

802.11 Wireless Attacks

Pavol Lupták

Copyright © 2005 Pavol Lupták, OpenWeekend 2005

802.11

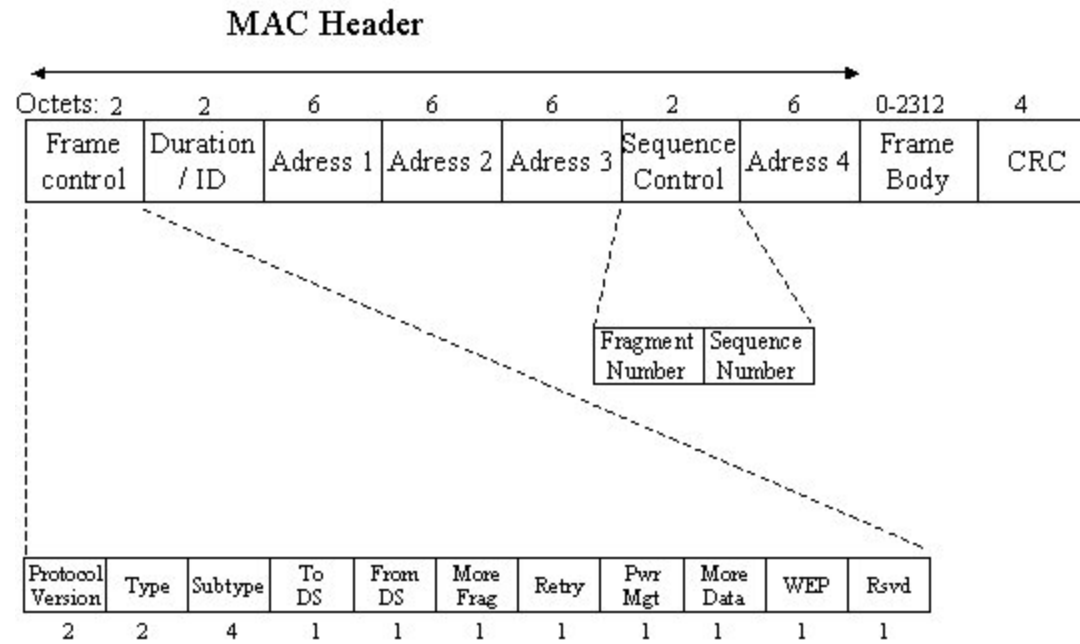
802.11

Link layer

Provides:

- **basic authentication mechanisms (CSMA/CA, RTS/CTS)**
- **fragmentation support**
- **network separation on the same frequencies using BSSID**
- **mobility between BSS**
- **funny "security" using WEP**
- **power management (STA has an ability to switch to sleep mode while AP is spooling data)**

MAC header



AUTH frame

Authentication frame

- **type = 0x00 (Management), subtype = 0x0B**
- **provides an authentication of device to AP. The device can be authenticated to many AP. There are various ways of the authentication:**
 - ***Open authentication (null authentication algorithm)* - if no encryption is enabled on the network, any device that knows the SSID can gain access to the network. Otherwise if a device does not have the correct WEP key (and its authentication is successful), the device will be unable to transmit data through the AP.**
 - ***Shared key authentication* - the client and AP have to use static shared WEP key. After the client sends an authentication request to the AP,**

AUTH frame (Continued)

the AP responds with an authentication response containing challenge text. The client encrypts the challenge text with its WEP key and reply with a subsequent authentication request. If the AP can decrypt this request and retrieve the original challenge text, it grants the client access.

- ***MAC address authentication*** - an additional mechanism to the open shared key authentications that reduces the likelihood of unauthorized devices accessing the network

DEAUTH frame

Deauthentication frame

- **type = 0x00 (Management), subtype = 0x0C**
- **provides a deauthentication of the authenticated clients (the AP sends DEAUTH before its reboot, shutdown)**
- **potential security weakness (It is possible to deauthenticate all authenticated clients by sending spoofed DEAUTH frame with BSSID of the AP)**

