VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

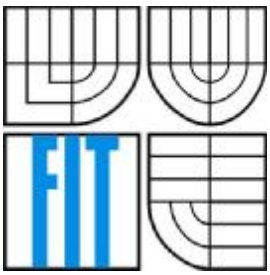
METODY AUTENTIZACE NAPOJENÍ K WIFI SÍTI

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE JAN SEDLÁŘ
AUTHOR

BRNO 2007

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

METODY AUTENTIZACE NAPOJENÍ K WIFI SÍŤI

METODS OF AUTHENTICATION TO WIFI NETWORK

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — JAN SEDLÁŘ
AUTHOR

VEDOUCÍ PRÁCE — Ing. Petr Lampa
SUPERVISOR

BRNO 2007

Vysoké učení technické v Brně - Fakulta informačních technologií

Centrum výpočetní techniky Akademický rok 2006/2007

Zadání diplomové práce

Řešitel: **Sedlář Jan**
Obor: Výpočetní technika a informatika
Téma: **Metody autentizace napojení k WiFi síti**
Kategorie: Počítačové sítě

Pokyny:

1. Prostudujte způsoby autentizace klientů na úrovni L2 k počítačové síti typu WiFi dle standardu IEEE 802.1x.
2. Seznamte se s dostupnými implementacemi Radius serveru a způsoby jejich konfigurace.
3. Otestujte funkčnost a spolehlivost implementací autentizace klientů v různých operačních systémech (WinXP, Linux, BSD) ve spojení se zvolenými servery Radius a WiFi AP HP, Asus, Avaya a Cisco.
4. Otestujte vliv použitého šifrování a rotace klíčů na přenosovou rychlost.
5. Zvolte vhodnou kombinaci autentizace a Radius serveru, vypracujte nástroje pro konfiguraci, správu a účtování přenesených dat včetně uživatelské dokumentace.
6. Zhodnoťte navržený nástroj v porovnání s dostupnými aplikacemi.

Literatura:

1. http://www.open1x.org/
2. http://www.freeradius.org/
3. http://www.ieee802.org/1/pages/802.1X-rev.html
4. http://www.eduroam.cz/cz/index.html

Při obhajobě semestrální části diplomového projektu je požadováno:

- První tři body zadání.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci ročníkového a semestrálního projektu (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním paměťovém médiu (disketa, CD-ROM), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Lampa Petr, Ing.**, CVT FIT VUT
Datum zadání: 1. listopadu 2006
Datum odevzdání: 22. května 2007

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
612 66 Brno, Božetěchova 2

L.S.

doc. Ing. Jaroslav Zendulka, CSc.
vedoucí ústavu

LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami

1. Pan

Jméno a příjmení: **Jan Sedlář**
Id studenta: 22377
Bytem: Na vyhlídce 347, 742 42 Šenov u Nového Jičína
Narozen: 20. 06. 1982, Zlín
(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta informačních technologií
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....
(dále jen "nabyvatel")

Článek 1
Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
diplomová práce

Název VŠKP: Metody autentizace napojení k WiFi síti
Vedoucí/školitel VŠKP: Lampa Petr, Ing.
Ústav: Centrum výpočetní techniky
Datum obhajoby VŠKP: .....

VŠKP odevzdal autor nabyvateli v:
tištěné formě počet exemplářů: 1
elektronické formě počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2
Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
   - ☐ ihned po uzavření této smlouvy
   - ☐ 1 rok po uzavření této smlouvy
   - ☐ 3 roky po uzavření této smlouvy
   - ☐ 5 let po uzavření této smlouvy
   - ☒ 10 let po uzavření této smlouvy
   - (z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3
Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísni a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: .....

....

Nabyvatel Autor

Abstrakt

Tato práce zachycuje fyzikální vlastnosti bezdrátových sítí a principy autentizace klientů, které se od těchto vlastností odvíjí. Dále pak standard 802.1x a jeho principy, využití serveru RADIUS a jeho implementaci a návrh informačního systému na správu uživatelů a přenesených dat.

Klíčová slova

WiFi, AP, Autentizace, Bezpečnost, RADIUS server, MySQL, Apache, Linux

Abstract

Thist thesis describes the fyzical principles of wireless communication and principles of client authentification. It describes the Standard 802.1x too, RADIUS server use and its implementation and design of information system for user management and traffic accounting .

Keywords

WiFi, AP, Autentization, Security, RADIUS server, MySQL, Apache, Linux.

Metody autentizace napojení k WiFi síti

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Petra Lampy
Další informace mi poskytli Ing. Dulík Tomáš
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Jan Sedlář
21. května 2007

Poděkování

Rád bych poděkoval Ing. Petru Lampovi za počáteční inspiraci a všem, kteří mi vytvořili podmínky pro tvůrčí činnost.

Obsah

1 Úvod

V poslední čtvrtině 20. století došlo ke značnému rozvoji komunikačních technologií, zvláště pak v odvětví počítačové komunikace. Počítačové systémy byli z počátku propojovány do menších a později do větších sítí, které umožňovali mezipočítačovou komunikaci. V dnešní době je největší celosvětovou sítí Internet.

S rostoucím počtem připojených počítačů v sítích a v Internetu, rostl také důraz na ochranu datových toků v síti nebo dat v jednotlivých počítačích. Postupem času se zabezpečení takovýchto sítí stalo nezbytným a jedním ze základních požadavků pro vytváření sítí.

V této práci se budeme zabývat zabezpečením bezdrátových sítí a to zejména způsobem jakým se jednotliví klienti autentizují k dané síti. Bezdrátové sítě v současné době zažívají velkou popularitu. Je to zejména tím, že bezdrátové připojení je pohodlné a „bez drátů“. Tato popularita je také dána nízkou pořizovací cenou jednotlivých zařízení. Lokální sítě dnes již nabízejí kavárny, hotely letiště a další místa, kde mohou návštěvníci použít své přenosné počítače či PDA, přečíst si poštu či surfovat na Internetu.

Ve druhé kapitole si připomeneme základy protokolu 802.11, dle kterých se komunikace v bezdrátovém světě neobejde. V kapitole třetí pak již jsou zahrnuty bezpečnostní prvky bezdrátové komunikace, jednotlivé principy zabezpečení a ověřovacích protokolů. Čtvrtá kapitola nám nastiňuje druhy autentizační nástrojů – serverů, jejich principy a postupný vývoj a vzájemné srovnání těchto systémů. Pátá kapitola zachycuje postup při instalaci RADIUS serveru, nejprve s modulem LDAP. Dále pak s databází MySQL. Šestá kapitola popisuje proprientální mechanismy výrobků CISCO, které patři mezi špičku v síťových technologiích. Sedmá kapitola pak nastiňuje možnost propojení dnes dynamicky se rozvíjejícího operačního systému MIKROTIK a serveru RADIUS, v této kapitole se také zabývám vlivem rotace sdílených klíčů na přenosovou rychlost. V osmé kapitole najdeme návrh Informačního systému pro správu a účtování klientů bezdrátové sítě, také nástin následné implementace.

.

2 Standard 802.11

Bezdrátová síť a standard 802.11 jsou pojmy o kterých v poslední době slyšíme stále častěji. Specifikace 802.11 byla přijata po několika letech diskusí a to v roce 1997. Podporovala přenosové rychlosti 1 nebo 2 megabity za sekundu. Protokol pokrýval první (fyzickou) a druhou (linkovou) vrstvu modelu OSI. Na úrovni druhé vrstvy definoval standard 802.11 tyto služby:
Autentizace a deautentizace
Asociace, disociace reasociace
Privátnost (WEP)
Doručování MSDU (Mac Service Data Unit)

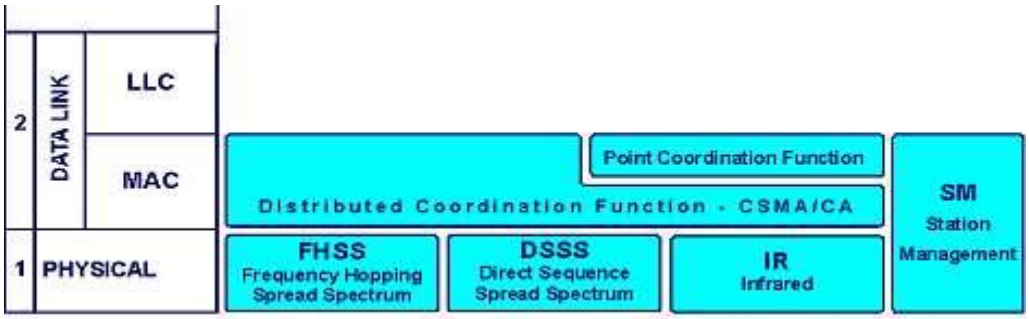
Na fyzické vrstvě pak byly definovány tři metody:

- DSSS (Direkt Sequence Spread Spektrum)
- FHSS (Frequency Hopping Spread Spektrum)
- Infračervený přenos

V roce 1999 byly uvedeny dva vysokorychlostní doplňky. 802.11a a 802.11b. V roce 2003 pak byl přijat doplněk 802.11g. Tyto standardy podporují vyšší rychlost (11 Mb/s – 802.11b, 54Mb/s – 802.11a/g)

2.1 802.11 - fyzická vrstva

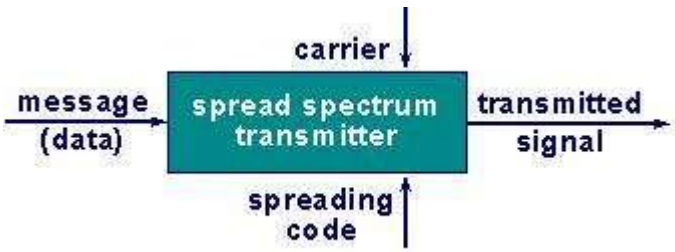
Jako všechny standardy řady 802.x zahrnuje popis první a druhé vrstvy OSI modelu, přesněji řečeno fyzické a MAC vrstvy viz. obr. č. 1



Obrázek 1

Pro fyzickou vrstvu je definován přenos pomocí infračerveného světla a rádiový přenos v rozprostřeném spektru a to technikou přímé sekvence (DSSS) nebo technikou přeskoku kmitočtů (FHSS). Systémy pracující infračerveným přenosem pracující v pásmu 850 – 950 nm jsou schopny pokrýt prakticky jen jednu místnost, protože pevné překážky infračervené světlo nepropouští, a z tohoto důvodu nejsou příliš zajímavé.

Co si představit pod pojmem rozprostřené spektrum? Šířka pásma vysílaného signálu je mnohem větší než šířka pásma originálního přenášeného datového signálu - zprávy. Vysílaný signál je určen datovou zprávou a rozprostírací funkcí (kódovou sekvencí, Spreading code; viz. obr. č.2), nezávislou na datové zprávě a známou jen vysílači a určenému přijímači.



Obrázek 2

V praxi to znamená, že systémy jsou imunní vůči interferencím generovaným jinými signály, ať toho už rozprostřenými nebo úzkopásmovými, přítomnými ve stejném frekvenčním pásmu. Také jsou obtížně zachytitelné.

V důsledku mohou být systémy s rozprostřeným spektrem umístěny v jednom místě bez nutnosti koordinace, jinými slovy bez přidělování frekvencí. Výsledkem toho je, že jejich provoz není zpoplatňován. Pro bezlicenční provoz je nejen u nás vyhrazena pásma 2,4 – 2,4385 GHz (802.11b/g) a 5470-5725 GHz (802.11a). Z technického pohledu používají tyto systémy dva modulační procesy (viz. obr. č.2 a 3):

a) modulace prováděná kódovou sekvencí (Spreading Code)

b) modulace prováděná datovou zprávou.

U DSSS jsou jednotlivé bity přenášeny pomocí jedenácti tzv. chipů. Důsledkem toho je, že zpráva je přenášena v širším frekvenčním spektru, každý datový bit je reprezentován známou sekvencí a ne všechny chipy jsou tudíž potřebné pro správnou demodulaci. Použití odlišných sekvenčních kódů pak umožňuje umístění více DSSS systémů v jednom místě.

U FHSS je jako sekvenční kód použita sekvence až 78 možných frekvencí. Datová zpráva je tak vysílána pomocí mnoha nosných frekvencí tzv. hops. Vysoké spolehlivosti je dosaženo díky tomu, že nepotvrzené tj. chybně přenesené rámce jsou znovu přenášeny s jinou nosnou frekvencí tj. v dalším hopu. Umístění více systémů v jednom místě je umožněno použitím různých sekvencí v každém systému.

Standard 802.11 podporuje rychlosti 1 a 2 Mbps pro oba systémy. Standard 802.11b definuje rychlost 11; 5,5; 2 a 1Mbps, ale pouze pro systémy pracující DSSS technikou.

Oba systémy mají své výhody a nevýhody:

- FHSS umožňuje koexistenci více systémů (System Collocation) v jedné lokalitě. Teoreticky až 26, prakticky cca 15.
- U DSSS jsou to pouze 3 systémy bez vzájemného rušení. Je to dáno tím, že pro koexistenci více systémů by byl nutný větší počet chipů, např. pro 16 systému by to bylo 255 chipů. To by znamenalo požadavek na mnohonásobně rychlejší rádiový přenos než je prakticky možné.
- DSSS systém má větší propustnost. FHSS spotřebovává část času na přeskok a synchronizaci na jinou frekvenci.
- FHSS má menší problémy s vícecestným šířením signálů. DSSS pracuje s vyšší modulační frekvencí, tím pádem s kratšími symboly a je tak více citlivý na různá zpoždění přijímaných signálů.
- DSSS systém je schopný si poradit s vyšší úrovní interferencí. Při silném rušení, které blokuje některé frekvence, je naopak FHSS systém schopný fungovat na nerušených frekvencích. Totéž platí pro tzv. near/far problém, kdy blízký zdroj interferencí může způsobit zablokování přijímače. FHSS systém může dále fungovat na neblokovaných frekvencích.
- DSSS používá pro příjem a vysílaní různá oddělená pásma, může tak i v plném duplexu používat pouze jednu anténu s filtrem na vstupu přijímače.

Pokud jde o složitost rádiové části a tím de facto i ceny, platí trochu zjednodušeně, že implementace FSK (Frequency Shift Key) pro FHSS je jednodušší než PSK (Phase Shift Key) používané DSSS systémy.

2.2 802.11 – MAC vrstva

Jak je vidět na obrázku č.2 standard 802.11 definuje dvě přístupové metody – DCF (Distributed Coordination Function) a PCF (Point Cordination Function). PCF je pouze volitelný mechanismus, který slouží pro přenos aplikací citlivých z hlediska času, například hlasu a videa.
Základním přístupovým mechanismem neboli distribuční koordinační funkcí je CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance).
CSMA je mechanismus použitý u klasického Ethernetu. CS (Carrier Sense) znamená, že stanice před vysíláním naslouchá na médiu a začne vysílat pouze pokud je médium volné. MA (Multiple Access) znamená, že je umožněn současný přístup více stanic k médiu. Rozdíl je v tom, že klasický Ethernet používá mechanismus detekce kolizí (Carrier Detection). U bezdrátového Ethernetu je použit mechanismus předcházení kolizí (Collision Avoidance). Proč?
U klasického Ethernetu, např. na koaxu, může každá stanice slyšet vysílání jiné stanice a detekovat kolizi. Tento základní předpoklad pro detekování kolizí u bezdrátového Ethernetu neplatí. Stanice může detekovat volné médium ve svém okolí, to však neznamená, že je volné i u přijímače. Jak je uvedeno výše, stanice komunikují prostřednictví AP a nemusí se tak vůbec přímo slyšet s jinou stanici ani detekovat její vysílání. Proto je použit mechanismus předcházení kolizím spolu s kladným potvrzováním. To znamená, že stanice naslouchá a pokud je médium volné počká ještě určený čas (DIFS,Distributed Inter Frame Space) a teprve pak začne vysílat. Přijímající stanice zkontroluje kontrolní součet (CRC) přijatého paketu a odešle potvrzení (ACK). Přijetí potvrzujícího paketu znamená pro odesílající stanici, že nedošlo ke kolizi. Pokud stanice ACK paket nedostane, opakuje vysílání.
Pro snížení pravděpodobnosti kolizí způsobených tím, že se stanice nemohou slyšet, definuje standard "virtuální" naslouchací mechanismus. Stanice, která chce vysílat, pošle nejdříve krátký řídící paket (RTS, Request To Send), který obsahuje kromě zdroje a cíle i trvání následujícího přenosu. Cílová stanice odpoví jiným řídícím paketem (CTS, Clear To Send), který rovněž obsahuje dobu trvání následujícího přenosu. Stanice slyšící RTS a/nebo CTS paket si nastaví indikátor virtuálního naslouchání, tzv. NAV (Network Allocation Vector) na dobu trvání přenosu. Jinými slovy bude po tuto dobu brát médium jako obsazené. Snižuje se tak pravděpodobnost kolize ze strany ostatních stanic v lokalitě příjemce pouze na dobu vysílání RTS, protože pak už zachytí paket CTS a budou brát médium jako obsazené. Takový mechanismus je efektivní pouze pro delší pakety, proto standard umožňuje také přenos bez RTS/CTS mechanismu. Tato možnost je volitelně nastavitelná na stanici (RTS Treshold). Rovněž multicasty a broadcasty se nepotvrzují.

2.3 Vzájemná Kompatibilita

V roce 1999 byla vytvořena aliance WECA (Wireless Ethernet Kompatibility Alliance), v roce 2002 byla přejmenována na WiFi alianci. Jde o neziskovou průmyslovou skupinu sdružující na 300 společností, jejíž cílem je zajistit vzájemnou kompatibilitu výrobků založených na standardu 802.11. Aliance vytvořila testovací laboratoře. Po testování získává produkt logo Wi-Fi, které prokazuje jeho shodu s testovacími standardy.

3 Bezpečnost a metody autentizace

Bezpečnost slouží k zajištění potřebné ochrany osobních nebo obchodních informací. Je důležité pochopit skutečnost, že bezpečnost nelze definovat v absolutních hodnotách. Musíme bohužel přiznat, že neexistuje dokonale zabezpečený počítačový systém. Dokonce se říká, že nejbezpečnější počítač je vypnutý počítač. Otázkou pak ale zůstává, jak moc je takový počítač užitečný. S pojmem bezpečnosti u bezdrátových sítí úzce souvisí pojem autentizace, tedy jakési ověření klienta, zda-li má právo přístupu.

3.1 Metody autentizace v bezdrátové komunikaci

Standard 802.11 specifikuje dvě možné metody autentizace (ověření "totožnosti" uživatele) na linkové vrstvě – Open systém a shared key. Dále pak je tu autentizace MAC adresou nebo pomocí standardu 802.1x.

3.1.1 Open system (otevřený systém)

Přístupový bod přijme klienta pouze na základě údajů, které mu toto zařízení poskytne. Tyto informace už dále nikde neověřuje, pouze je akceptuje. Toto řešení je výhodou u veřejných přístupových bodů, kde je neomezený přistup záměrem.

3.1.1.1 SSID

Jedná se o první a základní stupeň ochrany bezdrátové sítě. Každé AP (Access Point) vysílá administrativní signalizaci (tzv. beacon), kterým ohlašuje svou přítomnost. Zpráva obsahuje několik informací o AP, např. SSID (Service Set Identifier – název sítě), podporované rychlosti, sílu signálu, atd. Vysílání SSID má své výhody i nevýhody. Fakt, že se snadno prohledáte na svém bezdrátovém

klientu okolní sítě – tedy jakási reklama, je nesporně výhodou. Ovšem také fakt, že takovýmto potenciálním klientem může být i potenciální útočník je více méně pravdou. Tedy z hlediska bezpečnosti je vysílání SSID spíše nevhodné, neboť útočník díky tomu získává snadnou možnost nalézt a identifikovat vaši síť. Na AP jde také nastavit to, aby se SSID nevyplňovalo, tedy jakýsi stupeň ochrany. Pokud ale útočník odposlouchává provoz bezdrátové sítě – datové rámce, je schopen tuto bariéru prolomit. Navíc Deaktivací signálních rámců může dojít ke zhoršení interoperability wireless zařízení.

3.1.2 Shared key (sdílený klíč)

Tento systém autentizace pracuje na principu klíčů, které znají pouze žádaní uživatelé. Standard 802.11 dokonce vyžaduje, aby bylo každé zařízení s WEP schopné používat autentizaci sdíleným klíčem.
Sdílený klíč vyžaduje znalost WEP klíče(slouží i k šifrování dat). Princip:

- Klient pošle na AP autorizační požadavek
- AP pošle klientovi 64 nebo 128 bajtů dlouhou výzvu
- Klient zašifruje výzvu svým WEP klíčem a zašifrovaný text pošle zpátky na AP
- AP využije svou znalost WEP klíče a ověří, zda klientem odeslaná odpověď odpovídá původní výzvě
- AP klientovi oznámí úspěšnou či neúspěšnou autentizaci

Bohužel však tento typ autentizace přináší více bezpečnostních rizik, než by se dalo očekávat. Tím, že se posílá jeden text (náhodné číslo) nejprve jako plain text, a nazpět již zašifrované, může tak útočník odposlouchávající náš přenos získat hodnotné informace - dvojici nezašifrovaného a zašifrovaného textu, ze kterého již pak jednoduchým způsobem získá použitý klíč.

