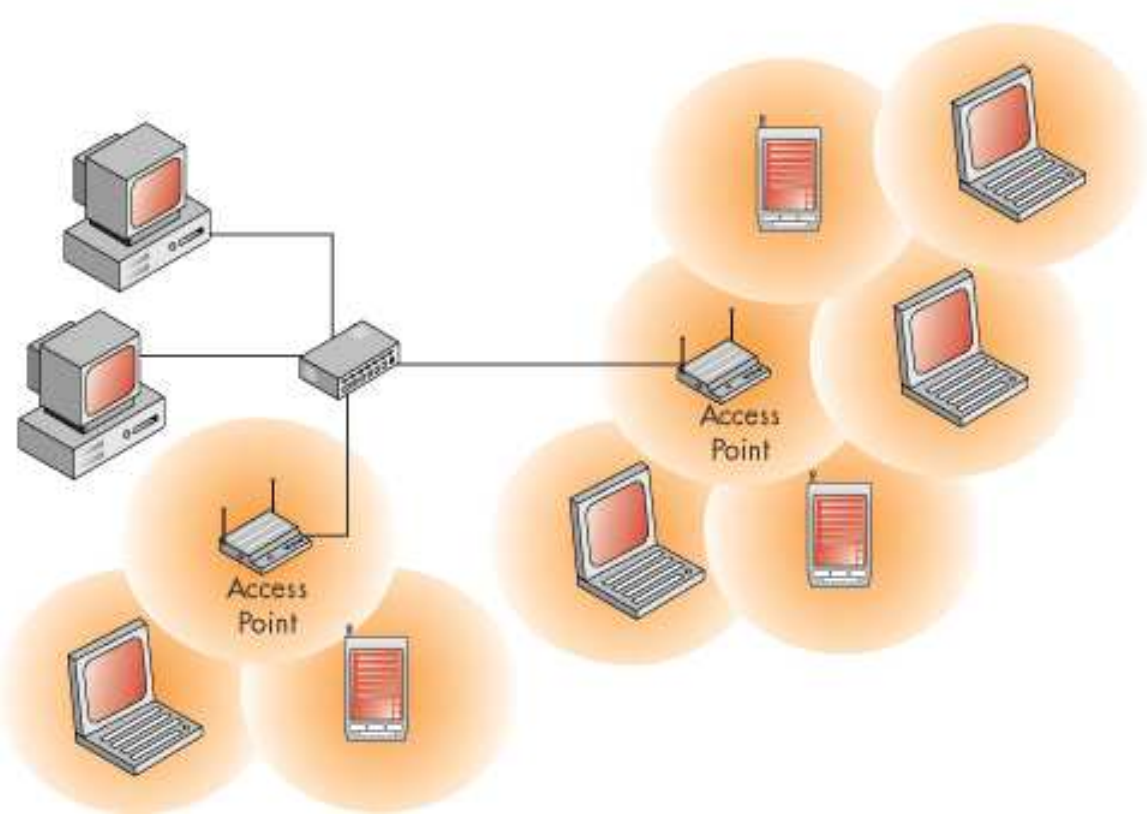
Tesina di
Sicurezza delle reti e commercio elettronico
Docente: Antonio Lioy
Relatore: Marco Aime

Wardriving e network auditing



Bovero Matteo Onofrio
Matricola: 93152

Typeset with LaTeX

matteo@assi.polito.it
http://www.voyager.it

Indice

1 Reti Wireless **1**
- 1.1 Apparati . . . 1
- 1.2 Crittografia . . . 2
  - 1.2.1 Descrizione . . . 3

2 Tool di auditing **5**
- 2.1 Apparecchiature . . . 5
- 2.2 Tool per Linux . . . 7
  - 2.2.1 Kismet . . . 7
  - 2.2.2 Airtraf . . . 17
  - 2.2.3 Airsnort . . . 18
- 2.3 Tool per iPaq . . . 22
  - 2.3.1 Kismet . . . 22
  - 2.3.2 Wscan . . . 22
  - 2.3.3 PrismStumbler . . . 23
- 2.4 Utilizzo e comparazione dei software . . . 25
- 2.5 Tabelle riassuntiva . . . 26

3 Possibili attacchi **28**
- 3.1 WEP: introduzione . . . 28
- 3.2 Autenticazione del messaggio . . . 29
  - 3.2.1 Modifica del messaggio . . . 29
  - 3.2.2 Generazione di messaggi “autentici” . . . 30
  - 3.2.3 Spoofing dell’autenticazione . . . 30
  - 3.2.4 Decodifica del messaggio . . . 31
- 3.3 Possibili rimedi . . . 32

4 Appendice A: formati di cattura di Kismet **33**
- 4.1 Cattura in formato network . . . 33
- 4.2 Cattura in formato csv . . . 33
- 4.3 Cattura in formato xml . . . 34

5 Appendice B: installazione Ipaq **41**

6 Riferimenti **44**

Elenco delle tabelle

2 Comparazione software per pc 27
3 Attacchi a forza bruta[2, B] 29

Elenco delle figure

1 Rete ad infrastruttura 2
2 Schema di crittografazione tramite wep 3
3 Schermata principale di Kismet 9
4 Schermata principale di Kismet (massimizzata) 9
5 Possibili ordinamenti delle reti “scoperte” 10
6 Elenco dei server 11
7 Frequenza di arrivo 12
8 Statistiche sui pacchetti 13
9 Informazioni sull’access point 13
10 Informazioni sui singoli pacchetti 14
11 Selezione di un gruppo 14
12 Client collegati ad un AP 15
13 Dettagli su un client 15
14 Schermata principale di Airtraf 18
15 Scansione delle reti wireless 19
16 Visualizzazione delle reti disponibili 19
17 Selezione della rete 20
18 Dettagli sulla rete selezionata 20
19 Informazioni sui pacchetti 21
20 Statistiche sullo stack tcp 21
21 Impostazioni di acquisizione 22
22 Interfaccia di Airsnort 22
23 Interfaccia di Wscan 23
24 Interfaccia di PrismStumbler 24

1 Reti Wireless

Le reti wireless sono standardizzate dall'IEEE con il codice 802.11. La prima versione fu redatta nel 1997, oggi si sta lavorando alla versione g:

	802.11	802.11b	802.11a	802.11g
Data riferimento	1997	1999	1999	2001-oggi
Velocità (a livello fisico, in Mbps)	1 e 2	11	6, 9, 12, 18, 24, 36, 48, 54	54
Velocità (a livello 3, in Mbps)	1.2	5	32	54
Frequenza di funzionamento (GHz)	2.4	2.4	5	2.4
Modulazione	DSSS	DSSS	OFDM	OFDM
Stardardizzato come		WiFi	WiFi5	

1.1 Apparati

Gli apparati necessari per implementare una rete wireless si dividono in due categorie: gli access point e le stazioni. Gli access point sono apparati con una doppia interfaccia di rete: la prima permette il collegamento alla rete fisica, mentre la seconda serve per le connessioni wireless. Le stazioni sono le macchine (pc, palmari, etc...) che necessitano il collegamento alla rete tramite l'access point. In questo caso si parla di **reti ad infrastruttura**. Il caso opposto, cioè reti che non necessitano di un access point per la comunicazione tra le varie macchine, vengono chiamate **reti ad-hoc**.

Da qui in poi, con il termine rete, si intederà rete ad infrastruttura.

Ogni rete dispone di un Service Set Identifier (SSID), che in pratica è il nome della rete ed è composto da 32 ottetti. Inoltre c'è il Basic Service Set Identifier (BSSI) che rappresenta l'identificativo della cella e corrisponde al mac dell'access point.

Senza scendere nel dettaglio dalla struttura dei singoli pacchetti, alcuni dei più caratteristici sono:

1. Probe Request: richiesta di sincronizzazione;
2. Probe Response: risposta ad un probe;
3. Beacon: pacchetti inviati periodicamente dagli access point per la sincronizzazione.

