Wi-Fi trickery, or how to secure (?), break (??) and have fun with Wi-Fi

ShmooCon2006, Washington – January 13-15, 2006

Laurent BUTTI & Franck VEYSSET – France Telecom Division R&D

{laurent.butti;franck.veysset} AT francetelecom.com

Who are we?

- **Network security experts in R&D labs**
  - Employed by France Télécom (major telco)
- **Speakers at security-focused conferences**
  - ShmooCon, ToorCon, FIRST, EuroSec…
- **ShmooCon 2005 speakers ;-)**
  - « Design and Implementation of a Wireless IDS »

Agenda

- **State of the art of (some) useful 802.11 attacks**
  - Starting with WiFi 101
  - Non exhaustive, we only have a one hour timeslot ;-)
- **Wireless frames and injection quick overview**
  - Description of 802.11 frames
  - Description of `RAW` injection
- **Let's present new stuff!**
  - An enhanced Fake AP
  - A Glue AP
  - A covert channel

WIFI 101

Different Modes

- Managed (Client mode)
- Adhoc (IBSS / *Independent Basic Service Set*)
- Master (ie AP mode)
- Monitor ☺

Different “channels”

Different SSID (networks)

- Essid = network name
- Bssid = Mac @

WIFI 101: Different frames

Management frames

- Authentication / Deauthentication
- Association / Disassociation
- Beacon frame
- Probe request / probe response

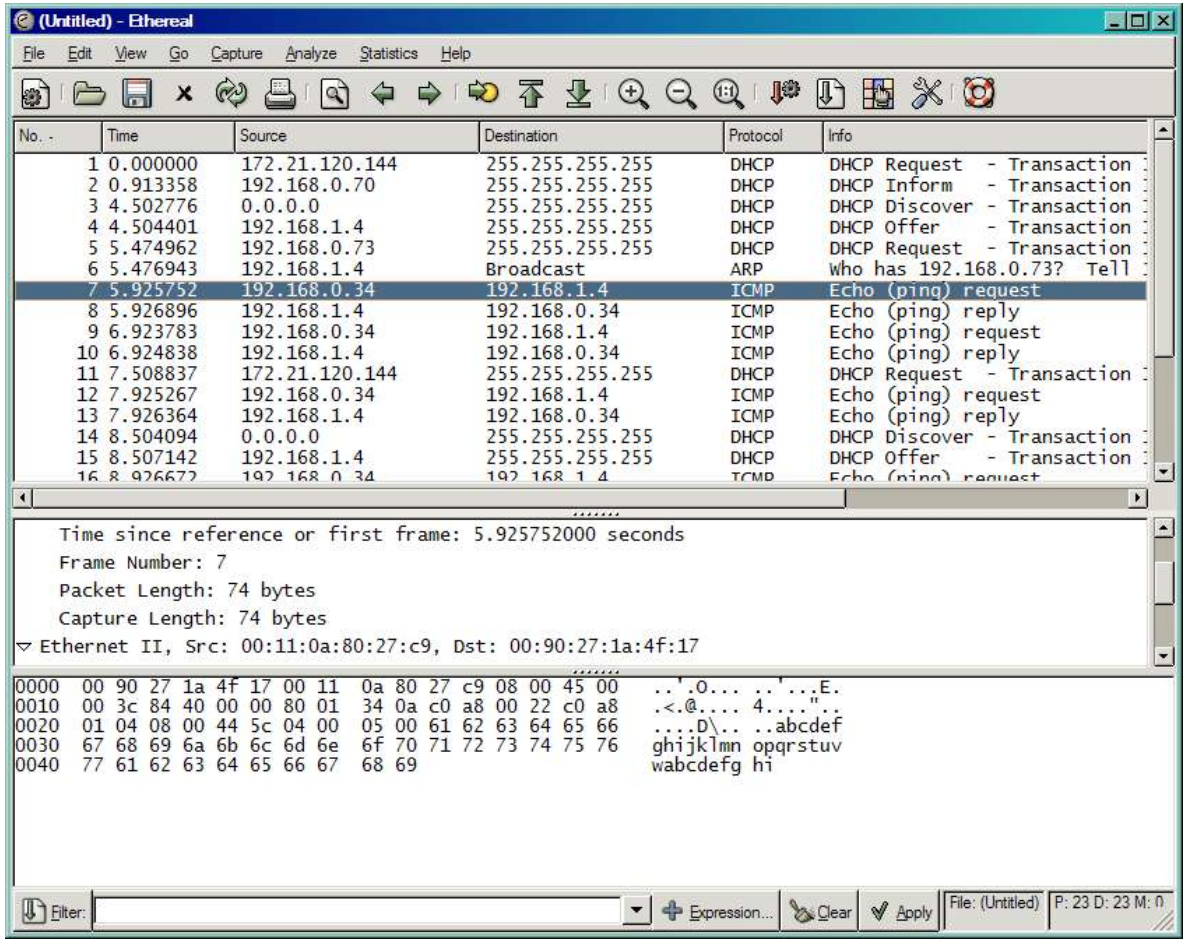
Control frames

- RTS/CTS
- Acknowledgement frame

Data frame

Ethereal

- You guys all know about ethereal…
- Easier to use under *Nix
- http://www.ethereal.com/
- Good 802.11 support (monitor mode)