3.1.2.1 WEP

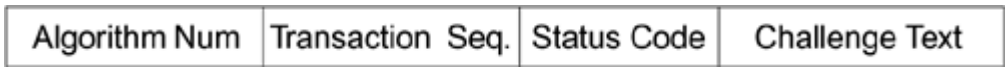
Protokol WEP (Wired Equivalent Privacy) pracuje jako volitelný doplněk k 802.11b pro řízení přístupu k síti a zabezpečení přenášených dat. Dnes již tento protokol využívá většina sítí.
WEP používá k šifrování zpráv symetrickou šifru RC4 - princip spočívá v tom, že se odesílána zpráva na vysílači zašifruje nějakým klíčem, a přijímač ji stejným klíčem rozšifruje. Tento klíč musí být znám jak vysílající stanici, tak přijímací (ve standardu se jedná o 40-bitový klíč).
Aby to nebylo tak jednoduché, klič se expanduje na délku stejnou jako má vysílaná zpráva, a to pomocí tzv. inicializačního vektoru. Jedná se o 24-bitový pseudonáhodný sled znaků, který se na straně vysílače přidá k tajnému klíči (tím vzniká 64-bitová "šifra", se kterou se pak zašifruje zpráva) a

také se pošle (nijak nezakódovaný! - toto patří mezi jednu z hlavních nevýhod WEP) přijímači, aby ten ho mohl přidat ke svému tajnému klíči a tento složený klíč pak použít pro dešifrování.
Někteří výrobci poskytují i vyšší úrovně zabezpečení ve formě 128-bitového šifrování (sdílený klíč má délku 104 bitů, inicializační vektor poté 24 bitů).
I WEP je však možné poměrně snadno obejít. Je bohužel možné prolomit WEP klíč pouhým sledováním a odposloucháváním provozu na síti. Hlavní problémy WEP spočívají především ve statických klíčích (nijak neřeší automatickou distribuci nových klíčů, a tak si ho v případě změny musí každý uživatel sám ručně znovu nastavit), a ve slabém inicializačním vektoru (posílá se "vzduchem" nezakódovaný a ještě se jeho kombinace poměrně "brzy" vyčerpají - jedná se "pouze" o 2^{24} možností).

3.1.3 MAC Adress authentication

Některé přístupové body umožňují omezit přístup do sítě podle MAC adres. MAC adresa je jednoznačný identifikátor síťové karty (ať už "drátové", nebo bezdrátové). Ani filtrování MAC adres není všespasitelné, přináší totiž několik problémů - mezi ty základní patří distribuce seznamu MAC adres a možnost falšovat MAC adresu. Každý přístupový bod si totiž musí udržovat vlastní databázi povolených MAC adres. Tento způsob autentizace také není zahrnut ve standardu 802.11, tudíž závisí na výrobci jestli tento způsob autentifikace zahrne do vlastností AP nebo ne. Dnes již také existují nástroje na falšování MAC adres a útočník si může odposlouchávat komunikaci na síti a odchytit si jednu z povolených MAC adres, kterou později použije

3.1.4 Formát autentizačních zpráv



Význam jednotlivých polí:
Algorithm Number (2B) - označuje číslo použité autentifikace:
0 - open-system
1 - shared key
Transaction Seq. (2B) - indikuje kde se právě nacházíme v autentizační sekvenci. První zpráva se označuje 1, druhá 2, atd...
Status code (2B) - posílá se v poslední zprávě jako indikace úspěchu/neúspěchu autentizačního požadavku.

tabulka 1

Hodnota	Význam
0	Úspěch
1	Nespecifikovaná chyba
2-9	Rezervováno
10	Nejsou podporovány všechny požadované funkce z Capability Information field
11	Reasociace byla odmítnuta kvůli nemožnosti potvrdit současnou asociaci
12	Asociace byla odmítnuta kvůli důvodu mimo tento standard
13	Odpovídající stanice nepodporuje specifikovaný autentifikační protokol
14	Byl přijat autentizační rámec s "Transaction sequence" jiným než je očekáván
15	Autentizace odmítnuta kvůli špatné odpovědi (challenge failure)
16	Autentizace odmítnuta kvůli vypršení časového limitu při čekání na další rámec v pořadí
17	Asociace odmítnuta protože AP není schopno obsloužit další stanice
18	Asociace odmítnuta protože stanice nepodporuje všechny vyžadované přenosové rychlosti
19-65535	Rezervováno

Challenge Text (3B - 255B) - je používán u shared key autentizace

Formát těla zprávy:

ElementID (1B)	Length (1B)	Challenge Text (1-253B)

3.1.5 PPP(Point to Point Protokol)

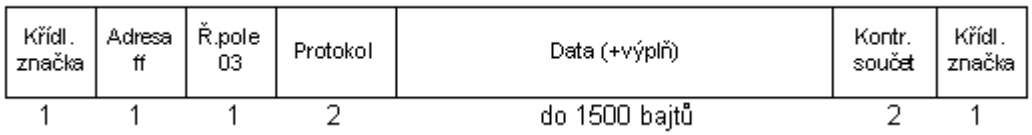
Abychom pochopili protokol 802.1x musíme nejprve uvést principy protokolu PPP ze kterého 802.1x vychází.

Protokol PPP (Point-to-Point Protocol) poskytuje standardní služby pro přenos datagramů různých formátů skrz dvoubodové spoje (sériové, telefonní případně ISDN linky).

- Nevyžaduje žádné řídící signály (RTS, CTS, DCD, DTR atp.). Řídící signály však mohou být využity pro zvýšení efektivity.
- Může používat jak asynchronní, tak bitově či znakově synchronní přenos dat.
- Vyžaduje plně duplexní dvojbodové spoje (point-to-point), které mohou být pevné i komutované.
- Využívá zpravidla 16 nebo 32 bitů pro kontrolní součet, aby mohl zjistit, zda nebyl rámec během přenosu poškozen.
- Skládá se ze dvou vrstev:
    - Link Control Protocol (LCP) užívaný pro navázání spojení po telefonní lince, konfiguraci a testování, případně i autentizaci. V okamžiku, kdy je tímto protokolem zavázáno spojeni, použije se jeden či více protokolu typu
    - Network Control Protocol (NCP) pro přenost dat. Protokolu NCP existuje více a lze je využívat i současně v jednom navázaném spojeni:
        - IPCP - protokol NCP zajišťující přenos IP verze 4
        - IPXCP - protokol NCP přenášející IPX
        - IPV6 - pro IP verze 6
        - SNACP, DNCP, OSINLCP a další

Cílem protokolu PPP je umožnit po jedné lince přenášet více síťových protokolů současně (mixovat protokoly).

3.1.5.1 Tvar rámce PPP



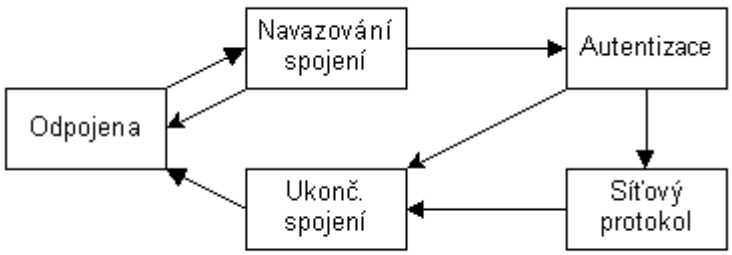
Obrázek 3

Křídlová značka - uvozuje i ukončuje každý PPP-rámec. Obsahuje binárně 01111110 (hexadecimálně 7e). Aby bylo možno přenášet tuto hodnotu jako platná data, používá se technika Esc-sekvencí a nebo bit-stuffing.
Adresa - obsahuje vždy hodnotu 11111111 (broadcast). Důsledek - ppp neumožňuje určit příjemce paketu.
Řídící pole - obsahuje hodnotu 00000011.
Pokud se na lince vyskytují rámce pouze s těmito adresami a řídícími poli, pak oba konce linky mohou použít kompresi (Address-and-Control-Field-Compression). Při této kompresi se prostě při vysílání tato dvě pole vypustí a při příjmu se opět doplní.

Protokol - obsahuje číslo identifikující typ protokolu přenášeného jako data.
Data - nula nebo více bajtů obsahujících datagram protokolu specifikovaného v poli protokol. Maximální délka je závislá na implementaci konkrétního PPP a lze ji měnit, typicky 1500 bajtů.
Kontrolní součet - implicitně dvoubajtová hodnota zajišťující detekci chyb. Velikost lze ale dohodnout na 4 bajty.

3.1.5.2 Protokol LCP

Protokol LCP se používá ještě před tím, než se vůbec uvažuje o tom, jaký síťový protokol na lince poběží. LCP je tedy společný (na rozdíl od protokolů NCP) pro jednotlivé síťové protokoly. Protokol LCP je určen pro navázání spojení, ukončení spojení, výměnu autentizačních informací a pod. Linka může být v následujících fázích:



Obrázek 4

Linka odpojena je fáze, ze které se vždy začíná a končí. Když dojde k nějaké externí události (např. ztráta nosné - modemy ztratí mezi sebou spojení nebo síťový administrátor vydá příkaz k ukončení spojení), přechází linka do této fáze.
Z fáze linka odpojena se přechází do fáze **navazování spojení**. Navazování spojení se provádí výměnou konfiguračních paketů. Během navazování spojení se žádné datové pakety (tj. pakety síťového protokolu - např. IP) nepřenáší . V případě výskytu datového paketu během navazování spojení se takový paket zahazuje.

Autentizace je fáze, kdy klient prokazuje svou totožnost. Klientem je ta strana (stanice), která je vyzvána k prokázání své totožnosti. Ovšem, že si po prokázání totožnosti jedné stanice mohou stanice svou roli vyměnit a k prokázání své totožnosti může být vyzvána i druhá strana. V praxi většinou prokazuje svou totožnost jen strana jedna (např. uživatel PC proti Internet Service Providerovi). Autentizace je nepovinná, tj. může být přeskočena. Během autentizace opět nemohou být přenášeny datové pakety (síťový protokol).

Důležité je, že autentizace pouze přenáší data používaná k vlastnímu prokazování totožnosti. Tj. protokol LCP nepopisuje žádný autentizační algoritmus, pouze přenáší data, která pak následně využijí autentizační protokoly. Jako autentizační algoritmus se používá zpravidla buď protokol PAP nebo protokol CHAP. Navíc ještě je zpravidla možná terminálová autentizace.

Fáze, který je na předchozím obrázku označena jako **síťový protokol** v sobě může obsahovat celou řadu kroků. V tomto okamžiku totiž přicházejí ke slovu jednotlivé protokoly NCP. Každý síťový protokol, který chce linku využívat si musí jednotlivě přivést pomocí svého protokolu NCP linku do otevřeného stavu. Pokud se objeví datové pakety síťového protokolu, pro který není linka otevřena, pak se tyto pakety zahodí.

Poslední fází je fáze **ukončování spojení**. Během této fáze jsou všechny jiné pakety než protokolu LCP zahazovány. Fyzické vrstvě je signalizováno ukončení spojení. Ta může pak reagovat např. zavěšením komutované linky.

3.1.5.3 Metody autentizace u PPP

Prokazovat totožnost lze v případě protokolu PPP trojím způsobem (neuvažujeme-li jako čtvrtou možnost eventualitu - autentizace zcela vynechána):

- **Terminálový dialog**. Terminálový dialog nesouvisí s protokolem PPP, nýbrž s jeho implementací. Zpravidla se totiž uživatel přihlašuje po sériové lince k serveru. Na serveru sedí na této lince terminálový proces vyžadující jméno uživatele a heslo. Teprve ze jména uživatele pozná, že se nejedná o běžného uživatele terminálu, ale uživatele, pro kterého má na lince startovat protokol PPP (např. proces pppd). Pokud je takováto autentizace na serveru možná a je dostačující, pak je možné autentizační fázi protokolu PPP přeskočit.
  Výhodou je, že uživatel může předat providerovi heslo šifrované (ve tvaru v jakém je např. v UNIXu v souboru /etc/passwd) a pak jej žádný zaměstnanec providera nemůže zneužít a přihlásit se místo uživatele.

- **Password Authentication Protocol** (PAP). Tento protokol je obdobou autentizace pomocí terminálového dialogu. Tj. uživatel prokazuje svou totožnost také pomocí jména uživatele a hesla. Pro výměnu autentizačních informací se ale použije protokol LCP, tj. jméno uživatele a heslo se nevkládá přímo na linku, ale balí se do protokolu LCP.
- **Challenge Handshake Authentication Protocol** (CHAP). Je považován za dokonalejší a je mu dávána přednost. Oba konce sdílí stejný šifrovací klíč symetrické šifry (pochopitelně tajný, tzv. sdílené tajemství). Stanice, která autentizaci inicializuje, vygeneruje náhodný

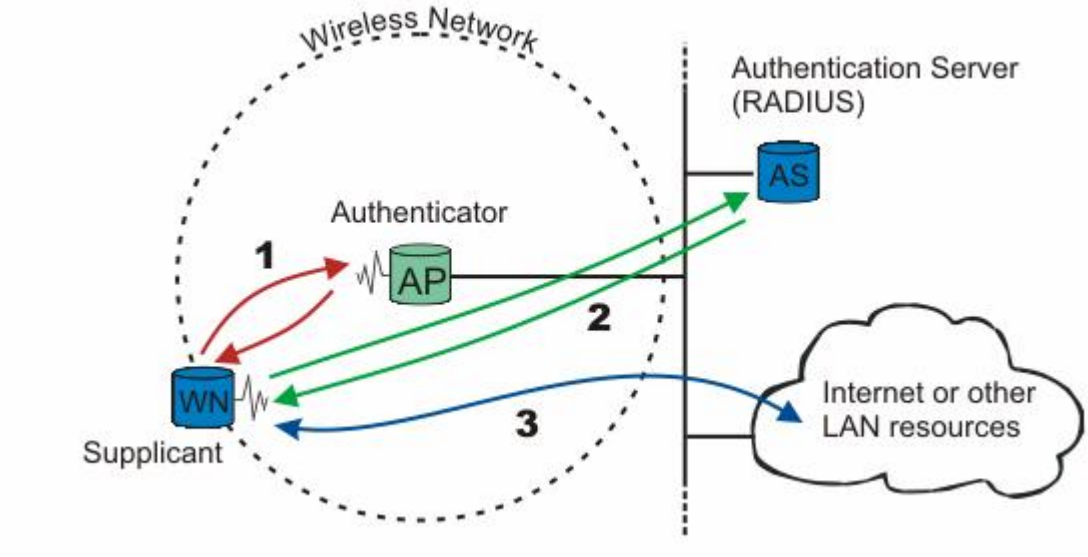
řetězec jako dotaz (challenge), který odešle druhé straně. Druhá strana tento řetězec zašifruje odešle zpět. Stanice, která autentizaci inicializovala tak obdržela zašifrovaný řetězec. Poté si vezme původní řetězec a zašifruje jej sama. Porovná oba výsledky. Jsou-li stejné, pak protějšku potvrdí úspěšný výsledek autentizace. V opačném případě odpoví, že autentizace proběhla neúspěšně a může se začít znovu s navazováním spojení.
Server musí být obezřetný v generování challenge. Musí si dát pozor, aby v krátkém časovém intervalu nevygeneroval stejný challenge. Pak by nezvaný host mohl použít odposlechnutou odpověď a přihlásit se. Výhodou protokolu CHAP je skutečnost, že oba konce znají stejný sdílený šifrovací klíč - je tak snadno možné provádět autentizaci oboustranně.

3.1.6 802.1x

IEEE 802.1x není produktem pracovní skupiny 802.11 (která se věnuje bezdrátovým sítím), ale skupiny 802.1, která se zabývá LAN protokoly na vyšších vrstvách modelu ISO/OSI. Tím pádem je možné ho využít nejen v drátových sítích (kam byl původně určen), ale i v bezdrátových.
IEEE 802.1x zajišťuje autentizaci uživatelů, integritu zpráv (šifrováním) a distribuci klíčů. Ověřování provádí přístupový bod na základě výzvy klienta pomocí externího autentizačního systému (např. Kerberos, nebo Radius).

Samotný protokol 802.1x vychází z protokolu PPP (Point-toPoint Protocol). PPP je ale omezen na autentizaci založenou pouze na kombinaci uživatelského jména a hesla.

Poznámka: 802.1x není dostatečně odolný vůči minimálně dvěma způsobům útoku (hijacking a man-in-the-middle), které díky jednostranné autentizaci umožní útočníkovi snadno podvrhnout přístupový bod.



Obrázek 5 - princip 802.1x

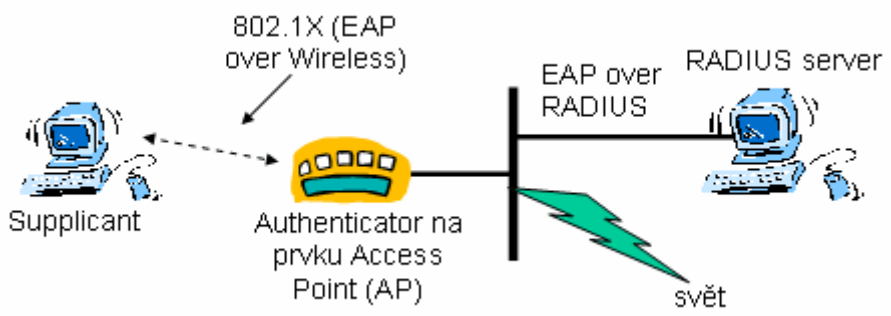
3.1.6.1 EAP

Protokol EAP byl původně vytvořen jako rozšíření protokolu PPP. Cílem bylo vytvořit obecnou platformu pro různé autentizační metody. Tedy je to v podstatě PPP se „zásuvnými“ autentizačními moduly. Díky EAP můžeme autentizovat klienty jak se nám zlíbí, tedy používat hesla, certifikáty, tokeny, čipové karty, Kerberos, aj. Otevřený standard nám zajišťuje, že kdykoliv v budoucnu bude vynalezen lepší způsob autentizace, vyvine se nový EAP.

3.1.6.2 EAP/802.1x

802.1x je protokol, který nám umožňuje používat EAP na metalických nebo bezdrátových sítích. Procesu ověřování identity se zúčastňují obvykle 3 strany :

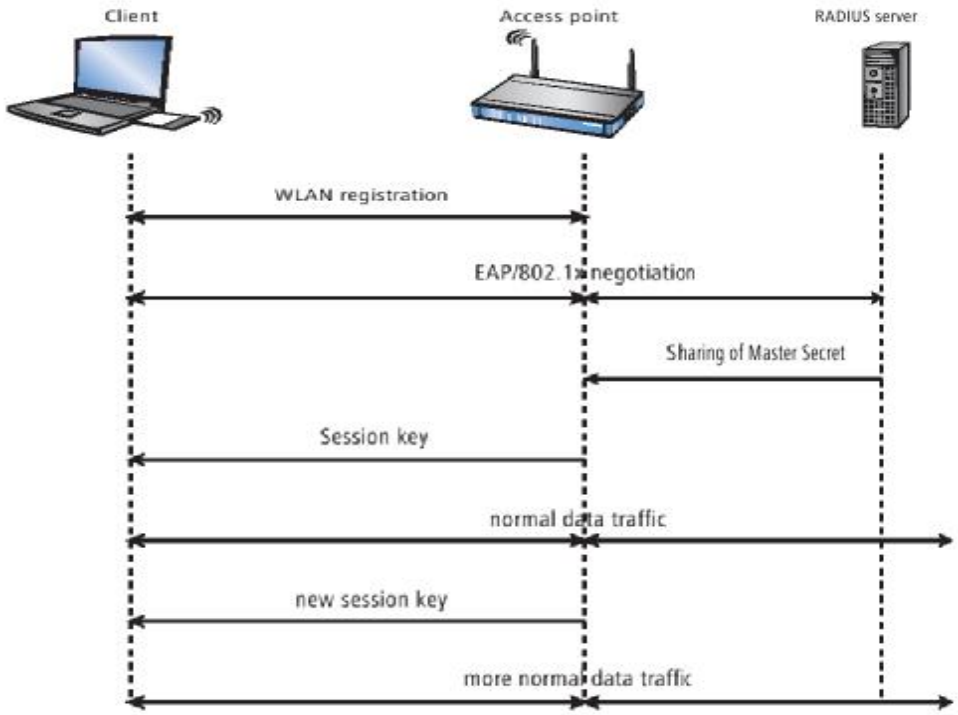
- Klient (supplicant), který se snaží o připojení - jedná se o softwarový modul na pracovních stanicích. Některé operační systémy ho mají pevně zabudovaný (WinXP)
- Systém, který ověřování provádí (autentizátor - access point či přepínač)
- Server, kde jsou uloženy veškeré informace o klientech (typicky RADIUS server). Doplňkovou entitou tohoto modelu může být PAE (Port Access Entity)



Obrázek 6

Mechanismus založený na komunikaci typu klient/server, specifikuje přístup k portu zařízení (přepínače). 802.1x blokuje veškerý provoz na daném portu až do doby, než se klient autentizuje prostřednictvím)údajů, které jsou uloženy na back-end serveru, kterým je typicky RADIUS
Definuje 2 základní stavy portů :

- Řízený port (controlled) - otevře se pouze jen po úspěšně provedené autentizaci.
- Neřízený port (uncontrolled) - ten propouští pouze rámce, které přenášejí autentizační informace přes síť. Tedy slouží ke komunikaci autentizátora s autentizačním serverem.



Obrázek 7 EAP/ 802.1x

Princip je následující. Nejprve se klient standardně přihlásí k přístupovému bodu, takže si s ním může vyměňovat pakety. Spojení je chráněno zatím jen běžným WEPem. V tomto stavu přijímá přístupový bod od klienta jen určité pakety(EAP Start). Jedná se právě o pakety protokolu EAP/802.1x. Díky využití standardu IEEE 802.1x jsou tyto pakety snadno odlišitelné od ostatních, neboť mají speciální typ Ethernet rámce *0x888E*.
Pokud přístupový bod přijme od klienta EAP/802.1x paket, vytvoří z něho žádost RADIUS a tuto žádost pošle autentizačnímu serveru RADIUS. Naopak, přístupový bod vytváří z odpovědí od serveru

opět EAP pakety a posílá je klientovi. Přístupový bod tak vytváří jen jakýsi tunel mezi klientem a autentizačním serverem. Navíc převod mezi EAP pakety a pakety pro RADIUS server neklade na přístupový bod žádné velké nároky a lze ho snadno naimplementovat. Tímto tunelem si klient a server mohou vzájemně vyměňovat autentizační informace.
Pro vzájemnou autentizaci serveru a klienta se používají různé mechanismy. Tomu odpovídají i různé varianty protokolu EAP, viz tabulka.

tabulka 2

Tabulka Varianty protokolu EAP.