Durante l'inizializzazione della scheda wireless in una stazione, vengono cercate le reti disponibili. Nel caso ce ne siano più di una, si seleziona quella con cui si comunica meglio. Questo può essere fatto tramite due modalità di scan:

- attivo: invio di probe request sui canali e analisi della risposta;

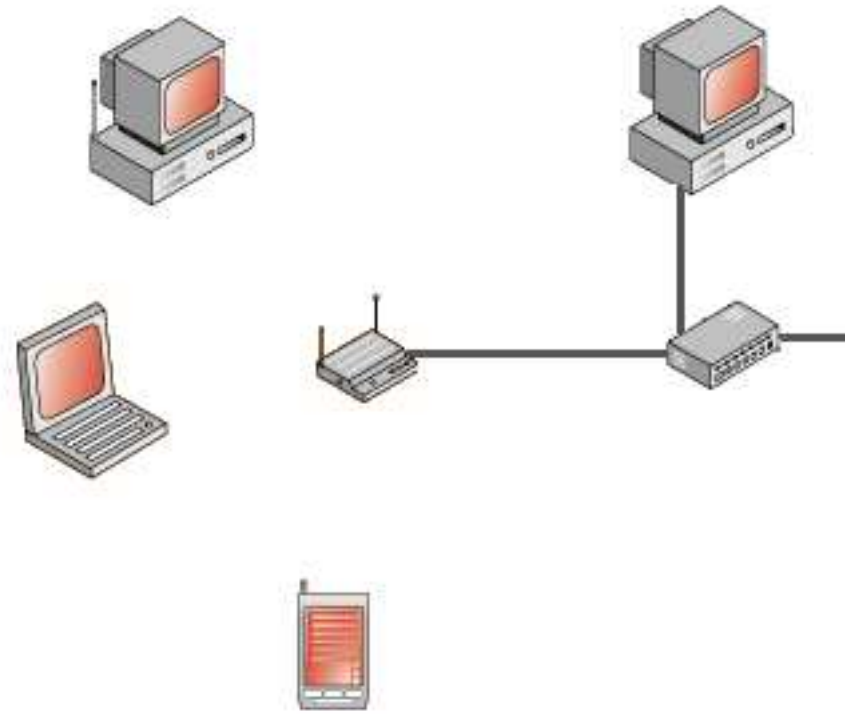


Figura 1: Rete ad infrastruttura

- passivo: verifica della qualità del canale tramite i messaggi di beacon.

Gli access point tengono traccia di tutte le stazioni connesse.

La situazione appena descritta si verifica quando non viene impostato il protocollo di autenticazione. Attualmente il protocollo di autenticazione standard si basa su un segreto condiviso, impostato come Extended Service Set Identifier (ESSID). Se questo è impostato, le stazioni cercano il miglio access point con l'ESSID specificato.

1.2 Crittografia

Lo standard 802.11 prevede la possibilità di crittografare i pacchetti con il protocollo WEP - Wired Equivalent Privacy.

Gli obbiettivi di questo protocollo sono tre:

1. privacy: il suo scopo principale è di evitare la possibilità che estranei possano accedere a dati sensibili

2. access control: garantisce l'accesso alla rete solo al personale autorizzato (infatti il protocollo 802.11 prevede la possibilità di rifiutare i pacchetti non crittografati)

3. integrità dei dati: il protocollo è stato pensato per rilevare la modifica di pacchetti effettuata da terze parti, per questo un checksum è incluso nel pacchetto.

1.2.1 Descrizione

Il WEP si basa su un segreto condiviso ***k*** per la comunicazione tra le parti per proteggere i dati trasmessi[2, W].

La crittografazione del frame avviene in questo modo:

- checksum: dato un messaggio ***M*** si calcola il checksum ***c(M)*** e lo si concatena al messaggio originale ***P = [M,c(M)]***. Da notare che nè ***c(M)*** nè ***P*** dipendono da ***k***.

- crittografazione: viene crittografato il messaggio ***P*** utilizzando l'algoritmo RC4 con la chiave ***k*** e un vettore di inizializzazione ***iv***: ***RC4(v,k)***. Quindi si esegue un xor tra il risultato di quest'ultima operazione e ***P***: ***C=P⊕RC4(v,k)***

- trasmissione: viene trasmetto il vettore di inizializzazione (***iv***) con il testo cifrato (***C***).

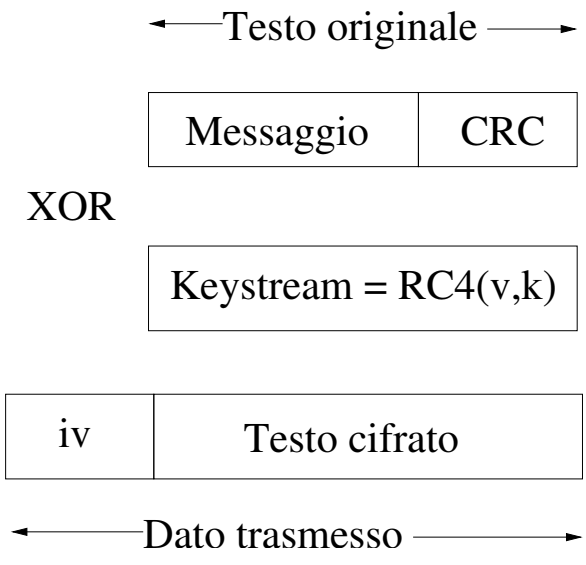


Figura 2: Schema di crittografazione tramite wep

La decodifica del messaggio avviene semplicemente con un processo di crittografazione inverso:

$$P' = C \oplus RC4(v,k) = (P \oplus RC4(v,k)) \oplus RC4(v,k) = P$$

Il ricevente scompone ***P'*** in ***[M',c']*** e calcola il checksum su ***M'***: ***c(M')*** e verifica che sia uguale a ***c'***.

2 Tool di auditing

2.1 Apparecchiature

Le prove di laboratorio sono state effettuate con le seguenti apparecchiature:

- access point:
  - produttore: Compaq
  - modello: WL410
  - standard di rete: IEEE 802.11
  - crittografia: wep 128bit
  - canali: 11
  - frequenze: da 2,4 a 2,4897 GHz
  - altre caratteristiche: compatibile IEEE 802.1D (Spanning Tree), possibilità di filtri mediante Access control Table e autenticazione mediante Radius
  -

velocità	11 Mbps	5.5 Mbps	2 Mbps	1 Mbps
copertura in spazio aperto	160 m	270 m	400 m	550 m
copertura in spazio semi aperto	50 m	70 m	90 m	115 m
copertura al chiuso	25 m	35 m	40 m	50 m
sensibilità di ricezione	-82dBm	-87dBm	-91dBm	-94dBm
intervallo di ritardo tipico	65ns	225ns	400ns	500ns

- pc1:
  - processore: Pentium 3
  - ram: 128 MB
  - scheda di rete:
    - produttore: Cisco
    - modello: Aironet 350
    - firmware: 4.25.05
    - velocità dati supportate: 1, 2, 5,5 e 11 Mbps
    - standard di rete: IEEE 802.11b
    - banda di frequenza: da 2,4 a 2,4897 GHz
    - tipi di architettura di rete: infrastruttura e ad hoc
    - supporto radio: DSSS (Direct Sequence Spread Spectrum)
    - Media Access Protocol: CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance)
    - modulazione: DBPSK a 1 Mbps, DQPSK a 2 Mbps, CCK a 5,5 e a 11 Mbps
    - canali operativi: Nord America: 11, ETSI: 13, Giappone: 14

    - canali non sovrapposti: Tre
    -

velocità	1 Mbps	2 Mbps	5.5 Mbps	11 Mbps
sensibilità di ricezione	-94 dBm	-91 dBm	-89 dBm	-85 dBm
intervallo di ritardo (tipico)	500 ns	400 ns	300 ns	140 ns

    - impostazioni disponibili per la potenza di trasmissione: 100 mW (20 dBm), 50 mW (17 dBm), 30 mW (15 dBm), 20 mW (13 dBm), 5 mW (7 dBm), 1 mW (0 dBm)
    - portata (tipica):

	11 Mbps	1Mbps
interni	40 m	107 m
esterni	244 m	610 m

    - antenna: esterna rimovibile 2,2 dBi dipolo con connettore RP-TNC
    - tipo di autenticazione: LEAP
  - OS: Debian 3.0, kernel 2.4.20