(Untitled) - Ethereal
File Edit View Go Capture Analyze Statistics Help
No. - Time Source Destination Protocol Info
1 0.000000 172.21.120.144 255.255.255.255 DHCP DHCP Request - Transaction
2 0.913358 192.168.0.70 255.255.255.255 DHCP DHCP Inform - Transaction
3 4.502776 0.0.0.0 255.255.255.255 DHCP DHCP Discover - Transaction
4 4.504401 192.168.1.4 255.255.255.255 DHCP DHCP Offer - Transaction
5 5.474962 192.168.0.73 255.255.255.255 DHCP DHCP Request - Transaction
6 5.476943 192.168.1.4 Broadcast ARP Who has 192.168.0.73? Tell
7 5.925752 192.168.0.34 192.168.1.4 ICMP Echo (ping) request
8 5.926896 192.168.1.4 192.168.0.34 ICMP Echo (ping) reply
9 6.923783 192.168.0.34 192.168.1.4 ICMP Echo (ping) request
10 6.924838 192.168.1.4 192.168.0.34 ICMP Echo (ping) reply
11 7.508837 172.21.120.144 255.255.255.255 DHCP DHCP Request - Transaction
12 7.925267 192.168.0.34 192.168.1.4 ICMP Echo (ping) request
13 7.926364 192.168.1.4 192.168.0.34 ICMP Echo (ping) reply
14 8.504094 0.0.0.0 255.255.255.255 DHCP DHCP Discover - Transaction
15 8.507142 192.168.1.4 255.255.255.255 DHCP DHCP Offer - Transaction
Time since reference or first frame: 5.925752000 seconds
Frame Number: 7
Packet Length: 74 bytes
Capture Length: 74 bytes
Ethernet II, Src: 00:11:0a:80:27:c9, Dst: 00:90:27:1a:4f:17
0000 00 90 27 1a 4f 17 00 11 0a 80 27 c9 08 00 45 00 ..'.O... ..'...E.
0010 00 3c 84 40 00 00 80 01 34 0a c0 a8 00 22 c0 a8 .<.@.... 4....".
0020 01 04 08 00 44 5c 04 00 05 00 61 62 63 64 65 66 ....D\.. ..abcdef
0030 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 ghijklmn opqrstuv
0040 77 61 62 63 64 65 66 67 68 69 wabcdefg hi
Filter: Expression... Clear Apply File: (Untitled) P: 23 D: 23 M: 0

Stumbler vs. Sniffer

- Sniffers like Ethereal, Tcpdump, or Kismet capture raw data frames. Kismet always operates in monitor mode, other sniffers can. Sniffers can see data packets.

- Stumblers query the card firmware to see what networks are detectable in the area. They usually see fewer networks than sniffers, and can't capture data packets, but they don't require special drivers, either.

(Thanks to Dragorn Kismet presentation)

Netstumbler

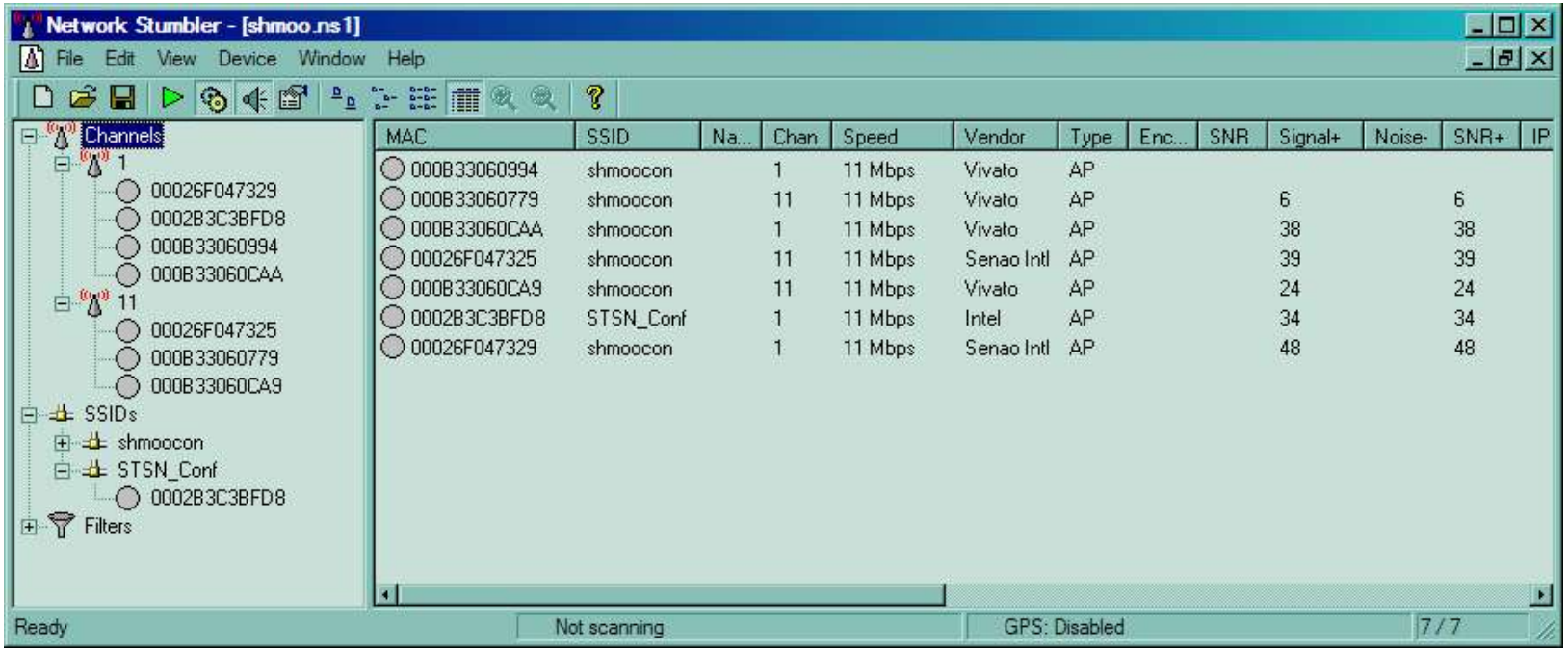
- http://www.netstumbler.com/
- **Current release: Netstumber 0.4 / MiniStumbler 0.4**

