UNIVERZITA PARDUBICE

FAKULTA ELEKTROTECHNIKY A INFORMATIKY

BAKALÁŘSKÁ PRÁCE

2008 Lukáš Cír

Univerzita Pardubice

Fakulta elektrotechniky a informatiky

Wi-Fi Router
Ověřování protokolem RADIUS
a
RADIUS Accounting

Lukáš Cír

Bakalářská práce

2008

Poděkování

Chtěl bych poděkovat Mgr. Tomáši Hudcovi, za cenné rady, informace, připomínky a vedení celé mé práce.

SOUHRN

Účelem práce bylo vypracovat popis dvou protokolů Internetu. Jedná se o protokol RADIUS a RADIUS Accounting. V druhé části navrhuji postup konfigurace softwarového serveru RADIUS a klienta RADIUS od firmy MikroTik pro společnou práci v sítích Wi-Fi.

První z protokolů slouží pro ověřování klientů sítě při přístupu k routeru Wi-Fi (nebo jinému přístupovému serveru). Druhý pak k evidenci informací o klientovi (jak dlouho byl připojen k bodu, nebo kolik dat za tuto dobu přenesl). Práce též účelně popisuje požadavek Change Of Authorization, jeho strukturu, funkci, ale i důvody použití.

KLÍČOVÁ SLOVA

RADIUS, RADIUS Accounting, Change Of Authorization, FreeRADIUS, MySQL, MikroTik, RouterOS, autentizace, autorizace, Wi-Fi, Wireless, HotSpot.

TITLE

Wi-Fi Router – RADIUS protocol authentication and RADIUS Accounting

ABSTRACT

In my work, I focused an describing of two Intenet protocols. The two protocols are RADIUS and RADIUS Accounting. In the second part of my work I project the scheme configuration of software server and client RADIUS from MikroTik company for collective work in Wi-Fi networks.

The first protocol serves for authorization network clients accessing to router Wi-Fi (or another access server). The other one, for the evidence of information about network client (how long was the client online or how many dates did he transport). My work also describes the request Change Of Authorization, its structure, function and also some reasons for using it.

KEYWORDS

RADIUS, RADIUS Accounting, Change Of Authorization, FreeRADIUS, MySQL, MikroTik, RouterOS, authentication, authorization, Wi-Fi, Wireless, HotSpot.

Obsah

Úvod

Cílem této bakalářské práce je seznámení s protokolem RADIUS a RADIUS Accounting a návrh použití protokolů RADIUS v sítích Wi-Fi.

Popis těchto protokolů, jejich využití a formáty paketů jsou hlavní náplní teoretické části této bakalářské práce. Součástí je i informace o paketu Change Of Authorization, jeho formátu, funkci, ale i využití.

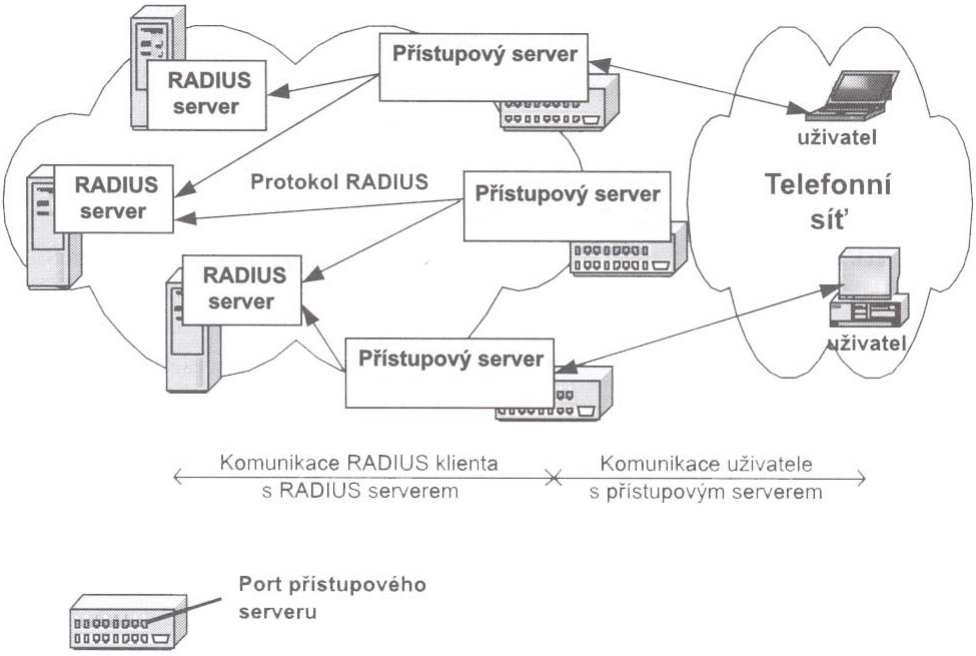
V praktické části navrhuji praktické užití protokolu RADIUS a RADIUS Accounting v sítích Wi-Fi. Pro tento účel používám server FreeRADIUS a databázový systém MySQL. Klientskou část tvoří router od firmy MikroTik. V poslední kapitole praktického užití jsou veškerá nastavení otestována.

1. Popis Protokolů

1.1 Protokol RADIUS

Protokol RADIUS je popsán dokumentem RFC-2865 [3]. Jeho cílem je provádět centralizovanou autentizaci uživatelů připojujících se na přístupový server NAS (Network Access Server).

Přístupový server je box obsahující směrovač s baterií asynchronních portů, na které se uživatelé na příklad připojují pomocí telefonického připojení [1]. Pokud se klient připojí k přístupovému serveru, pak přístupový server prověřuje oprávnění uživatele pro přístup do sítě. Tato oprávnění mohou být lokálně nastavena v přístupovém serveru, při použití více přístupových serverů je to však velice nepraktické. Viz obr. 1.



Obr. 1: Schéma použití protokolu RADIUS. [1]

V dnešní době lze za přístupový server NAS pokládat různá zařízení podporující funkci klient RADIUS. Můžou jimi být například routery či switche atd. Název, přístupový server je používán více méně jen z historických důvodů. Ovšem funkce protokolu RADIUS zůstává stejná.

Jak už vyplývá z úvodu kapitoly, je protokol RADIUS navržen pro centralizované ověřování uživatelů přistupujících do sítě. Samotný protokol ovšem uživatele neověřuje, pouze se stará o komunikaci mezi serverem RADIUS a přístupovým serverem NAS (klientem RADIUS). Ověření uživatele obstarává server RADIUS, který obsahuje několik autentizačních mechanismů.

Přístupový server NAS coby klient protokolu RADIUS se dotazuje serveru RADIUS, zda může uživatele do sítě propustit a za jakých podmínek. Server RADIUS odpoví buď kladně, nebo záporně. Pokud kladně, přibalí do odpovědi podmínky připojení uživatele. Jakmile přístupový server odpověď příjme, připustí uživatele do sítě a aplikuje na jeho připojení specifické podmínky.

1.1.2 Funkce protokolu

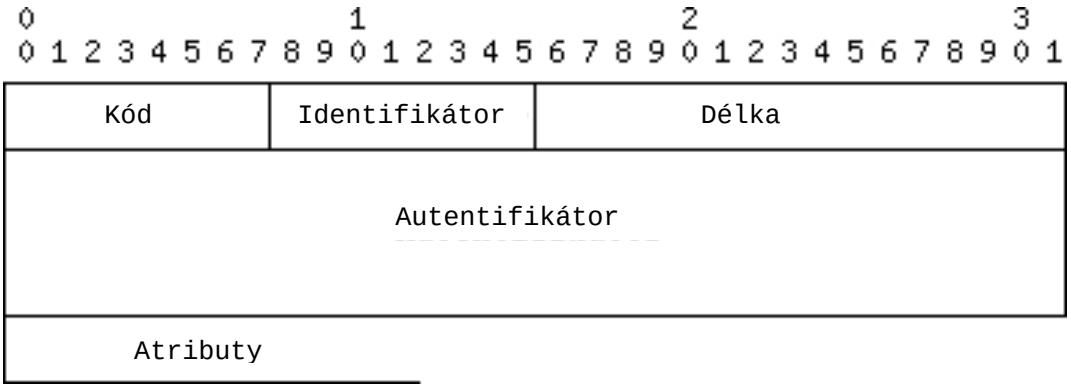
Protokol RADIUS je protokolem vytvářejícím komunikační tok mezi přístupovým serverem a serverem RADIUS. Server RADIUS ověřuje uživatele a umožňuje jim předat přístupové informace.

Protokol RADIUS je aplikační protokol využívající User Datagram Protocol (UDP). Jedná se o datagramovou službu, podobně jako Domain Name System (DNS). Důvodem volby protokolu UDP je nutnost rychlé odezvy přístupového serveru NAS v momentě požadavku o přihlášení uživatele.

Další výhodou používání UDP je fakt, že lze používat, stejně jako u DNS, více serverů RADIUS. V případě, když přístupový server NAS neobdrží odpověď od prvního serveru RADIUS, zopakuje dotaz na další server RADIUS. Server protokolu RADIUS obvykle naslouchá na portu UDP 1812.

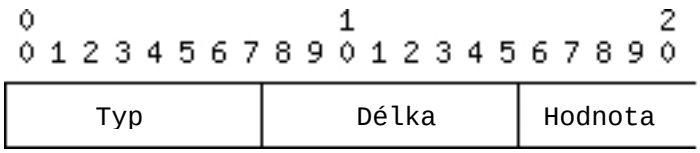
Protokol RADIUS používá čtyři typy zpráv. Každý druh zprávy má svůj odlišný číselný kód, tento kód se nachází v závorce (viz kód v obr. 2):

- Access-Request (kód = 1) – touto zprávou posílá přístupový server NAS požadavek serveru RADIUS na ověření uživatele.
- Access-Accept (kód = 2) – touto zprávou sděluje server RADIUS přístupovému serveru, že uživateli má být povolen přístup do sítě.
- Access-Reject (kód = 3) – tato odpověď je opakem předchozí a server RADIUS sděluje informaci o zamítnutí přístupu uživatele do sítě.
- Access-Challenge (kód = 11) – jedná se o výzvu zaslanou serverem RADIUS přístupovému serveru, aby klient zadal heslo.



Obr. 2: Formát paketu protokolu RADIUS. [9]

Při pohledu na obr. 2 paket začíná kódem, který specifikuje typ zprávy. Následně identifikátor, sloužící k párování dotazů a odpovědí, délka paketu, autentifikátor, který je určen pro prokázání správnosti odpovědi serveru, a poslední jsou atributy.



Obr. 3: Formát atributů paketu. [9]

Jak je vidět z obr. 3 první typ určuje druh atributu. Value určuje druhou část páru Attribute-Value a to hodnotu atributu.

Protokol RADIUS je navržen pro práci se dvěma druhy autentizace:

- Hlavní náplní protokolu RADIUS je ověřování klienta přihlašujícího se k přístupovému serveru NAS. Protokol RADIUS ovšem přenáší pouze data autentizačních protokolů. V serveru RADIUS můžou být ovšem implementovány různé autentizační mechanismy. Přitom na protokolu RADIUS to nic nemění.
- Vzhledem k bezpečnosti síťové komunikace je nutné, aby byla zabezpečena komunikace mezi serverem RADIUS a přístupovým serverem NAS (klientem RADIUS). Toto je tím druhým typem autentizace. Tato komunikace je riziková, protože by mohlo dojít k podvržení serveru RADIUS a podvržený server RADIUS by mohl povolit vstup do sítě neoprávněným uživatelům. Z tohoto důvodu si musí být klient RADIUS jist, že se jedná o odpověď od nepodvrženého serveru RADIUS.

Zabezpečení komunikace serveru RADIUS a přístupového serveru je na principu sdíleného tajemství, který zná pouze server RADIUS a NAS. Sdílené tajemství je řetězec znaků, který je sdílen mezi klientem a serverem RADIUS za účelem bezpečné komunikace. Jelikož komunikace server – klient musí být zabezpečena hlavně proto, aby nebyl do sítě umístěn podstrčený server RADIUS, je komunikace zabezpečena pouze při zaslání odpovědi serverem RADIUS přístupovému serveru NAS.

Při pokládání dotazu vygeneruje přístupový server NAS do pole autentifikátor náhodný řetězec, který slouží pouze pro identifikaci odpovědi (zda se jedná o odpověď na dotaz). Server RADIUS pak do téhož pole vrátí kontrolní součet vypočítaný hashovacím algoritmem MD5 z celého vraceného paketu. Tento kontrolní seznam spojí s řetězcem sdíleného tajemství a odešle přístupovému serveru NAS. Přístupový server NAS po přijetí odpovědi serveru RADIUS porovná řetězec sdíleného tajemství přijatý v odpovědi serveru RADIUS s řetězcem sdíleného tajemství, který má přiřazen k danému serveru RADIUS. Pokud se oba řetězce shodují, odpověď serveru RADIUS je považována za legitimní, v opačném případě za falzifikát.

Server RADIUS je komplexní ověřovací systém s podporou velké škály autentizačních mechanizmů. Lze využít ověřování např. pomocí uživatelského jména a hesla. Nebo použití protokolů PAP (Password Authentication Protocol), CHAP (Challenge-Handshake Authentication Protocol), až po autentizaci jednorázovým heslem.

Je-li zvolené ověření uživatele heslem, tak jméno i heslo uživatele odesílá přístupový server NAS ve zprávě požadavku Access-Request. Server RADIUS může na tento požadavek, po ověření hesla, odpovědět buď akceptováním žádosti, nebo jejím zamítnutím. Při akceptaci požadavku odpoví server RADIUS zprávou Access-Accept. Pokud je požadavek zamítnut, zpráva serveru RADIUS bude Access-Reject.

Při tomto způsobu ověřování se uživatelské jméno přenáší atributem „User-Name“ a heslo atributem „User-Password“. Nebezpečí této metody ověření spočívá v tom, že uživatelské heslo se přenáší v čistém textu.

Je ovšem důležité si uvědomit, že při této jednoduché ověřovací technice existují dva komunikační kanály. Prvním z komunikačních kanálů je dialog mezi uživatelem a přístupovým serverem NAS. Cílem této komunikace je získat ucelené informace pro vygenerování zprávy požadavku Access-Request. V podstatě se může jednat o několik dotazů serveru NAS a odpovědí uživatele. Mezi takové dotazy patří například dotazy na uživatelské jméno, heslo a další.

