr/local/etc/openldap/certs/ldap-machine-client-cert-and-chain.pem
# und der entsprechende Schluessel zum Client-Zertifikat
#TLS_KEY /usr/local/etc/openldap/certs/ldap-machine-client-key.pem
```

- Die CA-Zertifikatskette für die Client-Authentisierung bei EAP-TLS wird in einzelnen PEM Dateien mit `.pem` Dateiendung nach
 - `/usr/local/etc/raddb/certs/trustedCAsForRoamingClients/`
kopiert
- Dann wird in diesem Verzeichnis `c_rehash ./` aufgerufen
- Wenn kein EAP-TLS verwendet werden soll, dann bleibt dieses Verzeichnis leer!

- Die CA-Zertifikatskette des LDAP-Servers wird in einzelnen PEM Dateien mit `.pem` Dateiendung nach
 - `/usr/local/etc/raddb/certs/trustedCAs-ldap-backend/`
und
 - `/usr/local/etc/openldap/certs/trustedCAs-ldap-server/`
kopiert
- Dann wird jeweils in diesen Verzeichnissen `c_rehash ./` aufgerufen

- aktuelle CRLs per Cronjob und Poll-Skript
 - Beispielskript `fetch-crl` von <http://dist.eugridpma.info/distribution/util/fetch-crl/>

- Erstellen von `<hash>.crl_url` Konfigdateien in den Verzeichnissen
 - `/usr/local/etc/raddb/certs/trustedCAs-ldap-backend/` (LDAP Server)
 - `/usr/local/etc/raddb/certs/trustedCAsForRoamingClients/` (EAP-TLS, Client-AuthN)
 - `/usr/local/etc/openldap/certs/trustedCAs-ldap-server/` (LDAP Server)
- Aufruf von `fetch-crl` & Re-Start des `radiusd` per cron

Sichere Kommunikation
mit einem LDAP-Verzeichnisdienst
OpenLDAP als einfache Passwort-
Datenbank für FreeRADIUS
www.openldap.org

OpenLDAP

- Lightweight Directory Access Protocol Verzeichnisdienst
- OpenLDAP hat SSL/TLS-Unterstützung durch Cyrus-SASL Framework
- Absicherung der Abfragen
- Client-Zertifikate für Benutzer Authentifizierung möglich
- Serverzertifikat mit **serverAuth** und **clientAuth** als erweiterter Schlüssel-Verwendungszweck
- Profil **LDAP Server** in der DFN-PKI

- OpenLDAP Konfigurationsverzeichnis in und unterhalb von `/usr/local/etc/openldap`
- LDAP Schemata liegen unter `/usr/local/etc/openldap/schema`
 - Mitgelieferte Schemata sind ausreichend für die Nutzung als FreeRADIUS Passwort Datenbank

- Verzeichnisse für vertrauenswürdige CAs und deren CRLs, bleiben je nach Fall ggf. leer(!)
 - `/usr/local/etc/openldap/certs/trustedLDAPclientCAs/`
 - `/usr/local/etc/openldap/certs/trustedCAs-ldap-server/`

OpenLDAP Konfig-Dateien (2)

- Datei mit geheimen Schlüssel des LDAP-Servers
`/usr/local/etc/openldap/certs/ldap-domain-de-key.pem`
- Datei mit dem dazu passenden X.509 Zertifikat und der zugehörigen CA-Kette
`/usr/local/etc/openldap/certs/ldap-domain-de-cert-and-chain.pem`
- Datei mit Diffie-Hellman Parametern
`/usr/local/etc/openldap/certs/dh` erzeugt mittels `openssl dhparam -out /usr/local/etc/openldap/certs/dh 1024`

Konfig-Datei - slapd.conf (1)

```
# /usr/local/etc/openldap/slapd.conf

# alles LDAP-Server-Events loggen
loglevel any

# lesbar durch ldap Gruppe
include /usr/local/etc/openldap/schema/core.schema
include /usr/local/etc/openldap/schema/cosine.schema
include /usr/local/etc/openldap/schema/inetorgperson.schema

pidfile /usr/local/var/run/slapd.pid
argsfile /usr/local/var/run/slapd.args
```

Konfig-Datei - slapd.conf (2)

