# Téma 12: WiFi s PSK a EAP v CentOS

Vytvoření WiFi přístupového bodu s PSK a s EAP, nastavení FreeRADIUS serveru

Téma 12: WiFi s PSK a EAP v CentOS

# Teoretické znalosti

Bezdrátové sítě jsou dnes téměř všude. O to důležitější je zajistit jejich zabezpečení. Jsou dva základní typy, jak může být síť zabezpečená:

- Sdílené heslo (WPA-PSK = WPA-Personal)
- Samostatné ověření uživatele (WPA-EAP = WPA-Enterprise *(využívá Extensible Authentication Protocol)*)

Nastavení sdíleného hesla (PSK) je velice jednoduché. Problém je, že všichni uživatelé budou mít **stejné** heslo. To se výborně hodí pro domácí použití, nebo firmy s několika málo zaměstnanci. Všude jinde toto řešení není možné – dřív nebo později heslo unikne. I dnes se najdou zařízení, která neumí nic jiného, než WPA-PSK. Příkladem může být čtečka elektronických knih Amazon Kindle 3 – WPA-EAP neumí.

Druhý způsob spočívá v ověření uživatel proti nějakému centrálnímu bodu, tím bývá nejčastěji RADIUS server. Zde máme prakticky neomezené možnosti v nastavování. Mezi základní typy ověření patří využití jména a hesla (každý uživatel své), nebo využití certifikátů pro každého uživatele.

Na rozdíl od koupených řešení, si můžeme v Linuxu vše přizpůsobit na míru požadavkům. Všechen potřebný software je navíc k dispozici zdarma.

Při nákupu WiFi karty je důležité vybírat karty s čipem Atheros – mají nejlepší podporu v Linuxu a ušetříme si spoustu možných potíží.

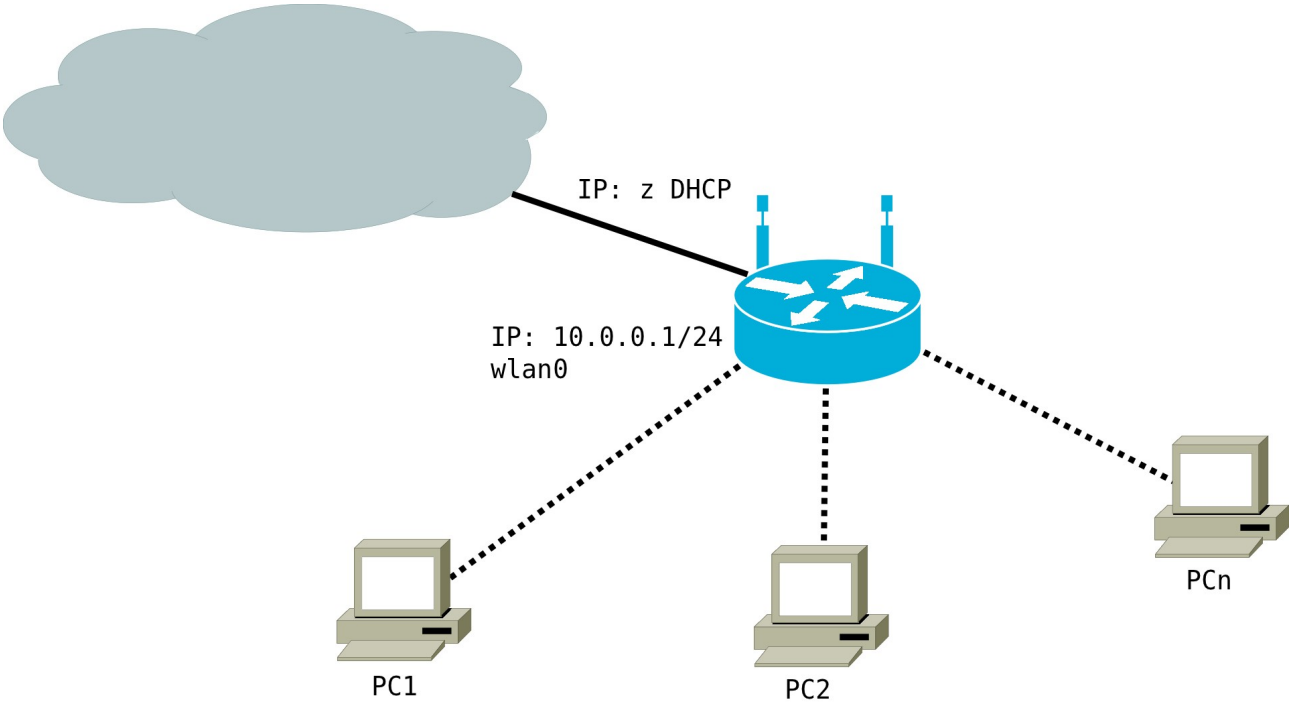
Použité šifrování je vždy nejlepší volit WPA2, není-li to kvůli staří hardware možné, tak za cenu nižší bezpečnosti WPA. Dnes už nikdy nepoužíváme WEP! Je totiž snadno prolomitelný během pár minut.

Interakci mezi klientem a serverem zpracovává program jménem wpa_supplicant. Ten je ve všech nových linuxových distribucích. Mac OS X a Windows mají své vlastní suplikanty.

Téma 12: WiFi s PSK a EAP v CentOS

# Zadání cvičení

1. Nastavte DHCP, rozsah přidělovaných adres bude 10.0.0.50 – 10.0.0.150/24
2. Nastavte NAT
3. Vytvořte WiFi přístupový bod se společným heslem PSK
4. Vytvořte WiFi přístupový bod s WPA-EAP zabezpečením, nastavte FreeRADIUS
   4.1. Vytvořte nového uživatele, pro ověření použijte jméno a heslo
   4.2. Vytvořte nového uživatele, pro ověření použijte SSL certifikáty
5. Zajistěte automatické spuštění všech potřebných služeb po startu po startu systému



Téma 12: WiFi s PSK a EAP v CentOS

# Řešení

## *Zjištění dostupných WiFi karet*

Nejdřív si zjistíme, jestli máme dostupnou nějakou WiFi kartu – **iwconfig** :

```
[root@localhost pokus]# iwconfig
lo        no wireless extensions.

wlan0     IEEE 802.11bg  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
          Retry  long limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off

eth0      no wireless extensions.

eth1      no wireless extensions.

[root@localhost pokus]#
```

Můžeme se také příkazem **lspci | grep "Network\|Ethernet"** podívat, jaký síťový hardware je v PC:

```
[root@localhost pokus]# lspci | grep "Network\|Ethernet"
02:03.0 Ethernet controller: Atheros Communications Inc. AR2417 Wireless Network Adapter [AR5007G 802.11bg] (rev 01)
02:04.0 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL-8139/8139C/8139C+ (rev 10)
02:0b.0 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL-8139/8139C/8139C+ (rev 10)
[root@localhost pokus]#
```