Následně je navázán druhý dialog protokolu RADIUS. Tato komunikace probíhá mezi serverem RADIUS a klientem RADIUS. Až v této komunikaci dochází k výměně požadavku o ověření zprávy Access-Request. A odpovědi na požadavek Access-Accept nebo Access-Reject.

Pokud server RADIUS zjistí, že pro ověření jistého klienta je požadována autentizace jednorázovým heslem, to znamená ověření na principu výzva/odpověď (Challenge/Response), odpoví server RADIUS na příchod požadavku Access-Request zprávou typu Access-Challenge, obsahující výzvu o zadání jednorázového hesla. Zprávu přístupový server NAS zpracuje a odešle uživateli výzvu k zadání jednorázového hesla. NAS následně sestaví novou zprávu požadavku Access-Request s novým identifikátorem jednorázového hesla v atributu „User-Password“.

K vytvoření páru mezi výzvou Access-Challenge s nově vytvořenou zprávou Access-Request slouží atribut zprávy „State“. Server RADIUS do zprávy Access-Challenge vloží identifikaci atributu „State“. Přístupový server tento atribut zkopíruje do nově vytvářené zprávy Access-Request.

Pro protokol CHAP tento mechanismus, ale není třeba. Když uživatel kontaktuje přístupový server NAS a použije ověřování systémem CHAP, pozná přístupový server, že jde o ověření protokolem CHAP. Z tohoto důvodu nemusí přístupový server NAS pro zadání výzvy kontaktovat server RADIUS. Vygeneruje sám generátorem náhodných čísel výzvu pro protokol CHAP, kterou pošle uživateli. Uživatelský software vygeneruje jednorázové heslo a odešle přístupovému serveru. Po obdržení jednorázového hesla přístupový server vygeneruje zprávu požadavku Access-Request. Tento požadavek naplní kromě jiného atributy:

- User-Name – uživatelské jméno (CHAP Username).
- CHAP-Password – atribut je naplněn dvěma údaji: CHAP ID a jednorázovým heslem.
- CHAP-Challenge – naplní náhodnou výzvou generovanou přístupovým serverem.

Princip CHAP mechanismu [10]:

1. Jakmile se klient připojí k přístupovému serveru NAS, NAS vygeneruje výzvu protokolu CHAP a odešle klientovi.

2. Jako odpověď na výzvu, klientský software vygeneruje jednorázové heslo (často hashovací funkcí md5 z uživatelského hesla, identifikátoru a dalších hodnot, jejichž hodnota se s každou výzvou mění) a odešle zpět přístupovému serveru NAS.
3. Přístupový server NAS předá požadavek Access-Request serveru RADIUS. Zpráva obsahuje atribut CHAP-Password naplněn údaji: CHAP ID a jednorázové heslo klienta.
4. Server RADIUS sám vygeneruje předpokládané heslo a porovná ho s jednorázovým heslem atributu CHAP-Password.
5. Pokud se obě jednorázová hesla shodují, server RADIUS odešle zprávu Access-Accept přístupovému serveru NAS, v opačném případě odpoví zprávou Access-Reject.
6. Podle druhu odpovědi přístupový server NAS klienta příjme, či odmítne.
7. V náhodných intervalech přístupový server NAS posílá nové výzvy klientovi. Kroky 1 až 6 se opakují.

Vzhledem k vygenerování jednorázového hesla (např. funkcí md5) a následné distribuci v tomto formátu je tento postup bezpečnější než ověřování pomocí hesla v čistém textu. Dešifrování jednorázového hesla vygenerovaného jednosměrnou funkcí md5 je sice dnes možné (např. metodou brutální síly), ale velmi náročné (náročnost brutální síly spočívá v čase), z čehož vyplývá, že napadení takového jednorázového hesla je obtížné. Zvláště, když jednorázové heslo je platné pouze pro danou výzvu protokolu CHAP.

Mezi další zajímavé funkce serveru RADIUS patří i funkce jako „proxy“. Využívá se v případě, když server RADIUS není schopen si ověřit požadavek o autentizaci uživatele zaslaný od přístupového serveru NAS z důvodu, že informace o klientovi jsou uloženy v jiném serveru RADIUS. Pak požádá server RADIUS, co by klient, vzdálený server RADIUS o poskytnutí ověření. V tomto případě pracuje server RADIUS jako proxy.

1.1.3 Popis některých běžných atributů protokolu

Jak už jsem se zde v textu zmínil dříve, protokol RADIUS se skládá z párů Attribute-Value. V paketu je atribut zastoupen jeho typem. Typ atributu je číselná hodnota. V paketu se dále nachází délka atributu a nakonec hodnota atributu. Hodnota atributu může být vyjádřena následujícími způsoby:

- řetězcem bajtů,
- textovým řetězcem ve formátu UTF-8,
- IP-adresou,
- časovým razítkem.

V následujícím výpise je popsáno několik základních atributů. Jejich význam je vysvětlen. Nejsou zde však uvedeny všechny atributy. Některé atributy jsou specifické přímo pro určitý typ přístupového serveru NAS, respektive každý výrobce může mít určité odlišnosti. Následující atributy jsou standardními. V závorce se nachází číselná hodnota typu atributu. V atributu paketuje použita právě tato číselná hodnota (položka Typ), která zastupuje slovní název atributu (viz obr. 3).

Název atributu (číselné vyjádření typu atributu) – význam:

- User-Name (1) – uživatelské jméno žadatele.
- User-Password (2) – jeho heslo.
- CHAP-Password (3) – CHAP ID + jednorázové heslo (jako jediný druh atributu obsahuje dvě hodnoty).
- NAS-IP-Address (4) – adresa IP klienta RADIUS (přístupového serveru NAS), žádajícího o autentizaci serveru. Ve zprávě Access-Request musí být buď atribut NAS-IP-Address, nebo atribut NAS-Identifier. Obvykle jsou ovšem oba atributy.
- NAS-Port (5) – číslo portu přístupového serveru NAS, na kterém se hlásí uživatel.
- Framed-Protocol (7) – protokol linkové vrstvy. Tento protokol je vyžádán uživatelem (1 = PPP, 2 = SLIP).

- Framed-IP-Address (8) – pomocí tohoto atributu může být zprávou Access-Accept dynamicky přidělena adresa IP uživateli.
- Framed-IP-Netmask (9) – síťová maska uživatele. Takto může být přidělena dynamicky síťová maska uživateli.
- Filter-Id (11) – aktivace filtru pro přístupový port, na kterém se uživatel hlásí.
- Framed-Compression (13) – tento atribut indikuje kompresní metodu.
- Reply-Message (18) – specifikace textu, který má být zobrazen uživateli.
- Callback-Number (19) – určení telefonního čísla pro zpětné volání.
- State (24) – atribut obsahuje párovací informaci pro zprávu Access-Challenge a následnou zprávu Access-Request.
- Idle-Timeout (28) – časový interval, po kterém je nečinný uživatel odpojen.
- NAS-Identifier (32) – Identifikace přístupového serveru NAS (např. jméno).
- CHAP-Challenge (60) – výzva protokolu CHAP generovaná přístupovým serverem.

Předchozí výpis obsahuje velké množství běžných atributů, které vycházejí ze standardu RFC. Firma MikroTik, jako většina firem zabývajících se tímto problémem, definuje více možných atributů pro svoji potřebu. Na příklad při použití balíku HotSpot lze najít atribut obsahující odhlašovací stránku klienta, atributy definující maximální rychlost klientů a další.

1.2 RADIUS Accounting

Protokol RADIUS Accounting slouží k zaznamenávání informací o přihlášení a odhlášení uživatelů k přístupovému serveru NAS. Vzhledem k jeho komplexní struktuře je možno ho požít z řady důvodů. Protokol RADIUS Accounting je popsán dokumentem RFC-2866 [4].

RADIUS Accounting se používá hlavně k zaznamenávání přístupů zaměstnanců z Internetu do firemní sítě. Z těchto informací lze vytvořit ucelený report o tom, kdy a jak byl zaměstnanec přihlášen.

V případě poskytovatelů Internetu se protokolem RADIUS Accounting sleduje, kdy a jak byl který zákazník přihlášen, kolik přenesl dat, jak dlouho byl přihlášen a jiné. V podstatě RADIUS Accounting shromažďuje pro poskytovatele Internetu nezbytné informace, které mohou sloužit i pro vystavení faktur koncovému zákazníkovi.

RADIUS Accounting je velmi podobný protokolu RADIUS. Server RADIUS Accounting obvykle naslouchá na portu UDP 1813. Použití protokolu UDP řeší nedostupnost serveru. Nevýhodou je, že je nutné seskupit záznamy serverů RADIUS Accounting dohromady za účelem vypracování statistik nebo tvorby zákaznických faktur. Při sjednocení záznamů můžou vzniknout duplicitní údaje. Většinou jde o údaje duplicitních požadavků ověření, či chyby otevření spojení. Proto je vhodné tyto záznamy čistit (např. pomocí filtrů databáze).

1.2.1 Popis protokolu

RADIUS Accounting používá stejně jako protokol RADIUS jak stejnou strukturu paketu, tak i stejný princip. RADIUS Accounting využívá dva druhy zpráv. Požadavek a odpověď.

Accounting-Request (kód = 4) odesílá klient protokolu RADIUS Accounting serveru informace, které mají být zapsány (např. informaci o přihlášení či odhlášení klienta).

Accounting-Response (kód = 5) server odpovídá při přijetí zprávy Accounting-Request.

Podobnost protokolů RADIUS Accounting a RADIUS je zřejmá z jejich struktury. Struktura obou protokolů je založena na principu párů Attribute-Value. protokol RADIUS Accounting využívá atributy stejné jako protokol RADIUS (User-Name, NAS-IP-Address, NAS-Port, Framed-Protocol, Framed-IP-Address, Filter-ID atd.). Obsahuje mimo jiné, ale i své specifické atributy, které jsou zajímavé právě pro účtování. V závorce se nachází číselná hodnota typu atributu. V atributu paketu je použita právě tato číselná hodnota (položka Typ), která zastupuje slovní název atributu (viz obr. 3).

Název atributu (číselné vyjádření typu atributu) – význam:

- Acct-Status-Type (40) – atribut specifikuje, o jaký typ zprávy se jedná, zdali se jedná o přihlášení uživatele (1), odhlášení uživatele (2), zapnutí účtování (7), či vypnutí uživatele (8).
- Acct-Delay-Time (41) – čas, jak dlouho se uživatel pokoušel záznam odeslat.
- Acct-Input-Octets (42) – počet přijatých bajtů portem přístupového serveru.
- Acct-Output-Octets (43) – počet odeslaných bajtů do portu přístupového serveru.
- Acct-Session-ID (44) – identifikace sezení uživatele. Tato identifikace je stejná po celou dobu aktivity uživatele při jednom sezení.
- Acct-Session-Time (46) – jak dlouho je uživatel přihlášen.
- Acct-Input-Packets (47) – kolik paketů bylo přijato portem uživatele.
- Acct-Output-Packets (48) – kolik paketů bylo portem odesláno uživateli.
- Acct-Terminate-Cause (49) – důvod přerušení spojení. Může být např.:
    - = 1 … ukončeno uživatelem
    - = 2 … ztráta signálu DCD
    - = 4 … důvodem nečinnosti uživatele
- Acct-Multi-Session-ID (50) – společný identifikátor spolu souvisejících relací.

1.3 Change Of Authorization

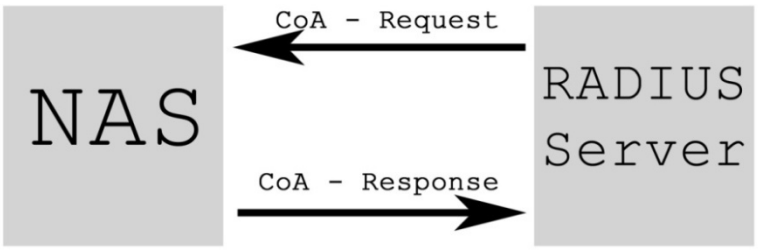
Účelem Change of Authorization (CoA) je dynamická změna (přidání, odebrání) parametrů připojení uživatele inicializované pomocí protokolu RADIUS v rámci už započatého sezení uživatele. Díky paketu Change Of Authorization je možné například změnit parametr Framed-IP-Address (uživatelovu adresu IP). Změna se vyvolá z klientské části serveru RADIUS. Pomocí požadavku CoA-Request se zpráva odešle na přístupový server NAS. Požadavek vyvolá změnu parametru připojení na přístupovém serveru NAS. V tomto případě změnu nastavení serveru DHCP. Uživatel dostane novou adresu IP. Následně odešle NAS odpověď na požadavek serveru RADIUS. Pokud se akci podařilo vyvolat, parametr připojení byl změněn, pak server NAS odpoví zprávou CoA-ACK. Pokud by došlo k chybě změny autorizace, server NAS by odpověděl zprávou CoA-NAK.

Paket Change of Authorization patří do skupiny paketů Dynamic Authorization Extensions to Remote Authentication Dial In User Service (RADIUS) popsaných v RFC 3576 [5]. Do této skupiny patří taktéž zprávy Disconnect.

Podobné aktivity jako u Change of Authorization lze inicializovat též zprávou Disconnect, která, jak už je zřejmé z názvu, slouží k vyvolání přerušení sezení uživatele. Všechny zprávy dynamického rozšíření RADIUS jsou opět odesílané protokolem UDP.

1.3.1 Princip Change Of Authorization

Jak už bylo zmíněno, probíhá komunikace stejně jako u protokolů RADIUS pomocí protokolu UDP. Rozdílem zpráv Change Of Authorization a Disconnect od zpráv protokolu je RADIUS, že požadavky jsou odesílány klientskou částí serveru RADIUS a až následně na požadavek odpovídá přístupový server NAS.