```
# erlaubte Krypto-Cipher, absteigend nach Staerke sortiert
TLSCipherSuite \
    @STRENGTH:HIGH:MEDIUM:!LOW:!EXP:!aNULL:!eNULL:!MD5:!ADH

TLSRandFile /dev/urandom

# Parameter fuer Diffie-Hellmann Krypto
TLSDHParamFile /usr/local/etc/openldap/certs/dh
```

Konfig-Datei - slapd.conf (3)

```
# LDAP-Server-Zertifikat
TLSCertificateFile \
    /usr/local/etc/openldap/certs/ldap-domain-de-cert-and-chain.pem
# Schluessel zum LDAP Server-Zertifikat
TLSCertificateKeyFile \
    /usr/local/etc/openldap/certs/ldap-domain-de-key.pem

# vertraenswuerdige CAs fuer LDAP Client-Auth-Zertifikate und deren
# CRLs (fetch-crl, c_rehash wie im FreeRADIUS Beispiel)
TLSCACertificatePath \
    /usr/local/etc/openldap/certs/trustedLDAPclientCAs/

# TLSVerifyClient Optionen: never, allow, try, demand
TLSVerifyClient demand

# TLSCRLCheck Optionen: none, peer, all
TLSCRLCheck all
```

Konfig-Datei - slapd.conf (4)

```
# Setzen des SecurityStrengthFactor auf mindestens 128 bit,  
# da ansonsten Passwoerter im Klartext ueber unverschlusselte  
# Verbindungen gehen koennten  
security ssf=128  
  
# Standard default deny access Policy auf alle LDAP DITs  
access to dn.base=""  
    by * none  
  
access to *  
    by none  
  
sizelimit 10
```

Konfig-Datei - slapd.conf (5)

```
# vor dem Starten des slapd immer ein db_recover im  
# DB Verzeichnis fahren; bei SuSE gibt es die  
# Konfigurations-Option OPENLDAP_RUN_DB_RECOVER in  
# /etc/sysconfig/openldap  
database bdb  
  
suffix "dc=domain,dc=de"  
rootdn "cn=ldapadmin,dc=domain,dc=de"  
  
# seeded sha1 hash von 'secret' - erzeugt mittels  
# 'slappasswd -s 'secret' -h '{SSHA}''  
rootpw          {SSHA}VYiFoo0lbaoY7ppvsaoAjstKPF+SshE2  
  
# Lese- und Schreibrechte für Gruppe ldap  
directory       /usr/local/openldap/data/dfnroaming-userdb  
index           objectClass eq  
  
# wir greifen in diesem Bsp. immer als DB-Eigner zu  
access to *  
    by * none
```

- OpenLDAP Konfigurationsdatei
 - `-f /usr/local/etc/openldap/slapd.conf`
- listen auf allen den entsprechenden IP-Adressen auf den Ports 389 & 636 (SSL)
 - `-h 'ldap://ldap.domain.de
ldaps://ldap.domain.de'`
- User und Gruppe festlegen
 - `-u ldap -g ldap`
- bei SuSE entsprechende Optionen in der Datei `/etc/sysconfig/openldap`

```
# /usr/local/etc/openldap/ldap.conf
loglevel -1 # alle LDAP-Client-Events loggen
TLS_RANDFILE /dev/urandom
# erlaubte Krypto-Cipher, absteigend nach Staerke sortiert
TLS_CIPHER_SUITE \
  @STRENGTH:HIGH:MEDIUM:!LOW:!EXP:!aNULL:!eNULL:!MD5:!ADH
# LDAP Server muss ein gueltiges Server-Zertifikat haben
TLS_REQCERT demand
# vertraenswuerdige CAs fuer LDAP-Server-Zertifikate und deren
# CRLs (fetch-crl, c_rehash wie im FreeRADIUS Beispiel)
TLS_CACERTDIR \
  /usr/local/etc/openldap/certs/trustedCAs-ldap-server/
# LDAP Server-Zertifikate in CRLs pruefen
TLS_CRLCHECK all
```

```
# LDAP Client Zertifikat-Konfiguration fuer ldap(search|add|modify)
#TLS_CERT \  
# /usr/local/etc/openldap/certs/ldap-machine-client-cert-and-chain.pem  
  