Téma 12: WiFi s PSK a EAP v CentOS

## *Nastavení IP adres*

Síťové kartě připojené do Internetu (eth0) nastavíme IP adresu z DHCP a WiFi kartě (wlan0) nastavíme IP adresu staticky. Ty musí být uloženy v adresáři **/etc/sysconfig/network-scripts/** a musí mít jméno **ifcfg-JmenoSitoveKarty**, tedy např. **ifcfg-wan0** :

```
[root@localhost /]# cat /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=dhcp
[root@localhost /]# cat /etc/sysconfig/network-scripts/ifcfg-wlan0
DEVICE=wlan0
IPADDR=10.0.0.1
PREFIX=24
ONBOOT=yes
BOOTPROTO=none
[root@localhost /]#
```

- **DEVICE** – jméno fyzického zařízení
- **BOOTPROTO** – **dhcp** = získat nastavení z DHCP, **none** = ruční nastavení
- **IPADDR** – IP adresa
- **PREFIX** – místo prefixu je možné použít „**NETMASK=255.255.255.0**“
- **ONBOOT** – aktivovat zařízení při bootování

Na adrese https://www.centos.org/docs/2/rhl-rg-en-7.2/ch-networkscripts.html je k dispozici podrobný popis.

Zakážeme **NetworkManager** a povolíme **network**:

```
[root@localhost /]# chkconfig NetworkManager off
[root@localhost /]# chkconfig network on
[root@localhost /]#
```

Téma 12: WiFi s PSK a EAP v CentOS

## *DHCP*

Abychom nemuseli na všech klientských počítačích nastavovat IP adresy, nainstalujeme a nastavíme DHCP server. Výchozí dnsmasq vypneme a nainstalujeme a povolíme ISC DHCPD:

```
[root@localhost /]# yum install dhcp
[root@localhost /]# chkconfig dnsmasq off
[root@localhost /]# chkconfig dhcpd on
```

V souboru **/etc/sysconfig/dhcpd** nastavíme parametrem **DHCPDARGS** síťové rozhraní, na kterém bude DHCP server poslouchat:

```
[root@localhost /]# cat /etc/sysconfig/dhcpd
# Command line options here
DHCPDARGS=wlan0
[root@localhost /]#
```

Samotné nastavení je v souboru **/etc/dhcp/dhcpd.conf** :

```
[root@localhost /]# cat /etc/dhcp/dhcpd.conf
max-lease-time 7200;
default-lease-time 3600;

subnet 10.0.0.0 netmask 255.255.255.0 {
    option domain-name-servers 8.8.8.8, 8.8.4.4;
    option domain-name "test";
    option broadcast-address 10.0.0.255;
    option routers 10.0.0.1;
    range 10.0.0.50 10.0.0.150;
    authoritative;

#    host jmeno {
#      hardware ethernet 1a:2b:3c:4d:5e:6f;
#      fixed-address 10.0.0.70;
#    }
}
[root@localhost /]#
```

V případě potřeby můžeme určitému počítači nastavit přidělení vždy stejné IP adresy (podle MAC adresy).

Téma 12: WiFi s PSK a EAP v CentOS

## *Firewall*

Nastavíme jednoduchý firewall. Aby bylo možné použít počítač jako router, je nutné povolit předávání paketů. V souboru **/etc/sysctl.conf** změníme net.ipv4.ip_forward=0 na **net.ipv4.ip_forward=1** . A znovu načteme nastavení příkazem **sysctl -p** .
Spouštění firewallu zajistíme init skriptem. Vytvoříme soubor **/etc/rc.d/init.d/firewall** s následujícím obsahem:

```
#!/bin/sh
#
# firewall   Start iptables firewall
#
# chkconfig: 2345 11 99
# description:      Start iptables firewall
#
# config: /usr/local/bin/firewall.sh
#
### BEGIN INIT INFO
# Provides: firewall
# Required-Start: $network
# Required-Stop:
# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
# Short-Description: start and stop iptables firewall
# Description: Start, stop and save iptables firewall
### END INIT INFO

# Source function library.
. /etc/init.d/functions

# only usable for root
[ $EUID = 0 ] || exit 4

case "$1" in
    start)
       /usr/local/bin/firewall.sh
       ;;
    status)
       echo -e "--------------------------------\nROUTE:\n"
       route -n
       echo -e "\n--------------------------------\nIPTABLES:\n"
       iptables -L -n -v
       ;;
```

```
    *)
       echo "Pouzijte 'start', nebo 'status'"
       ;;
esac
exit
```

A skript s firewallem uložíme do **/usr/local/bin/firewall.sh** :

```
#!/bin/bash

iptables -F
iptables -t nat -F
iptables -t mangle -F
iptables -X

iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT

iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT

iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

iptables -A INPUT -p icmp -j ACCEPT
iptables -A FORWARD -p icmp -j ACCEPT

iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

iptables -A INPUT -m state --state INVALID -j DROP

iptables -A INPUT -j REJECT --reject-with icmp-admin-prohibited
```

Přidáme právo spuštění pro **/usr/local/bin/firewall.sh** a **/etc/rc.d/init.d/firewall** a zapneme init skript **chkconfig firewall on**:

```
[root@localhost /]# chmod +x /usr/local/bin/firewall.sh
[root@localhost /]# chmod +x /etc/rc.d/init.d/firewall
[root@localhost /]# chkconfig firewall on
[root@localhost /]#
```

Téma 12: WiFi s PSK a EAP v CentOS

## *hostapd*

Abychom mohli vytvořit WiFi přístupový bod, budeme potřebovat program **hostapd**. Protože v CentOS hostapd není, budeme si muset stáhnout zdrojové kódy z oficiálních stránek http://hostap.epitest.fi/hostapd/ a zkompilovat je.

Nejdřív si nainstalujeme všechny potřebné balíčky, bez kterých se nám hostapd nepodaří zkompilovat:

```
[root@jlocalhost /]# yum groupinstall "Development Tools"
[root@jlocalhost /]# yum install libnl-devel
[root@jlocalhost /]# yum install openssl-devel
```

poté přejdeme do adresáře **/usr/local/src** , stáhneme zdrojové kódy a rozbalíme je:

```
[root@localhost src]# wget http://hostap.epitest.fi/releases/hostapd-0.7.3.tar.gz
[root@localhost src]# tar xzvf hostapd-0.7.3.tar.gz
```

v rozbaleném adresáři přejdeme do adresáře **hostapd** a soubor **defconfig** zkopírujeme do **.config** :

```
[root@localhost src]# cd hostapd-0.7.3/hostapd
[root@localhost hostapd]# cp defconfig .config
```

