UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

Využití protokolu 802.1x pro ISP

Václav Jelínek

Bakalářská práce
2014

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: **2013/2014**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Václav Jelínek**
Osobní číslo:	**I11087**
Studijní program:	**B2646 Informační technologie**
Studijní obor:	**Informační technologie**
Název tématu:	**Využití protokolu 802.1x pro ISP**
Zadávající katedra:	**Katedra informačních technologií**

Zásady pro vypracování:

Cílem práce je navrhnout využití protokolu 802.1x a jeho využití na směrovačích Mikrotik pro potřeby ISP. Autor podrobně představí protokol 802.1x, jeho principy a možnosti nasazení. Dále autor představí technologie směrovačů Mikrotik pro specifické potřeby ISP. Autor navrhne nasazení vhodných směrovačů Mikrotik pro potřeby ISP včetně komplexního zabezpečení. Navržené řešení realizuje v laboratorním prostředí, kde prověří navržená bezpečnostní opatření.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

IER, James T a Neil ANDERSON. Implementing 802.1X security solutions for wired and wireless networks: an illustrated home networking handbook for the everyday user. Hoboken, N.J.: Wiley, c2008, xxiii, 330 p. ISBN 04-701-6860-9. HECKMANN, Oliver a Neil ANDERSON. The competitive Internet service provider: network architecture, interconnection, traffic engineering and network design. Hoboken, NJ: J. Wiley, c2006, xxvii, 370 p. ISBN 978-047-0012-932.

Vedoucí bakalářské práce: **Mgr. Josef Horálek**
Katedra softwarových technologií

Datum zadání bakalářské práce: **20. prosince 2013**
Termín odevzdání bakalářské práce: **9. května 2014**

Univerzita Pardubice Česká republika

L.S.

prof. Ing. Simeon Karamazov, Dr.
děkan

Ing. Lukáš Čegan, Ph.D.
vedoucí katedry

V Pardubicích dne 31. března 2014

Prohlášení autora

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše. Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 5. 5. 2014

Václav Jelínek

Poděkování

Rád bych poděkoval Mgr. Josefu Horálkovi, Ph.D. za cenné rady a informace při vedení mé práce.

Anotace

Cílem práce je navrhnout využití protokolu 802.1x a jeho využití na směrovačích Mikrotik pro potřeby ISP. Autor podrobně představí protokol 802.1x, jeho principy a možnosti nasazení. Dále autor představí technologie směrovačů Mikrotik pro specifické potřeby ISP. Autor navrhne nasazení vhodných směrovačů Mikrotik pro potřeby ISP včetně komplexního zabezpečení. Navržené řešení realizuje v laboratorním prostředí, kde prověří navržená bezpečnostní opatření.

Klíčová slova

IEEE, EAP, 802.1X, FreeRADIUS, RADIUS server, Mikrotik, RouterOS, autentizace, Wireless, MySQL.

Title

Use of protocol 802.1x for ISP.

Annotation

The goal of this thesis is to project utilization of protocol 802.1x and its application in Mikrotik routers for purposes of ISP. Author will describe protocol 802.1x in detail, describe its principles and possibilities of its use. Author will also describe technologies of Mikrotik routers for specific needs of ISP. Author projects utilization of suitable Mikrotik routers including complete security for needs of ISP. He will implement projected solution in laboratory environment and verify its security.

Keywords

IEEE, EAP, 802.1X, FreeRADIUS, RADIUS server, Mikrotik, RouterOS, authentication, Wireless, MySQL.

Obsah

Seznam zkratek

AAA	Authentication, Authorization, Acounting
CCP	Compression Control Protocol
DHCP	Dynamic Host Configuration Protocol
EAP	Extensible Authentication Protocol
EAPoL	EAP over LAN
ECP	Encryption Control Protocol
CHAP	Challenge Authentication Protocol
IEEE	Institute of Electrical and Electronics Engineers
ISP	Internet service provider
LAN	Local Area Network
LCP	Link Control Protocol
MAC	Media Access Control
MD5	Message-Digest algorithm
PAP	Password Authentication Protocol
PPP	Point-to-Point Protocol
RADIUS	Remote Authentication Dial In User Service
SSID	Service Set Identifier
TLS	Transport Layer Security
TTLS	Tunneled Transport Layer Security
WLAN	Wireless Local Area Network

Seznam obrázků

Seznam tabulek

Úvod

Cílem této bakalářské práce je seznámení s IEEE 802.1x, jeho principy a možností nasazení na zařízení Mikrotik.

První část bakalářské práce popisuje jednotlivé protokoly. Pro pochopení samotného protokolu 802.1x je nutné se seznámit s protokolem PPP, EAP a RADIUS. Popis těchto protokolů je hlavní náplní teoretické části. Součástí je i informace o ISP a technologii směrovačů Mikrotik.

V praktické části navrhuji využití protokolu 802.1x na směrovačích Mikrotik. Pro tento účel používám server FreeRADIUS a databázový systém MySQL na zařízení Raspberry PI jako autentizační server RADIUS. V poslední kapitole praktické části je nastavení otestováno.

1 Protokol 802.1x

Informační technologie je rychle se rozvíjející obor. Zasahuje do všech oblastí života. Ať už se jedná o využívání služeb internetu, komunikace přes počítačové sítě nebo návštěvy prostor budov veřejného a soukromého sektoru. S rozvojem technologií a jejich využívání rostou i nároky na samotnou ochranu informací. Bezpečná manipulace s informacemi se proto stává důležitou součástí celkových bezpečnostních strategií firem a společností. Téma, které bude dále popisováno, vychází z prostudovaných zdrojů (Zandl, 2003; Dostálek, 2003; Pužmanová, 2004; *802.1X-2010: IEEE standard*, 2010).

Zabezpečení počítačových sítí je velice rozsáhlá problematika, která v sobě zahrnuje celou řadu relativně samostatných okruhů. Patří sem například problematika šifrování dat, topologií sítí, firewallů, přístupu k sítí a celková bezpečnostní politika organizací a firem. V této práci se budu věnovat pouze zabezpečení a řízení přístupu k počítačové síti.

Domnívám se, že řízení přístupu k počítačovým sítím a službám je nezbytné pro správnou bezpečnostní politiku. Zvláště v případech, kdy nemáme pod kontrolou všechny fyzické a logické porty sítě, je nutné zamezit přístup nepovoleným osobám. Znemožňujeme tak přístup neautorizovaných osob bez toho, abychom museli fyzicky ověřovat bezpečnost všech LAN portů sítě. V případě bezdrátových sítí WLAN je to nemožné a je potřeba tento problém řešit. Řízení přístupu zabezpečí a znemožní přístup těm, co by mohli počítačovou síť zneužít nebo jinak poškodit. Nezabezpečená síť v dnešní době představuje otevřenou cestu pro útočníky, kteří mohou nepozorovaně zneužívat datové připojení, vydávat se za někoho jiného, odposlouchávat provoz nebo provádět další nelegální činnosti, jako odcizení důvěrných nebo citlivých informací. Existuje mnoho technologií, jak tuto problematiku řešit, například standart IEEE 802.1x, IEEE 802.11i.

Protokol 802.1x si můžeme představit jako vrátného, který brání a kontroluje vstup do budovy. Abychom mohli vstoupit, potřebujeme se nějakým způsobem legitimovat, například ID kartou. V případě protokolu 802.1x to funguje podobně. Pro vstup do sítě, musíme ověřit naši totožnost a to pomocí kombinace jména a hesla nebo pomocí jiného mechanismu. Protokol 802.1x je velmi silný mechanismus, ale nesmíme zapomínat, že řeší pouze přístup a neřeší komplexní zabezpečení. Pokud útočník prolomí přístupové zabezpečení, nic mu nebrání v jakékoliv činnosti na síti.

Významným přínosem využívání protokolu 802.1x je i centralizovaná správa a účtování. V případě decentralizované zprávy je velice obtížně nastavení bezpečnostních pravidel. Na

každém AP nebo switch je třeba individuálně nastavit, jaká bezpečnostní pravidla má používat. Při větším počtu AP nebo switchu hrozí riziko nestejně definovaných bezpečnostních pravidel, které může vést k případnému přístupu neoprávněných uživatelů a ohrožení celé sítě.

1.1 Standart IEEE 802.1x

Standard IEEE (Institute of Electrical and Electronics Engineers) [1] 802.1x definuje protokol pro řízení síťového přístupu na linkové vrstvě modelu ISO/OSI jako součást bezpečnostní architektury AAA (Authentication, Authorization, Acounting), který se vztahuje nejen na pevné, ale i na bezdrátové sítě. Zajišťuje transportní mechanismus pro ověřování. Vychází z protokolu PPP (Point-to-Point Protocol), který je používán pro spojení ve WAN sítí. Standart 802.1x funguje na třech komponentech – žadatel (klient), autentizátor (switch, AP) a autentizační server RADIUS.(*802.1X-2010: IEEE standard*, 2010)

Klient (Supplicant):

Je zařízení, které žádá o přístup do sítě a musí být ověřena jeho identita. Aby mohl klient žádat o přístup pomocí 802.1x a EAP (Extensible Authentication Protocol), musí zařízení disponovat 802.1x klientským softwarem. Ten je dnes běžně implementovaný v systémech Windows, v případě Linuxu zajišťuje funkci například program Xsupplicant. (*Cisco*, 2012; *Eduroam*, 2011)

Autentizátor (Authenticator):

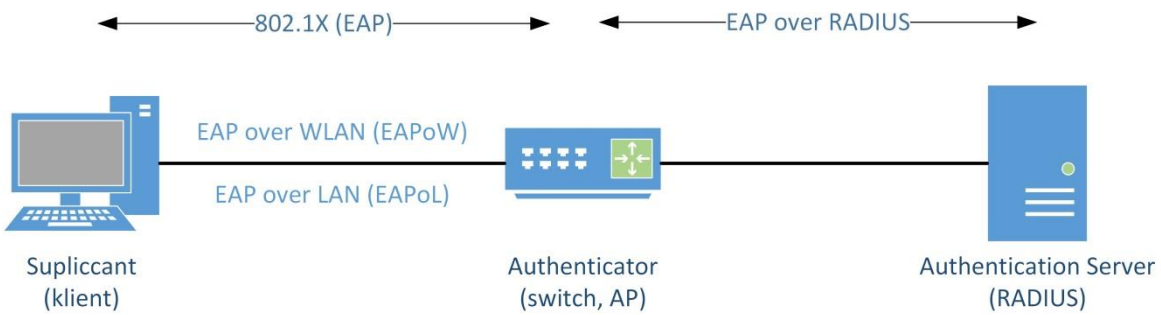
Zařízení funguje jako prostředník mezi klientem a ověřovacím serverem, který kontroluje fyzický přístup k síti na základě stavu autentizace klienta. Autentizátor obsahuje RADIUS klienta, který je zodpovědný za zapouzdření a vybalení EAP rámců a interakci s RADIUS serverem. Může se jednat o AP, router nebo switch. (*Cisco*, 2012)

[1] „sdružuje přes 350 000 elektroinženýrů a informatiků v cca 150 zemích ve všech světadílech“. *Československá sekce IEEE* [online]. 2013 [cit. 2014-03-9]. Dostupný z WWW: <http://www.ieee.cz/>.

Autentizační server (Authentication server):

Zařízení, které provádí autentizaci připojeného klienta. Autentizační server ověří identitu klienta a oznámí autentizátorovi, zda je nebo není klient oprávněn přistupovat ke službám sítě. Nejčastěji se využívá bezpečnostní systém RADIUS. RADIUS pracuje v režimu klient/server. Nejen že umí autentizaci uživatelů, ale je schopen uživatele přiřadit i do určité VLAN(virtual local area network) s omezeným přístupem ke službám. (*Cisco*, 2012)

1.2 Model protokolu 802.1x



Obrázek 1 Model protokolu 802.1x

Jedná se o autentizaci na úrovni portu switche LAN (Port-based Network Access Control) nebo virtuálního portu přístupového bodu WLAN. Při komunikaci mezi klientem a autentizátorem jsou EAP zprávy zapouzdřeny a přenášeny pomocí ethernetových rámců, ty jsou v literatuře (Zandl, 2003, s. 134; Pužmanová, 2004, s. 229) označovány jako EAPoL (EAP over LAN) nebo pro IEEE 802.11 jako EAPoW (EAP over WLAN). Mezi autentizátorem a autentizačním serverem jsou EAP zprávy zapouzdřeny a přenášeny prostřednictvím zpráv protokolu RADIUS.

1.3 PPP

PPP (Point-to-Point Protocol) je protokol linkové vrstvy. Používá se pro spojení mezi dvěma uzly. Díky jeho podpoře pro synchronní i asynchronní přenos a ověřování CHAP a PAP se stal standardem pro komunikaci na sériových linkách. Je také používán pro WAN spojení. Někteří ISP (Internet service provider) [2]jej využívají ve formě PPPoE (PPP over Ethernet) pro ověření uživatelů připojených přes xDSL. Umožňuje autentizaci, šifrování pomocí ECP (Encryption Control Protocol), kompresi dat pomocí CCP (Compression Control Protocol) a testuje kvalitu spojení. Kompletní popis lze najít v dokumentu RFC1661 (Simpson, 1994).

PPP ve skutečnosti není jediný protokol, ale skládá se hned ze dvou úrovní. Z protokolu pro řízení spojení LCP (Link Control Protocol) a protokolu řízení sítě NCP (Network Control Protocol).

- LCP – Tento protokol slouží pro navázání spojení, dohaduje konfiguraci a testuje spojení. Probíhá ve čtyřech fázích, z nichž jsou pouze dvě povinné: navázání a udržování spojení. Další dvě fáze, autentizace a zjištění kvality spoje, jsou volitelné a závisí na konkrétní implementaci.
- NCP – Slouží jako podpora pro jednotlivé protokoly vyšší síťové vrstvy (např. pro zapouzdření, adresaci apod.). Může sloužit pro každou síťovou architekturu (TCP/IP, AppleTalk, NetWare).

Při vytváření spojení na lince point-to-point nejprve PPP protokol zašle několik LCP paketů, které nastaví a otestují komunikaci. Dále PPP pokračuje posíláním NCP paketů, které vyberou a nastaví jeden či více protokolů síťové vrstvy. Po dokončení této konfigurace může začít komunikace na síťové vrstvě.

Ukončení spojení lze provést zasláním speciálních LCP nebo NCP paketů. K ukončení spojení může dojít také při různých událostech (vypršení časové odezvy).

[2] Poskytoval internetového připojení

Obrázek 2 Rámec protokolu PPP

Křídlová značka (8 bitů) – značí začátek a konec PPP rámec. Obsahuje bin. 01111110 (0x7E). Pokud je znak 7E přenášen v datech, je u synchronních linek použit Bit-stuffing a u asynchronních espace sekvence.

Adresa (8 bitů) – obsahuje vždy hodnotu 11111111 (0xFF) broadcast. Důsledkem je, že nemůžeme určit příjemce paketu.

