Zentyal 3.5.5 Community edition (small bussiness server)

🔧 Fix Me!

Přizpůsobení konfiguračních souborů

Přes zentyal se dá pár věcí naklikat přes webové rozhraní a následně je vygenerován konfigurační soubor pro příslušnou službu. Dříve nebo později ovšem zjistíme, že přes webové rozhraní nelze naklikat vše a proto je nutné upravit šablony, ze kterých se konfigurační soubor generuje, aby nedošlo k přepsání našich lokálních změn při změně nastavení přes webové rozhraní.

- Šablony jsou uloženy v adresáři `/usr/share/zentyal/stubs`
- Zde nalezneme adresáře k jednotlivým službám, např. `mail` pro poštovní služby.
- šablony jsou označeny příponou `.mas`, tedy např. pro soubor main.cf editujeme soubor `main.cf.mas`
- Po dokončení editace provedeme restart příslušné zentyal služby (odpovídá názvu adresáře). tj. pro poštu to bude `/etc/init.d/zentyal mail restart`. Tímto se provede vygenerování nových konfiguračních souborů restart souvisejících služeb. **Restart je nutné provádět přes démona zentyal, protože jinak nedojde k přegenerování konfiguračních souborů a tím pádem se nám změny provedené v šablonách neaplikují**

Konfigurace pořadí spouštění modulů zentyalu

Pořadí ve kterém se moduly zavádějí skriptem /etc/init.d/zentyal a při bootu se konfigurují nastavením závislostí v souborech *.yaml v /usr/share/zentyal/modules/ příklad ipsec:

```
class: 'EBox::IPsec'
depends:
    - network
    - firewall

bootdepends:
    - samba

models:
    - Connections
    - SettingsIPsec
    - SettingsL2TP
    - ConfPhase1
    - ConfPhase2
    - RangeTable
    - UsersFile
    - Users
```

```
composites:
    IPsecConf: [SettingsIPsec, Auth]
    IPsecL2TPConf: [GeneralL2TPSettings, UsersSettings]
    GeneralL2TPSettings: [SettingsL2TP, RangeTable]
    Auth: [ConfPhase1, ConfPhase2]
    UsersSettings: [Users, UsersFile]
```

 warning

!!!!Poškození služeb aktualizací libldb1!!!!

Předem doporučuji provádět jakoukoliv aktualizaci z konzole a ne přes webové rozhraní kde nastává při aktualizacích mnoho problémů a to klasicky:

```
apt-get update
apt-get upgrade
```

V repozitářích zentyalu se objevila nová verze libldb1 1:1.1.17-0ubuntu0.14.04.1 a 1:1.1.24-0ubuntu0.14.04.1. Tyto verze v žádném případě neinstalujte, jejich instalace vede k znefunkčnění zentyal-samba a všech zentyal balíků na této službě závislých, vede ke ztrátě konfigurace zentyalu. libldb1 a na něm závislý balík python-ldb jsou ve stavu přidržení nicméně je možné je aktualizovat přes web rozhraní zentyalu kde se nabízí k aktualizaci, neaktualizujte je!!!!! Pokud nedopatřením aktualizaci provedete je potřeba provést downgrade balíků ldb-tools, libldb1, python-ldb na verzi 1:1.1.16-1 následovně:

```
apt-get install $nazev_baliku=1:1.1.16-1
```

Po downgrade lze znovu nainstalovat komponenty zentyalu nicméně s největší pravděpodobností nebude možné obnovit předchozí konfiguraci, toto jsem nezkoušel řešit.

 warning

!!!!Zruseni sluzby zentyal-mailfilter aktualizaci dovecot!!!!

po instalaci dovecot verze 1:2.2.13-2c~trusty a vyssi se odinstaluje sluzba mailfilter, tato verze je hold takze lze nainstalovat pres webove rozhrani ale apt-get upgrade ji podrzi ve verzi 1:2.2.9-1ubuntu2.1

 warning

!!!!Znefunkcneni Openchange pri deprovision/reprovision!!!!

Pokud zalozite organizaci openchange a nasledne ji zrusite nebude pravdepodobne openchange nadale fungovat (tyka se i reinstalace openchange) Pri zalozeni nove organizace openchange nebo pouziti jiz existujici zalozene drive nelze pridavat uzivatele ani odebirat z openchange, problem nastal po reinstalaci openchange a naslednem zalozeni nove organizace. Puvodni organizace nelze odebrat prikazem:

```
openchange_provision --deprovision --firstorg="NazevOrganizace"
```

vyhlasi totiz:

```
[!] Unable to unregister this server, it's being used for: primary receipt
update service server
```

a na toto jsem nenasel reseni. 

 important

ntlm_auth

Nekteré služby využívají pro přístup k LDAP program ntlm_auth, který je součástí balíku winbind je proto potřeba jej doinstalovat jelikož v default konfiguraci systemu není nainstalován :)

```
sudo apt-get install winbind
```

L2TPIPsec VPN na serveru

Pokud nefunguje konfigurace L2TPIPsec VPV pro konkretní skupinu uživatelů je potřeba oeditovat soubor /usr/share/zentyal/stubs/ipsec/options.xl2tpd.mas a v něm zaměnit v řádku:

```
ntlm_auth-helper "/usr/bin/ntlm_auth --helper-protocol=ntlm-server-1 --
require-membership-of='<% $group %>'"
parametr --require-membership-of='<% $group %> na:
ntlm_auth-helper '/usr/bin/ntlm_auth --helper-protocol=ntlm-server-1 --
require-membership-of="domena\\skupina"'.
```

Pri pripojovani zarizeni a iOS a Android je potreba pridat do souboru /usr/share/zatyal/stubs/ipsec/ipsec.conf.mas nasledujici:

```
dpddelay=15
dpdtimeout=30
```

```
dpdaction=clear
```