Autentizace serveru	--	hash hesla	veřejný klíč (certifikát)	veřejný klíč (certifikát)	veřejný klíč (certifikát)
Autentizace klienta	hash hesla	hash hesla	veřejný klíč (certifikát nebo Mart karta)	CHAP, PAP, MS-CHAP(v2), EAP	jakýkoliv EAP, např EAP-MS-CHAPv2 nebo veř. klíč
Dynamické posílání klíčů	ne	Ano	ano	ano	ano
Poznámka	nejjednodušší a nejslabší varianta	LEAP = Lightweight EAP	TLS = Transport Layer Security	TTLS = Tunelled TLS	PEAP = Protected EAP
	EAP-MD5	LEAP	EAP-TLS	EAP-TTLS	PEAP

Po dokončení fáze autentizace se vytváří bezpečný tunel mezi klientem a serverem, aniž by pro to byl potřeba WEP. Po této fázi pošle RADIUS server přístupovému bodu klíč "od tunelu" - tzv. *"Master Secret"*, čímž umožní příst. bodu komunikovat s klientem přes tento bezpečný tunel.
Díky tomuto bezpečnému tunelu může nyní přístupový bod poslat klientovi WEP klíč. Buď se jedná o tzv. *Session Key*, což je WEP klíč vygenerovaný pro spojení s konkrétním klientem, nebo o tzv. *Group Key*, což je WEP klíč použitý pro větší počet klientů současně. Typicky většina běžně používaných přístupových bodů podporuje pouze skupinové klíče.
Díky EAP protokolu je tedy možné dynamicky distribuovat WEP klíče klientům. Obměna klíčů se děje typicky po 5 minutách, což je doba, během které je riziko prolomení mechanismu WEP zanedbatelné.

Výhodou použití EAP/802.1x je poměrně vysoké zabezpečení a nutnost jen drobných změn v původním hardwaru podporujícím WEP. Mezi nevýhody patří složitost, nutnost mít autentizační server atd., což předurčuje toto řešení pouze pro větší podnikové sítě a ne pro domácí a malé sítě. Navíc také nebyla stanovena žádná minimální sada autentizačních mechanismů, které musí klient i server podporovat, takže se může stát se si klient se serverem při autentizaci nebude "rozumět".

3.1.6.3 MD5

Metoda MD5 představuje nejnižší možnou úroveň zabezpečení a nejsnázeji se implementuje. Původně je označovaná v PPP protokolu jako CHEAP. Je napadnutelná řadou útoků. Kromě toho tato metoda neumožňuje vzájemnou autentizaci. Jde pouze o jednosměrnou autentizace, takže AP ověří totožnost klienta, klient si ale nemá možnost ověřit totožnost APOD. V kontextu PPP to příliš nevadí. V kontextu bezdrátových sítí se vzájemná autentizace však považuje za nezbytnou. Metoda MD5 nepodporuje dynamické generování WEP/TKIP (sdílených) klíčů. Neobsahuje také žádné mechanismy umožňující vytvářet individuální klíče pro jednotlivé klienty a relace.

3.1.6.4 LEAP

Protokol LEAP (Lightweight Extensible Authetication Protocol) poskytuje jak vzájemnou autentizaci, tak i dynamickou obnovu WEP klíčů. Tento protokol navrhla v roce 2000 společnost Cisco jak dočasné řešení před schválením standardu 802.1x. Jde bohužel o protokol nestandardní a proprietární. Podporovala jej pouze zařízení Cisco a nedočkal se široké podpory u ostatních výrobců. Výhodou bylo, že zařízení se dají použít na celé řadě platforem, neboť Cisco podporuje klientské adaptéry pro celou řadu operačních systémů včetně Windows, Macintosh a Linuxu. Nevýhodou tohoto řešení je, že je třeba použít výhradně adaptérů a zařízení Cisco. Jedině pak bylo možno LEAP použít.

3.1.6.5 TLS

Protokol TLS (Transport Layer Security) představuje z pohledu bezpečnosti nejsilnější řešení. Jeho nasazení je ala zároveň nejobtížnější. TLS poskytuje vzájemnou autentizaci i dynamickou obnovu WEP klíčů. Protokol prostřednictvím PKI (infrastruktury veřejného klíče) vytváří šifrovaný tunel, jímž probíhá výměna autentizačních údajů. Tedy musí být na straně serveru i na straně klienta instalovány digitální certifikáty.Výhodou je že poskytnutí bezpečnosti je na nejvyšší možné úrovni. Nevýhodou pak že vybudování infrastruktury s podporou PKI je komplikované.

3.1.6.6 TTLS a PEAP

TTLS (Tunneled Transport Layer Security) a PEAP ( ) představují rozšíření TLS. U těchto metod se prostřednictvím TLS autentizuje AP, autentizace uživatele následně proběhne jiným tunelovaným protokolem. Jinak řečeno, TLA se použije k vytvoření bezpečného kanálu (certifikace na straně serveru) a uživatel se autentizuje jinou EAP metodou s využitím vytvořeného bezpečného kanálu.

4 Autentizační nástroje

Existuje řada autentizačních protokolů, které lze použít. Autentizačním nástrojem myslíme zejména druh serveru a jeho princip

4.1 Kerberos

V řecké mytologii je Kerberos strážce vchodu do podsvětí - trojhlavý pes, který má zabránit návratu duší do normálního světa. Ale "norma" RFC1510 nám tvrdí o Kerberovi něco jiného. Podle tohoto RFC je Kerberos protokol zajišťující bezpečné ověření totožnosti (autentizaci) přes nezabezpečené sítě.
Autentizace pomocí protokolu Kerberos umožňuje využít principu jednotného přihlášení (SSO - Single Singn-On) mezi různými systémy.

4.1.1 Historie

Protokol Kerberos byl vyvinut jako součást projektu Athena na MIT (Massachusetts Institute of Technology). První veřejně uvolněná verze byl Kerberos V4. Kerberos V5, který se používá dnes, byl vypuštěn roku 1996. Z důvodu omezení vývozu šifrovacích technologií z USA vznikl v Evropě projekt Heimdal, který si dal za úkol kompatibilitu s protokolem Kerberos V4 a V5 a umožnil tak vyžití tohoto protokolu i mimo USA.

4.1.2 Popis činnosti systému

Autentizace uživatelů pomocí protokolu Kerberos je založena na důvěryhodné třetí straně. Touto důvěryhodnou třetí stranou je centrální autentizační server (KDC - Key Distribution Center). KDC spravuje databázi uživatelů a přiděluje jim tikety. Autentizace pomocí Kerberos systému je navržena tak, aby nemuselo po síti pohybovat heslo, místo něj se používá přidělený lístek, který neobsahuje žádné tajné informace.

KDC se skládá ze dvou služeb. První je autentizační server (AS, Authentication Server) a druhou službou je server pro přidělování lístků (TGS, Ticket Granting Server). Tyto dvě služby jsou samostatné, ale většinou se provozují na společném počítači. Tento počítač by měl být velmi dobře zabezpečený, protože při jeho kompromitaci může útočník získat přístup do celé sítě.

Lístky přidělené uživateli mají omezenou platnost, která jde v případě potřeby prodlužovat. Po propadnutí lístku si může uživatel požádat o nový.

4.1.3 Dostupné Implementace

Původní implementaci Kerbera najdeme u MIT (MIT Kerberos). Další dostupnou implementací je Heimdal, Heimdal používá stejné API (rozhranní pro programování aplikací) jako MIT Kerberos, proto je téměř jedno kterou verzi použijete ve svém programu.
Další implementaci Kerbera najdeme u Microsoftu, kde se od MS Windows 2000 používá jako autentizační mechanizmus v Active Directory.
Podpora protokolu Kerberos se najde i u firmy SUN Microsystems a v její javě.
Mechanismus Kerberos je podporován řadou aplikací. Pro přihlašování na lze použít klienty telnet, příp. ssh, kteří podporují autentizaci a následné šifrování přenosového kanálu. Pro přenos souborů lze použít ftp, aj. Tyto aplikace také přenášejí uživatelovy lístky, což umožňuje pohodlné přecházení mezi více stroji bez nutnosti opakovaného zadávání hesla.

4.2 TACACS

TACACS (*Terminal Access Controller Access-Control System*, česky *kontrolor terminálového přístupu k systému řízení přístupu*) je vzdálený autentikační protokol používaný ke komunikaci s autentikačním serverem často používaný v UNIX sítích. TACACS umožňuje ověřit každého uživatele na individuální bázi před přístupem ke směrovači nebo komunikačnímu serveru.
TACACS umožňuje klientu přijmout uživatelské jméno a heslo a poslat požadavek na TACACS autentikační server (TACACS démon, TACACSD) . Tento server rozhodne zda přijmout nebo zamítnout požadavek a pošle zpět odpověď.
Takto je rozhodovací proces otevřený a algoritmy a informace k němu použité jsou zcela na tom, kdo provozuje TACACS démona.

Novější verze TACACS od roku 1990 byly nazývány XTACACS nebo extended (rozšířený) TACACS. TACACS lze používat ve spolupráci se systémem Kerberos. Obě verze TACACS A XTACACS byly většinou nahrazeny novějšími protokoly TACACS+, RADIUS nebo DIAMETER.

Příkladem softwarové implementace je DialWays 3.0 nebo tac_plus

4.3 Radius Server

4.3.1 Co je to RADIUS ?

RADIUS (Remote Authentication Dial In User Service, česky Uživatelská vytáčená služba pro vzdálenou autentizaci)

4.3.2 Authorization, Authentication Accounting a RADIUS

Všechny tyto tři věci od RADIUSu požadujeme (A všechny lze obstarat s pomocí SQL), proto je vhodné načrtnout jak konkrétně RADIUS server provádí `A+A+A`. Toto je součástí protokolu RADIUS, proto, ať už se setkáte s jakýmkoliv softwarovým balíčkem RADIUS serveru, tyto principy by měly být stejné.

Výměna informací mezi klientem a serverem je řešena pomocí **`Attribute-Value`** párů (Atribut-Hodnota). Attribute je určen svým číslem, kterému odpovídá určitý název (Například `User-Password`, `User-Name`, nebo `Service-Type`). Atribut určuje způsob jakým se naloží s hodnotou, kterou nese. Pro různé druhy komunikace mezi klientem a serverem jsou používány obecně různé A/V páry. Kompletní specifikaci atributů lze najít na internetu.

A/V páry se dále, podle způsobu s jakým se s nimi pracuje, rozdělují na dvě skupiny, *Left-Hand-Side* (LHS) a *Right-Hand-Side* (RHS) atributy.

4.3.2.1 Left-Hand-Side atributy

LHS jsou atributy, které slouží k porovnávání hodnot, proti záznamům v databázi. Například k porovnání hesla poskytnutým uživatelem a tím, které je uložené v databázi.

4.3.2.2 Right-Hand-Side atributy

RHS atributy definují, co se má udělat, sedí-li porovnání LHS.

4.3.2.3 *Ověření uživatele* neboli authentication

Spočívá v tom, že uživatel poskytne o sobě všechny požadované informace, které se uloží do LHS, tyto jsou poslány serveru, který provede porovnání s údaji ve své databázi a v kladném případě následují akce definované v RHS. Ty například definují také, jak se mají porovnávat hesla, jestli například metodou CHAP, nebo jinak. Většinu RHS však pošle *NASu* (Network Access Server), který tyto informace může využít k vytvoření přístupu do sítě. A vlastně tímto proběhne též autorizace.

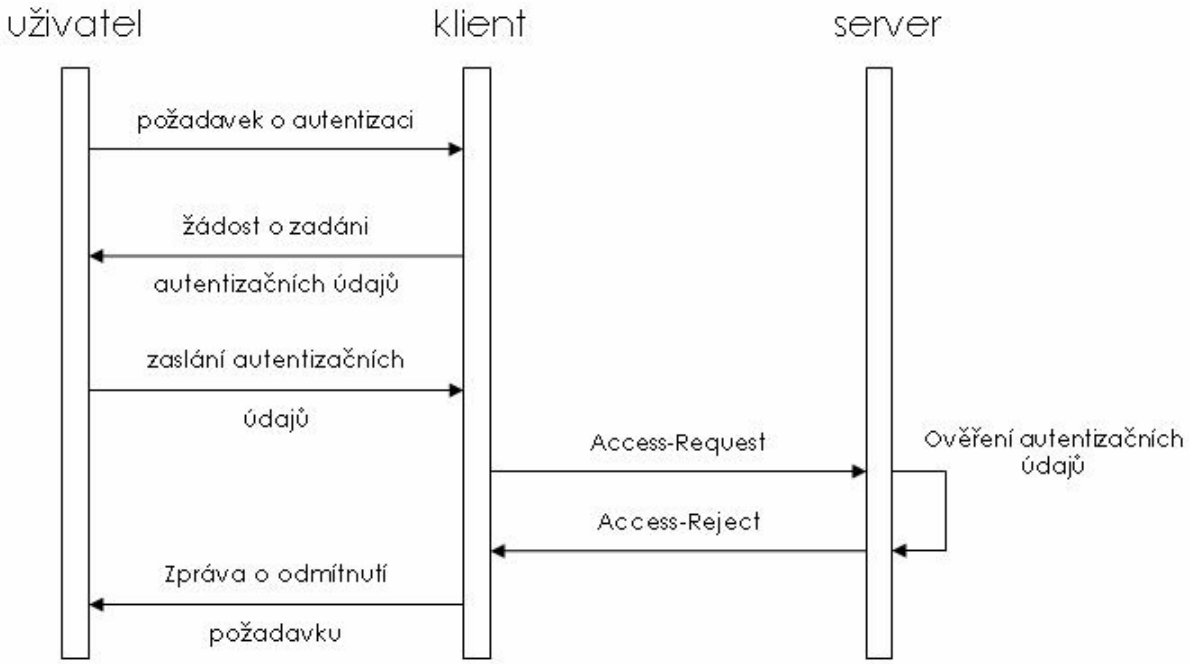
4.3.2.4 Accounting

Tedy vedení záznamů o používání účtů (časy připojení, počet přenesených bajtů ap.) Tyto informace slouží jen uživateli či správci, například k výběru jakým způsobem bude provádět accounting. Při zasílání žádosti o accounting (Account-Request packet), poskytne NAS všechny informace, samozřejmě opět v podobě `Attribute-Value` páru. Server podle uvážení provede accounting.

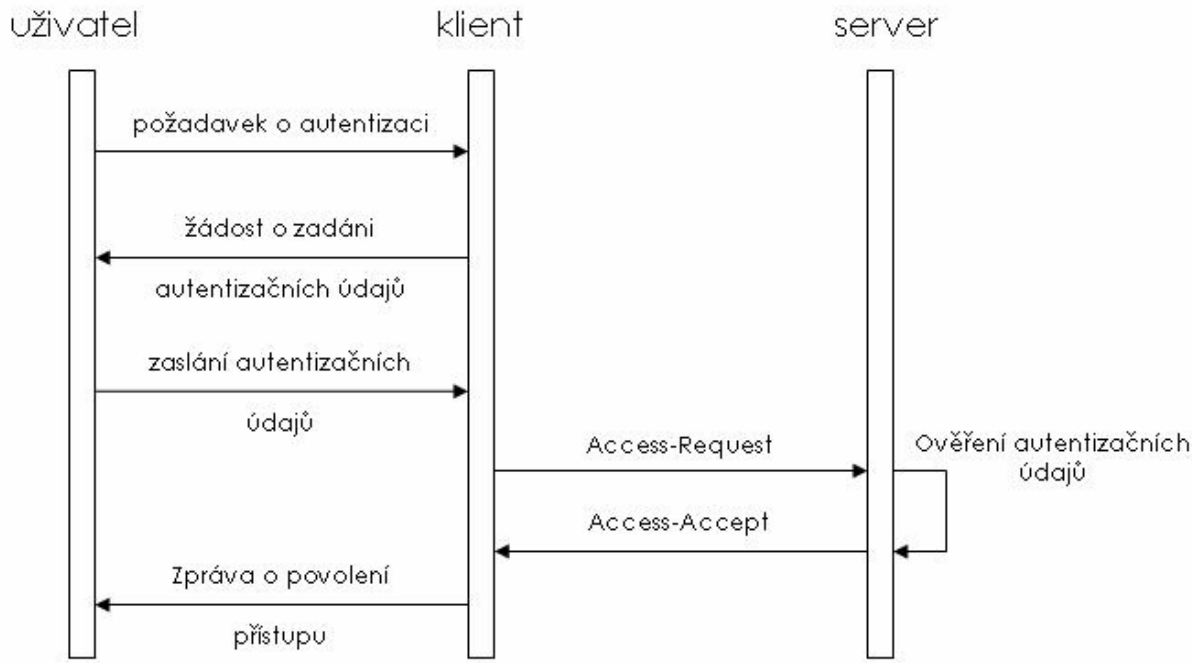
4.3.3 Funkce RADIUS v detailech

- Pokud je klient nakonfigurován k použití RADIUS protokolu, každý z uživatelů musí klientovi předat své autentizační údaje. To může být například provedeno pomocí příkazové řádky na přístupovém serveru (očekáváno od uživatele přihlašovací jméno a heslo) nebo například přes 802.1x.
- Klient informace od uživatele obdrží jednou a provede autentizaci pomoci RADIUS protokolu. To udělá tak, že klient vytvoří **Access-Request** (požadavek o přístup) , obsahující atributy uživatelské jméno, uživatelské heslo a ID portu, přes který je uživatel připojen. Pokud uživatel nějaké heslo zadal, je toto heslo ukryto metodou založenou na RSA Message Digest algoritmu MD5.
- Požadavek Access-Request je odeslán RADIUS serveru přes síť. Jestliže se nevrátí od RADIUS serveru žádná odezva v určeném čase, požadavek je opakovaně odeslán znovu. Klient může také přeposlat požadavek alternativnímu serveru nebo serverům v případě, že primární server je vypnut nebo nedostupný. Alternativní server může být použit po určitém počtu pokusů, kdy primární server selhal.
- RADIUS server příjme požadavek a ověří odesílajícího klienta. Požadavek od klienta, pro kterého RADIUS server nemá sdílené tajemství, by měl být tiše zahozen. Jestliže je totožnost klienta správná, RADIUS server se podívá do databáze uživatelů a vyhledá jméno uživatele, jež je obsaženo v požadavku. Uživatelský záznam v databázi (soubor *users* RADIUS serveru) obsahuje seznam parametrů (například uživatelova IP-adresa nebo IP-adresa RADIUS klienta, přes který se uživatel snaží přistupovat) a které musí souhlasit s údaji pro umožnění přístupu uživateli. Pro umožnění přístupu uživateli se ověřuje heslo, které může také specifikovat RADIUS klienta nebo port přístupového serveru, přes který je uživateli umožněn přístup.
- Jestliže některá z podmínek není splněna, RADIUS server odešle **Access-Reject** (zamítnutí přístupu), odezvu indikující, že tento uživatelský požadavek je neplatný. Pokud je toto požadováno, server může vložit textovou zprávu do odpovědi Access-Reject, která smí být zobrazena pomocí klienta uživateli. Žádné další atributy nejsou v odpovědi Access-Reject povoleny.

- Celý průběh komunikace mezi uživatelem, klientem a serverem je znázorněn na obrázcích. Obrázek 1 je případ, kdy nejsou splněny všechny podmínky nutné pro autentizaci a tudíž dochází k zamítnutí požadavku.

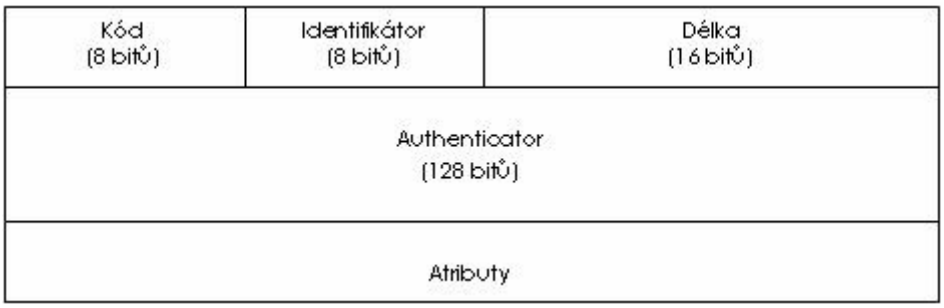


- Jestliže jsou všechny podmínky splněny, seznam konfiguračních hodnot pro uživatele je umístěn do Access-Accept odpovědi. Tyto hodnoty obsahují typ služby například: IP adresu, masku sítě, login uživatele a všechny hodnoty, které je potřeba předat požadované službě.



4.3.4 Formát RADIUS paketu

RADIUS paket je zabalen do datové části UDP segmentu, kde cílový port je nastaven na hodnotu 1812. Při generování odpovědi požadavku dojde k přehození cílového a zdrojového portu.



Kód (8 bitů) identifikuje typ RADIUS paketu. Pokud je paket přijat s neplatnou hodnotou v poli *Kód*, pak je tento paket tiše zahozen. Pole *Kód* může obsahovat následující hodnoty:

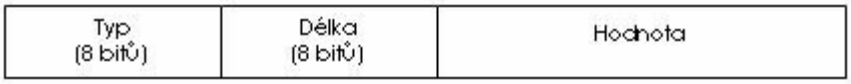
1) Access-Request - paket je odeslán RADIUS serveru a zprostředkovává informace použité pro rozhodování, zda-li má být uživateli umožněn přístup přes daný přístupový server (RADIUS klienta). Klient musí RADIUS paket odeslat s hodnotou 1 v poli Kód. Paket Access-Request musí obsahovat atributy User-Name, User-Password. Dále pak může obsahovat atributy NAS-IP Address,NAS-Identifier,NAS-Port,NAS-Port-Type.
2) Access-Accept - paket je odesílán RADIUS serverem a poskytuje specifické konfigurační informace potřebné pro službu, která je poskytována uživateli.
3) Access-Reject - při odmítnutí požadavku RADIUS server odesílá Access-Reject. Tento paket může obsahovat zprávou, která je zobrazena uživateli o zamítnutí přístupu.
4) Accounting-Request
5) Accounting-Response
6) Access-Challenge

Identifikátor (8 bitů) pomáhá při správném párování odpovídajících požadavků a odpovědí.
Délka (16 bitů) určuje velikost RADIUS paketu obsahující pole kód, identifikátor, délku, authenticator a atributy. Jestliže je paket menší, než je určeno v poli délka, může to znamenat chybu a paket může být tiše zahozen. Minimální délka je 20B, maximální délka je 4096B.
Authenticator (128 bitů - ve specifikaci 4 řady po 32bitech) jeho hodnota je použita při autentizaci odpovědi z RADIUS serveru a dále je použita při šifrování posílaného hesla.

