

Freeradius telepítése RHEL/Almalinux 9 (x86_64) rendszeren

Előkészítés

Kiindulásként AlmaLinux 9 (x86_64) operációs rendszer telepítését kell elvégezni. A telepítés minimal telepítés.

Telepítést követő lépések

Módosítani kell a /etc/dnf/dnf.conf állományt a felesleges gyenge függőségek telepítésének tiltásához:

```
# if [ ! -f /etc/dnf/dnf.conf.orig ] ; then cp -a /etc/dnf/dnf.conf
/etc/dnf/dnf.conf.orig && echo 'install_weak_deps=False' >>
/etc/dnf/dnf.conf ; fi
```

Érdemes kikapcsolni a kernel üzenetek megjelenítését

```
# grubby --update-kernel=ALL --args=quiet
```

A SELinux-ot átmenetileg megengedő módba kell kapcsolni

```
# sed -i 's/^SELINUX=.*SELINUX=permissive/' /etc/selinux/config
```

Be kell állítani a rendszer locale-t en_US.UTF-8-ra

```
# localectl
```

Listázzuk az elérhető csomagokat

```
# dnf list freeradius*
Last metadata expiration check: 1:14:21 ago on Mon 08 Sep 2025 07:24:49 PM
CEST.
Available Packages
freeradius.x86_64
3.0.21-44.el9_6
appstream
freeradius-devel.x86_64
3.0.21-44.el9_6
appstream
freeradius-doc.x86_64
3.0.21-44.el9_6
appstream
freeradius-krb5.x86_64
```

```
3.0.21-44.el9_6
appstream
freeradius-ldap.x86_64
3.0.21-44.el9_6
appstream
freeradius-utils.x86_64
3.0.21-44.el9_6
appstream
```

Telepítsük a szükséges csomagokat

```
# dnf install freeradius freeradius-utils
```

Futtassuk a radiusd programot debug módban ahol az alábbi hibát kapiuk

```
tls-config tls-common {
    verify_depth = 0
    ca_path = "/etc/raddb/certs"
    pem_file_type = yes
    private_key_file = "/etc/raddb/certs/server.pem"
}
Unable to check file "/etc/raddb/certs/server.pem": No such file or
directory
/etc/raddb/mods-enabled/eap[183]: Failed parsing configuration item
"private_key_file"
rlm_eap_tls: Failed initializing SSL context
rlm_eap (EAP): Failed to initialise rlm_eap_tls
/etc/raddb/mods-enabled/eap[14]: Instantiation failed for module "eap"
```

Inicializáljuk a tanúsítványokat

```
# cd /etc/raddb/certs

# make ca.pem
make[1]: Entering directory '/etc/raddb/certs'
make[1]: Leaving directory '/etc/raddb/certs'
make[1]: Entering directory '/etc/raddb/certs'
make[1]: Leaving directory '/etc/raddb/certs'
openssl req -new -x509 -keyout ca.key -out ca.pem \
    -days '60' -config ./ca.cnf \
    -passin pass:'whatever' -passout pass:'whatever' -noenc
.+.....+...+.....+..++++++*..+...+...+
.....+...+...+.....+.+..++++++*..+...+...
.++++++
...+.....+.....+.....+.+...+.....++++++
++++++*.....+.+.....+.+.....++++++
++++++*.....+.+.....+.+.....+.+.....+.+.....
...+..+...+...+.....+.+.....+.+.....+.+.....+.
.....+.+.....+.+.....+.+.....+.+.....+.+.....
.+...+.+.....+.+.....+.+.....+.+.....+.+.....
.+.....+.+.....+.+.....++++++
-----
```

[illegible]

```
chmod g+r server.p12
openssl pkcs12 -in server.p12 -out server.pem -passin pass:'whatever' -
passout pass:'whatever'
chmod g+r server.pem
chown root:radiusd server.*
chmod 640 server.*
```

Amennyiben szeretnénk az eredeti állaporta állni, akkor az alábbi utasításokat futtassuk

```
# cd /etc/raddb/certs && rm -f *.pem *.der *.csr *.crt *.key *.p12 serial*
index.txt*
```

