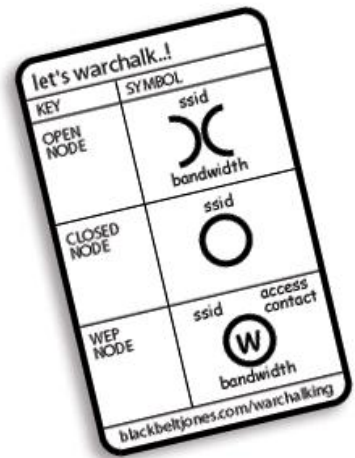




www.warchalking.org

Wireless Links Security



Poland MUM – Krakow – February, 2008

Eng. Wardner Maia
Brazil

Introduction

Name: Wardner Maia

Country: Brazil

- Electronic/Telecommunications Engineer
- Internet Service Provider since 1995
- Wireless Internet Service Provider since 2000
- Teaches Wireless for WISP's since 2002
- Mikrotik Certified Trainer since June, 2007
- Works as engineer to companies: MD Brasil and Rede Global Info

Introduction

MD Brasil Information Technology and Telecommunications

→ Wireless Internet Service Provider with about 3 thousand customers in Sao Paulo State

→ Mikrotik Distributor

→ Mikrotik Training Partner

www.mdbrasil.com.br / www.mikrotikbrasil.com.br

Rede Global Info (Global Info Network)

→ The biggest association of independent Internet Service Providers in Brazil (maybe one of the biggest in the world)

→ 684 small and medium WISP's covering 1300 cities

→ More than 6 Gbps aggregated bandwidth.

Our goal is to become a 100% secure network and of course a 100% Mikrotik Network powered ☺

www.redeglobalinfo.com.br

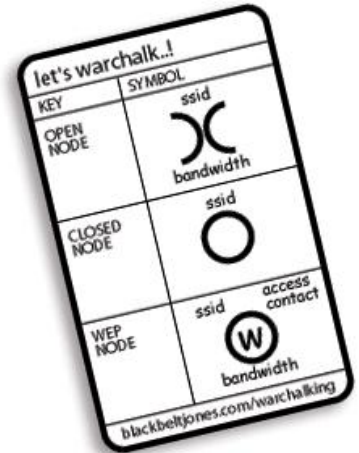


Why Wireless Security ?



- Wireless is the only solution for a lot of cities and rural areas not covered by traditional Telecom's Companies.
- Wireless is the easiest and fastest way to gain market share.
- With a good deployment performance can be as good as DSL and Cable
- WISP's are the real competitors for Telecom Companies.
- *Security is the Achilles heel for Wireless Networks based on Wi-Fi equipment.*

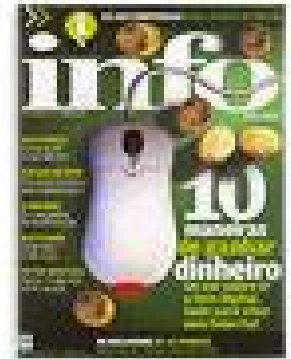
Objectives



- To give an overview of the theoretical concepts involved in 802.11 Wireless Links Security and how to use RouterOS to ensure security
- Critical Analysis of the actual adopted models by WISP's
- Layer 2 attacks, and the challenge of protecting against them.



“The power of the Potatoes”

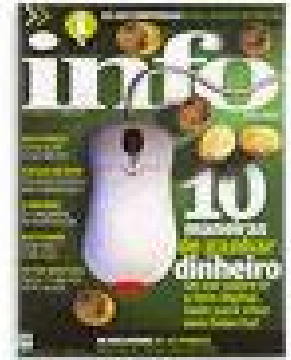


Among 43 Wireless Networks located at the most Important financial region in São Paulo, only 8 had “recommended” security configuration.

Specialized IT Magazine - Info Exame
News article published in 2002



“The power of the Potatoes”

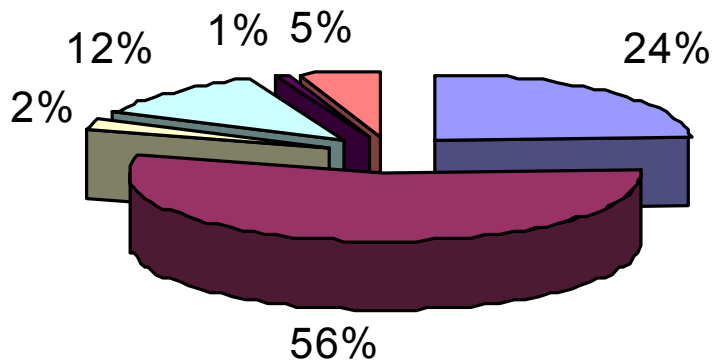


In 2002, according to the Author, the recommended security measures were:

- Hidden SSID (Network Name)
- MAC Acces Control Lists
- WEP

WISP's Security Methods in Brazil - 2002

Seguranca provedores 2002



■ Nenhuma Medida

■ Controle de MAC - ACL

■ Controle de MAC - Radius

■ Controle de MAC + IP

■ PPPoE

■ WEP

“Rudimentary” Security Measures (what is not real security)

1 – Hidden SSID

Access Points by default broadcasts its SSID in Beacons packets. This behavior can be modified in today available hardware to send null strings as SSID's or don't send anything.

It's a good measure to avoid a casual client to associate, but cannot be considered a real Security measure.

- SSID's are stored in clear text in Client's Machines.
- Passive Scanners can easily discover them by listen probe requests from authorized clients
- Some kind of old hardware has problem in associating with hidden SSID

“Rudimentary” Security Measures (what is not real security)

2 – MAC’s Access Lists

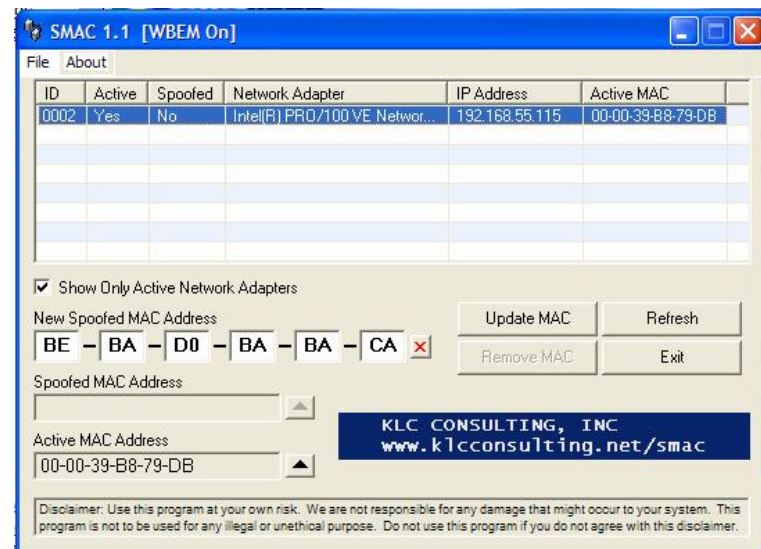
- Discover authorized MAC’s is possible with passive scanners
 - Airopeek for Windows
 - Kismet, Wellenreiter, etc for Linux
- Forgery MAC is trivial under Linux/BSD
and even under Windows

- FreeBSD :

```
ifconfig <interface> -L <MAC>
```

- Linux :

```
ifconfig <interface> hw ether <MAC>
```



“Rudimentary” Security Measures (what is not real security)

3 – WEP Encryption

→ “Wired Equivalent Privacy” – a non mandatory feature for securing 802.11 Wireless Lan's.

→ Based in a shared secret and generation of encryption keys with RC4 algorithm

→ 40 bit WEP can be cracked without any sophisticated technique – just dictionary attack in less than 24 hours !

→ 104 bit WEP in practice cannot be cracked by dictionary attacks

Compromising WEP (definitively)

1 – A Paper from UC Berkeley revealing WEP Weakness due to key reuse and inadequate message authentication.

Borisov, Nikita, Goldberg e Wagner

<http://www.isaac.cs.berkeley.edu/isaac/mobicom.pdf>

2 – A paper from the University of Michigan highlighting weaknesses in 802.11 access control mechanisms.

Arbaugh, Shankar e Wan

<http://www.cs.umd.edu/~waa/wireless.pdf>

3 – A paper published in Security Focus identifying weakness in WEP due to improper utilization of RC4 algorithm.

Fluhrer, Mantin e Shamir

http://downloads.securityfocus.com/library/rc4_ksaproc.pdf

Compromising WEP (definitively)

4 – A paper in 2005 published by Andrea Bittau describing a “fragmentation attack” enhancing the other inductive attacks – WEP cracked in less then 5 minutes !

<http://www.toorcon.org/2005/conference.html?id=3www.aircrack-ng.org/doku.php?id=fragmentation&DokuWiki=71f9be8def4d820c6a5a4ec475dc6127>

5 – Huge “support” at the Internet for cracking WEP

The FEDs can own your WLAN too

<http://www.tomsnetworking.com/Sections-article111.php>

How to crack WEP

<http://www.tomsnetworking.com/Sections-article118.php>

Breaking 104 bit WEP in less than 60 seconds

<http://eprint.iacr.org/2007/120.pdf>

Compromising WEP (definitively)

5 – Very good “support” to crack wep

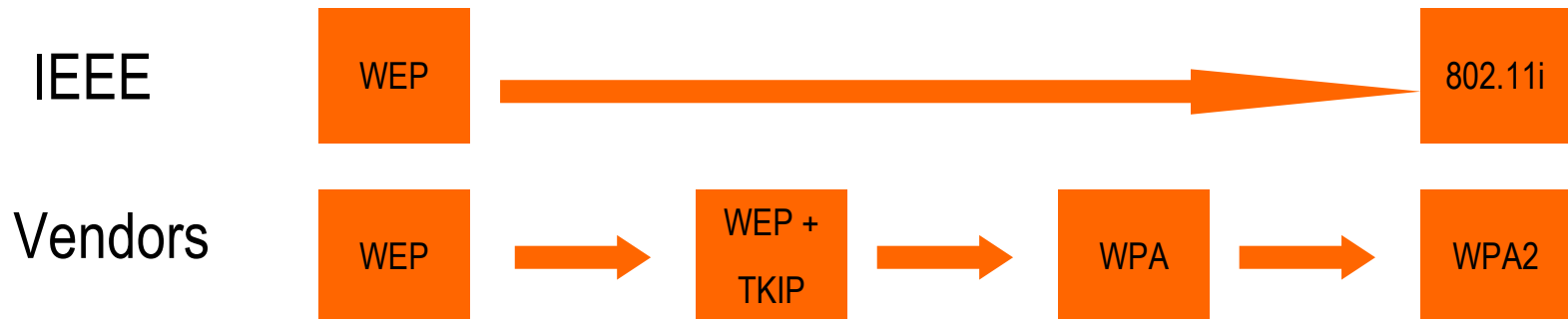
You Tube Vídeo

<http://www.youtube.com/watch?v=PmVtJ1r1pmc>



IEEE 802.11i

- In order to replace WEP, the IEEE has created a new Task Group – 802.11i
- Before the ratification of the amendment the Wi-Fi Alliance has created the WPA (Wireless Protected Access)
- The Amendment proposed by 802.11i task group was finally aproved in june of 2004.
- Wi-Fi Alliance created what it called WPA2, compatible with 802.11i
- To ensure interoperation, WPA2 should be interoperable with WPA



802.11i Goals

→Authentication

AP → Client: Prevent unauthorized network access

Client → AP: Make sure that the AP is not a rogue AP trying to hijack important data (man-in-the-middle attack)

→Confidentiality

→Use of encryption to ensure privacy of data

→Data Integrity

→Protect against modification or destruction of data

How 802.11i works

802.11i is composed by 3 entities:

- Supplicant
- Authenticator
- Authentication Server

And is made by a combination of other protocols:

- 802.1X – A Port Based Network Access Control
- EAP – Extensible Authentication Protocol
- RADIUS – Remote Access Dial In User Service

Authentication:

→ Home Mode:

→ Pre Shared Key (PSK)

→ Enterprise Mode:

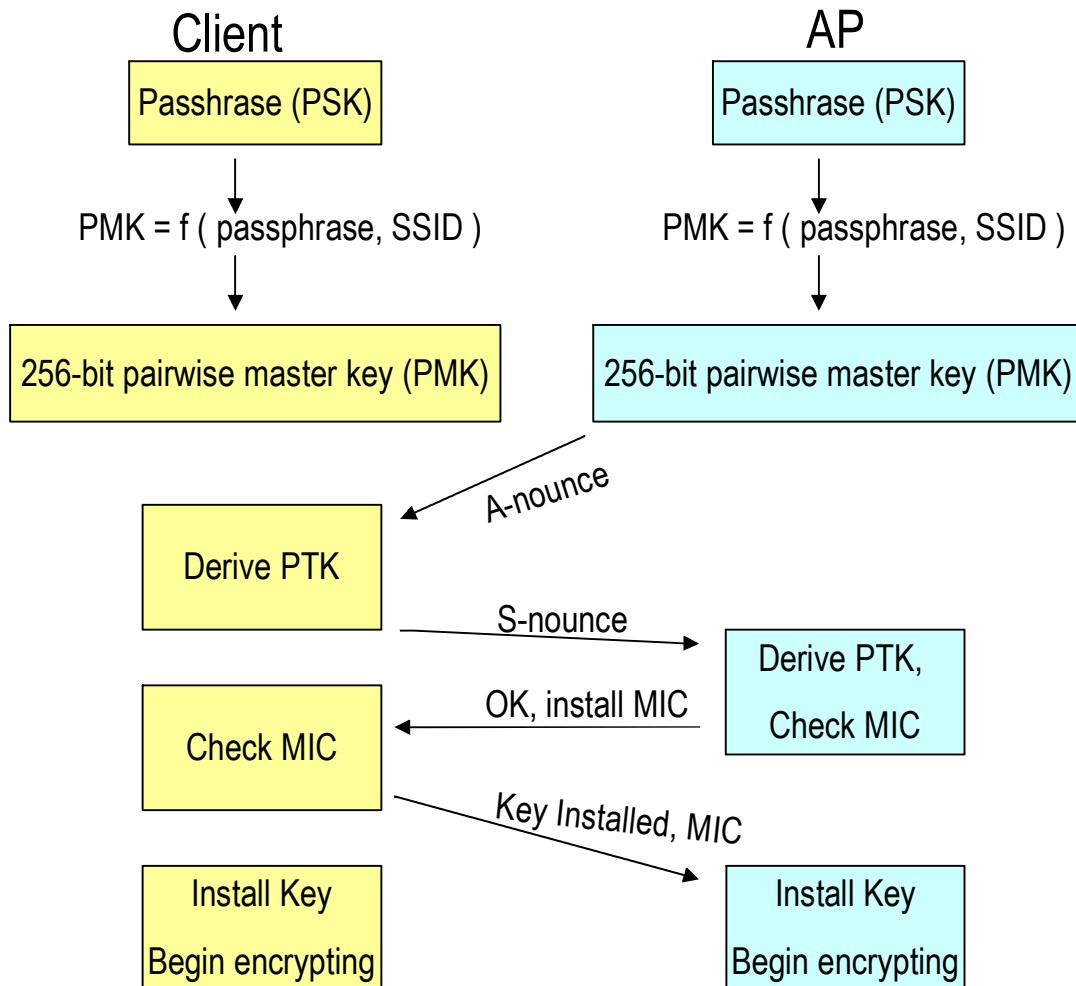
→ 802.1X/EAP

802.11i PSK

A Key called Pairwise Master Key (PMK) Is Created By hashing the Passphrase 4096 times and SSID is used too. It's Stored at Registry in Windows or supplicant.conf in Linux.

The Pairwise Transient Key is created dinamicaly after a exchange of ramdom numbers and are different always a client connects to the AP

MIC is used to ensure the client has the PMK



Confidentiality in 802.11i

Confidentiality

- After authentication, both sides – AP and the Client have a PMK – Pairwise Master Key that remains the same for all the session time.
- For the data transmission 802.11i will derive a PTK – Pairwise Transient Key that is a pseudo random number function of both sides and *exclusive for each client* – even if PSK is used.

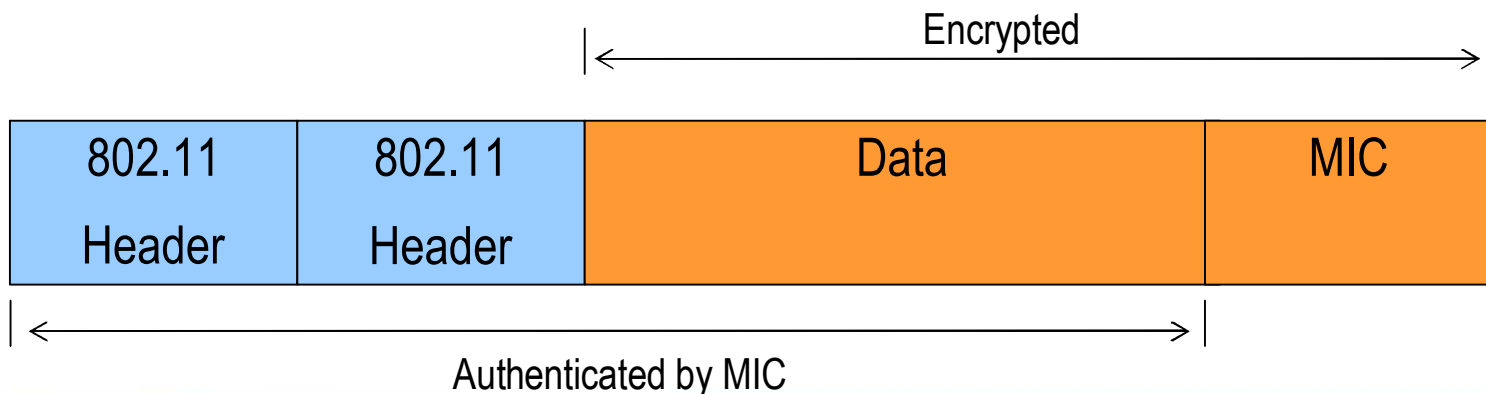
Integrity in 802.11i

One part of PTK is responsible to protect the integrity of the messages – is the Message Integrity Check (MIC). With the MIC, in every communication the Sender computes a hash of the data plus a secret key – the temporal integrity key.

MIC = hash (packet, temporal integrity key)

WPA uses TKIP → Hashing algorithm called “Michael”

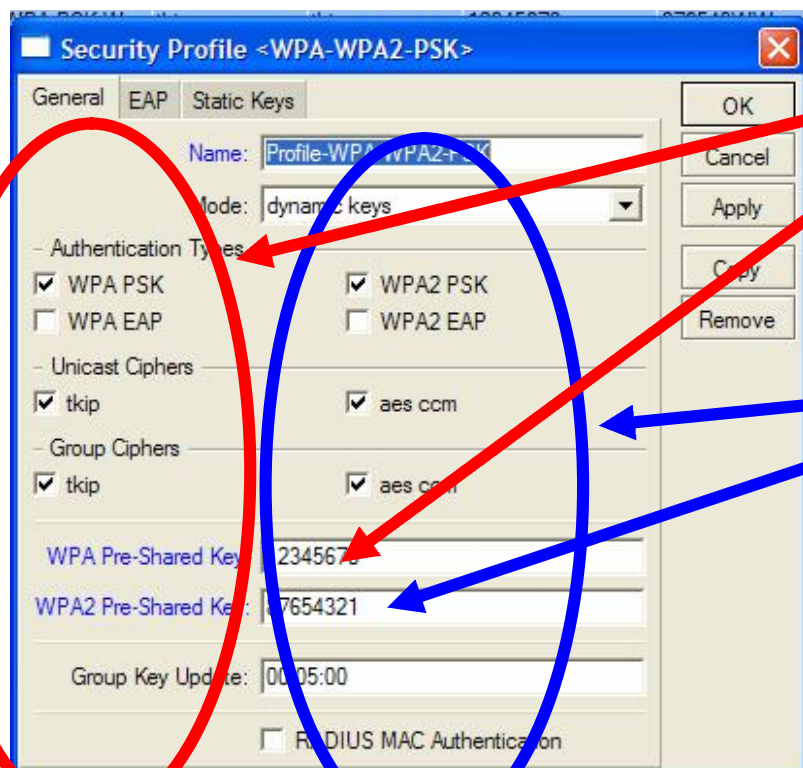
WPA2 uses CCMP → Cipher Block Chaining Message Authentication Check– CBC-MAC



Setup with WPA/WPA2 using PSK

Using WPA/WPA2 – PSK

It's very easy to configure WPA/WPA2 PSK con Mikrotik



→ WPA - PSK

choose dynamic keys, WPA-PSK, and the key (8 to 63 characters long)

→ WPA2 – PSK

choose dynamic keys, WPA2-PSK, and the key (8 to 63 characters long)

Group key update is the time to update the group key (there was a bug for versions older than 2.9.38)

How secure is WPA / WPA2 PSK ?

- The way to break WPA/WPA2 PSK is a dictionary attack.
- Because of the PMK is not just a hash of the PSK, but also of the SSID pre-computing hashes are ineffective (dictionary attacks based only in common words won't work)
- There is no difference in the difficulty to break WPA-PSK or WPA2-PSK, because they use the same hashing function to generate PMK. Only the MIC changes.
- Tools for breaking WPA/WPA2 – PSK Cowpatty
<http://sourceforge.net/projects/cowpatty>
- PSK biggest weakness is due to the fact that the key is present in clear text at customers computers (or radio equipment).

How secure is WPA / WPA2 PSK ?

When an attacker has the PSK, it's possible:

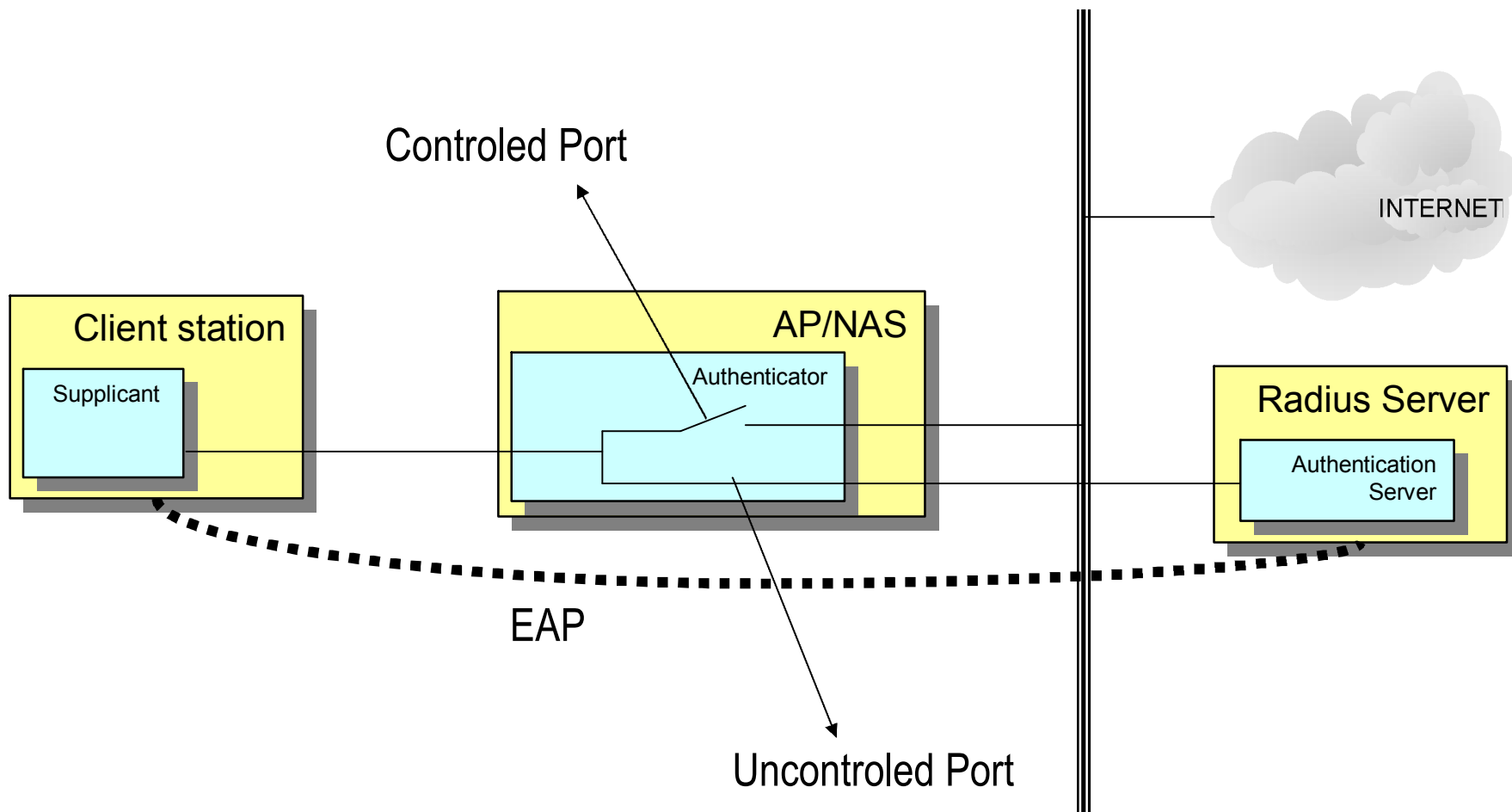
- To gain unauthorized access
- To impersonate an Access Point to launch the man-in-the middle attack

Recommendations to WISP's:

- Use PSK only if you're absolutely sure that the keys are protected (when you use radio equipments that nobody else has access)
- Don't forget that in Mikrotik v2.9 the keys appear in clear text in Winbox. And in V3, are the passwords really hidden ?