- Request Authenticator (128 bitů)- náhodně velké číslo, obsaženo v paketu Access-Request. Tato hodnota by měla být nepředvídatelná a jedinečná po dobu existence sdíleného hesla mezi RADIUS klientem a RADIUS serverem. RADIUS klient (NAS) a RADIUS server mají sdílené tajemství. Na sdílené tajemství společné s Request Athenticatorem je aplikována jednocestná hashovací funkce MD5, pomocí které je vytvořena 128 bitů velká hodnota, která je xorována s heslem jež zadal uživatel.

- Response Authenticator - použit v Access-Accept, Access-Reject, Access-Challenge paketech a obsahuje výsledek jednocestné funkce MD5, která je počítána z RADIUS paketu, Request Athenticator z Access-Request paketu, z atributů obsažených v odpovědi následované sdíleným tajemstvím.

RADIUS atributy nesou specifické autentizační, autorizační, informační a konfigurační detaily pro požadavky a odpovědi. Konec seznamu atributů je určen délkou RADIUS paketu. Na následujícím obrázku je vidět struktura atributu.



Obrázek 8 - Struktura Atributu

Typ (8 bitů) přípustné hodnoty, které toto pole může obsahovat jsou vypsány níže společně s názvem daného atributu. RADIUS server i klient ignoruje atributy, které obsahují neznámý typ. Typ je dán atributem.
Délka (8 bitů) označuje velikost atributu zahrnující políčka typ, délka a hodnota.
Hodnota má proměnnou velikost a obsahuje informace, které jsou specifické pro tento atribut.
Políčko hodnota musí obsahovat jeden z následujících typů:

- string 1 - 253 Bytů
- adresa 32 bitů
- integer 32 bitů
- time 32 bitů

4.3.5 Atributy RADIUS protokolu

1. User-Name
2. User-Password
3. CHAP-Password
4. NAS-IP-Address
5. NAS-Port
6. Service-Type
7. Framed-Protocol
8. Framed-IP-Address
9. Framed-IP-Netmask
10. Framed-Routing
11. Filter-Id
12. Framed-MTU
13. Framed-Compression
14. Login-IP-Host

15. Login-Service
16. Login-TCP-Port
17. (unassigned)
18. Reply-Message
19. Callback-Number
20. Callback-Id
21. (unassigned)
22. Framed-Route
23. Framed-IPX-Network
24. State
25. Class
26. Vendor-Specific
27. Session-Timeout
28. Idle-Timeout
29. Termination-Action
30. Called-Station-Id
31. Calling-Station-Id
32. NAS-Identifier
33. Proxy-State
34. Login-LAT-Service
35. Login-LAT-Node
36. Login-LAT-Group
37. Framed-AppleTalk-Link
38. Framed-AppleTalk-Netw.
39. Framed-AppleTalk-Zone

40-59 rozšíření o nové typy atributu pro RADIUS Acounting

40. Acct-Status-Type
41. Acct-Delay-Time
42. Acct-Input-Octets
43. Acct-Output-Octets
44. Acct-Session-Id
45. Acct-Authentic
46. Acct-Session-Time
47. Acct-Input-Packets
48. Acct-Output-Packets
49. Acct-Terminate-Cause
50. Acct-Multi-Session-Id

51-59 Acct-Link-Count

60. CHAP-Challenge
61. NAS-Port-Type
62. Port-Limit
63. Login-LAT-Port

4.3.6 Historie RADIUS

RADIUS byl původně vyvinut společností Livingston Enterprises pro jejich PortMaster série Network Access Servers a později (1997) zveřejněny jako RFC 2058 a RFC 2059 (současné verze jsou RFC 2865 a RFC 2866).

Nyní existuje několik komerčních a open-source RADIUS serverů.Vlastnosti se liší, ale většina umožňuje dohledávat uživatele v textových souborech, LDAP serverech, různých databázích a podobně. Účtovací informace se mohou zapisovat do textových souborů, různých databází, přeposílat na externí servery a podobně. SNMP je často používáno pro vzdálené monitorování. RADIUS proxy servery s jsou používány pro centrální správu a mohou přepisovat RADIUS pakety za běhu (z bezpečnostních důvodů, nebo pro překlady mezi dialekty jednotlivých výrobců).

4.3.7 Využití RADIUS

RADIUS je jako autentizační protokol bežně používán v IEEE 802.1x bezpečnostním standardu (často používán v bezdrátových sítích). I když nebyl RADIUS původně vytvořen pro autentizační metody v bezdrátových sítích, vylepšuje WEP zabezpečení ve spojení s ostatními bezpečnostními metodami jako EAP-PEAP. RADIUS je rozšiřitelný a většina výrobců zařízení a software používají vlastní RADIUS dialekty. Oficiálně přidělené čísla portů pro RADIUS protokol jsou pro autentizaci 1812 a pro účtování 1813.

Různé implementace

- Freeradius (opensource)
- Cisco Secure Access Control Server
- Funk Steel Belted RADIUS
- Radiator
- Interlink Merit

4.4 Diameter

DIAMETER je AAA protokol (*authentication, authorization and accounting*, česky *autentizace, autorizace a účtovací*) používaný pro přístup k síti nebo pro IP mobilitu. Hlavní koncept tvoří

základní protokol, který může být rozšířen pro poskytování AAA služeb novým přístupovým technologiím. Může pracovat jak lokálně tak i v roamingu.

4.4.1 Rozšíření oproti RADIUS protokolu

RADIUS protokol je předchůdce protokolu DIAMETER (anglické *diameter* je česky *průměr*, což je dvakrát více než *poloměr*, anglicky *radius*). Diameter není přímo zpětně kompatibilní, ale poskytuje rozšířenou cestu pro RADIUS. Hlavní rozdíly protokolu DIAMETER oproti protokolu RADIUS jsou:

- používá spolehlivý transportní protokol (TCP nebo SCTP, nepoužívá nespolehlivý UDP)
- může použít zabezpečení na transportní vrstvě (IPsec nebo TLS)
- podporuje přenos RADIUS
- má větší adresní prostor pro dvojice hodnot atributů (anglicky *Attribute Value Pairs*, AVPs) a širší identifikátory (32bitové místo 8bitových)
- jde o klient-server protokol, s výjimkou podpory některých zpráv inicializovaných serverem
- lze použít stavový i bezstavový model
- má dynamické objevování uzlů (používá DNS, SRV a NAPTR)
- má schopnost vyjednávání
- podporuje dohody na aplikační vrstvě, definuje metody odolávající chybám a stavové stroje (RFC 3539)
- oznamuje chyby
- má lepší podporu roamingu
- je snadněji rozšiřitelný; lze definovat nové příkazy a atributy
- je zarovnán na 32bitové hranice
- má základní podporu uživatelských sezení a účtování

4.4.2 Popis protokolu

Základní protokol Diameteru (anglicky *Diameter Base Protocol*) je definován v RFC 3588. Určuje minimální požadavky AAA protokolu. Aplikace Diameteru (anglicky *Diameter Applications*) mohou rozšířit základní protokol přidáním nových příkazů nebo atributů. Aplikace zde není program, nýbrž protokol založený na Diameteru. Zabezpečení protokolu Diameter je poskytováno protokolem IPSEC nebo TLS.

4.5 TACACS +

TACACS+ (Terminal Access Controller Access-Control System, česky kontrolor terminálového přístupu k systému řízení přístupu) je AAA protokol poskytující řízení přístupu k routerům, serverům

pro přístup k síti a dalším síťovým zařízením, přes jeden nebo více centralizovaných serverů. TACACS+ poskytuje AAA služby odděleně.

TACACS+ je založen na protokolu TACACS, nicméně jde o zcela nový protokol neporovnatelný s žádnou předchozí verzí TACACS a spojuje je jen jméno. Není tedy zpětně kompatibilní s protokoly TACACS nebo XTACACS. TACACS+ a RADIUS většinou nahradily své předchůdce protokoly v nově tvořených nebo aktualizovaných sítích, ačkoli TACACS a XTACACS stále běží na mnoha starších systémech.

Zatímco RADIUS spojuje autentizaci a autorizaci v uživatelském profilu, TACACS+ tyto dvě operace odděluje. Dalším rozdílem je že zatímco TACACS+ používá spolehlivý Transmission Control Protocol (TCP) na portu 49, RADIUS používá nespolehlivý User Datagram Protocol (UDP). Rozšíření TACACS+ protokolu poskytují více typů authentikačních požadavků a více typů kódů v odpovědích než bylo v původní specifikaci.

TACACS+ nabízí více protokolů jako jsou IP a AppleTalk. Běžně pracuje se zcela šifrovaným tělem paketu pro bezpečnější komunikaci. Jde o Cisco vylepšení původního TACACS protokolu.

TACACS+	**RADIUS**
využívá transportního protokolu TCP	využívá transportního protokolu UDP
provádí šifrování celého paketu	šifruje pouze heslo
nezávislé na architektuře AAA	kombinace autentizace a autorizace
hesla v databázi mohou být zašifrovaná	hesla v databázi jsou nezašifrovaná

Obrázek 9

5 Postup instalace a konfigurace RADIUS serveru

5.1 Instalace s modulem LDAP

Pro testování jsem použil free linuxovou distribuci Red Hat http://www.redhat.com/ distribuce Fedora Core 4 http://fedora.redhat.com/
Instalace LINUXOVEHO SERVERU jsem provedl dle instalační příručky
http://fedora.redhat.com/docs/fedora-install-guide-en/fc4/

V instalační části jsem zvolil instalační baliček Freeradius 1.0.25 http://www.freeradius.org/
Daná baliček lze stáhnout i samostatně na stránce ftp://ftp.freeradius.org/pub/radius/freeradius-1.0.5.tar.gz
V případě stažení samostatného balíčku můžeme provést instalaci v příkazovém řádku následovně:

Do systému musíme být přihlášení jako administrátor uživatel (root)

su root

1 / Provedeme rozbalení balíčku do připraveného adresáře. Nejlépe /usr/src/radius

tar –zxvf freeradius-1.0.5.tar.gz

2 / Provedeme konfiguraci RADIUS serveru

./configure

3 / Provedeme překonfigurování závislosti přeložení.

./make

4 / Provedeme instalaci na Freeradius SERVERU na daném PC.

.make install

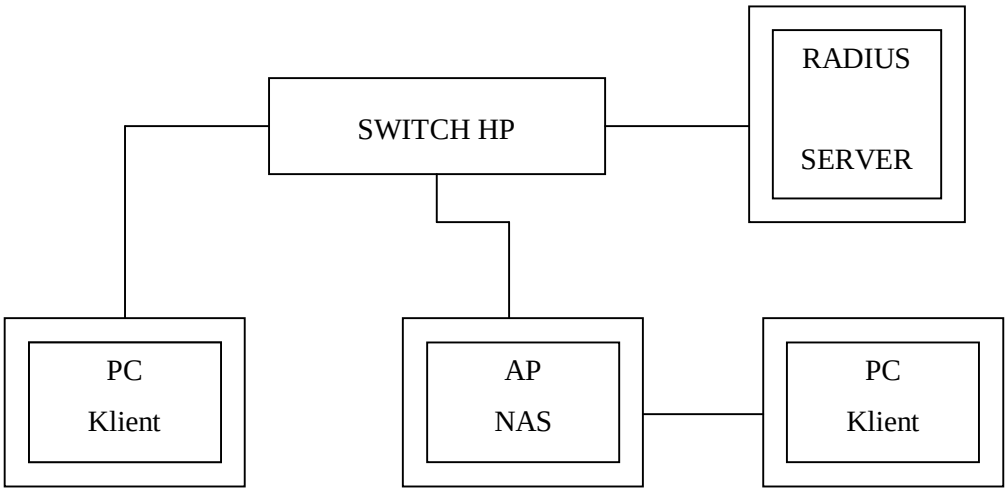
5 / Před prvním spuštěním musíme upravit konfigurační soubor

radiusd.conf

Daný soubor je velmi podrobně popsán komentáři.
Závislosti balíčků v jádru Fedora jsou již vyřešeny takže jsem pro testování využil vnitřně předkompilovaný baliček.
Freeradius se vyznačuje podporou RFC a VSA atributu jež jsou možné volit pomoci specifické konfigurace autentizace účtování.

Pro testování SERVERU v rámci vnitřní meziaplikační komunikace jsem použil na daném PC radiusclient 0.3.2 jež je též součástí vnitřního jádra distribuce Fedora.
Samostatný software lze stáhnout také na odkazu http://ftp.cvut.cz/debian/pool/main/r/radiusclient/

Zapojení při testu



5.2 RADIUS klient

Funk Odyssey

Instalace probíhá standardním způsobem. Program se sám automaticky spouští po startu Windows a schovává se v Tray. Ovládání je poměrně snadné, je rozděleno do hlavního klientského okna, kde se nastavují věci týkající se síťového rozhraní, apod., a okno na editaci profilů. V programu může být uloženo více profilů pro autentizaci (každý může využívat jiný způsob autentizace), a u každého profilu je možné nastavit odpovídající informace.

Aegis Client

Instalace probíhá také typicky. Program se také schová do traye, odkud je možné vyvolat stavové okno, ve kterém jsou vypsána všechna síťová rozhraní a jejich aktuální stav. Stejně jako u Odyssey je možno mít několik autentizačních profilů

Testování jsem prováděl na Notebooku Acer Travelmate 240 s PCMCIA kartou Compex WL 11B+

	Funk Odyssey	Aegis Client
Podporované OS	Windows XP, 2000, 98, Me, Pocket PC, Windows Mobile 2003.	Windows XP, 2000, NT, 98, ME, Pocket PC 2002 (ale i MacOS X)
Podporované metody	TTLS, PEAP, TLS, LEAP, MD5	TTLS, PEAP, TLS, LEAP, MD5
Podpora WPA	Ano (od verze 2.2) - je zmiňována pouze podpora TKIP	Ano
Cena	za 1 licenci asi 40$)	39.99$/1 licence
Web	www.funk.com	www.mtghouse.com
Pre-Config	Odyssey Client Administrator - umožňuje vytvářet "custom" verze instalátoru (možnost nastavit defaultní profil, seznam sítí, nainstalovat certifikáty serveru)	Aegis Client Enterprise Deployment Tool (Meetinghouse zmiňuje především možnost "bundlovat" klíč serveru k instalačce)

5.3 RADIUS server a databáze mySQL

Jedním z dalších možných kombinací použití je kombinace s oblíbeným databázovým serverem MySQL. Tuto kombinaci jsem také použil při implementaci informačního systému pro účtování a správu klientů bezdrátové sítě. Vybral jsem si opět instalaci Freeradius s modulem MySQL

5.3.1 Výhody RADIUS + MySQL

- Oddělení databáze uživatelů (MySQL) od mechanizmu ověřování (Freeradius) - umožňuje správci sítě mnohem flexibilnější přístup k práci. Rozhodne-li se v budoucnu změnit jeden z těchto dvou prvků, při troše štěstí nebude muset do druhého příliš zasahovat. Tento přístup také umožňuje mít více RADIUS serverů, nebo třeba záložní databáze uživatelů.
- Flexibilnější přístup k práci - Správu kont může správce přenechat jiným lidem, neboť přidávání a odebírání uživatelů pomocí php rozhranní SQL je vcelku snadné a bez rizika vůči RADIUS serveru
- MySQL je výkonný systém pro práci s rozsáhlými tabulkami -
- Možnost programového napojení na PERL, C či PHP

5.3.2 Vývojové prostředí

Konfigurace operačního systému:

Daný RADIUS server jsem testoval na této konfiguraci

- Debian GNU/ Linux, kernel 2.4.27-3-286
- MySQL 5.0.32
- Apache 2.0.54 (Debian GNU/Linux)
- PHP 5,2,1-0.dodeb.1

5.3.3 Postup instalace Freeradius-mysql

Po úspěšně zvládnuté instalaci Debianu a MySQL serveru a serveru Apache můžeme přistoupit k instalaci Freeradius serveru.

1) Do systému musíme být přihlášení jako administrátor uživatel (root)
   su root
2) Pro distribuci Debian je instalační baliček Freeradius serveru připojeného k databázi MySQL oddělen, a musíme jej tedy nainstalovat zvlášť
   $apt-get install freeradius-mysql
3) Po úspěšném nainstalování je nutno běžící Freeradius server vypnout a znovu spustit
   $freeradius –X –l stdout
4) Pokud je vše zdárně funkční měl by Freeradius napsat, že je připraven na zpracovávání požadavků.

5.3.4 Konfigurace Freeradius-mysql

Tento odstavec se věnuje popisu prostředí a konfiguračním souborům, kterým se při práci s Freeradiusem nevyhneme. Na různých počítačích se nacházejí konfigurační soubory v různých adresářích. Často je tato adresa */etc/raddb,* nebo */etc/freeradius.*

5.3.4.1 Konfigurační soubory

radiusd.conf - Soubor s hlavní konfigurací serveru. Pro nás je důležité, že mimo jiné specifikuje, jaké postupy zvolit při žádostech o authentication, authorization a accounting

users - Tento soubor slouží jako jednoduchá databáze uživatelů. My jej potřebovat nebudeme, ale je velmi vhodné se do tohoto souboru podívat. Je dobře okomentovaný a jsou v něm příklady záznamů uživatelů, které se v podstatě neliší od těch, které budeme vytvářet my v databázi MySQL

sql.conf - Velmi důležitý soubor. Definuje kde se nachází SQL (implicitně MySQL) databáze a jak z ní dostávat všechny potřebné informace. Jsou zde definovány SQL dotazy. Je vhodné pročíst si komentáře.

clients.conf - Soubor definující povolené přípojné body NAS. Nemá-li NAS záznam v tomto souboru pak s ním Freeradius nekomunikuje. Vyjímkou je, pokud je uveden v SQL databázi a Freeradius je nakonfigurován, aby se do ní díval. Implicitně je zde definován NAS *localhost*, který je zde pro účely testování.

5.3.5 Testování Freeradius-MySQL

Pokud nám tedy Freeradius server hlásí, že je schopen přijímat požadavky přistoupíme k otestování serveru

a) V souboru ***users*** vytvoříme buďto nový záznam uživatele, nebo nejste-li si jisti, odkomentujte nějaký příklad a vyzkoušet si, zda Freeradius běží opravdu dobře.

Například: steve

steve Auth-Type := Local, User-Password == "testing"
Service-Type = Framed-User,
Framed-Protocol = PPP,
Framed-IP-Address = 172.16.3.33,
Framed-IP-Netmask = 255.255.255.0,
Framed-Routing = Broadcast-Listen,
Framed-Filter-Id = "std.ppp",
Framed-MTU = 1500,
Framed-Compression = Van-Jacobsen-TCP-IP

b) Pro potřeby testování využijeme program radclient se zapnutým debugováním. První proveďte opět restart Freeradiusu. Dále pak napište:

$echo "User-Name=steve, User-Password=testing" | radclient localhost auth testing123 –xx

c) Za příkazem ***radclient*** je uveden host, v tomto případě, tedy náš počítač. Za ním je uveden identifikátor požadavku `auth`, authentication, chceme tedy uživatele ověřit. Následuje tajný

klíč, který by měl sdílet pouze NAS a Freeradius server a nikdo jiný, by jej neměl znát. Klíč `testing123` bývá již implicitně nastaven pro klienta přihlašujícího se z localhostu. Je k nalezení v ***clients.conf***. Podle něj se šifrují přenášená hesla. Přepínač `-xx`, znamená podrobné výpisy - debugging.
Pokud je vše v pořádku, zobrazí se vám podobný výpis:

Sending Access-Request of id 101 to 127.0.0.1:1812
User-Name = "steve"
User-Password = "testing"
rad_recv: Access-Accept packet from host 127.0.0.1:1812, id=101, length=71
Service-Type = Framed-User
Framed-Protocol = PPP
Framed-IP-Address = 172.16.3.33
Framed-IP-Netmask = 255.255.255.0
Framed-Routing = Broadcast-Listen
Filter-Id = "std.ppp"
Framed-MTU = 1500
Framed-Compression = Van-Jacobson-TCP-IP

Pozn.: Ujistěte se, jestli máte v konfiguračním souboru ***radiusd.conf*** v bloku *authorization* nezakomentovanou řádku se slovem: ***files*** .

5.3.6 Napojení Freeradius-MySQL na MySQL

5.3.6.1 Vytvoření databáze

Důležitý krok. Předpokládáme, že máme v systému funkční MySQL server. Freeradius dává uživateli v tomto ohledu vcelku volnou ruku. Záleží na nás, jakou strukturu tabulek si zvolíme. Je vhodné pročíst si soubor ***sql.conf***, z něj je dobře patrné co se od nás žádá.

V souboru ***sql.conf*** jsou uvedeny příklady dotazů do databáze, po jejich přečtení a po přečtení komentářů je jasně vidět, co provádějí. Pro autorizaci jsou zde dva páry dotazů:

a) **authorize_check_querry** který má získat z databáze údaje pro porovnání s LHS.
b) **authorize_reply_querry** který má následně vrátit RHS atributy a hodnoty.

Obdobně pak ***authorize_group_check_querry*** a ***authorize_group_reply_querry***.

Freeradius po nás požaduje, aby řádka výsledku dotazu byla přesně v tomto pořadí:

0. Číslo řádky, zatím Freeradius nevyužívá
1. Uživatelské jméno / jméno skupiny uživatelů
2. Název atributu
3. Hodnota atributu
4. Operátor atributu. Specifikace operátorů (=, ==, :=, ...) viz dokumentace ve freeradius/doc/

.