- **Active monitoring (send empty probe request frame)**
  - And do channel hopping
  - Can be configured with a GPS
  - To build map…

Netstumbler

Screenshot

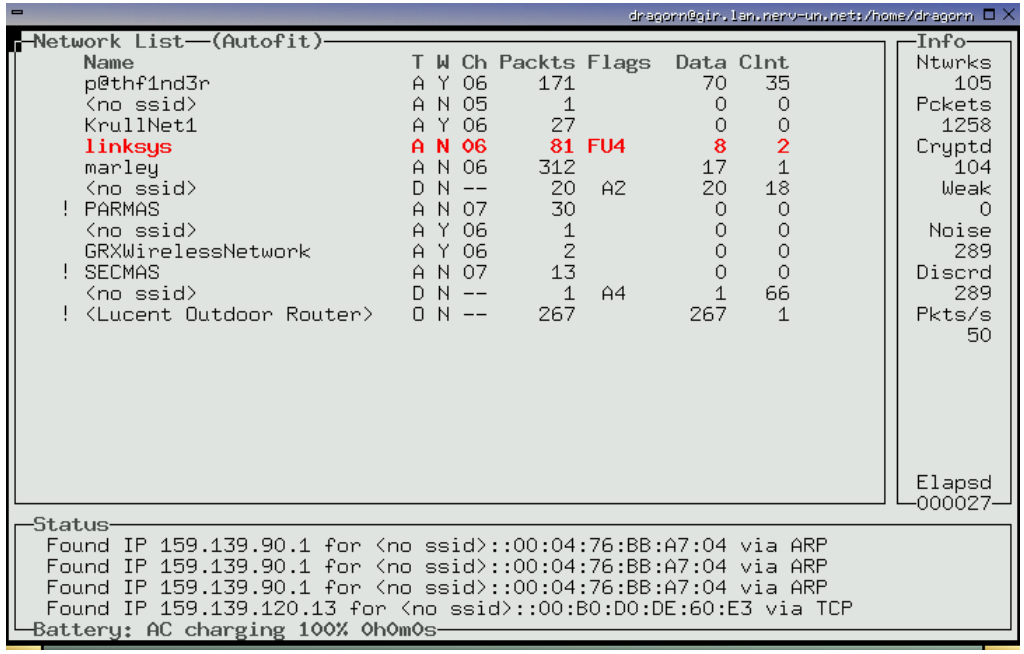
NETSTUMBLER.COM



KISMET

- **Very famous tool**
- **http://www.kismetwireless.net/**
  - Current release: Kismet-2005-08-R1
- **Passive monitor (ie listen to beacon / probe response)**
  - Also do channel hopping
  - Can use a GPS

Screenshot

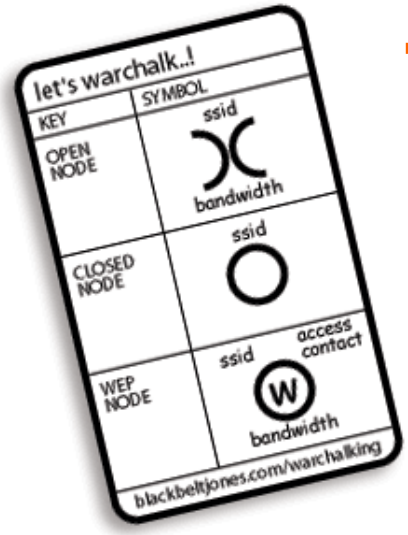

dragorn@gir.lan.nerv-un.net:/home/dragorn
Network List—(Autofit)
Name T W Ch Packts Flags Data Clnt
p@thf1nd3r A Y 06 171 70 35
<no ssid> A N 05 1 0 0
KrullNet1 A Y 06 27 0 0
linksys A N 06 81 FU4 8 2
marley A N 06 312 17 1
<no ssid> D N -- 20 A2 20 18
! PARMAS A N 07 30 0 0
<no ssid> A Y 06 1 0 0
GRXWirelessNetwork A Y 06 2 0 0
! SECMAS A N 07 13 0 0
<no ssid> D N -- 1 A4 1 66
! <Lucent Outdoor Router> O N -- 267 267 1
Info
Ntwrks 105
Pckets 1258
Cryptd 104
Weak 0
Noise 289
Discrd 289
Pkts/s 50
Elapsd 000027
Status
Found IP 159.139.90.1 for <no ssid>::00:04:76:BB:A7:04 via ARP
Found IP 159.139.90.1 for <no ssid>::00:04:76:BB:A7:04 via ARP
Found IP 159.139.90.1 for <no ssid>::00:04:76:BB:A7:04 via ARP
Found IP 159.139.120.13 for <no ssid>::00:B0:D0:DE:60:E3 via TCP
Battery: AC charging 100% 0h0m0s

WarDriving

- **Just listen for any IEEE 802.11 activity!**
  - Stealth…
- **Or send Probe Requests and listen for Probe Responses…**
  - Not stealth… ;-)