A szerver tesztelését újra futtathatjuk

```
Listening on auth address 127.0.0.1 port 18120 bound to server inner-tunnel
Listening on auth address * port 1812 bound to server default
Listening on acct address * port 1813 bound to server default
Listening on auth address :: port 1812 bound to server default
Listening on acct address :: port 1813 bound to server default
Listening on proxy address * port 44473
Listening on proxy address :: port 54094
Ready to process requests
```

File alapú hitelesítés

Mentsük le a felhasználókat tartalmazó adatbázist

```
# cp -a /etc/raddb/mods-config/files/authorize /etc/raddb/mods-
config/files/authorize.orig
```

Hozzunk létre teszt felhasználót

```
# cat > /etc/raddb/mods-config/files/authorize <<EOF
testing Cleartext-Password := "password"
EOF
```

Újraindítás nélkül teszteljük a hozzáférést

```
# radtest testing password 127.0.0.1 0 testing123
Sent Access-Request Id 146 from 0.0.0.0:59541 to 127.0.0.1:1812 length 77
  User-Name = "testing"
  User-Password = "password"
  NAS-IP-Address = 192.168.110.11
  NAS-Port = 0
  Cleartext-Password = "password"
Received Access-Reject Id 146 from 127.0.0.1:1812 to 127.0.0.1:59541 length
38
  Message-Authenticator = 0x6fdf89ab2cd775d9cbecc980edf715f3
(0) -: Expected Access-Accept got Access-Reject
```

Töltsük újra a szerver konfigurációt

```
# systemctl restart radiusd
```

Teszteljük a módosítást

```
# radtest testing password 127.0.0.1 0 testing123
Sent Access-Request Id 161 from 0.0.0.0:44720 to 127.0.0.1:1812 length 77
  User-Name = "testing"
  User-Password = "password"
  NAS-IP-Address = 192.168.110.11
  NAS-Port = 0
  Cleartext-Password = "password"
Received Access-Accept Id 161 from 127.0.0.1:1812 to 127.0.0.1:44720 length
38
  Message-Authenticator = 0x88d891160ee4e45002497ff6e3ca4b59
```

Teljes teszt

```
# radtest --help
Usage: radtest [OPTIONS] user passwd radius-server[:port] nas-port-number
secret [ppphint] [nasname]
  -d RADIUS_DIR          Set radius directory
  -t <type>              Set authentication method
                        type can be pap, chap, mschap, or eap-md5
  -P protocol            Select udp (default) or tcp
  -x                    Enable debug output
  -4                    Use IPv4 for the NAS address (default)
  -6                    Use IPv6 for the NAS address
  -6                    Mandate checks for Blast RADIUS (this is not set
by default).

# radtest -t pap testing password 127.0.0.1 0 testing123
Sent Access-Request Id 146 from 0.0.0.0:59648 to 127.0.0.1:1812 length 77
  User-Name = "testing"
  User-Password = "password"
  NAS-IP-Address = 192.168.110.11
  NAS-Port = 0
  Cleartext-Password = "password"
Received Access-Accept Id 146 from 127.0.0.1:1812 to 127.0.0.1:59648 length
38
  Message-Authenticator = 0x29c262c37719c7b5490dade0eba018a2

# radtest -t chap testing password 127.0.0.1 0 testing123
Sent Access-Request Id 46 from 0.0.0.0:46131 to 127.0.0.1:1812 length 78
  User-Name = "testing"
  CHAP-Password = 0xcaf4ca1ab5cbd80cfd255c0a20b5d0d7a3
  NAS-IP-Address = 192.168.110.11
  NAS-Port = 0
  Cleartext-Password = "password"
Received Access-Accept Id 46 from 127.0.0.1:1812 to 127.0.0.1:46131 length
```



```
EAP-Type-MD5-Challenge = 0x1070cf2ce2eb2e0cf2b1283d43583ae23b
EAP-Id = 245
State = 0x1d2d846a1dd8802f3111415ec636f0a3
EAP-Message = 0x02f50016041070cf2ce2eb2e0cf2b1283d43583ae23b
Received Access-Accept Id 38 from 127.0.0.1:1812 to 0.0.0.0:57235 length 53
Message-Authenticator = 0x8d94e8b9207f6f19bc078121280292f6
EAP-Message = 0x03f50004
User-Name = "testing"
EAP-Id = 245
EAP-Code = Success
```