Řídící pole (8 bitů) – obsahuje hodnotu 000000011 (0x03). Pokud se na lince vyskytují rámce pouze s těmito adresami a řídícími poli, pak se oba konce linky mohou dohodnout pomocí LCP na použití komprese. Při vysílání se tato pole vypustí a při příjmu doplní.

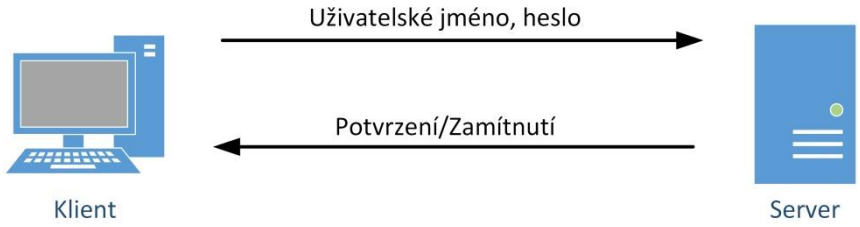
Protokol (8 / 16 bitů dle ISO 3309) – identifikuje typ protokolu zapouzdřených v rámcích.

Data (0 nebo více byte) – obsahuje samotnou přenášenou informaci.

Kontrolní součet (16 nebo 32 bitů) – zajišťuje detekci chyb dle předpisu HDCL.

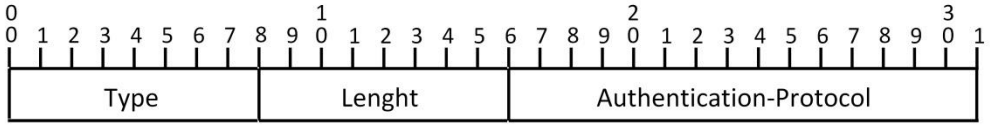
1.3.1 PAP

Nejjednodušší způsob autentizace v PPP nabízí protokol PAP (Password Authentication Protocol). Používá jednoduché, jednorázové ověření s přenosem hesla v otevřené formě po síti. Patří mezi nejstarší autentizační protokoly, se kterými se dnes můžeme setkat. Autentizaci inicializuje vždy klient na počátku spojení v autentizační zprávě, kde posílá jméno a heslo. Nevýhodou je posílání hesla po síti bez šifrování. Je popsán v RFC 1334 (Lloyd a Simpson, 1993).



Obrázek 3 Autentizační protokol PAP

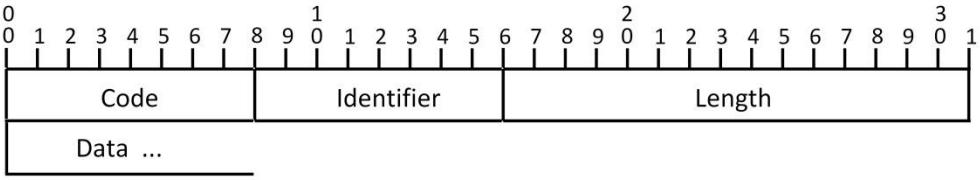
Protokol PAP využívá k autentizaci dvoucestnou výměnu (2-way handshake). Klient posílá serveru svoje ID a heslo. To opakuje, dokud server nepotvrdí správnou autentifikaci nebo informuje o chybné autentifikaci a celé datové spojení zruší. PAP není silná bezpečnostní metoda. Odesílá ID a heslo v čistě textové podobě. Úroveň zabezpečení je podobná jako u programu telnet.



Obrázek 4 PAP - konfigurační paket

Při autentizaci je nutné informovat server, jakým způsobem chceme ověřit identitu. Proto vyšleme serveru paket, který říká: „K ověření použij metodu PAP“. Nejdůležitější je pole *Authentication-Protocol*, které obsahuje informaci o použitém autentizačním protokolu. V tomto případě obsahuje hodnotu 0xC023 pro PAP.

PAP paket je zapouzdřen v informačním poli rámce PPP datové vrstvy, kde pole *Protocol* obsahuje informaci o typu ověření (0xC023) (Lloyd a Simpson, 1993, s. 4).



Obrázek 5 Obecný PAP paket

Code – je jeden oktet a identifikuje typ PAP paketu. PAP kódy jsou určeny takto:

1. Authenticate-Request (požadavek autentizace)
2. Authenticate-Ack (autentizace úspěšná)
3. Authenticate-Nak (autentizace neúspěšná)

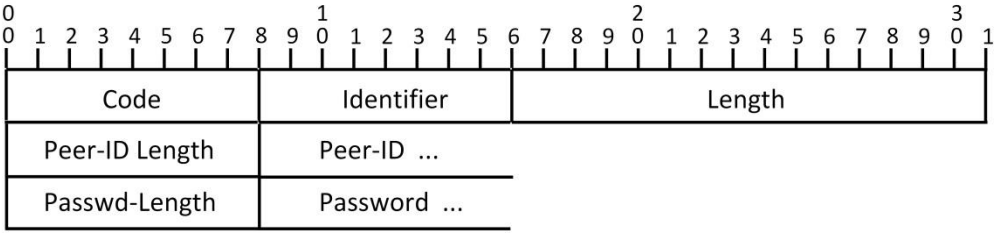
Identifier – je jeden oktet a pomůcka na odpovídající požadavky a odpovědi.

Length – velikost pole jsou dva oktety a udává délku PAP paketu včetně *Code*, *Identifier*, *Lenght* a *Data*.

Data – velikost pole je nula nebo více oktetů. Formát dat je určený podle pole *Code.*

Požadavek autentizace

Authenticate-Request paket používá klient k zahájení PAP autentizace. Tyto pakety se posílají tak dlouho, dokud nedorazí od serveru odpověď, případně dokud nevyprší volitelný čítač opakovaného odesílaní paketů.



Obrázek 6 PAP - Authenticate-Request paket

Code – obsahuje typ paketu Authenticate-Request.

Peer-ID Lenght – udává délku jména klienta.

Peer-ID – obsahuje jméno klienta.

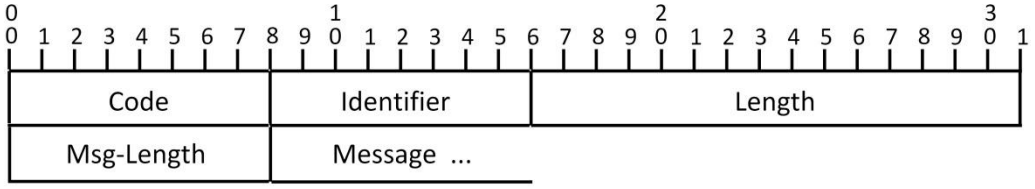
Passwd-Leght – udává délku hesla.

Password – obsahuje uživatelské heslo v textové podobě.

Výsledek autentizace

Můžou nastat dvě situace a to úspěšné ověření nebo neúspěšné ověření. Pokud server obdržel Authenticate-Request paket s ID klienta a heslem klienta, pak musí ověřovatel vyhodnotit informace, zkopírovat identifikátor a odeslat informaci o výsledku.

Pokud server vyhodnotí ověření jako úspěšně, pošle klientovi PAP paket s *Code* 2 (Authenticate-Ack) a považuje autentizaci za ukončenou. V opačném případě server posílá PAP paket s *Code* hodnotou 3 (Authenticate-Nak) a ukončí spojení.



Obrázek 7 PAP - Authenticate-Ack a Authenticate-Nak

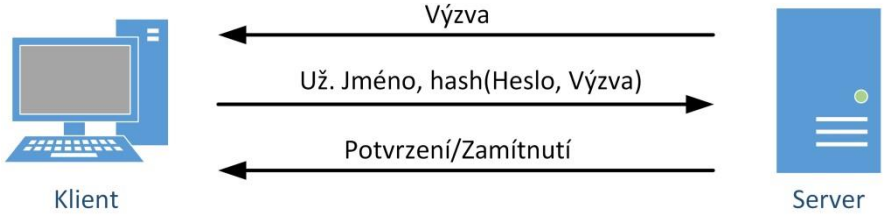
Code – obsahuje informaci, zda se jedná o Ack a Nak paket.

Msg-Lenght – udává délku zprávy (Message).

Message – obsah je závislý na implementaci. Obsahuje doplňující zprávu ve formě textového řetězce. Obsah tohoto pole nesmí ovlivnit funkčnost protokolu. Proto je doporučeno, aby se používaly znaky ASCII z rozsahu 32 až 126 (Lloyd a Simpson, 1993, s. 6).

1.3.2 CHAP

Protokol CHAP (Challenge Handshake Authentication Protocol) oproti PAP používá pro ověření složitější mechanismus. Ten je založen na tom, že ověřovací server pošle klientovi náhodně vygenerovaný řetězec (výzva). Klient přidá k výzvě heslo a vytvoří MD5 kontrolní součet vzniklého celku. Tento kontrolní součet nazývaný hash se odešle ověřovacímu serveru, který zná všechny potřebné údaje (řetězec výzvy, heslo i algoritmus MD5). Server vypočte hash a porovná ho s hashem od klienta. To přináší výhodu, že heslo není posíláno po síti. CHAP umožnuje i ověření v průběhu relace na rozdíl od PAP, který ověřuje pouze na začátku. V dnešní době se nejčastěji setkáme s verzí MS-CHAPv2, která používá obousměrnou autentizaci a MD4 hashovací funkci. Protokol CHAP je popsán v RFC 1994 (Simpson, 1996).



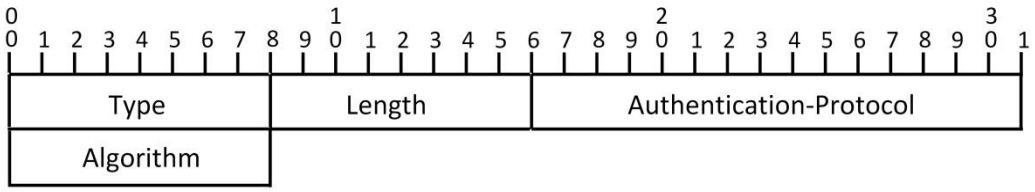
Obrázek 8 Autentizační protokol CHAP

Protokol CHAP ověřuje klienta na základě třícestné výměny, kterou zahajuje server. Nejprve příkazem Challenge (výzva) je klientovi odeslána výzva, která obsahuje náhodný řetězec. Klient použije na sdílené tajemství spojené s výzvou jednocestnou funkci (algoritmem MD5) a vytvoří hash. Hash vloží do odpovědi (Response), kterou odešle. Výsledkem je, že server autentizaci povolí nebo zamítne.

Klíčovou roli zde hrají zasílané challenge a identifikátory jednotlivých paketů. Serverem odeslané pakety obsahují jinou challenge s rostoucím identifikátorem. Tím zajišťuje bezpečnost a znemožňuje odchycení hesla pro pozdější použití. Klient proto musí do svých odpovědí důkladně kopírovat identifikátory, aby server věděl, kterou z klientovi zaslaných challenge má pro svůj vlastní výpočet použít.

Nevýhodou protokolu CHAP je, že tajný klíč musí být uložený na obouch komunikujících zařízení v otevřené formě, protože se používá jako klíč k jednosměrnému hashování. Tajný

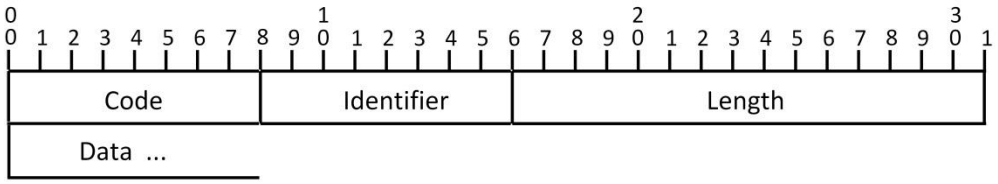
klíč může být hashovaný, ale před použitím je stejně nutné ho dešifrovat. (Dostálek, 2003, s. 27-30)



Obrázek 9 CHAP - konfigurační paket

Konfigurační paket je téměř shodný s konfiguračním paketem PAP. Přibývá pouze pole *Algorithm,* který definuje jednocestný algoritmus.

CHAP paket je zapouzdřen v informačním poli rámce PPP datové vrstvy, kde pole *Protocol* obsahuje informaci o typu ověření (0xC223) (Simpson, 1996, s. 5). Obecný tvar paketu je shodný jako obecný tvar PAP paketu.



Obrázek 10 Obecný CHAP paket

Code – je jeden oktet a identifikuje typ CHAP paketu. CHAP kódy jsou určeny takto:

1. Challenge (výzva),
2. Response (odpověď),
3. Success (úspěšné ověření),
4. Failure (neúspešné ověření.

Identifier – je jeden oktet a pomůcka na odpovídající požadavky a odpovědi.

Length – délka pole jsou dva oktety a udává délku PAP paketu včetně *Code, Identifier, Lenght* a *Data*.

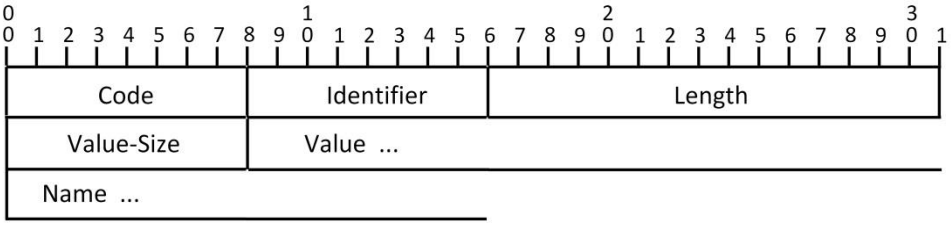
Data – velikost pole je nula nebo více oktetů. Formát dat je určen podle pole *Code.*

Výzva a odpověď

Challenge (*Code* = 1) paket se používá k zahájení CHAP. Server je zasílá tak dlouho, dokud se od klienta neobdrží odpověď *Response* nebo dokud nevyprší čítač opakovaného odesílání paketů. Challenge pakety mohou být rovněž posílány kdykoliv během připojení, aby server zajistil, že se spojení nezměnilo.

Kdykoliv klient přijme challenge paket, musí odpovědět zasláním response paketu. Response paket obsahuje hash challenge (klient hashuje challenge svým heslem) a zkopírovaný identifikátor z challenge paketu.

Když server obdrží odpověď (response), zkontroluje identifikátor, čímž zjistí, jaká byla výchozí challenge. Provede svůj výpočet hashe a porovná svůj výsledek s hashem od klienta. Poté server zasílá klientovi výsledek ověření.



Obrázek 11 CHAP - Challenge a Response paket

Code – obsahuje 1 pro Challenge nebo 2 pro Response.

Identifier – musí být pokaždé změněn, když je odeslán Challenge. Response Identifier musí být zkopírovaný z Challenge.

Value-Size – udává délku pole Value.