Nyní povolíme mac80211[1] ovladač pro Atheros karty. V první části souboru **.config** najdeme řádek **#CONFIG_DRIVER_NL80211=y** a odkomentujeme ho:

```
# Driver interface for drivers using the nl80211 kernel interface
CONFIG_DRIVER_NL80211=y
```

to je vše a můžeme spustit kompilaci a instalaci:

```
[root@localhost hostapd]# make
[root@localhost hostapd]# make install
```

Protože se nenainstaluje init skript, vytvoříme si vlastní. Do souboru **/etc/rc.d/init.d/hostapd** vložíme následující kód:

```
#!/bin/sh
#
# start/stop the hostapd server
#
```

1 http://linuxwireless.org/en/developers/Documentation/mac80211

Téma 12: WiFi s PSK a EAP v CentOS

```
# chkconfig: 2345 64 10
# description: hostap daemon
# processname: hostapd
# config: /etc/hostapd.conf
# pidfile: /var/run/hostapd.pid
#
PATH=/bin:/usr/bin:/usr/local/bin:/sbin:/usr/sbin
export PATH

# Source function library.
. /etc/rc.d/init.d/functions

stop()
{
 echo -n "Stopping hostapd daemon: "
        killproc hostapd
        echo
        rm -f /var/lock/subsys/hostapd
}

start()
{
        echo -n "Starting hostapd daemon: "
        daemon /usr/local/bin/hostapd /etc/hostapd.conf -P /var/run/hostapd.pid -B
        echo
        touch /var/lock/subsys/hostapd
}

case "$1" in
    start)
        start
        ;;
    stop)
        stop
        ;;
    status)
        status hostapd
        ;;
    restart)
        stop
        start
        ;;
    *)
        echo "Usage: hostapd {start|stop|status|restart}"
```

```
        exit 1
esac
exit 0
```

a nastavíme automatické spouštění:

```
[root@localhost hostapd]# chkconfig hostapd on
```

Hostapd je nutné spouštět před startem DHCP serveru!

```
[root@localhost rc2.d]# ls -l | grep "hostapd\|dhcp"
lrwxrwxrwx. 1 root root 17 26. bře 15.40 S64hostapd -> ../init.d/hostapd
lrwxrwxrwx. 1 root root 15 26. bře 15.45 S65dhcpd -> ../init.d/dhcpd
```

Téma 12: WiFi s PSK a EAP v CentOS

## *WiFi se společným heslem*

Nastavení přístupového bodu se společným heslem (PSK) je velice jednoduché. Nastavení pro hostapd umístíme do souboru **/etc/hostapd.conf** :

```
[root@localhost /]# cat /etc/hostapd.conf
interface=wlan0
driver=nl80211

wpa=2
  #1 = WPA, 2 = WPA2, 3 = oboji

wpa_key_mgmt=WPA-PSK

wpa_pairwise=CCMP
  #Do wpa_pairwise lze zadat TKIP nebo CCMP nebo oboji (wpa_pairwise=TKIP CCMP)

hw_mode=g
channel=13

ssid=CentOS
#psk="tajneHeslo123"
wpa_psk=e5db6c86a4a7d0988507a9b2dc85d8df8ca19302644be532d7b76d38485230ee
```

Nejbezpečnější je nastavit pouze WPA2[2]: **wpa=2** a **wpa_pairwise=CCMP**, ale velmi stará WiFi zařízení se nebudou umět připojit.

Pro zajištění kompatibility (za cenu nižší bezpečnosti) lze nastavit **wpa=3** a **wpa_pairwise=TKIP CCMP**.

Řádek **wpa_psk** si vygenerujeme programem **wpa_passphrase** :

```
[root@localhost /]# wpa_passphrase
usage: wpa_passphrase <ssid> [passphrase]

If passphrase is left out, it will be read from stdin
[root@localhost /]#
[root@localhost /]# wpa_passphrase CentOS tajneHeslo123
network={
        ssid="CentOS"
        #psk="tajneHeslo123"
        psk=e5db6c86a4a7d0988507a9b2dc85d8df8ca19302644be532d7b76d38485230ee
}
[root@localhost /]#
```

Vygenerované PSK je závislé na SSID!

2 WPA2 - http://cs.wikipedia.org/wiki/IEEE_802.11i

Téma 12: WiFi s PSK a EAP v CentOS

## *WiFi s 802.1X a FreeRADIUS*

FreeRADIUS je dostupný ve zdrojích distribuce, takže stačí nainstalovat příslušné balíčky:

```
[root@localhost /]# yum install freeradius freeradius-utils
```

Veškeré nastavení FreeRADIUSu je umístěno v adresáři **/etc/raddb/** .

Nejdřív je nutné nastavit soubor clients.conf. V něm nastavíme radius klienty, jako hostapd, nebo různé WiFi krabičky.

```
[root@localhost raddb]# cat clients.conf
client localhost {
       ipaddr       = 127.0.0.1
       secret       = testing123
       require_message_authenticator = no
       nastype     = other # localhost isn't usually a NAS...
#client 192.168.0.0/24 {
#      secret       = testing123-1
#      shortname    = private-network-1
#}
[root@localhost raddb]#
```

Sekce localhost bude sloužit pro připojení hostapd. Samozřejmě je vhodné zvolit lepší heslo.

Místo jednotlivých počítačů je také možné specifikovat celou síť.

Je více způsobů, jak lze mít uloženou databázi s uživateli. Nejjednodušší je použít soubor users :