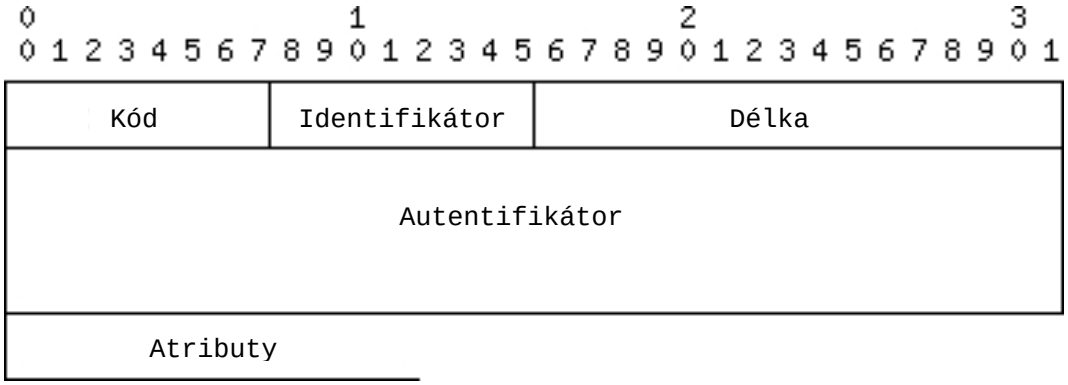


Obr. 4: Průběh komunikace Change of Authorization

Z obr. 4 lze vyčíst, že požadavek CoA-Request oproti protokolům RADIUS odesílá server RADIUS (klientskou aplikací je např. *radclient*). Požadavek je odeslán přístupovému serveru NAS. Ve zprávě CoA-Request (43) jsou obsaženy Attribute-Value páry změny autorizace. Po přijetí přístupový server požadavek zpracuje. Pokud zjistí, že požadavek byl ilegální, aplikování změn odmítne a odpoví zprávou CoA-NAK (45). Naopak pokud požadavek byl akceptovatelný, NAS změní atributy nastavení podle požadavku a odešle kladnou odpověď CoA-ACK (44) serveru RADIUS.

1.3.2 Popis paketu

Struktura paketu Change Of Authorization je identická s protokoly RADIUS. Atributy jsou též odesílány ve stejné podobě jako u předchozích protokolů, tzn. využívají párů Attribute-Value (typ, velikost, hodnota). Pro opětovné přiblížení struktury je přiložen následující obr. 5.



Obr. 5: Struktura paketu Change Of Authorization. [9]

Následně jsou uvedeny dva případy komunikace Change Of Authorization. První požadavek dopadne neúspěšně. A při druhém se změna povede. V požadavku je atribut, který chci u uživatele změnit. Je to parametr Framed-IP-Address.

```
received CoA-Request with id 185 from 82.202.96.34:57139
      Signature = 0xa651062e2b62d10dc6fc25cad00c651f
      User-Name = "Peny"
      Framed-IP-Address = 10.255.200.131

sending CoA-NAK with id 185 to 82.202.96.34:57139
      Signature = 0x9a068b991b88ccbdce789ea48f52f48e
      Error-Cause = 406
      NAS-Identifier = "Peny"
      NAS-IP-Address = 10.255.200.254
```

V příkladu je vidět, že změna se nepodařila. Přístupový server NAS odeslal odpověď CoA-NAK s atributem chyby 406. Tento chybový kód je, ale trochu zavádějící. Chyba znamená nepodporované rozšíření. Důvodem chyby byl pokus o změnu neexistujícího sezení uživatele.

V druhém případě byl otestován uživatel s aktivním sezením. A jak je vidět, změna se podařila, neboť přístupový server NAS odpověděl zprávou CoA-ACK.

```
received CoA-Request with id 218 from 82.202.96.34:46472
      Signature = 0x35f6663dc41168fd2248e5bde23ec67b
      User-Name = "00:60:B3:8C:8E:03"
      Framed-IP-Address = 10.255.200.130

sending CoA-ACK with id 218 to 82.202.96.34:46472
      Signature = 0x9c14c0899874e2e841f1c3c56a79c436
      NAS-Identifier = "Peny"
      NAS-IP-Address = 10.255.200.254
```

Požadavek Change Of Authorization může využít velkého množství atributů. Některé z nich jsou popsány v dokumentu RFC. Většina z nich je ovšem opět v rukách výrobců zařízení, kteří si své atributy definují podle potřeby.

1.3.3 Praktické využití

Praktické využití Change Of Authorization vyplývá už z názvu. Pokud je potřeba změnit parametry uživatelského sezení, je třeba využít tento paket.

Mezi dobré příklady použití patří změny rychlostí připojení uživatele v případě, že politika poskytovatele připojení sítě definuje ve smlouvě na příklad množství dat, které uživatel smí stáhnout plnou rychlostí připojení. Tak při překročení této mezní hodnoty je potřeba vykonat nějakou akci. Tyto akce jsou běžné spíše lokálně přímo na routeru. Ale je možné využít právě možnosti paketu Change Of Authorization. Jakmile uživatel překročí hodnotu, hlídací služba (obvykle skript, spouštěný jako služba, či periodicky, hlídající určité parametry připojení), spustí klientskou část serveru RADIUS s parametrem paketu CoA a atributem určující maximální možnou rychlost připojení uživatele. Klientská část RADIUS serveru odešle požadavek přístupovému serveru NAS, který aplikuje nové nastavení. A odešle odpověď serveru RADIUS o provedení změny.

V podstatě lze požadavek CoA využít k veškerým administrativním úkonům. Pokud je potřeba přidělit nové rozsahy adres IP uživatelům, změnit rychlost připojení a další. Jediné, co představivost ohraničuje, je fakt, že záleží na výrobci, které atributy využívá. Nelze proto využít speciální atributy firmy MikroTik u routeru od firmy Cisco.

2. Praktické nastavení server/klient RADIUS

Protokol RADIUS slouží jako centrální autorizační protokol pro různá řešení HotSpot (kapitola 2.1.2) nebo ho lze použít při autorizaci uživatelů v rozsáhlé síti (kapitola 2.1.1). Cílem je seznámení krok po kroku právě s nasazením protokolu RADIUS k autorizaci bezdrátových klientů v rozsáhlé síti, stejně tak seznámení s balíkem HotSpot od firmy MikroTik.

Dále budu informovat o nasazení protokolu RADIUS Accounting a jeho využití právě ve zmíněných místech HotSpot (kapitola 2.1.2), jako jsou například kavárny a jiná veřejná místa nabízející placené připojení k síti Internet.

Dříve než začnu, je vhodné informovat o mnou navrženém řešení a použitých technologiích, s nimiž budu v následujícím textu pracovat.

2.1 Navrhované řešení

Hlavní důraz je kladen především na konfiguraci propojení se serverem RADIUS, konfiguraci využití klienta a povolení možnosti požadavku CoA (Change of Authorization). Součástí je i nastavení potřebných služeb, které mne osobně nejvíce zajímají, a hlavně takových, jejichž požadavky bude vyřizovat server RADIUS. Řešení, které sám využívám, je založeno na autorizacích klientů pomocí adres MAC.

Uváděné řešení má několik výhod pro klienty. K připojení Internetu stačí, aby klient poskytl správci adresu MAC svého zařízení. Tato adresa se zaregistruje do databáze klientů sítě a o nic více se nemusí klient zajímat. Nicméně toto řešení není dokonalé. Ovšem pro síť s několika sty klienty je dostačující a je velice flexibilní z hlediska jednoduchosti.

Jak už zde bylo uvedeno, volba autorizace pomocí adresy MAC padla hlavně z praktických důvodů. Dalším důvodem je i fakt, že některé služby operačního systému firmy MikroTik (pojednání o operačním systému RouterOS firmy MikroTik následuje v kapitole 2.2.2) zmiňované zde v textu využívají právě jen toto ověření. Jedině služba HotSpot umožňuje více druhů ověření: ověření pomocí adresy MAC, pomocí protokolů PAP, CHAP, dále pomocí cookie webového prohlížeče a v poslední řadě trial (časově omezené připojení).

Firma MikroTik dává architektu sítě v podstatě na výběr. Buď se může použít ověření na principu adres MAC, to ovšem přináší určitá rizika (je známo, že klientskou adresu MAC lze získat odchycením síťového provozu, např. aplikací *wireshark*), ale díky této volbě je architektu sítě dostupná větší škála služeb podporujících ověření pomocí protokolu RADIUS (např. služba filtrující požadavky na bezdrátové připojení k serveru NAS – nikoli připojení do sítě). Na druhou stranu protokol CHAP přináší větší bezpečnost ověření jednorázovým heslem. Problém je v tom, že toto ověření podporuje pouze služba HotSpot. Ideálně bezpečné řešení by bylo využití obou možností zároveň, avšak to je řešení složité na administraci a udržovatelnost databáze. Výběr možnosti záleží na účelu sítě. I když ověření pomocí adres MAC není bezpečnostně ideální (adresu MAC akceptovaných klientů sítě lze snadno zjistit a následně padělat), vzhledem k možnosti využití více služeb podporujících ověření protokolem RADIUS, je podle mého názoru dostačující.

Přesto, že jsem zvolil univerzální řešení autorizace pomocí adres MAC, rozdělil bych navrhované řešení na dvě kategorie:

- Statičtí klienti – řešení je navrženo pro klienty dlouhodobějšího rázu.
- HotSpot – systém vytvořený ryze pro mobilní klienty.

Každá kategorie má své specifické vlastnosti. Pod pojmem statický klient se rozumí klient platící za dlouhodobé připojení k síti, vlastnící obvykle jedno zařízení připojené k síti.

V této souvislosti je tedy možné definovat i pojem mobilní klient. Mobilní klient je zákazník vlastnící více přenosných zařízení jako je notebook, mobilní telefon či PDA. Pomocí těchto zařízení by měl zájem se připojit k síti.

Za ideální demonstrační řešení v následujících kapitolách jsem zvolil kombinaci obou kategorii. Spousta konfiguračních kroků je technologicky závislá na mnou zvoleném řešení. Následující řešení jsem vybral jako univerzální odrazový můstek pro detailnější konfiguraci. Pro demonstraci funkčnosti plně postačuje.

2.1.1 Statičtí klienti

Jak už jsem se zmínil, tuto kategorii jsem navrhl převážně pro statické klienty rozsáhlé sítě. Hlavním důvodem tohoto navržení je fakt, že klient v rozsáhlé síti je obvykle fixován na jednu stálou adresu MAC. Velice pravděpodobně takový klient bude i vázán na jedno místo. Není to ovšem pravidlo, a pokud klientova adresa MAC náleží například notebooku, tak má díky navrhovanému řešení volný pohyb mezi všemi přístupovými servery NAS.

Klient je povinen před prvním připojením do sítě uvést adresu MAC jeho zařízení. Administrátor adresu MAC následně zaregistruje do databáze serveru RADIUS. Jakmile se tak stane, klientovo připojení k síti je aktivní. O nic jiného se klient už nestará (pouze v případě změny zařízení je povinen dodat poskytovateli novou adresu MAC).

Tento scénář považuji výhodným jak pro klienta, který se nemusí po podpisu smlouvy už o nic starat, tak pro administrátora sítě, který je povinen pouze zákazníkovu adresu MAC zaregistrovat.

K tomu, abych mohl řešení navržené kategorie aplikovat na přístupovém serveru NAS, je potřeba nastavit propojení se serverem RADIUS a aktivovat služby routeru MikroTik pro ověřování protokolem RADIUS. Veškeré nastavení klienta RADIUS, popsané v textu je prováděno za pomocí klienta protokolu SSH (Secure Shell – OpenSSH klient, putty). Klient RADIUS firmy MikroTik je možné také nastavit pomocí aplikace winbox (graficka nadstavba konzole).

Nejdůležitější službou je *Wireless*. Tedy služba, která má za úkol připojovat uživatele k routeru pomocí bezdrátového připojení. Pokud je tato služba nastavena špatně, význam jakéhokoli dalšího nastavování klientských služeb RADIUS nemá cenu, protože se klient buď připojí bez jakéhokoli ověření, nebo se nepřipojí vůbec. Pojem *Wireless* v politice zabezpečení routeru MikroTik zastupuje nastavení například WEP, WPA a pro mne hlavně ověření RADIUS pomocí adres MAC.

Další službu, kterou je dobré nastavit pro komunikaci se serverem RADIUS, je „server DHCP“. Pokud je tato služba nastavena, server DHCP se dotazuje serveru RADIUS, zda může přidělit klientovi adresu IP (popřípadě jakou), či nikoli. Tato služba je spíše kosmetickou volbou, ale v rámci flexibility sítě nezbytnou. Opět se jedná o službu, která ověřuje klienta na bázi adres MAC.

Nevýhodou řešení je v podstatě to samé, co je jeho výhodou. Ověřovaní pomocí adres MAC. Na jednu registraci adresy MAC není možné připojit více zařízení.

2.1.2 HotSpot

Pokud by poskytovatel sítě chtěl nabízet připojení mobilních zákazníků na veřejných místech, mohl by využít služeb systému HotSpot. Předchozí problémy s ověřováním na bázi adres MAC může eliminovat například využitím protokolu CHAP.

HotSpot je navržen pro poskytování Internetu na veřejných místech jako jsou například kavárny, školy a jiná. Jeho úkolem je autorizace uživatelů podle specifických postupů a pomocí rozdílných nástrojů a algoritmů. Je to komplexní nástroj, který lze spojit se serverem RADIUS a využívat jeho služeb. Ovšem není to nutné, nýbrž flexibilní. Hlavní výhodou tohoto systému je autorizace klientů i na jiných principech než pouze podle adresy MAC. Zajímavou taktikou je ovšem nejdříve použití ověření na principu adresy MAC, a pokud se ověření nezdaří, může přijít na řadu ověření například pomocí uživatelského jména a hesla.

Firma MikroTik počítala s nasazením svých routerů jako servery HotSpot. Proto lze v seznamu balíků operačního systému RouterOS (pojednání o RouterOS následuje v kapitole 2.2.2) najít balík s názvem HotSpot.