802.1x

- **network standard for medium access (does not refer to 802.11 only)**
- **Port-based access control, a client is authenticated to an AP using authentication server (RADIUS, TACACS+)**
- **At the beginning, clients are authenticated to AP, their communication depends on the authentication to EAP server**
- **For achieving a complex security there is inevitable to combine 802.1x with:**
 - **1. Dynamic (per-session) WEP keys**
 - **2. (Per Packet Keying, TKIP, MIC, Broadcast key rotation) = 802.11i**
 - **3. Securing on higher layers (SSH, SSL protocols)**

802.11i

- **CCMP (Cipher Block Chaining MAC protocol) - 128-bits encryption using AES**
- **TKIP (Temporal Key Integrity Protocol) - each packet is encrypted with unique WEP key generated using a one-way transformation from the primary key and incremental IV vector. After all IV vectors are tried (16.7 mil combinations) the primary WEP key is regenerated using 802.1x. A packet integrity is provided using MIC (Message Integrity Check) that represents 32-bit value calculated from the frame header, payload, bit-flipping-proof sequential number - there is no possibility to forge datagrams or use the same IV**

802.11i (Continued)

- **Key management and replacement of RC4 with AES, WRAP (Wireless Robust Authenticated Protocol), EAP a Radius standard**

Attacks

Attacks

O/S/M Vulnerabilities

- **Open authentication vulnerabilities** - no way for the the AP to determine whether a client is valid or not (WEP encryption should be always implemented)
- **Shared key authentication vulnerabilities** - requires the client use a preshared WEP key. An attacker can capture both the plain-text challenge text and the cipher-text response and performs an exclusive (XOR) function on the plain-text with the cipher-text to produce the key stream
- **MAC address authentication vulnerabilities** - an attacker can subvert the MAC authentication process by "spoofing" valid MAC address

WEP Encryption

- based on the weak RC4 algorithm (symmetric key stream cipher)
- IV (initialization vector) is a 24-bit value that augments a 40-bit WEP key to 64 bits and 104-bit WEP key to 128 bits and is sent in the clear (i.e. the effective key strength is only 40bits and 104bits)
- IV + WEP key produce the key stream - ciphertext = plaintext XOR keystream, plaintext = ciphertext XOR keystream
- the knowledge of the ciphertext and plaintext is enough for the computation of the key stream

Passive Attacks

- the key scheduling algorithm (KSA) issue - several weak IVs can reveal key bytes after statistical analysis
- researchers at AT&T/Rice University and the developers of the AirSnort demonstrated this vulnerability and verified that WEP keys can be derived after as few as 4 million frames
- using dynamic WEP keys can mitigate this vulnerability, but not eliminate
- WEP injection
[<http://www.dachb0den.com/users/h1kari/work/.0-day/>] can be used to inject new packets to increase weak IV (not "pure" passive attack) (a source/destination address and payload remain same, the AP responds with "duplication" error on network layer)

Passive Attacks (Continued)

- **Passive WPA PSK Dictionary Attack**
[<http://wifinetnews.com/archives/002452.html>]

All these attacks have been practically implemented!

Airsnort [<http://airsnort.shmoo.com/>], **dwepcrack**

[<http://www.e.kth.se/~pvz/wifi/>], **Wepecrack**

[<http://wepcrack.sourceforge.net/>], **Aircrack**

[<http://www.cr0.net:8040/code/network/aircrack/>], **We-**

pLab [<http://weplab.sourceforge.net/>]

Active Attacks

Due to the lack of effective message integrity the following attacks are feasible:

- **initialization vector reply attacks**
- **bit-flipping attacks**
- **man-in-the middle attacks**

All these attacks have been practically implemented!

Aireplay, WEPWedgie

[<http://sourceforge.net/projects/wepwedgie/>], Chopchop

[<http://www.netstumbler.org/showthread.php?t=12489>],

Airjack [<http://sourceforge.net/projects/airjack/>]

IV Reply Attack

- 1. A known plain-text message is sent to an wireless client (an email message, ICMP request, ..)**
- 2. The attacker will sniff the wireless looking for the predicted cipher-text**
- 3. The attacker will find the known frame and compute the key stream**
- 4. The attacker can grow the key stream to any size required**
 - The attacker builds a frame one byte larger than the known key stream size (ICMP frame should be ideal for obtaining the response)**
 - The attacker augments the key stream by one byte - he tries all possible values (i.e. he sends 256 ICMP requests)**

