

Attacking WiFi networks with traffic injection

Why open and WEP 802.11 networks really suck

Cédric BLANCHER

cedric.blancher@eads.net
EADS Corporate Research Center
EADS/CCR/DCR/SSI

sid@rstack.org
Rstack Team
<http://sid.rstack.org/>

Libre Software Meeting - Dijon - France
Security Topic
5-9 July 2005

<http://www.rencontresmondiales.org/>



Agenda

- 1 Introduction
- 2 Really quick 802.11 101
 - WiFi injection basics
- 3 Attacking WiFi networks
 - Where's the police - Managing management traffic
 - In the darkness bind them - Rogue APs
 - Breaking the shell - WEP cracking
 - Let me free - Bypassing captive portals
 - All naked - Attacking stations
- 4 WPA, WPA2 and 802.11i
- 5 Conclusion
- 6 Bibliography

Plan

- 1 Introduction
- 2 Really quick 802.11 101
 - WiFi injection basics
- 3 Attacking WiFi networks
 - Where's the police - Managing management traffic
 - In the darkness bind them - Rogue APs
 - Breaking the shell - WEP cracking
 - Let me free - Bypassing captive portals
 - All naked - Attacking stations
- 4 WPA, WPA2 and 802.11i
- 5 Conclusion
- 6 Bibliography

Introduction

We already know 802.11 networks are weak

- Open networks are prone to any well-known LAN perimeter attack
- WEP is vulnerable

So why this talk ?

Introduction

This talk is yet another "people never learn" story

Goals

- Understand that WiFi open networks are unsecure for users
- Understand that WEP really sucks and should not be used anymore
- Understand that there's no salvation outside WPA/WPA2

Maybe make some people learn something¹, at least (in case they don't know yet)

¹Must see website[ABOB]

Introduction

Traffic injection has changed things

- Increased DoS capabilities
- Dramatically decreased WEP cracking achievement time
- Allows traffic tampering
- Allows stations attacks

But still...

- Most ISPs selling wireless/router/modem boxes only provide WEP support
- Many WiFi compliant devices only support WEP (PSP, Zaurus, etc.)
- Most commercial hotspots are still open networks...

Plan

- 1 Introduction
- 2 Really quick 802.11 101
 - WiFi injection basics
- 3 Attacking WiFi networks
 - Where's the police - Managing management traffic
 - In the darkness bind them - Rogue APs
 - Breaking the shell - WEP cracking
 - Let me free - Bypassing captive portals
 - All naked - Attacking stations
- 4 WPA, WPA2 and 802.11i
- 5 Conclusion
- 6 Bibliography

802.11 basics

802.11[IEEE99] is a wireless communication IEEE standard also known as WiFi and pushed by WiFi Alliance[WIFI] lobby

- CSMA/CA based
- Infrastructure vs. Ad-hoc
- Distribution System (DS)
- Association concept
- Management vs. data traffic

802.11 "early" security

WiFi initial protection scheme is WEP (Wired Equivalent Privacy)

- Authentication through challenge/response (sort of) handshake
- Privacy with RC4 cipher using 24bits IV plus fixed key
- Integrity with ciphered CRC32 on cleartext payload

WEP is still widely deployed :(

- 1 Introduction
- 2 Really quick 802.11 101
 - WiFi injection basics
- 3 Attacking WiFi networks
 - Where's the police - Managing management traffic
 - In the darkness bind them - Rogue APs
 - Breaking the shell - WEP cracking
 - Let me free - Bypassing captive portals
 - All naked - Attacking stations
- 4 WPA, WPA2 and 802.11i
- 5 Conclusion
- 6 Bibliography

Arbitrary frames injection

Quite old but non common fonctionnality

- Needs appropriate firmware
- Needs appropriate driver
- Needs appropriate library/software

Some drivers/libs/tools exist[AIRJ], but most focus on management traffic

Toolkit

Proper adapter and driver

- Hostap[HAP], but apparently limited by both firmware and driver (i.e. needs patch)
- Wlan-ng[WLAN] (patched)
- Atheros/Madwifi[MADW] (patched)
- Intersil Prism54[PR54] (development SVN snapshot)

Atheros seems to be the best chipset

Traffic injection 101

Traffic injection quick HOWTO

- 1 Insert adapter
- 2 Load driver and activate adapter
- 3 Set driver into monitor mode (real 802.11 mode)
- 4 Set appropriate channel
- 5 Open PF_PACKET/RAW socket on interface (Linux only)
- 6 Use your socket and play