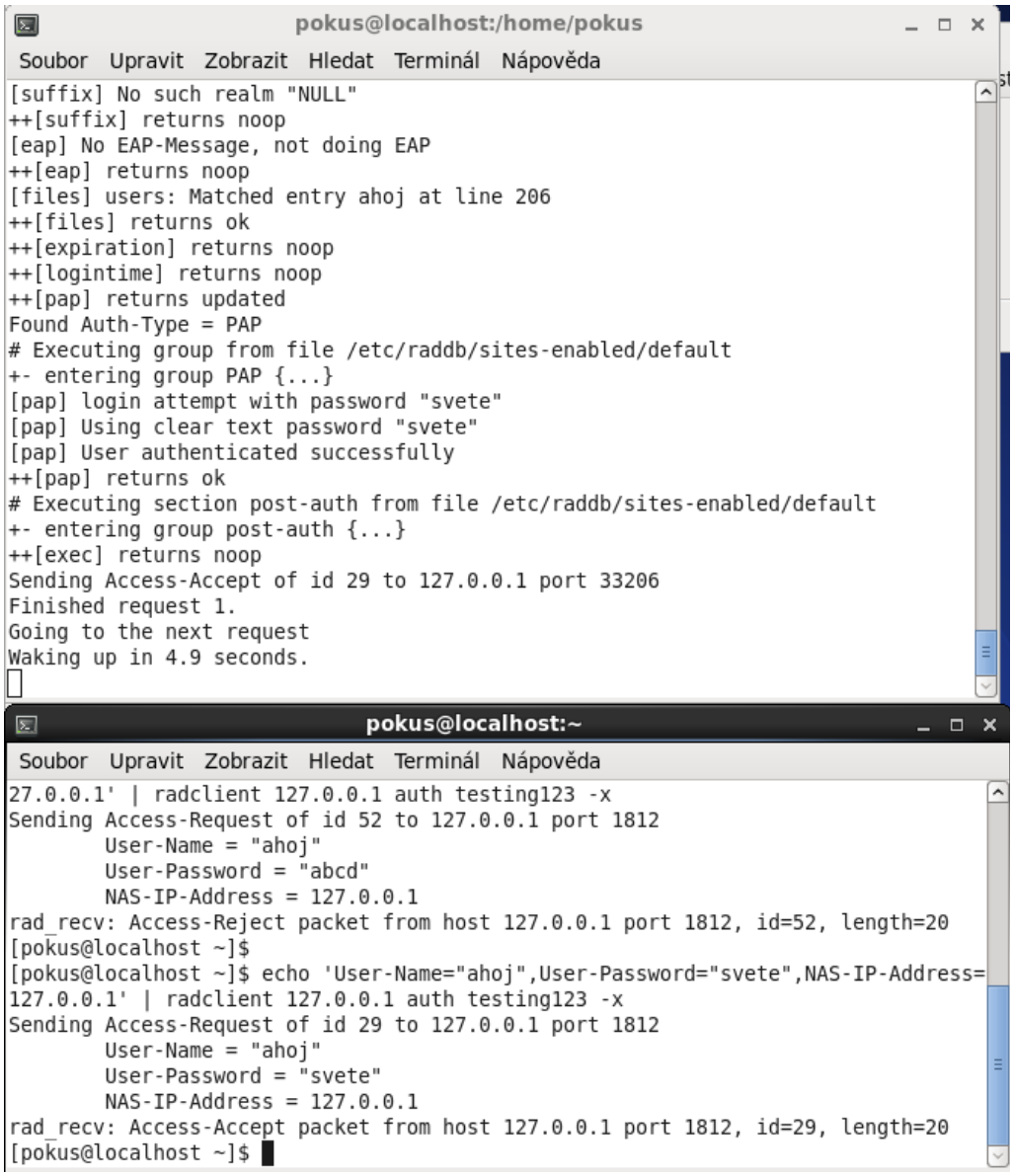
```
[root@localhost raddb]# cat users
ahoj   Cleartext-Password  :=     "svete"
[root@localhost raddb]#
```

Na konec přidáme záznam pro nového uživatele se jménem **ahoj** a heslem **svete**.

Nyní si spustíme FreeRADIUS v debug módu **radiusd -X** a provedeme test:

Téma 12: WiFi s PSK a EAP v CentOS

```
[root@localhost raddb]# echo 'User-Name="ahoj",User-Password="svete",NAS-IP-
Address=127.0.0.1' | radclient 127.0.0.1 auth testing123 -x
```




Aby hostapd využíval radius, musíme ho k tomu nastavit. Opět v souboru **/etc/hostapd.conf** :

```
interface=wlan0
driver=nl80211

wpa=2
wpa_key_mgmt=WPA-EAP
wpa_pairwise=CCMP
hw_mode=g
channel=13
ssid=CentOS_802.1x

ieee8021x=1
eap_server=0
```

# Téma 12: WiFi s PSK a EAP v CentOS

```
own_ip_addr=127.0.0.1
auth_server_addr=127.0.0.1
auth_server_port=1812
auth_server_shared_secret=testing123
eapol_key_index_workaround=1
```

Téma 12: WiFi s PSK a EAP v CentOS

Pro otestování spustíme radiusd a hostapd v debug módu:

**radiusd -X** a **hostapd -dd /etc/hostpad.conf**

Vidíme tak celý průběh připojování klienta a případnou chybu:

## Nastavení certifikátů

Nejbezpečnější ověření uživatele je při použití certifikátů. Uživatel má certifikát certifikační autority a svůj vlastní certifikát s klíčem, ten je navíc chráněný heslem.

Abychom mohli vydávat uživatelům certifikáty, musíme si vytvořit certifikační autoritu. Použijeme tu před-připravenou v adresáři **/etc/raddb/certs/** .

Ihned po instalaci nejdeme v tomto adresáři nějaké hotové certifikáty, ty jsou však vydané na fiktivní jméno. Proto si vytvoříme vlastní autoritu.

Nejdřív smažeme všechny staré, pro nás nepotřebné soubory:

```
[root@localhost certs]# cd /etc/raddb/certs
[root@localhost certs]# rm -f *.pem *.der *.csr *.crt *.key *.p12 serial* index.txt*
[root@localhost certs]# ls
bootstrap  ca.cnf  client.cnf  dh  Makefile  random  README  server.cnf  xpextensions
[root@localhost certs]#
```

Pak upravíme soubor **ca.cnf** – jde o nastavení, podle kterého bude OpenSSL generovat certifikáty:

```
[root@localhost certs]# cat ca.cnf
[ ca ]
default_ca          = CA_default

[ CA_default ]
dir                 = ./
certs               = $dir
crl_dir                   = $dir/crl
database            = $dir/index.txt
new_certs_dir       = $dir
certificate         = $dir/ca.pem
serial              = $dir/serial
crl                 = $dir/crl.pem
private_key         = $dir/ca.key
RANDFILE            = $dir/.rand
name_opt            = ca_default
cert_opt            = ca_default
default_days        = 60
default_crl_days    = 30
default_md          = sha1
preserve            = no
policy              = policy_match
```

```
[ policy_match ]
countryName         = match
stateOrProvinceName = match
organizationName    = match
organizationalUnitName    = optional
commonName          = supplied
emailAddress        = optional

[ policy_anything ]
countryName         = optional
stateOrProvinceName = optional
localityName        = optional
organizationName    = optional
organizationalUnitName    = optional
commonName          = supplied
emailAddress        = optional

[ req ]
prompt              = no
distinguished_name  = certificate_authority
default_bits        = 2048
input_password            = pokus
output_password           = pokus
x509_extensions           = v3_ca

[certificate_authority]
countryName         = CZ
stateOrProvinceName = Czech Republic
localityName        = Hradec Kralove
organizationName    = Pokus
emailAddress        = admin@example.com
commonName          = "Pokus CA"

[v3_ca]
subjectKeyIdentifier      = hash
authorityKeyIdentifier    = keyid:always,issuer:always
basicConstraints    = CA:true
```

Důležité je nastavit heslo a informace o autoritě (organizationName musí být stejné i u ostatních certifikátů!).