Princip balíku HotSpot v routeru MikroTik spočívá v tom, že pokud se uživatel připojí k síti, router ho odkáže na server HotSpot, respektive na jeho webovou stránku. Zde uživatel zadá své heslo a jméno. A autorizační služba, v našem případě server RADIUS, uživatele ověří, a pokud ho shledá přijatelným, webová stránka HotSpotu se uzavře a uživatel je od této chvíle připojen. Vzhledem k tomuto řešení je uživatel nucen použít stránku přihlášení, jinak mu nepůjde žádná síťová služba. Proto si myslím, že balík HotSpot není vhodný pro sítě se statickými klienty.

Pokud ovšem chci využít RADIUS Accounting, nebo využití autorizace klienta na jiném principu než je adresa MAC, je využití tohoto balíku více než nutností. Protože jedině při aktivním serveru HotSpot lze uplatnit a aktivovat služby protokolu RADIUS Accounting. Názorně aktivaci předvedu.

K aplikování mého řešení systému HotSpot je potřeba aktivovat komunikaci klienta RADIUS se službou *HotSpot*. Mezi způsoby ověřování jsem vybral ověření na bázi adresy MAC. Přestože toto ověřování nepatří mezi hlavní výhodu systému HotSpot, je podle mého uvážení dostatečně názorné.

2.2 Popis architektury

Před vlastním nastavením několik slov k použitým technologiím, o kterých se budu v textu zmiňovat.

2.2.1 Server RADIUS

Pro serverovou část topologie RADIUS jsem využil řešení GNU/Linux z důvodu stability, škálovatelnosti a v neposlední řadě ceně. V rozsáhlé síti je důležité počítat s tím, že jediný server RADIUS nebude stačit. Musí se počítat se záložními servery. Nelze zapomenout, že špatné rozmístění serverů na síti může přinést velmi hodně nepříjemností. Ovšem popis dobré či špatné topologie přesahuje rámec této práce. Pouze okrajově se zde zmíním o využití více serverů.

Jako distribuci GNU/Linux je použit GNU/Debian. Jedná se podle mne o jednu z nejlépe administrovatelnou distribuci. Nesmím ovšem zapomenout, že lze server RADIUS provozovat na jakékoli distribuci. Jedna z vynikajících předností distribuce GNU/Debian je perfektní systém pro správu balíků APT.

V dnešní době existuje spousta serverů RADIUS, proto jsem se při volbě serveru RADIUS inspiroval dokumentací klienta RADIUS od firmy MikroTik [2]. Zde tvůrci doporučují tři servery RADIUS, které byly s jejich klientem RADIUS otestovány. Server RADIUS Steel-Belted Radius Server od firmy Juniper Networks je záležitostí hardwarovou. Takové řešení je zajímavé pro řadu administrátorů (řada administrátorů dává přednost hardwarovému zařízení, protože je často lépe sladěné: operační systém s aplikací a administrací), ovšem pro mne nedostupné. Dalším serverem RADIUS zmiňovaným v dokumentaci firmy MikroTik je XtRadius. Od tohoto serveru RADIUS mne odradila velice slabá dokumentace programu, malý počet rozšíření a absence podpory protokolu MS-CHAP. Poslední testovaný server – FreeRADIUS – má naopak dokumentaci velmi dobře propracovanou, dobrou podporu ověřovacích algoritmů a jedná se o aplikaci licencovanou veřejnou licencí GPL.

Z výše uvedených důvodů byl vybrán server FreeRADIUS. V definici typu přístupových serverů NAS je možné u něj najít typ přímo určený pro komunikaci s routery MikroTik. Použití této definice je velkým přínosem, protože obsahuje seznam specifických atributů této firmy.

Jedinou nevýhodou serveru FreeRADIUS je, že jeho databáze klientů v základní instalaci balíku v mnou zvolené distribuci je lokálního charakteru (v souboru `users`). To by mělo za následek špatnou distribuci databáze záložním serverům. Je tedy velice dobré se uchýlit k použití modulu pro FreeRADIUS, který je schopen pracovat s databázemi SQL.

Jako databáze můžou sloužit MySQL, PostgreSQL, Lightweight Directory Access Protocol (LDAP), Microsoft SQL Server, Oracle SQL, či Kerberos 5. Volba druhu systému databáze víceméně záleží na vkusu administrátora. Osobně bych zvolil systémy SQL. Díky jazyku SQL je možná data následně snadno zpracovávat. Jelikož popisuji nasazení serveru RADIUS do prostředí tvořeného operačním systémem GNU/Linux, nelze použít Microsoft SQL Server. Systém Oracle SQL je velice oblíbený a používaný systém SQL. Bohužel má několik nedostatků: jedná se o komerční produkt. Pro GNU/Debian existuje pouze verze Express s absencí možnosti clusterování a pro komunikaci s oblíbeným skriptovacím jazykem *php* je nutno doinstalovat rozšíření. Moje podmínky (licence GPL, funkce v operačním systému GNU/Linux, clusterovatelný systém, podpora skriptovacích jazyků *php* a *perl*) splňují pouze MySQL a PostreSQL. Já jsem z těchto dvou systémů zvolil systém MySQL především kvůli znalosti funkcí jazyka *php* pro operace se systémem MySQL. Tento faktor byl pro mne ve výběru systému databáze rozhodující.

2.2.2 Klient RADIUS

Mé řešení je postaveno na routerech od firmy MikroTik, respektive na operačním systému RouterOS od zmiňované firmy. RouterOS je modulární (snadná instalace rozšíření v podobě balíků) proprietální operační systém od firmy MikroTik, určený pro veškerá jejich zařízení.

Tento operační systém jsem vybral, protože se jedná o komplexní systém primárně určený pro Wi-Fi routery. Obsahuje veškeré potřebné nástroje pro poskytovatele sítě Internet (routovací protokoly, NAT, paket filter, monitorovací nástroje, nástroje pro řízení datového toku a mnoho dalších). A dává možnost výběru architektům sítě (např. možnost využití různých druhů routovacích protokolů – RIP, OSPF, BGP). Dalšími důležitými determinanty mého výběru jsou odlišné možnosti administrace operačního systému (telnet, ssh, grafická nadstavba ssh – aplikace winbox, www). Posledním určujícím faktorem výběru bylo rozšíření operačního systému RouterOS. Vzhledem k ceně a rozsáhlosti funkcí se stal mezi poskytovateli bezdrátového připojení k síti velice populární.

RouterOS je relativně levný operační systém jednoznačně určený pro směrovače, ať už postavené na architektuře x86, nebo přímo pro produkt s názvem RouterBoard od stejnojmenné firmy MikroTik. RouterBoard je malý router disponující procesory od různých firem (např. Infineon), takže je potřeba vlastnit operační systém určený přímo pro tyto procesory (druh/výrobce procesoru se liší ve verzích RouterBoardu).

Operační systém GNU/Linux disponuje veškerou funkčností jako RouterOS, takže bych mohl ho brát jako alternativu. Ovšem díky rozdílné architektuře procesorů není možné používat operační systém GNU/Linux pro všechny routery RouterBoard. Čím vyšší počet různých operačních systémů je v síti nasazeno, tím se zvyšuje obtížnost administrace. A to je poslední důvod mé volby operačního systému RouterOS.

Router MikroTik je velice zajímavé zařízení v poměru cena/výkon. Obsahuje precizní nastavení klientského připojení v takzvaných „Simple Queue“ (nástroj pro řízení datového toku). Umožňuje například nastavit rychlost připojení, prioritu a jiné. Velice lákavá je i možnost psaní vlastních skriptů, které mohou například periodicky měnit nastavení klientského připojení ve zmiňovaném „Simple Queue“.

Avšak implementace balíku HotSpot má jeden malý nedostatek. Po připojení klienta si vytvoří v „Simple Queue“ dynamický záznam, který vychází z přijatého nastavení ze strany serveru RADIUS. Ten nelze ze strany routeru měnit. V momentě, kdy politika poskytovatele Internetu závisí na změně rychlosti připojení klienta při stažení velkého množství dat, je tento nedostatek fatální. Tímto se stane nemožné aplikovat dynamické změny parametrů „Simple Queue“ pro specifického klienta pomocí vlastních skriptů (např. změna maximální rychlosti připojení klienta).

K tomu, abych toto omezení obešel, je potřeba použít pouze některé atributy předávající se v paketu Access-Accept. Takto lze docílit vytvoření dynamického záznamu v „Simple Queue“ pro server HotSpot, ale ne už pro klienty. Pak jde uplatnit statické „Simple Queue“ vytvořené například pomocí tabulky ARP a skriptem.

Na druhé straně MikroTik RADIUS klient podporuje práci s CoA (Change of Authorization). Požadavek CoA je odesílán ze strany serveru RADIUS k routeru. Tento požadavek je velice přínosný v momentě, kdy potřebujete routeru říci, že má změnit nějaký parametr v připojení síťového klienta, který je ověřen serverem RADIUS.

MikroTik RADIUS klient sice poskytuje funkce i pro další služby systému RouterOS, jako například ppp, login či služby s názvem telephony. Jedná se ale o služby, které nejsou určeny pro bezdrátovou komunikaci. Pro tuto práci tedy neadekvátní.

2.3 Nastavení serverové části

Tato kapitola obsahuje popis instalace aplikace FreeRADIUS v distribuci GNU/Debian a následovně její konfiguraci pro použití SQL databází a zařízení od firmy MikroTik.

2.3.1 Instalace aplikace FreeRADIUS a databáze MySQL

Díky balíčkovacímu systému APT je instalace aplikace FreeRADIUS více než snadná. Jako správce (uživatel root) je třeba spustit příkaz:

```
apt-get install freeradius freeradius-mysql
```

FreeRADIUS je nainstalovaný. Následuje konfigurace souborů. Nejčastěji si FreeRADIUS ve vašem systému vytvoří adresáře */etc/raddb*, nebo */etc/freeradius*. V distribuci Debian se nachází konfigurační soubory v adresáři */etc/freeradius*. Po nahlédnutí do adresáře lze zjistit, že se zde nachází řada konfiguračních souborů. Popíšu zde jen nejdůležitější z nich.

Nejdůležitějším souborem je radiusd.conf s hlavní konfigurací serveru. Je důležité, že mimo jiné specifikuje jaké algoritmy zvolit při žádostech o autentifikaci, autorizaci a účtování (accounting). Lze zde najít i volbu různých databází pro ověřování a řadu dalších parametrů serveru FreeRADIUS, které více méně překračují rámec této práce.

Druhý nejdůležitější soubor pro implementaci FreeRADIUS s využitím SQL je `sql.conf`. Jsou zde uvedeny připojovací informace k databázi, použitý typ databázového systému, definování názvu tabulek a dokonce i definice dotazů SQL do databáze.

Poslední dva soubory, které budu popisovat, mají pro následující implementaci pouze informativní účel, ale je dobré o nich vědět, protože se budu o nich v textu zmiňovat.

Soubor `users` slouží jako jednoduchá textová databáze uživatelů. Vzhledem k tomu, že systém MySQL jeho účel důkladně zastoupí, není potřeba definovat uživatele lokálně zde. Jak jsem uvedl v úvodu, nastavování přes tyto lokální databáze není zas tak flexibilní v momentě, když používáme více serverů RADIUS v síti. Doporučuji ovšem do tohoto souboru informativně nahlédnout. Soubor obsahuje kvalitní komentáře a jsou v něm vytvořeny příklady záznamů uživatelů, které se v podstatě neliší od těch, které budeme vkládat do databáze MySQL.

Posledním z nejzajímavějších souborů je soubor `clients.conf`. Uvnitř jsou definované přístupové servery NAS (Network Access Server). Nemá-li NAS záznam v tomto souboru, pak s ním FreeRADIUS nekomunikuje. I tento soubor opět nahradíme databází. Je potřeba se ujistit, zda je FreeRADIUS nakonfigurován, aby se do databáze serverů NAS díval.

2.3.2 Konfigurace FreeRADIUS pro práci s MySQL

Pro připojení FreeRADIUS k databázi MySQL je zapotřebí učinit několik nepatrných změn v konfiguračních souborech.

První z nich je `sql.conf`. Do souboru je nutno doplnit údaje o tom, jaký driver k databázovému systému je použit, kde se databáze nachází, uživatelské jméno a heslo k ní.

```
driver = "rlm_sql_mysql"
server = "localhost"
login = "user"
password = "heslo"
radius_db = "radius_database"
```

Konfigurace připojení k databázovému systému MySQL je dokončena a teď je pouze za potřebí serveru FreeRADIUS sdělit, že má modul *sql* využívat. Tento krok se zrealizuje editací souboru `radiusd.conf`.

Uvádím příklady bloků, které je potřeba pozměnit. Nezmiňované části bloků jsou v mém případě nezměněné a doporučuji je měnit pouze podle uvážení. Zájemce odkazuji na dokumentaci serveru FreeRADIUS [6]. Lze například aktivovat i jiné autorizační systémy, jako je například Lightweight Directory Access Protocol a jiné.

V bloku `authorize` se musí aktivovat modul *sql* smazáním komentáře. Je dobré přidat komentář před modul *files*, jinak se server bude poohlížet po souboru `users`. U bloku `preacct` platí to samé o modulu *files* jako u předchozího příkladu. Nakonec se musí v bloku `accounting` opět aktivovat modul *sql*.

```
authorize {
      # files
      sql
}

preacct {
      # files
}

accounting {
      sql
}
```

Po provedení těchto kroků je FreeRADIUS nastavený pro spojení s databází MySQL.

2.3.3 Vytvoření tabulek v databázi MySQL

Nejprve je třeba si vytvořit skupinu řídících tabulek v databázi. FreeRADIUS dává uživateli v tomto ohledu vcelku volnou ruku. Záleží na každém, jakou strukturu tabulek si zvolí. Je vhodné pročíst si soubor `sql.conf`. Z něj je dobře patrné, co se žádá.

V souboru `sql.conf` jsou uvedeny příklady dotazů do databáze. Po jejich přečtení a po přečtení komentářů je zřejmé, co provádějí (viz příloha A).

Ze struktury souboru `sql.conf` vyplývá, že ani tabulky nejsou předem určeny. Každý si je může definovat podle potřeby. Nicméně pro ulehčení práce tvůrci serveru FreeRADIUS dali k dispozici soubor `db_mysql.sql`, který má v sobě defaultní strukturu tabulek. Díky tomuto souboru je zaručena 100% kompatibilita se souborem `sql.conf`. Vytvoření tabulek v MySQL je proto velice příjemnou záležitostí. Uvedený soubor je přiložen jako příloha této práce.