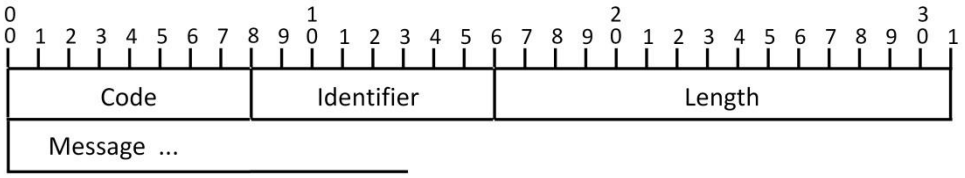
Value – je jeden nebo více oktetů. Nejvýznamnější oktet je přenášen jako první. Hodnota Challenge je proměnný proud oktetů (challenge řetězec). Challenge hodnota musí být změněna při každém dalším odeslání Challenge. Hodnota Response je jednosměrný hash spočítaný klientem (16 oktetů pro MD5).

Name – slouží pro přenos dalších údajů. Nejsou zde žádná omezení. Například může obsahovat ASCII znakové řetězce nebo globálně jedinečné identifikátory.

Výsledek autentizace

Opět mohou nastat dva stavy ověření a to úspěšné (Success) nebo neúspěšné (Failure). Pokud se rovná klientem zaslaná hodnota (hash) v poli Value očekávané (serverem spočítaný hash), autentizace proběhla úspěšně a server posílá klientovi CHAP paket s Code hodnotou 3 (Success).

V opačném případě autentizace proběhla neúspěšně. Server pošle klientovi CHAP paket s Code hodnotou 4 (Failure) a měl by přijmout opatření, pro ukončení spojení.



Obrázek 12 CHAP - Success a Failure paket

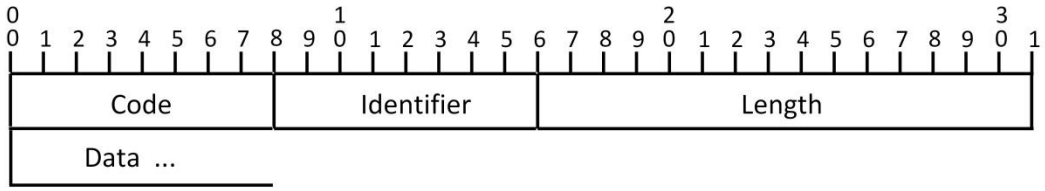
Code – obsahuje v závislosti na výsledku autentizace 3 (Success) nebo 4 (Failure).

Message – je závislé na implementaci. Obsahuje čitelný řetězec a nesmí mít vliv na provoz protokolu. Doporučuje se, aby zpráva obsahovala ASCII znaky 32 až 126. Mechanismy pro rozšíření na jiné znakové sady jsou tématem výzkumu (Simpson, 1996, s. 6).

1.4 EAP

Protokol EAP představuje další vývojový krok v autentizačních protokolech. EAP byl původně vyvinut jako autentizační protokol pro PPP. Rozšiřuje protokol PPP tak, že umožňuje využití libovolné metody ověřování pomocí výměny pověření a informací o libovolné délce. Slouží jako transportní mechanismu pro ověřovací systémy, nikoliv jako samotný ověřovací systém. Dnes se s ním můžeme setkat při autentizaci v datových sítích standartu IEEE 802.3, 802.11 nebo 802.16, jako součást autentizačního rámce IEEE 802.1x. Protokol EAP je popsán v RFC3748 (Aboba a Blunk, 2004).

Velikou výhodou EAP je, že představuje pouze obecný rámec pro autentizaci na principu klient-server, ale samotný proces autentizace vlastně vůbec neřeší. Jednotlivé varianty – tzv. EAP-metoda řeší, jak konkrétně se bude autentizace provádět (bude vzájemná, nebo jednostranná, jestli k autentizaci mají být použity certifikáty X.509 [3]nebo jméno/heslo…). V současné době je definováno přibližně 40 metod.



Obrázek 13 Struktura rámce EAP

Code – Identifikuje typ EAP paketu. Existují čtyři typy:

- Reguest
- Response
- Success
- Failure

Identifier – napomáhá ke spárování dotazů a odpovědí.

Lenght – udává délku celého paketu.

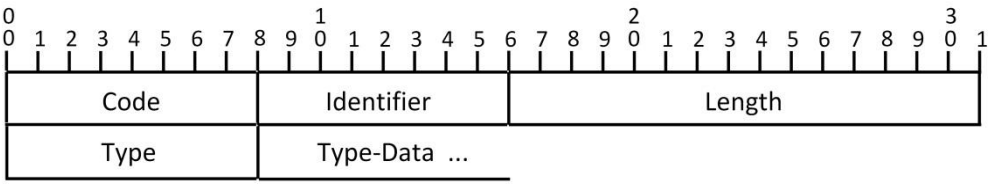
Data – obsahuje vlastní informaci. Obsah a význam informace závisí na typu paketu.

[3] Všeobecně používaná struktura certifikátu zavedená doporučením ITU (Mezinárodní telekomunikační unie)

1.4.1 Request/Response paket

Request paket zasílá server klientovi. Každý request paket má svůj typ, který určuje, jakou informaci si server žádá od klienta. Server neustále posílá klientovi request pakety, dokud nedostane od klienta odpověď. Poté se mění význam Identifier. Při posílání stejného typu žádosti server identifikátor nemění. Novou žádost signalizuje nový identifikátor.

Klient odešle odpověď response paket, do které zkopíruje identifikátor, typ paketu a doplní data požadovaná serverem.



Obrázek 14 EAP Request/Response paket

Code – určuje, zda se jedná o výzvu nebo odpověď.

Type – udává, jaký typ informace je požadován/zasílán. V případě response paketu, může být zaslána hodnota Nak, která sděluje serveru, že klient neumí poskytnout požadovanou informaci.

Type-Date – obsahuje čitelnou zprávu od serveru nebo odpověď zaslanou klientem.

1.4.2 Základní typy Request/Response paketů

Je šest základních typů paketů, první čtyři typy musí být podporovány v každé implementaci protokolu EAP.

1) *Identity* – Používá se pro zjištění totožnosti klienta (uživatelské jméno). V odpovědi může být zaslána textová zpráva, která vyzývá uživatele k zadání jeho ID. Odpověď obsahuje v Type-Data zadané uživatelské jméno.

2) *Notification* – Podobný typ jako Identity. Může obsahovat textovou zprávu pro klienta. Například doba platnosti hesla, důvody neúspěšné autentizace.

3) *Nak* – Je povolen pouze v response paketu. Sděluje serveru, že požadovanou informaci nelze poskytnout.

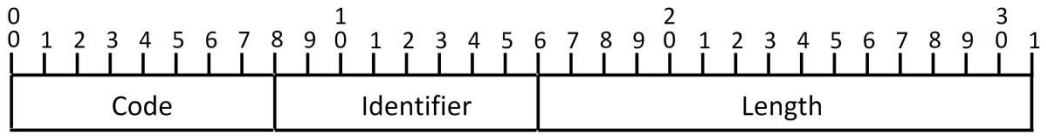
4) *MD5-challenge* – Je velmi podobný protokolu CHAP. Požadavky ověřovatele a odpovědi klienta se odesílají jako zprávy protokolu EAP.

5) *One-Time Password (OTP)* – Je to systém jednorázových hesel. Výzva zahrnuje řetězec obsahující OTP challenge. Odpověď obsahuje několik zadaných slov uživatelem ze slovníku OTP.

6) *Generic Token Card* – Request paket obsahuje ACSII řetězec, který klient opíše do karty a vygeneruje nový řetězec, který se zašle pomocí response paketu serveru.

1.4.3 Success/Failure paket

Výsledek autentizace může být úspěšný nebo neúspěšný. V případě úspěšného ověření EAP, musí autentizátor informovat o úspěchu pomocí Success paketu (pole Code je nastaveno na 3). Pokud autentizátor nemůže ověřit klienta, pak po neúspěšném dokončení EAP, musí autentizátor poslat Failure paket (pole Code je nastaveno na 4). Server může zaslat i několik request paketů před neúspěšným ověřením, ve kterých klienta informuje v čitelných zprávách o důvodech neúspěšné autentizace. Tím je možné omezit lidské chyby jako překlepy a podobně.



Obrázek 15 EAP Success/Failure paket

Code – obsahuje 3 (úspěšná autentizace) nebo 4 (neúspěšná autentizace).

Identifier – je pomůcka pro spárování odpovědi danému Response paketu. Pole musí odpovídat identifikátoru z pole Response paketu.

1.4.4 Metody EAP

EAP-MD5 – Snadná na implementaci, ale má velkou nevýhodu a to minimální zabezpečení. Využívá hashovací funkci MD5, která je náchylná na slovníkové útoky. Dále nepodporuje generování klíčů, které jsou nezbytné pro použití WEP nebo WPA/WPA2. Neumí vzájemnou autentizaci a z toho důvodu je náchylná na útoky typu man-in-the-middle. Klient se může ověřit, ale už nemá možnost ověřit AP. (Aboba a Blunk, 2004, s. 48).

EAP-LEAP – LEAP (Light Extensible Authentication Protocol) je proprietární protokol vyvinutý firmou Cisco v roce 2000, kdy ještě nebyl publikován standart IEEE 802.11i. Mezi jeho vlastnosti patří dynamické WEP klíče a vzájemná autentizace. Je založen na modifikaci MS-CHAPv2. Není bezpečný z důvodu existence nástroje pro získání hesla. Cisco doporučuje LEAP nahradit protokoly EAP-PAEP nebo EAP-TLS.

EAP-TLS – EAP-Transport Layer Security poskytuje velkou míru zabezpečení s širokou podporou mezi výrobci zařízení. Používá PKI (Public Key Infrastructure) s jehož využitím vytváří zabezpečenou komunikaci s RADIUS serverem. Poskytuje vzájemnou autentizaci i obnovu WEP klíčů. Jediná nevýhoda je složitější implementace z důvodu nutnosti certifikátu X.509v3 na straně serveru, tak i na straně klienta. To však přináší výhodu v případě prozrazeného hesla. Útočníkovi je takové heslo bez certifikátu zbytečné. Lze dosáhnout vyšší bezpečnosti uložením certifikátu na smart-kartu.(Simon a Aboba, 2008, s. 3-4)

EAP-TTLS – Metoda TTLS (Tunneled Transport Layer Security) je rozšíření TLS. Byl vyvinut ve Funk Software a Certicom. Odstraňuje problém metody EAP-TLS, jako je potřeba existence certifikátů pro klienty. Server disponuje certifikátem, který klient využije pro vytvoření šifrovaného TLS. To znamená, že celý přenos je zabezpečený. Nativně je podporovaný až od Windows 8, proto se v praxi moc neujala. (Funk, 2008, s. 3-4).

EAP-PSK – Je popsán v experimentálním RFC 4764(Bersani a Tschofenig, 2007). Pro vzájemnou autentizaci používá PSK (Pre-Shared Key). Poskytuje zabezpečenou komunikaci v případě úspěšného ověření.

EAP-PEAP – Metoda PEAP (Protetected EAP) je podobná metodě EAP-TTLS. Byla vyvinuta společností Microsoft, Cisco a RSA Security. Jedná se o jednu z nepoužívanějších metod v praxi. PEAP funguje ve dvou krocích. V prvním kroku klient autentizuje server pomocí jeho certifikátu a tím doje k vytvoření šifrovaného TLS tunelu. Ve druhém kroku se uživatel autentizuje serveru. Existují tři verze:

- PEAPv0/EAP-MSCHAPv2
    - vytvořen Microsoftem
    - velká podpora
- PEAPv1/EAP-GTC
    - vytvořen firmou Cisco
    - není podpora MS Windows, proto se nepoužívá
- PEAPv2
    - podpora pro řízení více EAP metod

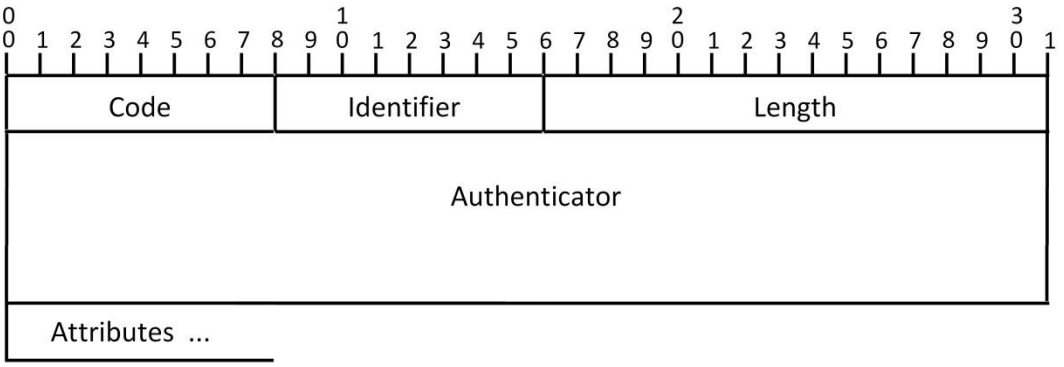
1.5 RADIUS protokol

Protokol RAIDUS (Remote Authentication Dial In User Service) je popsán dokumentem RFC 2865 (Rigney a Williens, 2000), kde najdeme bližší popis. Jedná se pravděpodobně o nejrozšířenější AAA protokol. Provádí centralizovanou autentizaci uživatelů připojujících se na přístupový bod/server NAS (Network Access Server) neboli Authenticator. Vytváří komunikační tok mezi přístupovým bodem a serverem RADIUS. Protokol RADIUS používá na úrovni transportní vrstvy protokol UDP (User Datagram Protocol) z nutnosti rychlé odezvy v momentě požadavku o přihlášení.

NAS funguje jako klient RADIUS, který je zodpovědný za předávání informací o uživateli na určený RADIUS server. V dnešní době lze za přístupový server NAS považovat různá zařízení podporující funkci klient RADIUS. Například routery, switche či AP.

RADIUS server poskytuje služby autentizace a autorizace na UDP portu 1812, accouting je provozován na UDP portu 1813. Samotný protokol RADIUS uživatele neověřuje, pouze obstarává komunikaci mezi serverem RADIUS a NAS (RADIUS klientem).

NAS se dotazuje serveru RADIUS, jestli může uživatele do sítě pustit a za jakých podmínek. Server RADIUS odpoví kladně nebo záporně. V případě kladné odpovědi, připojí do odpovědi podmínky připojení uživatele. Přístupový server si tuto odpověď převezme a umožní přístup uživateli do sítě.



Obrázek 16 Hlavička paketu RADIUS

Code – identifikuje typ RADIUS paketu. Pokud je paket s neplatným kódem, je zahozen. Existuje 9 typů zpráv.