# und der entsprechende Schluessel zum Client-Zertifikat  
#TLS_KEY /usr/local/etc/openldap/certs/ldap-machine-client-key.pem
```

ldapadd & ldapmodify

- befüllen des LDAP Servers mit LDIF Daten
 - `ldapadd -v -c -x -W \
-H ldap://ldap.domain.de \
-D 'cn=ldapadmin,dc=domain,dc=de' \
-f ldapdata-db-init.ldif \
-S ldapdata-db-ini-err.ldif`
- ändern von LDAP Einträgen mit LDIF Daten
 - `ldapmodify -v -c -x -W \
-H ldap://ldap.domain.de \
-D 'cn=ldapadmin,dc=domain,dc=de' \
-f ldapdata-mod.ldif \
-S ldapdata-mod-err.ldif`

LDIF Daten-Beispiel für roamenden Nutzer (1)

```
# LDAP Baum (DIT) Struktur anlegen

dn: dc=domain,dc=de
objectClass: domain
dc: domain

dn: c=DE,dc=domain,dc=de
objectClass: country
c: DE

dn: o=Einrichtung ABC,c=DE,dc=domain,dc=de
objectClass: organization
o: Einrichtung ABC

dn: ou=Team XYZ,o=Einrichtung ABC,c=DE,dc=domain,dc=de
objectClass: organizationalUnit
ou: Team XYZ
```

LDIF Daten-Beispiel für roamenden Nutzer (2)