Postup vytvoření tabulek pomocí souboru db_mysql.sql je následující:

```
mysql -h adresa_serveru -u uziv_jmeno -pheslo nazev_databaze \
< db_mysql.sql
```

Tabulky v databázi MySQL jsou tedy vytvořeny a databáze je připojena k serveru FreeRADIUS. Zbývá pouze vysvětlit, co tabulky vyjadřují, a jaké hodnoty lze do nich vkládat, respektive co každý sloupec tabulek vyjadřuje. Podrobný popis uvádím v příloze B.

Na straně serveru je tedy vše nastaveno a připraveno k použití. Zbývá nakonfigurovat klientskou část, otestovat připojení a nasadit do ostrého provozu.

2.4 Nastavení klienta

Klient RADIUS v RouterOS firmy MikroTik zvládne spoustu věcí. Autorizaci HotSpotu, PPP, PPPoE, PPTP, L2TP, ISDN atd.

Pro nastavení klienta RADIUS je potřeba přejít do jeho úrovně v konzoli příkazem */radius***.**

```
[Peny@IvDoma] > /radius print
Flags: X – disabled
 #   SERVICE                              CALLED-ID DOMAIN ADDRESS
SECRET
```

Jak je z příkladu vidět, není nyní žádný klient nadefinován. Je možné začít výpisem možností.

```
[Peny@IvDoma] > /radius ?
MikroTik RouterOS can authenticate for PPP, PPPoE and PPTP
connections

.. -- go up to root
add -- Create a new item
incoming -- Incoming messages management
```

Ve výpise jsem uvedl pouze dvě volby, které mne jako jediné zajímají:

- **add** – volba přidá nové propojení k serveru RADIUS (výpis možností viz příloha C).
- **incoming** – je volba konfigurace CoA.

Pro přidání serveru RADIUS s parametry je potřeba využít syntaxe příkazu:

```
[Peny@IvDoma] > /radius add address=82.202.96.34 secret=xxx
service=wireless,dhcp
```

Přidal jsem server RADIUS s adresou IP 82.202.96.34. Sdílené tajemství bylo nastaveno na řetězec xxx a byly aktivovány služby, které budou komunikovat se serverem RADIUS. Pro komunikaci RADIUS byly aktivovány služby HotSpot, Wireless a DHCP.

Pro ověřování byly učiněny všechny potřebné kroky. Doporučuji ověřit nastavení následovně:

```
[Peny@IvDoma] > /radius print detail
Flags: X – disabled
 0 service= HotSpot,wireless,dhcp called-id="" domain=""
address=82.202.96.34 secret="xxx" authentication-port=1812
accounting-port=1813 timeout=300ms accounting-backup=no realm=""
```

Parametry *called-id*, *domain* ani *realm* není potřeba primárně nastavovat, pokud pro jejich využití není opodstatnění. Nastavení *portů*, *timeout* a *accounting-backup* zůstane na výchozích hodnotách. S těmito hodnotami není zapotřebí manipulovat, leda v případě, kdy například server RADIUS naslouchá na jiném portu.

2.5 Nastavení RADIUS Authorization

Nyní je přidaný komunikační most mezi routerem (klientem RADIUS) a serverem RADIUS. V následující kapitole se zabývám úpravou služeb routeru, aby komunikovaly s autorizačním serverem.

Popisuji pouze změny, které je nutno udělat, aby bylo možné dokončit komunikační kanál. Nastavení samotných služeb je mimo rámec této práce, a proto předpokládám, že služby jsou již nakonfigurovány a funkční.

2.5.1 Wireless

V první části této kapitoly se věnuji službám, které komunikují se serverem RADIUS a autorizují uživatele pomocí adres MAC. Nejdůležitější službou je určitě *Wireless*. Tedy služba, která má za úkol připojovat uživatelé k routeru pomocí bezdrátového připojení. Pojem *Wireless* v politice zabezpečení routeru MikroTik zastupuje nastavení například WEP, WPA a pro nás hlavně ověření RADIUS pomocí adres MAC.

Sám o sobě router MikroTik vlastní lokální databázi povolených uživatelských adres MAC. V případě většího počtu routerů se stává tato databáze neefektivní. Příklad takového access listu je zde:

```
[radmin@CK_Lhota] > /interface wireless access-list print
Flags: X - disabled
 0   ;;; Dudek
     mac-address=00:4F:62:03:C1:7A interface=Josi_lhota_nova
authentication=yes forwarding=yes ap-tx-limit=0  client-tx-limit=0
private-algo=none private-key=""
 1   ;;; Drozd
     mac-address=00:4F:62:05:0F:EB interface=Josi_lhota_nova
authentication=yes forwarding=yes ap-tx-limit=0  client-tx-limit=0
private-algo=none private-key=""
```

I když je takový *access list* velice podrobný, správa 20 access listů může být (finančně) nákladná. Proto nastavím službu *Wireless* pro komunikaci s autentizačním serverem RADIUS. A celou tuto databázi uživatelů centralizuji.

K tomu, aby bylo možné toto nastavení uplatnit, je potřeba si vytvořit nový bezpečnostní profil, který bude politiku uplatňovat. V poslední řadě je nutno tento profil aktivovat na specifickém bezdrátovém interface.

Nyní si přidám nový bezpečnostní profil, přidělím mu jméno RADIUS a pomocí volby *radius-mac-authentication* zapnu komunikaci se serverem RADIUS:

```
[radmin@CK_Lhota] > /interface wireless security-profiles add
name=RADIUS radius-mac-authentication=yes
```

Pro ujištění, zda se profil opravdu vytvořil, lze využít opět volbu *print* k vypsání seznamu profilů. Ve výpise by měla být vidět zapnuta volba *radius-mac-authentication=yes*. Takže mám vytvořen profil a nyní už jen zbývá profil aktivovat na specifickém zařízení. Aktivaci provedu příkazem:

```
[radmin@CK] > /interface wireless set smerovka_2,4 security-
profile=RADIUS
```

A ověřím opět příkazem print:

```
[radmin@CK_Lhota] > /interface wireless print
Flags: X - disabled, R - running
 1  R name="smerovka_2,4" … security-profile=RADIUS
```

2.5.2 Server DHCP

Další službu, kterou je potřeba nastavit pro komunikaci se serverem RADIUS, je „server DHCP“. Tato služba je v rámci flexibility sítě nezbytná.

Protože zde pojednávám pouze o autorizaci RADIUS, tak si opět na funkčním serveru DHCP tuto volbu pouze zapnu a podrobněji se jí zabývat nebudu. Nejdříve je potřeba zjistit, kde volbu hledat a zda je na mém příkladu aktivní.

```
[radmin@CK_Lhota] > /ip dhcp-server print detail
Flags: X - disabled, I - invalid
 0 name="dhcp1" … use-radius=yes
 1 name="dhcp2"
```

Z výpisu je zřejmé, že jsou nakonfigurovány dva servery DHCP. První z nich s aktivovanou volbu *use-radius=yes*, pro komunikaci se serverem RADIUS. Aktivace volby u druhého serveru se realizuje následovně:

```
[radmin@CK_Lhota] > /ip dhcp-server set dhcp2 use-radius=yes
```

Pro ověření změny je možné si opět nechat vypsat seznam serverů DHCP pomocí příkazu *print*.

Toto nastavení je dobré uplatňovat pro statické uživatele. Vytvořit databázi o stovkách uživatelských jmen a hesel je v rozsáhlé síti pracné. Uživatele by jenom zdržovala logovací webová stránka serveru HotSpot. Proto nastavení služeb *Wireless* a *DHCP* pro zabezpečení přístupu statických klientů je dostačující. Toto nastavení též zajišťuje možnost mobility klientů, ale pouze při použití stále stejného zařízení (např. notebooku).

2.5.3 HotSpot

Pro zajištění plné mobility klienta, to znamená klienta platícího za připojení k síti nezávisle na zařízení, bych musel využít služeb balíku *HotSpot*.

HotSpot lze připojit k serveru RADIUS, jak už je v routeru MikroTik běžné, pouze aktivováním přepínače. Tento přepínač se nachází v profilu serveru HotSpot, stejně jako tomu bylo u služby *Wireless*.

```
[radmin@CK_Lhota] > /ip hotspot profile print
lags: * - default
 0 * name="default" … login-by=mac,cookie,http-chap … use-radius=no
```

Z předchozího výpisu je vidět, že v tomto profilu není využití serveru RADIUS aktivováno. HotSpot je nastaven na ověření na bázi adresy MAC, protokolu CHAP a cookie. Aktivace se provede editací záznamu:

```
[radmin@CK_Lhota] > /ip hotspot profile set default use-radius=yes
```

Po aktivaci se zvýšil počet konfigurací v profilu. Většina těchto nových možností je věnována protokolu RADIUS Accounting.

```
[radmin@CK_Lhota] > /ip hotspot profile print
Flags: * - default
 0 * name="RADIUS" … use-radius=yes radius-accounting=yes radius-
interim-update=received nas-port-type=wireless-802.11 radius-
default-domain="" radius-location-id="" radius-location-name=""
```

Jediné, co z nových vlastností je zde vhodné pospat, je *nas-port-type=wireless-802.11*. V příloze B, zmiňuji informaci o sloupci port v tabulce *nas*. Nechal jsem ve sloupci port vyplněno NULL s poznámkou, že NAS odesílá číslo portu v paketu. Jedná se právě o tento port. Zde lze změnit typ portu.

```
[radmin@CK_Lhota] > / ip hotspot profile set default nas-port-type=?
NasPortType ::= cable | wireless-802.11 | ethernet
```

V příkladu je vyplněn port wireless-802.11. Posledním krokem nastavení profilu HotSpot jest jeho přiřazení ke správnému serveru. Postup je podobný jako při nastavení služeb Wireless:

```
[radmin@CK_Lhota] > ip hotspot set server1 profile=RADIUS
```

Takto se aktivují služby pro práci s serverem RADIUS. Jsou to služby, které jsou nezbytnou součástí při poskytování zabezpečeného připojení Wi-Fi.

MikroTik RADIUS klient sice poskytuje funkce i pro další služby systému RouterOS jako například ppp, login či telefony, pro tuto práci tedy neadekvátní.

2.6 Nastavení RADIUS Accounting

V nastavení autorizace RADIUS jsem uvedl, že funkce RADIUS Accounting je přímo závislá na využití balíku HotSpot a jeho aktivaci. V minulé kapitole jsem ukázal jak nastavit profil HotSpot a jak ho následně přiřadit k aktivnímu serveru HotSpot.

Teď upravím zmiňovaný profil tak, aby odesílal serveru RADIUS pakety RADIUS Accounting, které bude server zpracovávat a též ukládat do databáze.

Na straně serveru je vše připraveno, takže server už čeká pouze na požadavky. Je tedy potřeba nastavit pouze klientskou část, která se nachází na routeru MikroTik.

V poslední části toto nastavení otestuji a zkontroluji, jak požadavek vypadá v režimu „debug“ serveru FreeRADIUS a také jak vypadá v logu MikroTik.

2.6.1 Aktivace RADIUS Accounting

Aktivace RADIUS Accounting v routeru MikroTik je závislá na konfiguraci serveru HotSpot a jeho profilu, jak už zde bylo uvedeno. Pro informaci o současném nastavení RADIUS Accountingu je dobré si vypsat nastavení profilu:

```
[radmin@CK_Lhota] > /ip hotspot profile print
Flags: * - default
 0 * name="RADIUS" … radius-accounting=no
```

Jak je vidno z výpisu *radius-accounting=no*, RADIUS Accounting je vypnut. Pro aktivaci funkce je potřeba zadat příkaz:

```
[radmin@CK_Lhota] > /ip hotspot profile set RADIUS radius-
accounting=yes
```

Toto je z nastavení RADIUS Accounting vše.

2.7 Zapnutí přijímání požadavku Change of Authorization

V úvodu kapitoly jsem uvedl, co znamená požadavek Change of Authorization (dále jen CoA). Nyní se budu věnovat tomu, jak ho lze v routeru MikroTik jednoduše aktivovat.

Jednoduchým příkazem se tedy požadavek aktivuje. Výpis níže podává důkaz, že je vskutku aktivován:

```
[Peny@IvDoma] > /radius incoming set accept=yes
[Peny@IvDoma] > /radius incoming print
  accept: yes
    port: 1700
```

CoA požadavek má nastaven výchozí port 1700, který není potřeba měnit.

2.8 Testování

Nastavil jsem autorizaci klientů Internetu na základě adresy MAC v routeru MikroTik. Teď už zbývá popsat, které parametry je nutné vložit do databáze, aby autorizace proběhla v pořádku. Tyto údaje z databáze pak použije server FreeRADIUS.

Než se zadá informace o uživateli, respektive jeho ověřovací informace, je zapotřebí vložit do databázové tabulky *nas* testovací router MikroTik.

```
mysql> INSERT INTO nas VALUES (NULL,"10.255.200.254","Peny",
"mikrotik",NULL,"xxx","JCKSMTP","NAS");
```

Vytvořil jsem záznam o serveru NAS s adresou IP 10.255.200.254. Je nutné zvolit adresu IP síťové karty, která je topologicky nejblíže serveru RADIUS. Pojmenoval jsem ji NAS „Peny“. A upřesnil typ MikroTik. Porty jsem nechal nastaveny na výchozí hodnotu, tzn. na hodnotu zaslanou požadavkem NASu. Sdílené tajemství (vysvětlení pojmu sdílené tajemství na straně 13) jsem zvolil stejné jako při konfiguraci klienta xxx. Komunita SNMP (Simply Network Management Protocol – protokol navržen pro monitoring a management sítě [13]) je zvolena taková, kterou provozuji. Komunita SNMP je sdílené tajemství, které zabezpečuje komunikaci mezi serverem a klientem protokolu SNMP. Jako popis jsem nastavil jednoduše NAS. Tímto je uzavřeno nastavení klienta pro testovaní.