Still, you need a 802.11 stack over your socket and/or good libs and tools so you can communicate

Plan

- 1 Introduction
- 2 Really quick 802.11 101
 - WiFi injection basics
- 3 Attacking WiFi networks**
 - Where's the police - Managing management traffic
 - In the darkness bind them - Rogue APs
 - Breaking the shell - WEP cracking
 - Let me free - Bypassing captive portals
 - All naked - Attacking stations
- 4 WPA, WPA2 and 802.11i
- 5 Conclusion
- 6 Bibliography

Disclaimer

All materials described in this talk are for educational and demonstration purposes only.

DO NOT USE THEM ON OTHERS' NETWORKS WITHOUT THEIR AUTHORIZATION

You could break the law and face prosecution...

- 1 Introduction
- 2 Really quick 802.11 101
 - WiFi injection basics
- 3 Attacking WiFi networks**
 - Where's the police - Managing management traffic**
 - In the darkness bind them - Rogue APs
 - Breaking the shell - WEP cracking
 - Let me free - Bypassing captive portals
 - All naked - Attacking stations
- 4 WPA, WPA2 and 802.11i
- 5 Conclusion
- 6 Bibliography

Management traffic

Description

Management traffic :

- is a regulation traffic
- is completely unprotected ! ?

It's a target of choice...

Lots of tools for playing with it

Management traffic

Tampering

You alter DS current state by tampering management traffic

- Reject association requests
- Inject disassociation frame
- Inject fake associations
- Wake up devices in sleep mode
- Etc.

Mainly DoSes...

Management traffic Injection

Management traffic is easy to generate and inject
See Scapy[SCAP] packets classes

- Dot11
- Dot11Disas
- Dot11AssoResp
- Dot11ReassoResp
- Dot11Deauth
- etc.

See Scapy in action[BIO04]

- 1 Introduction
- 2 Really quick 802.11 101
 - WiFi injection basics
- 3 Attacking WiFi networks**
 - Where's the police - Managing management traffic
 - In the darkness bind them - Rogue APs**
 - Breaking the shell - WEP cracking
 - Let me free - Bypassing captive portals
 - All naked - Attacking stations
- 4 WPA, WPA2 and 802.11i
- 5 Conclusion
- 6 Bibliography

Rogue APs

Building AP from scratch

For AP mode, you need to inject

- Beacon frames
- Answers to associations requests
- Management traffic
- Forwarded data frames

Rogue APs

Enter the game

If you can be an AP, you can also be a fake one...

- Cheap solution for low level traffic redirection
- Cool attacks against automatic "WiFi network managers" [KARM]

Rogue AP is the poor man attack that works so well

- 1 Introduction
- 2 Really quick 802.11 101
 - WiFi injection basics
- 3 Attacking WiFi networks**
 - Where's the police - Managing management traffic
 - In the darkness bind them - Rogue APs
 - Breaking the shell - WEP cracking**
 - Let me free - Bypassing captive portals
 - All naked - Attacking stations
- 4 WPA, WPA2 and 802.11i
- 5 Conclusion
- 6 Bibliography

WEP cracking

Attacks overview

WEP is RC4 based, which is XOR based...

- IV collisions
- (Almost) Arbitrary frame injection
- Cleartext attacks (e.g. authentication challenge) and authentication bypass
- PRGA² output/IV couple table construction
- Fluhrer, Mantin and Shamir attack (weak IVs attack)
- Korek optimization of FMS attack based on solved cases
- Korek Chopchop attack

PRGA output/IV and FMS attacks need traffic gathering

²Pseudo Random Generation Algorithm

WEP cracking

IV collisions

First WiFi (in)security paper published in 2000[WAL00]

- Key space is 2^{24} whatever WEP key length
- More than 99% IV collision after only 12000 frames

Let C and C' two cleartexts ciphered using the same key K

Key collision info extraction

$$P = C \oplus RC4(IV \parallel K)$$

$$P' = C' \oplus RC4(IV \parallel K)$$

$$\Rightarrow P \oplus P' = C \oplus C'$$

RC4 weak keys problem mentionned[RW95]