- pc2:
  - processore: Pentium 3
  - ram: 128 MB
  - scheda di rete:
    - produttore: Compaq
    - modello: WL110
    - firmware: 0x8002A
    - velocità dati supportate: 1, 2, 5,5 e 11 Mbps
    - standard di rete: IEEE 802.11b
    - banda di frequenza: 2400-2483.5 MHz
    - supporto radio: DSSS (Direct Sequence Spread Spectrum)
    - Media Access Protocol: CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance)
    - modulazione: DBPSK, DQPSK, CCK
    - canali operativi: 11
    -

velocità	11 Mbps	5.5 Mbps	2 Mbps	1 Mbps
copertura in spazio aperto	160 m	270 m	400 m	550 m
copertura in spazio semi aperto	50 m	70 m	90 m	115 m
copertura al chiuso	25 m	35 m	40 m	50 m
sensibilità di ricezione	-82dBm	-87dBm	-91dBm	-94dBm
intervallo di ritardo tipico	65ns	225ns	400ns	500ns

    - potenza in uscita: 15dBm
    - tipo di autenticazione: wep
  - OS: Red Hat 7.3

- palmare:
  - produttore: Compaq
  - modello: H3600
  - processore: Strong Arm
  - ram: 16 Mb
  - scheda di rete:
    - produttore: Compaq
    - modello: WL110
  - OS: Gpe2 v0.7 pre7 for h3600, kernel 2.4.19

Nella prima fase i tool di auditing sono stati installati sul pc1 e il pc2 o il palmare (in mutua esclusione visto che entrambi usavano la stessa scheda wireless) sono stati usati per generare traffico.

Nella seconda fase sono stati sperimentati i tool di auditing sul palmare e quindi il pc1 è servito per la generazione del traffico.

In entrambi i casi si è usata una rete wireless di tipo ad infrastruttura. Per utilizzare la scheda Cisco in modalità promiscua e quindi "vedere" tutti i pacchetti, anche quelli non destinati alla macchina, è necessario eseguire due comandi manualmente:

```
echo 'Mode: r' > /proc/driver/aironet/eth1/Config
echo 'Mode: y' > /proc/driver/aironet/eth1/Config
```

supponendo che eth1 sia la scheda di rete wireless.

2.2 Tool per Linux

Sulla macchina "pc1", su cui vi è installata la distribuzione Linux Debian 3.0 (Stable), sono stati testati i seguenti tool: Kismet, Airtraf e Airsnort.

2.2.1 Kismet

Scheda del prodotto:

- sviluppatore: Mike Kershaw <dragon@kismetwireless.net>
- sito internet: http://www.kismetwireless.net
- licenza: GPL
- versione: 2.8.1
- sistemi operativi supportati: Linux, Linux-Arm, BSD, Win32 (Cygwin), MacOS X

- schede supportate: tutte quelle che possono lavorare in rfmon mode (accesso raw ai pacchetti), quindi tutte quelle basate su prism2 (Linksys, D-Ling, Rangelan, ecc), Cisco Aironet e Orinoco
- supporto per il gps
- wep crack

Configurazione

Seguendo la consuetudine dei sistemi Unix, la configurazione del programma si trova nella directory di sistema /etc/kismet. Il file principale relativo all configurazione è kismet.conf. Tra i parametri possiamo evidenziare:

- "source=tipo,interfaccia,nome" che definisce la scheda di rete da utilizzare per l'acquisizione. In particolare tipo definisce il tipo della scheda (da scegliere tra cisco, prism2, orinoco, wsp100, general), interfaccia definisce l'interfaccia software a cui è collegate la scheda (verificabile tramite il comando ifconfig) e nome è il nome con cui è identificata la scheda ed il suo valore è arbitrario.
- "logtype=formato" abilita la possibilità di salvare su file i dati acquisiti. I possibili formati (da indicare separati da virgola) sono: dump, network, csv, xml, weak, cisco e gps
- "gps=true/false" abilita o disabilita la funzione gps
- "ap_manuf=ap_manuf client_manuf=client_manuf" questi due parametri indicano dove è possibile reperire i fingerprint delle schede di rete e degli access point conosciuti.

Sempre in /etc/kismet si trova il file kismet_ui.conf, ap_manuf e client_manuf. Il primo definisce i parametri dell'interfaccia grafica, mentre gli altri due sono stati descritti qui sopra.

Interfaccia

Kismet si presenta con un'interfaccia testuale suddivisa in tre zone (figura 3): in quella centrale c'è la lista delle reti disponibili con le loro caratteristiche, nella colonna di destra c'è un riepilogo di tutti i pacchetti che sono stati analizzati e in basso lo status.
Più in dettaglio si può analizzare il significato dei parametri per ogni entry del riquadro principale:

- "Name": è l'essid della rete
- "T": è il tipo di rete:
  - "P": probe request: una scheda client non ancora on-line cerca una una rete

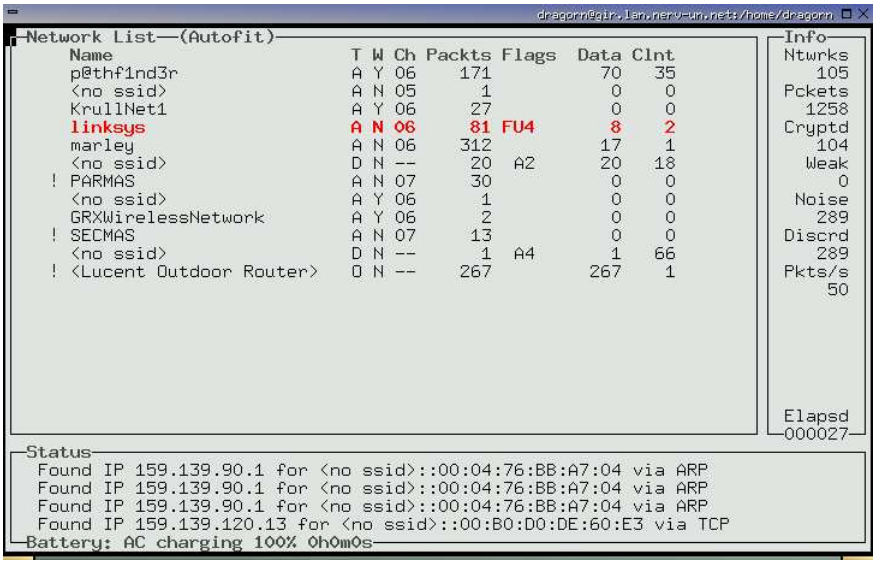


Figura 3: Schermata principale di Kismet

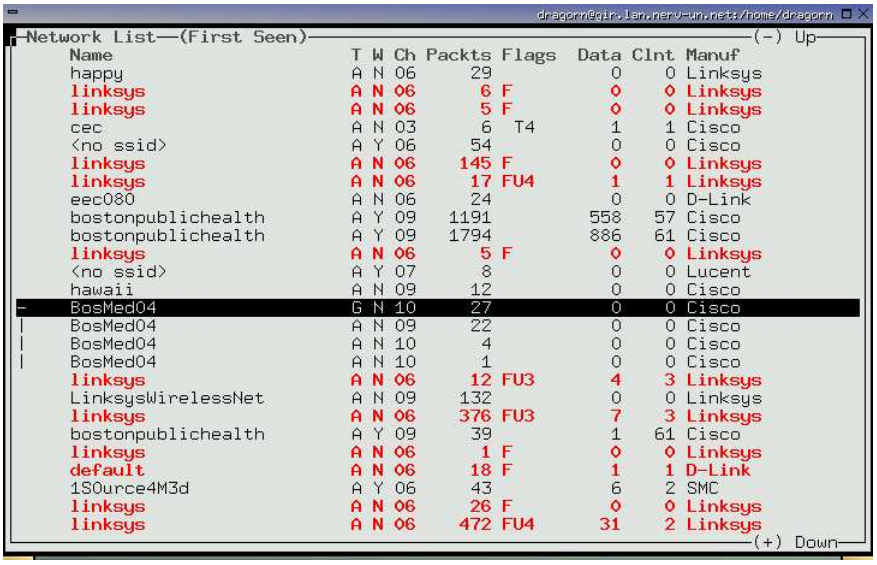


Figura 4: Schermata principale di Kismet (massimizzata)