Při použití RouterOS verze 2.9 je klientova adresa MAC považována v serveru RADIUS za parametr *UserName*. V případě HotSpotu, záleží na typu zvolené autorizace. Pro další práci je tato informace důležitá. Pro novou verzi 3.0 už ale není aktuální. V novější verzi je možné si atribut *UserName* nastavit podle potřeby.

Server FreeRADIUS pokládá do databázového systému dotazy ve specifickém pořadí. Z následovně uvedeného pořadí je vidět, že individuální nastavení klienta má přednost před skupinou:

```
authorize_check_query
authorize_group_check_query
authorize_reply_query
authorize_group_reply_query
```

Prvním krokem k přidání uživatele do databáze v tabulce *usergroup* je přiřazení skupiny a její priority k *UserName*. V praxi to znamená vložit za *UserName* klientovu adresu MAC, za *GroupName* skupinu Wi-Fi a prioritu například 1.

Dále je potřeba v tabulce *radgroupcheck* nastavit, aby server RADIUS propouštěl skupinu Wi-Fi s ověřením prázdného hesla. Ve výsledku bude tabulka obsahovat záznamy jako *GroupName*, opět skupinu Wi-Fi, Attribute bude *user-password*, *op* bude „==“ a Value bude prázdná hodnota. Tím se oznamuje serveru RADIUS, že tento klient se bude ověřovat prázdným heslem. Připomínám, že popis tabulek uvádím v příloze B.

V tabulce *radgroureply* je možné detailněji nastavit parametry klientského připojení. Informace z tabulky *radgroupreply* jsou odesílány serverem RADIUS v paketu Access-Accept.

Lze nastavit též individuální nastavení pro každého klienta v tabulkách *radcheck* a také *redreply*. Skupinové nastavení má výhodu pro větší počet klientů. Proto předvedu na svém příkladu vložení klientské adresy MAC. Následně si nastavím ověřovací postupy pro celou skupinu. Přidám si do databáze i některou informaci RHS určenou pouze jednomu klientovi.

Uvedu nyní příklady dotazů SQL pro vložení správných hodnot do databáze SQL. Začnu přiřazením klienta do skupiny v tabulce *usergroup*.

```
mysql> INSERT INTO usergroup VALUES ("00:13:02:BB:A7:23","Wi-Fi",1);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM usergroup;
+-------------------+-----------+----------+
| UserName          | GroupName | priority |
+-------------------+-----------+----------+
| 00:50:FC:A7:D5:71 | Wi-Fi     |        1 |
| 00:13:02:BB:A7:23 | Wi-Fi     |        1 |
| Peny              | HotSpot   |        2 |
+-------------------+-----------+----------+
```

Jak je vidět z výpisu, vložení proběhlo úspěšně. Následně vložím do tabulky *radgroupcheck* hodnoty LHS (atributy, které slouží k porovnávání hodnot v databázi, např. k porovnání hesla poskytnutého uživatelem s heslem uloženým v databázi), aby server FreeRADIUS kladně ověřoval skupinu Wi-Fi s prázdným heslem. Toto nastavení je možné aplikovat i v tabulce *radcheck*, ale vzhledem k tomu, že v mém případě je nastavení hesla stejné pro všechny klienty, volím skupinové nastavení.

```
mysql> INSERT INTO radgroupcheck VALUES (NULL,"Wi-Fi","user-
password","==","");
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM radgroupcheck;
+----+-----------+---------------+----+-------+
| id | GroupName | Attribute     | op | Value |
+----+-----------+---------------+----+-------+
|  5 | Wi-Fi     | user-password | == |       |
+----+-----------+---------------+----+-------+
```

Tyto informace by ale stále nestačily. Stále nedojde k samotnému ověření, protože server zpravidla ještě neví, jakou metodou hesla porovnávat. Vzhledem k tomu, že metody porovnání hesel budou pro celou skupinu stejné, tak je dobré vložit informaci o porovnání hesel též do tabulky *radgroupcheck*.

```
mysql> INSERT INTO radgroupcheck VALUES (NULL,"Wi-Fi","Auth-
Type","==","local");
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM radgroupcheck;
+----+-----------+---------------+----+-------+
| id | GroupName | Attribute     | op | Value |
+----+-----------+---------------+----+-------+
|  6 | Wi-Fi     | Auth-Type     | == | local |
|  5 | Wi-Fi     | user-password | == |       |
+----+-----------+---------------+----+-------+
```

Ověřovací typ local určuje, že je možno ověřovat hesla uložená v čistém textu (Plain-Text), tedy nezašifrovaná. Není to optimální nastavení, ale jediná možnost při použití metody CHAP.

Takto nastavené hodnoty v tabulkách LHS jsou už nyní schopny ověřit klienta na bázi adresy MAC. V dalším kroku uvedu příklad hodnoty RHS (atributy definující, co se má udělat, pokud souhlasí porovnání LHS), která je pouze informativní. Jiné hodnoty RHS není zapotřebí v mém příkladu vkládat. Pouze bych chtěl poukázat na princip hodnot RHS, jejichž využití bude patrné z upovídaného (debug) režimu serveru FreeRADIUS, který budu popisovat později.

Následující příklad reprezentuje princip hodnot RHS. Pro ukázku se vloží do tabulky *radreply* atribut, který má při správné implementaci klientské části protokolu RADIUS zobrazit přístupovému serveru NAS řetězec znaků definovaný jako hodnota atributu. I když se tato zpráva nikde výrazně neprojeví, v logu přístupového serveru NAS bude řádně zaznamenána.

```
 mysql> INSERT INTO radreply VALUES
(NULL,"00:13:02:BB:A7:23","Reply-Message",":=","Ahoj Test!");
```

V tomto případě zde není žádná podstatná informace. Je ale důležité ukázat princip. Tato tabulka totiž slouží k určení atributů a hodnot, které jsou pro daného uživatele specifické a nevztahují se na celou skupinu uživatelů. Užitečnější je určitě případ, kdy je potřeba uživateli přidělit napevno adresu IP:

```
 mysql> INSERT INTO radreply VALUES (NULL,"00:13:02:BB:A7:23",
"Framed-IP-Address",":=","10.255.200.134");
```

Pokud bych měl zájem přidělit atributy celé skupině uživatelů, třeba jako rozsah DHCP IP, mohl bych zvolit tabulku *radgroupreply*. Tabulka pro přidělení atributů skupině má stejnou strukturu jako *radreply*. Vzhledem k vytvoření pouze jednoho záznamu RHS není potřeba tuto tabulku využívat.

2.8.1 RADIUS Authorization

Teď, když už jsou veškeré potřebné informace vloženy do databáze, je vhodné autorizaci otestovat. Z tohoto důvodu je rozumné regulérně vypnout běžící proces FreeRADIUS a následně ho zapnout v režimu „debug“.

Pro zjištění, jakým přepínačem se zapíná režim „debug“ serveru FreeRADIUS, je dobré prolistovat nápovědu:

```
email:~# freeradius –help
```

Z výpisu nápovědy lze vyčíst, že hledaný přepínač je `–X`. Pokud by nebyl výpis dostatečně informativní, příkazem `man freeradius` si lze prostudovat manuálové stránky programu.

Vypnutí FreeRADIUS a spuštění v režimu „debug“ (jako uživatel root):

```
/etc/init.d/freeradius stop
freeradius –X
```

Nejdříve proběhne řada inicializačních informací o serveru FreeRADIUS, například jaké má zavedeny moduly pro ověřování, accounting a mnoho dalších. Co je ale podstatné, začne vypisovat informace o požadavcích o ověření.

Teď je možné se přihlásit a sledovat zároveň ve výpise serveru FreeRADIUS co se bude dít:

```
rad_recv: Access-Request packet from host 10.255.200.254:34947,
id=116, length=183
      NAS-Port-Type = Wireless-802.11
      Calling-Station-Id = "00:13:02:BB:A7:23"
      Called-Station-Id = "server1"
      NAS-Port-Id = "AP"
      User-Name = "00:13:02:BB:A7:23"
      NAS-Port = 2150629399
      Acct-Session-Id = "80300017"
      Framed-IP-Address = 10.255.200.134
      Mikrotik-Host-IP = 10.255.200.134
      User-Password = ""
      Service-Type = Login-User
      WISPr-Logoff-URL = http://10.255.200.129/logout
      NAS-Identifier = "Peny"
      NAS-IP-Address = 10.255.200.254
```

Z požadavku o přístup je jasné, že NAS komunikuje se serverem RADIUS. Při prostudování požadavku Access-Request je z něho možné zjistit spoustu zajímavých informací. Je naprosto jasné, že NAS zasílá informace v syntaxi párů Attribut-Value. Pod tímto pojmem rozumím dvojici atribut a k němu přiřazena hodnota (popis atributů viz příloha D).

Když se podívám na uživatelské jméno a heslo žadatele, je mi jasné, že tyto informace jsou přesně těmi požadovanými informacemi. Přesto je dobré se přesvědčit a podívat se na odpověď serveru FreeRADIUS. Zajímá nás, zda odpověď server RADIUS odeslal, jakého je typu a co obsahovala za atributy. Pokud je vše správně nastaveno, po prohledání výpisu upovídaného (debug) režimu serveru FreeRADIUS lze najít odpověď podobnou této:

```
Sending Access-Accept of id 116 to 10.255.200.254 port 34947
      Reply-Message := "Ahoj Test!"
      Service-Type = Framed-User
```

Z identifikačního čísla je zřejmé, že jde o odpověď k předchozímu požadavku. V tomto případě je odpověď kladná. Jde o odpověď „Access-Accept“. Z toho vyplývá, že mé nastavení je funkční a testovací klient byl přihlášen.

V odpovědi lze najít oba mnou nastavované atributy. Pokud by přihlášení nedopadlo jak má, odpovědi by se daly najít jak ve výpise režimu „debug“, který by s největší pravděpodobností ohlásil chybu. Tak i v logu routeru MikroTik. Pro ověření svého přihlášení si můžu vypsat záznam tabulky radpostauth.

```
mysql> SELECT pass,reply,date FROM radpostauth WHERE user =
"00:13:02:BB:A7:23";
+---------------+--------------+---------------------+
| pass          | reply        | date                |
+---------------+--------------+---------------------+
| Chap-Password | Access-Accept | 2008-04-08 12:05:00 |
+---------------+--------------+---------------------+
```

Tato tabulka slouží k zaznamenávání informací o přihlášení. Doporučoval bych při neúspěšném přihlášení zkontrolovat její obsah.

Jednoduchou cestou si můžu též ověřit v routeru MikroTik, zda byl klient autorizován. Balík HotSpot obsahuje seznam klientů. V tomto seznamu lze ověřit podle příznaku klienta, zda byl autorizován, či nikoli.

```
[Peny@Peny] > /ip hotspot host print detail
Flags: S - static, H - DHCP, D - dynamic, A - authorized, P -
bypassed
 0 HA mac-address=00:13:02:BB:A7:23 address=10.255.200.134
      to-address=10.255.200.134 server=server1 uptime=13m26s
      keepalive-timeout=2m found-by="TCP :49958 -> 66.228.113.26:80"
```

Z výpisu, podle příznaku A je vidět, že klient byl vskutku autorizován. Druhý příznak H sděluje, že byla klientovi přidělena adresa IP serverem DHCP. Takto podobný příznak existuje i v seznamu zapůjčených adres serverem DHCP. Tentokrát jde o příznak R.

```
[Peny@Peny] > /ip dhcp-server lease print detail
Flags: X - disabled, R - radius, D - dynamic, B – blocked
16 R address=10.255.200.130 mac-address=00:13:02:BB:A7:23
     client-id="1:0:13:2:bb:a7:23" server=Wi-Fi status=bound
     expires-after=2d23h58m47s active-address=10.255.200.130
     active-mac-address=00:13:02:BB:A7:23
     active-client-id="1:0:13:2:bb:a7:23" active-server=Wi-Fi
     host-name="Peny-PC"
```

Z předchozích indicií vím, že se přihlášení povedlo. Tak si vyzkouším malý experiment. Přidám do tabulky radcheck záznam, který zapříčiní odepření ověření. Pod tímto záznamem mám na mysli Auth-Type := Reject. Klient bude odmítnut.

```
mysql> INSERT INTO radcheck VALUES (NULL,"00:13:02:BB:A7:23","Auth-
Type",":=","Reject");
Query OK, 1 row affected (0.03 sec)

mysql> SELECT * FROM radcheck WHERE UserName = "00:13:02:BB:A7:23";
+---------+-------------------+-----------+----+--------+
| id      | UserName          | Attribute | op | Value  |
+---------+-------------------+-----------+----+--------+
| 5500033 | 00:13:02:BB:A7:23 | Auth-Type | := | Reject |
+---------+-------------------+-----------+----+--------+
1 row in set (0.00 sec)
```

Při pokusu o připojení klienta, pro kterého jsem nastavil odmítnutí, se ukáže v režimu „debug“ serveru FreeRADIUS následující výpis:

```
Sending Access-Reject of id 0 to 10.255.200.254 port 44108
      Reply-Message := "Ahoj Test!"
```

Z výpisu je zřejmé, že klient byl odmítnut odpovědí Access-Reject. Pokud by se hodnota Reject změnila na Accept, bude klient přijat, ať už zadá jakékoliv heslo. Tento záznam odeberu, protože sloužil pouze jako příklad.

```
mysql> DELETE FROM radcheck WHERE Value = "Reject";
```

Tímto bych ukončil část věnovanou testování ověření klienta a budu pokračovat testováním RADIUS Accounting. Jen bych chtěl říci, že toto testování je přínosem při ladění použitých atributů, nebo při nasazení nových ověřovacích metod. Je možné si takto dokonce ověřit funkčnost dotazů SQL, popřípadě vyzkoušet funkci jejich modifikací.

2.8.2 RADIUS Accounting

Testování RADIUS Accounting probíhá téměř identicky jako ověřování. Je nutné nejdříve ukončit běh serveru FreeRADIUS obvyklým způsobem, nebo ukončit proces. Následovně spustit FreeRADIUS v režimu „debug“ (jako uživatel root):

```
/etc/init.d/freeradius stop
freeradius –X
```

V tomto režimu vypisuje FreeRADIUS veškeré informace o sobě, o svém běhu, ale i informace o požadavcích Accounting-Request. Nyní mám pro testování nastaven server HotSpot s ověřováním adres MAC.