takto:

```
<%args>
    @tunnels
</%args>
# /etc/ipsec.conf - Openswan IPsec configuration file
# RCSID $Id: ipsec.conf.in,v 1.16 2005/07/26 12:29:45 ken Exp $

# This file:  /usr/share/doc/openswan/ipsec.conf-sample
#
# Manual:     ipsec.conf.5


version 2.0     # conforms to second version of ipsec.conf specification

# basic configuration
config setup
        # Do not set debug options to debug configuration issues!
        # plutodebug / klipsdebug = "all", "none" or a combation from below:
        # "raw crypt parsing emitting control klips pfkey natt x509 dpd
private"
        # eg:
        # plutodebug="control parsing"
        #
        # enable to get logs per-peer
        # plutoopts="--perpeerlog"
        #
        # Again: only enable plutodebug or klipsdebug when asked by a
developer
        #
        # NAT-TRAVERSAL support, see README.NAT-Traversal
        nat_traversal=yes
        # exclude networks used on server side by adding %v4:!a.b.c.0/24
        virtual_private=%v4:10.0.0.0/8,%v4:192.168.0.0/16,%v4:172.16.0.0/12
        # OE is now off by default. Uncomment and change to on, to enable.
        oe=off
        # which IPsec stack to use. netkey,klips,mast,auto or none
        protostack=netkey
        #
        interfaces=%none


# Add connections here

# sample VPN connection
# for more examples, see /etc/ipsec.d/examples/
#conn sample
#               # Left security gateway, subnet behind it, nexthop toward
right.
```

```
#               left=10.0.0.1
#               leftsubnet=172.16.0.0/24
#               leftnexthop=10.22.33.44
#               # Right security gateway, subnet behind it, nexthop toward
left.
#               right=10.12.12.1
#               rightsubnet=192.168.0.0/24
#               rightnexthop=10.101.102.103
#               # To authorize this connection, but not actually start it,
#               # at startup, uncomment this.
#               #auto=start

% foreach my $tunnel (@tunnels) {

# VPN: <% $tunnel->{'name'} %> (<% $tunnel->{'type'} %>): <%
$tunnel->{'left_ipaddr'} %> <=> <% $tunnel->{'right_ipaddr'} %>
conn <% $tunnel->{'name'} %>
        left=<% $tunnel->{'left_ipaddr'} %>
        right=<% $tunnel->{'right_ipaddr'} %>
%   if ($tunnel->{'type'} eq 'ipsec') {
        rekey=yes
        keyingtries=0
        leftsubnet=<% $tunnel->{'left_subnet'} %>
        rightsubnet=<% $tunnel->{'right_subnet'} %>
%     if ( $tunnel->{'pfs'} ) {
        pfs=yes
%     } else {
        pfs=no
%     }
        auth=esp
        keyexchange=ike
%     if ( $tunnel->{'ike-enc'} ne 'any' ) {
        ike=<% $tunnel->{'ike-enc'} %>-<% $tunnel->{'ike-auth'} %>
%     }
        ikelifetime=<% $tunnel->{'ike-keylife'} %>s
%     if ( $tunnel->{'ike-enc'} ne 'any' ) {
        esp=<% $tunnel->{'phase2-enc'} %>-<% $tunnel->{'phase2-auth'} %>;<%
$tunnel->{'phase2-dhgroup'} %>
%     }
        keylife=<% $tunnel->{'phase2-keylife'} %>s
%   } elsif ($tunnel->{'type'} eq 'l2tp') {
        rekey=no
        keyingtries=3
        pfs=no
        leftprotoport=17/1701
        rightprotoport=17/%any
        dpddelay=15
        dpdtimeout=30
        dpdaction=clear
%   }
        authby=secret
```

```
        auto=start
% }
```

L2TPIPsec VPN na klientovi (Windows 7)

Pokud je Zentyal server za NATem je potřeba na klientovi nastavit následující: Vyhledejte a vyberte následující podklíč registru: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PolicyAgent V nabídce Úpravy přejděte na příkaz Nový a potom klepněte na příkaz Hodnota DWORD (32bitová) název hodnoty: AssumeUDPEncapsulationContextOnSendRule, a stiskněte klávesu ENTER. AssumeUDPEncapsulationContextOnSendRule klepněte pravým tlačítkem myši a potom klepněte na příkaz změnit.
Do pole Údaj hodnoty zadejte jednu z následujících hodnot:
0 - Hodnota 0 (nula) nastaví systém Windows tak, že jej nelze vytvořit přidružení zabezpečení se servery, které jsou umístěny za zařízení NAT. Toto je výchozí hodnota.
1 - Hodnota 1 nastaví systém Windows tak, že ji lze vytvořit přidružení zabezpečení se servery, které jsou umístěny za zařízení NAT.
2 - Hodnota 2 nastaví systém Windows tak, aby je vytvoření přidružení zabezpečení, když server a klientský počítač virtuální privátní sítě založené na systému Windows Vista nebo systémem Windows Server 2008 jsou za zařízení NAT.

 important

U l2tp/IPsec je potreba pocitat s tim ze pokud se pripojuje vice klientu ze stejne IP za NATem po pripojeni prvniho zarizeni na VPN nemusi dalsim klientum fungovat pripojeni na VPN. Napr. kdyz se 1. pripoji zarizeni s Windows7 je mozne se na VPN pripojit ze zarizeni Android ale pokud se prvni pripoji zarizeni Android nepujde se pripojit s Windows7 dokud se Android neodpoji. Je to zpusobeno nestandardni implementaci l2tp/IPsec v Androidu a iOS kdy ip xfrm policy nejsou správně definovány při začátku a mazány při ukončování komunikace.

 important

!!!!Vypršení hesla administratora!!!!

Toto je dulezity bug zentyal-samba, je třeba opravit při všech implemetacích. Heslo administratora je nastaveno tak že po 1 roce vyprší, problém je že zentyal zavádí některé důležité moduly pod účtem administrator. Symptomy: zentyal po rebootu/restartu nezavede některé muduly a v logu je vidět hláška "password has expired". Workaround:

```
sudo samba-tool user setexpiry administrator --noexpiry
```

Problém se zálohováním duplicity

Po neůspěšném pokusu o zálohování nelze načíst zálohy přes webadmin a zálohovací proces hlásí do logu že se nepodařil spustit s hláškou "backup process allready running". Je potřeba smazat zálohovací flag:
/var/cache/zentyal/duplicity/řetězec zálohovacího úkolu/lockfile.lock nebo celý adresář zálohovací cache.
a restartovat zálohovací službu:
/etc/init.d/zentyal ebackup restart

Po přihlášení stroje s Windows 10 do domény se nelze přihlásit k účtům google, microsoft atd.

Je potřeba vytvořit na klientovy klíč registru:
HKEY_LOCAL_MACHINE\Software\Microsoft\Cryptography\Protect\Providers\df9d8cd0-1501-11d1-8c7a-00c04fc297eb\ProtectionPolicy jako DWORD s hodnotou 1

Problém s cestovními profily

V některých případech není po založení nového uživatele vygenerován adresář pro jeho cestovní profil, z tohoto důvodu cestovní profil není funkční. Řešení: po přidání novéhu uživatele odškrtnout "enable roaming profiles" na kartě Doména/Nastavení → uložit změny → znovu zaškrtnout "enable roaming profiles" → uložit změny, po této operaci by měly být vygenerovány chybějící adresáře pro profily v cestě /home/samba/profiles/ (Tato operace nepřepíše existující adresáře cestovních profilů :)
Tento problém vzniká pravděpodobně updatem zentyalu, protože ve starsi verzi 3.5.x se neobjevuje.