- “A”: access point: rete ad infrastruttura
- “H”: ad-hoc: rete poit-to-point
- “T”: turbocell (aka KarInet e Lucent Outdoot Router)
- “G”: gruppo di reti (creato dall’utente con il comando “g”)

  - "D": rete di soli dati senza pacchetti di controllo

- "W": indica se il wep è attivo o no (nel caso si utilizzi un protocollo più avanzato, come l'eap, questo flag è settato "N", non crittografato)
- "Ch": indica il canale di lavoro della rete
- "Packts": indica il numero di pacchetti analizzati
- "Data": quantità di dati trasferiti
- "Clnt": indica il numero di client collegati
- "Manuf": produttore dell'access point (figura 4)

Per quanto riguarda la colonna laterale i parametri rappresentano:

- "Ntwrks": il numero di reti scoperte
- "Pckets": numero di pacchetti analizzati
- "Cryptd": numero di pacchetti crittografati
- "Weak": numero di pacchetti deboli
- "Noise": livello di rumore
- "Pkts/s": numero di pacchetti al secondo

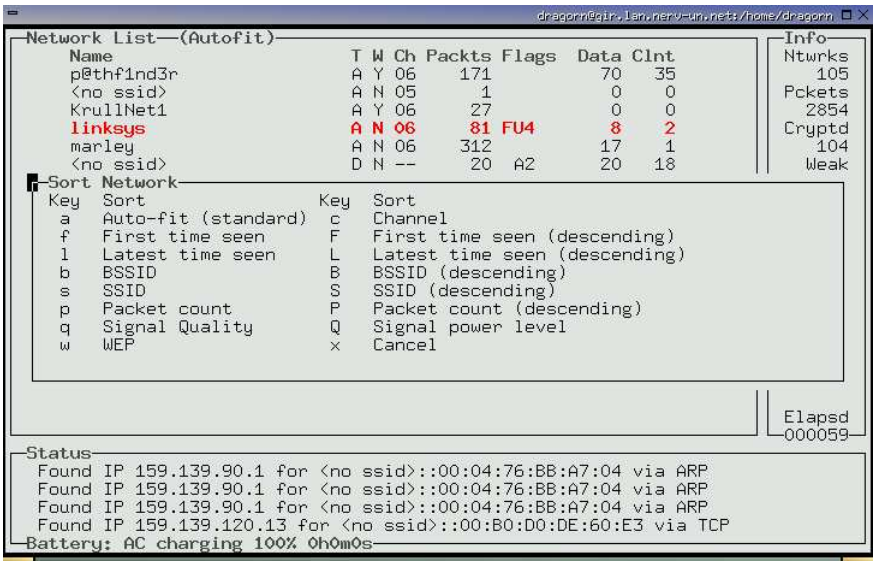


Figura 5: Possibili ordinamenti delle reti "scoperte"

I comandi della schermata principale sono i seguenti:

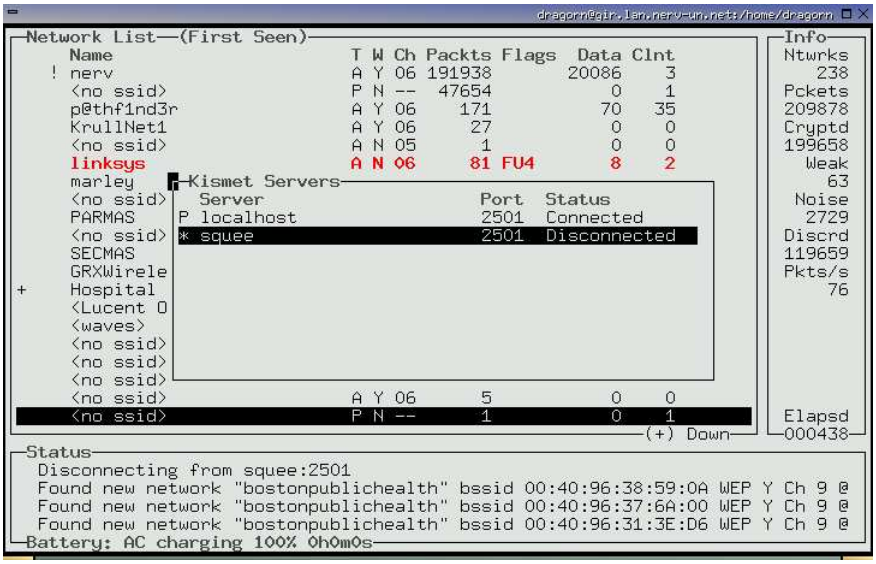


Figura 6: Elenco dei server

- “s”: permette di cambiare l’ordine di visualizzazione delle reti scoperte (figura 5)
- “e”: apre la finestra dei server di Kismet (in questo modo è possibile monitorare più server in contemporanea, figura 6)
- “r”: frequenza di arrivo dei pacchetti (figura 7)
- “a”: statistiche sui pacchetti acquisiti (figura 8) e dei canali
- “i”: mostra i dettagli sulla rete selezionata (figura 9)
- “p”: visualizza le informazioni sui pacchetti ricevuti (figura 10)
- “g/u”: crea / elimina un gruppo di reti (figura 11)
- “c”: mostra la lista dei client wireless individuati (figura 12)
  - “s”: seleziona l’ordinamento dei client
  - “i”: mostra le informazioni sui singoli client (figura 13)
- “t”: seleziona o deseleziona una rete o un gruppo di reti
- “z”: passa dalla visualizzazione principale a uno zoom solo sul riquadro principale (da figura 3 a figura 4)
- “n”: rinomina la rete o il gruppo di reti selezionato

- “l”: visualizza le informazioni sui livelli di segnale, rumore e potenza delle schede
- “d”: visualizza le stringhe “stampabili” contenute nei pacchetti
- “f”: stima il centro fisico della rete e mostra la bussola (necessita gps)
- “w”: mostra tutti i precedenti alert e warning

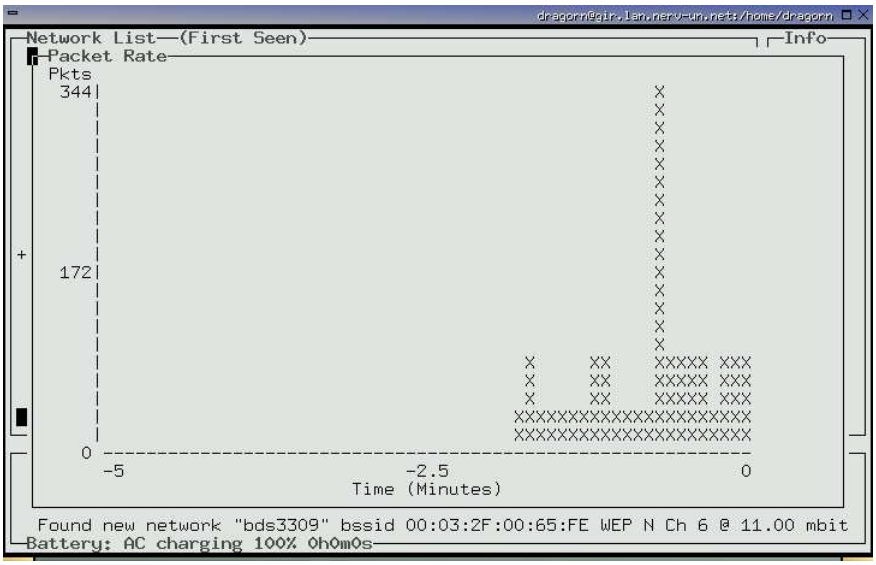


Figura 7: Frequenza di arrivo

Nella schermata di dettaglio delle reti, se la rete è un gruppo, si hanno i seguenti parametri:

- “Name”: nome del gruppo
- “Networks”: numero di reti nel gruppo
- “Min Loc”: minima area geografica coperta
- “Max Loc”: massima area geografica coperta
- “Range”: range del gruppo