Na zkoušku se zkusím přihlásit. Pokud vše funguje jak má, měl by se ukázat blok podobný tomuto:

```
rad_recv: Accounting-Request packet from host 10.255.200.254:57415,
id=226, length=141
      Acct-Status-Type = Start
      NAS-Port-Type = Wireless-802.11
      Calling-Station-Id = "00:13:02:BB:A7:23"
      Called-Station-Id = "server1"
      NAS-Port-Id = "AP"
      User-Name = "00:13:02:BB:A7:23"
      NAS-Port = 2150629393
      Acct-Session-Id = "80300011"
      Framed-IP-Address = 10.255.200.134
      Mikrotik-Host-IP = 10.255.200.134
      Event-Timestamp = "Apr  7 2008 13:07:02 CEST"
      NAS-Identifier = "Peny"
      NAS-IP-Address = 10.255.200.254
      Acct-Delay-Time = 0
```

Z požadavku lze vyčíst spoustu zajímavých informací. Všechny tyto informace odesílá server NAS pomocí následujících párů Attribute-Value serveru RADIUS (popis atributů viz příloha D).

Z výstupu routeru MikroTik je možné vyčíst, zda přišla odpověď ze strany serveru Accounting-Response.

```
Accounting-Response with id 15 from 82.202.96.34:1813
Signature = 0x6c77091cbc51f372f4720a0d4df4f64f
```

Následně uvádím ještě příklad požadavku Interim-Update, který je zachycen režimem „debug“ serveru FreeRADIUS.

```
rad_recv: Accounting-Request packet from host 10.255.200.254:36292,
id=186, length=183
      Acct-Status-Type = Interim-Update
      NAS-Port-Type = Wireless-802.11
      Calling-Station-Id = "00:13:02:BB:A7:23"
      Called-Station-Id = "server1"
      NAS-Port-Id = "AP"
      User-Name = "00:13:02:BB:A7:23"
      NAS-Port = 2150629393
      Acct-Session-Id = "80300011"
      Framed-IP-Address = 10.255.200.134
      Mikrotik-Host-IP = 10.255.200.134
      Event-Timestamp = "Apr  7 2008 13:43:03 CEST"
      Acct-Input-Octets = 58723
      Acct-Output-Octets = 470994
      Acct-Input-Gigawords = 0
      Acct-Output-Gigawords = 0
      Acct-Input-Packets = 609
      Acct-Output-Packets = 592
      Acct-Session-Time = 2161
      NAS-Identifier = "Peny"
      NAS-IP-Address = 10.255.200.254
      Acct-Delay-Time = 0
```

Tento typ požadavku slouží pro aktualizaci informací o uživateli. Proto se zde nacházejí další páry hodnot, jako jsou například příchozí/odchozí oktety dat, nebo počet odeslaných, přijatých paketů a délka trvání sezení. Tyto informace se následně ukládají do databáze.

Pomocí testovaní lze zjistit informace o komunikaci mezi klient/server RADIUS. Pokud by komunikace neprobíhala podle představ, pravděpodobně bude mít upovídaný režim serveru FreeRADIUS odpověď, proč tomu tak je.

Poslední test, který pro úplnost provedu, je testem obsahu databáze. Vzhledem k tomu, že upovídaný režim serveru FreeRADIUS neohlásil žádnou chybu či varování, měly by být v databázi uloženy informace o mém současném připojení. Tuto hypotézu si ověřím jednoduchým dotazem SQL:

```
mysql> SELECT AcctSessionId, UserName, AcctStartTime FROM radacct;
+---------------+-------------------+---------------------+
| AcctSessionId | UserName          | AcctStartTime       |
+---------------+-------------------+---------------------+
| 80300010      | 00:13:02:BB:A7:23 | 2008-04-07 11:18:46 |
| 80300011      | 00:13:02:BB:A7:23 | 2008-04-07 13:07:02 |
+---------------+-------------------+---------------------+
2 rows in set (0.00 sec)
```

Z výsledku dotazu je zřejmé, že druhý řádek odpovídá mému současnému připojení. *AcctSessionId*, *UserName* i *AcctStartTime* se shodují s požadavky zachycenými v režimu „debug“ serveru. Z toho vyplývá, že nastavení RADIUS Accountingu je správné.

3. Závěr

V této práci jsem popsal protokoly RADIUS, navrhl a nastavil jsem komunikaci mezi aplikací FreeRADIUS (server RADIUS) a routerem MikroTik (klient RADIUS) a na konec celé nastavení otestoval.

Popis protokolů nezabíhá do detailů. Zájemcům, kteří by chtěli protokoly studovat detailněji, třeba za účelem tvorby klienta RADIUS či serveru, doporučuji nastudovat dokumenty RFC. Rozsah použitých dokumentů RFC je mnohem vyšší, cca 200 stran.

Jak jsem uvedl, FreeRADIUS je velice mocný nástroj s velkou škálou možností autorizace, lze ho nakonfigurovat pro komunikaci s různými databázovými systémy a povoluje úpravu dalších režijních voleb (např. nastavení portu, aktivace spolupráce s protokolem SNMP a další). Jako databázový systém jsem zvolil MySQL, především z důvodu jeho možnosti clusterování. A také mimo jiné pro jeho obvyklé použití se skriptovacími jazyky *perl* a *php*. Jazyky *perl* a *php* jsou v dnešní době používány pro tvorbu dynamických webových stránek. Za jejich pomoci je možné naprogramovat webovou administrační stránku databáze serveru RADIUS a tím ulehčit práci administrátorům sítě. Popis mnou vytvořené administrační aplikace, jsem přiložil do přílohy E.

Clusterování databáze MySQL je velikým přínosem pro servery RADIUS, protože je možné postavit řadu serverů RADIUS nad replikovaným clusterem databáze. Tímto způsobem lze docílit dostačujícího stupně redundance.

Pokud bych měl shrnout informace o kapitolách nastavování a testování, rád bych ještě jednou zdůraznil, že se jedná pouze o příklady atributů. Firma MikroTik podporuje celou řadu dalších, svých vlastních atributů. Veškeré jejich atributy jsou popsány v dokumentaci operačního systému RouterOS. Některé tyto atributy jsou velice zajímavé. Protože každý výrobce definuje použití svých specifických atributů, je důležité zvolit správný typ přístupového serveru NAS.

Ze svých zkušeností vím, že testování serveru RADIUS v jeho režimu „debug“ je velice přínosné. Je to jeden ze způsobů, kterým je možno ladit kombinace atributů.

Když se zamyslím nad bezpečností protokolů RADIUS, jsem toho názoru, že tvůrci bezpečnost podcenili. Při útoku typu „Man In The Middle“, by mohl útočník odposlechnout velmi důležité informace. A podle mne existuje reálné nebezpečí podstrčení povržené odpovědi Access-Accept. Doporučuji vyzkoušet následovně:

- Fyzicky přemostit spoj mezi přístupovým serverem NAS a serverem RADIUS.
- Odchytit komunikaci (např. aplikací WireShark) mezi těmito zřízeními a následně ji analyzovat.
- Z analýzy zjistit řetězec sdíleného tajemství.
- Přerušit spojení mezi přístupovým serverem NAS a serverem RADIUS a nasadit vlastní podstrčený RADIUS server.

Zajímavým řešením otázky bezpečnosti by bylo použití šifrování paketů, či dokonce veškeré komunikace pomocí známých způsobů. Např. prostřednictvím IPSec [11] (použití šifrované linky mezi přístupovým serverem NAS a serverem RADIUS). Routery MikroTik protokol síťové vrstvy modelu ISO/OSI [12] IPSec umožňují. Implementace protokolu IPSec u routerů firmy MikroTik ale zatím není natolik stabilní, aby se dala prakticky využít při komunikaci s centrálním autorizačním systémem (Stává se, že někdy kvůli špatnému nastavení času, spojení protokolu IPSec u routerů firmy MikroTik kolabuje). Snad v novějších verzích operačního systému RouterOS bude implementace protokolu IPSec opravená.

Jsem toho názoru, že i přes veškerá rizika se zabezpečením a redundancí je nasazení server/klient RADIUS pro centralizované ověřování klientů sítě velikým přínosem a ulehčením práce administrátorů. Při řádném návrhu sítě je možné rizika využití protokolu RADIUS téměř eliminovat.

Díky této práci jsem se o protokolech RADIUS dozvěděl mnoho informací. Přezkoumání jejich principu a atributů bylo pro mne velice zajímavé a přínosné. Na základě nových znalostí jsem mohl navrhnout efektivní nasazení serveru RADIUS pro komunikaci s routerem MikroTik. Správnost mého řešení jsem otestoval a zjistil, že komunikace probíhá tak, jak jsem ji navrhl.

Jako jediné dvě problematické věci, na které jsem při práci narazil, byla volba správné kombinace atributů v tabulkách RHS (viz strana 47). Druhý problém spočíval ve volbě adresy IP přístupového serveru NAS v tabulce databáze serveru RADIUS. O tomto problému pojednávám na straně 44.

Rozhodně by bylo zajímavé vyzkoušet mnou navrhované řešení bezpečnější komunikace mezi serverem a klientem RADIUS.

Má práce by mohla být zajímavá pro všechny poskytovatele bezdrátových sítí. Přesto, že mnou navržené řešení nelze aplikovat na všechny sítě, popis a teorie protokolů bude vždy stejná.

Seznam použité literatury

[1] Libor Dostálek a kol.: *Velký průvodce protokoly TCP/IP: Bezpečnost.* Computer Press, 2003. ISBN: 80-7226-849-X

[2] *MikroTik: RADIUS client* [online]. MikroTik, 2006-07-03 [cit. 2006-11-14]. URL: http://www.mikrotik.com/docs/ros/2.9/guide/aaa_radius, http://www.mikrotik.com/docs/ros/2.9/guide/aaa_radius.pdf

[3] Rigney, C. a kol.: *RFC 2138 – Remote Authentication Dial In User Service (RADIUS)* [online]. 1997 [cit. 2006-11-14]. URL: http://www.faqs.org/rfcs/rfc2138.html

[4] Rigney, C. a kol.: *RFC 2139 – RADIUS Accounting* [online]. 1997 [cit. 2006-11-14]. URL: http://www.faqs.org/rfcs/rfc2139.html

[5] Chiba, M. a kol.: *RFC 3576 – Dynamic Authorization Extensions to Remote Authentication Dial In User Service (RADIUS)* [online]. 2003 [cit. 2006-11-14]. URL: http://www.ietf.org/rfc/rfc3576.txt

[6] *The FreeRADIUS Server Project* [online]. 2004 [cit. 2006-11-14]. URL: http://www.freeradius.org/

[7] Bartlett, S.: SB's *very rough notes to FreeRadius and MySQL* [online]. 2005-03-20 [cit. 2006-11-14]. URL: http://www.frontios.com/freeradius.html

[8] Mikolášek, V.: *Napojení RADIUS serveru na SQL* [online]. 2004 [cit. 2006-11-14]. URL: http://www.kiv.zcu.cz/~simekm/vyuka/pd/zapocty-2004/radius_mysql-mikolasek/

[9] Rigney, C. a kol.: RFC *2138 – Remote Authentication Dial In User Service (RADIUS)* [online]. 1997 [cit. 2006-11-14]. URL: http://www.scit.wlv.ac.uk/rfc/rfc21xx/RFC2138.html

[10] Simpson, W. a kol.: *RFC 1994 - PPP Challenge Handshake Authentication Protocol (CHAP)* [online]. 1996 [cit. 2006-11-14]. URL: http://www.faqs.org/rfcs/rfc1994.html

[11] *IPsec* [online]. [cit. 2006-11-14]. URL: http://cs.wikipedia.org/wiki/IPsec

[12] *Referenční model ISO/OSI* [online]. [cit. 2006-11-14]. URL: http://cs.wikipedia.org/wiki/Referen%C4%8Dn%C3%AD_model_ISO/OSI

[13] Case, J. a kol.: *RFC 1157 - Simple Network Management Protocol (SNMP)* [online]. 1990 [cit. 2006-11-14]. URL: http://www.faqs.org/rfcs/rfc1157.html

Seznam obrázků a tabulek

Příloha A: Popis dotazů do MySQL

Autorizaci zde provádějí 4 dotazy. První dva z nich patří do skupiny porovnání LHS. Druhý pár dotazů vrací atributy a hodnoty RHS:

- authorize_check_query – provádí individuální autorizaci,
- authorize_group_check_query – skupinovou autorizaci,
- authorize_reply_query – vrací individuální atributy,
- authorize_group_reply_query – skupinové atributy.

Ostatní dotazy patří spíše do kategorie informativních. Jejich význam je následující:

- group_membership_query – ověřuje, do které skupiny patří klient, popřípadě UserName.
- postauth_query – vkládá informace o úspěšném ověření klienta.

Pro RADIUS Accounting k zaznamenávání slouží:

- simul_verify_query,
- accounting_onoff_query,
- accounting_update_ query,
- accounting_update_query_alt,
- accounting_start_query,
- accounting_start_query_alt,
- accounting_stop_query,
- accounting_stop_query_alt.

Z komentářů souboru `sql.conf` lze vyčíst, že dotaz *simul_verify_query* vrací informace o současném připojení. Následujících sedm dotazů do databáze MySQL upravují tabulku *radacct* podle typu paketů (start, stop, update, on/off). Tři z uvedených dotazů mají v názvu přídavek _alt. Dotazy s tímto přídavkem jsou alternativními dotazy pro případ, že primární dotaz do databáze selže.

Příloha B: Popis tabulek MySQL

Po zadání dotazu v MySQL „*show tables*“ se zobrazí seznam 8 tabulek:

- Nas,
- radacct,
- radcheck,
- radgroupcheck,
- radgroupreply,
- radpostauth,
- radreply ,
- usergroup.