Upgrade blíčků systému

Při upgrade balíčků systému pravděpodobně dojde k chybě při upgrade IDS/IPS Suricata a do logu hlásí "NFQOEUE not supported" nebo něco v tom smyslu, za následek to má že zentyal nestartuje.
Workaround:
před upgradem balíčků zavést do systému modul nfnetlink_queue následovně:

```
sudo modprobe nfnetlink_queue
```

pro ověření zavedení modulu provést:

```
lsmod | grep nfnet
```

provést upgrade:

```
apt-get update
apt-get upgrade
```

RADIUS

S radiusem je hned nekolik problemu:

- Nestartuje a do logu hlasi:

```
Tue Feb 23 21:15:32 2016 : Error: /etc/freeradius/users[3]: Parse error
(check) for entry DEFAULT: Expected end of line or comma
Tue Feb 23 21:15:32 2016 : Error: Errors reading /etc/freeradius/users
Tue Feb 23 21:15:32 2016 : Error: /etc/freeradius/modules/files[7]:
Instantiation failed for module "files"
Tue Feb 23 21:15:32 2016 : Error: /etc/freeradius/sites-
enabled/default[152]: Failed to load module "files".
Tue Feb 23 21:15:32 2016 : Error: /etc/freeradius/sites-enabled/default[62]:
Errors parsing authorize section.
Tue Feb 23 21:15:32 2016 : Error: Failed to load virtual server <default>
```

Reseni:
Oeditovat DEFAULT LDAP-Group v /usr/share/zentyal/stubs/radius/users.mas takto:

```
DEFAULT LDAP-Group == "<% $group %>"
```

- Dalsim problemem je selhavani autentifikace, pro pouziti freeradius v kombinaci samba-ldap se doporucuje take odkomentovat radky:

```
chase_referrals = yes
rebind = yes
```

v /usr/share/zentyal/stubs/radius/ldap.mas

- Pro otestovani spravneho fungovani autentifikace je mozne pouzit radtest z localu nasledovne:

```
radtest user password localhost 1812 sdilenehesloradius
```

pred testovanim je potreba pridat do /etc/freeradius/clients.conf:

```
client localhost {
        ipaddr = 127.0.0.1
        secret = sdilenehesloradius
        nastype = other
}
```

a restartovat radius.

V dashboardu hlasi ze nebezi i kdyz bezi
Reseni:
Pridat:

```
post-start script
```

```
PID=`status freeradius | egrep -oi '([0-9]+)$' | head -n1`
echo $PID > /var/run/freeradius/freeradius.pid
end script
```