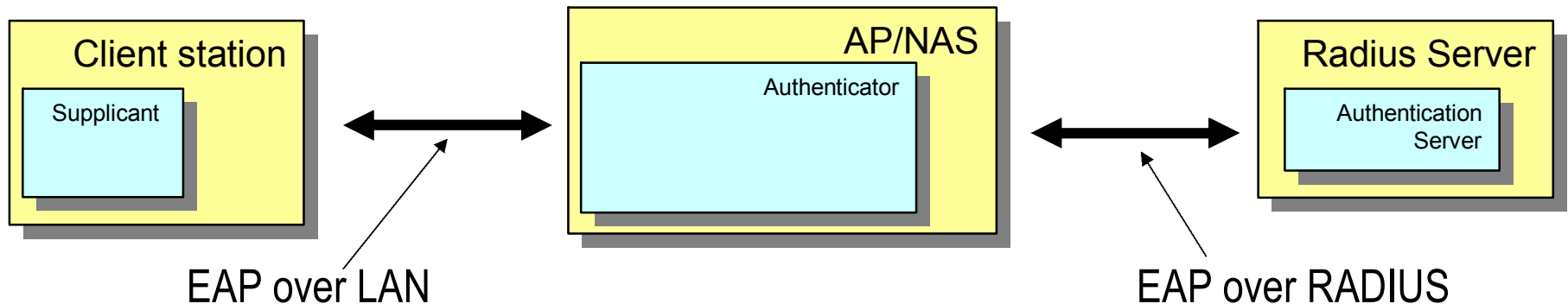
Setup in Enterprise Mode (802.1x + EAP)

Authentication via 802.1X



EAP

EAP is a general protocol first defined for PPP and used for host or user identification.



There are several types of EAP. The most common are: EAP TLS, EAP TTLS, PEAP and LEAP

Some types of EAP

LEAP: (Lightweight EAP)

It's Cisco's proprietary protocol developed before 802.11i and WPA. Based in a challenge-response schema with username/password. It can be ported to various clients but only works on Cisco AP's. There are 2 flavors of LEAP, before and after 802.11i

Requires username/password for clients. Doesn't require certificates Ensures 2 way authentication, but is vulnerable to dictionary attacks. The only way to provide some security with LEAP is a very strong password policy.

→ Tool to crack LEAP: Asleap - <http://asleap.sourceforge.net/>

Some types of EAP

PEAP: (Protected EAP) and EAP-TTLS (EAP tunneled TLS)

Both methods requires Certificates at the server and username/password for Clients. Authentication occurs in the following order:

- 1 - The server sends an EAP request identity as usual
- 2 - Once the identity (any) is sent, a TLS Tunnel is established
- 3 – Inside the Tunnel, the Client passes the real username and password

The problem with this methods is the man-in-the-middle attack and to avoid it the Clients must install the CA Certificate.

OBS: The difference between PEAP and TTLS is that TTLS is compatible with old EAP protocols, like LEAP

Some types of EAP

EAP-TLS (EAP – Transport Layer Security)

Provide the highest level of Security. Requires Certificates both in clients and Server.

- 1 – The server provides a Certificate to the Client
- 2 – The client sends his Certificate to the server
- 3 – If both sides validate themselves, a random number is generated and used to Create a dynamic PMK

This is the most secure method at all. It's only disadvantage of this method is that it requires extensive support for administering Certificates.

EAP types comparison

EAP Type	Open/ Proprietary	Mutual Auth	Authentication Credentials		User Name In Clear	Man in The middle Vulnerable
			Supplicant	Authenticator		
TLS	Open	Yes	Certificate	Certificate	NA	No
TTLS	Open	Yes*	Username/Pwd	Certificate	No	Maybe*
PEAP	Open	Yes*	Username/Pwd	Certificate	No	Maybe*
LEAP	Proprietary	No	Username/Pwd		Yes	Yes

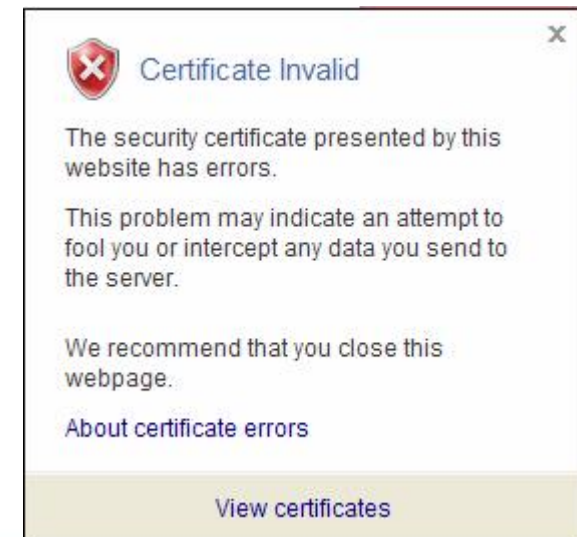
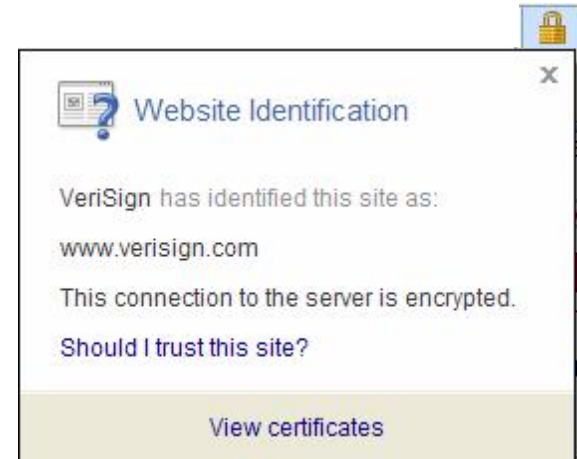
* Depends on client configuration

Deploying EAP-TLS with Certificates

A digital certificate is a file that uniquely identifies its owner. A certificate contains owner identity information and owner's public key. Certificates are created by entities called Certificate Authorities (CAs ')

Certificates can be :

- Signed by a "trusted" CA or
- Self signed Certificates



Security Profiles – TLS Mode

→ verify certificates

require a certificate and verify that it has been signed by the available CA certificate

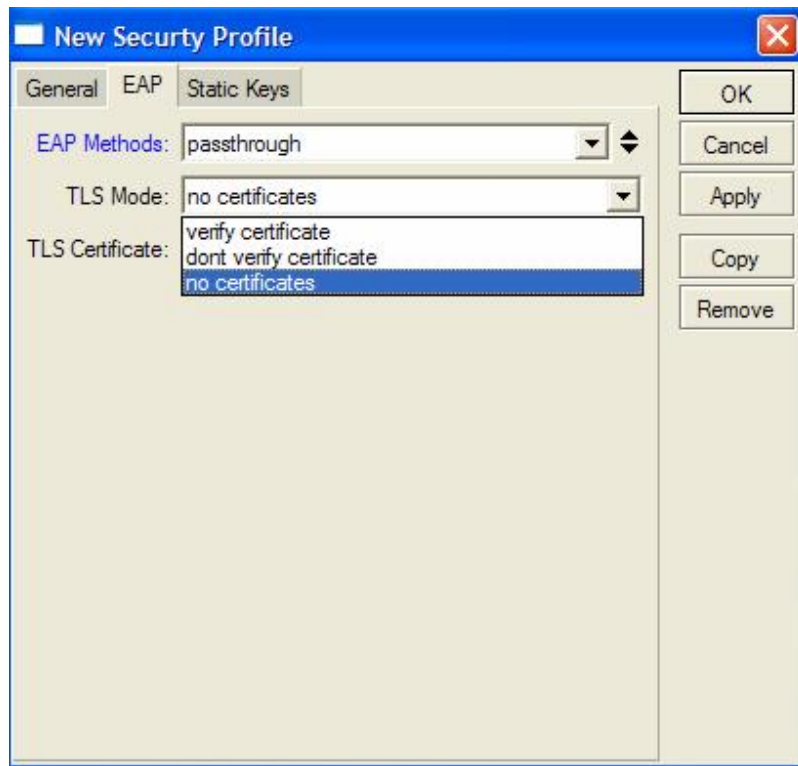
→ don't verify certificates

require a certificate, but don't check if it has been signed by the available CA certificate

→ no certificates

certificates are negotiated anonymously using Diffie-Hellman algorithm.

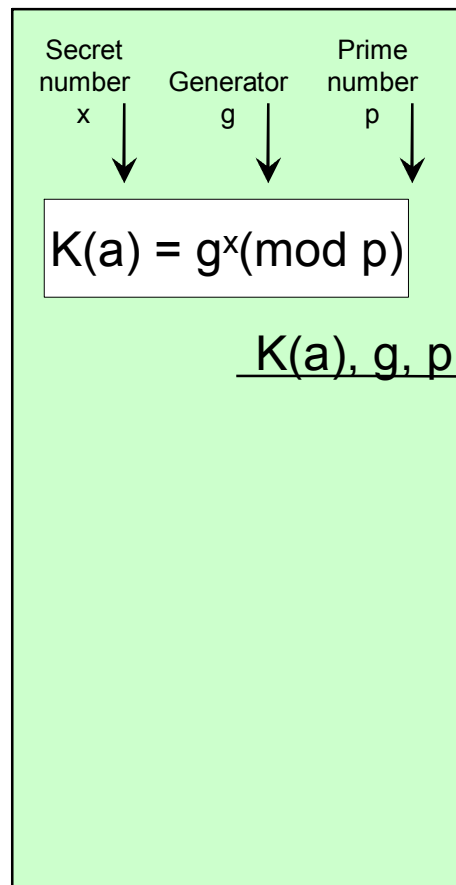
(further explained)



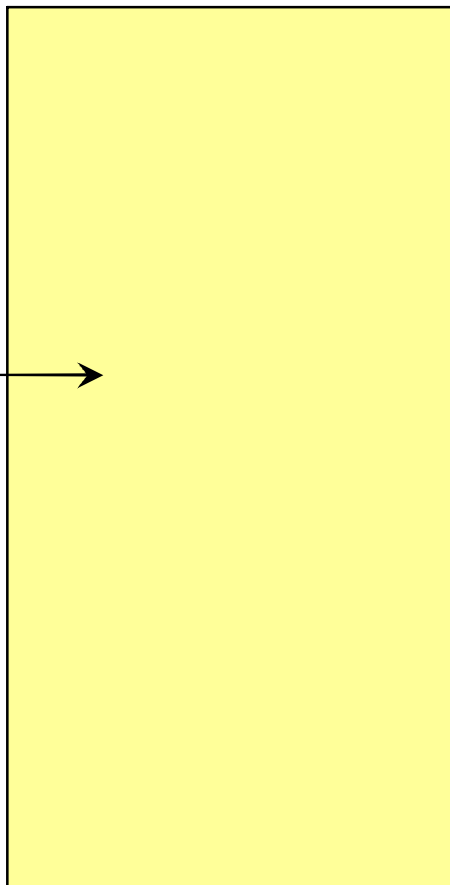
What does it mean to work with EAP-TLS, but
without certificates ???