Nel caso, invece, che la rete selezionata sia una rete fisica si ha (figura 9):

- “SSID”: ssid della rete
- “server”: quale server kismet riporta la presenza della rete
- “BSSID”

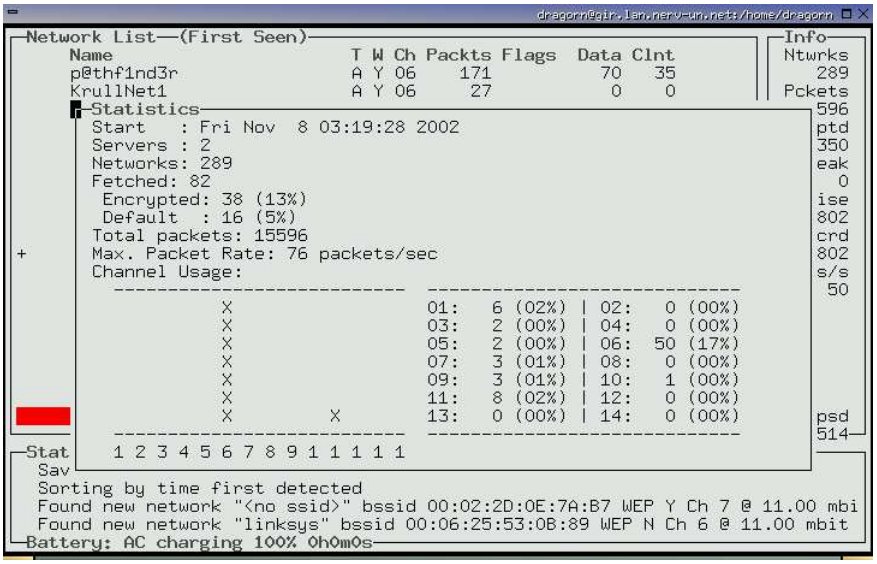


Figura 8: Statistiche sui pacchetti

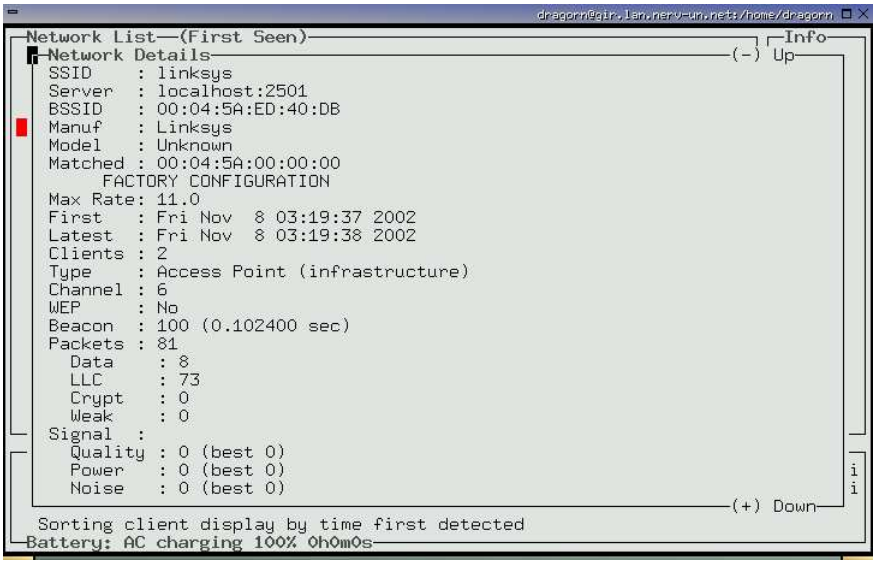


Figura 9: Informazioni sull'access point

- "Manuf": produttore della scheda basata su bssid mac
- "Model": modello, nel caso in cui ci sia un riscontro
- "Matched": parte del bssid mac usato per verificare il produttore e il modello

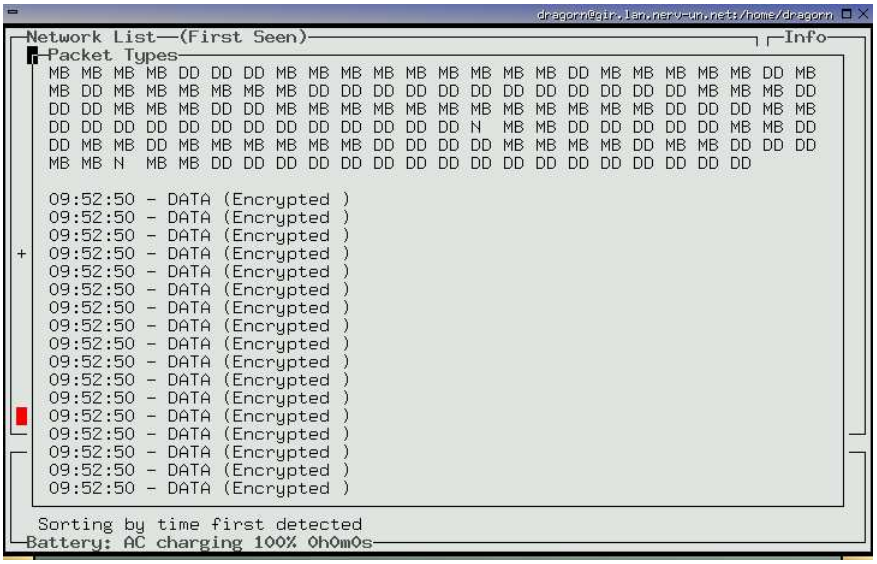


Figura 10: Informazioni sui singoli pacchetti

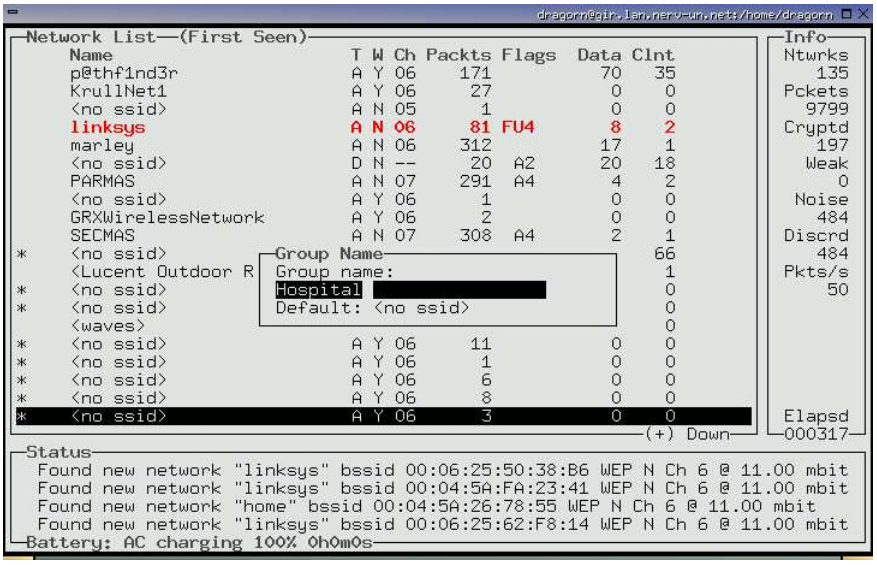


Figura 11: Selezione di un gruppo

- “Max rate”: massima velocità supportata dalla rete
- “First”: il tipo del primo pacchetto “visto” sulla rete
- “Lastest”: il tipo del’ultimo pacchetto “visto” sulla rete