do souboru:
/etc/init/freeradius.conf

Pri testovani pres radtest se user bez problemu zaloguje, ovsem pri prihlaseni pres klienta (Windows7) hlasi login incorect. Reseni:
Zde je tolik zmen ze radeji pastuju primo cele konfiguraky:
/usr/share/zentyal/stubs/radius/eap.conf.mas

```
<%args>
    $capath
</%args>
# -*- text -*-
##
##  eap.conf -- Configuration for EAP types (PEAP, TTLS, etc.)
##
##      $Id$

######################################################################
#
#  Whatever you do, do NOT set 'Auth-Type := EAP'.  The server
#  is smart enough to figure this out on its own.  The most
#  common side effect of setting 'Auth-Type := EAP' is that the
#  users then cannot use ANY other authentication method.
#
#  EAP types NOT listed here may be supported via the "eap2" module.
#  See experimental.conf for documentation.
#
        eap {
                #  Invoke the default supported EAP type when
                #  EAP-Identity response is received.
                #
                #  The incoming EAP messages DO NOT specify which EAP
                #  type they will be using, so it MUST be set here.
                #
                #  For now, only one default EAP type may be used at a time.
                #
                #  If the EAP-Type attribute is set by another module,
                #  then that EAP type takes precedence over the
                #  default type configured here.
                #
                default_eap_type = peap

                #  A list is maintained to correlate EAP-Response
                #  packets with EAP-Request packets.  After a
                #  configurable length of time, entries in the list
                #  expire, and are deleted.
                #
```

```
                timer_expire     = 60

                #  There are many EAP types, but the server has support
                #  for only a limited subset.  If the server receives
                #  a request for an EAP type it does not support, then
                #  it normally rejects the request.  By setting this
                #  configuration to "yes", you can tell the server to
                #  instead keep processing the request.  Another module
                #  MUST then be configured to proxy the request to
                #  another RADIUS server which supports that EAP type.
                #
                #  If another module is NOT configured to handle the
                #  request, then the request will still end up being
                #  rejected.
                ignore_unknown_eap_types = no

                # Cisco AP1230B firmware 12.2(13)JA1 has a bug.  When given
                # a User-Name attribute in an Access-Accept, it copies one
                # more byte than it should.
                #
                # We can work around it by configurably adding an extra
                # zero byte.
                cisco_accounting_username_bug = no

                #
                #  Help prevent DoS attacks by limiting the number of
                #  sessions that the server is tracking.  Most systems
                #  can handle ~30 EAP sessions/s, so the default limit
                #  of 4096 should be OK.
                max_sessions = 4096

                # Supported EAP-types

                #
                #  We do NOT recommend using EAP-MD5 authentication
                #  for wireless connections.  It is insecure, and does
                #  not provide for dynamic WEP keys.
                #
                md5 {
                }

                # Cisco LEAP
                #
                #  We do not recommend using LEAP in new deployments.  See:
                #  http://www.securiteam.com/tools/5TP012ACKE.html
                #
                #  Cisco LEAP uses the MS-CHAP algorithm (but not
                #  the MS-CHAP attributes) to perform it's authentication.
                #
                #  As a result, LEAP *requires* access to the plain-text
                #  User-Password, or the NT-Password attributes.
```

```
#  'System' authentication is impossible with LEAP.
#
leap {
}

#  Generic Token Card.
#
#  Currently, this is only permitted inside of EAP-TTLS,
#  or EAP-PEAP.  The module "challenges" the user with
#  text, and the response from the user is taken to be
#  the User-Password.
#
#  Proxying the tunneled EAP-GTC session is a bad idea,
#  the users password will go over the wire in plain-text,
#  for anyone to see.
#
gtc {
        #  The default challenge, which many clients
        #  ignore..
        #challenge = "Password: "

        #  The plain-text response which comes back
        #  is put into a User-Password attribute,
        #  and passed to another module for
        #  authentication.  This allows the EAP-GTC
        #  response to be checked against plain-text,
        #  or crypt'd passwords.
        #
        #  If you say "Local" instead of "PAP", then
        #  the module will look for a User-Password
        #  configured for the request, and do the
        #  authentication itself.
        #
        auth_type = PAP
}

## EAP-TLS
#
#  See raddb/certs/README for additional comments
#  on certificates.
#
#  If OpenSSL was not found at the time the server was
#  built, the "tls", "ttls", and "peap" sections will
#  be ignored.
#
#  Otherwise, when the server first starts in debugging
#  mode, test certificates will be created.  See the
#  "make_cert_command" below for details, and the README
#  file in raddb/certs
#
#  These test certificates SHOULD NOT be used in a normal
```

```
                #  deployment.  They are created only to make it easier
                #  to install the server, and to perform some simple
                #  tests with EAP-TLS, TTLS, or PEAP.
                #
                #  See also:
                #
                #  http://www.dslreports.com/forum/remark,9286052~mode=flat
                #
                tls {
                        #
                        #  These is used to simplify later configurations.
                        #
                        certdir = ${confdir}/certs
                        cadir = ${confdir}/certs

                        private_key_password = whatever
                        private_key_file = ${certdir}/freeradius.pem

                        #  If Private key & Certificate are located in
                        #  the same file, then private_key_file &
                        #  certificate_file must contain the same file
                        #  name.
                        #
                        #  If CA_file (below) is not used, then the
                        #  certificate_file below MUST include not
                        #  only the server certificate, but ALSO all
                        #  of the CA certificates used to sign the
                        #  server certificate.
                        certificate_file = ${certdir}/freeradius.pem

                        #  Trusted Root CA list
                        #
                        #  ALL of the CA's in this list will be trusted
                        #  to issue client certificates for authentication.
                        #
                        #  In general, you should use self-signed
                        #  certificates for 802.1x (EAP) authentication.
                        #  In that case, this CA file should contain
                        #  *one* CA certificate.
                        #
                        #  This parameter is used only for EAP-TLS,
                        #  when you issue client certificates.  If you do
                        #  not use client certificates, and you do not want
                        #  to permit EAP-TLS authentication, then delete
                        #  this configuration item.
                        CA_file = <% $capath %>

                        #
                        #  For DH cipher suites to work, you have to
                        #  run OpenSSL to create the DH file first:
                        #
```

```
                        #       openssl dhparam -out certs/dh 1024
                        #
                        dh_file = ${certdir}/dh
                        random_file = /dev/urandom

                        #
                        #  This can never exceed the size of a RADIUS
                        #  packet (4096 bytes), and is preferably half
                        #  that, to accomodate other attributes in
                        #  RADIUS packet.  On most APs the MAX packet
                        #  length is configured between 1500 - 1600
                        #  In these cases, fragment size should be
                        #  1024 or less.
                        #
                #       fragment_size = 1024

                        #  include_length is a flag which is
                        #  by default set to yes If set to
                        #  yes, Total Length of the message is
                        #  included in EVERY packet we send.
                        #  If set to no, Total Length of the
                        #  message is included ONLY in the
                        #  First packet of a fragment series.
                        #
                #       include_length = yes

                        #  Check the Certificate Revocation List
                        #
                        #  1) Copy CA certificates and CRLs to same
directory.
                        #  2) Execute 'c_rehash <CA certs&CRLs Directory>'.
                        #    'c_rehash' is OpenSSL's command.
                        #  3) uncomment the line below.
                        #  5) Restart radiusd
                #       check_crl = yes
                #       CA_path = /path/to/directory/with/ca_certs/and/crls/

                       #
                       #  If check_cert_issuer is set, the value will
                       #  be checked against the DN of the issuer in
                       #  the client certificate.  If the values do not
                       #  match, the cerficate verification will fail,
                       #  rejecting the user.
                       #
                #       check_cert_issuer =
"/C=GB/ST=Berkshire/L=Newbury/O=My Company Ltd"

                       #
                       #  If check_cert_cn is set, the value will
                       #  be xlat'ed and checked against the CN
                       #  in the client certificate.  If the values
```

```
                        #  do not match, the certificate verification
                        #  will fail rejecting the user.
                        #
                        #  This check is done only if the previous
                        #  "check_cert_issuer" is not set, or if
                        #  the check succeeds.
                        #
                #       check_cert_cn = %{User-Name}
                #
                        # Set this option to specify the allowed
                        # TLS cipher suites.  The format is listed
                        # in "man 1 ciphers".
                        cipher_list = "DEFAULT"

                        #

                        #  This configuration entry should be deleted
                        #  once the server is running in a normal
                        #  configuration.  It is here ONLY to make
                        #  initial deployments easier.
                        #
                        #make_cert_command = "${certdir}/bootstrap"

                        #
                        #  Session resumption / fast reauthentication
                        #  cache.
                        #
                        cache {
                              #
                              #  Enable it.  The default is "no".
                              #  Deleting the entire "cache" subsection
                              #  Also disables caching.
                              #
                              #  You can disallow resumption for a
                              #  particular user by adding the following
                              #  attribute to the control item list:
                              #
                              #         Allow-Session-Resumption = No
                              #
                              #  If "enable = no" below, you CANNOT
                              #  enable resumption for just one user
                              #  by setting the above attribute to "yes".
                              #
                              enable = no

                              #
                              #  Lifetime of the cached entries, in hours.
                              #  The sessions will be deleted after this
                              #  time.
                              #
                              lifetime = 24 # hours
```

```
                #
                #  The maximum number of entries in the
                #  cache.  Set to "0" for "infinite".
                #
                #  This could be set to the number of users
                #  who are logged in... which can be a LOT.
                #
                max_entries = 255
        }
}

#  The TTLS module implements the EAP-TTLS protocol,
#  which can be described as EAP inside of Diameter,
#  inside of TLS, inside of EAP, inside of RADIUS...
#
#  Surprisingly, it works quite well.
#
#  The TTLS module needs the TLS module to be installed
#  and configured, in order to use the TLS tunnel
#  inside of the EAP packet.  You will still need to
#  configure the TLS module, even if you do not want
#  to deploy EAP-TLS in your network.  Users will not
#  be able to request EAP-TLS, as it requires them to
#  have a client certificate.  EAP-TTLS does not
#  require a client certificate.
#
#  You can make TTLS require a client cert by setting
#
#       EAP-TLS-Require-Client-Cert = Yes
#
#  in the control items for a request.
#
ttls {
        #  The tunneled EAP session needs a default
        #  EAP type which is separate from the one for
        #  the non-tunneled EAP module.  Inside of the
        #  TTLS tunnel, we recommend using EAP-MD5.
        #  If the request does not contain an EAP
        #  conversation, then this configuration entry
        #  is ignored.
        default_eap_type = md5

        #  The tunneled authentication request does
        #  not usually contain useful attributes
        #  like 'Calling-Station-Id', etc.  These
        #  attributes are outside of the tunnel,
        #  and normally unavailable to the tunneled
        #  authentication request.
        #
        #  By setting this configuration entry to
```

```
                #  'yes', any attribute which NOT in the
                #  tunneled authentication request, but
                #  which IS available outside of the tunnel,
                #  is copied to the tunneled request.
                #
                # allowed values: {no, yes}
                copy_request_to_tunnel = no

                #  The reply attributes sent to the NAS are
                #  usually based on the name of the user
                #  'outside' of the tunnel (usually
                #  'anonymous').  If you want to send the
                #  reply attributes based on the user name
                #  inside of the tunnel, then set this
                #  configuration entry to 'yes', and the reply
                #  to the NAS will be taken from the reply to
                #  the tunneled request.
                #
                # allowed values: {no, yes}
                use_tunneled_reply = no

                #
                #  The inner tunneled request can be sent
                #  through a virtual server constructed
                #  specifically for this purpose.
                #
                #  If this entry is commented out, the inner
                #  tunneled request will be sent through
                #  the virtual server that processed the
                #  outer requests.
                #
                virtual_server = "inner-tunnel"

                #  This has the same meaning as the
                #  same field in the "tls" module, above.
                #  The default value here is "yes".
        #       include_length = yes
        }

        ##################################################
        #
        #  !!!!! WARNINGS for Windows compatibility  !!!!!
        #
        ##################################################
        #
        #  If you see the server send an Access-Challenge,
        #  and the client never sends another Access-Request,
        #  then
        #
        #               STOP!
        #
```

```
#  The server certificate has to have special OID's
#  in it, or else the Microsoft clients will silently
#  fail.  See the "scripts/xpextensions" file for
#  details, and the following page:
#
#       http://support.microsoft.com/kb/814394/en-us
#
#  For additional Windows XP SP2 issues, see:
#
#       http://support.microsoft.com/kb/885453/en-us
#
#  Note that we do not necessarily agree with their
#  explanation... but the fix does appear to work.
#
##################################################

#
#  The tunneled EAP session needs a default EAP type
#  which is separate from the one for the non-tunneled
#  EAP module.  Inside of the TLS/PEAP tunnel, we
#  recommend using EAP-MS-CHAPv2.
#
#  The PEAP module needs the TLS module to be installed
#  and configured, in order to use the TLS tunnel
#  inside of the EAP packet.  You will still need to
#  configure the TLS module, even if you do not want
#  to deploy EAP-TLS in your network.  Users will not
#  be able to request EAP-TLS, as it requires them to
#  have a client certificate.  EAP-PEAP does not
#  require a client certificate.
#
#
#  You can make PEAP require a client cert by setting
#
#       EAP-TLS-Require-Client-Cert = Yes
#
#  in the control items for a request.
#
peap {
        #  The tunneled EAP session needs a default
        #  EAP type which is separate from the one for
        #  the non-tunneled EAP module.  Inside of the
        #  PEAP tunnel, we recommend using MS-CHAPv2,
        #  as that is the default type supported by
        #  Windows clients.
        default_eap_type = mschapv2

        #  the PEAP module also has these configuration
        #  items, which are the same as for TTLS.
        copy_request_to_tunnel = no
        use_tunneled_reply = no
```

```
                        #  When the tunneled session is proxied, the
                        #  home server may not understand EAP-MSCHAP-V2.
                        #  Set this entry to "no" to proxy the tunneled
                        #  EAP-MSCHAP-V2 as normal MSCHAPv2.
                #       proxy_tunneled_request_as_eap = yes

                        #
                        #  The inner tunneled request can be sent
                        #  through a virtual server constructed
                        #  specifically for this purpose.
                        #
                        #  If this entry is commented out, the inner
                        #  tunneled request will be sent through
                        #  the virtual server that processed the
                        #  outer requests.
                        #
                        virtual_server = "inner-tunnel"
                }

                #
                #  This takes no configuration.
                #
                #  Note that it is the EAP MS-CHAPv2 sub-module, not
                #  the main 'mschap' module.
                #
                #  Note also that in order for this sub-module to work,
                #  the main 'mschap' module MUST ALSO be configured.
                #
                #  This module is the *Microsoft* implementation of MS-
CHAPv2
                #  in EAP.  There is another (incompatible) implementation
                #  of MS-CHAPv2 in EAP by Cisco, which FreeRADIUS does not
                #  currently support.
                #
                mschapv2 {
                }
        }
```