WEP cracking

Modified frame injection

Let C be our cleartext message and C' a modification of C
Let $Mod = C \oplus C'$

Arbitrary message modification

$$\begin{aligned} P &= WEP(C \parallel ICV(C)) \\ &= (C \parallel ICV(C)) \oplus RC4(IV \parallel K) \\ P' &= (C' \parallel ICV(C')) \oplus RC4(IV \parallel K) \\ &= (C \parallel ICV(C)) \oplus RC4(IV \parallel K) \oplus (Mod \parallel ICV(Mod)) \\ &= P \oplus (Mod \parallel ICV(Mod)) \end{aligned}$$

This means you can inject arbitrary layer 2 consistent WEP frames and have them decrypted...

WEP cracking

Arbitrary injection consequences

We can inject arbitrary traffic through WEP without key knowledge

- Launch oracle based attacks
- Stimulate network in order to create traffic

Full WEP cracking is no more relying on passive listening

WEP cracking

Cleartext attack

WEP authentication is vulnerable to cleartext attack
Let C be a cleartext challenge.

PRGA extraction

$$\begin{aligned} P &= WEP(C \parallel ICV(C)) \\ &= (C \parallel ICV(C)) \oplus RC4(IV \parallel K) \\ \Rightarrow RC4(IV \parallel K) &= P \oplus (C \parallel ICV(C)) \end{aligned}$$

Payload header is 8 bytes, C is 128 bytes and $ICV(C)$ is 4 bytes
So we can grab 140 bytes of PRGA output for given IV

Authentication bypass

"Your 802.11 Wireless Network Has No Clothes" [ASW01]

Challenge answer computation

$$P' = (C' \parallel ICV(C')) \oplus RC4(IV \parallel K)$$

Once one authentication is captured, we can compute any further answer P' to challenge C' using known PRGA output

PRGA output/IV tables

For every IV, grab PRGA output

- We know how to grab 140 bytes of PRGA output
- We can generate traffic with known PRGA output (e.g. GET / HTTP/1.0)
- We can have traffic generated and grab longer PRGA output (e.g. HTTP reply)

We can end up with a huge PRGA output/IV table ($\approx 25\text{GB}$)
allowing one to decrypt any packet on the air

We can boost this attack playing with disassociations :)

WEP cracking

Fluhrer, Mantin and Shamir attack

Article "Weaknesses in the Key Scheduling Algorithm of RC4" [FMS01], based on Roos and Wagner work

- Weak key = info about internal RC4 state
- Weak key + known first bytes of stream = info about K

So, what do we have ?

- RC4 key is $IV || K$ and IV is known
- C is a 802.11 frame, so we can guess first bytes

We have "known weak IVs" that provide informations about K and lead to an effective attack against WEP

Korek added other "solved cases" [KO04a]

WEP cracking

Korek Chopchop attack

Arbaugh first published an inductive attack against WEP[ARB01]
Korek published a similar (reversed) inductive attack[KO04b] with
a PoC called Chopchop

- 1 Grab a multicast/broadcast frame
- 2 Strip the last data byte
- 3 Assume last byte cleartext value
- 4 Correct frame ICV and reinject
- 5 See if AP forwards the new frame

Extremely effective on ARP traffic (10-20s per packet).

WEP cracking

Devine aircrack/aireplay WEP cracking

Using FMS and Korek optimizations, Christophe Devine released aircrack and aireplay[AIRC]

- 1 Capture an ARP request, optionnaly with Chopchop
- 2 Inject ARP request again and again
- 3 Stimulate traffic and unique IV collection
- 4 Crack WEP key with optimized FMS

Full WEP cracking is now a matter of minutes[WWCR]

And aircrack can still get optimized...

- 1 Introduction
- 2 Really quick 802.11 101
 - WiFi injection basics
- 3 Attacking WiFi networks**
 - Where's the police - Managing management traffic
 - In the darkness bind them - Rogue APs
 - Breaking the shell - WEP cracking
 - Let me free - Bypassing captive portals**
 - All naked - Attacking stations
- 4 WPA, WPA2 and 802.11i
- 5 Conclusion
- 6 Bibliography

Bypassing captive portals

Commercial WiFi hospots

Commercial public Internet access

- Captive portal based system
- Authentication to billing system through web portal
- Authorization for Internet access
- Authorization tracking

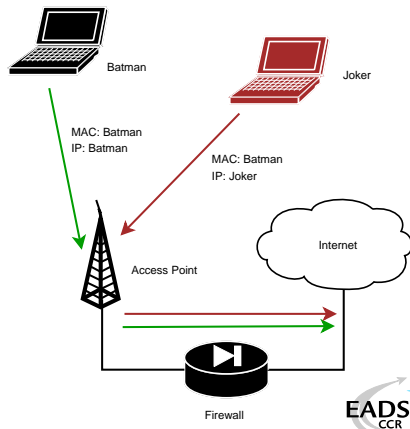
It would be nice to be free... For free !