Tabulka 1 RADIUS typy zpráv

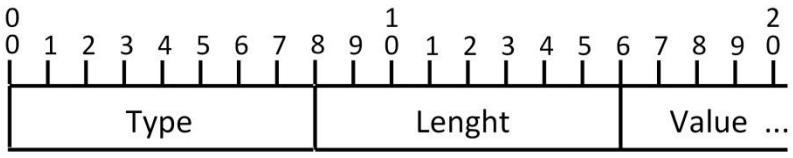
Hodnota pole Code	Typ zprávy	Význam
1	Access-Request	NAS -> RADIUS, žádost o autentizaci
2	Access-Accept	RADIUS -> NAS, autentizace úspěšná, autorizace
3	Access-Reject	RADIUS -> NAS, autentizace neúspěšná
4	Accouting-Request	NAS -> RADIUS, žádost o accountingu
5	Accouting-Response	RADISU -> NAS, potvrzení accountingu
11	Access-Challenge	RADIUS -> NAS, Žádost o další informace od klienta (pokračování EAP)

Identifier – Slouží k porovnání dotazů a odpovědí. Umožňuje identifikovat duplicitní pakety. Toto pole je nezbytné z důvodu používání bez stavového protokolu UDP.

Lenght – Udává délku paketu (minimálně 20B a maximálně 4096B).

Authenticator – Používá se pro ověření odpovědi z RADIUS serveru. Při pokládání dotazu vygeneruje NAS náhodný řetězec, který slouží pro identifikaci odpovědi. Server RADIUS pak do stejného pole vrátí kontrolní součet vypočítaný pomocí MD5 z celého vráceného paketu. Tento kontrolní součet spojí s řetězcem sdíleného tajemství a odešle serveru NAS. Server NAS přijme odpověď od RADIUS serveru, porovná řetězec sdíleného tajemství přijatý v odpovědi s řetězcem sdíleného tajemství, který má NAS přiřazen k danému RADIUS serveru. Pokud se řetězce rovnají, odpověď je považována za správnou.

Attributes – Toto pole udává specifické atributy. Ty udávají konkrétní autentizaci, autorizaci, informační a konfigurační údaje pro žádosti a odpovědi. Konec seznamu atributů je řízen délkou RADIUS paketu. (Rigney a Williens, 2000, s. 17),



Obrázek 17 Formát atributů protokolu RADIUS

Type – specifikuje typ atributu. V následujícím seznamu je popsáno několik základních atributů. Všechny typy jsou vypsány v RFC 2865 (Rigney a Williens, 2000, s. 23),

1. User-Name — jméno uživatele, který má být ověřen
2. User-Password — heslo uživatele
3. CHAP-Password — CHAP ID + jednorázové heslo
4. NAS-IP-Address — IP adresa NAS, žádajícího
5. NAS-Port — číslo portu NAS

Length – udává velikost hodnoty v oktetech

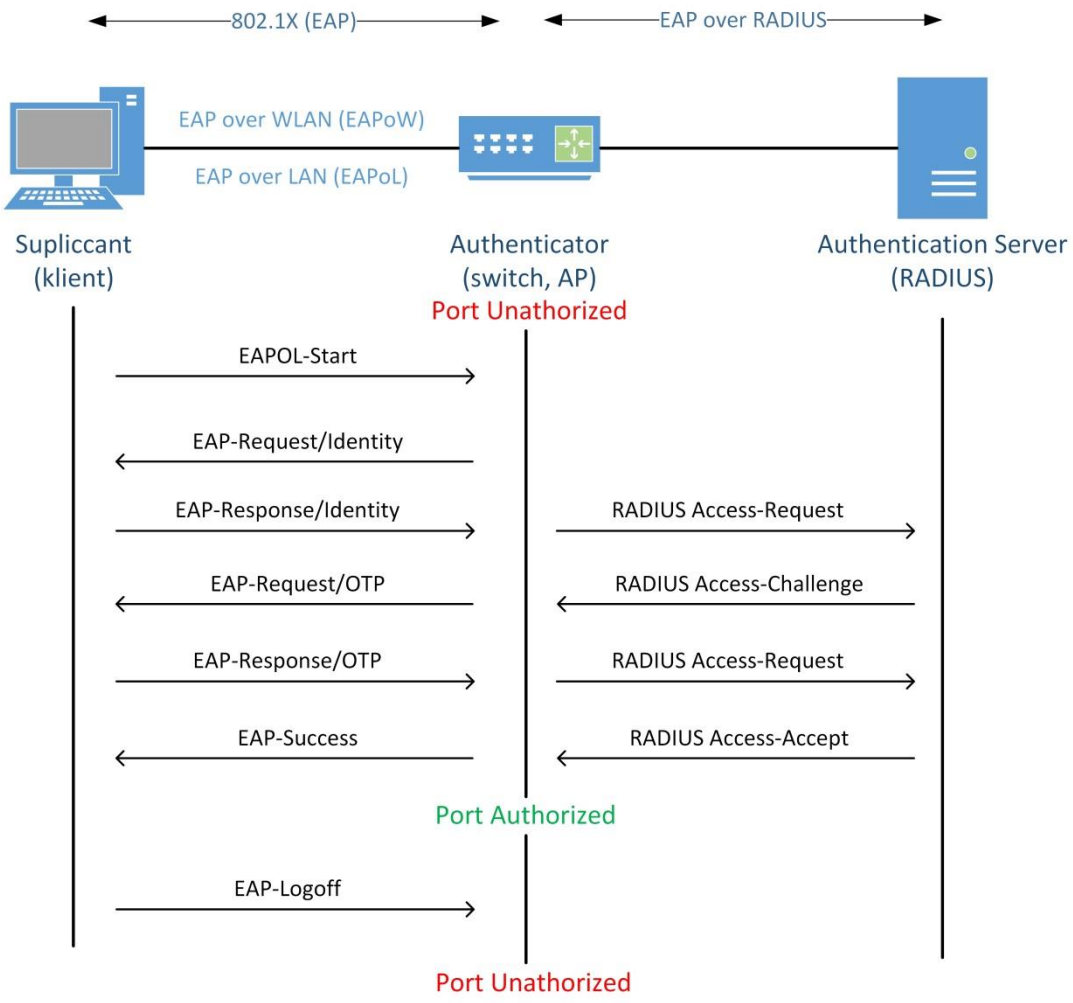
Value – Obsahuje informace pro specifický atribut. Formát a délka hodnoty pole se určí podle typu. Může obsahovat textový řetězec v kódování UTF-8, binární data, 32 bitové číslo nebo čas v sekundách od 1. Ledna 1970.

1.6 Proces ověření pomocí 802.1x

Je to jedna ze součástí AAA mechanismu. Pro přehlednost AAA (Authentication, Authorization, Accountig).

- Autentizace – poskytuje metody identifikace uživatele a kontrolu oprávnění přístupu k síťovým službám.
- Autorizace – definuje přístup k síťovým zdrojům na základě autorizace uživatele.
- Accountig (Účtování) – zaznamenává informace o využívání síťových služeb nebo zdrojů uživatelem.

Klient se připojí na fyzický port u switche nebo virtuální port u AP, který je ve stavu „uzavřen“. Je povolen pouze protokol EAP. Suplikant (speciální program), běžící na straně klienta zahájí ověření přes EAP protokol. Suplikant vyšle na NAS žádost o autentizaci, následně NAS naváže spojení s RADIUS serverem a zprostředkuje ověření klienta vůči RADIUSu. O výsledku informuje RADIUS server NAS. Ten v závislosti na odpovědi zapíná nebo vypíná porty. (Wendell, Russ a Naren, 2009, s. 631-632)



Obrázek 18 Průběh autentizace

1) Klient posílá informaci „EAPOL-Start“ o tom, že se chce přihlásit do sítě.
2) Autentizátor pošle požadavek „EAP-Request/Identity“ klientovi na ověření totožnosti zabalenou do rámce EAPoL nebo EAPoW a to v případě, kdy autentizátor přijme „EAPOL-Start“ nebo detekuje aktivní port.
3) Klient žádost vyhodnotí a odpoví autentizátorovi informací o své totožnosti (uživatelské jméno) „EAP-Response/Identity“.

   Autentizátor informaci přijme, vybalí ji z EAPoL rámce a zabalí do datagramu protokolu RADIUS, který odešle na autentizační server RADIUS „RADIUS Access-Request“.
4) Autentizační server posílá autentizátoru výzvu pro ověření „RADIUS Access-Challenge“.

Autentizátor přebírá paket, který z datagramu protokolu RADIUS zabalí do EAPoL rámce, a posílá klientovi „EAP-Request“.

Klient odpoví autentizátorovi „EAP-Response“ např. k výzvě přidá své heslo, provede hash a ten použije jako odpověď. Autentizátor odpověď přijme, vybalí ji z EAPoL rámce, zabalí do protokolu RADIUS a pošle na autentizační server „RADIUS Access-Request“.

5) Autentizační server provede ověření. V případě úspěchu odpoví autentizátorovi rámcem „ RADIUS Access-Accept“ a při neúspěchu „RADIUS Access-Reject“.

   Pokud autentizátor obdrží rámec „RADIUS Access-Accept“, přepne daný port z neautorizovaného stavu na autorizovaný a povolí normální síťovou komunikaci s možností restrikcí na základě atributů od autentizačního serveru (přiřazení do určité VLAN, nastavení filtrování). Autentizátor posílá klientovi rámce „EAP-Success“, kterým informuje o úspěšné autentizaci.

   V případě rámce „RADIUS Access-Reject“ žádost zamítne a nechá port ve stavu neautorizovaném. Klient pak obdrží EAPoL rámce „EAP- Failure“.

Pokud klient spojení ukončí, může poslat zprávu EAP-Logoff, kterou autentizátorovi oznamuje, že nebude dále komunikovat. Ten port převede do stavu neautorizovaného. Do neautorizovaného stavu se může přepnout i v případě, dojde-li k fyzickému odpojení klienta (vytažením kabelu ze zásuvky) nebo z důvodu vypršení časového limitu, kdy se měl klient autentizovat.

2 Mikrotik

Velmi často se název „Mikrotik“ zaměňuje za samotný hardware, což je špatně. Mikrotik je název firmy, která vyvíjí a vyrábí RouterOS (operační systém) a RouterBoard (hardware). Firma byla založena v roce 1995 a už od samého začátku se zabývala vývojem bezdrátových technologií a směrovačů.

V roce 1997 firma začala vytvářet samotný RouterOS (routovací operační systém), který poskytuje rozsáhlé funkce, kontrolu a flexibilitu pro všechny druhy síťových rozhraní a směrovačů. Od roku 2002 firma vyrábí vlastní hardware (RouterBOARD).

Může to znít překvapivě, ale sídlo má v hlavním městě Lotyšska v Rize a má 80 zaměstnanců. (*Mikrotik*, 2014)

2.1 RouterOS

Operační systém využívaný na RouterBOARDech. Je vhodný na bezdrátové spoje nebo jako HW firewall, či router se snadnou GUI konfigurací. Je distribuován v podobě instalačního balíčku NPK pro embedded systémy, v podobě předinstalovaného systému na RouterBOARDU, nebo v podobě ISO souboru pro vypálení na CD. Je koncipován pro platformy i386, mips, powerpc. Z toho důvodu může být nainstalovaný i na běžný počítač, ze kterého RouterOS vytvoří router se všemi potřebnými funkcemi.

RouterOS je postavený na základě linuxového jádra verze 2.6. Je zaměřen na rychlou a jednoduchou instalaci a snadno použitelné rozhraní. Podporuje multi-core a multi-CPU počítače (SMP). Může fungovat na nejnovějších základních deskách a procesorech od firmy Intel. Má velké množství podporovaných sítových rozhraní včetně 10 Gigabit Ethernet karty. Dále podporuje 802.11/a/b/g/n bezdrátové karty a 3G modemy. Samotný RouterOS má mnoho funkcí a různých nastavení. Mezi nejzákladnější patří:

- Bezpečnostní firewall,
- omezující firewall (QoS)
- VPN tunel s podporou PPP, PPTP, L2TP, IPsec,
- Hotspot,
- proxy server,

- bridge,
- router,
- syslog.

2.1.1 Konfigurace

RouterOS podporuje různé metody konfigurace – lokální přístup, sériovou komunikaci s využitím terminálové aplikace, Telnet a zabezpečený SSH přístup, konfigurační nástroj s GUI (Winbox), webové konfigurační rozhraní a programovací API pro vytvoření vlastního rozhraní. Také podporuje přístup založený na úrovni MAC – Mac-Telnet a Winbox nástroje v případě, že je problém s komunikací na úrovni IP. (*Mikrotik RouterOS,* 2010)

2.1.2 Firewall

RouterOS implementuje firewall, který provádí filtrování paketů a tím poskytuje bezpečnostní funkce, které jsou nutné pro správu proudících dat přes router, či AP. Spolu s NAT (Network Address Translation) zabraňuje neoprávněnému přístupu do připojených sítí. Slouží i jako filtr pro odchozí komunikaci.

Dále RouterOS obsahuje stavový firewall, který provádí stavovou kontrolu paketů a udržuje stav připojeních sítí, přes které se komunikuje. Firewall obsahuje funkce, které využívají vnitřních připojení, směrovacích a paketových značek. Je možné filtrovat podle IP adres, rozsah adres, portů a rozsahů portů, IP protokolů, DSCP a dalších parametrů. Také podporuje statické a dynamické seznamy adres. RouterOS podporuje firewall i pro IPv6 verzi. (*Mikrotik RouterOS,* 2010)

2.1.3 Routing

RouterOS podporuje velké množství směrovacích protokolů.

- Pro IPv4 – RIP v1 a v2, OSPF v2 a BGP v4,
- pro IPv6 – RIPng, OSPF v3 a BGP.

Také podporuje virtuální routing a forwarding (VRF).

2.1.4 Forwarding

RouterOS podporuje forwarding (přeposílání), bridging (přemostění), Mesh a WDS (Wireless distribution system).

WDS umožňuje vytvářet vlastní bezdrátové pokrytí pomocí více AP.

2.1.5 HotSpot

Další zajímavou funkcí, kterou RouterOS disponuje, je HotSpot. Ten umožňuje veřejný přístup k síti pro klienty využívající bezdrátové nebo drátové připojení. Při prvním otevření webového prohlížeče se uživateli zobrazí přihlašovací obrazovka. Po správném vyplnění přihlašovacích údajů je klientovi umožněn přístup ke službám. Toto řešení je ideální pro hotely, letiště, internetové kavárny nebo pro kterékoliv veřejné místo. Není potřeba žádná konfigurace sítě nebo instalace softwaru na straně klienta.

Správa uživatelů je řešena přes uživatelské profily. Každý profil může mít nastavený uptime, omezený upload a download, omezené množství přenesených dat. Hotspot také umožňuje autentizaci vůči RADIUS serveru nebo proti uživatelům vytvořených přímo v RouterOS. (*Mikrotik RouterOS*, 2010)

2.1.6 Licence

V případě předinstalovaných RouterOS na RouterBOARDech není potřeba se o licenci starat. Licence se kupuje spolu s hardwarem. Pro systémy x86, je nutné licenční klíč zakoupit.