Podobně upravíme nastavení pro serverový certifikát:

```
[root@localhost certs]# cat server.cnf
```

```
[ ca ]
default_ca          = CA_default

[ CA_default ]
dir                 = ./
certs               = $dir
crl_dir                   = $dir/crl
database            = $dir/index.txt
new_certs_dir       = $dir
certificate         = $dir/server.pem
serial              = $dir/serial
crl                 = $dir/crl.pem
private_key         = $dir/server.key
RANDFILE            = $dir/.rand
name_opt            = ca_default
cert_opt            = ca_default
default_days        = 60
default_crl_days    = 30
default_md          = sha1
preserve            = no
policy              = policy_match

[ policy_match ]
countryName         = match
stateOrProvinceName = match
organizationName    = match
organizationalUnitName    = optional
commonName          = supplied
emailAddress        = optional

[ policy_anything ]
countryName         = optional
stateOrProvinceName = optional
localityName        = optional
organizationName    = optional
organizationalUnitName    = optional
commonName          = supplied
emailAddress        = optional

[ req ]
prompt              = no
distinguished_name  = server
default_bits        = 2048
input_password            = pokus
```

```
output_password             = pokus

[server]
countryName             = CZ
stateOrProvinceName = Czech Republic
localityName            = Hradec Kralove
organizationName    = Pokus
emailAddress            = freeradius@example.com
commonName              = "RADIUS server"
```

A spustíme bootstrap skript, který vytvoří vše potřebné:

```
[root@localhost certs]# ./bootstrap
```

Aby FreeRADIUS mohl pracovat s certifikátem, musíme mu sdělit heslo. V souboru **/etc/raddb/eap.conf** upravíme v sekci **tls** položku **private_key_password**:

```
[root@localhost /]# cat /etc/raddb/eap.conf
...
       tls {
              ...
              private_key_password = pokus
              ...
```

Cesty k certifikátům nemusíme měnit.

Na závěr vygenerujeme certifikáty pro klienty. Vrátíme se do adresáře **/etc/raddb/certs** a tentokrát upravíme soubor **client.cnf**:

```
[root@localhost certs]# cat client.cnf
[ ca ]
default_ca          = CA_default

[ CA_default ]
dir                 = ./
certs               = $dir
crl_dir             = $dir/crl
database            = $dir/index.txt
new_certs_dir       = $dir
certificate         = $dir/server.pem
serial              = $dir/serial
crl                 = $dir/crl.pem
private_key         = $dir/server.key
```

```
RANDFILE            = $dir/.rand
name_opt            = ca_default
cert_opt            = ca_default
default_days        = 60
default_crl_days    = 30
default_md          = sha1
preserve            = no
policy              = policy_match

[ policy_match ]
countryName         = match
stateOrProvinceName = match
organizationName    = match
organizationalUnitName    = optional
commonName          = supplied
emailAddress        = optional

[ policy_anything ]
countryName         = optional
stateOrProvinceName = optional
localityName        = optional
organizationName    = optional
organizationalUnitName    = optional
commonName          = supplied
emailAddress        = optional

[ req ]
prompt              = no
distinguished_name  = client
default_bits        = 2048
input_password             = pokus
output_password            = pokus

[client]
countryName         = CZ
stateOrProvinceName = Czech Republic
localityName        = Hradec Kralove
organizationName    = Pokus
emailAddress        = uzivatel.DVA@example.com
commonName          = uzivatel.DVA@example.com
```

zajímají nás pouze tučně zvýrazněné řádky. Vytvoříme klientovi certifikát:

```
[root@localhost certs]# make client.pem
```

tím vznikne soubor **uzivatel.DVA@example.com.pem** (podle hodnoty emailAddress), ten spolu se souborem **ca.pem** předáme uživateli.


Když budeme generovat certifikát pro dalšího uživatele, upravíme v souboru **client.cnf** tučně označené řádky a znovu spustíme **# make client.pem**.

Nakonec nastavíme práva pro certifikáty, aby se k nim radius dostal a můžeme se z klienta připojit:

```
[root@localhost certs]# chown root:radiusd *
```

### *Připojení z klientského počítače*

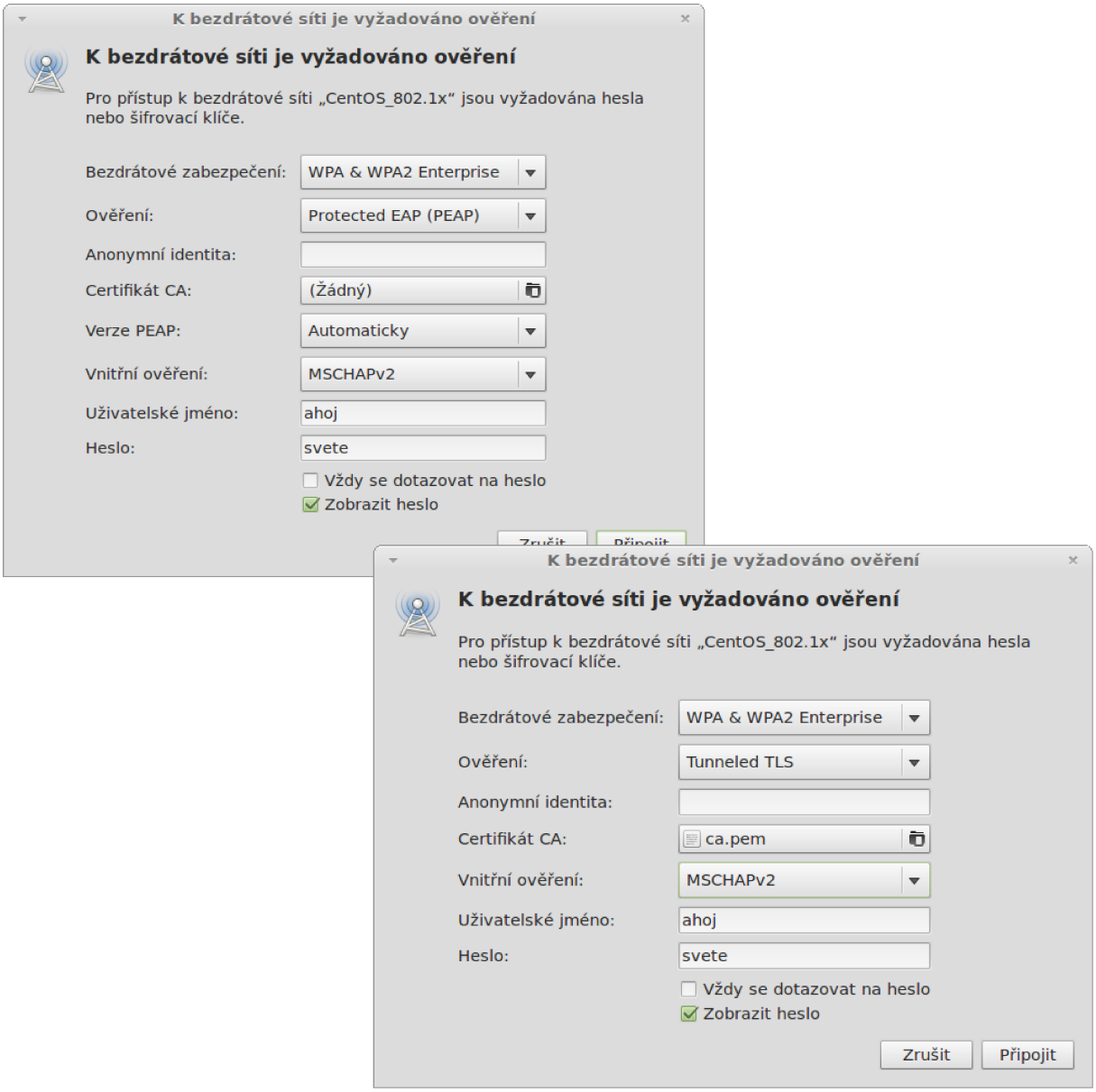
Při použití společného hesla stačí na klientském počítači vybrat WiFi síť a při připojení zadat heslo.