LDAP hitelesítés

Szükséges csomagok telepítése

```
# dnf install freeradius freeradius-utils freeradius-ldap
```

Létre kell hozni az ldap szimbolikus linket

```
# ln -s ../mods-available/ldap /etc/raddb/mods-enabled/ldap
```

Menteni kell a /etc/raddb/mods-available/ldap állományt majd módosítani a tartalmát

```
# [ ! -e /etc/raddb/mods-available/ldap.orig ] && cp -a /etc/raddb/mods-
available/ldap /etc/raddb/mods-available/ldap.orig

cat > /etc/raddb/mods-available/ldap <<'EOF'
ldap {
    server = 'ldaps://dc1.adomain.lan'
    identity = 'cn=radiusbind,cn=users,dc=adomain,dc=lan'
    password = 12345678
    base_dn = 'dc=adomain,dc=lan'

    update {
        control:Password-With-Header += 'userPassword'
        control: += 'radiusControlAttribute'
        request: += 'radiusRequestAttribute'
        reply: += 'radiusReplyAttribute'
    }

    user {
        base_dn = "${..base_dn}"
        filter = "(sAMAccountName=%{%{Stripped-User-Name}:-{%{User-Name}}})"
    }

    group {
        base_dn = "${..base_dn}"
        filter = '(objectClass=posixGroup)'
        membership_attribute = 'memberOf'
    }
}
```

```
}

client {
    base_dn = "${..base_dn}"
    filter = '(objectClass=radiusClient)'
    attribute {
        ipaddr = 'radiusClientIdentifier'
        secret = 'radiusClientSecret'
    }
}

accounting {
    reference = "%{tolower:type.%{Acct-Status-Type}}"

    type {
        start {
            update {
                description := "Online at %S"
            }
        }

        interim-update {
            update {
                description := "Last seen at %S"
            }
        }

        stop {
            update {
                description := "Offline at %S"
            }
        }
    }
}

post-auth {
    update {
        description := "Authenticated at %S"
    }
}

options {
    chase_referrals = yes
    rebind = yes
    res_timeout = 10
    srv_timelimit = 3
    net_timeout = 1
    idle = 60
    probes = 3
    interval = 3
    ldap_debug = 0x0028
}
```

```
}

tls {
    start_tls = no
    require_cert = 'allow'
}

pool {
    start = ${thread[pool].start_servers}
    min = ${thread[pool].min_spare_servers}
    max = ${thread[pool].max_servers}
    spare = ${thread[pool].max_spare_servers}
    uses = 0
    retry_delay = 30
    lifetime = 0
    idle_timeout = 60
}
}
EOF
</core>
```

Menteni kell a /etc/raddb/sites-available/default állományt majd módosítani a tartalmát

<code>

```
# [ ! -e /etc/raddb/sites-available/default.orig ] && cp -a
/etc/raddb/sites-available/default /etc/raddb/sites-available/default.orig
```

```
cat > /etc/raddb/sites-available/default <<'EOF'
```

```
server default {
    listen {
        type = auth
        ipaddr = *
        port = 0
        limit {
            max_connections = 16
            lifetime = 0
            idle_timeout = 30
        }
    }

    listen {
        ipaddr = *
        port = 0
        type = acct
        limit {
        }
    }

    listen {
        type = auth
```

```
    ipv6addr = ::      # any.  ::1 == localhost
    port = 0
    limit {
        max_connections = 16
        lifetime = 0
        idle_timeout = 30
    }
}

listen {
    ipv6addr = ::
    port = 0
    type = acct
    limit {
    }
}

authorize {
    filter_username
    preprocess
    chap
    mschap
    digest
    suffix

    eap {
        ok = return
    }

    files
    -sql

    ldap

    if ((ok || updated) && User-Password && !control:Auth-Type) {
        update {
            control:Auth-Type := ldap
        }
    }

    expiration
    logintime
    pap
}

authenticate {
    Auth-Type PAP {
        pap
    }

    Auth-Type CHAP {
```

```
        chap
    }

    Auth-Type MS-CHAP {
        mschap
    }

    mschap

    digest

    Auth-Type LDAP {
        ldap
    }

    eap
}

preacct {
    preprocess
    acct_unique
    suffix
    files
}

accounting {
    detail
    unix
    -sql
    exec
    attr_filter.accounting_response
}

session {
}

post-auth {
    if (session-state:User-Name && reply:User-Name && request:User-Name
&& (reply:User-Name == request:User-Name)) {
        update reply {
            &User-Name !* ANY
        }
    }

    update {
        &reply: += &session-state:
    }

    -sql
    exec
    remove_reply_message_if_eap
}
```

```
    Post-Auth-Type REJECT {
        -sql
        attr_filter.access_reject
        eap
        remove_reply_message_if_eap
    }

    Post-Auth-Type Challenge {
    }
}

pre-proxy {
}

post-proxy {
    eap
}
}
EOF
```