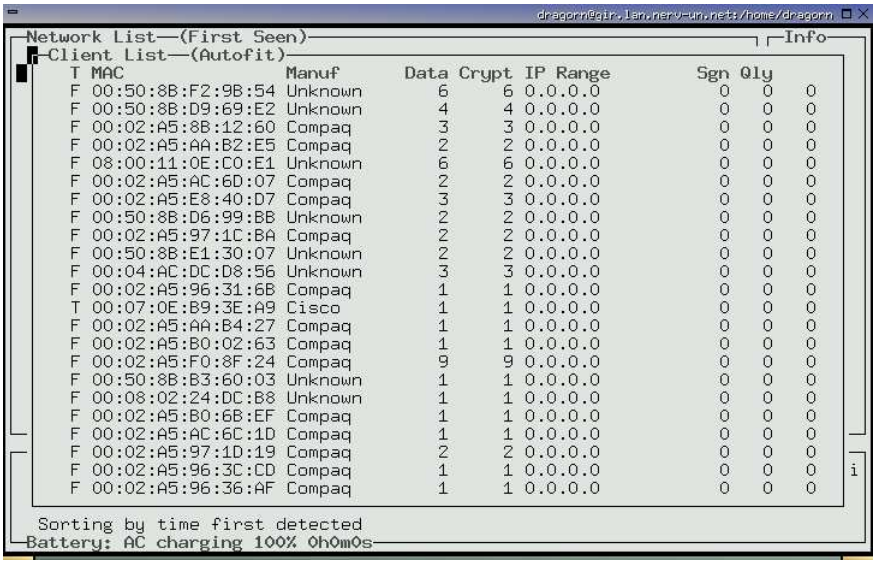


Figura 12: Client collegati ad un AP

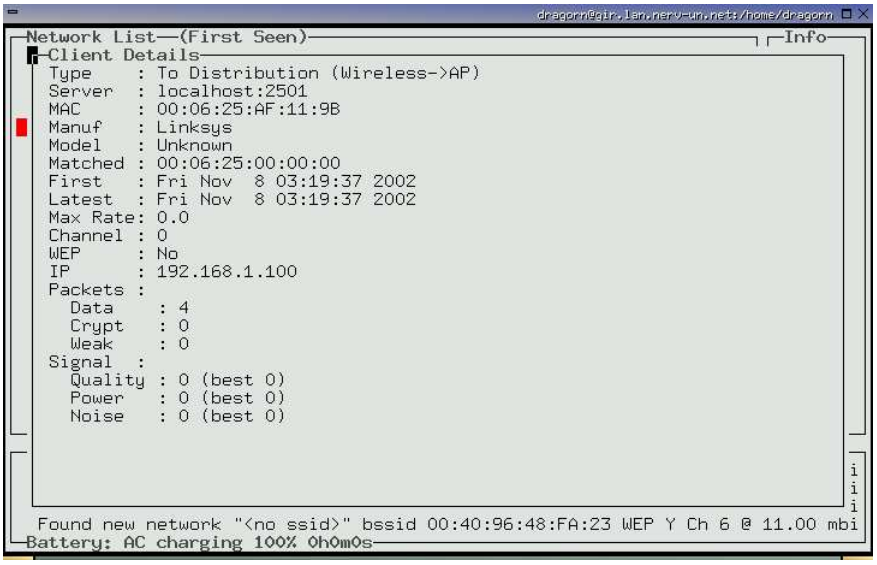


Figura 13: Dettagli su un client

- “Clients”: numero di client collegati alla rete
- “Type”: tipo di rete
- “Channel”: canale su cui la rete lavora

- “Wep”: indica se il wep è abilitato o no
- “Beacon”: frequenza del beacon
- “Packets”: numero e tipo di pacchetti acquisiti
- “Data”: quantità di dati trasferita sulla rete
- “Signal”: livelli di segnale attuale e migliore sulla rete
- “IP”: gruppo di IP riscontrato sulla rete
- “Min Loc”: minima area geografica coperta
- “Max Loc”: massima area geografica coperta
- “Range”: range della rete

Visualizzando la lista dei client on-line, si hanno i seguenti parametri (figura 12):

- “T”: indica il tipo di client:
  - “F” - From DS: client broadcast dalla rete wireless del sistema distribuito
  - “T” - To DS: client trasmette sulla rete al sistema distribuito
  - “I” - Intra DS: il client è un nodo del sistema distribuito che trasmette ad un altro nodo del sistema
  - “E” - Established: il client è entrato nel sistema ed è subito uscito
- “MAC”: mac della scheda di rete
- “Manuf”: produttore
- “Data”: numero di pacchetti inviati
- “Crypt”: numero di pacchetti crittografati

Gli altri tre parametri riguardano statistiche della rete non sempre disponibili e significano:

- “IP RANGE“: ip con i quali dialoga
- “Sgn”: potenza del segnale
- “Qly”: qualità del segnale

Entrando nella schermata di dettaglio dei client si ha (figura 13):

- “Type”: tipo di connessione
- “Server”: server che vede il client

- “MAC”: mac della scheda
- “Manuf”: produttore basato sul mac
- “Matched”: parte del mac usato per verificare il produttore e il modello
- “Max rate”: massima velocità supportata dal client
- “First”: il tipo del primo pacchetto “visto” sul client
- “Lastest”: il tipo del’ultimo pacchetto “visto” sul client
- “Channel”: canale su cui il client lavora
- “Wep”: indica se il wep è abilitato o no
- “IP”: gruppo di IP riscontrato sul client
- “Min Loc”: minima area geografica coperta
- “Max Loc”: massima area geografica coperta
- “Range”: range del client
- “Packets”: numero e tipo di pacchetti acquisiti
- “Data”: quantità di dati trasferita sulla rete
- “Signal”: livelli di segnale attuale e migliore

2.2.2 Airtraf

Scheda del prodotto

- sviluppatore: Peter K. Lee <saint@elixar.com>
- sito internet: www.elixar.com
- licenza: GPL
- versione: 1.0
- sistemi operativi supportati: Linux (necessita della libreria ncurses e una risoluzione di 120 colonne per 45 righe)
- schede supportate: quelle basate su prism2, Cisco Aironet (kernel 2.4.7 e superiori) e Orinoco

Configurazione

A differenza di Kismet, Airtraf non utilizza un file di configurazione. Di default esegue una scansione delle schede disponibili per accedere alla rete wireless. Nel caso in esame, per fargli riconoscere la scheda si è dovuto forzargli a mano i parametri che indicano l’interfaccia e la il modello:

```
airtraf -Iwifi0 -Caironet
```

Interfaccia[1]

Airtraf presenta un'interfaccia testuale a menù (figura 14).

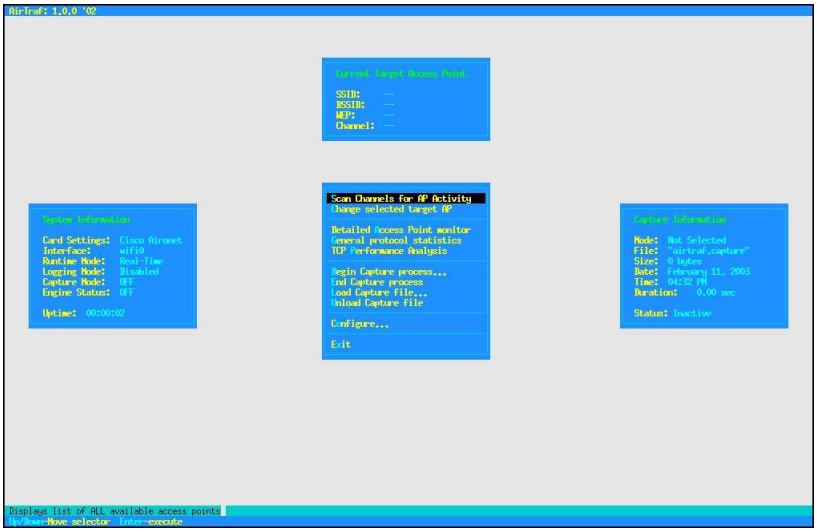


Figura 14: Schermata principale di Airtraf

Il primo comando da impartire è la scansione dei canali in cerca di access point attivi. Airtraf inizierà una scansione preliminare (figura 15) e in seguito mostrerà le reti trovate, eventualmente aggiornando la lista in caso di modifche (figura 16).

Con il comando "x" è possibile tornare alla schermata principale, dove viene chiesto di selezionare una rete di riferimento per le azioni successive (figura 17) Dal menù principale è ora possibile visualizzare informazioni dettagliate sulla rete selezionata (figura 18), sui pacchetti in generale (figura 19) o in particolare sul protocollo tcp (figura 20).