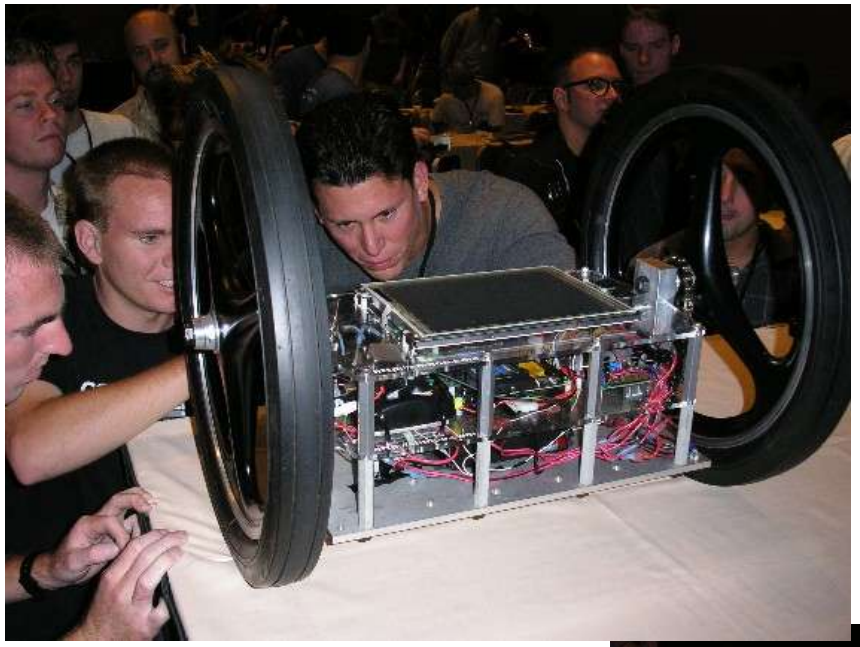
➔WarDriving

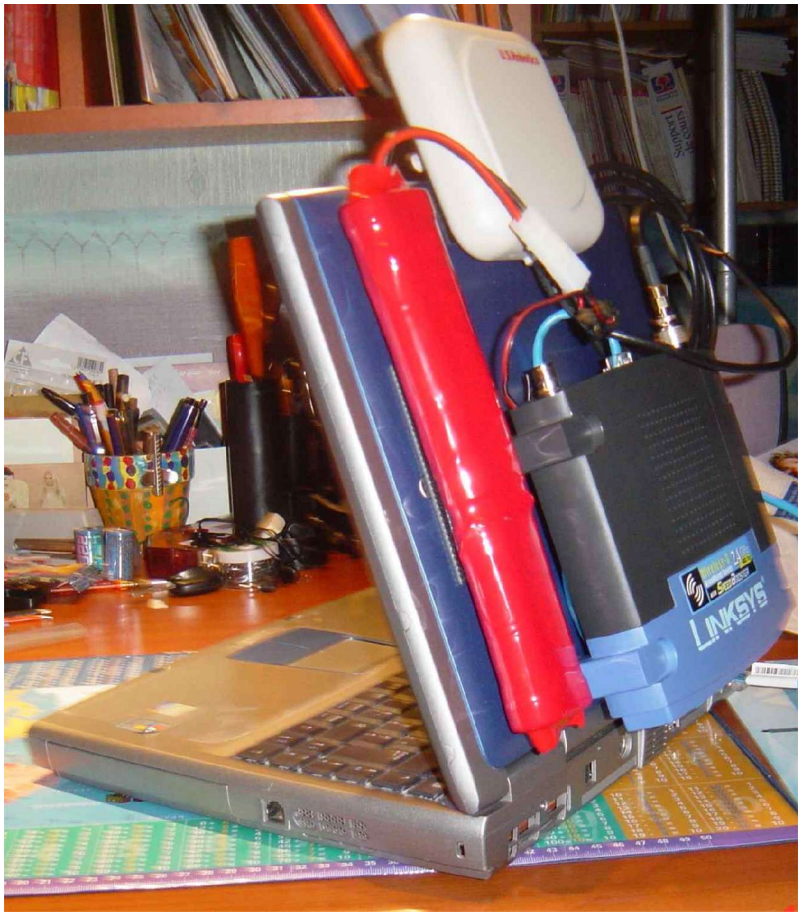
➔WarChalking



➔WarFlying

➔WarParking




LINKSYS

Defcon, a few years ago ☺

Definitions (1/2)

A rogue access point

- is a wireless access point that has been installed on a secure company network without explicit authorization from a local network management

A wireless intrusion detection system (WIDS)

- is a network device that monitors the radio spectrum for the presence of unauthorized, rogue access points

Source: Wikipedia, the free encyclopedia

Definitions (2/2)

- **No definition for ‘fake access point’ on Wikipedia**
- **Could be (in bad english) ;-)**
  - is an illegitimate wireless access point which purpose is to fool wireless users that usually connect themselves to legitimate access points
- **Could also be defined as**
  - a security nightmare!

RAW Injection (1/3)

- **We mean layer 2 frame injection**
  - 802.11 management, control and data frames
  - Could be extremely powerful!
- **Goal: inject any arbitrary frame**
  - Userland tool gives it to the kernel/driver
  - Driver gives it to the firmware
- **Was really tricky 2 or 3 years ago…**
  - Prism2/2.5/3 with HostAP was one of the only mean for frame injection
  - But with limitations (some 802.11 fields mastered by the firmware)
    - –Fragmentation, sequence number, BSS timestamp…

RAW Injection (2/3)

- **Today a large choice of chipsets and drivers supports it**
  - Prism2/2.5/3 with HostAP or wlan-ng
  - Prsim54 with prism54
  - Atheros with madwifi
  - Ralink RT2x00 with rt2x00
  - Realtek RTL8180 with rtl8180
- **Check Christophe Devine's aircrack for additional patches**
- **Injection and sniffing are performed in 'monitor' mode**
  - `socket(PF_PACKET, SOCK_RAW, htons(ETH_P_ALL))`
  - `iwconfig interface mode monitor`
  - `ifconfig interface up`