Ted tedy můžeme přistoupit ke tvorbě tabulek. Nabízí se nám dvě možnosti. Buďto dané tabulky vytvořit ručně nebo využít dostupný skript, který lze také stáhnout z internetových stránek (je součástí elektronické projektové dokumentace této práce – db_mysql.sql).

Vytvoří se vám několik tabulek:

mysql> show tables;

```
+--------------------+
| Tables_in_nicklaus |
+--------------------+
| nas                |
| radacct            |
| radcheck           |
| radgroupcheck      |
| radgroupreply      |
| radpostauth        |
| radreply           |
| usergroup          |
+--------------------+
8 rows in set (0.00 sec)
```

Obrázek 10

5.3.6.2 Změny konfigurace Freeradius-mysql

a) Do souboru ***sql.conf*** musíme doplnit údaje o tom, kde se naše databáze nachází a uživatelské jméno a heslo.

 # Connect info

server = "adresa_serveru"

login = "uziv_jmeno"

password = "heslo"

radius_db = "nazev_databaze"

b) Dále je nunté nakonfigurovat server, aby věděl, že má ověřovat požadavky pomocí SQL. V souboru ***radiusd.conf***. Ujistěte se, že v bloku *authorize* máte nezakomentovanou řádku s sql:

authorize {

preprocess

chap

mschap

#attr_filter

#eap

suffix

sql

#files

}

Se zakomentovaným ***files*** se server nebude po údajích poohlížet v souboru ***users***.

c) V bloku `authenticate`:

authenticate {

authtype PAP {

pap

}

authtype CHAP {

chap

}

authtype MS-CHAP{

mschap

}

#digest

#pam

#unix

#authtype LDAP {

 # ldap

#}

}

d) V *accounting*:

preacct {

acct_unique

preprocess

suffix

#files

}

accounting {

detail

#unix

sql

#radutmp

#sradutmp

}

Zbytek souboru ***radiusd.conf*** ponecháme beze změn.

5.3.6.3 Vložení dat do databáze, struktura tabulek

Je důležité vložit do databáze testovací balík dat. Například uživatel Pavlík:

1) přidělíme Pavlíkovi skupinu

mysql> INSERT INTO usergroup VALUES (0,'pavlik','students');

*mysql> select * from usergroup;*

```
+----+----------+-----------+
| id | UserName | GroupName |
+----+----------+-----------+
|  1 | pavlik   | students  |
+----+----------+-----------+
1 row in set (0.01 sec)
```

2) Vložíme údaje do ***radcheck*** tabulky, která, jak název napovídá, slouží Freeradius serveru k ověření uživatele.

mysql> INSERT INTO radcheck VALUES (1,'pavlik','Password','==','pavlik');

*mysql> select * from radcheck;*

```
+----+----------+-----------+----+--------+
| id | UserName | Attribute | op |  Value |
+----+----------+-----------+----+--------+
|  1 | pavlik   |  Password | == | pavlik |
+----+----------+-----------+----+--------+
1 row in set (0.01 sec)
```

To jaký atribut dáme v této tabulce a jaká bude jeho hodnota záleží na nás. Konkrétně v této tabulce nemáme moc na výběr a nejčastější atribut bude ***Password***, ale můžeme zde třeba mít atribut ***Auth-Type := Accept*** a tím zaručíme, že uživatel bude ověřen ať už zadá jakékoliv heslo i žádné. Chceme-li naopak konkrétnímu uživateli zakázat přístup do sítě (než si třeba odviruje počítač), musíme to udělat v tabulce ***radcheck*** s tím, že tam bude ***Auth-Type := Reject***.

3) Protože server zpravidla ještě neví jakou metodou bude hesla porovnávat, řekneme mu to v tabulce ***radgroupcheck***, protože se dá předpokládat, že metoda porovnání hesel bude pro celou skupinu uživatelů stejná.

mysql> INSERT INTO radgroupcheck VALUES (0,'students','Auth-Type',':=','Local');

mysql> select * from radgroupcheck;

```
+----+-----------+-----------+----+--------+
| id | GroupName | Attribute | op | Value  |
+----+-----------+-----------+----+--------+
|  1 | students  | Auth-Type | := | Local  |
+----+-----------+-----------+----+--------+
1 row in set (0.00 sec)
```

Local znamená, že je nám umožněno ověřovat hesla uložená v *Plain-Text* formě, tedy nijak nezašifrovaná (chceme-li použít metodu CHAP) .

4) Tabulka ***radreply***:

mysql> INSERT INTO radreply VALUES (0,'pavlik','Reply-Message',':=','Ahoj Pavle!');
*mysql> select * from radreply;*

```
+----+----------+-----------------+----+----------------------+
| id | UserName | Attribute       | op | Value                |
+----+----------+-----------------+----+----------------------+
|  1 | pavlik   | Reply-Message   | := | Ahoj Pavle!          |
+----+----------+-----------------+----+----------------------+
       1 row in set (0.01 sec)
```

Tato tabulka slouží k určení atributů a hodnot, které jsou pro daného uživatele specifické a nevztahují se na celou skupinu uživatelů. Příklad využití je, chceme-li konkrétnímu uživateli přidělit IP adresu napevno. ***Framed-IP-Address := 47.238.77.117***

5) Tabulka ***radgroupreply***
Obsahuje informace následně posílané NASu. Jsou to informace převážně definující spojení, které má NAS vytvořit.

mysql> INSERT INTO radgroupreply VALUES (0,'students','Framed-Ip Address',':=','255.255.255.254',0);
mysql> INSERT INTO radgroupreply VALUES (0,'students','Framed-Ip-Netmask',':=','255.255.255.0',0);
*mysql> select * from radgroupreply;*

```
+----+-----------+-------------------+----+-----------------+------+
| id | GroupName | Attribute         | op | Value           | prio |
+----+-----------+-------------------+----+-----------------+------+
|  1 | students  | Framed-Ip-Address | := | 255.255.255.254 |    0 |
|  2 | students  | Framed-Ip-Netmask | := | 255.255.255.0   |    0 |
+----+-----------+-------------------+----+-----------------+------+
              2 rows in set (0.00 sec)
```

6) Tabulka ***radpostauth***

Je určena k logování všech přicházejících požadavků o autorizaci. Údaje do ní vkládá server příkazem ***postauth_querry***. Pokud je k tomu server nakonfigurován, což implicitně není. Dá se to nastavit v souboru radiusd.conf

5.3.6.4 Accounting

S accountingem je to velice jednoduché. Tabulku plní server, jakmile mu přijdou požadavky na accounting. Potom toto obstarává voláním těchto dotazů:

accounting_onoff_query - query pro Accounting On/Off pakety
accounting_update_query - query pro Accounting update pakety
accounting_update_query_alt - query pro Accounting update pakety
(alternativní v případě, že první selhala)
accounting_start_query - query pro Accounting start pakety
accounting_start_query_alt - query pro Accounting start pakety
(alternativní v případě, že první selhala)
accounting_stop_query - query rpo Accounting stop pakety
accounting_stop_query_alt - query pro Accounting start pakety
(alternativní v případě, že první selhala)

5.3.6.5 Test ověřování a accountingu

1) Pokud RADIUS server běží, ukončete jej a spusťte v *debugg* módu.
$freeradius -X -l stdout

2) Přepněte se do jiné konzole a napište příkaz:
$echo "User-Name=pavlik, CHAP-Password=pavlik" | radclient localhost auth testing123 –xx

Atributem `CHAP-Password` dáváme najevo, že požadujeme použít metodu CHAP pro přenos hesla přes síť. Pokud vše proběhlo v pořádku, dostanete podobný výpis:

```
Sending Access-Request of id 154 to 127.0.0.1:1812
                User-Name = "pavlik"
                CHAP-Password = "0x8aaf2db342d132894d6f5d46ae00d20156"
rad_recv: Access-Accept packet from host 127.0.0.1:1812, id=138, length=33
                Reply-Message = "Ahoj Pavle!"
                Framed-IP-Address = 255.255.255.254
                Framed-IP-Netmask = 255.255.255.0
```

3) Pošleme serveru žádost o provedení accountingu.

$echo "User-Name=pavlik, Acct-Status-Type=Start" | radclient localhost acct testing123 –xx

Podle atributu `Acct-Status-Type=Start`, pozná server, že má začít s accountigem. Že se tak stalo, poznáme z výpisu, kde se uvádí, že server poslal *Accounting-Response packet*

```
Sending Accounting-Request of id 182 to 127.0.0.1:1813
                User-Name = "pavlik"
                Acct-Status-Type = Start
rad_recv: Accounting-Response packet from host 127.0.0.1:1813, id=140,
length=20
```

4) Chvíli vyčkáme a pošleme žádost o ukončení accountingu.

$echo "User-Name=pavlik, Acct-Status-Type=Stop" | radclient localhost acct testing123 -xx

```
Sending Accounting-Request of id 184 to 127.0.0.1:1813
                User-Name = "pavlik"
                Acct-Status-Type = Stop
rad_recv: Accounting-Response packet from host 127.0.0.1:1813, id=184,
length=20
```

tabulka ***raddacct***:

mysql> select UserName, AcctStartTime, AcctStopTime from radacct;

```
+----------+---------------------+---------------------+
| UserName | AcctStartTime       | AcctStopTime        |
+----------+---------------------+---------------------+
| pavlik   | 2007-05-25 01:22:47 | 2007-05-25 01:27:07 |
+----------+---------------------+---------------------+
1 row in set (0.00 sec)
```

V tabulce je několik sloupců s informacemi jako je počet přijatých, či odeslaných oktetů, číslo spojení a mnohé další.

5.3.7 Doplňující informace

Je možnost udržovat seznam přípustných klientů, tedy NASů též v databázi MySQL. Stejný skript, který nám vytvořil tabulky pro AAA nám také vytvořil tabulku `nas`. Do té můžeme všechny informace o našich NASech vložit a potom říct serveru, že se má dívat i do MySQL. V konfiguračním souboru `sql.conf`, zajistíme aby tam byla řádka

readclients = yes

například když používám localhost

mysql> INSERT INTO nas VALUES (0,'127.0.0.1','localhost','other',NULL,'t4jn3',NULL,'RADIUS Client');

*mysql> select * from nas;*

Vypíše nám tabulku ***nas***.

5.3.7.1 Uchovávání zašifrovaných hesel

Někdy je samozřejmě potřeba (z různých bezpečnostních důvodů) uchovávat hesla v databázi zašifrovaná. Toto rozhodnutí je potřeba důkladně zvážit, protože jsou-li hesla zašifrována, znemožňuje to pro ověřování použít metodu CHAP. To znamená, že se hesla mezi serverem a Klientem přenášejí v otevřené formě.

Uchovávání hesel v zašifrované formě vyžaduje tři jednoduché kroky:

1) Upravit tabulku `radcheck`. Ve sloupci ***Attribute***, místo dosavadního ***'Password'*** bude hodnota ***'Crypt-Password'***

UPDATE radcheck SET Attribute='Crypt-Password' WHERE UserName='pavlik';

2) Zašifrovat heslo. Samozřejmě potřebujeme heslo ještě zašifrovat. To je potřeba udělat unixovskou metodu, kterou nám MySQL poskytuje funkcí ***ENCRYPT.***

UPDATE radcheck SET Value=ENCRYPT('pavlik') WHERE UserName='pavlik';

3) Upravit ***radgroupcheck*** tak, že skupina uživatele, který má zašifrované heslo, bude mít pro atribut ***Auth-Type hodnotu 'Crypt-Local'***.

UPDATE radgroupcheck SET Value='Crypt-Local' WHERE GroupName='students';

*mysql> select * from radgroupcheck;*

6 Proprietární mechanismy Cisco

Základní rozdíl mezi dnes běžně dostupnými prvky levných výrobců a značkovými produkty výrobců zaměřených na podnikovou sféru je především v úrovni bezpečnosti. Levní výrobci těží ze standardů, které se snaží dodržovat, ale nerozvíjejí je a spoléhají na mechanismy, které jsou již dávno překonané. Progresivní výrobci (Cisco, HP, aj.) se snaží implementovat mechanismy, které odstraňují známé problémy na základě nových standardů (802.1x), případně definují nové mechanismy, které mají potenciál stát se základem dalších standardů. Tím, že jejich výrobky a mechanismy jsou žádané, dochází k jejich přejímání dalšími výrobci a tím i dalšímu nárůstu množství produktů podporujících „nestandard“ a tím zmírnění negativních vlastností používání nestandardu (viz. např. LEAP).

6.1.1 TKIP

Proprietární mechanismus fy Cisco nazývaný **TKIP** (Temporal Key Integrity Protocol) zajišťuje zlepšení bezpečnosti šifrování dat na bezdrátových sítích – tedy snaží se eliminovat základní nedostatky protokolu WEP. Jeho lepší bezpečnost je zajištěna použitím těchto mechanismů:

- **PPK** (*per-packet key hashing*), umožňuje změnu klíče pro každý paket; tím je odstraněna slabina standardní definice WEP, jež pracuje se statickým klíčem, ten se během spojení nemění (a není-li použit protokol 802.1x se vynuceným pravidelným ověřováním, nemění se dlouhodobě).
- **MIC** (*Message Integrity Check*), je v podstatě digitální podpis nesený v každém paketu; tím je odstraněna možnost útoku nazývaného "man-in-the-middle", tedy takového útoku, kdy útočník zachytává pakety od vysílajícího, modifikuje je a posílá příjemci.
- **rotace broadcastových klíčů,** PPK zajišťuje změnu klíčů pro unicastovou komunikaci; protože je 802.11 založen (jako všechny 802.x sítě) na broadcastovém mechanismu, je nutné zajistit změnu i klíčů používaných pro broadcasty a multicasty.

6.1.2 Cisco Identity Based Networking Services

Autentizační funkce v LAN přepínačích Catalyst, bezdrátových AP Aironet a autentizačních serverech Cisco Secure ACS jsou rozvíjeny v rámci programu Cisco Identity Based Networking Services (IBNS). IBNS zajišťují jednotnou implementaci autentizačních funkcí založených na 802.1x protokolu v bezdrátových i přepínaných sítích. IBNS využívají existující standardy (802.1x, EAP, EAP-TLS, RADIUS), podporují rozvoj nových doporučení (LEAP, EAP-FAST), zajišťují integraci autentizačních funkcí se síťovými a bezpečnostními vlastnostmi síťových zařízení.

6.1.3 Cisco Wireless Security Suite (SWSS)

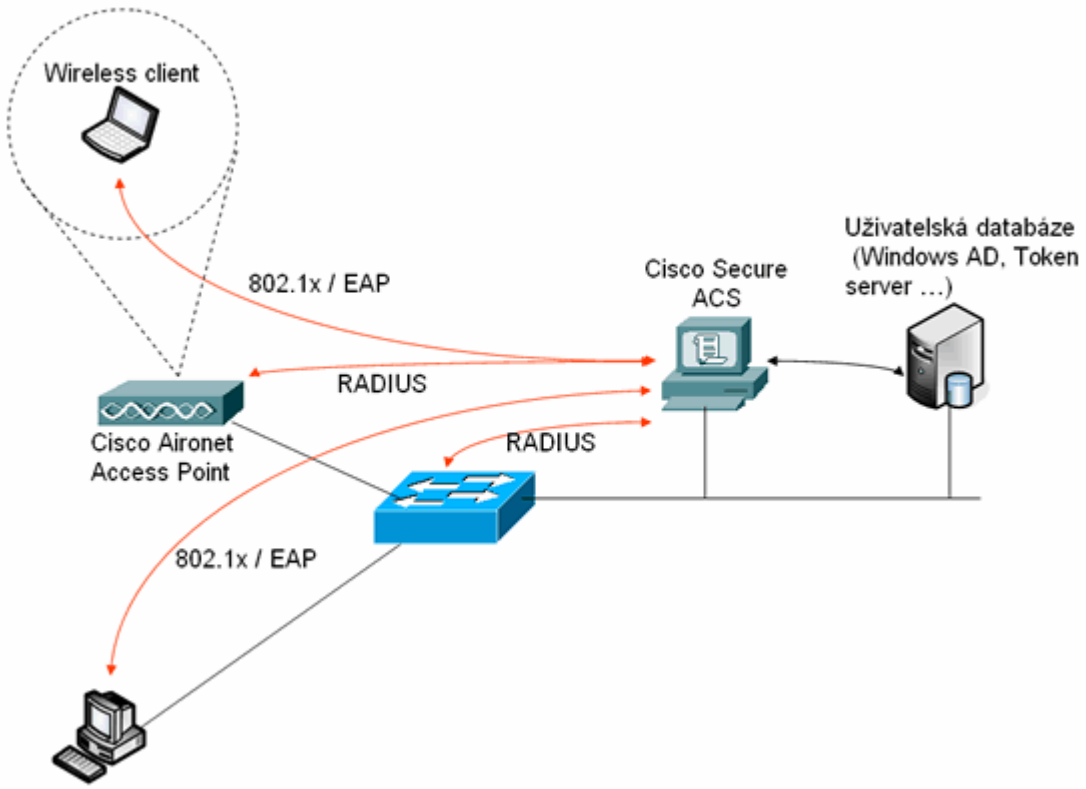
Sada technologii, kterými jsou vybaveny produkty Cisco Aironet. Pomocí standardizovaných prostředků, specifikací i firemními rozšířeními umožňuje realizaci bezpečné podnikové bezdrátové sítě.

Mezi rysy, které jsou v rámci CWSS podporovány, patří:

- Autentizace pomocí 802.1x a EAP.
- Mezi podporované EAP typy patří: Cisto EAP (LEAP), EAP-FAST, EAP-TLS, PEAP, EAP-TTLS a EAP-SIM.
- Autentizace je zajištěna pomocí centrálního autentizačního a autorizačního RADIUS serveru, jakým je např. Cisto Secure ACS.
- Enkrypce pomocí WEP, Cisto TKIP, WPA TKIP a AES.
- Rozšíření TKIP o message integrity check, initialization vector hashing, broadcast key rotation
- IEE 802.11i a WPA2.

6.1.4 Cisco Structured Wireless - Aware Network (SWAN)

Koncept pro správu bezdrátové sítě. Poskytuje stejnou úroveň bezpečnosti, rozšiřitelnosti a spolehlivosti, jakou známe u klasických LAN sítí. SWAN dokáže zajistit plynulý a velmi dobře zabezpečený přechod mezi access pointy jak v rámci jednoho subnetu (L2 roaming), tak i mezi nimi (L3 roaming). Rovněž je schopen rychle detekovat neautorizované (rogue) access pointy. Významnou součástí architektury SWAN je Cisco Works Wireless Solution Engine (WLSE), realizující management sítě.



Obrázek 11 - princip ověřování v sítích Cisco

7 RADIUS server a OS Mikrotik

Mikrotik je operační systém pro PC routery, který je dnes dostupný za přístupnou cenu. Nejedná se tedy o freeware, ale o komerční záležitost. Tento systém také podporuje technologii 802.1x. Nejvhodnější je použití kombinace RADIUS server + MySQL databáze, kde je nastavení připojení a vlastností triviální. Nicméně OS Mikrotik podporuje i další typy autentizačních serverů ne jen freeradius, ale i také Kerberos aj.

7.1.1 Využití RADIUS serveru v systému Mikrotik

- Centralizovaná autorizace bezdrátových klientů pomocí MAC adres.
- Centrální ověřování a účtování klientů.
- Je možno jej použít na všech zařízeních, které podporují RADIUS.
- RADIUS server je možné na Mikrotiku použít pro ověřování těchto služeb a protokolů:
  - PPP, PPPoE, PPTP, L2TP
  - Hotspot
  - Login
  - Wireless
  - Telephony

7.1.2 Slovník pojmů

NAS - Network Access Server

- Přístupová jednotka, která ověřuje uživatele proti RADIUS databázi (MIKROTIK).
- Je definována pomocí IP adresy a secret

REALM- doména, do které účet patří

- Vhodné použít pro rozlišení jednotlivých Mikrotik serverů (pokud je jich více v síti)
- REALM je NAS jednotkou doplněn za uživatelské jméno do tvaru “user@realm”

Secret

- heslo pro přístup NAS jednotky k RADIUS serveru

7.1.3 Access-Request atributy:

Service-Type – vždy hodnota "Framed" (pouze pro PPP)

Framed-Protocol – vždy je hodnota "PPP" (pouze pro PPP)

NAS-Identifier – identifikace routeru

NAS-IP-Address - IP adresa routeru

NAS-Port-Type

- async PPP - "Async“
- PPTP a L2TP - "Virtual";
- PPPoE a HotSpot - "Ethernet"

Calling-Station-Id

- PPPoE – MAC adresa klienta
- PPTP and L2TP – veřejná IP adresa klienta
- HotSpot - MAC adresa klienta, příp. IP adresa

Called-Station-Id

- PPPoE – název služby
- PPTP and L2TP – IP adresa serveru
- HotSpot - MAC adresa interface hotspotu, případně IP adresa interface hotspotu

NAS-Port-Id - async PPP

- PPPoE – název ethernetového interface
- HotSpot – název hotspot interface
- PPTP and L2TP není použit

Framed-IP-Address - IP adresa hotspot klienta

User-Name - login

Realm

7.1.4 Mikrotik OS a podpora různých RADIUS serverů

Nejvhodnější se zdá použití kombinace RADIUS server + MySQL databáze, kterou jsem zvolil. Nicméně OS Mikrotik podporuje i jiné produkty:

Komerční produkty: RADIATOR, EVOLYNX, ARadial, atp.

Open source: Freeradius, Openradius, a řada dalších nevyvíjených serverů.