Menteni kell a `/etc/raddb/sites-available/inner-tunnel` állományt majd módosítani a tartalmát

```
# [ ! -e /etc/raddb/sites-available/inner-tunnel.orig ] && cp -a
/etc/raddb/sites-available/inner-tunnel /etc/raddb/sites-available/inner-
tunnel.orig

cat > /etc/raddb/sites-available/inner-tunnel <<'EOF'
server inner-tunnel {
    listen {
        ipaddr = 127.0.0.1
        port = 18120
        type = auth
    }

    authorize {
        filter_username
        chap
        mschap
        suffix
        update control {
            &Proxy-To-Realm := LOCAL
        }
        eap {
            ok = return
        }

        files

        -sql

        ldap
    }
}
```

```
        if ((ok || updated) && User-Password && !control:Auth-Type) {
            update {
                control:Auth-Type := ldap
            }
        }

        expiration
        logintime

        pap
    }

    authenticate {
        Auth-Type PAP {
            pap
        }

        Auth-Type CHAP {
            chap
        }

        Auth-Type MS-CHAP {
            mschap
        }

        mschap

        Auth-Type LDAP {
            ldap
        }

        eap
    }

    session {
        radutmp
    }

    post-auth {
        -sql
        if (0) {
            update reply {
                User-Name !* ANY
                Message-Authenticator !* ANY
                EAP-Message !* ANY
                Proxy-State !* ANY
                MS-MPPE-Encryption-Types !* ANY
                MS-MPPE-Encryption-Policy !* ANY
                MS-MPPE-Send-Key !* ANY
                MS-MPPE-Recv-Key !* ANY
            }
        }
    }
}
```

```
    }

    update {
        &outer.session-state: += &reply:
    }
}

Post-Auth-Type REJECT {
    -sql
    attr_filter.access_reject
    update outer.session-state {
        &Module-Failure-Message := &request:Module-Failure-Message
    }
}

pre-proxy {
}

post-proxy {
    eap
}
}
EOF
```

Újra kell indítani a radiusd szolgáltatást

```
# systemctl restart radiusd
```

Létre kell hozni a DC-n a radiusbind felhasználót akinek a jelszava nem jár le

```
# samba-tool user create radiusbind 12345678
```

```
# samba-tool user setexpiry radiusbind --noexpiry
```

Tesztelhető a szolgáltatás

```
# radtest teszt.elek 12345678 127.0.0.1 0 testing123
Sent Access-Request Id 158 from 0.0.0.0:39864 to 127.0.0.1:1812 length 80
  User-Name = "teszt.elek"
  User-Password = "12345678"
  NAS-IP-Address = 192.168.110.11
  NAS-Port = 0
  Cleartext-Password = "12345678"
Received Access-Accept Id 158 from 127.0.0.1:1812 to 127.0.0.1:39864 length 38
  Message-Authenticator = 0xf111666b349fe1dcc5ea191026805f2d
```

Kerberos hitelesítés

Szükséges csomagok telepítése

```
# dnf install freeradius freeradius-utils freeradius-krb5 krb5-workstation
```

Hozzuk létre egy felhasználót a DC.n

```
# samba-tool user add radius-svc --random-password
User 'radius-svc' added successfully
```

Hozzuk létre SPN-t a radius szervernek

```
# samba-tool spn add radius/dc1.adomain.lan radius-svc
```