Licenční klíč je blok symbolů, který je potřeba zkopírovat z mikrotik účtu, nebo z e-mailu, který jsme obdrželi po zakoupení. Klíč můžeme vložit kdekoliv v terminálu nebo pomocí nástroje Winbox. Aby se projevily změny, je nutné zařízení restartovat. RouterOS licence jsou založeny na SoftwareID, který je vázán na paměťové medium (HDD, NAND).

K dispozici je celkem 6 úrovní licencí, z toho jsou čtyři placené. Licence 0 je pouze k vyzkoušení a je limitována časem. Licence 1 je demo a rozdíly mezi placenými a neplacenými licencemi je uveden v tabulce. Placené verze začínají od úrovně 3, která má

funkci bezdrátového klienta a omezený počet aktivních klientů. Naopak bez omezení je nejvyšší licence 6. (*Mikrotik RouterOS*, 2010)

Tabulka 2 Licence RouterOS (Mikrotik, 2013)

Level number	**0 (Demo)**	**1 (Free)**	**3 (WISP CPE)**	**4 (WISP)**	**5 (WISP)**	**6 (Controller)**
Price	no key	registration required	volume only	$45[4]	$95[4]	$250[4]
Upgradable To	-	no upgrades	ROS v7.x	ROS v7.x	ROS v8.x	ROS v8.x
Initial Config Support	-	-	-	15 days	30 days	30 days
Wireless AP	24h trial	-	-	yes	yes	yes
Wireless Client and Bridge	24h trial	-	yes	yes	yes	yes
RIP, OSPF, BGP protocols	24h trial	-	yes	yes	yes	yes
EoIP tunnels	24h trial	1	unlimited	unlimited	unlimited	unlimited
PPPoE tunnels	24h trial	1	200	200	500	unlimited
PPTP tunnels	24h trial	1	200	200	500	unlimited
L2TP tunnels	24h trial	1	200	200	500	unlimited
OVPN tunnels	24h trial	1	200	200	unlimited	unlimited
VLAN interfaces	24h trial	1	unlimited	unlimited	unlimited	unlimited
HotSpot active users	24h trial	1	1	200	500	unlimited
RADIUS client	24h trial	-	yes	yes	yes	yes
Queues	24h trial	1	unlimited	unlimited	unlimited	unlimited
Web proxy	24h trial	-	yes	yes	yes	yes
User manager active sessions	24h trial	1	10	20	50	Unlimited
Number of KVM guests	none	1	Unlimited	Unlimited	Unlimited	Unlimited

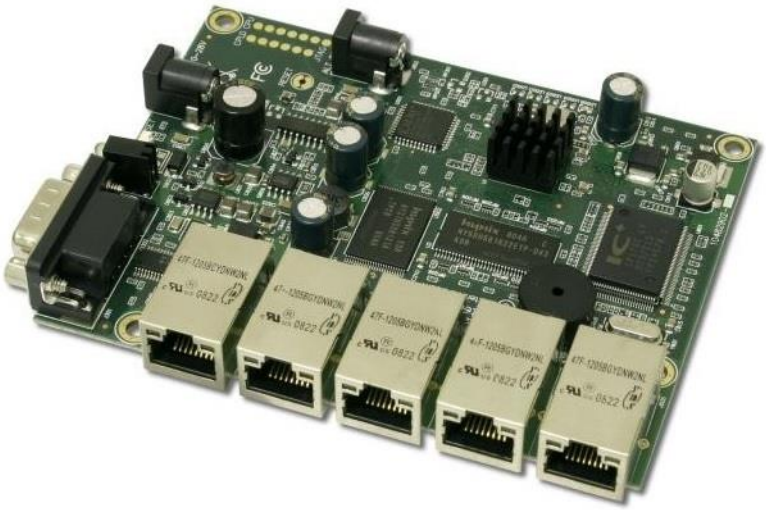
[4] Ceny jsou pouze informativní.

2.2 RouterBOARD

Jedná se o optimalizované základní desky firmy Mikrotik, které jsou specializované pro potřeby počítačových sítí. Lze je snadno rozšířit pomocí modulů miniPCI (rozšiřující karty ve standardu miniPCI pro bezdrátovou komunikaci, rozšíření ethernetových portů) a miniGBIC (slouží pro rozšíření RouterBoardů s SFP slotem pro připojení do optické trasy).

RouterBoard je dodáván spolu s RouterOS. Díky vlastnostem RouterOS a snadné rozšiřitelnosti RouterBordu lze využít, jako domácí AP, tak i pro potřeby ISP. Modely RouterBoardu se liší především výkonem, licencí RouterOS a možnostmi rozšířením. Jednotlivé modely se mohou lišit přizpůsobením danému použití (domácí, venkovní, v racku).

RouterBoardy obsahují procesory Atheros, MPC a PowerPC. Mají vlastní operační paměť z pravidla 32Mb – 256Mb. Samotný RouterOS je nainstalován v NAND paměti.



Obrázek 19 Ukázka RB450G (Mikrotik.cz, 2014)

3 ISP

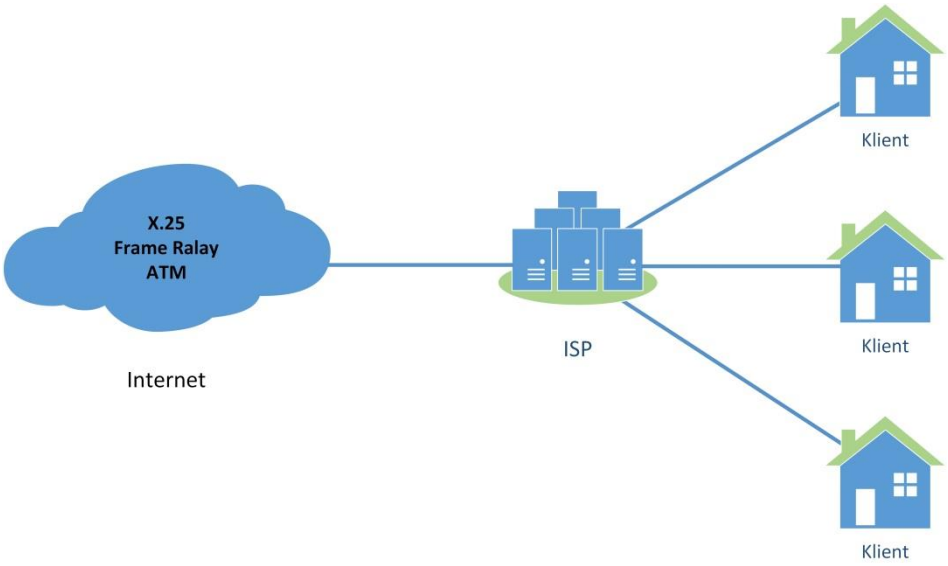
Dnes se ISP (Internet Service Provider) vyznačuje obrovskou rozmanitostí nabízených služeb, lišících se od tradičních telekomunikačních služeb přes použité technologie. Máme určitou podmnožinu internetových služeb, které nabízejí poskytovatelé internetových síťových služeb – INSPs (Internet network service providers).

ISP poskytuje jednu nebo více internetových služeb. ISP můžeme chápat jako společnost, která nabízí telefonní služby nebo jiné telekomunikační služby, jako je například připojení k internetu. Jednoduše ISP je firma umožňující hlasové nebo datové služby zákazníkům. Jedná se o firmu, která službu provozuje a vyúčtovává za ni zákazníkovi poplatky za užívání.

ISP nabízí připojení k internetu přes dial-up, DSL (Digital Subscriber Line) technologie jako je ADSL (Asymmetric Digital Subscriber Line), či VDSL - VHDSL(Very High Speed DSL), dále pomocí optických vláken, ISDN (Integrated Services Digital Network), bezdrátový přístup pomocí Wi-Fi.

Mezi domácnostmi je nejoblíbenější a nejdostupnější připojení přes ADSL nebo bezdrátové připojení. V případě panelových domů je nejrozšířenější připojení přes kabelovou televizi. Těžko si představit v dnešní době cloudu a náročnosti služeb, jako je sledování filmů a poslech hudby, připojení pomocí ISDN.

Tak jako ISP připojuje své klienty k internetu i samotní ISP musí být připojeni k internetu přes nadřízené ISP.



Obrázek 20 ISP

4 Hardware a software pro realizaci

V této části jsou popsány základní informace o navržených technologií.

Pro praktickou část bakalářské práce jsem volil z dostupných technologií a zařízení. Kladl jsem důraz na GPL licence a minimální finanční náročnost. Jako RADIUS server jsem zvolil zařízení Raspberry PI s distribucí Rasbian a softwarem FreeRadius. Pro klienta RADIUS serveru a DHCP serveru jsem využil zapůjčené zařízení RouterBoard RB750 a RB133 se systémem RouterOS.

4.1 Mikrotik RB750 a RB133

Mé řešení je založeno na zapůjčených RouterBordech RB750 a RB133 od firmy Mikrotik. Nevýhoda RB750 je, že nemá WLAN rozhraní a proto se zaměřím na konfiguraci RADIUS klienta a DHCP serveru. Na zařízení RB 133 otestuji konfiguraci WLAN a autentizaci vůči serveru RADIUS.

4.2 Raspberry PI

Jedná se o velmi levný počítač velikosti platební karty. Je vyvíjen britskou nadací Raspberry Pi Foundantion. Jeho hlavním cílem je podpora ve výuce informatiky ve školách. Umožňuje všem věkovým kategoriím prozkoumat výpočetní techniku a naučit se základy programování v jazycích Screcth a Python. Disponuje procesorem ARM s taktem 700 MHz, grafickým procesorem VedeoCore IV a v závislosti na verzi 256 MB RAM nebo 512 MB RAM s LAN portem. Obsahuje kompozitní výstup i HDMI konektor. Lze jej využít i do oboru automatizace, protože obsahuje rozhraní GPIO. Raspberry PI je poměrně výkonný a zvládne většinu práce, kterou běžný uživatel provádí na svém klasickém PC. Jako operační systém využívá hned několik distribucí, jako jsou Rasbian, Pidora, Arch Linux. (*Raspberry Pi Foundation,* 2013)

Obrázek 21 Raspberry PI (Alza.cz, 2014)

Jeho velikou přednostní je minimální spotřeba, která nepřekračuje 3W. Díky tomu je ideálním řešení pro nepřetržitě běžící aplikace. Může sloužit jako web server, síťové datové úložiště, tiskový server i jako RADIUS server. S ohledem na výkon Raspberry PI není vhodný pro služby, které využívají stovky uživatelů zároveň. Postačí pro méně náročné funkce, jako je autentizace několika uživatelů za hodinu.

Pro zvolené řešení jsem se rozhodl z důvodů minimální spotřeby, jeho rozměrům a nárokům na místo. Také chci dokázat, že pro dané účely postačí a není potřeba drahý server.

4.3 Server RADIUS

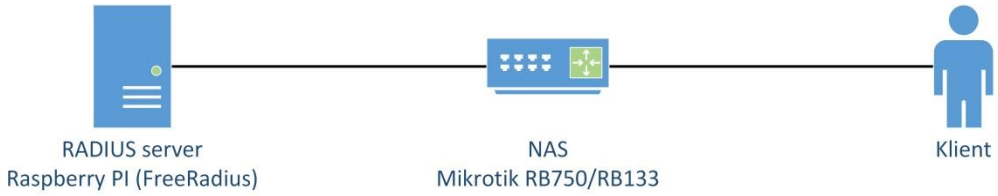
Existuje mnoho serverů RADIUS, jak pro platformy GNU/Linux tak i pro Windows. RouterOS klient by měl pracovat se všemi RADIUS servery kompatibilními s RFC dokumentem. Při výběru RADIUS serveru jsem volil ze tří, které byly na RouterOS testovány (*Mikrotik*, 2007). Server FreeRadius, XTRadius a Steel-Belted Radius. Z těchto nabízených serverů jsem zvolil FreeRadius, který má veřejnou licenci GPL a má velmi podrobnou dokumentaci na internetu i v samotných konfiguračních souborech.

Jako nevýhodu FreeRadius serveru můžeme považovat databázi klientů, která je v základním balíčku lokálního charakteru. To znamená, že uživatelé jsou ověřováni na základě souboru `etc/freeradius/users.conf`, což může vést ke špatné distribuci databáze uživatelů záložním serverům. Z tohoto důvodu je vhodné, zvolit balíček, který umí pracovat s SQL databázemi.

SQL databázi jsem zvolil MySQL a to z několika důvodů. Hlavním důvodem je GPL licence, funkčnost v operačním systému GNU/Linux a vlastní zkušenosti s administrací MySQL databázích. Existuje však mnoho dalších SQL databází. Například PostgreSQL, LDAP (Lightweight Directory Access Protocol), Microsoft SQL server, Oracle SQL nebo Kerberos. Zajímavé a oblíbené řešení je pomocí Oracle SQL. Jeho nevýhodou oproti zvolené MySQL je, že Oracle SQL je komerční produkt. Z důvodu použití systému GNU/Linux je nemožné použít Microsoft SQL server.

5 Návrh implementace 802.1x

Pro laboratorní testování jsem zvolil jednoduchou topologie sítě. Topologie obsahuje jeden RADIUS server. NAS server, který slouží jako klient RADIUS a DHCP server. A samotný klient, který bude ověřován vůči RADIUS serveru.



Obrázek 22 Schéma praktické ukázky

5.1 Instalace RADIUS serveru

Instalace aplikace FreeRADIUS je velmi snadná díky balíčkovacímu systému APT v distribuci Rasbian. Je třeba spustit následující příkaz jako administrátor (root uživatel):

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

Po instalaci si FreeRADIUS vytvoří konfigurační soubory v adresářích /etc/raddb, nebo /etc/freeradius. V distribuci Rasbian se konfigurační soubory nachází v adresáři /etc/freeradius. FreeRADIUS obsahuje řadu konfiguračních souborů, pomocí kterých se konfiguruje. Uvedu zde a popíšu jen ty nejdůležitější.

radiusd.conf

Základním konfiguračním souborem serveru FreeRadius je soubor radiusd.conf. Je popěrně rozsáhlý, nicméně většina jeho obsahu jsou komentáře. U většiny atributů lze ponechat výchozí hodnoty. Popis těchto atributů překračuje rámec této práce a nebudu se jimi zabývat.

eap.conf

V souboru eap.conf se nastavují bezpečnostní mechanizmy a jejich parametry používané při autentizaci.

clients.conf

Tento konfigurační soubor slouží pro definování přístupových NAS serverů (klientů RADIUS). Pokud zde není uveden záznam určitého NAS serveru, nemůže komunikovat s RADIUS serverem. Příklad níže obsahuje IP adresu (doménové jméno) NAS, tajné heslo pro komunikaci, název zařízení pro identifikaci a typ zařízení.

```
client 192.168.0.10 {
        secret          = testmikrotik
        shortname       = RB750
        nastype         = other
}
```

Doporučuje se pro každý NAS server zadávat vlastní tajné heslo a název.

sql.conf

Soubor sql.conf je jeden z nejdůležitějších pro konfiguraci FreeRadius serveru s využitím SQL databáze. Obsahuje informace pro připojení k databázi, typ SQL databáze a názvy jednotlivých tabulek.

users.conf

V souboru users.conf jsou uvedeni lokální uživatelé. Samotný soubor obsahuje mnoho příkladů, jak přidat uživatele. Vzhledem k tomu, že využiji SQL databázi pro správu klientů, nebudu tento konfigurační soubor dále využívat.