7.1.5 Nejpoužívanější atributy

Rate-Limit - limity rychlosti pro uživatele

- formát:
  - **rx-rate[/tx-rate] [rx-burst-rate[/tx-burst-rate] [rx-burst-threshold[/tx-burst-threshold] [rxburst-time[/tx-burst-time]]]]**
- pro zadávání je možno použít jednotky:
  - 'k' (1,000x)
  - 'M' (1,000,000x)
  - např. **`64k/128k 256k/256k 128k/128k 10/10`**
- není-li zadána hodnota rx-rate, použije se hodnota tx-rate
- není-li zadána hodnota pro burst, použijí se hodnoty rxrate a tx-rate

Framed-IP-Address

- IP adresa uživatele
- využití v tabulce RADREPLY

Framed-IP-Netmask

- síťová maska klienta
- využití v tabulce RADREPLY

Framed-IP-Pool

- Název poolu (na routeru) pro přiřazení adresy z DHCP

Recv-Limit

- limit dat pro klienta

Xmit-Limit

- limit dat pro klienta

Acct-Interim-Interval

- Interim update pro klienta

Acct-Session-Time

- délka připojení v sekundách

Acct-Input-Octets

- množství stažených dat (B)

Acct-Output-Octets

- množství odeslaných dat (B)

AcctTerminateCause

- příčina ukončení spojení (dle RFC2866)

Další atributy naleznete v dokumentaci OS Mikrotik. http://www.mikrotik.com .

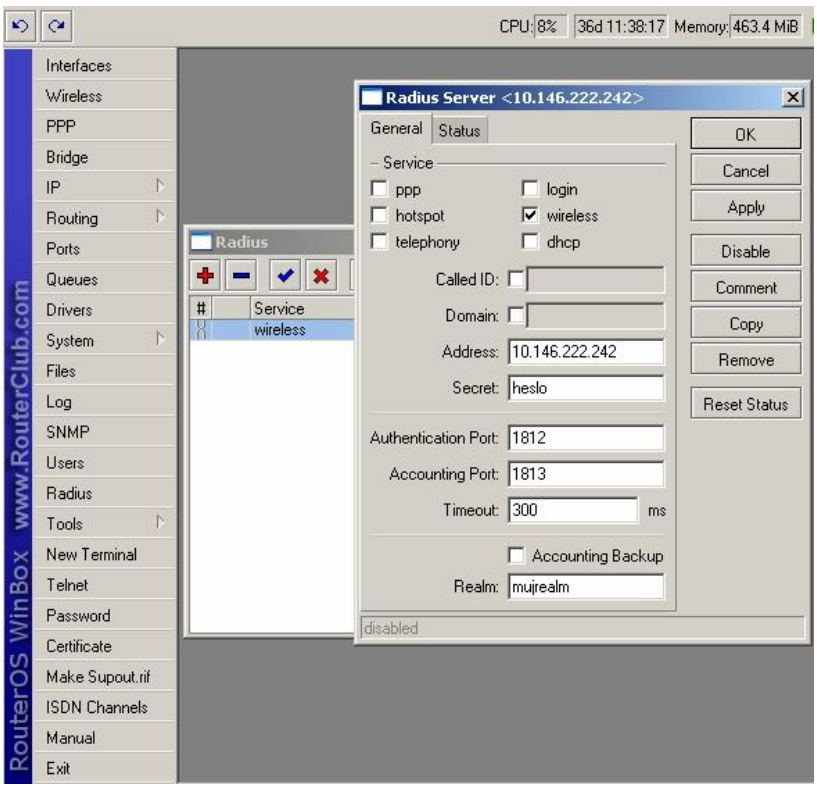
7.1.6 Ověřování MAC adres v Mikrotiku

Při ověřování MAC adres je jako UserName jako LHS atribut odesílána MAC adresa. V databázi radcheck musí být záznam pro MAC adresu. Např:

```
+-----+---------------------------+-----------+----+---------+
| id  | UserName                  | Attribute | op | Value   |
+-----+---------------------------+-----------+----+---------+
| 192 | michal@mujrealm           | Password  | == | mich@l  |
| 192 | 00:11:85:1E:D7:36@mujrealm | Password  | == |         |
+-----+---------------------------+-----------+----+---------+
```

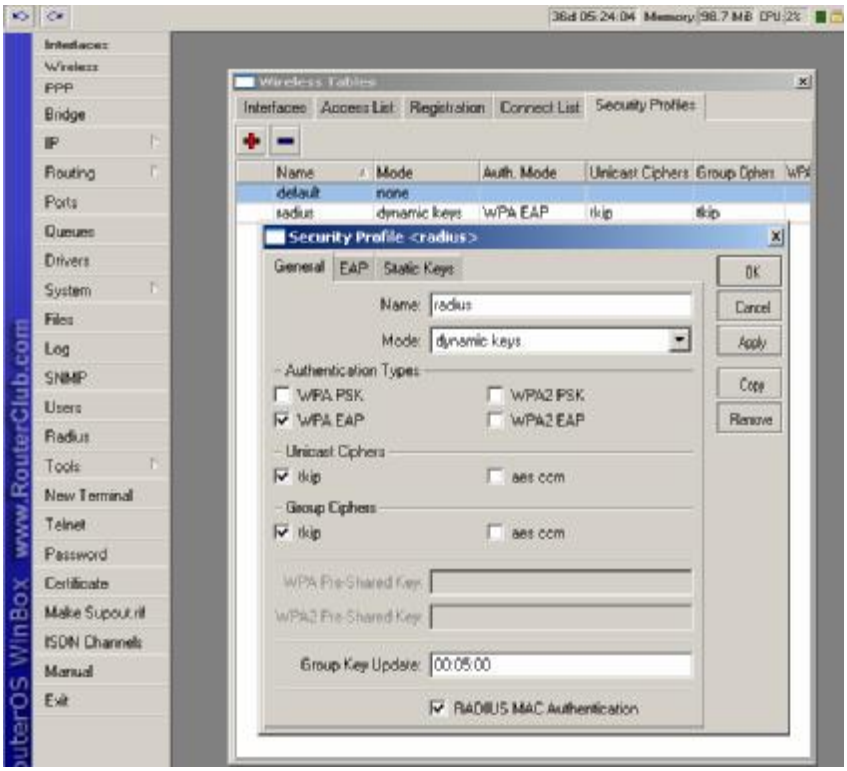
Nastavení na Mikrotiku:

/radius –> Add

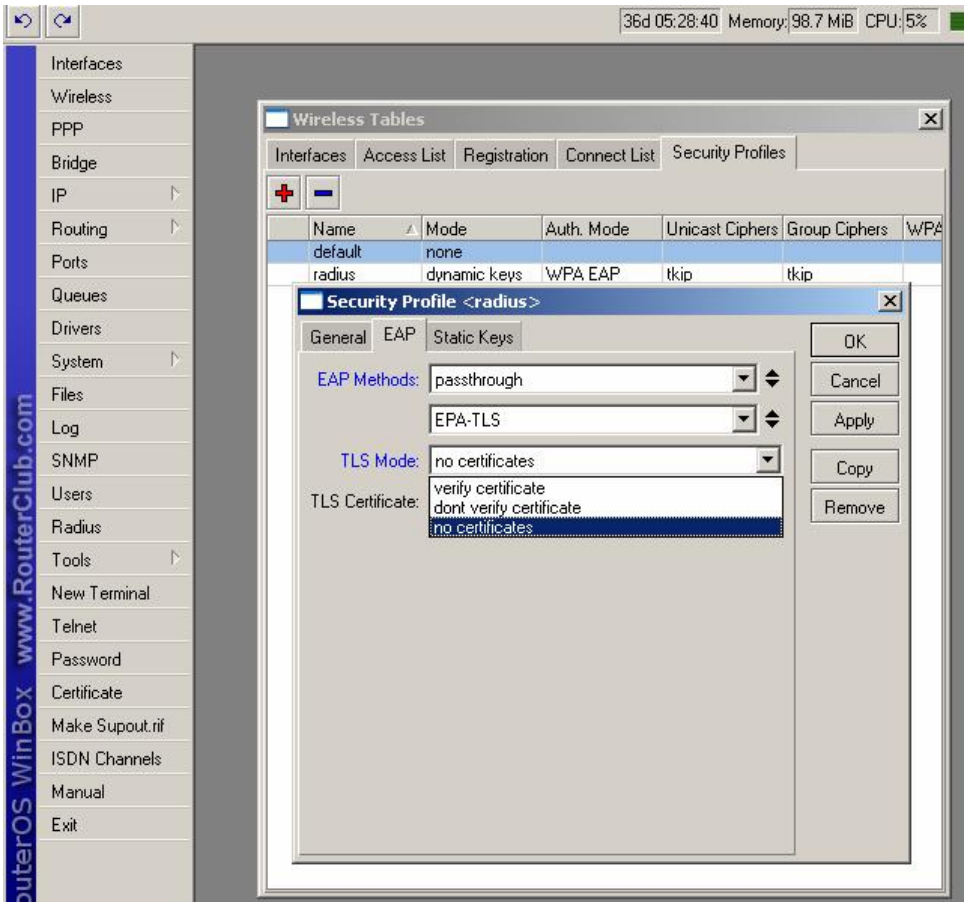


Hodnota timeout označuje prodlevu mezi opakováním požadavku. Hodnota Called ID : **PPPoE** – název služby, **PPTP** – IP adresa serveru, **L2TP** – IP adresa serveru.

/Wireless -> Security Profiles-> Add -> Radius MAC Authentication



V záložce EAP pak můžeme nastavit další atributy, zda je požadován certifikát či použít EAP-TLS.



7.2 Další použitelná zařízení

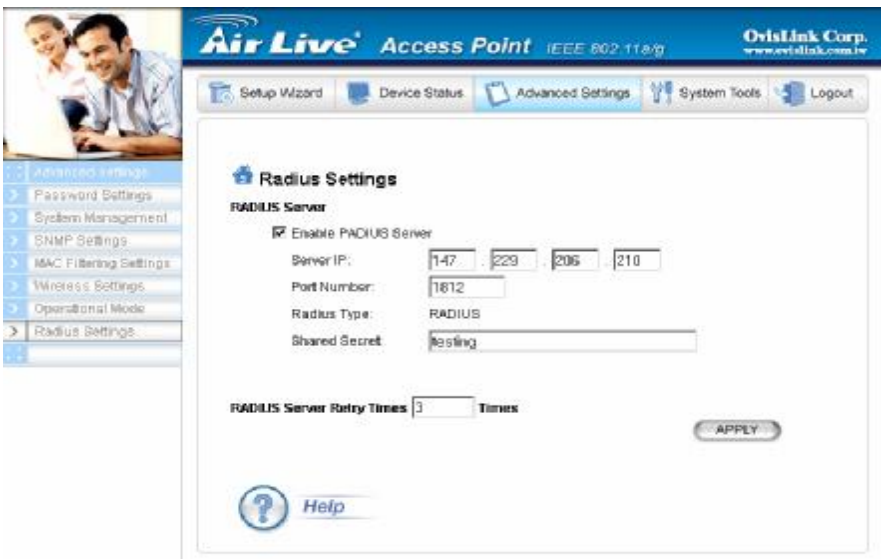
Nastavení v jednotlivých Access Pointech jsou jednoduché.

7.2.1 Ovislink 5000AP

Obrázek 12

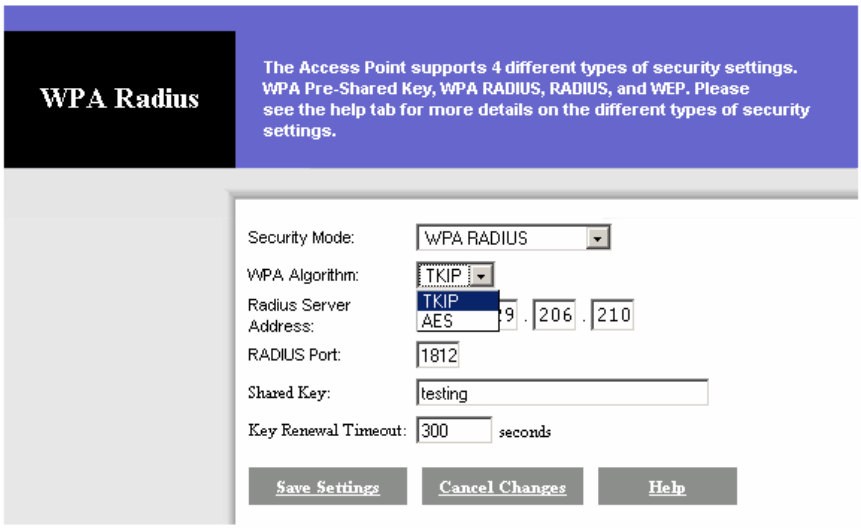
Vlastnosti:

Zařízení je funkční pouze s autentizačním serverem typu RADIUS a zvládá autentifikaci pomocí MAC nebo EAP.



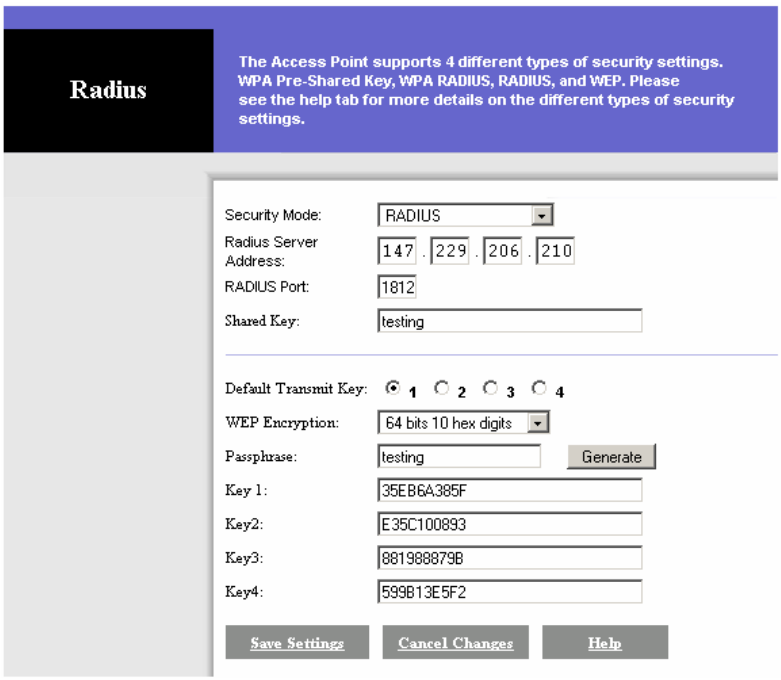
Obrázek 13

7.2.2 Linksys WAP54G



Obrázek 14

Vlastnosti:



Obrázek 15

7.3 Vliv rotace klíčů na rychlost přenosu

Pro testování je nutné nainstalovat na náš server spolu s Freeradiusem a servery Apache a MySQL také sadu certifikátů openssl, a nastavit freeradius v konfiguračním souboru httpd.conf na podporu autentifikace EAP, popřípadě EAP – TTL. Toto nastavení je netriviální a podařilo se mi na několikátý pokus.

7.3.1 Konfigurace zařízení pro test

Server: athlon 2200+, 768MB RAM, 266MHz

NAS: Mikrotik Router Board 532A

Test PC: Intel Celeron Duo Core 1600, 1GB RAM, 533MHz

7.3.2 Výsledek testů

Výsledek testů potvrdil, že rotace klíčů nemá zásadní vliv na přenosovou rychlost. Zásadním vlivem rozumíme omezení přenosové rychlosti v řádu Mbits. Testy byli provedeny pomocí přenosu souboru o velikosti 10MB ze zdrojového serveru na testovací PC pomoci FTP protokolu.

typ šifrování	AES-CCM	AES-TKIP	WEP	bez šifrování	AES2
síla signálu	-45dBm	-45dBm	-45dBm	-45dBm	-45dBm
přenosová rychlost	6,28Mbit	6,30Mbit	6,20Mbit	6,31Mbit	6,30Mbit
fyzická rychlost	54Mbit	54Mbit	54Mbit	54Mbit	54Mbit
standard	802.11g	802.11g	802.11g	802.11g	802.11g

Tabulka 3

8 Informační Systém pro správu a účtování uživatelů Wi-fi sítě pro neziskové organizace – FreeNetIS

Jako jeden z bodů své diplomové práce je navrhnout informační systém pro správu, účtování přenesených dat a také jej implementovat. Jelikož již čtvrtým rokem působím u neziskového sdružení provozující bezdrátovou síť s přístupem do internetu jako správce sítě, odhodlal jsem se tedy spojit tuto moji činnost s diplomovou prací.

8.1 Cíl projektu a srovnatelné dostupné systémy

Cílem je co nejvíce zjednodušit správu (bez)drátové sítě s orientací na uživatele. Urychlí registraci a prvotní spuštění uživatelského účtu, které může být plně automatické. Projekt je primárně určen pro neziskové organizace popř. malé ISP. Zdrojové kódy budou šířeny pod licencí GNU/GPL.

Nejde o žádnou díru do světa, podobné systémy fungují ve všech komerčních sítích – FreeNetIS je však speciální proto, že bude šířen s licencí GPL a také proto, že bude ušitý namíru freenetové síti, provozované občanským sdružením.

8.2 Vlastnosti systému (features)

Systém FreeNetIS kompletně automatizuje provoz freenetové sítě v následujících činnostech:

1) **registrace nových uživatelů sítě**
   a) Systém poskytuje funkci pro registraci uživatele, jejímž výsledkem je vytvoření uživatelského účtu a poskytnutí přihlašovacího jména/hesla novému uživateli
   b) Systém poskytuje funkci pro generování účtů street access uživatelů s časově/funkčně omezeným přístupem

2) **přihlašování uživatelů do sítě** – do sítě budou mít pouze uživatelé s platným jménem a heslem. Přihlašování do sítě může být realizováno přes:

a) Autentizace bude založena na infrastruktuře RADIUS a bude podporovat kterýkoli autentizační protokol, jehož implementace s RADIUSem spolupracuje (PPPoE, PPTP, 802.1x, atd.)

3) **správa plateb členských příspěvků**
   a) Systém pravidelně kontroluje příchozí platby na bankovním účtu a ukládá je do databáze plateb.
   b) Uživatelům, kteří včas nezaplatí příspěvky, deaktivuje účet.
   c) Min. 1 měsíc před deaktivací systém uživateli začne pravidelně zobrazovat výzvy k platbě pomocí funkce „Web redirection messaging“.
   d) Každý uživatel bude po přihlášení přístup ke svému uživatelskému účtu, kde si může ověřit zaplacení členských příspěvků, délku svého předplaceného období atd.

4) **Při prvním přihlášení po zaplacení vstupního členského příspěvku systém**
   a) Vygeneruje pro daného uživatele DNS záznam pro jeho tunelovací (PPPoE/PPTPD) adresu a tuto adresu mu natrvalo přidělí v nastavení jeho Radius účtu.
   b) „Fyzické“ IP adresy, přidělené DHCP serverem na daném AP, by bylo možné přidělovat automaticky také, ale systém by se musel na dané AP přihlásit pomocí SSH.

Je třeba vyzkoušet, co se stane, pokud se uživatel přihlásí stejným jménem/heslem ze 2 počítačů najednou. Dostane zřejmě stejnou IP, a pokud mu nebude síť kvůli tomu fungovat, musí být možné jednomu uživateli přiřadit 2 a více jmen/hesel, spojených pod 1 účet
systém pro nový uživatelský účet vytvoří frontu QoS, která bude společná pro IP všech zařízení, která daný uživatel má.

Stejně tak by měl vytvořit frontu QoS i na Apčku, protože provoz po lokále se k centrálnímu routeru často vůbec nedostane.

5) **Databáze plateb** bude **společná se systémem EkonomOS** – což je účetní SW pro podvojné účetnictví, postavený na databázi SQL. To umožní členům Správní rady kontrolovat stav financí ve sdružení mnohem jednodušeji než dnes, kdy s účetním SW může pracovat pouze 1 osoba (Money S3 není stavěné pro nasazení v distribuovaných sítích, datové soubory má pouze na lokálním disku).