IV Reply Attack (Continued)

- **When the attacker guesses the correct value, the expected response (e.g. the ICMP reply message) is received**
- **The attacker can repeat this process until the desired key stream length is obtained**

Bit-flipping Attack

- 1. The attacker captures the frame and flips random bits in the data payload of the frame**
- 2. The attacker modifies the ICV (Integrity Check Value - CRC32) and transmits the modified frame**
 - An original frame (F1) has an ICV (C1)**
 - The attacker creates a new frame (F2) containing the bits to flip and computes its ICV (C2)**
 - The attacker computes a new bit-flipped frame as $F3 = F1 \text{ XOR } F2$ and its new ICV $C3 = C1 \text{ XOR } C2$**
- 3. The receiver (a client or AP) receives the frame and calculates the ICV based on the frame contents and compares this value with the value in the ICV field of the frame - everything is ok - the receiver accepts the modified frame**

Bit-flipping Attack (Continued)

- 4. The receiver de-encapsulates the frame and processes the Layer 3 packet. Because bits are flipped in the layer packet, the Layer3 checksum fails and the receiver IP stack generates a predictable error**
- 5. The attacker sniffs the wireless LAN looking for the encrypted error message**
- 6. When the attacker receives the encrypted error message, he computes the key stream as with the IV replay attack**

ManInTheMiddle



MITM Attack

- 1. The attacker knows (=is able to find out) the victim wireless parameters (the MAC address, ESSID/BSSID, number of channel)**
- 2. The attacker sends (via broadcast or unicast) a DEAUTH request to the victim (on the same channel as the victim) with the spoofed source address of the victim's AP**
- 3. The victim is deauthenticated and starts to search all channels for a new valid AP**
- 4. The attacker spoofs on a new channel his forged AP with the original MAC address (BSSID) and ESSID of the victim's AP. The forged AP responses to all victim's AUTH, AS_RQ and REAS_RQ frames needed for the victim's authentication/association**

MITM Attack (Continued)

- 5. After the successful victim's authentication/association to the forged AP, the attacker spoofs on the original victim's channel the victim's MAC address and associates to the original victim's AP (the AP supposes the associated client is the victim, not attacker)**
- 6. The attacker is in the middle of the victim and his AP**

Cisco LEAP

- **Lightweight Extensible Authentication Protocol (easy to install, configure and support)**
- **LEAP specification is only opened to business partners under NDA**
- **Proprietary EAP method because of its requirement to use a Cisco AP**
- **Provides an authentication (uses a modified MS-CHAPv2 challenge/response/in clear, mutual authentication to mitigate MITM attacks, short-lived WEP keys to encrypt data, prevents usage of weak IV's from the AP)**
- **Still vulnerable to passive dictionary attacks (MS-CHAPv2 and LEAP client-AP Challenge/Response weaknesses)**

LEAP weaknesses

MS-CHAPv2 weaknesses:

- No salt in stored NT (double MD4) hashes - permits pre-computed dictionary attacks
- Weak DES key selection for challenge/response
- Username is sent in the clear-text

LEAP client-AP Challenge/response weaknesses:

1. The AP sends a random 8-byte challenge to the client
2. The client uses 16 byte NT hash (MD4) of the user password to generate 3 DES keys (NT₁-NT₇) (NT₈-NT₁₄) (NT₁₅-NT₁₆+"\0\0\0\0\0")
3. Each DES key is used to encrypt the challenge (each generating 8 bytes of output) and the 24-byte response is sent back to the AP.

LEAP weaknesses (Continued)

4. The AP responses with success or failure message

LEAP attack

The 3rd DES key is weak (due to the 5 NULL's in every challenge/response) there are only 2^{16} hash possibilities

```
'grep "BN-1BN$" ntdash-dict > possible-passwords'
```