/usr/share/zentyal/stubs/radius/inner-tunnel.mas

```
# -*- text -*-
######################################################################
#
#       This is a virtual server that handles *only* inner tunnel
#       requests for EAP-TTLS and PEAP types.
#
#       $Id$
#
######################################################################
```

```
server inner-tunnel {

#
#  Un-comment the next section to perform test on the inner tunnel
#  without needing an outer tunnel session.  The tests will not be
#  exactly the same as when TTLS or PEAP are used, but they will
#  be close enough for many tests.
#
#listen {
#       ipaddr = 127.0.0.1
#       port = 18120
#       type = auth
#}


#  Authorization. First preprocess (hints and huntgroups files),
#  then realms, and finally look in the "users" file.
#
#  The order of the realm modules will determine the order that
#  we try to find a matching realm.
#
#  Make *sure* that 'preprocess' comes before any realm if you
#  need to setup hints for the remote radius server
authorize {
        #
        #  The chap module will set 'Auth-Type := CHAP' if we are
        #  handling a CHAP request and Auth-Type has not already been set
        chap

        #
        #  If the users are logging in with an MS-CHAP-Challenge
        #  attribute for authentication, the mschap module will find
        #  the MS-CHAP-Challenge attribute, and add 'Auth-Type := MS-CHAP'
        #  to the request, which will cause the server to then use
        #  the mschap module for authentication.
        mschap

        #
        #  Pull crypt'd passwords from /etc/passwd or /etc/shadow,
        #  using the system API's to get the password.  If you want
        #  to read /etc/passwd or /etc/shadow directly, see the
        #  passwd module, above.
        #
#       unix

        #
        #  Look for IPASS style 'realm/', and if not found, look for
        #  '@realm', and decide whether or not to proxy, based on
        #  that.
#       IPASS
```

```
        #
        #  If you are using multiple kinds of realms, you probably
        #  want to set "ignore_null = yes" for all of them.
        #  Otherwise, when the first style of realm doesn't match,
        #  the other styles won't be checked.
        #
        #  Note that proxying the inner tunnel authentication means
        #  that the user MAY use one identity in the outer session
        #  (e.g. "anonymous", and a different one here
        #  (e.g. "user@example.com").  The inner session will then be
        #  proxied elsewhere for authentication.  If you are not
        #  careful, this means that the user can cause you to forward
        #  the authentication to another RADIUS server, and have the
        #  accounting logs *not* sent to the other server.  This makes
        #  it difficult to bill people for their network activity.
        #
        suffix
#       ntdomain

        #
        #  The "suffix" module takes care of stripping the domain
        #  (e.g. "@example.com") from the User-Name attribute, and the
        #  next few lines ensure that the request is not proxied.
        #
        #  If you want the inner tunnel request to be proxied, delete
        #  the next few lines.
        #
        update control {
               Proxy-To-Realm := LOCAL
        }

        #
        #  This module takes care of EAP-MSCHAPv2 authentication.
        #
        #  It also sets the EAP-Type attribute in the request
        #  attribute list to the EAP type from the packet.
        #
        #  The example below uses module failover to avoid querying all
        #  of the following modules if the EAP module returns "ok".
        #  Therefore, your LDAP and/or SQL servers will not be queried
        #  for the many packets that go back and forth to set up TTLS
        #  or PEAP.  The load on those servers will therefore be reduced.
        #
        eap {
                ok = return
        }

        #
        #  Read the 'users' file
        files
```

```
        #
        #  Look in an SQL database.  The schema of the database
        #  is meant to mirror the "users" file.
        #
        #  See "Authorization Queries" in sql.conf
#       sql

        #
        #  If you are using /etc/smbpasswd, and are also doing
        #  mschap authentication, the un-comment this line, and
        #  configure the 'etc_smbpasswd' module, above.
#       etc_smbpasswd

        #
        #  The ldap module will set Auth-Type to LDAP if it has not
        #  already been set
        ldap

        #
        #  Enforce daily limits on time spent logged in.
#       daily

        #
        # Use the checkval module
#       checkval

        expiration
        logintime

        #
        #  If no other module has claimed responsibility for
        #  authentication, then try to use PAP.  This allows the
        #  other modules listed above to add a "known good" password
        #  to the request, and to do nothing else.  The PAP module
        #  will then see that password, and use it to do PAP
        #  authentication.
        #
        #  This module should be listed last, so that the other modules
        #  get a chance to set Auth-Type for themselves.
        #
        pap
}


#  Authentication.
#
#
#  This section lists which modules are available for authentication.
#  Note that it does NOT mean 'try each module in order'.  It means
#  that a module from the 'authorize' section adds a configuration
#  attribute 'Auth-Type := FOO'.  That authentication type is then
```

```
#  used to pick the apropriate module from the list below.
#

#  In general, you SHOULD NOT set the Auth-Type attribute.  The server
#  will figure it out on its own, and will do the right thing.  The
#  most common side effect of erroneously setting the Auth-Type
#  attribute is that one authentication method will work, but the
#  others will not.
#
#  The common reasons to set the Auth-Type attribute by hand
#  is to either forcibly reject the user, or forcibly accept him.
#
authenticate {
        #
        #  PAP authentication, when a back-end database listed
        #  in the 'authorize' section supplies a password.  The
        #  password can be clear-text, or encrypted.
        Auth-Type PAP {
                pap
        }

        #
        #  Most people want CHAP authentication
        #  A back-end database listed in the 'authorize' section
        #  MUST supply a CLEAR TEXT password.  Encrypted passwords
        #  won't work.
        Auth-Type CHAP {
                chap
        }

        #
        #  MSCHAP authentication.
        Auth-Type MS-CHAP {
                mschap
        }

        #
        #  Pluggable Authentication Modules.
#       pam

        #
        #  See 'man getpwent' for information on how the 'unix'
        #  module checks the users password.  Note that packets
        #  containing CHAP-Password attributes CANNOT be authenticated
        #  against /etc/passwd!  See the FAQ for details.
        #
#       unix

        # Uncomment it if you want to use ldap for authentication
        #
        # Note that this means "check plain-text password against
```

```
        # the ldap database", which means that EAP won't work,
        # as it does not supply a plain-text password.
        Auth-Type LDAP {
                ldap
        }

        #
        #  Allow EAP authentication.
        eap
}

#######################################################################
#
#       There are no accounting requests inside of EAP-TTLS or PEAP
#       tunnels.
#
#######################################################################


#  Session database, used for checking Simultaneous-Use. Either the radutmp
#  or rlm_sql module can handle this.
#  The rlm_sql module is *much* faster
session {
        radutmp

        #
        #  See "Simultaneous Use Checking Queries" in sql.conf
#       sql
}


#  Post-Authentication
#  Once we KNOW that the user has been authenticated, there are
#  additional steps we can take.
post-auth {
        # Note that we do NOT assign IP addresses here.
        # If you try to assign IP addresses for EAP authentication types,
        # it WILL NOT WORK.  You MUST use DHCP.

        #
        #  If you want to have a log of authentication replies,
        #  un-comment the following line, and the 'detail reply_log'
        #  section, above.
#       reply_log

        #
        #  After authenticating the user, do another SQL query.
        #
        #  See "Authentication Logging Queries" in sql.conf
#       sql
```

```
        #
        #  Instead of sending the query to the SQL server,
        #  write it into a log file.
        #
#       sql_log

        #
        #  Un-comment the following if you have set
        #  'edir_account_policy_check = yes' in the ldap module sub-section
of
        #  the 'modules' section.
        #
#       ldap

        #
        #  Access-Reject packets are sent through the REJECT sub-section of
the
        #  post-auth section.
        #
        #  Add the ldap module name (or instance) if you have set
        #  'edir_account_policy_check = yes' in the ldap module
configuration
        #
        Post-Auth-Type REJECT {
                attr_filter.access_reject
        }

        #
        #  The example policy below updates the outer tunnel reply
        #  (usually Access-Accept) with the User-Name from the inner
        #  tunnel User-Name.  Since this section is processed in the
        #  context of the inner tunnel, "request" here means "inner
        #  tunnel request", and "outer.reply" means "outer tunnel
        #  reply attributes".
        #
        #  This example is most useful when the outer session contains
        #  a User-Name of "anonymous@....", or a MAC address.  If it
        #  is enabled, the NAS SHOULD use the inner tunnel User-Name
        #  in subsequent accounting packets.  This makes it easier to
        #  track user sessions, as they will all be based on the real
        #  name, and not on "anonymous".
        #
        #  The problem with doing this is that it ALSO exposes the
        #  real user name to any intermediate proxies.  People use
        #  "anonymous" identifiers outside of the tunnel for a very
        #  good reason: it gives them more privacy.  Setting the reply
        #  to contain the real user name removes ALL privacy from
        #  their session.
        #
        #  If you want privacy to remain, see the
        #  Chargeable-User-Identity attribute from RFC 4372.  In order
```

```
        #  to use that attribute, you will have to allocate a
        #  per-session identifier for the user, and store it in a
        #  long-term database (e.g. SQL).  You should also use that
        #  attribute INSTEAD of the configuration below.
        #
        #update outer.reply {
        #       User-Name = "%{request:User-Name}"
        #}

}

#
#  When the server decides to proxy a request to a home server,
#  the proxied request is first passed through the pre-proxy
#  stage.  This stage can re-write the request, or decide to
#  cancel the proxy.
#
#  Only a few modules currently have this method.
#
pre-proxy {
#       attr_rewrite

        #  Uncomment the following line if you want to change attributes
        #  as defined in the preproxy_users file.
#       files

        #  Uncomment the following line if you want to filter requests
        #  sent to remote servers based on the rules defined in the
        #  'attrs.pre-proxy' file.
#       attr_filter.pre-proxy

        #  If you want to have a log of packets proxied to a home
        #  server, un-comment the following line, and the
        #  'detail pre_proxy_log' section, above.
#       pre_proxy_log
}

#
#  When the server receives a reply to a request it proxied
#  to a home server, the request may be massaged here, in the
#  post-proxy stage.
#
post-proxy {

        #  If you want to have a log of replies from a home server,
        #  un-comment the following line, and the 'detail post_proxy_log'
        #  section, above.
#       post_proxy_log

#       attr_rewrite
```

```
        #  Uncomment the following line if you want to filter replies from
        #  remote proxies based on the rules defined in the 'attrs' file.
#       attr_filter.post-proxy

        #
        #  If you are proxying LEAP, you MUST configure the EAP
        #  module, and you MUST list it here, in the post-proxy
        #  stage.
        #
        #  You MUST also use the 'nostrip' option in the 'realm'
        #  configuration.  Otherwise, the User-Name attribute
        #  in the proxied request will not match the user name
        #  hidden inside of the EAP packet, and the end server will
        #  reject the EAP request.
        #
        eap

        #
        #  If the server tries to proxy a request and fails, then the
        #  request is processed through the modules in this section.
        #
        #  The main use of this section is to permit robust proxying
        #  of accounting packets.  The server can be configured to
        #  proxy accounting packets as part of normal processing.
        #  Then, if the home server goes down, accounting packets can
        #  be logged to a local "detail" file, for processing with
        #  radrelay.  When the home server comes back up, radrelay
        #  will read the detail file, and send the packets to the
        #  home server.
        #
        #  With this configuration, the server always responds to
        #  Accounting-Requests from the NAS, but only writes
        #  accounting packets to disk if the home server is down.
        #
#       Post-Proxy-Type Fail {
#                       detail
#       }

}

} # inner-tunnel server block
```