```
# Nutzer Daten fuer John Doe

dn: cn=John Doe Test,ou=Team XYZ,o=Einrichtung
ABC,c=DE,dc=domain,dc=de

objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson

cn: John Doe
sn: Doe

uid: jdoe@domain.de

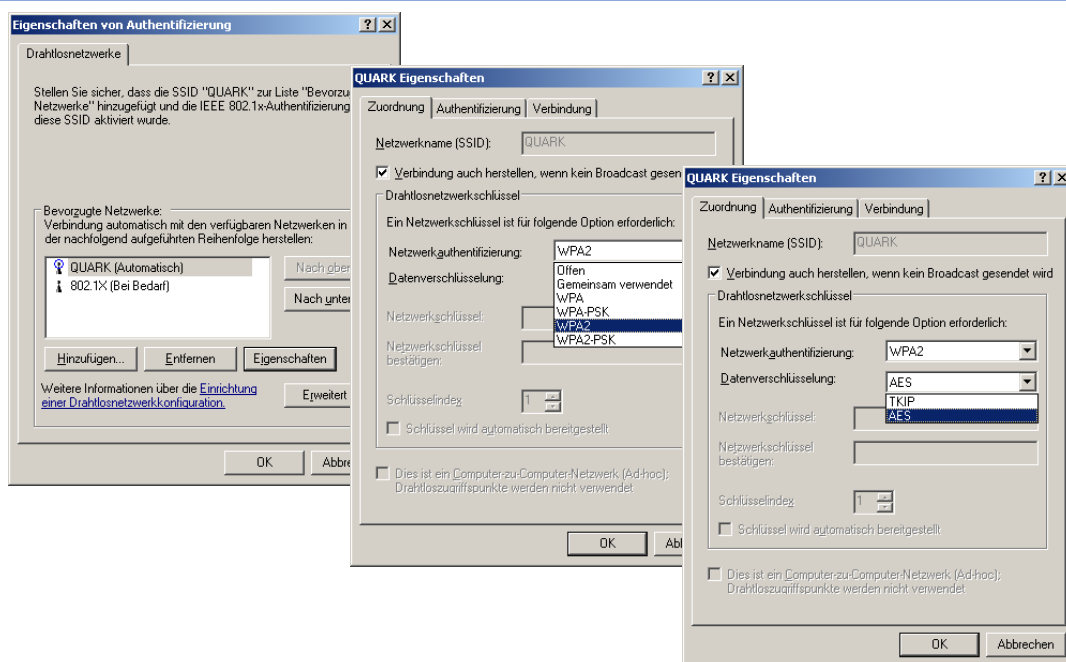
# Klartext Passwort des Nutzers John Doe
userPassword: {clear}passwort-geheimnis
```

- abfragen des LDAP Servers (cn und uid Attribut)
 - ```
ldapsearch -v -x -W \
-H ldap://ldap.domain.de \
-D 'cn=ldapadmin,dc=domain,dc=de' \
-b 'dc=domain,dc=de' \
'(objectclass=*)' \
cn uid
```
- ausführliches Debugging mit Option "-d -1"
- Option "-zz" für SSL mit STARTTLS
- Option "-H ldaps://" für LDAP SSL auf Port 636

## WLAN Konfiguration von Windows XP SP2 mit WPA2 - EAP

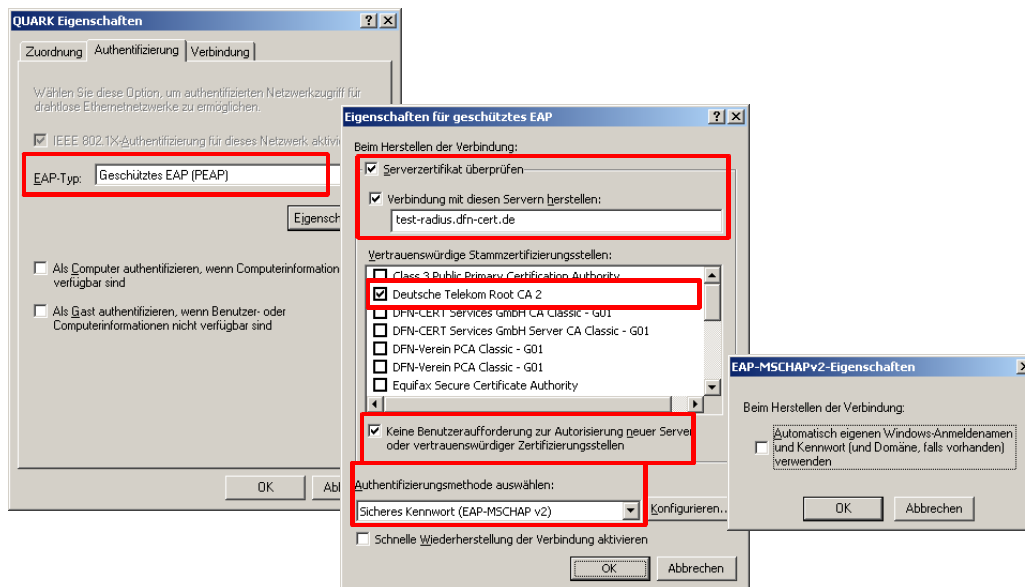
- Windows XP SP2
- Hotfix 917021 (Wireless Client Update)
  - <http://support.microsoft.com/kb/917021>
- Hotfix 893357 (WPA2 Update)
  - <http://support.microsoft.com/kb/893357>
- Hotfix 885453 (nur bei Problemen mit PEAP Authentisierung am RADIUS Server)