Diffie-Hellmann (Without Certificates)

Side A



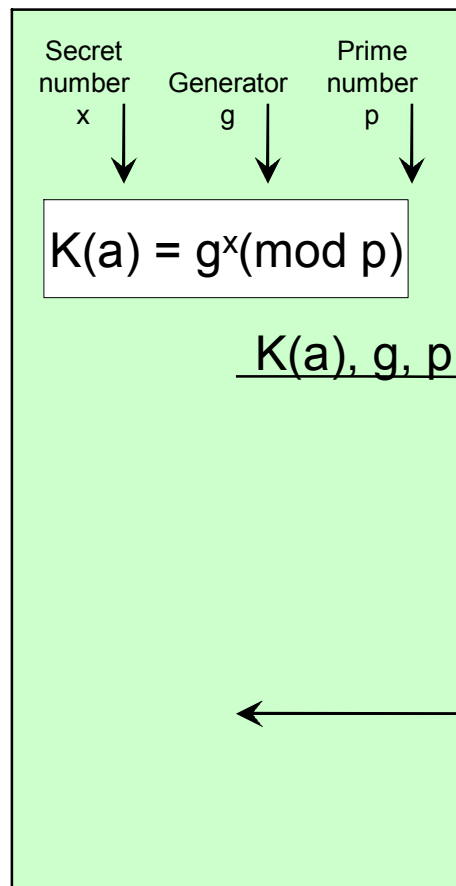
Side B



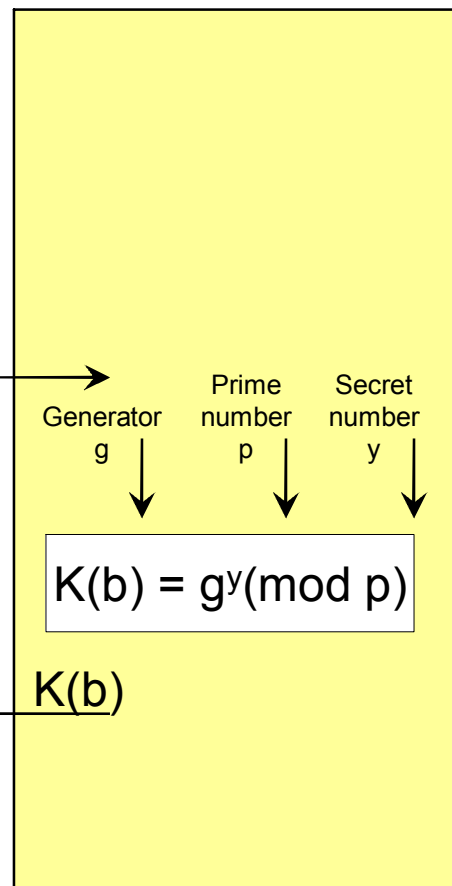
1. Each side selects a secret number x and y . These are referred to as the private keys.
2. Side A starts by selecting a large prime number (p) and a smaller integer called the generator (g)
3. Side A calculates using modulus mathematics its public key, $K(a)$ using the prime number and the secret key as following:
 $\rightarrow K(a) = g^x \pmod{p}$
4. Side A sends this Public Key, the prime number (p), and the generator (g) to side B

Diffie-Hellmann (Without Certificates)

Side A

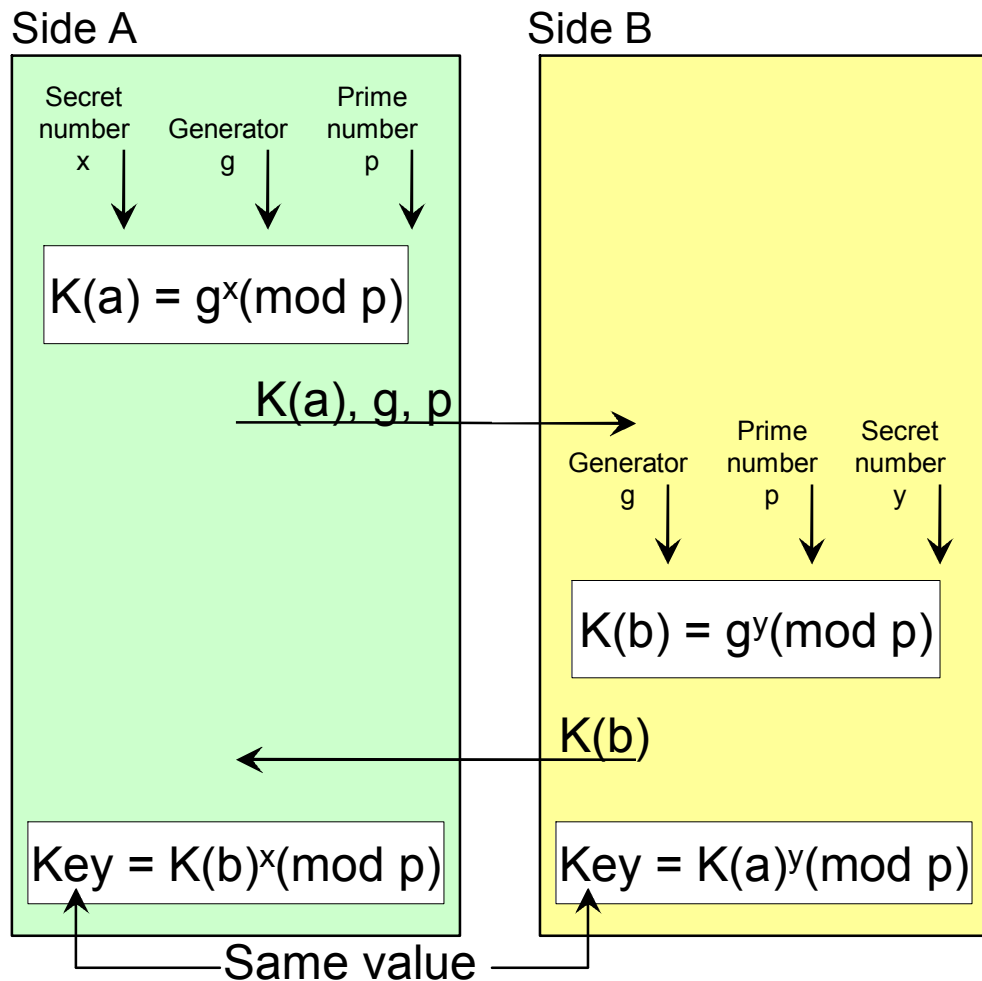


Side B



5. Side B performs a similar calculation with its secret key and the prime and generator to get its public key.
6. Side B sends its Public key to A.
7. Now both sides can calculate the shared key as follows
 - Shared key = $K(b)^x \pmod{p}$
 - Shared key = $K(a)^y \pmod{p}$

Diffie-Hellmann (Without Certificates)



8. The two shared key calculations produce the same value property of modulus arithmetic).
9. This key can be now used to start AES encryption

Setup with EAP-TLS using no Certificates (Diffie-Hellman Algorithm)

Setup with EAP-TLS – No Certificates

Station Configuration

Interface <wlan1>

General Wireless Data Rates Advanced WDS ...

Radio Name: 000C420C545B

Mode: station

SSID: ☒ AP_no_Cert

Band: 2.4GHz-B/G

Frequency: 2462

Scan List: ☐

Security Profile: Profile-no-Cert

Frequency Mode: manual txpower

Country: no_country_set

Antenna Gain: 0 dBi

DFS Mode: none

Proprietary Extensions: post-2.9.25

Default AP Tx Rate: ☐ bps

Default Client Tx Rate: ☐ bps

☒ Default Authenticate

☒ Default Forward

☐ Hide SSID

disabled running connected to ess

Security Profile

Security Profile <Profile-no-Cert>

General EAP Static Keys

EAP Methods: EPA-TLS

TLS Mode: no certificates

TLS Certificate: none

disabled running connected to ess

Setup with EAP-TLS – No Certificates

AP Configuration

Interface <AP_no_Cert>

General Wireless WDS Status Traffic

Master Interface: wlan2

SSID: ☒ AP_no_Cert

Area:

Security Profile: EAP-TLS-NoCert

Max Station Count: 2007

Proprietary Extensions: post-2.9.25

Default AP Tx Limit: ☐ bps

Default Client Tx Limit: ☐ bps

☒ Default Authenticate

☒ Default Forward

☐ Hide SSID

OK Cancel Apply Disable Comment Copy Remove

disabled running

Security Profile

Security Profile <EAP-TLS-NoCert>

General EAP Static Keys

EAP Methods: EPA-TLS

TLS Mode: no certificates

TLS Certificate: none

OK Cancel Apply Copy Remove

disabled running

How secure is EAP-TLS without Certificates ?

- After the anonymous negotiation results a PMK that is used to encryption AES (WPA2) or RC4 (WPA)
- Since there is no Pre Shared Key, the method itself is very secure. The (big) problem is if a hacker configures a rogue Mikrotik with the same method and put it in a position to impersonate AP or Client...
- The workaround for this potential risk is to close the link using EAP-TLS without certificates, but only allow communication after closing a L2TP or PPTP tunnel.

Deploying EAP-TLS with Certificates

Deploying EAP-TLS with Certificates

Step A → Create the Certificate Authority

Step B → Create the Certificate Requests

Step C → Take the Requests to be signed by the CA

Step D → Import the signed Certificates to the Mikrotik Boxes

Step E → If necessary, create Certificates for Windows Machines

Deploying EAP-TLS with Certificates

A) Creating the Certificate Authority (CA) [1/3]

→ In a Linux machine with OpenSSL installed edit the SSL config file with the data for the Certificates you will generate later:

/etc/ssl/openssl.cnf

dir	= ./MikrotikBrasil_CA
countryName_default	= BR
stateOrProvinceName_default	= Sao Paulo
0.organizationName_default	= MikrotikBrasil_Private_Network

Deploying EAP-TLS with Certificates

A) - Creating the Certificate Authority (CA) [2/3]

→ Edit the script that creates the CA (CA.sh) to the same directory you put in 1.1

`/usr/lib/ssl/misc/`

`CATOP=./MikrotikBrasil_CA`

→ Run the script with the option `-newca`

```
root@mikrotikbrasil:/etc/ssl# ./misc/CA.sh -newca
```

CA certificate filename (or enter to create)

→ Press <enter> and answer the questions

NB: Note that when you will be asked for Common Name it will be suggested to put your name.

You may prefer to put the Organization Name instead, because this name will appear for the clients in the Certificates created.

Deploying EAP-TLS with Certificates

A) - Creating the Certificate Authority (CA) [3/3]

→ The Certificate has been created and is stored at:

`/usr/lib/ssl/misc/MikrotikBrasil_CA/cacert.pem`

→ A DES protected private could be also found at:

`/usr/lib/ssl/misc/MikrotikBrasil_CA/private/cakey.pem`

Deploying EAP-TLS with Certificates

B - Creating the Certificate requests [1/1]

For Mikrotik boxes, certificates could be created either:

→ by RouterOS command line:

```
/certificates/create-certificate-request
```

→ by the command below in a Linux Machine with OpenSSL:

```
openssl req -new -keyout key_file.pem -out cert_req.pem -days 1825
```

Both methods create the private key and the Certificate Request file that needs to be signed. 1825 days means that the Certificate will expire in 5 years.