Bypassing captive portals

MAC based authorization tracking

Authorized clients are identified by their MAC address

- MAC address is easy to spoof
- No MAC layer conflict on WiFi network
- Just need a different IP



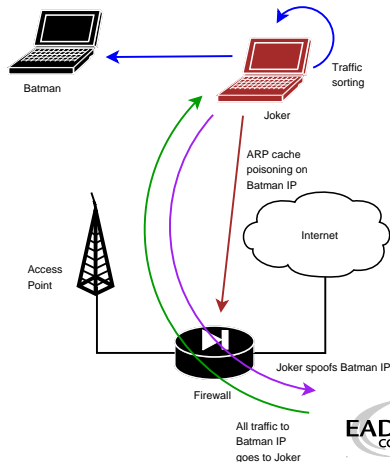
Bypassing captive portals

IP based authorization tracking

Authorized clients are identified by their IP address

- IP address are just a little more tricky to spoof
- ARP cache poisoning helps redirecting traffic
- Traffic redirection allows IP spoofing

See my LSM 2002 talk[BLA02], arp-sk website[ARPS] or MISC3[MISC]

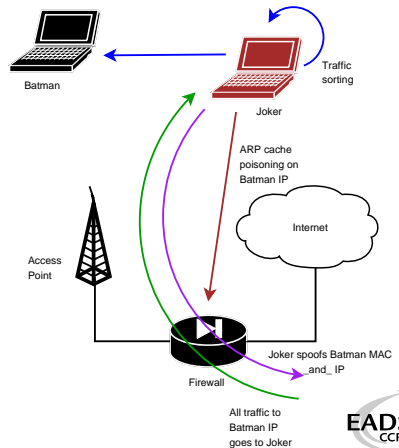


Bypassing captive portals

MAC+IP addresses based authorization tracking

The smart way for tracking people ?

- Previous technic won't help because of MAC address checking
- Send traffic with spoofed MAC address
- ARP cache poisoning and IP spoofing
- Hint : IP layer and MAC layer don't care much about each other



Bypassing captive portals

Misconfiguration and tricks

Some gateways are misconfigured

- HTTP proxy left open on gateway
- ESTABLISHED,RELATED -j ACCEPT prevents connections drop when authorization expires on Linux based systems
- Administration network on the same VLAN, accessible through WiFi
- Etc.

Misconfigurations tend to be less and less common
Nevertheless, DNS based communication[OZY] or tunneling[NSTX] always works :)

- 1 Introduction
- 2 Really quick 802.11 101
 - WiFi injection basics
- 3 Attacking WiFi networks**
 - Where's the police - Managing management traffic
 - In the darkness bind them - Rogue APs
 - Breaking the shell - WEP cracking
 - Let me free - Bypassing captive portals
 - All naked - Attacking stations**
- 4 WPA, WPA2 and 802.11i
- 5 Conclusion
- 6 Bibliography

Attacking stations

What about associated stations ?

Associated stations are almost naked

- LAN attacks (ARP, DHCP, DNS, etc.)
- Traffic interception and tampering
- Direct station attacks

Think of personal firewalls exception for local network...

Attacking stations

Station to station traffic prevention

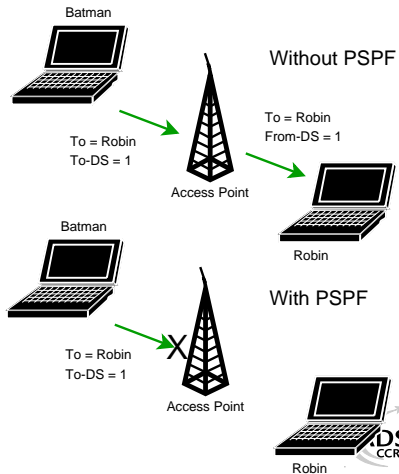
Security feature that blocks traffic within DS

Cisco calls this PSPF, each vendor has it's own name/flavor

- Station sends To-DS frame
- AP sees it's destined to DS
- AP drops the frame

No From-DS frame, so no communication^a : stations can't talk to each other...