  - <http://support.microsoft.com/kb/885453>
  - Auf Anfrage von MS oder z.B. unter <http://www.fh-fulda.de/index.php?id=3972>

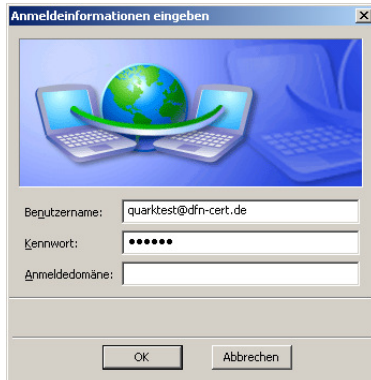
## Windows XP Supplikant – Konfig (1) WLAN – AP Zuordnung



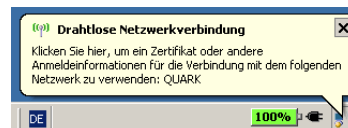
# Konfiguration des Supplikanten Windows XP für PEAP (Windows build-in)

## Windows XP Supplikant – Konfig AuthN – PEAP – MS-CHAPv2



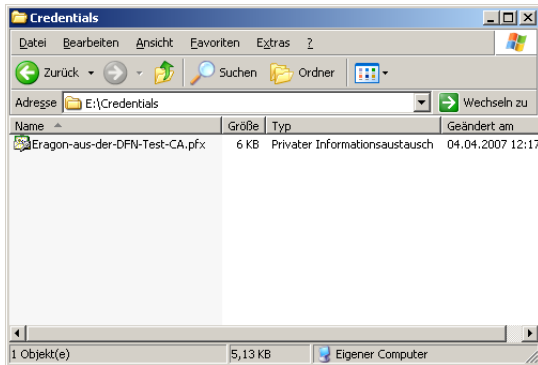


Beim Verbindungsaufbau fragt Windows zunächst über eine Blase in der Task-Leiste nach dem Nutzernamen (äußere RADIUS-Identität ist gleich der inneren RADIUS-Identität) und Passwort und baut dann die Verbindung auf.



## Konfiguration des Supplikanten Windows XP für EAP-TLS (Windows build-in)

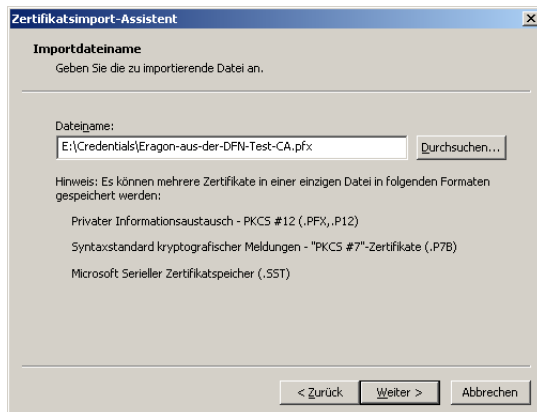
# Import von Nutzerzertifikaten (1)



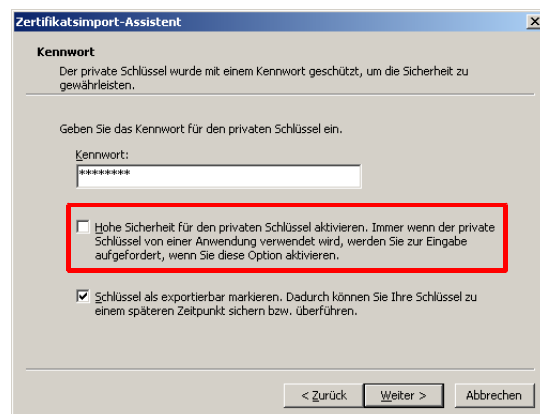
Doppelklick auf die PKCS#12 Datei startet den Import-Assistenten



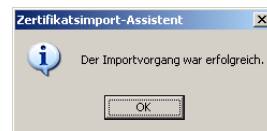