É inoltre possibile salvare i pacchetti acquisiti su file, selezionando le impostazioni di cattura (figura 21)

2.2.3 Airsnort

Scheda del prodotto

- sviluppatori: Jeremy Bruestle <melvin@melvin.net>, Blake Hegerle <blake @melvin.net>, Snax <snax@shmoo.com>
- sito internet: http://airsnort.shmoo.com

[1] L'interfaccia di Airtraf non è criptica come quella di kismet, quindi la sua descrizione sarà molto breve.

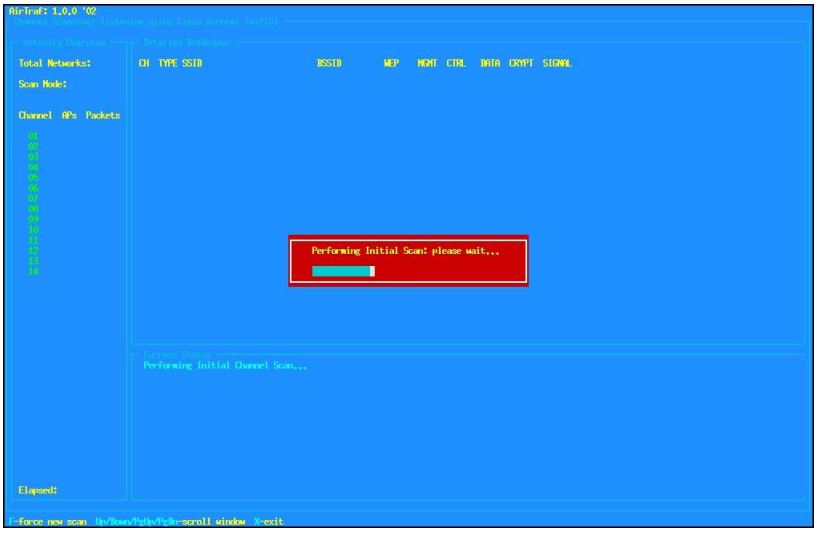


Figura 15: Scansione delle reti wireless

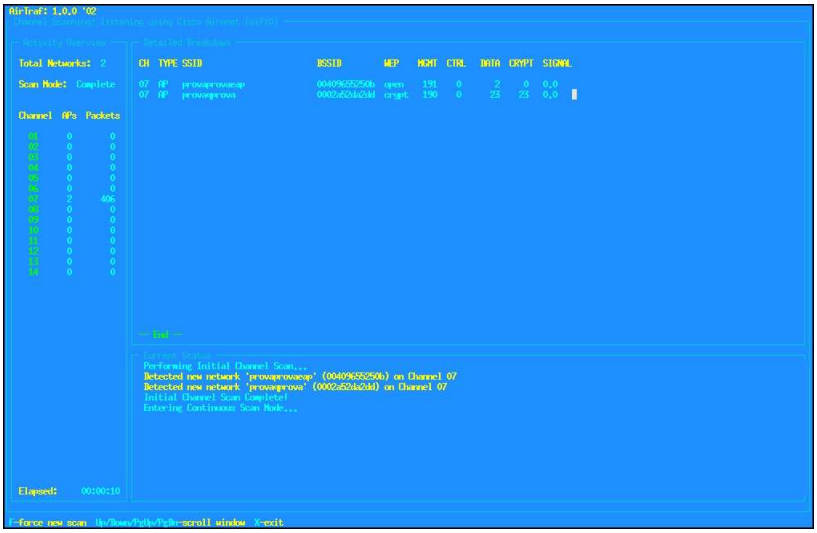


Figura 16: Visualizzazione delle reti disponibili

- licenza: GPL
- versione: 0.2.1b-2
- sistemi operativi supportati: Linux (necessita delle librerie gtk1.2)
- schede supportate: quelle basate su prism2, Cisco Aironet (ma non è in grado di portarle in monitor mode) e Orinoco

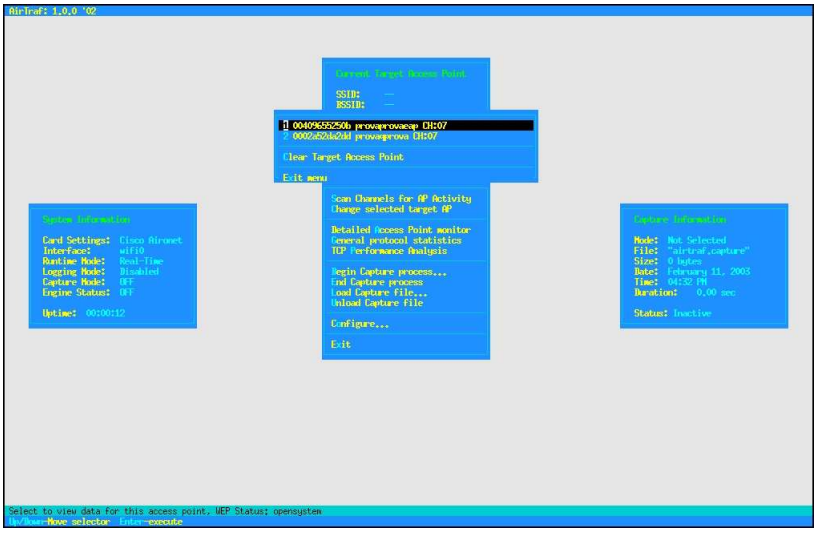


Figura 17: Selezione della rete

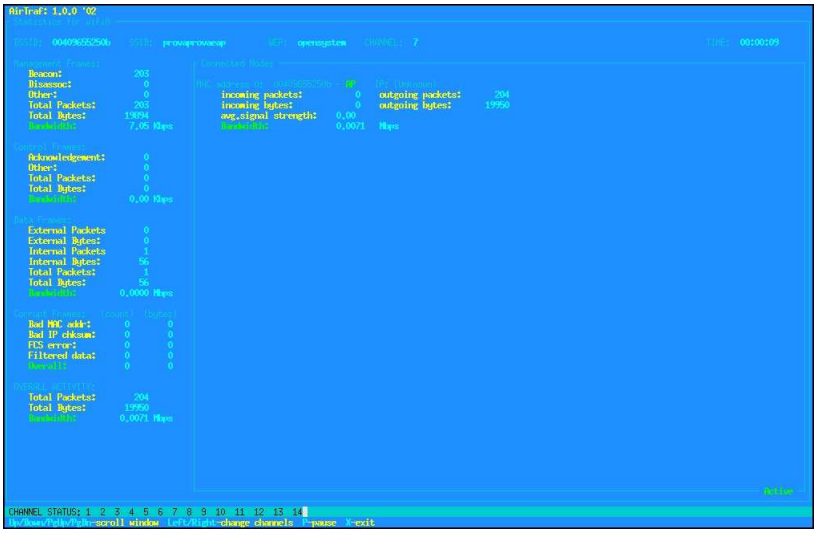


Figura 18: Dettagli sulla rete selezionata

Configurazione

Come Airtraf, Airsnort non necessita di file di configurazione e le impostazioni vengono modificate da interfaccia grafica. Quando il programma viene chiuso, salva le impostazioni correnti nel file .airsnortrc nella home directory dell'utente che lo ha utilizzato, riportando le modifiche rispetto alla configurazione di default.