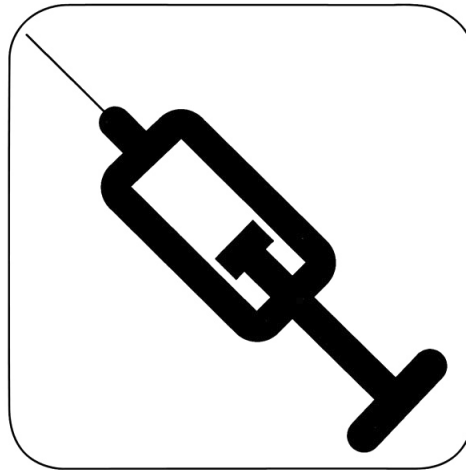
- 1. Take a large password list, calculate MD4 hashes and password+NT hash list**
- 2. Capture LEAP challenge/response, extract username, challenge, response, calculate the last 2 bytes bytes of the NT hash from the response**
- 3. Search through password+hash list for hashes with matching bytes**

LEAP attack (Continued)

4. **Use matching entries to encrypt the challenger - matching captured and calculated response will indicate the user's password**

ICV issue

- **Integrity Check Value - CRC32 vulnerable to XOR operation**
- **ICV does not apply to datagram header**
- **IV reply, bit-flipping attacks**



Passive WPA PSK

- **Each station may have its own PSK (pre-shared key)**
- **When no 802.1X is used, the PSK is used directly as the Pairwise Master Key (PMK). When PSK is passphrase, then $PMK = PBKDF2(\text{passphrase}, \text{ssid}, \text{ssidLength}, 4096, 256)$;**
- **The PTK (Pairwise Transient Key) is a keyed-HMAC function using the PMK on the two MAC addresses and the two nonces from the first two packets of the 4-Way Handshake unique for each session**

WPA PSK attacks

- **Any PTK can be generated by learning the two MAC addresses, nonces and selected ciphersuite during the initial exchange**
- **Anyone with knowledge of the PSK can determine any PTK in the ESS (Extended Service Set) through passive sniffing of the wireless network**
- **A key generated from a passphrase of less than about 20 characters is likely vulnerable to the dictionary attack**
- **Since the single PSK is used for the whole ESS, the attacker is now a member of the ESS, and the whole ESS is compromised**
- **WPA PSK should use only truly random keys!!!**

PSK attack tools

- **CoWPaaty**
[<http://www.securiteam.com/tools/6L00F0ABPC.html>]
and **Aircrack**
[<http://www.cr0.net:8040/code/network/aircrack/>]
- **For a dictionary attack to be effective, it must take each dictionary word and perform 4096 iterations of HMAC-SHA1 with two nonce values and the supplicant and authenticator MAC addresses - too slow (approximately 70 words/second on a Pentium 4 3.8 GHz system)**

Conclusion

Conclusion

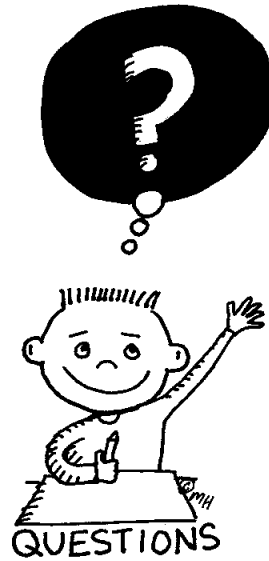
References

- ***Joshua Wright* - Detecting Wireless LAN MAC Address Spoofing**
[<http://home.jwu.edu/jwright/papers/wlan-mac-spoof.pdf>]
- ***Joshua Wright* - Layer 2 Analysis of WLAN Discovery Applications for Intrusion Detection**
[<http://home.jwu.edu/jwright/papers/l2-wlan-ids.pdf>]
- ***abaddon@802.11ninja.net* - Advanced 802.11 Attack**
[<http://802.11ninja.net/bh2002.ppt>]
- ***John Bellardo, Stefan Savaga* - 802.11 Denial-of-Service Attacks: Real Vulnerabilities and Practical Solutions**
[<http://www.cs.ucsd.edu/~savage/papers/UsenixSec03.pdf>]
- ***Michael Ossman* - WEP: Dead Again, Part 1**
[<http://www.securityfocus.com/infocus/1814>]

References (Continued)

- ***Michael Ossman - WEP: Dead Again, Part 2***
[<http://www.securityfocus.com/infocus/1824>]
- ***Robert Moskowitz & Glenn Fleishman - Weakness in Passphrase Choice in WPA Interface***
[<http://wifinetnews.com/archives/002452.html>]

Questions & Answers



Thanks!