/usr/share/zentyal/stubs/radius/ldap.mas

```
<%args>
    $url
    $dn
    $rootdn
    $password
</%args>
# -*- text -*-
```

```
#
#  $Id$

# Lightweight Directory Access Protocol (LDAP)
#
#  This module definition allows you to use LDAP for
#  authorization and authentication.
#
#  See raddb/sites-available/default for reference to the
#  ldap module in the authorize and authenticate sections.
#
#  However, LDAP can be used for authentication ONLY when the
#  Access-Request packet contains a clear-text User-Password
#  attribute.  LDAP authentication will NOT work for any other
#  authentication method.
#
#  This means that LDAP servers don't understand EAP.  If you
#  force "Auth-Type = LDAP", and then send the server a
#  request containing EAP authentication, then authentication
#  WILL NOT WORK.
#
#  The solution is to use the default configuration, which does
#  work.
#
#  Setting "Auth-Type = LDAP" is ALMOST ALWAYS WRONG.  We
#  really can't emphasize this enough.
#
ldap {
        #
        #  Note that this needs to match the name in the LDAP
        #  server certificate, if you're using ldaps.
        server = "<% $url %>"
        identity = "<% $rootdn %>"
        password = "<% $password %>"
        basedn = "<% $dn %>"
        filter = "(samAccountName=%{Stripped-User-Name:-%{User-Name}})"
        #base_filter = "(objectclass=radiusprofile)"

        #  How many connections to keep open to the LDAP server.
        #  This saves time over opening a new LDAP socket for
        #  every authentication request.
        ldap_connections_number = 5

        # seconds to wait for LDAP query to finish. default: 20
        timeout = 5

        #  seconds LDAP server has to process the query (server-side
        #  time limit). default: 20
        #
        #  LDAP_OPT_TIMELIMIT is set to this value.
        timelimit = 5
```

```
        #
        #  seconds to wait for response of the server. (network
        #   failures) default: 10
        #
        #  LDAP_OPT_NETWORK_TIMEOUT is set to this value.
        net_timeout = 1

        #
        #  This subsection configures the tls related items
        #  that control how FreeRADIUS connects to an LDAP
        #  server.  It contains all of the "tls_*" configuration
        #  entries used in older versions of FreeRADIUS.  Those
        #  configuration entries can still be used, but we recommend
        #  using these.
        #
        tls {
                # Set this to 'yes' to use TLS encrypted connections
                # to the LDAP database by using the StartTLS extended
                # operation.
                #
                # The StartTLS operation is supposed to be
                # used with normal ldap connections instead of
                # using ldaps (port 689) connections
                start_tls = no

                # cacertfile    = /path/to/cacert.pem
                # cacertdir             = /path/to/ca/dir/
                # certfile              = /path/to/radius.crt
                # keyfile               = /path/to/radius.key
                # randfile              = /path/to/rnd

                #  Certificate Verification requirements.  Can be:
                #    "never" (don't even bother trying)
                #    "allow" (try, but don't fail if the cerificate
                #               can't be verified)
                #    "demand" (fail if the certificate doesn't verify.)
                #
                #       The default is "allow"
                # require_cert  = "demand"
        }

        # default_profile = "cn=radprofile,ou=dialup,o=My Org,c=UA"
        # profile_attribute = "radiusProfileDn"
        # access_attr = "dialupAccess"

        # Mapping of RADIUS dictionary attributes to LDAP
        # directory attributes.
        dictionary_mapping = ${confdir}/ldap.attrmap

        #  Set password_attribute = nspmPassword to get the
```

```
        #  user's password from a Novell eDirectory
        #  backend. This will work ONLY IF FreeRADIUS has been
        #  built with the --with-edir configure option.
        #
        #  See also the following links:
        #
        #  http://www.novell.com/coolsolutions/appnote/16745.html
        #
https://secure-support.novell.com/KanisaPlatform/Publishing/558/3009668_f.SA
L_Public.html
        #
        #  Novell may require TLS encrypted sessions before returning
        #  the user's password.
        #
        # password_attribute = userPassword

        #  Un-comment the following to disable Novell
        #  eDirectory account policy check and intruder
        #  detection. This will work *only if* FreeRADIUS is
        #  configured to build with --with-edir option.
        #
        edir_account_policy_check = no

        #
        #  Group membership checking.  Disabled by default.
        #
        groupname_attribute = cn
    groupmembership_filter = "(&(objectClass=group)(member=%{Stripped-User-
Name:-%{User-Name}}))"
        groupmembership_attribute = memberOf

        # compare_check_items = yes
        # do_xlat = yes
        # access_attr_used_for_allow = yes

        #
        #  The following two configuration items are for Active Directory
        #  compatibility.  If you see the helpful "operations error"
        #  being returned to the LDAP module, uncomment the next
        #  two lines.

         chase_referrals = yes
         rebind = yes

        #
        #  By default, if the packet contains a User-Password,
        #  and no other module is configured to handle the
        #  authentication, the LDAP module sets itself to do
        #  LDAP bind for authentication.
        #
        #  THIS WILL ONLY WORK FOR PAP AUTHENTICATION.
```

```
        #
        #  THIS WILL NOT WORK FOR CHAP, MS-CHAP, or 802.1x (EAP).
        #
        #  You can disable this behavior by setting the following
        #  configuration entry to "no".
        #
        #  allowed values: {no, yes}
        # set_auth_type = yes

        #  ldap_debug: debug flag for LDAP SDK
        #  (see OpenLDAP documentation).  Set this to enable
        #  huge amounts of LDAP debugging on the screen.
        #  You should only use this if you are an LDAP expert.
        #
        #       default: 0x0000 (no debugging messages)
        #       Example:(LDAP_DEBUG_FILTER+LDAP_DEBUG_CONNS)
        #ldap_debug = 0x0028
}
```