RAW Injection (3/3)

- **Could be used by Wireless IDS for layer 2 countermeasures**
  - One goal is to prevent wireless clients from associating to rogue access points
    - Thanks to deauthentication / deassociation floods
- **Could be used for tricky things**
  - WEP cracking speedup (à la aircrack)
  - Denial of service, association floods
  - Fake access points and clients
  - And so on…
- **Drastically increased the range of feasible attacks…**

(Big) Issue For Any Wireless IDS

- **Dealing with 'unuseful' data is a classic issue for any IDS**
  - Data mastered by an attacker who intends to corrupt the WIDS
- **`RAW` injection is a key feature to corrupt any WIDS**
  - Inserting arbitrary data in databases
  - Aggregating and correlating unuseful data
  - Flooding the GUI (and system administrators) ☹
- **A major challenge for any Wireless IDS vendor**
  - How to deal with an attacker flooding at the wireless IDS?

`fakeap.pl` (1/4)

- **You guys, know about infamous BlackAlchemy's Fake AP!**
  - Available at: http://www.blackalchemy.to/project/fakeap/
- **Basically it's a `PERL` script using `ifconfig` and `iwconfig`**
  - (Randomly) change BSSID, ESSID, channel, WEP and `txpower`
  - Feed it with an ESSID list and MAC prefixes
- **A wireless havoc for stumblers and wireless IDS**
  - Filling tables and GUI with random fake access points
- **But…**

fakeap.pl (2/4)

- **As BSSIDs are randomized (and not cyclic), you may use**
  - A timeout window to flush 'old' fake access points
    - –Keep only those that are currently speaking
- **As the wireless card is in 'master' mode, all fields are mastered by the driver and firmware, especially**
  - Sequence number
  - BSS Timestamp
  - Supported capabilities (tagged parameters)
- **So what?**

`fakeap.pl` (3/4)

`fakeap.pl` could be detected

- Load of ESSIDs with (sometimes) funny ones ;-)
- Resetted BSS Timestamps*
  - A flood of low BSS timestamps from different sources is a clear sign of a `fakeap.pl` attack
- (Sometimes) Resetted sequence numbers
  - At the beginning of the attack
- Same tagged parameters for different beacons in a time period
  - Layer 2 fingerprinting of the attacker wireless card

* hint from Joshua Wright

`fakeap.pl` (4/4)

- `fakeap.pl` pcap capture file
- Take a look at BSS timestamps and tagged parameters…

Fakeap.cap

Wireless IDS and Fake APs…

- **Wireless IDS should have `fakeap.pl` detection engines**
  - Latter slides show means to achieve a good level of detection

- **But, if the attacker has `RAW` injection capabilities**
  - It could be a severe hurt for Wireless IDS and stumblers

Important Notice!

- **All code is in alpha/beta stage**
  - Raw Fake AP is fully functional
  - Raw Glue AP is in alpha stage (need to be extensively tested)
  - Raw Covert is fully functional but quite unuseful without extended capabilities (file transfer, remote shell)
- **These tools were developed for**
  - Wireless IDS testing
  - Proof-of-concept purposes
  - Showing how `RAW` injection could be powerful!
  - Fun! ;-)
- **Will be released under the GPL license…**

Raw Fake AP (1/7)

- **What about `RAW` injection in monitor mode?**
  - Today, supported by (most) wireless chipsets, firmwares and drivers
- **Could help for a 'Raw Fake AP'…**
  - A program that emulates IEEE 802.11 access points thanks to wireless raw injection
  - Only Probe Response and Beacon frames are supported
  - Going towards other management frames could lead to a (rather) complete Virtual AP…
- **Check for next slides…**

Raw Fake AP (2/7)

Some features

- Raw injection of beacon and probe response frames in monitor mode
- Try to forge coherent sequence numbers and BSS timestamps
  - (depending on driver injection capabilities)
- Try to have a coherent time interval between beacons
  - (which is hard to achieve without a real time kernel)
- Supports multiple capabilities advertisements
  - (cryptoprotocols like WPA/RSN, radio capabilities like data rates)

Raw Fake AP (3/7)

- **Should not be detected as a Fake AP attacks thanks to**
  - Coherent BSS Timestamps and sequence numbers
  - Emulated access points will constantly speak
- **Will test your wireless IDS**
  - Garbage data (invalid characters), high number of access points…
  - Becomes really hard for a wireless IDS to classify this as a Fake AP activity
- **Will hide your real networks from (novice) wardrivers**
  - How to distinguish between valid and emulated access points?
  - Could be a countermeasure activated by a wireless IDS detecting wardriving activity ;-)

Raw Fake AP (4/7)

Will fool passive and active stumblers / sniffers

- Thanks to advertised beacons regularly sent
- Thanks to probe responses sent back in responses to wireless clients probe requests

Beacon mode

- Choose channel X
- Send beacons of fake access points under channel X
- Switch channel and so on…

Probe response mode

- Wait on channel X for NULL probe requests
- Send back probe responses of fake access points under channel X
- Switch channel and so on…

Raw Fake AP (5/7)