Deploying EAP-TLS with Certificates

C) - Signing the Certificates requests [1/1]

→ Disregard the method used for creation, Certificates could be signed in the Linux Machine with :

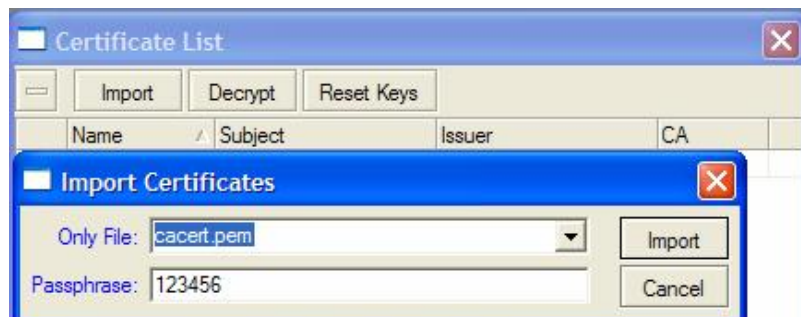
```
./openssl ca -config ./openssl.conf -policy policy_anything  
-out /cert_signed.pem -infiles /cert_req.pem
```

Now you can delete the req file because you're going to use only the file "cert_signed.pem" and the private key

Deploying EAP-TLS with Certificates

D) – Importing the Certificate to RouterOS [1/1]

Importation via Winbox



After Importation



You can import via terminal with

`/certificate import` and giving the pass phrase you used when created

Observe carefully the messages because the cert and the key are imported separately.

You'll need to give the password to import the protected Key.

Deploying EAP-TLS with Certificates

E) - Creating Certificates for Windows Machines:

For Windows machines we'll create Certificates in P12 format with the command:

Saying that we've created client1_cert.pem and client1_key.pem,

```
openssl pkcs12 -export -in cliente1_cert.pem -inkey cliente1_key.pem  
-out cliente1.p12
```

This will create client1.p12 – an appropriate format to import to Windows.

Deploying EAP-TLS with Certificates

With the Certificates created, with RouterOS you can choose:

- To work with Certificates both in AP and Clients
- To work with Certificates in Clients and Radius.

Setup with EAP-TLS using Certificates at AP and Clients

Setup with EAP-TLS (AP with Certificate) AP Configuration

AP Configuration

Interface <AP_with_Cert>

General Wireless WDS Status Traffic

Master Interface: wlan2

SSID: ☒ AP_with_Cert

Area: ☐

Security Profile: EAP-TLS-APCert

Max Station Count: 2007

Proprietary Extensions: post-2.9.25

Default AP Tx Limit: ☐ bps

Default Client Tx Limit: ☐ bps

☒ Default Authenticate

☒ Default Forward

☐ Hide SSID

OK Cancel Apply Disable Comment Copy Remove

disabled running

Security Profile

Security Profile <EAP-Cert>

General EAP Static Keys

EAP Methods: EAP-TLS

TLS Mode: verify certificate

TLS Certificate: cert1

OK Cancel Apply Copy Remove

Certificate

Certificate List

Import Decrypt Reset Keys

Name	Subject	Issuer	CA
KQR cert1	C=BR, ST=Sao Paulo,...	C=BR, ST=Sao Paulo,...	yes

K - decrypted private key, Q - private key, R - rsa

Setup with EAP-TLS (AP with Certificate) Client Configuration

Station Configuration

Security Profile

Certificate

Name	Subject	Issuer	CA
KQR cert1	C=BR, ST=Sao Paulo, ...	C=BR, ST=Sao Paulo, ...	yes

K - decrypted private key, Q - private key, R - rsa

Setup with EAP-TLS using Radius

Deploying EAP-TLS with Certificates

Step E → Installing Radius Server with EAP-TLS support

Step F → Creating Radius Server Certificate

Step G → Installing Radius Server Certificate

Step H → Configuring Radius

NB: The configurations showed in this presentation are for FreeRadius and Debian.
Make sure to adapt them for your own distribution.

Deploying EAP-TLS with Certificates

E) - Installing the Radius Server with EAP-TLS support [1/1]

Because of OpenSSL license is not compatible with FreeRadius GPL, some Linux distributions don't compile natively the library EAP-TLS.

So, you have to do some hack to get a Radius Server running with EAP-TLS

How to (in French) install Radius under Debian with EAP-TLS (and PEAP too):

<http://www.queret.net/blog/index.php/2007/04/04/72-freeradius-avec-support-eap-tls-eap-ttls-eap-peap-sur-linux-debian-etch>

NB: If you feel more comfortable in Portuguese language, just ask ☺

Deploying EAP-TLS with Certificates

F) - Creating the RADIUS Server's Certificate [1/1]

The Radius Server Certificate can be created in the same way other Certificates. Therefore, because Certificates are created with the Private Key encrypted you must type the Private Key on every Radius Startup.

To avoid this uncomfortable situation use the option `-nodes` when generating the Certificate request for Radius:

```
openssl req -nodes -new -keyout key_file.pem -out req_file.pem -days 1825
```

Sign it as usual and that's ready to use.

Deploying EAP-TLS with Certificates

G) - Instalng the Certificate at RADIUS Server

→ Create a random seed for RADIUS and Diffie-Hellman parameter:

```
cd /etc/freeradius  
dd if=/dev/random of=./certs/random count=2  
openssl dhparam -check -text -5 -512 -out ./certs/dh
```

→ Copy the Radius Certificate , Radius key and the CA Certificate files to /etc/freeradius/certs. The directory should contain:

```
ls /etc/freeradius/certs  
  
radius_cert radius_key cacert.pem dh random
```

Deploying EAP-TLS with Certificates

H) - Configuring the RADIUS Server [1/4]

→ Edit clients.conf informing the list of AP's (NAS's) that will use Radius

```
vi /etc/freeradius/clients.conf
client 192.168.100.1/32 {
    secret          = 123456
    shortname       = AP1
}
```

→ Edit radiusd.conf

```
vi /etc/freeradius/radiusd.conf
```

```
user = nobody
group = nogroup
```

Deploying EAP-TLS with Certificates

H) - Configuring the RADIUS Server [2/4]

Editing radiusd.conf - cont

```
authorize    {  
              preprocess  
              chap  
              mschap  
              suffix  
              eap  
              files  
            }
```

Deploying EAP-TLS with Certificates

H) - Configuring the RADIUS Server [3/4]

→ Editing eap.conf

```
root@radius:/usr/local/etc/raddb# aee eap.conf
```

```
default_eap_type = tls
tls {
    private_key_file = ${raddbdir}/certs/radius_key.pem
    certificate_file = ${raddbdir}/certs/radius_cert.pem
    CA_file = ${raddbdir}/certs/cacert.pem
    dh_file = ${raddbdir}/certs/dh
    random_file = ${raddbdir}/certs/cacert.pem
}
```

→ Finally start the Radius Server

```
root@radius:/usr/local/etc/raddb# ./radiusd -X
```


Deploying EAP-TLS with Certificates

H) Configuring the RADIUS server [4/4]

vi /etc/freeradius/users

user_1 Auth-Type := EAP

user_2 Auth-Type := EAP

...

DEFAULT Auth-Type := Reject

Reply-Message := "You were kicked by Radius"

→ Very important:

user_1, user_2, etc must match the same name you have used when you created the Certificate.

Station Configuration

Interface <wlan1>

General | Wireless | Data Rates | Advanced | WDS | ...

Radio Name: 000C420C545B

Mode: station

SSID: ☒ AP_to_Radius

Band: 2.4GHz-B/G

Frequency: 2462

Scan List: ☐

Security Profile: Profile-EAP-TLS

Frequency Mode: manual txpower

Country: no_country_set

Antenna Gain: 0 dBi

DFS Mode: none

Proprietary Extensions: post-2.9.25

Default AP Tx Rate: ☐ bps

Default Client Tx Rate: ☐ bps

☒ Default Authenticate

☒ Default Forward

☐ Hide SSID

disabled | running | connected to ess

Setup with EAP-TLS + Radius Client Configuration

Security Profile

Security Profile <EAP-Cert>

General | EAP | Static Keys

EAP Methods: EPA-TLS

TLS Mode: verify certificate

TLS Certificate: cert1

Certificate

Certificate List

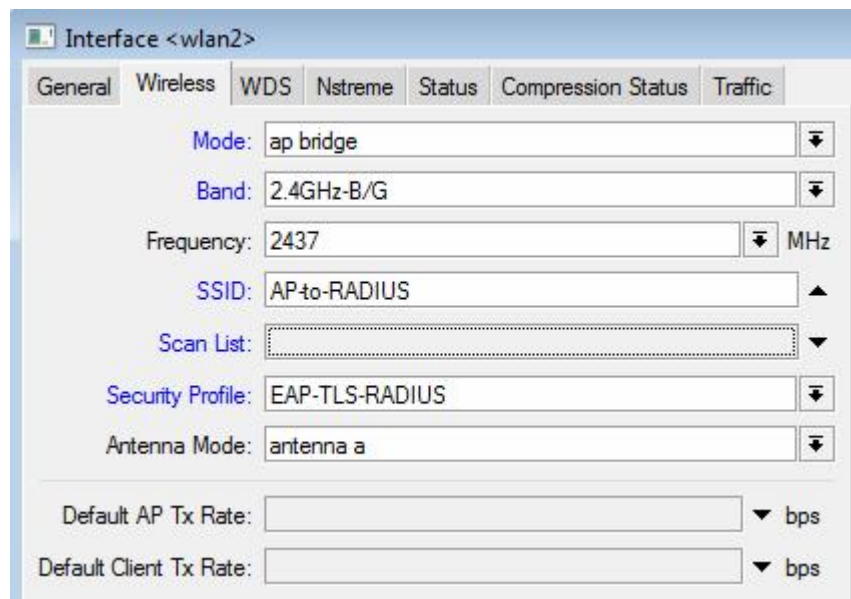
Import | Decrypt | Reset Keys

Name	Subject	Issuer	CA
KQR cert1	C=BR, ST=Sao Paulo,...	C=BR, ST=Sao Paulo,...	yes

K - decrypted private key, Q - private key, R - rsa

Setup with EAP-TLS + Radius AP Configuration

AP Configuration



Interface <wlan2>

General Wireless WDS Nstreme Status Compression Status Traffic

Mode: ap bridge

Band: 2.4GHz-B/G

Frequency: 2437 MHz

SSID: APto-RADIUS

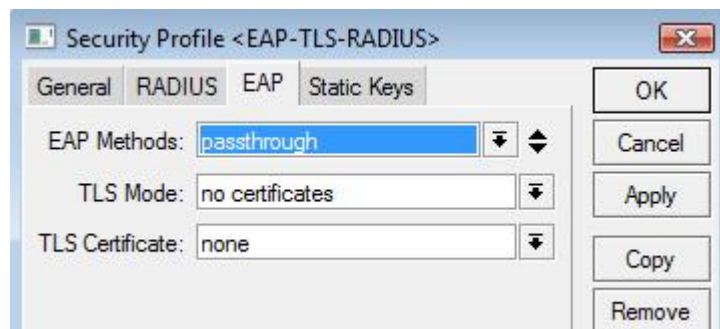
Scan List:

Security Profile: EAP-TLS-RADIUS

Antenna Mode: antenna a

Default AP Tx Rate: bps

Default Client Tx Rate: bps



Security Profile <EAP-TLS-RADIUS>

General RADIUS EAP Static Keys

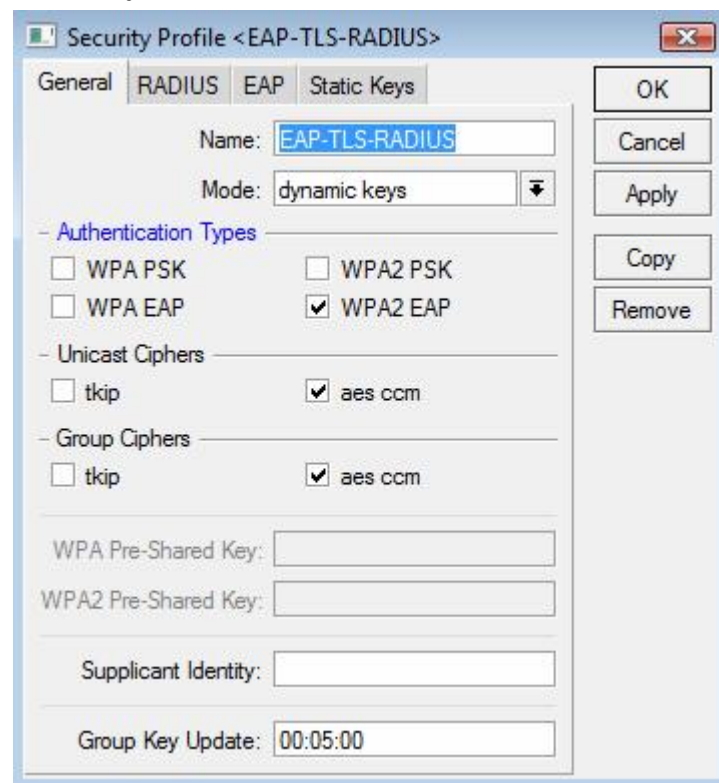
EAP Methods: passthrough

TLS Mode: no certificates

TLS Certificate: none

OK Cancel Apply Copy Remove

Security Profile



Security Profile <EAP-TLS-RADIUS>

General RADIUS EAP Static Keys

Name: EAP-TLS-RADIUS

Mode: dynamic keys

Authentication Types

☐ WPA PSK ☐ WPA2 PSK

☐ WPA EAP ☒ WPA2 EAP

Unicast Ciphers

☐ tkip ☒ aes ccm

Group Ciphers

☐ tkip ☒ aes ccm

WPA Pre-Shared Key:

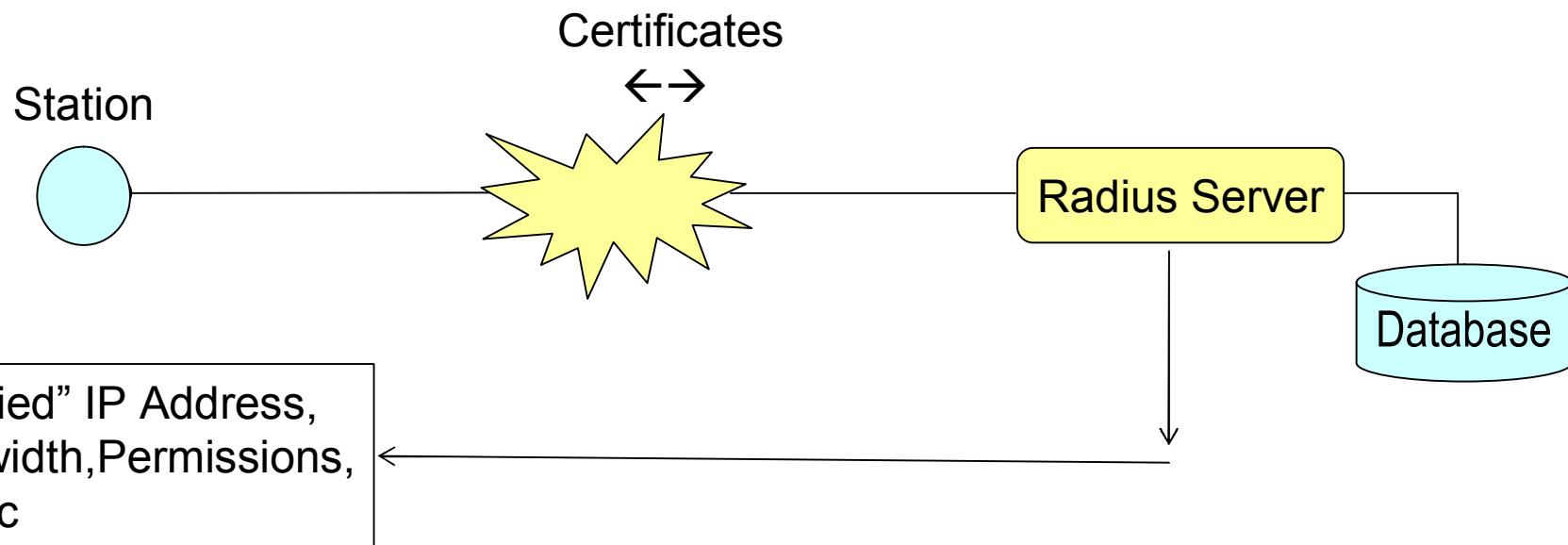
WPA2 Pre-Shared Key:

Supplicant Identity:

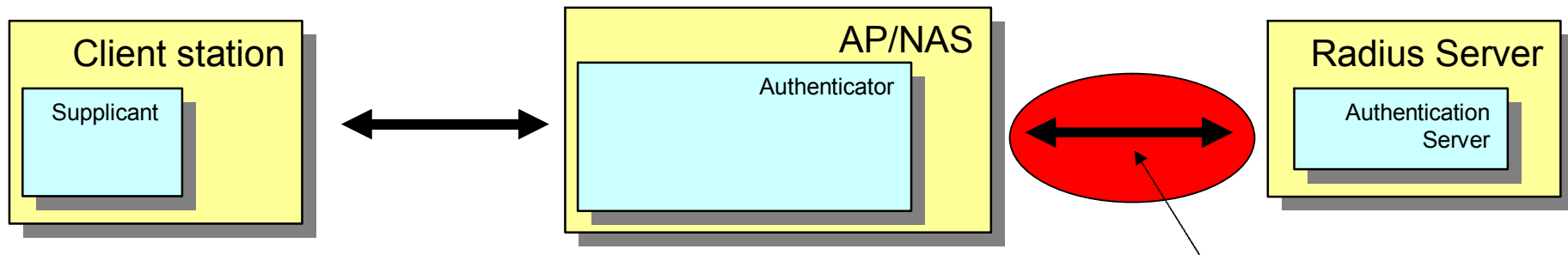
Group Key Update: 00:05:00

OK Cancel Apply Copy Remove

EAP-TLS Based Backbone



Is WPA2 EAP-TSL really “Bullet Proof”?



→ If an attacker has physical access to the link between Radius And AP, he could deploy a dictionary attack to discover the Radius secret and after this, discover the PMK's

Attacking delivery of PMK over RADIUS

→ To avoid this consider the possibility of making a L2PT tunnel with IPSec between Radius and AP.

802.11i

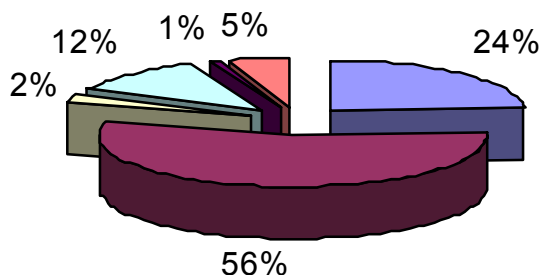
X

WISP's

Back to the past

In 2002 Brazilian WISP's with the "recommended" security measures felt themselves very secure !

Seguranca provedores 2002



■ Nenhuma Medida	■ Controle de MAC - ACL	■ Controle de MAC - Radius
■ Controle de MAC + IP	■ PPPoE	■ WEP

AND TODAY ???

Research in September of 2007

Total of WISP's that participated in the research: 74

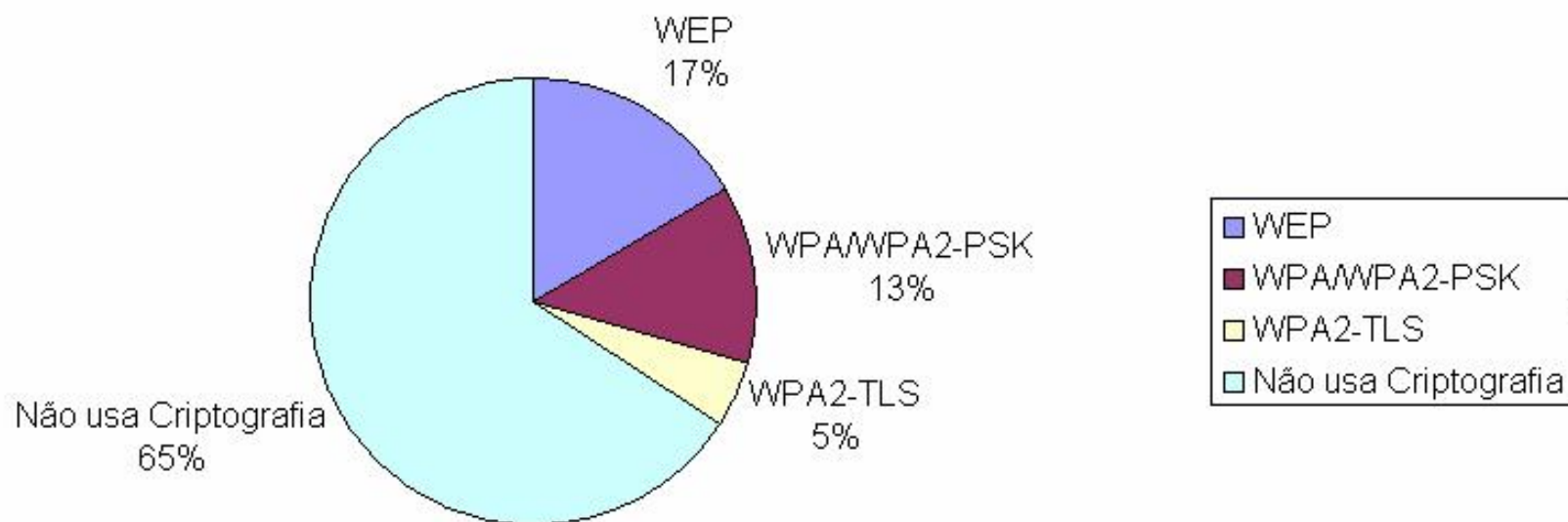
Total of clients covered: 52.385

Agregated Bandwidth to the Internet: 585.6 mbps

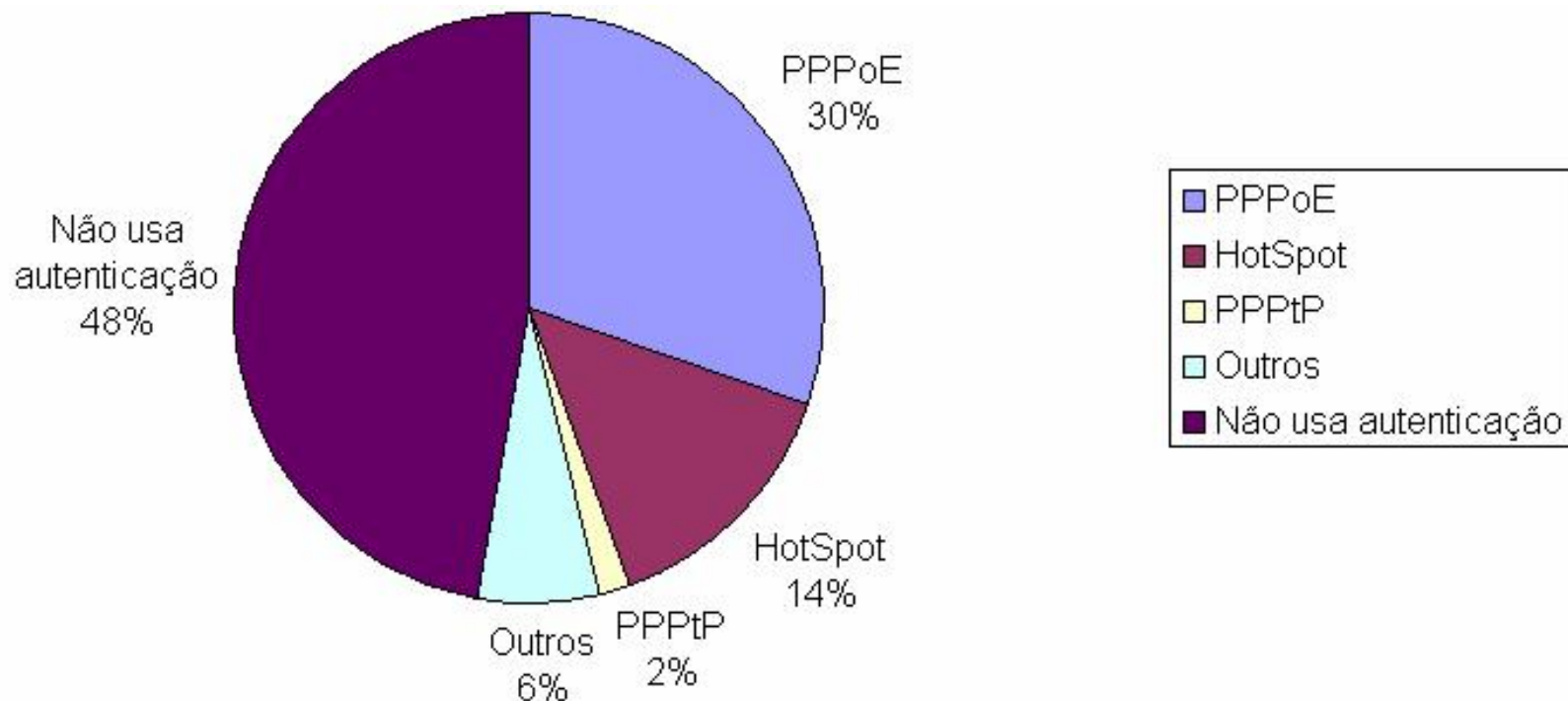
The results were compiled per user basis, i.e. taking in account the number of clients declared by each WISP.