Ellenőrizzük az SPN-t

```
# ldbsearch -H ldap://localhost -U administrator
'(servicePrincipalName=radius/dc1.adomain.lan)' sAMAccountName
servicePrincipalName
Password for [ADOMAIN\administrator]:
# record 1
dn: CN=radius-svc,CN=Users,DC=adomain,DC=lan
sAMAccountName: radius-svc
servicePrincipalName: radius/dc1.adomain.lan

# Referral
ref: ldap://adomain.lan/CN=Configuration,DC=adomain,DC=lan

# Referral
ref: ldap://adomain.lan/DC=DomainDnsZones,DC=adomain,DC=lan

# Referral
ref: ldap://adomain.lan/DC=ForestDnsZones,DC=adomain,DC=lan

# returned 4 records
# 1 entries
# 3 referrals
```

Ez a lépés nem kötelező . Beállíthatjuk a titkosítást a felhasználói fiók esetében.

```
# cat > ~/encryption-mod.ldif <<'EOF'
dn: CN=radius-svc,CN=Users,DC=adomain,DC=lan
changetype: modify
replace: msDS-SupportedEncryptionTypes
msDS-SupportedEncryptionTypes: 24
EOF

# ldbmodify -H ldap://localhost -U administrator ~/encryption-mod.ldif
Password for [ADOMAIN\administrator]:
```

Modified 1 records successfully

Exportáljuk a keytab-ot

```
# samba-tool domain exportkeytab /root/radius.keytab --  
principal=radius/dc1.adomain.lan  
Export one principal to /root/radius.keytab
```

Ellenőrizzük a keytab tartalmát

```
# klist -k -e /root/radius.keytab  
Keytab name: FILE:/root/radius.keytab  
KVNO Principal  
-----  
---  
    2 radius/dc1.adomain.lan@ADOMAIN.LAN (aes256-cts-hmac-sha1-96)  
    2 radius/dc1.adomain.lan@ADOMAIN.LAN (aes128-cts-hmac-sha1-96)
```

Másoljuk be a keytabot a megfelelő helyre és állítsuk be a jogosultságot

```
# cp -a /root/radius.keytab /var/lib/radiusd/keytab  
  
# chown radiusd:radiusd /var/lib/radiusd/keytab  
  
# chmod 0600 /var/lib/radiusd/keytab
```

Mentsük le az eredeti kerberos konfigurációt

```
# [ ! -f /etc/raddb/mods-available/krb5.orig ] && cp -a /etc/raddb/mods-  
available/krb5 /etc/raddb/mods-available/krb5.orig
```

Módosítsuk a konfigurációt

```
# cat > /etc/raddb/mods-available/krb5 <<'EOF'  
krb5 {  
    keytab = ${localstatedir}/lib/radiusd/keytab  
    service_principal = radius/dc1.adomain.lan  
    pool {  
        start = ${thread[pool].start_servers}  
        min = ${thread[pool].min_spare_servers}  
        max = ${thread[pool].max_servers}  
        spare = ${thread[pool].max_spare_servers}  
        uses = 0  
        lifetime = 0  
        idle_timeout = 0  
    }  
}  
EOF
```

Engedélyezzük a konfigurációt

```
# ln -s ../mods-available/krb5 /etc/raddb/mods-enabled/krb5
```

Indítsuk újra a szolgáltatást

```
# systemctl restart radiusd
```

Teszteljük a kapcsolatot

```
# radtest -t pap teszt.elek '12345678' localhost 0 testing123
Sent Access-Request Id 176 from 0.0.0.0:36009 to 127.0.0.1:1812 length 80
  User-Name = "teszt.elek"
  User-Password = "12345678"
  NAS-IP-Address = 192.168.110.11
  NAS-Port = 0
  Cleartext-Password = "12345678"
Received Access-Accept Id 176 from 127.0.0.1:1812 to 127.0.0.1:36009 length
38
  Message-Authenticator = 0x662c8de328ec1703342d74ece4225877
```

From:

<http://wiki.r-l.hu/> - **Reverse-Logic wiki**

Permanent link:

<http://wiki.r-l.hu/doku.php?id=linux:freeradius>

Last update: **2025/09/10 11:05**