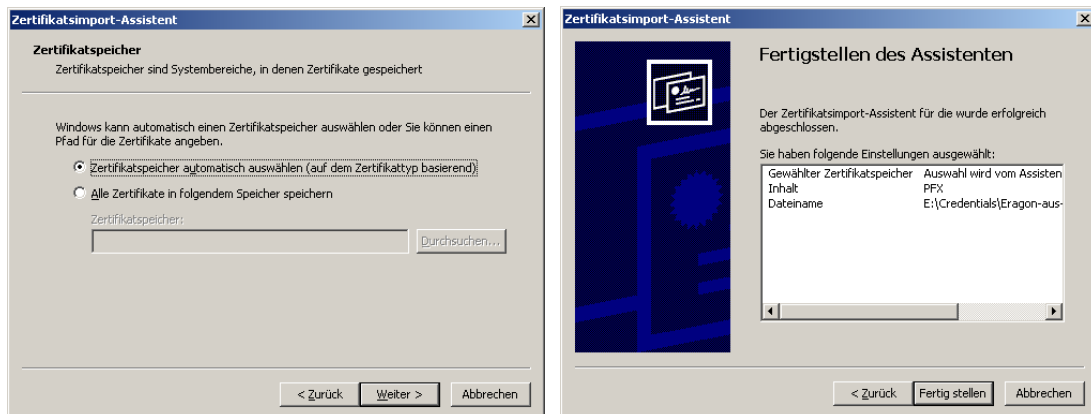
# Import von Nutzerzertifikaten (2) Import-Assistent



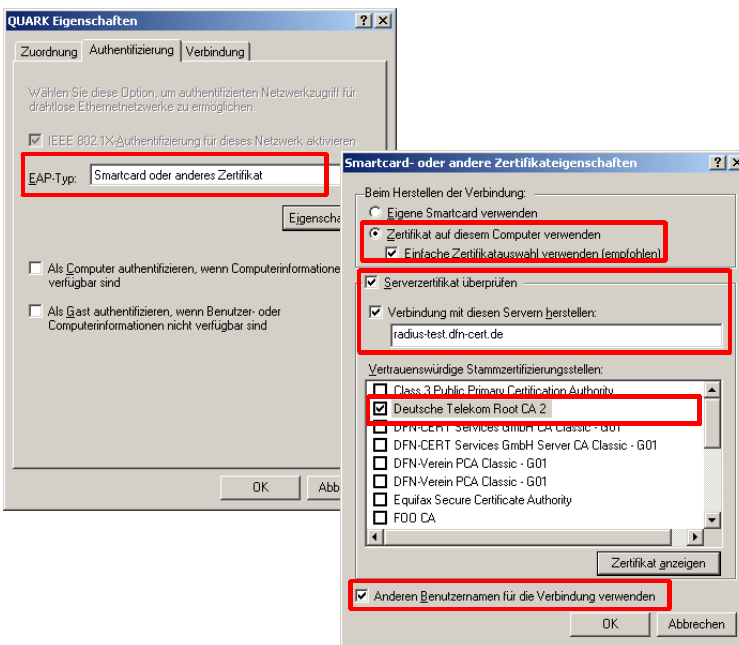
1. Passwort, mit dem die PKCS#12 Datei geschützt ist
2. Keine hohe Sicherheitsstufe, da das Zertifikat sonst nicht zur Nutzung für EAP-TLS ausgewählt werden kann



# Import von Nutzerzertifikaten (3) Import-Assistent



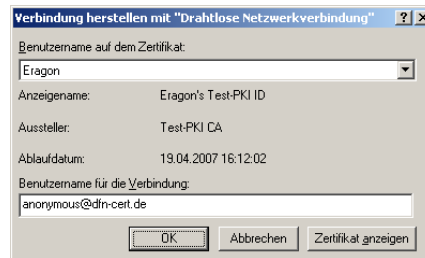
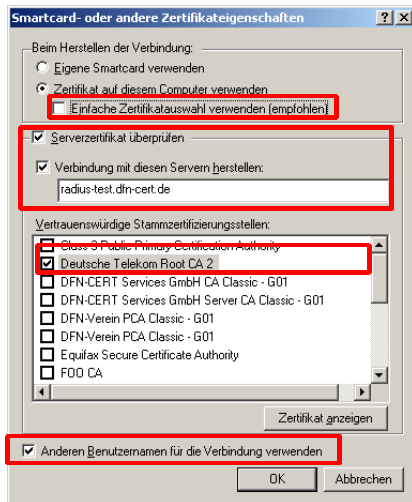
# Windows XP Supplikant – Konfig (1) AuthN Protokoll – EAP-TLS -Zertifikat



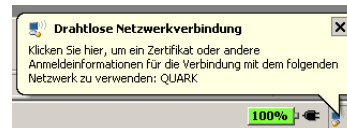
Windows versucht beim Verbindungsaufbau an Hand der vorhandenen Nutzerzertifikate automatisch das richtige auszuwählen. Zertifikate mit unpassenden extendedKeyUsages (z.B. MS Smart Card Logon oder fehlendem clientAuth) werden hierbei ignoriert.

Im Idealfall keine Nachfragen beim Nutzer, ggf. Frage nach äußerer RADIUS-Identität, z.B. *anonymous@domain.de*

# Windows XP Supplikant – Konfig (2) Zertifikat – erweiterte Zertifikatauswahl



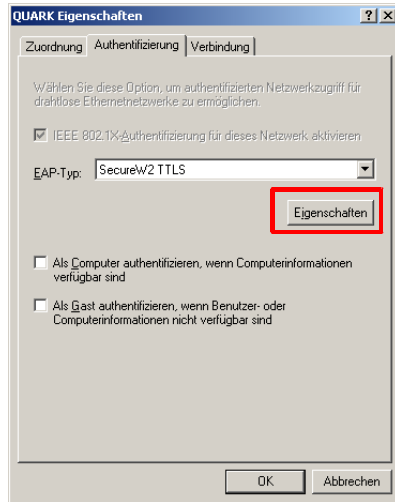