Při použití radiusu je to složitější.

## Připojení v Linuxu

V appletu Network Manageru vybereme síť a připojíme se.

Použijeme PEAP, nebo Tunneled TLS[3]:

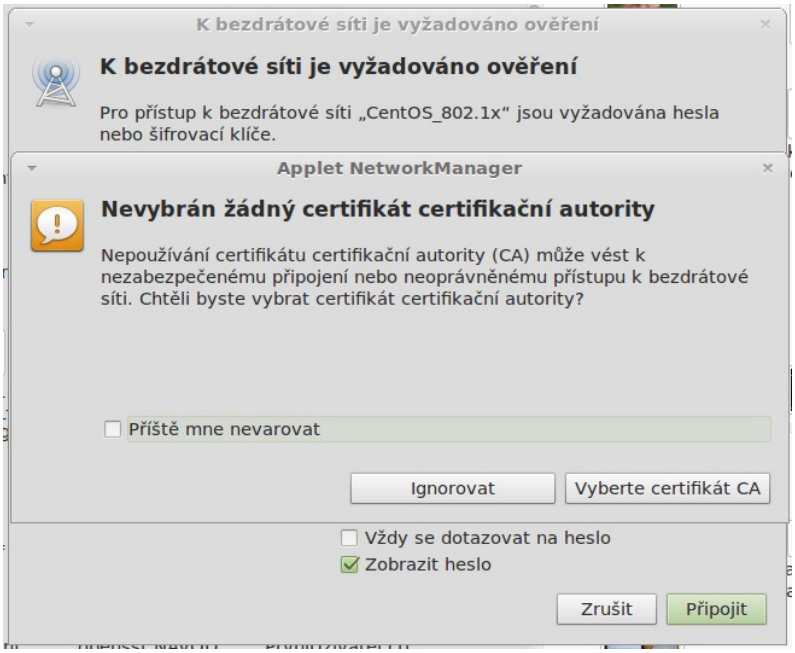



Máme dvě možnosti, použít certifikát certifikační autority (není to bezpečné, ale nemusíme klientům dávat žádný soubor), nebo použít certifikát CA, jméno a heslo.

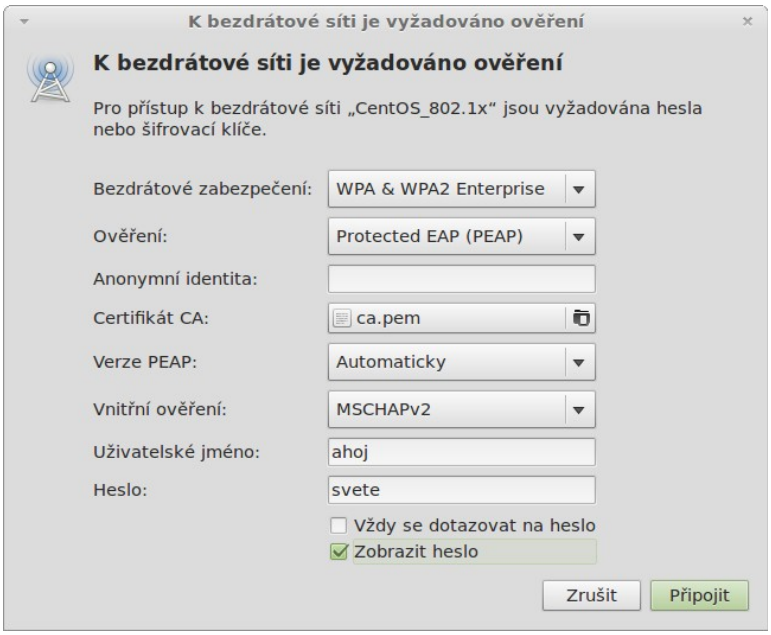
3 Porovnání PEAP a TTLS: http://www.opus1.com/www/whitepapers/ttlsandpeap.pdf

Téma 12: WiFi s PSK a EAP v CentOS

Certifikát certifikační autority, který budeme rozdávat uživatelům zkopírujeme z **/etc/radddb/certs/ca.pem**



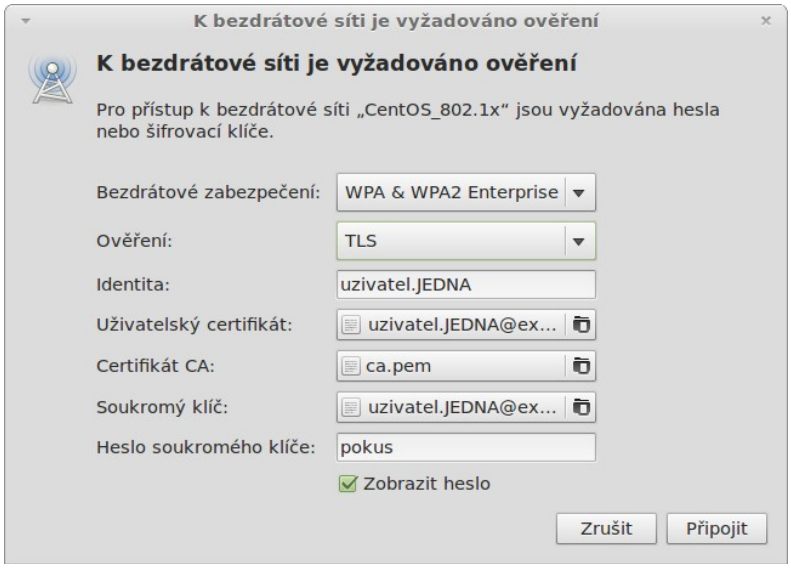

Použití certifikátu CA:




Po úspěšném připojení automaticky dostaneme IP adresy:

Téma 12: WiFi s PSK a EAP v CentOS


Připojení s certifikáty nastavíme skoro stejně:




## Připojení z Windows

Připojení z Windows je mnohem složitější, než z Linuxu.