^aDoes not work between 2 APs linked via wired network



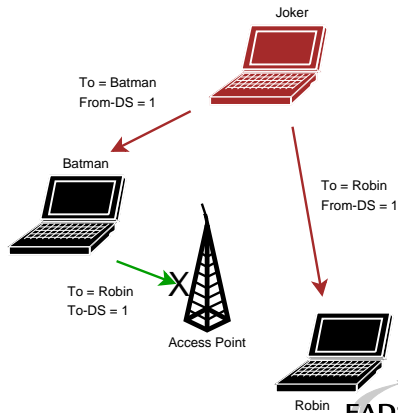
Attacking stations

PSPF bypass with injection

Joker can inject From-DS frames directly

- No need for AP benediction
- You can spoof about anyone
- You're still able to sniff traffic

Traffic injection allows complete PSPF bypass



Attacking stations

Traffic tampering with injection

WiFi communication are just opened on the air

- Listen to WiFi traffic
- Match interesting requests
- Spoof the AP and inject your own answers
- Clap clap, you've done airpwn-like[AIRP] tool

Only think of injecting nasty stuff in HTTP traffic, just in case someone would dare to use MSIE on an open WLAN

Attacking stations

Full communication with injection

Sending traffic directly to stations without AP authorization

- Allows station to station communication
- Allows communicating if AP is out of reach
- Allows communication if AP refuses association

A smart way for talking to stations without being associated

Attacking stations

Proof of concept : Wifitap

Needed a PoC for PSPF-like systems bypass and wrote Wifitap

- Written in Python[PYTH]
- Relies on Scapy[SCAP]
- Uses tuntap device and OS IP stack
- Use WiFi frame injection and sniffing

Wifitap allows communication with station despite of AP restrictions

Attacking stations

Wifitap usage

```
# ./wifitap.py -h
Usage: wifitap -b <BSSID> [-o <iface>] [-i <iface> [-p]]
        [-w <WEP key> [-k <key id>]]
        [-d [-v]] [-h]

-b <BSSID>      specify BSSID for injection
-o <iface>      specify interface for injection
-i <iface>      specify interface for listening
-p             No Prism Headers in capture
-w <key>        WEP mode and key
-k <key id>     WEP key id (default: 0)
-d             activate debug
-v             verbose debugging
-h             this so helpful output
```



Attacking stations

Wifitap in short

How Wifitap works

Sending traffic

- Read ethernet from tuntap
- Add 802.11 headers
- Add BSSID, From-DS and WEP
- Inject frame over WiFi

Receiving traffic

- Sniff 802.11 frame
- Remove WEP layer if needed
- Remove 802.11 headers
- Send ethernet through tuntap

Attacker does not need to be associated

Attacking stations

Quick demo...

We Proudly R3wt



Download Wifitap at

http://sid.rstack.org/index.php/Wifitap_EN

Hotspots with PSPF-like

Some hotspots implement PSPF-like in order to prevent clients from attacking each other

- Does not protect against "session" hijacking³
- Attacker then needs to take over victim's session
- Victim does not have access anymore, and still pays for it

And among all, it's pretty useless...

³Side effect : tools like arpspoof won't work

More hotspot bypassing...

Hijacking people authorization is not very kind

- Use Wifitap to bypass PSPF-like
- Now you can send your poor victim his traffic back

Your victim and you are both able to surf transparently

Now, you "can be a true gentlemanly [h|cr]acker" [ISCD];)

Plan

- 1 Introduction
- 2 Really quick 802.11 101
 - WiFi injection basics
- 3 Attacking WiFi networks
 - Where's the police - Managing management traffic
 - In the darkness bind them - Rogue APs
 - Breaking the shell - WEP cracking
 - Let me free - Bypassing captive portals
 - All naked - Attacking stations
- 4 WPA, WPA2 and 802.11i
- 5 Conclusion
- 6 Bibliography

WPA

Transitional recommendation[WPA] from WiFi Alliance (2003)
extracted from IEEE work for infrastructure networks only