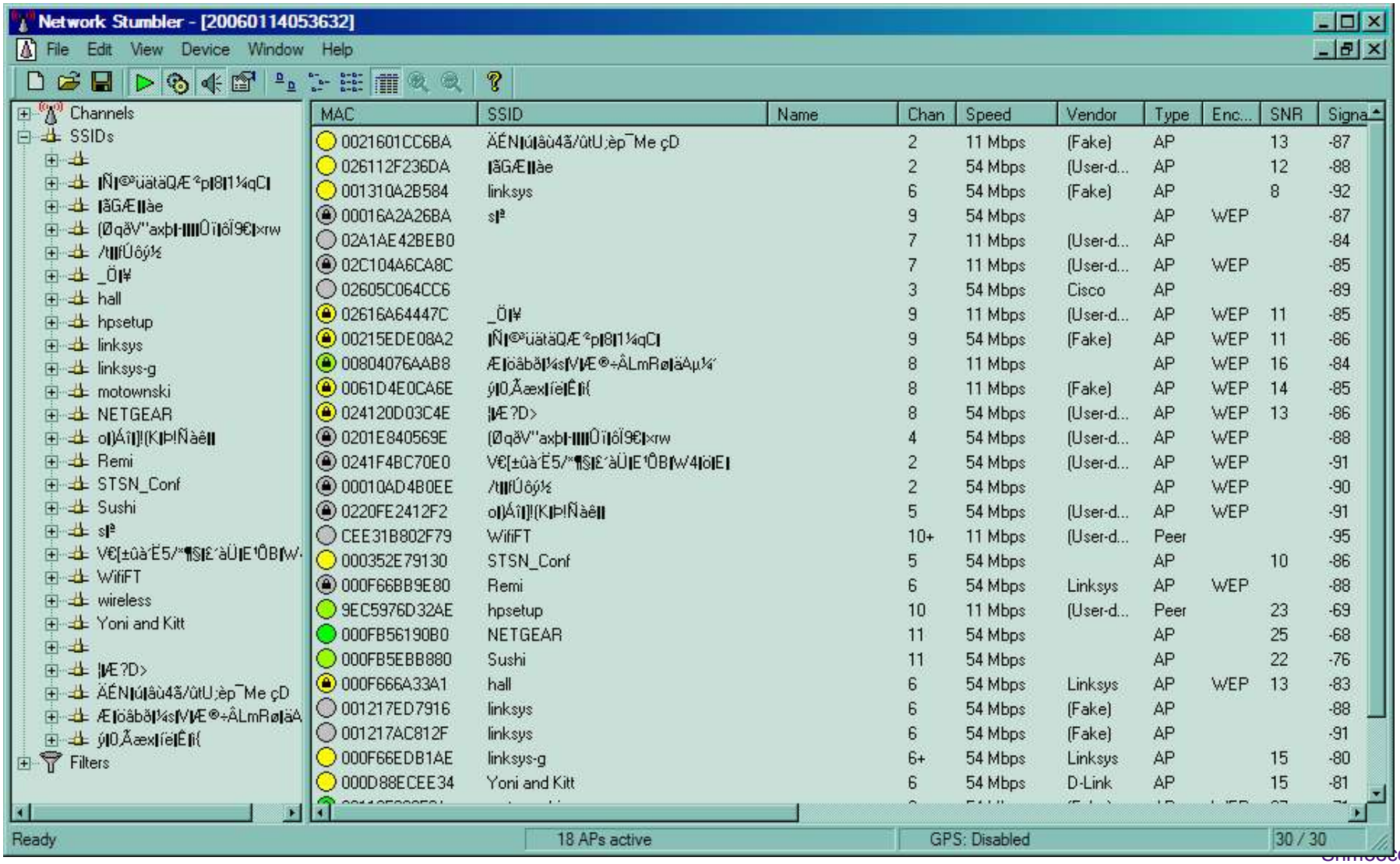
- **Command line interface will help you to choose**
  - Randomize Open/WEP/WPA/RSN crypto
  - Randomize b/g cards
  - Channel hopping
  - TXpower hopping
  - Randomize ESSIDs (allnum or not)
  - Randomize BSSIDs
  - Choose beacon interval
  - Choose number of fake access points
  - Choose a file with valid OUIs
  - Choose a file with ESSIDs
  - Choose between beacon or probe response mode
  - Select a destination MAC address