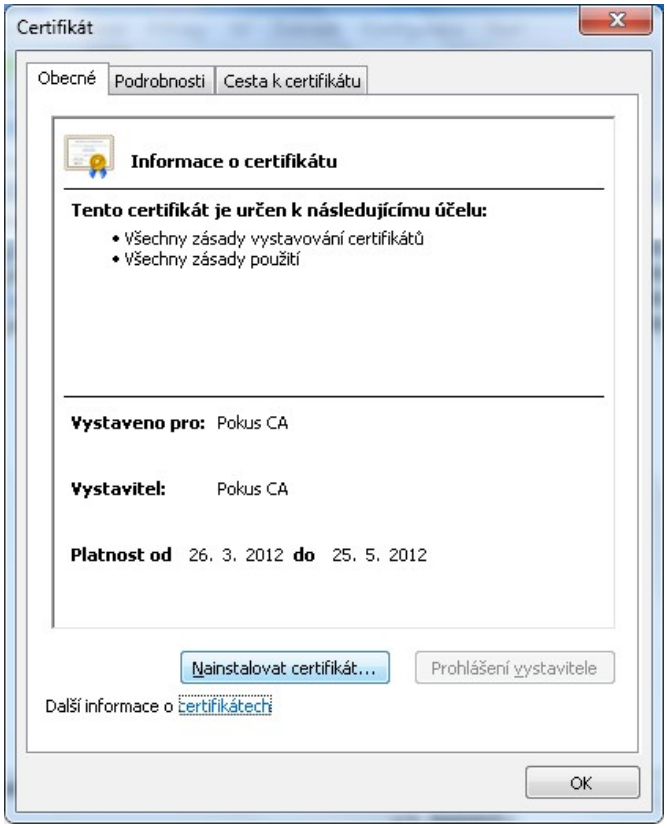
Postup by měl být skoro stejný, jako při připojení k *eduroam* -

http://www.eduroam.cz/doku.php?id=cs:uzivatel:sw:uvod

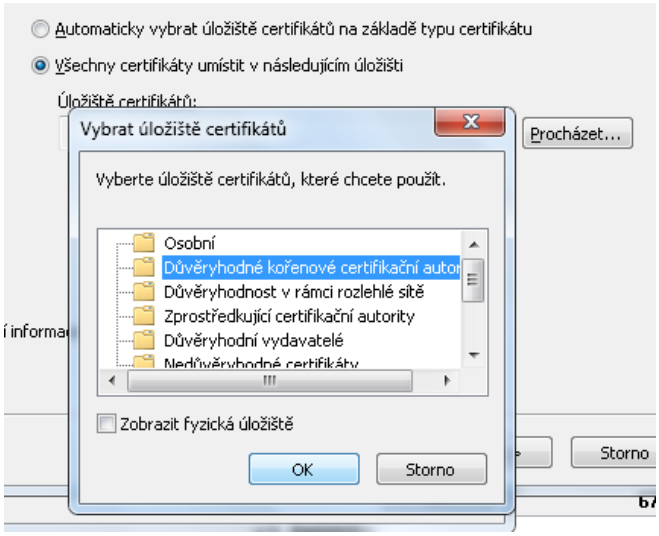
Pro Windows 7 - http://www.eduroam.cz/doku.php?id=cs:uzivatel:sw:win:seven