- New authentication scheme based on PSK or 802.1x
- New key generation and scheduling scheme for keys
- New integrity check through SHA1 based MIC with sequencing

Pretty solid solution that can prevent injection/replay

WPA2 and 802.11i

802.11i[IEEE04b] is a standard from IEEE for WiFi security
WPA2[WPA2] is a recommendation from WiFi Alliance based on 802.11i

- RSN⁴ concept : security algorithms negotiation
- Integrates Ad-Hoc security
- Authentication using 802.1x
- Ciphering using AES-CCMP
- Integrity check using CCMP MIC

Return to the roots and use of a real adapted ciphering solution

⁴Robust Security Network

WPA/WPA2 using Free Software

Building WPA/WPA2 aware network with free software

Client side

- wpa_supplicant[WPAS]
- WPA/WPA2/RSN supplicant
- Linux, BSD and...
Win32 :)

SoftAP side

- hostapd[HAPD]
- WPA/WPA2/RSN and
802.1x[IEEE04a]
authenticator
- Linux, BSD

Some flaws already

Yet some papers have been published regarding WPA security

- WPA weak PSK (<20 chars) bruteforce[MOS03]
- Injection of spoofed first handshake message leads to memory exhaustion[HM04] (DOS)
- TEK attack in 2^{105} instead of 2^{128} (requires key knowledge)[MRH04]
- Counter-measures abuse (DOS) : traffic replay, dumb traffic injection

Moreover, nothing will ever protect from layer 1 based DoS attacks (bandwidth reservation, jamming)

And then ?

Although some flaws, WPA provides strong mechanisms for end users

- Good authentication mechanisms if properly used
- Real session management
- Session key management and re-keying
- Real integrity check
- Anti-replay, anti-injection mechanisms

WPA2 is even better.

Plan

- 1 Introduction
- 2 Really quick 802.11 101
 - WiFi injection basics
- 3 Attacking WiFi networks
 - Where's the police - Managing management traffic
 - In the darkness bind them - Rogue APs
 - Breaking the shell - WEP cracking
 - Let me free - Bypassing captive portals
 - All naked - Attacking stations
- 4 WPA, WPA2 and 802.11i
- 5 **Conclusion**
- 6 Bibliography

Conclusion

WiFi environnement are highly insecure and tough to secure
You just can't cope with amateur style protection...

Then...

- Don't use WEP anymore, it "has no clothes" at all
- Don't use open networks for public access, use WPA/WPA2^a
- Migrate to WPA, then WPA2 as soon as possible

^aBTW, RADIUS is far better for AAA

Manufacturers, vendors, journalists, etc. : stop telling people WEP is OK! It's not at all!

Maybe ending WEP support would be a good idea...

Thank you for your attention and...

Greetings to...

- EADS CCR/DCR/SSI team
- **Rstack.org** team
<http://www.rstack.org/>
- **MISC Magazine**
<http://www.miscmag.com/>
- **French Honeynet Project**
<http://www.frenchhoneynet.org/>



Download these slides from <http://sid.rstack.org/>

Plan

- 1 Introduction
- 2 Really quick 802.11 101
 - WiFi injection basics
- 3 Attacking WiFi networks
 - Where's the police - Managing management traffic
 - In the darkness bind them - Rogue APs
 - Breaking the shell - WEP cracking
 - Let me free - Bypassing captive portals
 - All naked - Attacking stations
- 4 WPA, WPA2 and 802.11i
- 5 Conclusion
- 6 Bibliography

Bibliography I



[IEEE04a] IEEE Std 802.1x, Port-Based Network Access Control, 2004,

<http://standards.ieee.org/getieee802/download/802.1X-20>



[IEEE99] ANSI/IEEE Std 802.11, Wireless LAN Medium Access Control and Physical Layer Specifications, 1999,

<http://standards.ieee.org/getieee802/download/802.11-19>



[IEEE04b] IEEE Std 802.11i, Medium Access Control Security Enhancements, 2004,





<http://standards.ieee.org/getieee802/download/802.11i-2>



[WPA] WiFi Protected Access,

http://www.wi-fi.org/OpenSection/protected_access

Bibliography II

-  [WPA2] WiFi Protected Access 2,
http://www.wi-fi.org/OpenSection/protected_access.asp
-  [RW95] A. Roos and D.A. Wagner, Weak keys in RC4,
sci.crypt Usenet newsgroup
-  [WAL00] J. Walker, Unsafe at any key size; An analysis of
WEP encapsulation, 2000,
<http://www.dis.org/wl/pdf/unsafew.pdf>
-  [ASW01] W.A. Arbaugh, N. Shankar and Y.C.J. Wan, Your
802.11 Wireless Network Has No Clothes, 2001,
<http://www.cs.umd.edu/~waa/wireless.pdf>