5.2 Konfigurace RADIUS serveru

V této kapitole se věnuji samotné konfiguraci RADIUS server. Pro připojení FreeRADIUS k databázi MySQL je nutné mít nainstalovaný balíček freeradius-mysql a změnit konfiguraci.

5.2.1 Vytvoření tabulek v MySQL

Je nutné naistalovat MySQL databázi, která bude obsahovat řídící tabulky pro FreeRADIUS. Pro lepší přehled nad databází doporučuji nainstalovat phpMyAdmin. Postup instalace je popsán v příloze A.

Doporučuji si před samotným vytvářením tabulek přečíst soubor sql.conf, kde je podrobně popsáno, co se žádá. Jsou zde uvedeny příklady SQL dotazů a názvy tabulek. Tabulky si můžeme vytvořit sami. Pro zjednodušení využiji defaultní schéma SQL tabulek ze souboru /etc/freeradius/sql/.

Přihlášení do MySQL a vytvoření nové databáze:

```
mysql –u root -p
```

Poté vytvoříme novou databázi *radius* a uživatele *radius* s heslem *radiusheslo.*

```
mysql> create database radius;
mysql> grant all on radius.* to radius@localhost identidied by "radiusheslo";
```

Do vytvořené databáze radius se vloží schéma z FreeRadius.

```
mysql -u root -p radius < /etc/freeradius/sql/mysql/schema.sql
```

Tento příkaz vytvořil tabulky ze schématu schema.sql. Pro kontrolu si můžeme nechat vypsat jednotlivé tabulky.

```
mysql> show tables;
+------------------------+
| Tables_in_radius |
+------------------------+
| radacct |
| radcheck |
| radgroupcheck |
| radgroupreply |
| radpostauth |
| radreply |
| radusergroup |
+------------------------+
mysql>
```

Popis tabulek a jednotlivých sloupců je v příloze B.

5.2.2 Konfigurace radiusd.conf a sql.conf

Po úspěšném vytvoření MySQL databáze je zapotřebí nastavit i samotný FreeRADIUS. Nejprve se upraví konfigurační soubor sql.conf následovně:

```
database = "mysql"
driver = "rlm_sql_mysql"
server = "localhost"
login = "radius"
password = "radiusheslo"
radius_db = "radius"
```

Ukazuji pouze změněná nastavení. Pro další nastavení odkazuji na dokumentaci FreeRADIUS (*FreeRADIUS* , 2014).

Další úprava, kterou je nutné provést je v souboru `/etc/freeradius/sites-enable/default`. Je potřeba odkomentovat *sql* v sekci authorize{}, accounting{}, session{} a auth{}.

Nakonec musíme upravit konfigurační soubor radiusd.conf. Je třeba nastavit FreeRADIUS, aby používal modul *SQL*. Smazáním komentáře u `$INCLUDE sql.conf` docílíme, že FreeRADIUS začne tento modul využívat.

5.2.3 Konfigurace eap.conf

FreeRADIUS podporuje spoustu metod. Například EAP-MD5, EAP-TLS, PEAP, EAP-TTLS. Více jich nalezneme v dokumentaci (*FreeRADIUS*, 2014). Já zvolil EAP-TTLS s protokolem MS-CHAPv2 a to z důvodu, že není ze strany klienta vyžadován certifikát jako u metody EAP-TLS. Proto je konfigurace méně náročná.

Myslím si, že pro účely ISP ověřovat své vlastní zařízení by postačila metoda EAP-MD5. Bohužel zařízení od firmy Mikrotik tuto metodu nepodporují a celkově se metoda EAP-MD5 nedoporučuje používat. Hlavním důvodem jsou útoky typu man-in-the-middle. Proto ji ani nenajdeme například ve Windows 7.

Pokud bychom chtěli autentizovat zařízení od firmy Mikrotik je nutné využít metodu EAP-TLS, kterou RouterOS podporuje.

Pro konfiguraci využiji testovací certifikáty z adresáře /etc/freeradius/certs a déle je nebudu popisovat.

Je potřeba nakonfigurovat EAP-TTLS i EAP-TLS. V našem případě nám stačí výchozí konfigurace EAP-TLS.

V konfiguračním soboru provedeme tyto změny a zbytek ponecháme.

```
eap {
        default_eap_type=ttls
        .....
      ttls {
            default_eap_type=mschapv2
            .....
        }
      }
```

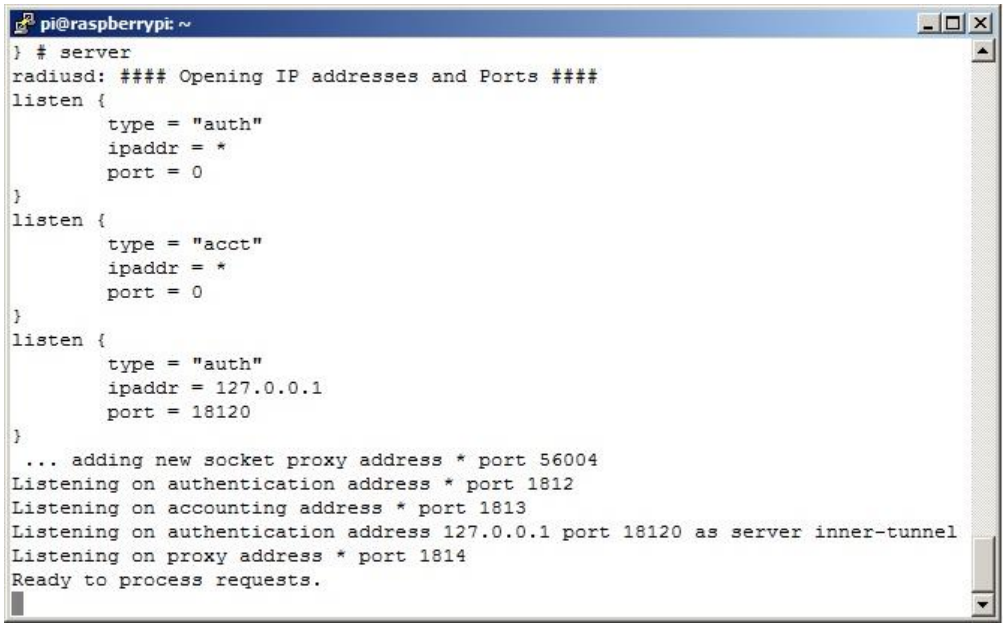
Tímto jsme nastavili jako výchozí ověřování pomocí EAP-TTLS s protokolem MS-CHAPv2. MS-CHAPv2 využívá šifrovaný tunel TLS, který vytvoří EAP-TTLS. Klient se díky MS-CHAPv2 ověřuje uživatelským jménem a heslem.

5.2.4 Funkčnost FreeRADIUS a MySQL

Než se pustíme do konfigurace NAS serveru (klienta RADIUS), doporučuji otestovat funkčnost FreeRADIUS serveru s MySQL databází. Vložíme do databáze záznam.

```
mysql> INSERT INTO radcheck (UserName, Attribute, Value) VALUES ('sqltest',
'Password', 'test');
```

Spustíme FreeRADIUS v „debug“ modu příkazem `freeradius-X`. Pokud je v konfiguraci chyba, zobrazí se ve výpisu, kde se chyba nachází. Na obrázku 22 je vidět standartní výpis



Obrázek 23 Výpis debug modu FreeRadius

bez chyb s porty, na kterých naslouchá.

Pro ověření funkčnosti s databází MySQL otevřeme nový shell, kde spustíme následující příkaz.

```
$ radtest sqltest test localhost 18128 testing123
```

```
radtest {username} {password} {hostname} 10 {radius_secret}
```

Příkaz vypisuje, jaké atributy se posílají v Requestu a zda byl požadavek úspěšný. Poslední řádek ukazuje výsledek ověření. V našem případě Access-Accept. Tím ověříme funkčnost FreeRadiusu a MySQL databáze.

```
Sending Access-Request of id 136 to 127.0.0.1 port 1812
        User-Name = "sqltest"
        User-Password = "heslo"
        NAS-IP-Address = 127.0.1.1
        NAS-Port = 18128
        Message-Authenticator = 0x00000000000000000000000000000000
rad_recv: Access-Accept packet from host 127.0.0.1 port 1812, id=136, length=20
```

5.3 Konfigurace RADIUS klienta a DHCP

Konfiguraci RouterBOARDu lze provést pomocí aplikace Winbox nebo příkazové řádky. Záleží na každém, pro co se rozhodne. Winbox je pohodlný nástroj, ale z důvodu velikého množství ukázkových obrázků jsem zvolil příkazovou řádku. V popisu konfigurace nebudu rozlišovat mezi RB750 a RB133 z důvodu téměř stejné konfigurace. Pouze na změny upozorním.

5.3.1 Konfigurace DHCP

Pro připojování klientů k sítí je nezbytné nakonfigurovat server DHCP. Služba DHCP umožňuje automatické přiřazování IP adres klientům. DHCP je protokol rodiny TCP/IP a pracuje na aplikační vrstvě.

Nastaví se IP adresu na zvolený interface.

```
[admin@MikroTik] /ip address> add address 192.168.0.1/24 interface eth3
```

V případě RB 133 změníme pouze interface.

```
[admin@MikroTik] /ip address> add address 192.168.0.1/24 interface wlan1
```

Výpis DHCP serverů se provede příkazem *print*. Z výpisu je zřejmé, že DHCP server není aktivní na žádném interfacu.

```
[admin@MikroTik] /ip> dhcp-server print
Flags: X - disabled, I – invalid
```

Nastavení DHCP serveru provedeme příkazem ip dhcp-server> setup, který nás provede konfigurací.

```
[admin@MikroTik] /ip dhcp-server> setup
Select interface to run DHCP server on

dhcp server interface: eth3
```

```
Select network for DHCP addresses

dhcp address space: 192.168.0.0/24
Select gateway for given network

gateway for dhcp network: 192.168.0.1
Select pool of ip addresses given out by DHCP server

addresses to give out: 192.168.0.2-192.168.0.254
Select DNS servers

dns servers: 8.8.8.8
Select lease time

lease time: 3d
```

V případě RB 133 je nutné zadat interface wlan1.

Konfigurace DHCP serveru je hotová. V prvním kroku jsme zvolili interface, na kterém DHCP server bude provozován. Poté jsme zvolili síť 192.168.0.0/24, výchozí bránu 192.168.0.1, rozsah přidělovaných adres (ip pool), DNS server 8.8.8.8 a dobu zapůjčení adresy IP.

Pomocí výpisu lze zkontrolovat nastavení.

```
[admin@MikroTik] /ip dhcp-server> print
Flags: X - disabled, I - invalid
 #   NAME     INTERFACE     RELAY           ADDRESS-POOL     LEASE-TIME ADD-ARP
 1   dhcp2    eth3                          dhcp_pool1       3d
```

5.3.2 Konfigurace RADIUS klienta

Pro komunikaci RADIUS klienta s RADIUS serverem je nutné nakonfigurovat RouterBOARD. Klient Radius umožňuje autentizace několik služeb jako je PPP, HotSpot, DHCP, login, wireless. Můžeme klienta využít i pro samotné přihlášení k RouterBOARDu, kdy uživatelské jméno a heslo se ověřuje vůči Radiusu. Konfigurace klienta je popsána níže.

Přejdeme v konzoli do úrovně Radiusu pomocí příkazu *radius*. Příkazem *print* vypisuje stávající nakonfigurované klienty.

```
[admin@MikroTik] > radius
[admin@MikroTik] /radius> print
Flags: X - disabled
 #   SERVICE   CAL...   DOMAIN   ADDRESS   SECRET
```

Z výpisu je zřejmé, že žádný klient není nakonfigurován. Ke konfiguraci využijeme příkaz *add*.

```
[admin@MikroTik] /radius> add address=192.168.0.253 secret=testmikrotik
service=dhcp,wireless,ppp
```

Tímto příkazem jsem přidal nový Radius server s IP adresou 192.168.0.253 a sdíleným tajemství „testmikrotik“. Také jsem aktivoval služby DHCP, wireless a PPP, které budou s Radius serverem komunikovat.

K ověření konfigurace doporučuji využít příkaz *print* nebo pro detailní výpis *print detail*.

```
[admin@MikroTik] /radius> print
Flags: X - disabled
 #   SERVICE   CAL... DOMAIN ADDRESS                         SECRET
1    wire...                 192.168.0.1                        mik...
     dhcp
```

5.4 Konfigurace RADIUS Authorization

Zařízení (RouterBOARDy) od firmy Mikrotik podporují autentizaci 802.1x pouze na WLAN rozhraní. Tuto informaci jsem zjistil až v průběhu praktické části. Byla mi potvrzena i technickou podporou mikrotik.cz. Z tohoto důvodu popíši na RB750 přidělování IP adres na základě MAC adresy a na zařízení RB 133 ukážu konfiguraci WLAN interface s autentizaci vůči RADIUS serveru (802.1x).

Tento problém lze vyřešit PPP protokolem, přesněji PPPoE protokolem s ověřováním vůči RADIUS serveru. Konfiguraci provádět nebudu. Není to tématem práce.

5.4.1 Wireless

Pokud je RADIUS klient a DHCP server připravený. Můžeme přejít ke konfiguraci

Ideální řešení pro ISP by bylo využít dva RB 133, jako je znázorněno na obrázku 24. Jeden RouterBOARD by sloužil jako AP (Access Point) a druhý jako klient, který by poskytoval připojení pro uživatele.

Obrázek 24 Konfigurace AP a klienta

Konfigurace AP je popsána níže. Klient by byl nakonfigurován jako „station“ s využitím EAP-TLS. K dispozici mám pouze jeden RouterBOARD RB133, který bude sloužit jako AP. Z toho důvodu konfiguraci klienta RB133 nepopisuje a odkazuji na dokumentaci (Mikrotik, 2014), Popsána konfigurace využívá pouze jeden RouterBOARD RB133 jako AP.