Téma 12: WiFi s PSK a EAP v CentOS

1. Certifikát CA si přejmenujeme z ca.pem na **ca.crt**
2. Otevřeme ho a zvolíme instalaci

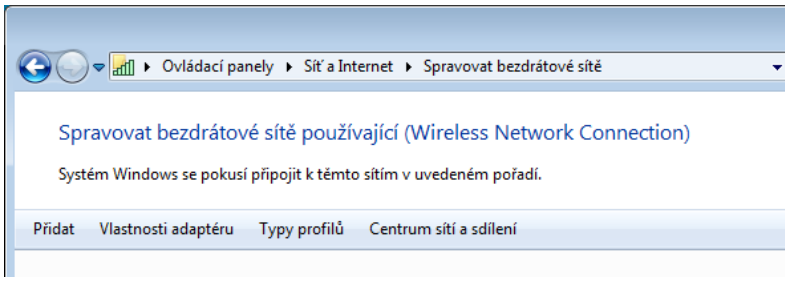



3. Úložiště vybereme **„Důvěryhodné kořenové a certifikační autority“**

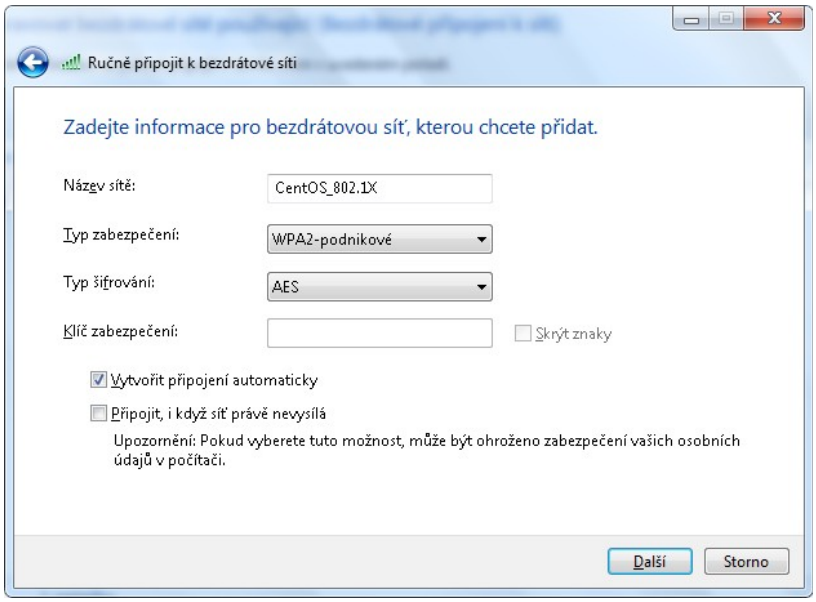



4. Přejdeme v Ovládacích panelech na nastavení bezdrátových sítí a vybereme Přidat

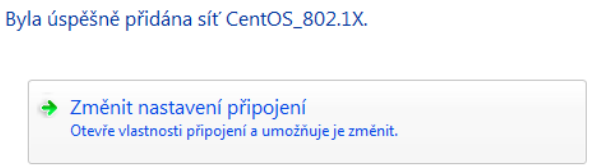
Téma 12: WiFi s PSK a EAP v CentOS



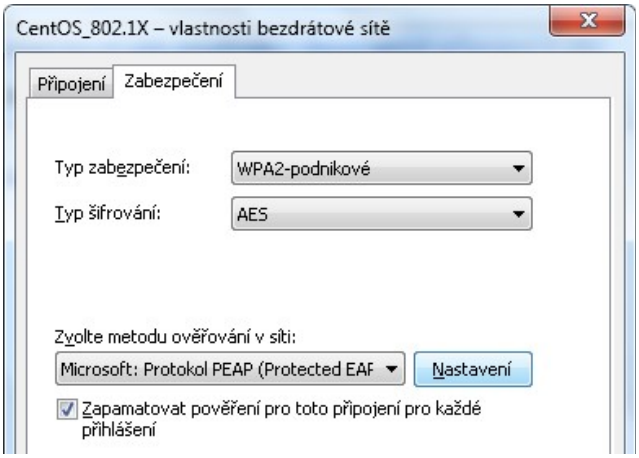

5. Typ zabezpečení bude WPA2-podnikové




6. Změnit nastavení

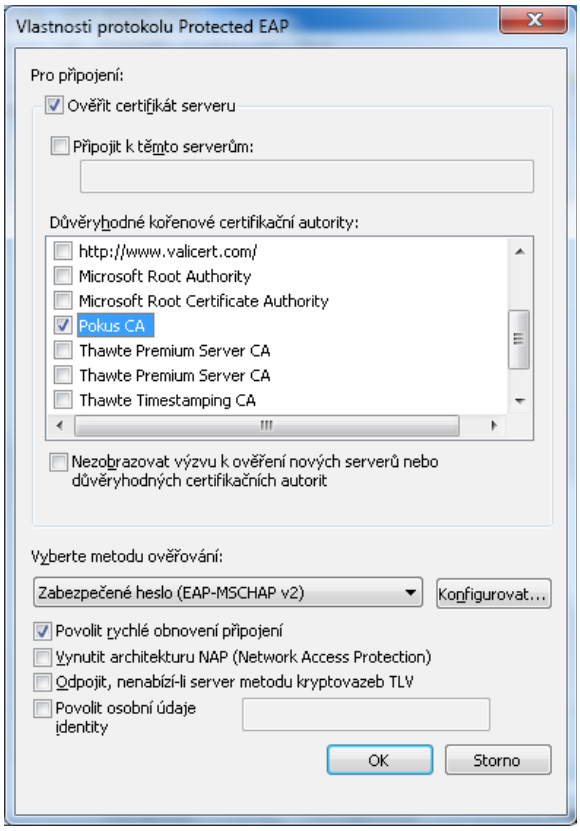



7. Metodu ověřování dáme PEAP

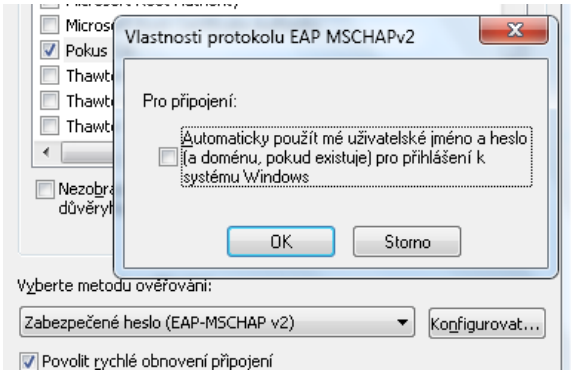



8. Vybereme naši CA a v Konfigurovat

Téma 12: WiFi s PSK a EAP v CentOS

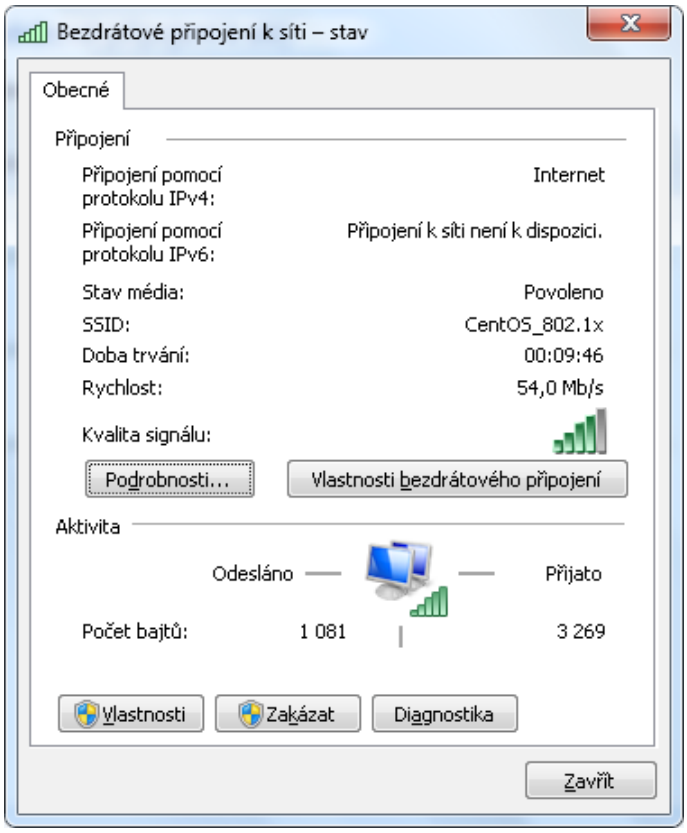



9. Odškrtneme použití jména z Windows




10. Nyní se pomocí ikony u hodin připojíme a po zadání jména a hesla máme funkční síť

Téma 12: WiFi s PSK a EAP v CentOS




Nastavení pro běžného uživatele je velmi zdlouhavé a obtížné. Lze však zautomatizovat.

### *Automatické nastavení připojení ve Windows*

1. Nastavíme (v Ovládacích panelech) připojení k WiFi síti, ale **nevyplníme jméno, ani heslo**
2. Exportujeme[4] nastavení WiFi do XML souboru příkazem:

```
netsh wlan export profile name="SSID"
```

Tím máme vytvořené nastavení.

K importování nastavení u klienta musíme nejdřív importovat certifikát certifikační autority.

V systému Windows Vista a 7 lze certifikát jednoduše importovat:

```
certutil -addstore -f Root "certifikat.crt"
```

dojde k automatickému importu, uživatel o ničem neví.

4 http://sudonetworks.com/wiki/index.php?title=Windows_Wireless_Control_with_netsh
http://msdn.microsoft.com/en-us/library/aa369853(v=vs.85).aspx

Téma 12: WiFi s PSK a EAP v CentOS

Ve Windows XP nelze certutil použít. Je potřeba využít:

```
rundll32.exe cryptext.dll,CryptExtAddCER certifikat.crt
```

tím dojde ke spuštění průvodce importem certifikátu. Uživatel jenom několikrát klikne na tlačítko další. Jiná možnost v systému Windows XP asi není.

Poté můžeme načíst nastavení z XML souboru:

```
netsh wlan add profile filename="sitovy_profil.xml"
```

při prvním připojení by uživateli mělo stačit už jenom zadat své jméno a heslo.
Je otázkou několika minut vytvořit jednoduchý program ve formě jednoho malého exe souboru, který koncovému uživateli automaticky vše nastaví.