For example, the numbers given by a provider with 1000 clients, had 10 times more weight than another with 100 clients.

Research in September of 2007 Encryption



Research in September of 2007 Authentication



NB: Among all providers that use PPPoE or Hotspot, only 4% use encryption (i.e. 96% use PPPoE or Hotspot as an exclusive security measure)

Research in September of 2007 MAC spoofing



Why don't WISP's use encryption ?

- Very complex to deploy
- Legacy Hardware doesn't support encryption
- WPA could be cracked in the future, like WEP did.
- Performance problems with encryption

Nowadays WISP's preferred security methods

To provide security, most of WISP's use as exclusive solution:

→ PPPoE Tunnels

→ Hotspot authentication

We are going to do a critical analysis of such methods, when used as security methods.

PPPoE Tunneling overview

- PPPoE : first developed for wired Networks means Point to Point Protocol over Ethernet.
- The PPPoE Server (PPPoEd) listens to PPPoE requests and client uses PPPoE discover protocol. PPPoE works at layer II
- In RouterOS users can be authenticated in the local database or a external Radius Server can be used
- User/password can be protected by means of using CHAP authentication. PAP passes in plain text.
- A “dialer” installation is necessary

PPPoE Tunelling overview

- One configuration parameter of the server is the “service name”. If it’s left in blank any requisition will be responded.
- The Interface that ‘listens” PPPoE requests do not have an routeable IP. If it has, any user can by pass PPPoE authentication configuring manually a valid IP in his machine.
- Like other tunneled protocols MTU and MRU should be decreased.
- PPPoE is very sensible to signal variations.

Security with PPPoE

- PPPoE is not an encrypted Tunnel by default. It can be configured with encryption, but the client must support encryption and at the server side encryption increases processor overhead.
- A Spoofed MAC doesn't allow the attacker to navigate, but causes a lot of problems to the legitimated user.
- .
- It's very easy to cause a Denial of Service against a PPPoE server in the air. Someone can flood the server with tons of requisitions and there is nothig we can do to avoid it.
- The worst thing with PPPoE is that the user doesn't authenticate the server (concentrator). Because of this a "man-in-the-middle" attack is trivial. In WISP's plants if an attacker puts a rogue AP with a better signal in relation of a victim client, it can capture PPPoE discover and forces user to connect in the rogue PPPoEd

Hotspots overview

- Usually used to provide access in Hotels, Shoppings, etc is being used as a framework to provide authentication in WISP's environment.
- The interface configured as a Hotspot captures the browsers requisitions and asks for a user/password
- RouterOS can authenticate in the local database or in a external Radius.
- With RouterOS Hotspots can run https by means of a Certificate that can ensure mutual authentication.

Security with Hotspots

- Since the client is authenticated and her/his pair IP+MAC is discovered and spoofed by an attacker, he gains the access to the network. Services will be compromised for both users (the real and the attacker). No conflict will happen.
- Running a Hotspot with Certificates and forcing the authentication via https, in theory you can ensure mutual authentication, but it depends on clients knowledge and actions.
- RouterOS Universal Client feature is useful to provide an extra level of security. When providing fixed service using Hotspot you should disable hotspot DHCP and configure whatever IP you want on client. If discovered the pair IP+MAC you can change the IP and avoid the problem

PPPoE x Hotspot - conclusions

PPPoE has a lot of benefits because in a PPPoE plant there is no IP traffic and a lot of problems due to network broadcasts, virus, etc are not present.

The man-in-the-middle and DoS's attacks to PPPoE plants are serious concerns and there is no countermeasures to avoid it when the physical medium is the air.

Hotspots installed with Certificates, can “avoid” man-in-the middle attack (clients must be well informed)

Wireless Security – (almost) final conclusions

Wireless Security method that ensures:

- Authentication (mutual)
- Confidentiality
- Data Integrity

Is only achieved by means of a implementation of:

802.11i + EAP-TLS + Radius

Other implementations like VPN's between wireless clients and a Concentrator, could be considered and could be effective, but they will cause extra administrative work and will demand more and more processor power with the network growth.

Why don't WISP's use encryption ? Should they ?

- Very complex to deploy
 - Not True. With Mikrotik all is very easy to deploy.
- Legacy Hardware doesn't support encryption
 - It's a fact, but not a reason. With Mikrotik you can have a lot of Security Profiles to connect all kinds of clients.
- WPA could be cracked in the future, like WEP did.
 - Who knows ?. But the differences between both techniques are enormous and there is no way to compare them.
- Performance problems with encryption
 - In the past this was true. Today with new Atheros chipsets this is not a real issue.

Availability Compromised with Layer II attacks

Availability Compromised – Layer II attacks

IEEE 802.11i cared about

- Authentication
- Confidentiality
- Integrity

Unfortunately it didn't care about availability. Wi-Fi service can be seriously compromised with 2 types of attack :

- High RF Power based (in fact a kind of layer I attack)
- Protocol Based Attacks

Availability Compromised – Layer II attacks

→ RF High Power based attacks (Jamming)

Since we are working with unlicensed bands, there is not much we can do but only call the authorities responsible for spectrum use.

We can be less vulnerable to such attacks with a good RF project.

→ Protocol Based Attacks

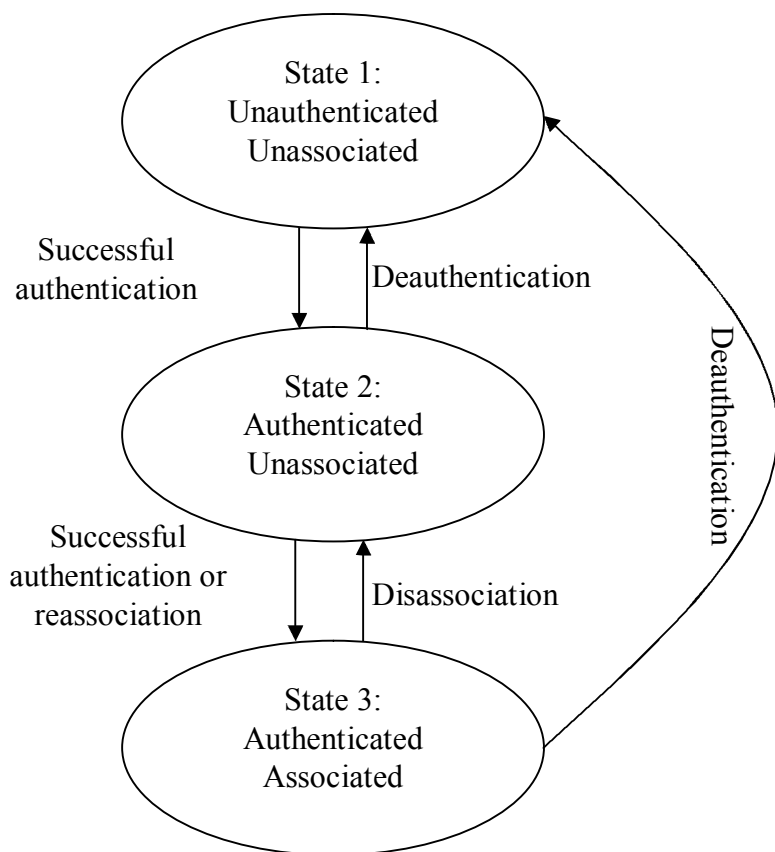
They are based on a weak design of 802.11 that is very dependent on MAC addresses.

There are a lot of tools in the internet that can be used to perform this attacks, like Void11, airreplay, etc. You can even get a Live CD with this tools and a lot of other hacking tools:

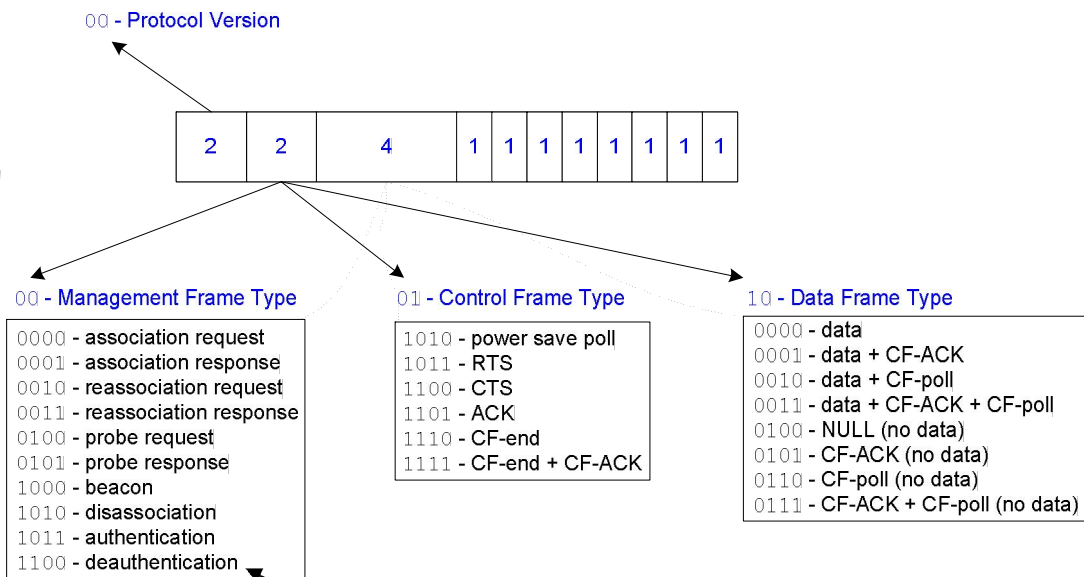
www.wlanbrasil.com.br/downloads/seguranca/cd1.iso