6) **Zasílání zpráv uživatelům přesměrováním webu** („Web Redirection Messaging“).
   a) Jednotlivé zprávy budou mít tyto příznaky:
      i) zobrazit všem nepřihlášeným

ii) zobrazit všem přihlášeným
iii) zobrazit jenom konkrétnímu přihlášenému uživateli
iv) zobrazit seznamu uživatelů nebo uživatelům v určitém rozsahu IP adres
v) datum/čas začátku a konce zobrazování
vi) maximální počet zobrazení danému uživateli
vii) pravidelnost zobrazování (kolikrát za měsíc)

a) systém musí u každého uživatele sledovat, které zprávy a kolikrát už mu byly zobrazeny (implementace scheduleru zpráv, který bude plánovat zobrazení zpráv pro jednotlivé uživatele)
b) U každé zprávy bude odpovědní formulář, který pošle autorovi zprávu na Jabber a email.
c) doplňkovým nástrojem k FreeNetIS, by měly být skripty, běžící na všech AP, které při výpadku spoje do zbytku sítě přesměrují uživatele na stránku s hlášením o výpadku – pozor, kromě webu je nutno přesměrovat i DNS a upravit jeho nastavení tak, aby pro všechny DNS dotazy vracelo IP adresu access pointu! (v Linuxu bez problémů, v Mikrotiku asi těžko – jedině použít Linux DNS připojené přímo k Apčku)

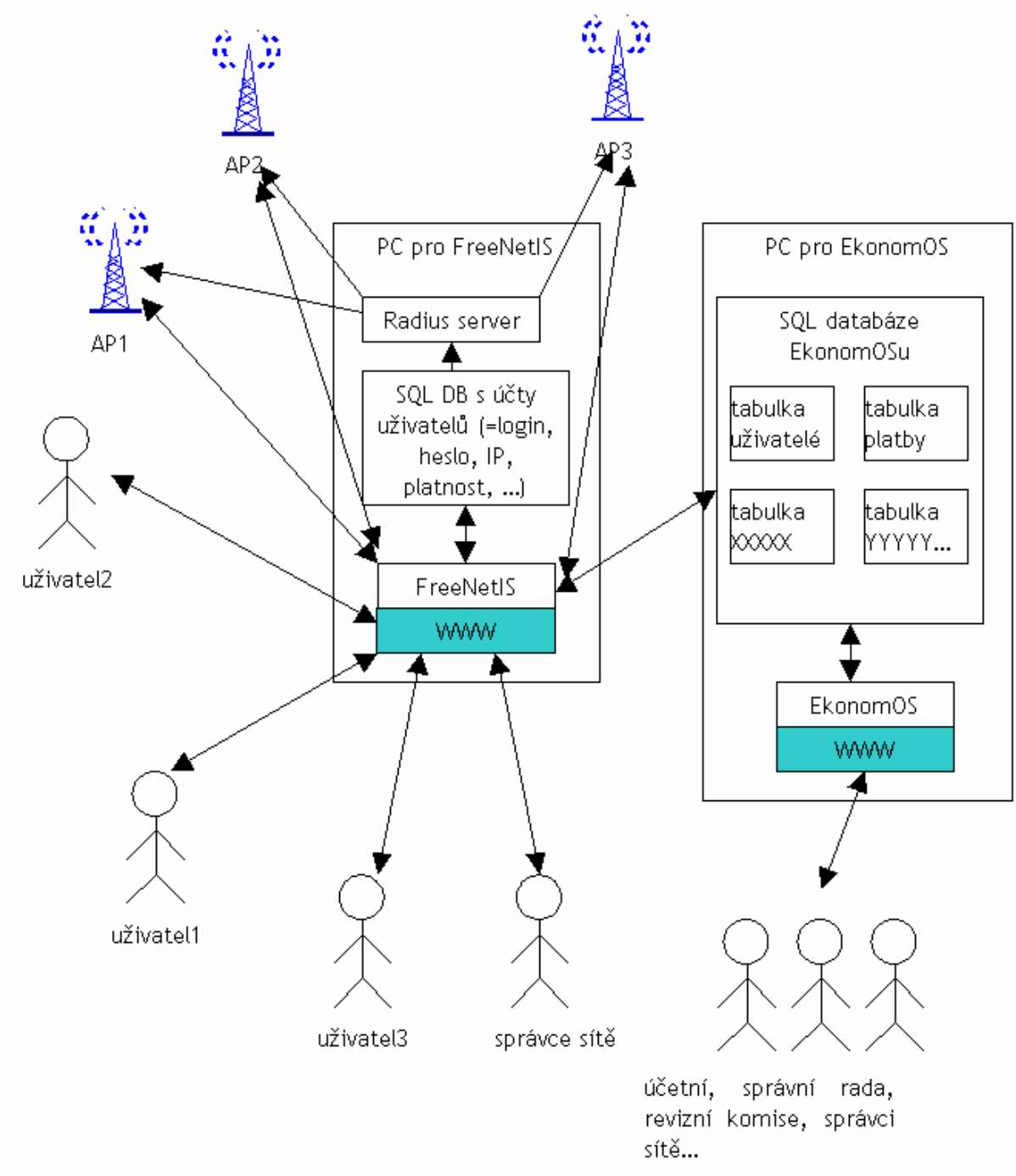
7) **Monitoring –** přenesená data, síla signálu, ztráty ping, atd.
8) **Správa uživatelských účtů** – v tomto modulu se budou „sbíhat“ informace ze všech předchozích modulů. Administrátoři systému budou moci měnit nastavení účtu všech uživatelů, uživatelé budou moci upravovat pouze svůj účet.

Poznámka:

Pokud stroj, na kterém poběží FreeNetIS, nebude centrální router/DNS, pak implementace bodů 4. a) a 4. d) bude realizována dálkovým přístupem na router/DNS přes SSH nebo jiný protokol.

8.3 Architektura Systému

8.3.1 „Deployment“ diagram



Pozn.: databáze uživatelských účtů **FreeNetIS** a **EkonomOS** by se měli vzájemně doplňovat, tj. neměla by v nich být žádná redundace – to by totiž komplikovalo aktualizaci (při změně redundantního údaje je potřeba jej změnit na obou místech – nutná vzájemná synchronizace) Projekt EkonomOS je také open source, jeho stránky : *www.sourceforge.net/projects/ekonomos* .

8.3.2 MVC (Model – View – Controler)

Je softwarová architektura, která rozděluje datový model aplikace, uživatelské rozhraní a řídicí logiku do tří nezávislých komponent tak, že modifikace některé z nich má minimální vliv na ostatní. Obecně řečeno, vytváření aplikací s využitím architektury MVC vyžaduje vytvoření tří komponent, mezi které patří:

Model (model), což je doménově specifická reprezentace informací, s nimiž aplikace pracuje.

View (pohled), který převádí data reprezentovaná modelem do podoby vhodné k interaktivní prezentaci uživateli.

Controller (řadič), který reaguje na události (typicky pocházející od uživatele) a zajišťuje změny v modelu nebo v pohledu.

8.3.3 Qcodo

Jeden z řady Framework založený na principu MVC. Vytvoří nám z naší databáze kostru této architektury. Jedná se o open source distribuci. Nevýhodou je, že je plně kompatibilní pouze s jazykem PHP5.

8.4 Struktura databáze

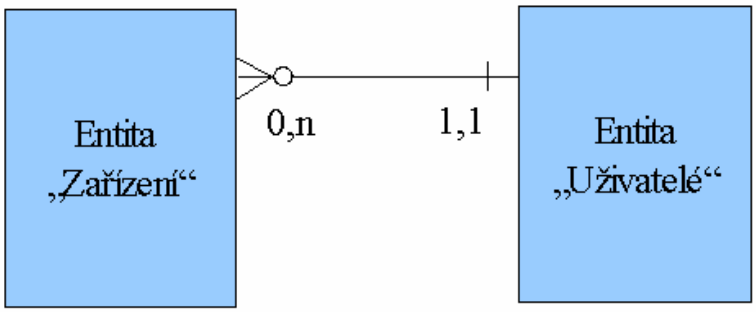
8.4.1 ER diagram

Na úvod je potřeba vysvětlit terminologii, používanou při návrhu databází:

Entita je třída objektů reálného světa. Příklady entit v naší databázi: "Členové" (Members), "Uživatelé" (Users), "Zařízení" (Devices) atd. Entita je v databázi implementovaná jednou tabulkou. Každá entita má nějaké vlastnosti – ***atributy***, např. Uživatel má nějaké jméno, příjmení, kontaktní údaje, ... - tyto vlastnosti v databázi odpovídají sloupcům tabulky entity.

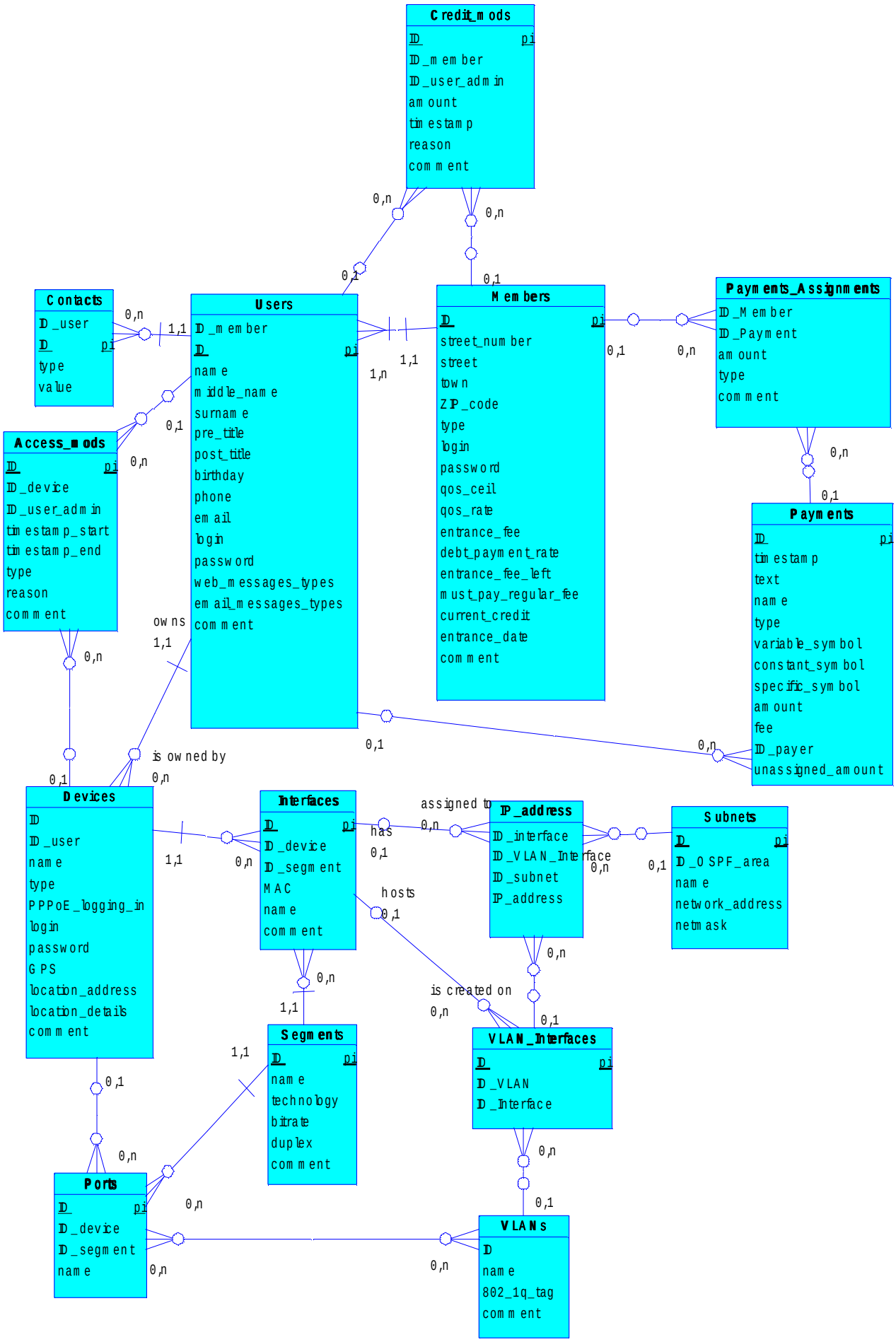
ER diagram (Entity Relationship) znázorňuje vztahy mezi jednotlivými entitami.

Příklad jednoduchého ER diagramu s vysvětlením symbolů:



- **Čára**, spojující obě tabulky, znamená, že entity, reprezentované tabulkami, mají spolu nějaký vztah – např. “Uživatel vlastní Zařízení” (nebo “Zařízení patří Uživateli”).
- “**Vidlička**” u tabulky Zařízení znázorňuje rozvětvený vztah – říká, že jeden uživatel může mít více různých zařízení (jedno zařízení je reprezentováno jedním záznamem=řádkem v tabulce “Zařízení”).
- “**Kolečko**” znamená nepovinnost, nebo nulu, tj. uživatel může mít více zařízení, ale **nemusí** mít žádné (může jich mít 0).
- Údaj “**0,n**” je pouze shrnutím výše uvedených bodů – říká, že Uživatel může mít 0 až *n* zařízení
- “Čárka” na straně tabulky “Uživatelé” znamená povinnost, tj. že zařízení **musí** patřit alespoň jednomu uživateli.
- Údaj “**1,1**” je opět jenom shrnutím předchozího bodu a toho, že na straně tabulky Uživatelé není “vidlička”. Říká, že zařízení patří právě jednomu uživateli (minimálně jednomu, maximálně jednomu)
- uvedený obrázek je typickým příkladem vztahu 1:N, který je v databázích nejčastější. Implementace tohoto vztahu v tabulkách je jednoduchá – každému uživateli i každému zařízení přidělíme jednoznačné identifikační číslo (ID), vazbu 1:N pak vyrobíme přidáním sloupce ID_uzivatele do tabulky zařízení.
- Tento příklad je typický i v dalším směru – vztah Uživatel – Zařízení totiž můžeme klidně definovat i jako M:N, protože jeden uživatel může mít více zařízení, ale zároveň každé zařízení může být vlastněno více uživateli. Vztah M:N je ale mnohem složitější na implementaci - v databázi se realizuje pomocí pomocné tabulky, která má 2 sloupce – ID_uzivatele a ID_zařízení, další práce je pak s uživatelským rozhraním databáze, které musí umožňovat ke každému zařízení přidělit více než jedoho uživatele. Samozřejmě, není to až takový problém, ale je to práce navíc, takže v návrhu databáze jsem se vztahům M:N snažil co nejvíce vyhnout.
- Dalším typickým kandidátem na vztah M:N je mezi tabulkami Member a User – každý uživatel může být přiřazen více členům – v naší síti je např. dost časté, že se nejdřív připojí syn, který pak do sdružení přivede i rodiče, přičemž jim koupí i spravuje veškeré zařízení. Pokud bude vztah Member-User pouze 1:N, budeme muset syna zadávat při tvorbě záznamu členství rodičů jako nového uživatele. Pokud bude vztah M:N, bude muset programátor uživatelského rozhraní k databázi implementovat přidělování stávajících uživatelů jednotlivým členům.

Následující obrázek je pouze kopií diagramu z OpenSystemArchitect.


Credit_mods
ID pi
ID_member
ID_user_admin
amount
timestamp
reason
comment
Contacts
ID_user
ID pi
type
value
Users
ID_member
ID pi
name
middle_name
surname
pre_title
post_title
birthday
phone
email
login
password
web_messages_types
email_messages_types
comment
Members
ID pi
street_number
street
town
ZIP_code
type
login
password
qos_ceil
qos_rate
entrance_fee
debt_payment_rate
entrance_fee_left
must_pay_regular_fee
current_credit
entrance_date
comment
Payments_Assignments
ID_Member
ID_Payment
amount
type
comment
Access_mods
ID pi
ID_device
ID_user_admin
timestamp_start
timestamp_end
type
reason
comment
Payments
ID pi
timestamp
text
name
type
variable_symbol
constant_symbol
specific_symbol
amount
fee
ID_payer
unassigned_amount
owns
is owned by
Devices
ID
ID_user
name
type
PPPoE_logging_in
login
password
GPS
location_address
location_details
comment
Interfaces
ID pi
ID_device
ID_segment
MAC
name
comment
has
assigned to
IP_address
ID_interface
ID_VLAN_Interface
ID_subnet
IP_address
Subnets
ID pi
ID_OSPF_area
name
network_address
netmask
hosts
is created on
Segments
ID pi
name
technology
bitrate
duplex
comment
VLAN_Interfaces
ID pi
ID_VLAN
ID_Interface
Ports
ID pi
ID_device
ID_segment
name
VLANs
ID
name
802_1q_tag
comment

8.4.2 Tabulky v databázi

Kompletní specifikace v angličtině se nachází na příslušném CD přiloženém k této práci jako samostatná příloha.

Tabulka Members (členové)

Obsahuje všechny důležité informace o členech občanského sdružení a doplňuje údaje uvedené v tabulce „users“ . Každý člen sdružení je zároveň tedy uživatelem tohoto IS.

Tabulka Users (uživatelé)

Obsahuje všechny důležité informace o jednotlivých uživatelích systému (jméno, příjmení, telefon, datum narození, atd.), je zde pamatováno na to že každý uživatel nemusí být zároveň členem sdružení, zejména pokud uživatel pouze pracuje s daný systémem – tzn. účetní, hospodář, ekonom atd.

Tabulka Credit_mods (kredity)

Každý člen nebo uživatel za svou činnost pro sdružení dostává tzv. kredity. Po získání jistého počtu kreditů za tyto kredity dostane peněžní odměnu.

Tabulka Contacts

Tabulka obsahuje doplňující kontakty jednotlivých členů.

Tabulka Payments_Assignment

Obsahuje informace o platebních transakcích

Tabulka Payments (platby)

Obsahuje informace o tom jak který člen má platit tedy jeho přidělené platební údaje.

Tabulka Device (zařízení)

Obsahuje všechny zařízení na síti : klientská a systémová. Dále nám říká jak jsou klientská zařízení nastavena k připojení k síti, dále pak obsahuje umístění těchto zařízení a způsob připojení.

Tabulka Interfaces (rozhranní)

Zde jsou uvedeny doplňující informace k tabulce DEVICES .. jsou zde všechny zařízení které mají MAC adresu.

Tabulka ports

Zde jsou uvedeny doplňující informace k tabulce DEVICES, jsou zde všechny zařízení které nemají vlastní MAC adresu pouze port (switch, Atp.)

Tabulka Segments

Obsahuje název ,technologii a přenosovou rychlost použitého média (segmentu). Obsahuje libovolný počet portů a libovolný počet MAC adres jednotlivých zařízení

Tabulka VLANs

Obsahuje údaje o jednotlivých VLAN na jednotlivých portech. 1 Vlan může obsahovat n portů.

Tabulka Interface

Nám říká jaký VLAN je přiřazen jakému Interface.

Tabulka IP_address

Obsahuje informace jakému interface, VLAN a subnetu je daná IP adresa přidělena.

Tabulka Subnets

Popis subnetu – maska a ID VLAN

8.5 Přístupová práva k jednotlivým částem systému

V nejjednodušší variantě přístupová práva musí podporovat alespoň tyto uživatelské role:

- nezaregistrovaný uživatel – nedostane se v systému nikam, pouze na stránku "registrace"
- zaregistrovaný uživatel – dostane se v systému pouze na svůj účet
- zaregistrovaný uživatel, který se stal členem sdružení
- člen-technik (má právo tvořit a měnit záznamy o technologiích ve vlastnictví sdružení)
- správní rada – má právo měnit úplně všechno
- účetní – má právo měnit účetní záznamy

Implementace přístupových práv nebude úplně triviální, protože to není jenom o tom, že "obsah tabulky X může vidět skupina Y". Přístupová práva musí mít granularitu na úrovni jednotlivých záznamů a sloupců v tabulkách, aby bylo možno zajistit např. to, že přihlášený nečlen nebude mít ani read-only přístup k informacím o technických detailech sítě – může se ale např. podívat na polohu přístupového bodu na mapě a nechat si vyhledat členy, kteří jsou připojeni ve vzdálenosti X metrů od jím zadaného bodu.
Přístupová práva se dají do systému naprogramovat "napevno", tj. přístup k jednotlivým objektům systému pro jednotlivé role bude ošetřen natvrdo přímo v kódu, ale toto řešení je velmi nevhodné – pokud bychom pak v budoucnosti chtěli přidat jakoukoli další roli, nebo třeba jenom omezit právo přístupu určitého člověka k určitému objektu v databázi, znamenalo by to modifikovat téměř všechny zdrojové kódy systému.

Jak tedy na to?

Jedním z myšlenkově elegantních řešení je použití knihovny PhpGACL – tato knihovna umožňuje definovat přístupová práva k jednotlivým objektům pomocí N-árně stromových struktur:

- "Access Requesting Object" (ARO) - každý uzel tohoto stromu ARO reprezentuje uživatelskou roli, kterou lze snadno modifikovat přidáním až N synů (jde tedy o N-ární

strom), kteří zdědí všechny původní vlastnosti rodičovské role a navíc přidají některé nové přístupové právo.

- “Access Control Object” (ACO) – říká, zda přístup nějaké role k nějakému objektu je povolen nebo zakázán.
- “Access eXtension Object” (AXO) – každý uzel tohoto stromu reprezentuje objekt, ke kterému chceme kontrolovat právo přístupu jednotlivými uživatelskými rolemi (ARO). Opět, aby nebylo potřeba definovat každý objekt zvlášť, stačí definovat rodičovský uzel, a z něj odvodit potomky, u nichž bude přístup od jednotlivých rolí stejný jako u rodičů, ale nemusí být.

Při programování jednotlivých modulů FreeNetISu je potřeba mít neustále na paměti, aby se ke každé funkci systému dostali pouze uživatelé, kteří na ni mají práva. Tj. je potřeba neustále volat funkci `gacl->acl_check` pro zjištění, jestli právě přihlášený uživatel má určitý objekt systému vidět, zda jej může editovat atd.

Protože phpGacl pro každé volání acl_check přistupuje do databáze, přináší tento přístup velké zpomalení běhu programu – naštěstí umí phpGacl vytvářet cache těchto přístupů do databáze, takže opakované stejné dotazy jsou pak velmi rychlé.

8.6 Implementace Systému

Systém byl vyvíjen na lokálním PC, tzv. localhostu. Pro tyto účely byla použita volně dostupná aplikace AppServ Open Project ve verzi 2.5.6, která obsahuje webový server Apache, skriptovací jazyk PHP, MySQL databázi a také pohodlné rozhraní pro práci s databází phpMyAdmin.
Samotné programování probíhalo v prostředí produktu PSPad 4.3.0, případně Poznámkovém bloku obsaženém v MS Windows.

Funkční systém můžete najít na adrese http://freenetis.unart.cz

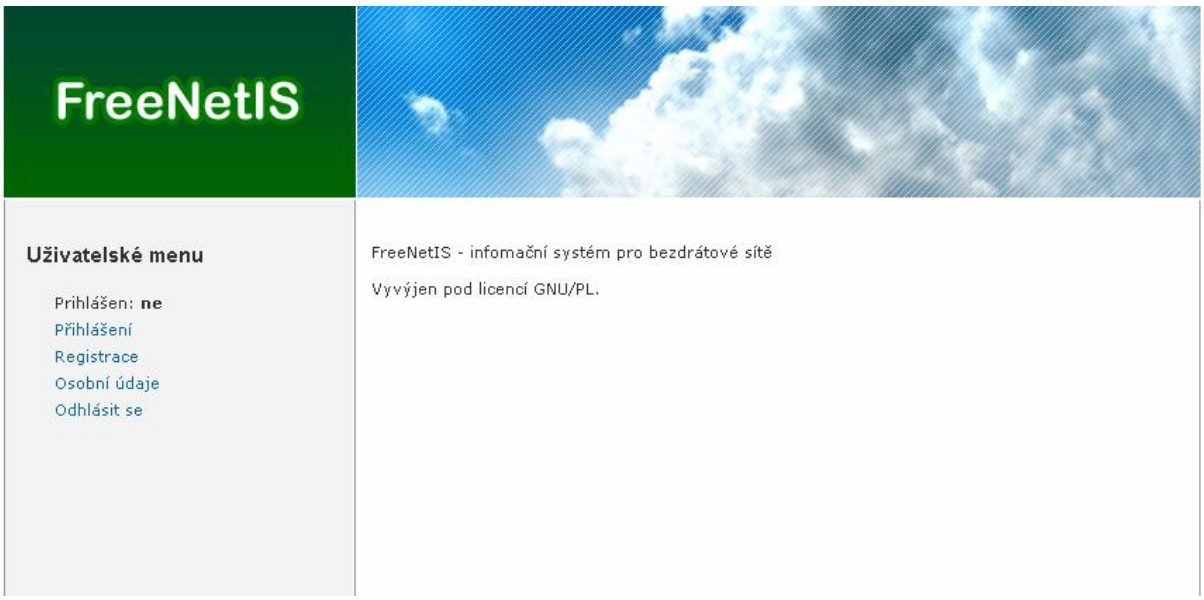
8.6.1 Instalace Qcodo Framework

Framework pro tvorbu tříd z tabulek databáze a také pro tvorbu jednotlivých formulářů databáze.
Postup instalace:
Je třeba tento balík komplet stáhnou z adresy : *http://www.qcodo.com/downloads/* .
Rozbalíme jej do www adresáře
Nastavíme konfigurační soubory (cesty k jednotlivým souborům a přístup do databáze MySQL). Dle příslušného videa: *http://www.qcodo.com/demos/* .