/etc/freeradius/modules/mschap

```
# -*- text -*-
#
#  $Id$

# Microsoft CHAP authentication
#
#  This module supports MS-CHAP and MS-CHAPv2 authentication.
#  It also enforces the SMB-Account-Ctrl attribute.
#
mschap {
        #
        #  If you are using /etc/smbpasswd, see the 'passwd'
        #  module for an example of how to use /etc/smbpasswd
authtype = MS-CHAP
        # if use_mppe is not set to no mschap will
        # add MS-CHAP-MPPE-Keys for MS-CHAPv1 and
        # MS-MPPE-Recv-Key/MS-MPPE-Send-Key for MS-CHAPv2
        #
        use_mppe = yes

        # if mppe is enabled require_encryption makes
        # encryption moderate
        #
        require_encryption = yes

        # require_strong always requires 128 bit key
        # encryption
        #
        require_strong = yes
```

```
        # Windows sends us a username in the form of
        # DOMAIN\user, but sends the challenge response
        # based on only the user portion.  This hack
        # corrects for that incorrect behavior.
        #
        with_ntdomain_hack = yes

        # The module can perform authentication itself, OR
        # use a Windows Domain Controller.  This configuration
        # directive tells the module to call the ntlm_auth
        # program, which will do the authentication, and return
        # the NT-Key.  Note that you MUST have "winbindd" and
        # "nmbd" running on the local machine for ntlm_auth
        # to work.  See the ntlm_auth program documentation
        # for details.
        #
        # If ntlm_auth is configured below, then the mschap
        # module will call ntlm_auth for every MS-CHAP
        # authentication request.  If there is a cleartext
        # or NT hashed password available, you can set
        # "MS-CHAP-Use-NTLM-Auth := No" in the control items,
        # and the mschap module will do the authentication itself,
        # without calling ntlm_auth.
        #
        # Be VERY careful when editing the following line!
        #
        # You can also try setting the user name as:
        #
        #       ... --username=%{mschap:User-Name} ...
        #
        # In that case, the mschap module will look at the User-Name
        # attribute, and do prefix/suffix checks in order to obtain
        # the "best" user name for the request.
        #
        ntlm_auth = "/usr/bin/ntlm_auth --request-nt-key --
username=%{%{Stripped-User-Name}:-%{%{User-Name}:-None}} --
challenge=%{%{mschap:Challenge}:-00} --nt-response=%{%{mschap:NT-
Response}:-00}"

        # For Apple Server, when running on the same machine as
        # Open Directory.  It has no effect on other systems.
        #
#       use_open_directory = yes

        # On failure, set (or not) the MS-CHAP error code saying
        # "retries allowed".
        allow_retry = yes

        # An optional retry message.
#       retry_msg = "Re-enter (or reset) the password"
}
```