Obrázek 25 Konfigurace AP

Je nutné nastavit wireless interface do režimu „ap-bridge“. Také nastavíme SSID na Mikrotik. Pro detailní konfiguraci odkazuji na dokumentaci (Mikrotik, 2014)

```
/interface wireless set wlan1 ssid=MikroTik mode=ap-bridge
```

Déle je potřeba vytvořit nový bezpečnostní profil, který dané nastavení bude uplatňovat. Poté je tento profil aktivován na specifickém bezdrátovém interface. Pro toto nastavení doporučuji použít nástroj Winbox. Příkaz je velmi dlouhý a nepřehledný.

```
[admin@MikroTik] /interface wireless> security-profiles add name="overeni"
mode=dynamic-keys authentication-types=wpa-eap,wpa2-eap unicast-ciphers=aes-ccm
group-ciphers=aes-ccm wpa-pre-shared-key="" wpa2-pre-shared-key="" supplicant-
identity="" eap-methods=passthrough tls-mode=no-certificates tls-certificate=none
mschapv2-username="" mschapv2-password="" radius-mac-authentication=yes radius-
mac-accounting=no radius-eap-accounting=no interim-update=0s radius-mac-
format=XX:XX:XX:XX:XX:XX radius-mac-mode=as-username radius-mac-caching=disabled
group-key-update=5m management-protection=allowed management-protection-key=""
```

Popíši pouze nejdůležitější vlastnosti:

- `security-profiles add name="overeni"` – nastaví jméno profilu,
- `mode=dynamic-keys` – WPA mod,
- `authentication-types=wpa-eap,wpa2-eap` – podporovaný autentizační typ,
- `eap-methods=passthrough` – AP bude předávat proces ověření na RADIUS sever,
- `radius-mac-authentication=yes` – tato vlastnost ovlivňuje způsob, jak AP bude zpracovávat klienty, kteří nejsou uvedeny v přístupovém seznamu. RADIUS server ověřuje MAC adresu klienta,
- `radius-mac-format=XX:XX:XX:XX:XX:XX` – formát v jakém bude MAC adresa posílána,
- `radius-mac-mode=as-username` – MAC adresa bude posílána v User-Name.

Popis ostatních vlastností je uveden v dokumentaci (Mikrotik, 2014)

Po vytvoření profilu stačí profil aktivovat na daném bezdrátovém interface.

```
[admin@MikroTik] /interface wireless> set wlan1 security-profile=overeni
```

5.4.2 DHCP server

Pro zajištění lepší bezpečnosti můžeme využit přidělování IP adres na základě ověření MAC adres vůči Radius serveru. To provedeme následujícím příkazem.

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

Pro ověření změn je možné si vypsat seznam DHCP serverů.

```
[admin@MikroTik] /ip dhcp-server> print detail
Flags: X - disabled, I - invalid
 3   name="dhcp2"… use-radius=yes
```

Toto nastavení je vhodné pro stále klienty. Z pohledu ISP je to užitečné řešení. Dokonce někteří ISP toto řešení používají jako jedinou autentizační metodu. S tím rozdílem, že pro každou MAC adresu mají jedinečnou IP adresu.

5.5 Zabezpečení

Zabezpečení směrovačů proti neautorizovaným pokusům o administraci je velmi důležité a je častou otázkou majitelů Mikrotik RouterOS. Pokusím se představit základní pokynu pro zajištění optimální míry zabezpečení.

Nejvyužívanějším prostředkem pro administraci je grafický nástroj WinBox. Pokud máme nainstalovaný balíček Security, je samotná komunikace mezi WinBox a RouterOS kryptována. Připojení k serveru probíhá na portu 80, kde běží webová služba. Další komunikace běží na portu 3987.

Další možnosti administrace jsou telnet a SSH. Telnet doporučuji nepoužívat, neboť se jedná o nezabezpečený přenos. SSH je bezpečný a šifrovaný.

Pro přenos souborů využívá RouterOS služby FTP a SCP. FTP také nedoporučuji ze stejných důvodů jako telnet. Pro přenos souborů pomocí SCP můžeme použít například WinSCP.

Další důležitou součástí jsou dostatečně silné hesla a jejich pravidelná změna.

Na RouterOS standardně běží služby: ftp, hotspot, hotspot-ssl, ssh, telnet, www. Příkazem *ip service* zobrazí služby.

```
[admin@MikroTik] /ip service> print
Flags: X - disabled, I - invalid
 #   NAME      PORT ADDRESS                         CERTIFICATE
 0 X telnet      23
 1 X ftp         21
 2   www         80
 3   ssh         22
 4 X www-ssl    443                                  none
 5 X api       8728
 6   winbox    8291
 7   api-ssl   8729                                  none
```

Z hlediska bezpečnosti je nevhodné, aby na RouterOS běžely nezabezpečené služby jako je ftp a telnet a také služby, které se nepoužívají. Ostatní služby je vhodné přesunout na jiné porty.

Volby portů se provádějí menu /ip service, kde lze také zadat, z jakých IP adres budou tyto služby přístupné.

```
[admin@MikroTik] /ip service> set ssh port=2121 address=192.168.0.1
```

Pro zadání více adres nebo subnetů, využijeme pravidla ve firewallu.

```
[admin@MikroTik] /ip firewall filter> add src-address=10.10.10.0/24 dst-
address=:2222 protocol=tcp action=accept disabled=no

[admin@MikroTik] /ip firewall filter> add dst-address=:2222 protocol=tcp
action=drop disabled=no
```

5.6 Testování

Pro testovací účely vytvoříme nového uživatele a přidáme MAC adresu.

Přihlásíme se do MySQL a použijeme databázi radius

```
mysql -u radius -p
mysql> use radius
```

Vytvoříme uživatele morce.

```
mysql> INSERT INTO radcheck(username, attribute, op, value) VALUES ("morce","User-Password","==","heslo");
```

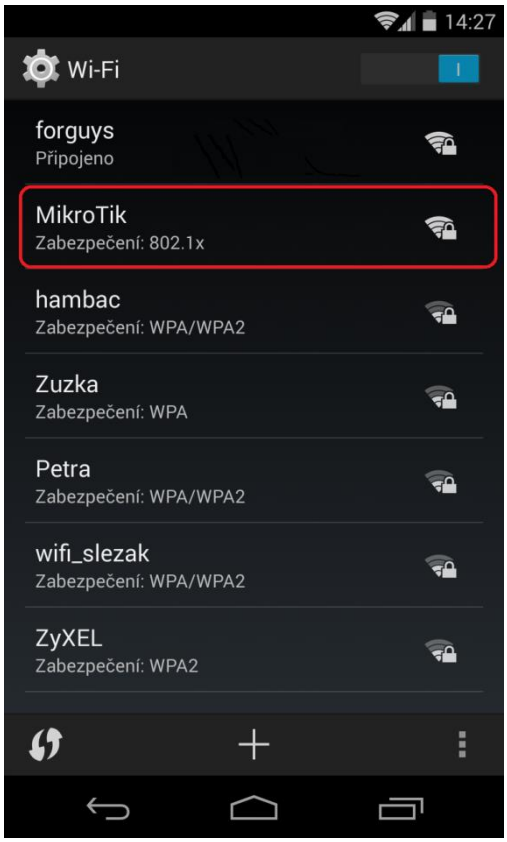
Přidáme záznam MAC adresy.

```
mysql> INSERT INTO radcheck(username, attribute, op, value) VALUES ("68:5D:43:89:71:E6","Auth-Type",":=","Accept");
```

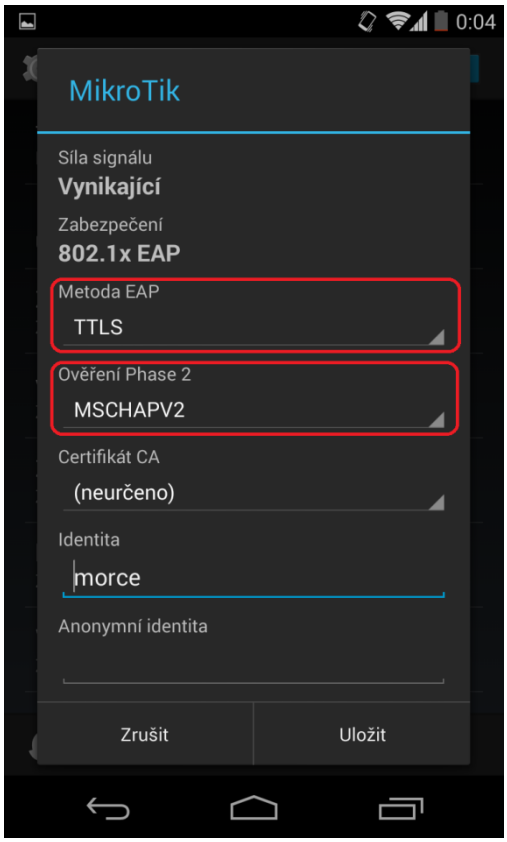
Příkazem `mysql> SELECT * FROM` radcheck vypíšeme tabulku.

```
+----+-------------------+--------------------+----+-----------+
| id | username          | attribute          | op | value     |
+----+-------------------+--------------------+----+-----------+
| 10 | morce             | User-Password      | == | heslo     |
| 11 | 68:5D:43:89:71:E6 | Auth-Type          | := | Accept    |
+----+-------------------+--------------------+----+-----------+
```

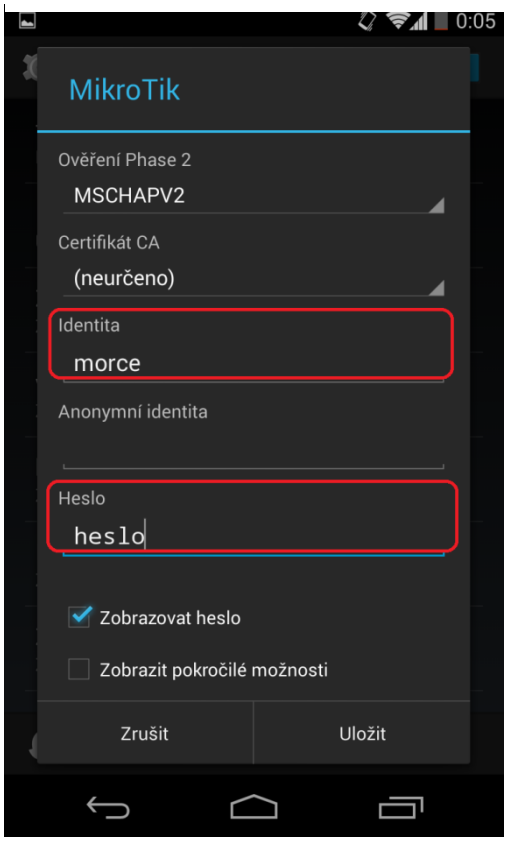
Testování ukáži na zařízení LG Nexus 4. V praxi by se použil RouterBOARD s wireless interface.



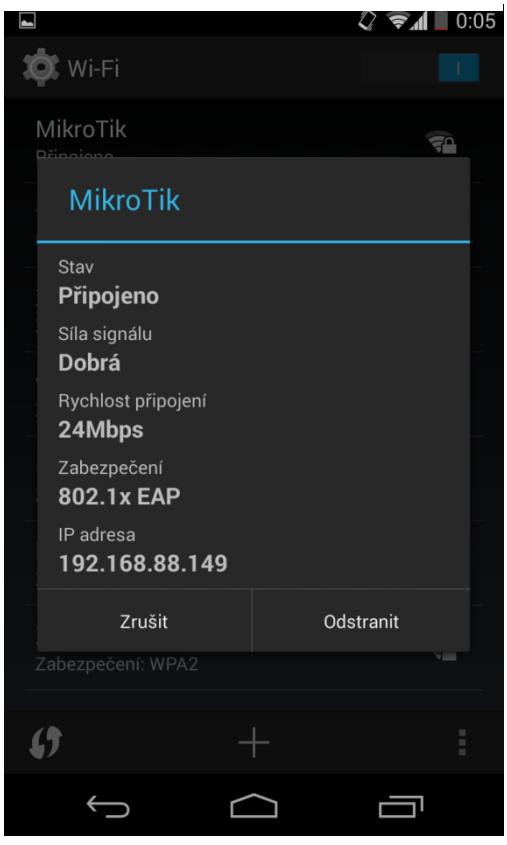
Obrázek 26 Výpis wireless



Obrázek 27 Metoda EAP a ověření



Obrázek 28 Nastavení jména a hesla



Obrázek 29 Kontrola připojení

Konfigurace je poměrně jednoduchá záležitost. Na obrázku 27 je ukázka nastavení EAP metody EAP-TTLS a ověření MS-CHAPv2. Na obrázku 28 je zadané uživatelské jméno a heslo. Na posledním obrázku 29 je stav připojení.

Jako první se ověří MAC adresa zařízení. Až po té následuje samotné ověření 802.1x, kdy TTLS vytvoří šifrovaný tunel a proběhne ověření pomocí MS-CHAPv2. To znamená, že uživatel, respektive zařízení musí mít správnou MAC adresu a znát uživatelské jméno a heslo.

Průběh autentizace si můžeme ověřit v tabulce radpostauth.

```
mysql> SELECT * FROM radpostauth;
+----+-------------------+------+--------------+---------------------+
| id | username          | pass | reply        | authdate            |
+----+-------------------+------+--------------+---------------------+
| 26 | 58:A2:B5:CF:C0:EA |      | Access-Accept | 2014-04-28 21:59:55 |
| 27 | morce             |      | Access-Accept | 2014-04-28 21:59:56 |
+----+-------------------+------+--------------+---------------------+
```

6 Závěr

Celá práce se skládá ze dvou hlavních částí. V teoretické části je popsán protokol 802.1x a jeho principy fungování. Velmi důležitou součástí protokolu 802.1x je protokol EAP, kterému je věnována patřičná část pro pochopení. Zaměřil jsem se i na starší protokol PPP, který se dodnes používá na sériových linkách. Okrajově je představen protokol RADIUS, Mikrotik a ISP. Praktická část obsahuje samotnou implementaci.

Popis protokolů nezabíhá do úplných detailů. Snažil jsem se vystihnout podstatné věci a popsat je co nejsrozumitelněji. Zájemcům, kteří by měli zájem o detailnější popis protokolů, doporučuji nastudovat jednotlivé RFC dokumenty.

Protokol 802.1x byl původně určen pro řízení přístupu k drátovým sítím a implementaci v přepínačích. V posledních letech je protokol 802.1x využíván hlavně v bezdrátových sítích, kde je řízení přístupu mnohem problematičtější. Protokol 802.1x je velmi silný nástroj pro řízení přístupu k sítím. Jeho nevýhodou je náročnější implementace. Nutnost pořídit a správně nakonfigurovat RADIUS server mluví za vše. Z toho důvodu nedoporučuji nasazení v domácím prostředí. Hlavní uplatnění najde ve firemním sektoru, kde je na bezpečnost kladen mnohem větší důraz.

Protokol 802.1x by mohl najít uplatnění i u ISP. Z vlastní zkušenosti mohu říct, že většina ISP poskytující bezdrátové připojení neklade na autentizaci a samotné zabezpečení spojení velký důraz. Vím o několika ISP, kde využívají pouze ověření na základě MAC adresy bez jakéhokoliv šifrování. To znamená, že kdokoliv může komunikaci odposlouchávat a MAC adresu snadno zachytit. Pro útočníka pak není problém se do sítě připojit.