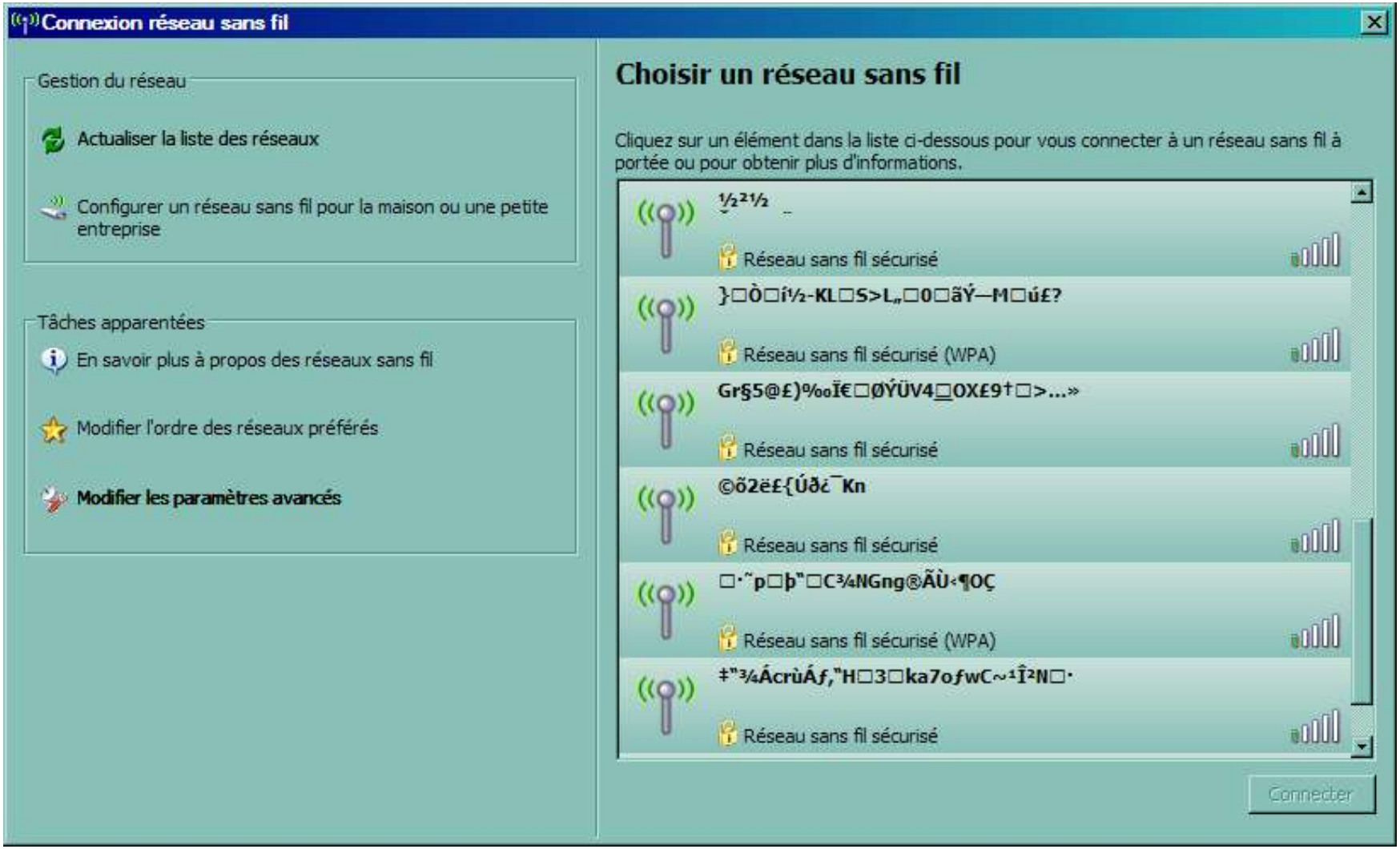
Raw Fake AP (6/7)

- **Proof-of-concept release**
  - Lack of features (no configuration file defining fake access points)
  - Monolithic, non threaded…
  - Do not blame us for ugly coding style!
  - Originally designed to test Wireless IDS and stumblers
- **Released under the GPL licence**

Raw Fake AP (7/7)

Live demo!


Network Stumbler - [20060114053632]
File Edit View Device Window Help
Channels
SSIDs
Filters
MAC SSID Name Chan Speed Vendor Type Enc... SNR Signa
0021601CC6BA 2 11 Mbps (Fake) AP 13 -87
026112F236DA 2 54 Mbps (User-d... AP 12 -88
001310A2B584 linksys 6 54 Mbps (Fake) AP 8 -92
00016A2A26BA 9 54 Mbps AP WEP -87
02A1AE42BEB0 7 11 Mbps (User-d... AP -84
02C104A6CA8C 7 11 Mbps (User-d... AP WEP -85
02605C064CC6 3 54 Mbps Cisco AP -89
02616A64447C 9 11 Mbps (User-d... AP WEP 11 -85
00215EDE08A2 9 54 Mbps (Fake) AP WEP 11 -86
00804076AAB8 8 11 Mbps AP WEP 16 -84
0061D4E0CA6E 8 11 Mbps (Fake) AP WEP 14 -85
024120D03C4E 8 54 Mbps (User-d... AP WEP 13 -86
0201E840569E 4 54 Mbps (User-d... AP WEP -88
0241F4BC70E0 2 54 Mbps (User-d... AP WEP -91
00010AD4B0EE 2 54 Mbps AP WEP -90
0220FE2412F2 5 54 Mbps (User-d... AP WEP -91
CEE31B802F79 WifiFT 10+ 11 Mbps (User-d... Peer -95
000352E79130 STSN_Conf 5 54 Mbps AP 10 -86
000F66BB9E80 Remi 6 54 Mbps Linksys AP WEP -88
9EC5976D32AE hpsetup 10 11 Mbps (User-d... Peer 23 -69
000FB56190B0 NETGEAR 11 54 Mbps AP 25 -68
000FB5EBB880 Sushi 11 54 Mbps AP 22 -76
000F666A33A1 hall 6 54 Mbps Linksys AP WEP 13 -83
001217ED7916 linksys 6 54 Mbps (Fake) AP -88
001217AC812F linksys 6 54 Mbps (Fake) AP -91
000F66EDB1AE linksys-g 6+ 54 Mbps Linksys AP 15 -80
000D88ECEE34 Yoni and Kitt 6 54 Mbps D-Link AP 15 -81
Ready
18 APs active
GPS: Disabled
30 / 30


Connexion réseau sans fil
Gestion du réseau
Actualiser la liste des réseaux
Configurer un réseau sans fil pour la maison ou une petite entreprise
Tâches apparentées
En savoir plus à propos des réseaux sans fil
Modifier l'ordre des réseaux préférés
Modifier les paramètres avancés
Choisir un réseau sans fil
Cliquez sur un élément dans la liste ci-dessous pour vous connecter à un réseau sans fil à portée ou pour obtenir plus d'informations.
½²½ _
Réseau sans fil sécurisé
}□Ò□í½-KL□S>L„□0□ãÝ—M□ú£?
Réseau sans fil sécurisé (WPA)
Gr§5@£)‰Ï€□ØÝÜV4□OX£9†□>...»
Réseau sans fil sécurisé
©õ2ë£{Úð¿¯Kn
Réseau sans fil sécurisé
□·˜p□þ"□C¾NGng®ÃÙ‹¶OÇ
Réseau sans fil sécurisé (WPA)
‡"¾ÁcrùÁƒ,"H□3□ka7oƒwC~¹Î²N□·
Réseau sans fil sécurisé
Connecter

Raw Glue AP (1/6)

- **A fact!**
  - Wireless clients are often the weakest link of any wireless infrastructure
  - They connect to any network or preferred networks (cf. WZC slides)
- **Wireless IDS/IPS (usually) try to mitigate this by**
  - Sending regularly deauthentication / deassociation floods to clients preventing them from associating to rogue access points
- **The purpose of this tool is trying to evaluate another option!**
  - Catch them in a virtual quarantine area!
- **Cf. Attacking Automatic Wireless Network Selection, Dino A. Dai Zovi, Shane A. Macaulay**
  http://www.theta44.org/karma/

Raw Glue AP (2/6)

What about a Virtual AP populating every ESSID?