Dale je potreba vyaktualizovat freeradius na v. 2.2.9 ktera neni v repozitarich zentyalu:

```
apt-get install software-properties-common
add-apt-repository ppa:freeradius/stable
apt-get update
apt-get install freeradius
apt-get install freeradius-common
```

AMAVIS

Amavis obcas spadne a v mailove fronte zustavaji viset maily s hlaskou ze se servis ukoncuje nebo neco v tom smyslu.

Reseni: v Zentyalu je pouzita stara verze amavisu, je potreba vyupdateovat, toto ovsem neni mozne protoze novejsi verze neni v repozitarich ktere zentyal pouziva, je proto potreba ho pridat, nainstalovat amavis a restartovat sluzbu nasledovne:

```
apt-get install software-properties-common
add-apt-repository ppa:patrickdk/general-lucid
apt-get update
apt-get install amavis
/etc/init.d/zentyal mailfilter restart
```

JABBER

Hlasi:

```
E(<0.357.0>:eldap_utils:166) : failed to parse LDAP filter:
** Filter: []
** Reason: {error,["syntax error before: ",[]]}
```

Reseni:
odebrat radek

```
{ldap_filter, ""},
```

v sekci mod_shared_roster_ldap
v /usr/share/zentyal/stubs/jabber/ejabberd.cfg.mas

Zajimave odkazy

https://wiki.zentyal.org/wiki/Building_and_maintaining_a_contrib_Zentyal_module