Beim Verbindungsaufbau fragt Windows zunächst über eine Blase in der Task-Leiste nach dem zu nutzenden Nutzerzertifikat und nach der äußeren RADIUS-Identität. Zertifikate mit unpassenden extendedKeyUsages (z.B. MS Smart Card Logon oder fehlendem clientAuth) werden hierbei ignoriert.



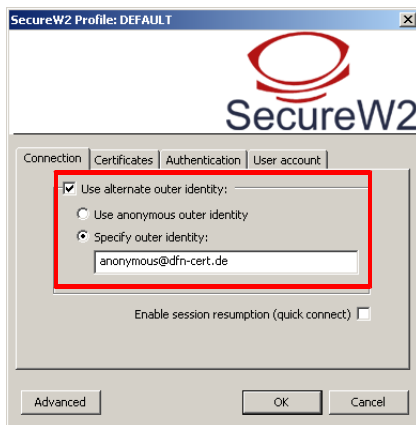
## Konfiguration des Supplikanten Windows XP für EAP-TTLS (SecureW2)

[www.securew2.org](http://www.securew2.org)

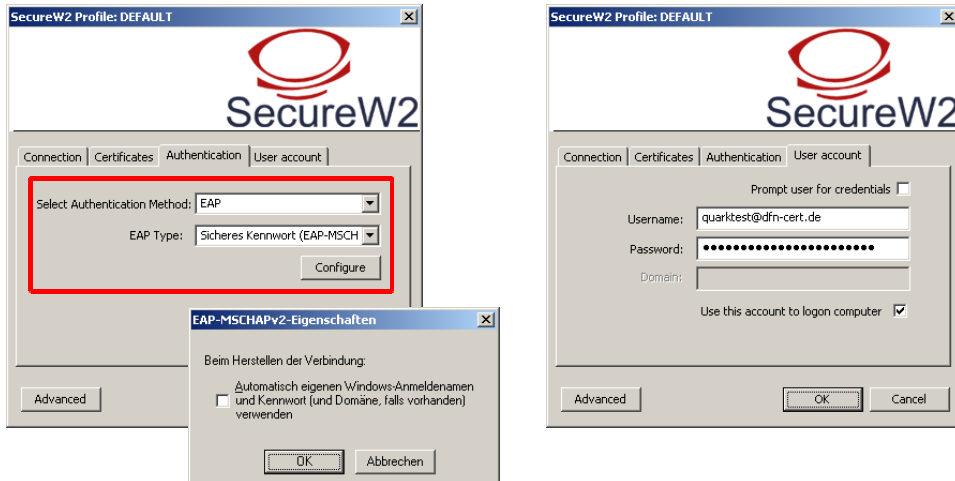
# SecureW2 Supplikant – Konfig (1) EAP-TTLS via SecureW2 - Konfiguration



# SecureW2 Supplikant – Konfig (2) Verbindung - RADIUS-Server-Zertifikate

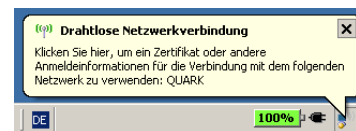
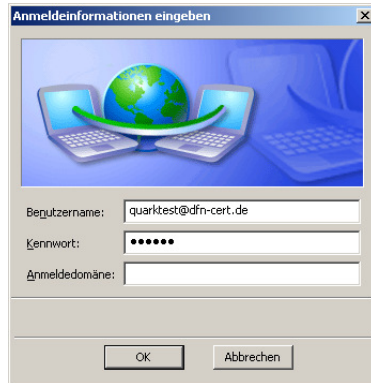


## SecureW2 Supplikant – Konfig (3) Authentisierung - Nutzerkonto



## SecureW2 Supplikant – Konfig (4) Erweiterte Optionen





Fragen?

Vielen Dank für Ihre Aufmerksamkeit!

[dfnpca@dfn-cert.de](mailto:dfnpca@dfn-cert.de)