Implicitně je tento Framework spojen s jazykem PHP5. Pokud chceme použít s tímto nástrojem jazyk PHP4 je třeba doinstalovat balík ADODB z:
http://amountaintop.com/adodb_adapter_for_qcodo_beta_2 .
Po úspěšné instalaci a propojení s naší databází se nám objeví startovací obrazovka Frameworku. Pokud klikneme na „code generace your tables“ vytvoříme třídy a formuláře pro jednotlivé tabulky databáze se kterými dále pracujeme.

8.6.2 Veřejná část webové aplikace

Tato část se zobrazí každému uživateli, který vstoupí na stránky. Na obr.16 vidíme základní vzhled a rozmístění prvků.



Obrázek 16

Hlavička stránky je tvořena nadpisem. V levé části je hlavní menu, které obsahuje ve veřejné části pouze odkazy na úvodní stránku, na přihlášení nebo na registraci nového uživatele, či na odhlášení. Hlavní obsah zabírá většinu stránky a zobrazuje se v pravé části. V patičce stránky je uveden odkaz na zdrojové kódy projektu na sourceforge.org , autor, vyhrazená práva.

8.6.3 Uživatelská část webové aplikace

Po úspěšném přihlášení je možnost zobrazit a editovat osobní údaje.

8.6.4 Administrátorská část webové aplikace

Po příhlášení administrátora má administrátor možnost editovat jakokoliv tabulku v databázi.

8.7 Funkce a algoritmy

Do systému byly implementovány vestavěné funkce jazyka PHP, kterých jsem využíval především při programování vlastních funkcí a algoritmů. Většina těchto funkcí je využívána všemi uživateli, ovšem některé jsou vytvořeny jen pro určitý typ uživatele. Také byli využity funkde Frameworku QCodo

8.7.1 Obecné funkce

Jedná se o funkce využívané všemi registrovanými uživateli systému. Slouží ke zvýšení bezpečnosti systému, kontrole práv uživatele, práci se soubory apod. Nejvýznamnější funkce jsou popsány podrobněji, možnosti ostatních funkcí jsou zmíněny jen v bodech.

Rozlišení přístupu různých typů uživatelů

K rozlišení přístupů jsme použili knihovnu phpGACL

Přihlášení

Pokud uživatel zadá login, heslo a stiskne tlačítko přihlásit, provede se:

- Ošetření vkládaných dat
- Heslo se upraví pomocí hašovací funkce
- Provede se SQL dotaz, zda v databázi existuje user - uživatel se zadaným loginem a souvisejícím heslem

Pokud takový uživatel existuje, uložíme si do relace session jeho id a login. Také se mu automaticky přiřadí práva na jaké stránky systému má přístup.

Odhlášení

Odhlášení uživatele spočívá ve vymazání proměnných uživatelské relace a jejím ukončení:

```
$_SESSION = array(); // Vymaže proměnné
session_destroy();   // Ukončí relaci
```

8.7.2 Formuláře

V systému jsme využili vygenerovaných formulářů z Frameworku Qcodo. Tyto pak jsme upravili pro danou šablonu stránky. Viz příloha A.

8.8 Srovnání s dostupnými systémy

Podobné systémy jako je Freenetis můžeme uvést asi dva. Jsou to WifiDog a CaféRadius.

WifiDog je systém, který komplexně zastřešuje správu připojených PC. Využívá k tomu svého autentizačního serveru. Je založen na 3 základních součástech (***OS Linux***, balíček ***netfilter*** zkompilovaný do kernelu, balíček **iptables**) a doplňkového balíčku **iproute2**. Je nejvíce využívaný pro sítě typu „HotSpot“ stejně tak jako druhý sytém CaféRadius.

CaféRadius je také využíván v „Hot Spot“ sítích. Pro svoji činnost potřebuje tento software: webový server s SSL, MySQL, PHP, Radius server spolupracující s MySQL a program ChilliSpot. Funkce jsou podobné jako u WifiDogu. Tzn. kompletní správa uživatelů – od prvotní registrace – přihlášení – zpřístupnění do internetu – odhlášení. Pokud jsi pohledneme své statistiky můžeme vidět kolik jsme přenesli dat, atp.

FreeNetIS databáze má podobné vlastnosti jako oba předchozí systémy, navíc ale umožňuje zachytit kompletní strukturu sítě, což je u větších sítí (jako třeba u sítě www.slfree.net) dost podstatné plus. V současné době je již naše síť tak rozsáhlá, že ani hlavní administrátor si nepamatuje, jaké IP adresy má na kterých přístupových bodech jaké Alany jdou kam, atd. S pomocí této databáze bude ale možné mít tyto informace zachycené a navíc udržované zcela automaticky (trvalým monitoringem všech přístupových bodů). Dále pak zde je možnost propojení s databází plateb – projekt EkonomOS.

8.9 Možná rozšíření systému

- Zasílání zpráv uživatelům přes webové rozhranní
- Rozšíření přístupových práv systému – dle návrhu
- Možnost přihlašování do sítě pomocí webového rozhranní
- Správa plateb – automatické omezení neplatičů
- EkonomOS – propojení
- Monitoring všech zařízení

9 Závěr

Tato diplomová práce byla pro mě velmi naučná. A to jak z pohledu standardu 802.1x, jakož je pochopení jeho principů, také z hlediska praxe. Neboť Instalování RADIUS serveru je sice triviální, ale sem tam jsem narazil i stinné stránky, jako je špatně stažený soubor .. či nedokonale zabalený a umístěný soubor na www stránkách – radius server. Instalace samotných certifikátu pro autentizaci 802.1x již tak jednoduchá není, spíše je netriviální.

Můžu říci, že daná práce je pro mě výzvou, neboť problematice bezdrátových sítí a jejich bezpečnosti se věnuji a chtěl bych se i nadále věnovat ve svém budoucím zaměstnání. Pomocí této

práce jsem se také seznámil i s jinými autentizačními nástroji, ne jen s RADIUS Serverem. Tyto nástroje dovedou řešit některé situace více efektivněji než samotný systém RADIUS. Dnes již je znám bezpečnostní standard 802.11i, který pomalu, ale jistě vchází do podvědomí středních a menší ISP – poskytovatelů internetu. A dnešní trend je nejen poskytovat internet bezdrátově, ale poskytovat zabezpečené spojení pro fyzické i právnické osoby, neboť věk elektronické komunikace se v posledních letech několika násobil. A udělat bankovní převod přes internet a jiné činnosti jsou ne ojedinělé, ale běžnou praxí.

Také bylo mým úkolem navrhnout systém na správu a účtování přenesených dat. Tento systém využívá funkce RADIUS serveru – „accounting“, kde sám server nám ukládá do databáze kolik dat daný ověřený uživatel přenesl. Tyto údaje je pak již jednoduché pomocí vztahů v databázi plně využívat. Daný systém – FreeNetIS není navržen pouze na tento specifický účel, ale jako komplexní nástroj pro správu celé sítě. Pod komplexní správou si představuji kompletní vedení databáze uživatelů, všech připojený aktivních i neaktivních síťových prvků, správu plateb či možnost zasílat uživatelům krátké zprávy přes webové rozhranní. Bohužel je systém tak rozsáhlý, že se mi jej podařilo naimplementovat z malé části. A samotná implementace by mohla býti samostatnou diplomovou prací.

Díky této práci jsem se seznámil nejen s autentizačními nástroji a jejich konfiguraci, ale i s tvorbou přístupných a použitelných webových aplikací pro správu uživatelů. Tento druh aplikací se stává čím dál rozšířenější zvláště u ISP. Zároveň pro mě byla přínosem komunikace se spoluvývojáři zde navrhnutého informačního systému.

Literatura

[1] Barken Lee Jak zabezpečit bezdrátovou síť Wi-Fi. Brno, Computer press 2004

[2] Bigelow S. J. Mistrovství v počítačových sítích. Brno, Computer press 2004

[4] Gutmans, Andi; Bakken, Stig; Rethans, Derick: Mistrovství v PHP5. Brno. CP Books, a.s., 2005, 655 stran. ISBN 80-251-0799-X

[3] internet: http://www.freeradius.org, stažení RADIUS serveru (leden 2007)

[4] internet: http://ftp.cvut.cz/debian/pool/main/r/radiusclient/, RADIUS klient (leden 2007)

[5] internet: ftp://ftp.freeradius.org/pub/radius/freeradius-1.0.5.tar.gz, freeradius (leden 2007)

[6] internet http://fedora.redhat.com/, Ret Hat linux – fedora core 4 (leden 2007)

[7] internet http://www.ieee802.org/1/pages/802.1X-rev.html, principy 802.1x (březen 2007)

[8] internet http://www.sourceforge.net/projects/phpmyprepaid, php skripty (březen 2007)

[9] internet http://www.caferadius.com/, caferadius (duben 2007)

[10] internet http://www.wifidog.org/, wifidog (květen 2007)

[11] internet http://tldp.org/HOWTO/html_single/8021X-HOWTO, nastaveni 802.1x (duben 2007)

[12] internet http://www.root.cz/clanky/poznamky-k-ldap/, intalace LDAP (březen 2007)

[13] internet http://www.debian.org, operační systém debian (květen 2007)

[14] internet http://interval.cz/clanky/instalace-phpmyadmin/, instalace phpmyadmin (březen 2007)

[15] internet http://www.open1x.org, instalace openssl na debbian (březen 2007)

[16] internet http://www.eduroam.cz/cz/index.html, implementace RADIUS (květen 2007)

[17] internet http://www.qcodo.com/, Framework QCODO (květen 2007)

Seznam příloh

Příloha A – Uživatelská příručka

Instalace systému

Všechny zdrojové soubory webové aplikace jsou umístěny ve složce „freenetis“ včetně potřebných adresářů. Systém je možné nainstalovat na webový server podporující PHP5 a MySQL 5.0 a vyšší.

1) Zkopírujte kompletní obsah složky „www“ na webový server.
2) Zkontrolujte, zda máte u adresáře „data“ povoleno čtení, zápis i spouštění pro všechny uživatele. Případně změňte nastavení těchto atributů, např. příkazem `chmod 777 data`.
3) K provedené dalších kroků je nutné, abyste měli vytvořenou databázi, ke které je nutné znát uživatelské jméno, heslo, název hostitele a název databáze.
4) Tedy zvolíme jméno databáze a pomocí sql skriptu install.sql provedeme vytvoření příslušných tabulek a jejich vazeb.
5) Celý adresář freenetis překopírujeme do „wwwroot“ – kořenový adresář serveru pro http dokumenty.
6) Editujte soubor „wwwroot“\freenetis_devtools_cli\path_to_pretend, zde napište cestu k podadresáři \includes
7) Editujte soubor „wwwroot“\frenetis\includes\configuration.inc.php v části:

```
define ('__DOCROOT__', 'c:/xampp/htdocs// - zde zadejte umístění „wwwroo“');
define ('__VIRTUAL_DIRECTORY__', ' ');
define ('__SUBDIRECTORY__', '/freenetis - podadresář');
```

a v části:

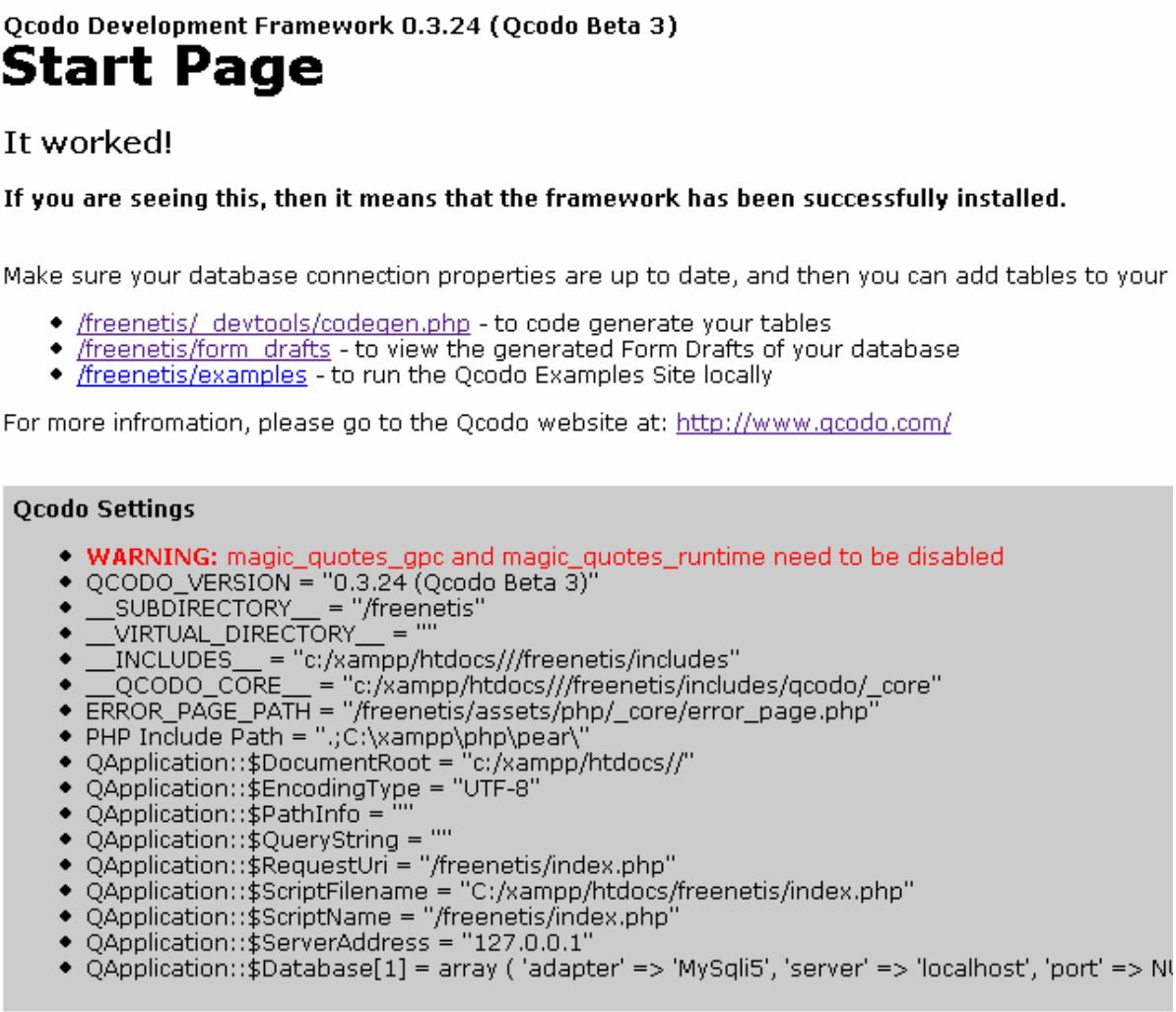
```
define('DB_CONNECTION_1', serialize(array(
                                    'adapter' => 'MySqli5',
                                    'server' => 'jméno serveru, implicitně localhost',
                                    'port' => null,
                                    'database' => 'freenetis – jmeno databaze',
                                    'username' => 'root',
                                    'password' => '',
                                    'profiling' => false)));
```

Kompletní nastavení balíku QCODO v tutoriál videu na adrese http://www.qcodo.com/demos/ .

8) Pro ověření funkčnosti zadejte do prohlížeče adresu webu, kde jste umístili tuto webovou aplikaci a připište do adresy cestu ke skriptu

   např. `localhost/freenetis/index2.php`

V případně chybného nastavení se vypíše chybové hlášení, jinak se zobrazí tato stránka stránka.



Obrázek 17

Která říká, že Framework QCODO funguje.

9) Poté zadejte do prohlížeče adresu webu, kde jste umístili tuto webovou aplikaci a připište do adresy cestu ke skriptu „start.php“.

   např. `localhost/freenetis/`

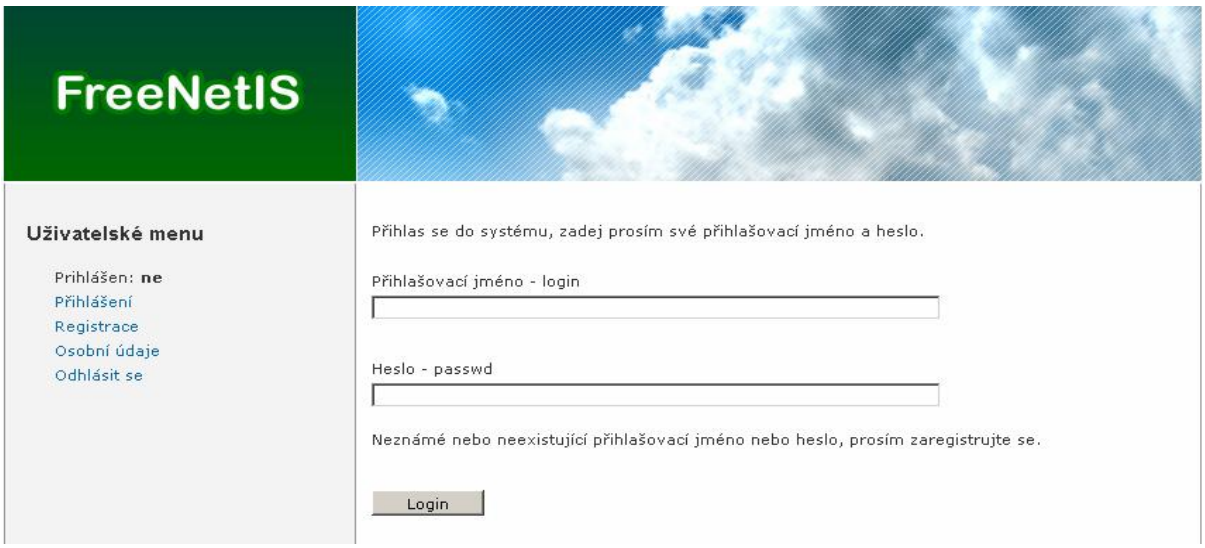
   Tím dojde ke spuštění aplikace freenetis – zobrazí se úvodní stránka

10) Tímto je instalace webové aplikace na webový server hotova.
11) Nyní můžete začít používat nainstalovanou webovou aplikaci.

Daný systém se nachází na adrese: http://freenetis/unart.cz

Práce se systémem

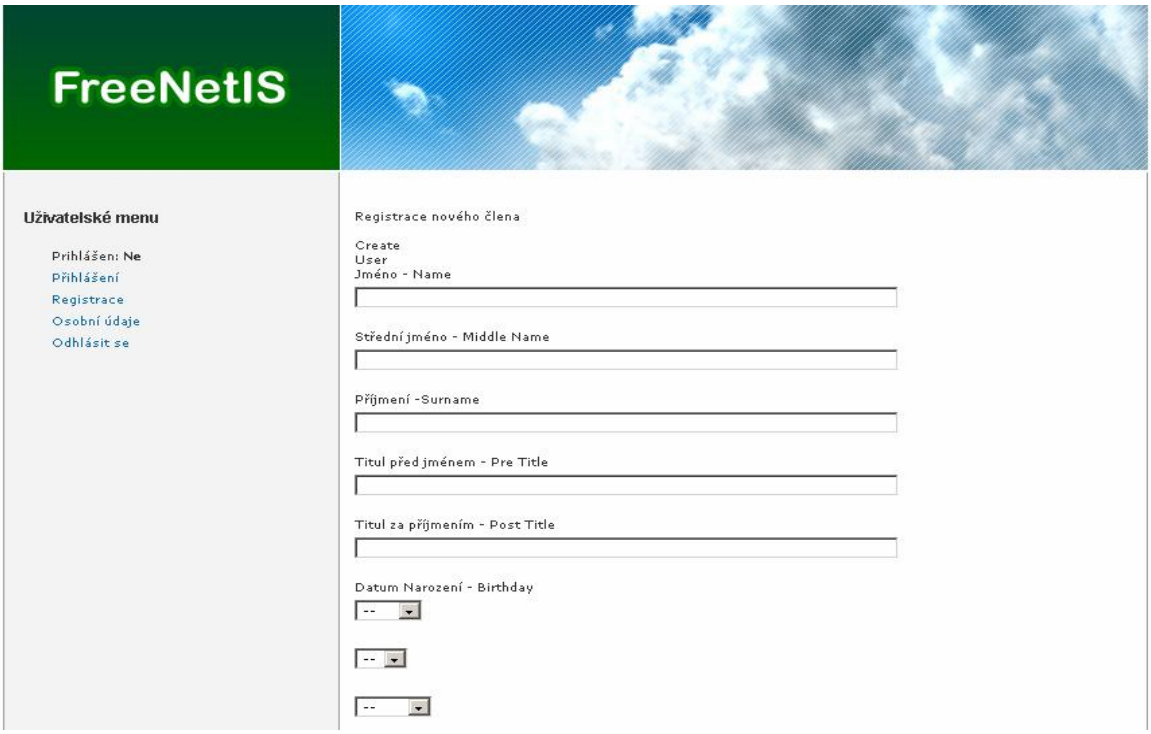
Přihlášení do systému



Obrázek 18

Pokud zadáte špatné nebo neexistující přihlašovací jméno nebo heslo, je třeba se zaregistrovat

Registrace



Obrázek 19

Další možnosti práce ze systémem viz stránky http://freenetis.unart.cz .