www.wlanbrasil.com.br/downloads/seguranca/cd2.iso

Association Process



802.11 Types and Subtypes

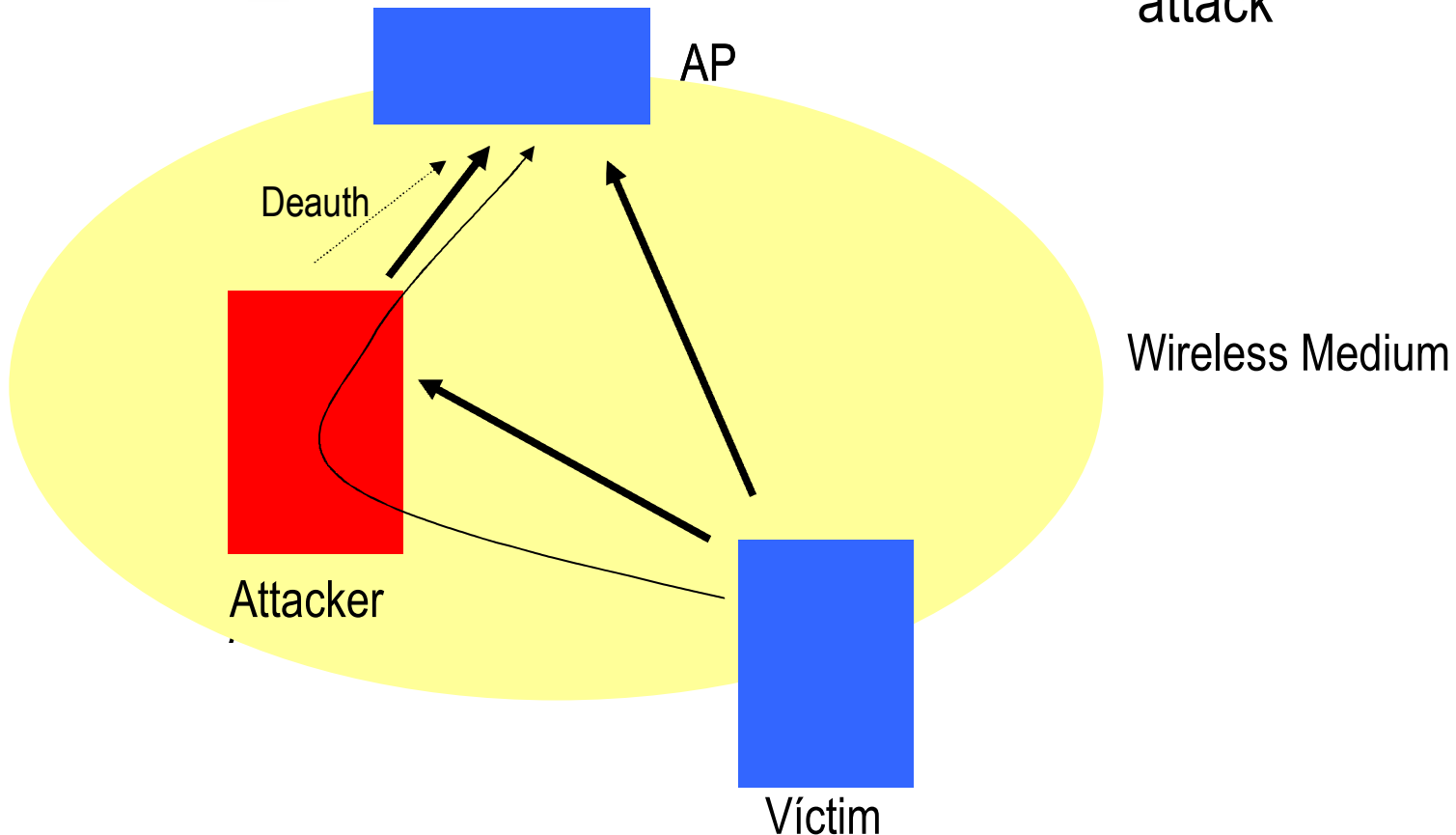


Deauth Attack

- 1 – Attacker uses any tool like airopeek, kismet, wellenreiter, etc to find out :
 - AP's MAC
 - Client's MAC
 - Channel in use
- 2 – Go to a position that AP can hear his frames (even a poor signal will work and he doesn't need to be authenticated neither associated)
- 3 – Launch the attack asking the AP to de-authenticate the client sending de-auth requests

If the attacker has a good position and good signal, this attack can be more sophisticated with another tools compelling the client to associate with the Rogue AP while the Rogue AP associates with the Real AP using Clients Credentials – That's another kind of the man-in-the-middle attack

Man-in-the-middle in the air (Monkey Jack) attack



In this case, encryption does not avoid the denial of service, but the man-in-the-middle

De-auth Attack countermeasures with RouterOS the definitely solution

A definitely solution is only achieved by means of a Protocol Modification

The idea is :

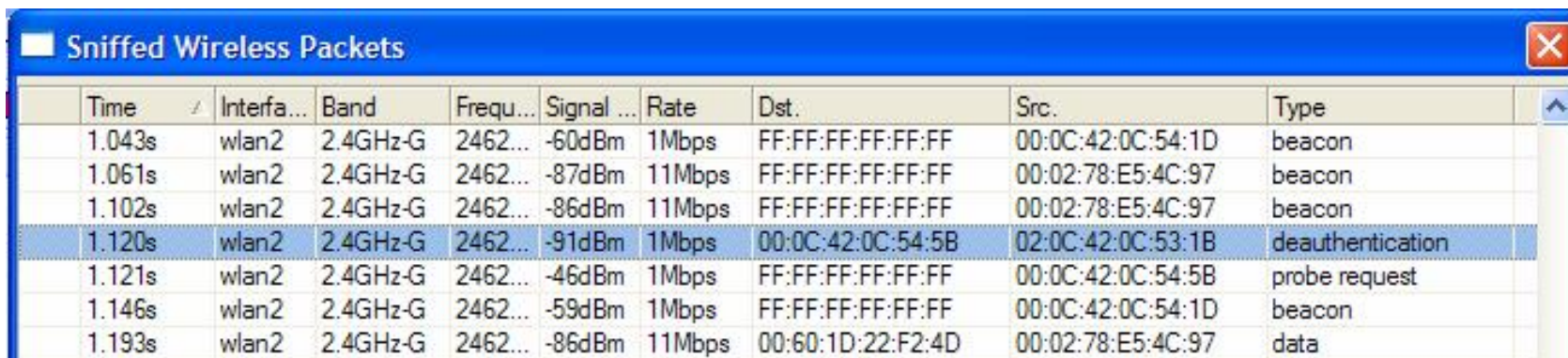
- AP delays honoring the de-authentication request for a short period of time, say (5 – 10 s)
- If no other frames are received from the source, then accept the de-auth request
- If source sends data, then discard the request

<http://sysnet.ucsd.edu/~bellardo/pubs/usenix-sec03-80211dos-slides.pdf>

Since such modification is at protocol level there is nothing we users can do.

De-auth Attack countermeasures with RouterOS

The first thing is to make sure you're really under a de-auth attack. Let's look at the Wireless packets sniffed under /interface/wireless/sniffer



Time	Interfa...	Band	Frequ...	Signal ...	Rate	Dst.	Src.	Type
1.043s	wlan2	2.4GHz-G	2462...	-60dBm	1Mbps	FF:FF:FF:FF:FF:FF	00:0C:42:0C:54:1D	beacon
1.061s	wlan2	2.4GHz-G	2462...	-87dBm	11Mbps	FF:FF:FF:FF:FF:FF	00:02:78:E5:4C:97	beacon
1.102s	wlan2	2.4GHz-G	2462...	-86dBm	11Mbps	FF:FF:FF:FF:FF:FF	00:02:78:E5:4C:97	beacon
1.120s	wlan2	2.4GHz-G	2462...	-91dBm	1Mbps	00:0C:42:0C:54:5B	02:0C:42:0C:53:1B	deauthentication
1.121s	wlan2	2.4GHz-G	2462...	-46dBm	1Mbps	FF:FF:FF:FF:FF:FF	00:0C:42:0C:54:5B	probe request
1.146s	wlan2	2.4GHz-G	2462...	-59dBm	1Mbps	FF:FF:FF:FF:FF:FF	00:0C:42:0C:54:1D	beacon
1.193s	wlan2	2.4GHz-G	2462...	-86dBm	11Mbps	00:60:1D:22:F2:4D	00:02:78:E5:4C:97	data

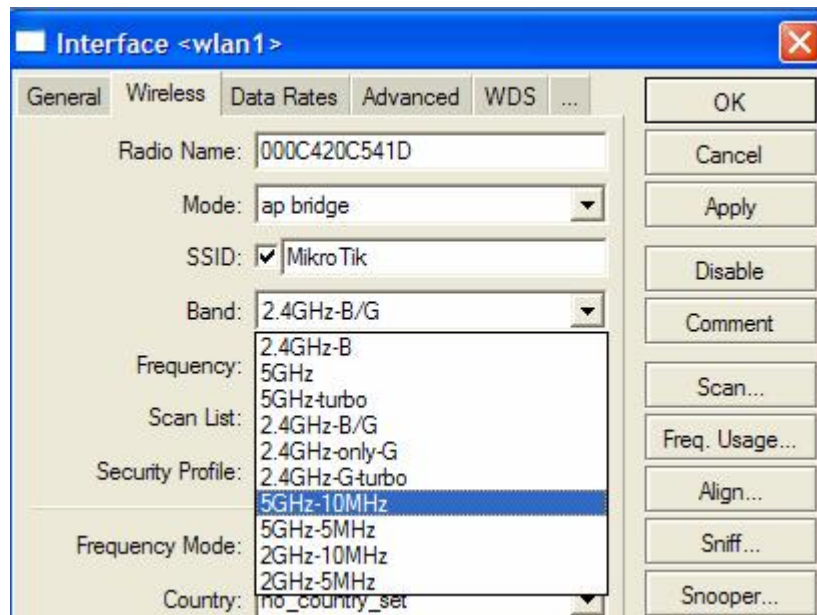
It's usual that you find some packets like this but in a deauth-attack the number will be very high. Look at specially the Destination and Source MAC

De-auth Attack countermeasures with RouterOS

The Band Modes in 5 Ghz or 2.4 Ghz that use 10 and 5 MHz of width are not Affected by usual de-auth tools.

This was tested in practice with void11 And air-replay.

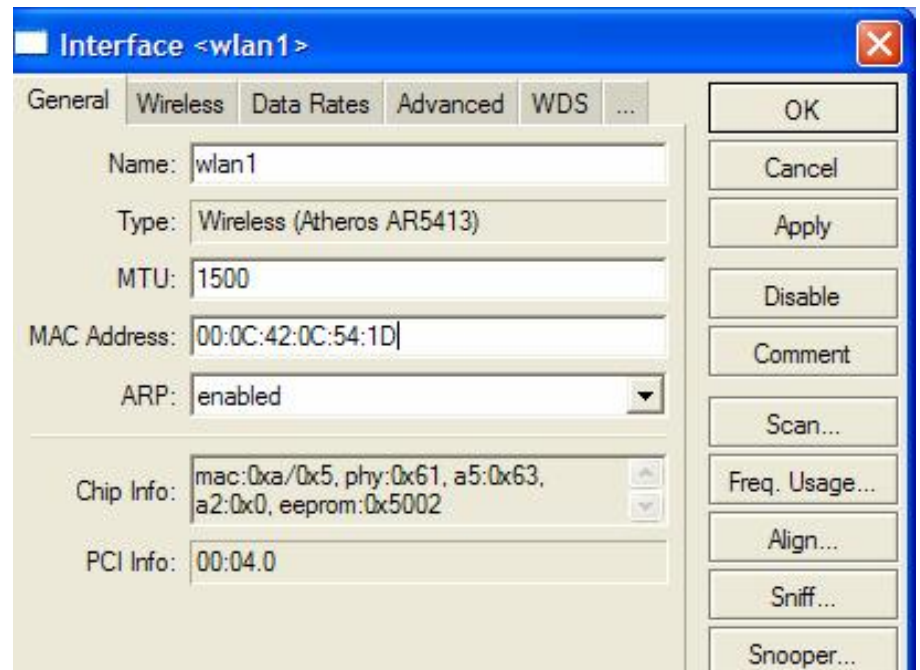
If you're being attacked in a Point to Point Link this can be a good solution.



De-auth Attack countermeasures with RouterOS

Since the attacks are performed using AP MAC, consider to change it in RouterOS.

This cannot be considered a elegant Security measure, but a workaround that can help until the attacker Discovers the new MAC



Countermeasures for de-auth attack with RouterOS

Security by means of “obscurity”

Using Virtual AP's with no practical function, but only to broadcast with a lot of SSID's and MAC's a hard environment could be created to avoid sniffers and MAC discovers.

Virtual AP + scripts can be used to create such environment dynamically

This technique was inspired in “Fake AP” – a perl script that do this in a Linux machine

<http://www.blackalchemy.to/project/fakeap>

Actual Work

- Implementing security measures in all providers of our association.
- Working on a low cost CPE with EAP-TLS support
- Working on an “W”-IDS to detect layer II attacks

Dziękuję.

Na zdrowie !

Wardner Maia

maia@mikrotikbrasil.com.br