Praktická část je věnována samotné implementaci protokolu 802.1x. Popisuje konfiguraci RADIUS serveru, RouterBOARD a klienta. V konfiguraci jsem se snažil postupovat systematicky a v rámci svých možností. Bohužel jsem měl k dispozici pouze jeden RouterBOARD s wireless interface, proto jsem konfiguraci musel přizpůsobit a jako klienta využít mobilní zařízení. Začal jsem konfigurací FreeRADIUS serveru s připojením k databázi

MySQL pro přehlednější administraci. Po konfiguraci FreeRADIUS serveru je popsána konfigurace RouterBOARDU a samotného klienta. Nakonec je celá implementace otestována. Samotný protokol 802.1x kombinuji s ověřením MAC adresy. Nejprve se ověří MAC adresa zařízení, až poté probíhá samotné ověření pomocí 802.1x. Pro autentizaci jsem zvolil protokol EAP-TTLS, který vytvoří šifrovaný tunel pro ověření MS-CHAPv2. To znamená, že klient musí disponovat správnou MAC adresou a kombinací uživatelského jména a hesla.

V praktické části jsem narazil na několik problémů, které jsem musel řešit. Původní myšlenka byla provést konfiguraci na drátových interface. Zde jsem narazil na omezení RouterOS. Po osobní konzultaci s podporou Mikrotik jsem zjistil, že drátový interface 802.1x nepodporuje. To byl pro mě zásadní problém, protože jsem měl k dispozici pouze RouterBOARD s drátovým interface. Jsem zvyklý pracovat na Cisco zařízení, kde tato možnost lze a nepočítal jsem s tím, že právě zde narazím. Z toho důvodu jsem byl nucen konfiguraci provést na bezdrátovém interface. Cílem práce není porovnat Cisco s Mikrotikem, ale musím říci, že v tomto ohledu má Cisco velikou výhodu.

Má práce by mohla být přínosná pro všechny poskytovatele bezdrátových sítí, nejen ISP, kteří se snaží o lepší zabezpečení a centralizovanou zprávu řízení přístupu k síti.

Literatura

1. ABOBA, B. a BLUNK, L., 2004. *RFC 3748: Extensible Authentication Protocol (EAP)* [online]. [cit. 2014-03-31]. Dostupné z: http://tools.ietf.org/html/rfc3748

2. BERSANI, F. a TSCHOFENIG, H., 2007. *RFC 4764: A Pre-Shared Key Extensible Authentication Protocol (EAP) Method* [online]. [cit. 2014-03-31]. Dostupné z: https://tools.ietf.org/html/rfc4764

3. DOSTÁLEK, Libor, 2003. *Velký průvodce protokoly TCP/IP: bezpečnost.* 2. aktualiz. vyd. Praha: Computer Press, xvi, 571 s. ISBN 80-722-6849-X.

4. FUNK, P. a BLAKE-WILSON, S., 2008. *RFC 5281: Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol* [online]. [cit. 2014-03-31]. Dostupné z: https://tools.ietf.org/html/rfc5281

5. LLOYD, B. a SIMPSON W., 1993. *RFC 1334: PPP Authentication Protocols* [online]. [cit. 2014-03-31]. Dostupné z: https://tools.ietf.org/rfc/rfc1334

6. ODOM, Wendell, Rus HEALY a Naren MEHTA, 2009 *Směrování a přepínání sítí: autorizovaný výukový průvodce*. Vyd. 1. Brno: Computer Press, s. 631-632. ISBN 978-80-251-2520-5.

7. PUŽMANOVÁ, Rita, 2004. *Širokopásmový Internet: přístupové a domácí sítě*. 1. vyd. Brno: Computer Press, 377 s. ISBN 80-251-0139-8.

8. RIGNEY, C. a WILLIENS, S., 2000. *RFC 2865: Remote Authentication Dial In User Service (RADIUS*[online]. [cit. 2014-03-31]. Dostupné z: https://tools.ietf.org/html/rfc2865

9. SIMON, D. a ADOBA, B., 2008. *RFC 5216: The EAP-TLS Authentication Protocol* [online]. [cit. 2014-03-31]. Dostupné z: http://tools.ietf.org/html/rfc5216

10. SIMPSON, W., 1994. *RFC 1661: The Point-to-Point Protocol (PPP)* [online]. [cit. 2014-03-31]. Dostupné z: http://tools.ietf.org/rfc/rfc1661

11. SIMPSON, W., 1996. *RFC 1994: PPP Challenge Handshake Authentication Protocol* [online]. [cit. 2014-03-31]. Dostupné z: http://tools.ietf.org/rfc/rfc1994

12. ZANDL, Patrick, 2003. *Bezdrátové sítě WiFi: praktický průvodce*. Vyd. 1. Brno: Computer Press, 190 s. ISBN 80-722-6632-2.

13. EAP Methods, 2014. *FreeRADIUS* [online]. [cit. 2014-05-02]. Dostupné z: http://freeradius.org/features/eap.html

14. FreeRADIUS Documentation, 2014. *FreeRADIUS* [online].[cit. 2014-05-02]. Dostupné z: http://freeradius.org/doc/

15. IEEE 802.1X Port-Based Authentication, 2012. *Cisco* [online]. [cit. 2014-03-10]. Dostupné z: http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst6500/ios/12-2SX/configuration/guide/book/dot1x.html

16. Konfigurace Xsupplicantu pro připojení k eduroamu, 2011. *Eduroam* [online]. [cit. 2014-03-10]. Dostupné z: https://www.eduroam.cz/cs/uzivatel/sw/nix/xsupplicant

17. Manual:Interface/Wireless, 2014. *Mikrotik* [online]. [cit. 2014-03-31]. Dostupné z: http://wiki.mikrotik.com/wiki/Manual:Interface/Wireless

18. Manual Licence, 2013. *Mikrotik.* [online]. [cit. 2014-03-31]. Dostupné z: http://wiki.mikrotik.com/wiki/Manual:License#License_Levels

19. *Mikrotik, 2014* [online].[cit. 2014-03-31]. Dostupné z: http://www.mikrotik.com/

20. Mikrotik RB450G, 2014. *Mikrotik* [online]. [cit. 2014-03-31]. Dostupné z: http://www.mikrotik.cz/RouterBoardy/RB450G-256-MB-RAM-680-MHz-5x-Gbit-LAN-vc-L5.html

21. Mikrotik RouterOS, 2010. *Mikrotik.* [online]. [cit. 2014-03-31]. Dostupné z: http://www.mikrotik.com/pdf/what_is_routeros.pdf

22. RADIUS client, 2007. *Mikrotik* [online]. [cit. 2014-04-07]. Dostupné z: http://www.mikrotik.com/testdocs/ros/2.9/guide/aaa_radius.php

23. RASPBERRY Pi Model B, 2014. *Alza.cz* [online]. [cit. 2014-04-07]. Dostupné z: http://www.alza.cz/raspberry-pi-model-b-d404121.htm

24. WHAT IS A RASPBERRY PI?, 2013. *RASPBERRY PI FOUNDATION* [online]. [cit. 2014-04-07]. Dostupné z: http://www.raspberrypi.org/help/what-is-a-raspberry-pi/

25. *802.1X-2010: IEEE standard for local and metropolitan area networks port-based network access control*, 2010. [online]. New York: Institute of Electrical and Electronics Engineers, [cit. 2014-05-01]. ISBN 978-073-8161-457. Dostupné z: http://standards.ieee.org/findstds/standard/802.1X-2010.html

Příloha A: Instalace MySQL a phpMyAdmin

Instalaci databázi MySQL se provede příkazem:

```
apt-get install mysql-server
```

V průběhu instalace budeme vyzváni k zadání root hesla.

K instalaci phpMyAdmin je potřeba naistalovat webserver. Jako webserver jsem vybral Apache2. Apache2 a phpMyAdmin se naistaluje následovně.

```
apt-get install apache2 phpmyadmin
```

Příloha B: Popis tabulek MySQL

Databáze MySQL nahrazuje klasické textové soubory. Umožňuje snadnější a přehlední práci s daty. Vzorové tabulky jsou umístěný v adresáři /etc/freeradius/sql/mysql/. Sedm tabulek je importovaných ze souboru schema.sql a jedna tabulka ze souboru nas.sql.

Výpis tabulek provedeme příkazem „*show tables*“.

```
mysql> show tables;
+------------------------+
| Tables_in_radius |
+------------------------+
| nas |
| radacct |
| radcheck |
| radgroupcheck |
| radgroupreply |
| radpostauth |
| radreply |
| radusergroup |
+------------------------+
mysql>
```

Velmi důležitou tabulkou je nas, která uchovává informace o jednotlivých NAS serverech (RADIUS klientech). Pokud chceme tuto tabulku využívat, je nutné v soubor sql.conf odkomentovat parametr „readclients = yes“. Tabulka obsahuje osm sloupců.

- id – pouze informativní. Slouží k řazení záznamů.
- nasname – slouží pro vkládání IP adres NAS.
- Shortname – uchovává informaci o názvu NAS serveru.
- type – říká RADIUS serveru, že klient může obsahovat specifické atributy, podle typu NAS serveru.
- ports – informativní sloupec, který udává počet dostupných portů NAS.
- secret – obsahuje informaci o sdíleném tajemství (heslu) mezi klientem a serverem RADIUS.

- community- název komunity SNMP. V našem případě obsahuje NULL.
- description- pouze informativní sloupec s popisem.

Tabulky radchech, radgroupcheck, radreply, radgroupreply mají téměř shodnou strukturu. Obsahují pět sloupců.

- id – tento sloupec je informativní. FreeRADIUS ho využívá k řazení výsledků dotazů.
- UserName / GroupName – obsahuje uživatelské / skupinové jméno.
- Attribute – zde se vkládají atributy. Například Password, Cleartext-Password.
- op – operátory ==, :=, =.
- Value – obsahuje hodnotu atributu. Například heslo, Accept nebo prázdnou hodnotu, která se využívá pro autentizaci vůči MAC adrese.

Poslední důležitou tabulkou je radusergroup, která přiřazuje uživatele do skupiny. Má velmi jednoduchou strukturu a obsahuje tři sloupce.

- UserName – uživatelské jméno.
- GroupName – jméno skupiny do které je uživatel přiřazen.
- Priority – využívá se v situaci, kde je uživatel ve více skupinách. Podle této hodnoty se server rozhodne. Jako první se uplatňují skupiny s nejnižší prioritou.

Tabulka radpostauth je pouze informativní. Obsahuje informace o autentizaci.

```
mysql> SELECT * FROM radpostauth ;
+----+------------------+-------+--------------+---------------------+
| id | username         | pass  | reply        | authdate            |
+----+------------------+-------+--------------+---------------------+
| 66 | sqltest          | test  | Access-Reject | 2014-04-21 20:17:38 |
| 67 | sqltest          | heslo | Access-Accept | 2014-04-21 20:18:05 |
+----+------------------+-------+--------------+---------------------+
```

Tabulka se skládá z pěti sloupců:

- id – pouze informativní,
- username – zde se ukládá uživatelské jméno,
- pass – informace o typu přihlášení
- reply – obsahuje informaci o typu paketu přihlášení, například Access-Accept,
- authdate – datum a čas přihlášení.

Tabulka radacct je určena pro RADIUS Accountig a je pouze informativní. Obsahuje 25 sloupců. Tato tabulka nebyla v práci použita, a proto není popisována.

Příloha C: Výpis RouterBOARD

```
[admin@MikroTik] /interface wireless> security-profiles print
 0 name="default" mode=none authentication-types="" unicast-ciphers=aes-ccm
   group-ciphers=aes-ccm wpa-pre-shared-key="" wpa2-pre-shared-key=""
   supplicant-identity="MikroTik" eap-methods=passthrough
   tls-mode=no-certificates tls-certificate=none static-algo-0=none
   static-key-0="" static-algo-1=none static-key-1="" static-algo-2=none
   static-key-2="" static-algo-3=none static-key-3=""
   static-transmit-key=key-0 static-sta-private-algo=none
   static-sta-private-key="" radius-mac-authentication=no
   radius-mac-accounting=no radius-eap-accounting=no interim-update=0s
   radius-mac-format=XX:XX:XX:XX:XX:XX radius-mac-mode=as-username
   radius-mac-caching=disabled group-key-update=5m
   management-protection=disabled management-protection-key=""

2 name="eap" mode=dynamic-keys authentication-types=wpa-eap,wpa2-eap
   unicast-ciphers=aes-ccm group-ciphers=aes-ccm wpa-pre-shared-key=""
   wpa2-pre-shared-key="" supplicant-identity="Mikrotik"
   eap-methods=passthrough tls-mode=dont-verify-certificate
   tls-certificate=none static-algo-0=none static-key-0="" static-algo-1=none
   static-key-1="" static-algo-2=none static-key-2="" static-algo-3=none
   static-key-3="" static-transmit-key=key-0 static-sta-private-algo=none
   static-sta-private-key="" radius-mac-authentication=yes
   radius-mac-accounting=no radius-eap-accounting=no interim-update=0s
   radius-mac-format=XX:XX:XX:XX:XX:XX radius-mac-mode=as-username
   radius-mac-caching=disabled group-key-update=5m
   management-protection=allowed management-protection-key=""

[admin@MikroTik] /interface wireless> print
Flags: X - disabled, R - running
 0    name="wlan1" mtu=1500 mac-address=00:0C:42:64:05:75 arp=enabled
      interface-type=Atheros AR5413 mode=ap-bridge ssid="MikroTik"
      frequency=2462 band=2.4ghz-b/g scan-list=default antenna-mode=ant-a
      wds-mode=disabled wds-default-bridge=none wds-ignore-ssid=no
      default-authentication=no default-forwarding=no default-ap-tx-limit=0
      default-client-tx-limit=0 hide-ssid=no security-profile=eap
      compression=no
```

```
[admin@MikroTik] /ip> dhcp-server export
# jan/02/1970 13:06:24 by RouterOS 4.17
# software id = YDF0-122B
#
/ip dhcp-server
add address-pool=pool1 authoritative=after-2sec-delay bootp-support=static \
    disabled=yes interface=ether3 lease-time=3d name=dhcp1
add address-pool=pool1 authoritative=after-2sec-delay bootp-support=static \
    disabled=no interface=ether2 lease-time=3d name=server2
add address-pool=pool1 authoritative=after-2sec-delay bootp-support=static \
    disabled=no interface=wlan1 lease-time=3d name=dhcp2
/ip dhcp-server config
set store-leases-disk=5m
/ip dhcp-server lease
add address=192.168.0.253 comment="" disabled=no mac-address=\
    B8:27:EB:19:4C:55 server=server2
/ip dhcp-server network
add address=192.168.0.0/24 comment="" gateway=192.168.0.1

[admin@MikroTik] /radius> export
# jan/02/1970 13:07:14 by RouterOS 4.17
# software id = YDF0-122B
#
/radius
add accounting-backup=no accounting-port=1813 address=192.168.0.253 \
    authentication-port=1812 called-id="" comment="" disabled=no domain="" \
    realm="" secret=testmikrotik service=login,wireless,dhcp timeout=3s
/radius incoming
set accept=yes port=1700
```