- Catch probe requests
- Catch authentication and association requests

A kind of Glue AP!

- Once caught, wireless clients may be associated during a certain time to a non existent access point!

Constraint

- Use monitor mode in order to perform both countermeasures and detection
- In order to (eventually) implement it within a wireless IDS/IPS

Raw Glue AP (3/6)

- **NULL probe requests are caught in order to deal with clients with automatic association to any ESSID**
  - A probe response is sent back with chosen BSSID and ESSID
- **Probe requests with a ESSID are caught in order to deal with clients associating to preferred networks**
  - A probe response is sent back with chosen BSSID and asked ESSID
- **Authentication request must be ACKnowledged**
  - And then answer by a successful authentication response
- **Association request must be ACKnowledged**
  - And then answer by a successful association response

Raw Glue AP (4/6)

Proof-of-concept release

- Not really tested ☹
- Not adapted to real world: catch everyone!
- Lack of features (no configuration file for ESSID/BSSID catching)
- Monolithic, non threaded…
- Do not blame us for ugly coding style!

Seems to work on some wireless drivers

- Unstable results, need further improvements
- Estimation of timeouts

Will only work on 'Open' mode

- But Fake APs cannot be in authenticated mode!

Raw Glue AP (5/6)

- **Main difficulties to achieve**
  - `ACK` frames should be sent back within a (small) timeframe (depends on wireless drivers, usually 300 microseconds)
  - Keep-alive packets from the client must be supported
- **Coded in `C` for speed purposes**
- **Will be released under GPL license**

Raw Glue AP (6/6)

- **Live demo!**
- **Who has associated to `30:77:6E:65:64:21`?**

Raw Covert Channel (1/8)

Covert channel

- In information theory, a covert channel is a communications channel that does a writing-between-the-lines form of communication.
- Source: Wikipedia, the free encyclopedia

Writing between-the-lines

- Use valid frames to carry additional information
- Valid frames could be management, control or data frames

This tool is 'only' an example! Possibilities are infinite!

Raw Covert Channel (2/8)

- **With 802.11, this may be performed by many means**
  - Using a proprietary protocol within valid or invalid frames
  - It gives infinite possibilities thanks to **`RAW`** injection
- **(Some) 802.11 frames are not considered as 'malicious'**
  - Control frames like **`ACK`** are lightweight and non suspicious!
    - Frame control (16 bits)
    - Duration Field (16 bits)
    - Receiver Address (48 bits)
  - (Usually) not analyzed by wireless IDS
    - No source nor BSSID addresses ;-)
- **(Some) 802.11 drivers do not give back `ACK` frames in monitor mode (managed in the firmware: e.g. HostAP)**
  - Increasing stealthyness

Raw Covert Channel (3/8)

How it works?

- A client encodes the information and sends **ACKs** over the air
- A server listens for **ACKs** and tries to decode the information

Basically, it uses a magic number in receiver address

- 2 bytes

Basically, it encodes the covert channel in receiver address

- 1 byte

Several ACK frames are needed to send information

Raw Covert Channel (4/8)

Issues

- `ACK` frames can be missed, wireless is not a reliable medium! ;-)
- Detection may be performed (only) with anomaly detection

Proof-of-concept release

- No enhanced features

Will be released under GPL license

Raw Covert Channel (5/8)

Possible enhancements

- Multiple encoding techniques
- Encryption techniques
- Remote shell
- File transfer
- Use invalid frames (see next slide)

Raw Covert Channel (6/8)

- **Invalid frames (in the 802.11 sense, i.e. proprietary frames)**
  - But should be detected by any wireless IDS performing sanity check on every frame
- **FCS invalid frames**
  - Should require driver/firmware modifications to inject bad FCS
  - Wireless IDSes do not analyze such bad frames
  - But should be detected with FCSerr statistics (even if harder to diagnose as a covert channel)

Raw Covert Channel (7/8)

Invalid FCS monitoring

- Usually a bit is set by the firmware when a FCS is invalid
- Most drivers discard packets with bad FCS thanks to this information
  - `HAL_RXERR_CRC` for madwifi
  - `rfmon_header->flags & 0x01` for prism54
- HostAP driver has a facility
  - `prism2_param interface monitor_allow_fcserr 1`

Raw Covert Channel (8/8)

- **Live demo!**
- **Did you detected it? ;-)**

Questions?

Thanks for your attention

References

- Attacking Automatic Wireless Network Selection, Dino A. Dai Zovi, Shane A. Macaulay
  http://www.theta44.org/karma/
- Fake AP, http://www.blackalchemy.to/project/fakeap/
- Kismet, http://www.kismetwireless.net/
- Netstumbler, http://www.netstumbler.com/
- Ethereal, http://www.ethereal.com/
- Aircrack, Christophe Devine home page (www.google.com !)

- Tools: to be released at http://rfakeap.tuxfamily.org