Mezi LHS patří tabulky radcheck a radgroupcheck a do skupiny RHS radreply a radgroupreply. Tyto čtyři tabulky budou společně popsány jako první, protože jejich struktura je téměř identická. Následující dotaz SQL vypisuje tabulku *radreply*.

```
mysql> SELECT * FROM radreply;
```

Tabulky mají pět sloupců:

- id – tento sloupec je víceméně pouze informativní. FreeRADIUS ho využívá pouze k řazení výsledku dotazů.
- UserName / GroupName – přeloženo uživatelské / skupinové jméno.
- Attribute – velice důležitý sloupec. Zde se vkládají atributy. Například v tabulkách LHS user-password.
- op – neboli operátor. Jako operátory lze např. použít: ==, :=, =.
- Value – posledním sloupcem tabulek je Value, hodnota atributu. Při použití autorizace pomocí adres MAC bývá u LHS hodnota prázdná, v jiném případě obsahuje například heslo.

Další z velmi důležitých tabulek je tabulka *nas*. Tabulka *nas* uchovává v sobě informace o serverech NAS (Network Access Server), jimiž jsou v našem případě routery firmy MikroTik. Použití této tabulky má malý háček. Problém tkví v tom, že je potřeba použití tabulky *nas* nejdříve aktivovat. Pro aktivaci je nutné zapnout parametr „readclients = yes“ v souboru `sql.conf`. Tento parametr se obvykle nachází až na konci souboru a je okomentován. Tabula *nas* lze vypsat dotazem SQL takto:

```
mysql> SELECT * FROM nas;
```

Tabulka nas obsahuje osm sloupců. Jejich význam je následující:

- id – nemá nějaký obzvlášť velký význam. Slouží pouze k řazení dotazů.
- nasname – do tohoto sloupce se vkládají adresy IP serveru NAS.
- Shortname – položka uchovává informace názvu serveru NAS.
- type - Nesmíme zapomenout zvolit správný typ NAS. Type říká serveru RADIUS, že klient může obsahovat specifické atributy, které jsou právě závislé na typu NASu.
- ports – sloupec ports udává počet dostupných portů NAS. Toto pole je informativní a slouží například programu dialupadmin, grafickému rozhraní pro správu databáze. Port jinak posílá sám NAS v rámci paketu Acces-Request. Pro využití čísla portu zaslané pomocí paketu Acces-Request, lze pole nastavit na hodnotu NULL.
- secret – položka uchovává informace o sdíleném tajemství mezi klient/server RADIUS.
- community – udává název komunity SNMP. V případě, že SNMP není použito, je pole nastaveno na NULL.
- description – poslední prvek, česky popis. Jde pouze o informativní prvek.

Poslední tabulkou důležitou pro autorizaci klienta, respektive klientských skupin, je *usergroup*. Tato tabulka má jednoduchý účel. Přiřazuje skupiny uživateli. Stejně jako má tabulka jednoduchý význam, má i jednoduchou strukturu. Následující dotaz SQL vypíše tabulku *usergroup*.

```
mysql> SELECT * FROM usergroup;
+-------------------+-----------+----------+
| UserName          | GroupName | priority |
+-------------------+-----------+----------+
| Peny              | HotSpot   |        2 |
| 00:60:B3:8C:8E:03 | Wi-Fi     |        1 |
+-------------------+-----------+----------+
```

Obsahuje pouze tři sloupce:

- UserName – jak už vyplývá z názvu, určuje uživatelské jméno.
- GroupName – přiřazuje skupinu uživateli.
- Priority – nastavuje prioritu skupin uživateli. To znamená, že pokud uživatel patří do více skupin, platňují se jako první pravidla skupiny s nejnižší prioritou.

Předposlední tabulkou je tabulka *radpostauth*. Tato tabulka má pouze informativní účel. Zapisují se zde informace o autorizaci. Obsah tabulky lze vypsat následovně:

```
mysql> SELECT * FROM radpostauth;
```

Tabulka se skládá z pěti sloupců:

- id – je pouze informativního účelu,
- user – do tohoto sloupce se ukládá uživatelské jméno,
- pass – sloupec informuje o typu přihlášení,
- reply – do sloupce reply server RADIUS zapisuje typ paketu přihlášení, takže třeba Access-Accept,
- date – datum a hodina přihlášení.

Poslední tabulka v databázi, neboli *radacct***,** je tabulka určená pro RADIUS Accounting. Taktéž se jedná o informativní tabulku vyplňovanou serverem RADIUS Accounting. Její struktura je nejsložitější ze všech ostatních tabulek. Je zajímavá kvůli studiu zpracování logů RADIUS Accounting. V routeru MikroTik je zapotřebí její využití nejdříve aktivovat. Tímto se budu zabývat až v sekci nastavování RADIUS Accountingu na routeru MikroTik.

Tabulka radacct obsahuje 25 sloupců. Jejich význam je následující:

- RadAcctId – udává identifikační číslo položky v tabulce,
- AcctSessionId – zde se zapisuji identifikační čísla sezení, přijaté od požadavku Accounting-Request,
- AcctUniqueId – obsahuje unikátní identifikační číslo,
- UserName – obsahuje uživatelské jméno klienta,
- Realm – Realm doména klienta,
- NASIPAddress – adresa IP serveru NAS, z něhož přišel požadavek,
- NASPortId – identifikace portu NAS,
- NASPortType – typ portu NAS. Např. Wireless-802.11,
- AcctStartTime – čas příchodu požadavku Start,
- AcctStopTime – čas příchodu požadavku Stop,
- AcctSessionTime – doba trvání sezení,
- AcctAuthentic – zaznamenání výběru autorizace RADIUS nebo Local authority (pouze služba PPP),
- ConnectInfo_start – u této položky není zatím definované využití,
- ConnectInfo_stop – u této položky není zatím definované využití,
- AcctInputOctets – počet příchozích oktetů,
- AcctOutputOctets – počet odchozích oktetů,
- CalledStationId – identifikace služby odesílající požadavek,
- CallingStationId – v tomto poli se nachází identifikace klienta,
- AcctTerminateCause – obsahuje důvod ukončení sezení,
- ServiceType – u routeru MikroTik se používá jedině „Framed“ (pouze služba PPP),
- FramedProtocol – protokol linkové vrstvy. U routeru MikroTik se používá jedině „PPP“ (pouze služba PPP),
- FramedIPAddress – přidělená adresa IP klientovi,
- AcctStartDelay – doba po jakou se klient snažil odeslat požadavek Accounting-Request typu Start,
- AcctStopDelay – doba po jakou se klient snažil odeslat požadavek Accounting-Request typu Stop,
- XAscendSessionSvrKey – u této položky není zatím definované využití.

Příloha C: Volby přidání propojení k serveru RADIUS

Volby přidání nového propojení (hodnoty v závorce říkají, jaký datový typ volby je použit a jaká je jeho výchozí hodnota, v případě, že je použit znak „|“ jedná se o přepínač a okolo tohoto znaku se nachází volby přepínače):

```
[Peny@IvDoma] > /radius add ?
creates new item with specified property values.
```

- accounting-backup (yes | no; default: no) – pokud *yes*, pak jde o záznam záložního serveru RADIUS pro Accounting
- accounting-port (integer; default: 1813) – server RADIUS Accounting port,
- address (IP address; default: 0.0.0.0) - adresa IP serveru RADIUS,
- authentication-port (integer; default: 1812) – server RADIUS port autentifikace,
- called-id (text; default: "") – hodnota závislá na PPP:
    - ISDN – telefonní číslo vytáčení (MSN),
    - PPPoE – jméno služby,
    - PPTP – IP Adresa serveru,
    - L2TP – IP Adresa serveru.
- domain (text; default: "") – doména Microsoft Windows pro klienty serveru RADIUS, který potřebuje ověření domény Windows,
- realm (text) – explicitně určený realm (uživatelská doména), uživatel nemusí mít doménu ISP v uživatelském jméně,
- secret (text; default: "") – sdílené tajemství pro přístup k serveru RADIUS,
- service (multiple choice: HotSpot | login | ppp | telephony | wireless | dhcp; default: "") – služby routeru, které využívají služeb serveru RADIUS:
    - HotSpot – autorizační služba HotSpot,

        - login – lokální přihlašování do routeru,
        - ppp – autorizace klientů Point-to-Point,
        - telephony – IP telephony accounting,
        - wireless – autorizace bezdrátových klientů (klientská adresa MAC slouží jako User-Name),
        - dhcp – klient DHCP autorizace (klientská adresa MAC slouží jako User-Name).
- timeout (time; default: 100ms) – timeout, po kterém je požadavek znovu odeslán.

Příloha D: Popis atributů přijatých při testování

Požadavek Access-Request:

- NAS-Port-Type = Wireless-802.11 – informace o typu portu NASu,
- Calling-Station-Id = "00:13:02:BB:A7:23" – id klienta, jeho adresa MAC,
- Called-Station-Id = "server1" – identifikace volající služby,
- NAS-Port-Id = "AP" – identifikace volajícího portu (použit název interface),
- User-Name = "00:13:02:BB:A7:23" – uživatelské jméno klienta,
- NAS-Port = 2150629399 – číslo portu, ke kterému se uživatel hlásí,
- Acct-Session-Id = "80300017" – identifikační číslo sezení,
- Framed-IP-Address = 10.255.200.134 – požadovaná adresa IP pro klienta,
- Mikrotik-Host-IP = 10.255.200.134 – původní adresa IP klienta,
- User-Password = "" – heslo klienta,
- Service-Type = Login-User – typ požadavku,
- WISPr-Logoff-URL = http://10.255.200.129/logout – odhlašovací stránka balíku HotSpot,
- NAS-Identifier = "Peny" – identifikace serveru NAS,
- NAS-IP-Address = 10.255.200.254 – adresa IP serveru NAS.

Požadavek Accounting-Request :

- Acct-Status-Type = Start – informuje o jaký požadavek jde. Tento typ určuje, prvotní paket, který začíná Accounting sezení. Dalšími typy jsou Interim-Update a Stop.
- NAS-Port-Type = Wireless-802.11 – informuje o typu portu,
- Called-Station-Id = "server1" – název serveru HotSpot,
- User-Name = "00:13:02:BB:A7:23" – uživatelské jméno,

- Event-Timestamp = "Apr 7 2008 13:07:02 CEST" – časová známka události,
- NAS-Identifier = "Peny" – identifikace serveru NAS,
- NAS-IP-Address = 10.255.200.254 – taktéž i jeho adresa IP.

Z požadavku Accounting-Request lze vyčíst více informací jako například adresu IP klienta, identifikaci klienta, identifikační číslo sezení a mnoho dalších.

Příloha E: Administrační aplikace

Jako přílohu k mé bakalářské práci jsem naprogramoval administrační aplikaci pro přidávání, editaci a mazání záznamů v databázi serveru RADIUS za účelem ulehčení správy klientů v sítích Wi-Fi. Vytvořil jsem zde dvě možnosti editace. Přidávání klientů sítě a přidání přístupových serverů NAS. Pro každou z nich je určen samostatný oddíl s formulářem. Vzhledem ke snaze o zjednodušení přidávání, editace a mazání záznamů v databázi jsem formulář navrhl tak, že se zde zadává pouze ta nejpodstatnější informace jak pro klienty (adresa MAC), tak pro přístupový server NAS (adresa IP). Ostatní důležité informace jsou definovány v konfiguračním souboru práce s databází (např.: sdílené tajemství, uživatelská skupina a jiné).

Administrační aplikace využívá nastavení popsané ve vlastní práci (ověření na principu adres MAC). Jak jsem už v ní uvedl, nastavení vychází z konfigurace přístupových serverů NAS a serveru RADIUS, které využívám. Proto je tato aplikace určená pro nasazení k administraci mnou zvoleného řešení.

Aplikace je napsána v jazyce *php* a je určena pro nasazení na webový server s podporou zmiňovaného jazyka. Jako programovací styl jsem využil objektově orientované programování. Jazyk *php* disponuje možností objektově orientovaného programování od verze 5. Při programování jsem narazil na trochu zvláštní chování instancí jazyka *php* (Jazyk disponuje zvláštní implementací konstruktoru i destruktoru. Instanci je téměř nemožné zrušit a parametry třídy se chovají jako globální proměnné nejen pro danou instanci). Vytvořená aplikace pracuje s databázovým systémem MySQL. Je proto nezbytné mít nainstalované rozšíření jazyka *php* podporující komunikaci s tímto systémem. Konfigurace připojení k MySQL je v programu umístěna do konfiguračního souboru `mysql.conf.php`.

Pro práci s databází jsem navrhl třídu `TMySQL` umístěnou v souboru `mysql.init.php`. Veškerá komunikace se systémem MySQL je umístěna právě zde (připojení, odpojení, odeslání dotazu SQL, vrácení výsledku dotazu).

Třída `TDatabase` definovaná v souboru `database.init.php` generuje dotazy SQL podle mnou vytvořené datové struktury (jedná se o vícerozměrné pole založené na principu klíč – hodnota). Z této třídy je volána předchozí třída `TMySQL`. Třída `TDatabase` dokáže generovat nejběžnější SQL dotazy.

Třída `TDBOperation` náležící souboru `dboperate.init.php`. Definuje různé druhy konkrétních operací (např.: vkládání, mazání, vypisování, atd.). Třída pro zpracování volá třídu `TDatabase` pro vygenerování dotazu SQL.

Třídy `TClientOperate` (soubor `client.ini.php`) a `TNASOperate` (soubor `nas.init.php`) definují konkrétní operace pro skupinu (klienti nebo NAS). Mezi takové operace patří třeba výpis klientů, mazaní přístupových serverů NAS a další. Obě třídy volají třídu `TDatabase`.

Třídy `TDynamicHTML` a `THTML` a `header` slouží k vygenerování validního kódu XHTML, který představuje webové rozhraní aplikace. První z nich generuje kód XHTML pomocí dat získaných z databáze. Třída `THTML` obsahuje definice kódu XHTML. Důvodem byla snaha oddělit XHTML od jazyka php. Poslední zmiňovaná třída obsahuje definici hlavičky dokumentu XHTML.

Vzhledem k tomuto mému rozvržení je možné snadno upravit editační formuláře bez nutnosti změny tříd pracujících s databázovým systémem MySQL. Díky této vlastnosti je možné jádro pracující se systémem MySQL použít i v jiné aplikaci.