Bibliography III



[FMS01] S. Fluhrer, I. Mantin and A. Shamir, Weaknesses in the Key Scheduling Algorithm of RC4, 2001,
http://www.drizzle.com/~aboba/IEEE/rc4_ksaproc.pdf








[MOS03] R. Moskowitz, Weakness in Passphrase Choice in WPA Interface, 2003,
<http://wifinetnews.com/archives/002452.html>






[HM04] C. He and J.C. Mitchell, 1 Message Attack on 4-Way Handshake, 2004,
<http://www.drizzle.com/~aboba/IEEE/11-04-0497-00-000i-1>

Bibliography IV

-  [MRH04] V. Moen, H. Raddum and K.J. Hole, Weakness in the Temporal Key Hash of WPA, 2004,
http://www.nowires.org/Papers-PDF/WPA_attack.pdf
-  [ABOB] Bernard Aboba, The Unofficial 802.11 Security Web Page, <http://www.drizzle.com/~aboba/IEEE/>
-  [WIFI] WiFi Alliance, <http://www.wi-fi.org/>
-  [MISC] MISC Magazine, <http://www.miscmag.com>
-  [WWCR] Cracking WEP in 10 minutes with Whoppix,
<http://whoppix.hackingdefined.com/Whoppix-wepcrack.html>

Bibliography V

-  [ARB01] W.A. Arbaugh, An Inductive Chosen Plaintext Attack against WEP/WEP2, 2001,
<http://www.cs.umd.edu/~waa/attack/v3dcmnt.htm>
-  [BIO04] P. Biondi, Packet generation and network based attacks with Scapy, 2004,
http://www.secdev.org/conf/scapy_csw05.pdf
-  [BLA02] C. Blancher, Switched environments security, a fairy tale, 2002,
http://sid.rstack.org/pres/0207_LSM02_ARP.pdf

Bibliography VI

-  [BLA03] C. Blancher, Layer 2 filtering and transparent firewalling, 2003
http://sid.rstack.org/pres/0307_LSM03_L2_Filter.pdf
-  [KO04a] Korek,
<http://www.netstumbler.org/showthread.php?p=89036>
-  [KO04b] Korek, Chopchop,
<http://www.netstumbler.org/showthread.php?t=12489>
-  [AIRC] C. Devine, Aircrack,
<http://www.cr0.net:8040/code/network/aircrack/>
-  [AIRJ] Airjack,
<http://sourceforge.net/projects/airjack/>

Bibliography VII

-  [AIRP] Airpwn, <http://www.evilscheme.org/defcon/>
-  [ARPS] Arp-sk, <http://www.apr-sk.org/>
-  [EBT] Ebttables, <http://ebtables.sourceforge.net/>
-  [HAP] Hostap Linux driver, <http://hostap.epitest.fi/>
-  [HAPD] Hostapd authenticator,
<http://hostap.epitest.fi/hostapd/>
-  [KARM] Karma, <http://theta44.org/karma/>
-  [MADW] MadWiFi project,
<http://madwifi.sourceforge.net/>

Bibliography VIII

-  [NSTX] Nstx, <http://nstx.dereference.de/nstx/>
-  [OZY] OzymanDNS,
http://www.doxpara.com/ozymandns_src_0.1.tgz
-  [PR54] Prism54 Linux driver, <http://prism54.org/>
-  [PYTH] Python, <http://www.python.org/>
-  [SCAP] Scapy, <http://www.secdev.org/projects/scapy/>
-  [WLAN] Linux Wlan-ng, <http://www.linux-wlan.org/>
-  [WPAS] Wpa_supplicant,
http://hostap.epitest.fi/wpa_supplicant/

Bibliography IX



[WTAP] Wifitap,

http://sid.rstack.org/index.php/Wifitap_EN



[ISCD] ISC Handler's Diary,

<http://isc.sans.org/diary.php?date=2005-06-26>