Interfaccia

Airsnort presenta un'interfaccia grafica in gtk. Le opzioni modificabili sono poche:

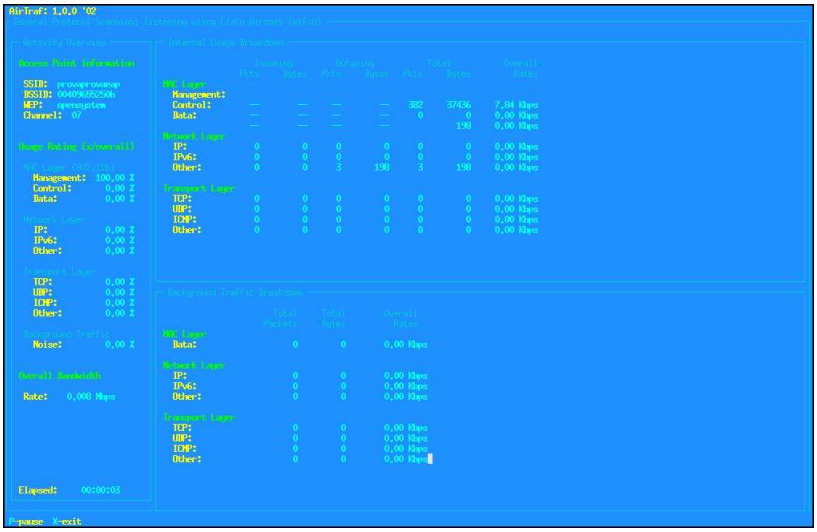


Figura 19: Informazioni sui pacchetti

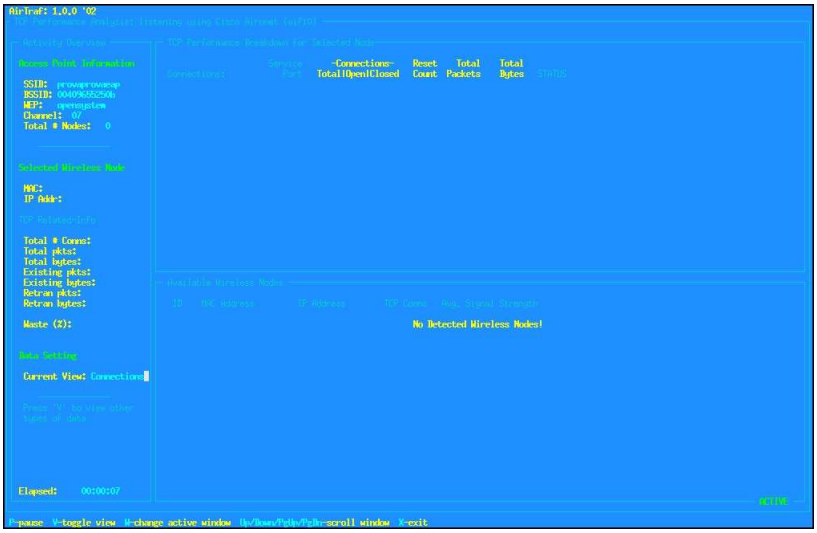


Figura 20: Statistiche sullo stack tcp

- Airsnort può scandire tutti i canali in modo automatico alla ricerca di una rete, oppure si può selezionare un canale su cui lavorare;
- "Network device": è l'interfaccia di rete che si desidera utilizzare;
- "Card type": è il modello di scheda. Si può scegliere tra prism2, Orinoco o general. In quest'ultimo caso la modalità scansione potrebbe non funzionare.

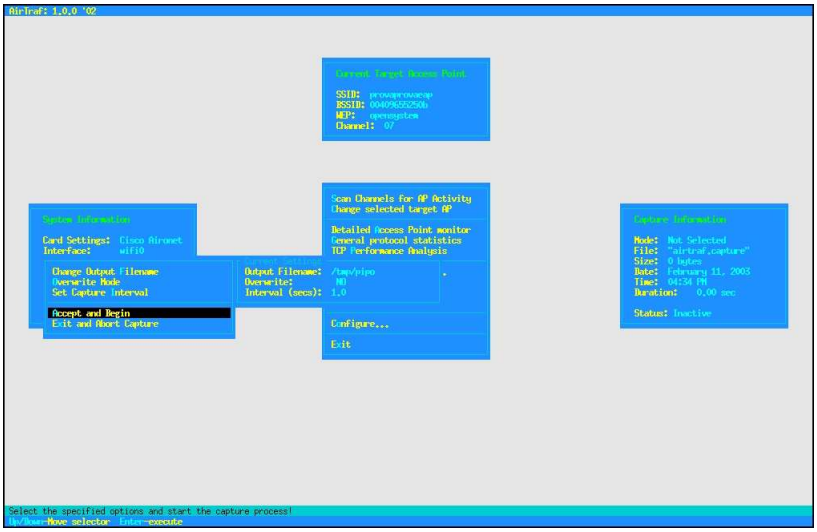


Figura 21: Impostazioni di acquisizione

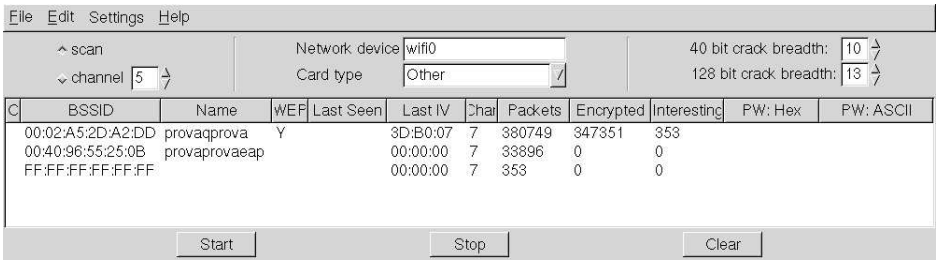


Figura 22: Interfaccia di Airsnort

2.3 Tool per iPaq

2.3.1 Kismet

É in tutto e per tutto identico alla versione per pc. Versione testata 2.4.6.

2.3.2 Wscan

Scheda del prodotto

- sviluppatori: Ginny Mak <makg@cs.pdx.edu>, Jim Binkley <jrb@cs.pdx.edu>
- sito internet: http://www.cs.pdx.edu/research/SMN/
- licenza: GPL like
- versione: 1.0

- sistemi operativi supportati: Linux, Freebsd (richiede fltk)
- schede supportate: quelle basate su prism2

Configurazione

Wscan non ha bisogno di particolari impostazioni. L'unica variabile è il tempo di aggiornamento

Interfaccia

Wscan presenta un'interfaccia grafica in fltk. Nella schermata principale viene

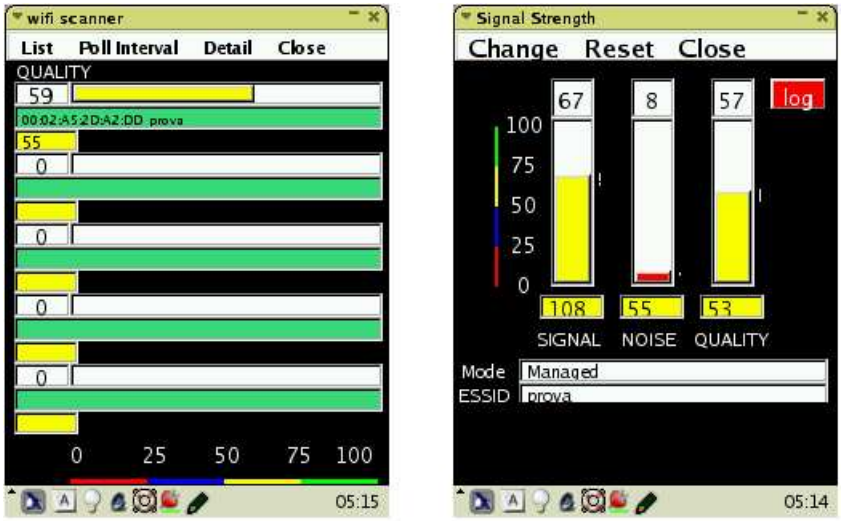


Figura 23: Interfaccia di Wscan

visualizzato un riassunto di tutte le reti "trovate" con l'indicazione del ssid e la qualità della rete. Nella schermata di dettaglio c'è un dettaglio della rete (tipo, essid, ssid), potenza del segnale, rumore e qualità.

2.3.3 PrismStumbler

Scheda del prodotto

- sviluppatore: Florian Boor <boor@unix-ag.org>
- sito internet: http://prismstumbler.sourceforge.net
- licenza: GPL
- versione: 0.6.0
- sistemi operativi supportati: Linux
- schede supportate: quelle basate su prism2

Configurazione

Non necessita di particolari configurazioni. Le uniche opzioni riguardano i canali da analizzare.

Interfaccia

PrismStumbler presenta un'interfaccia grafica in gtk. Presenta quattro tab:

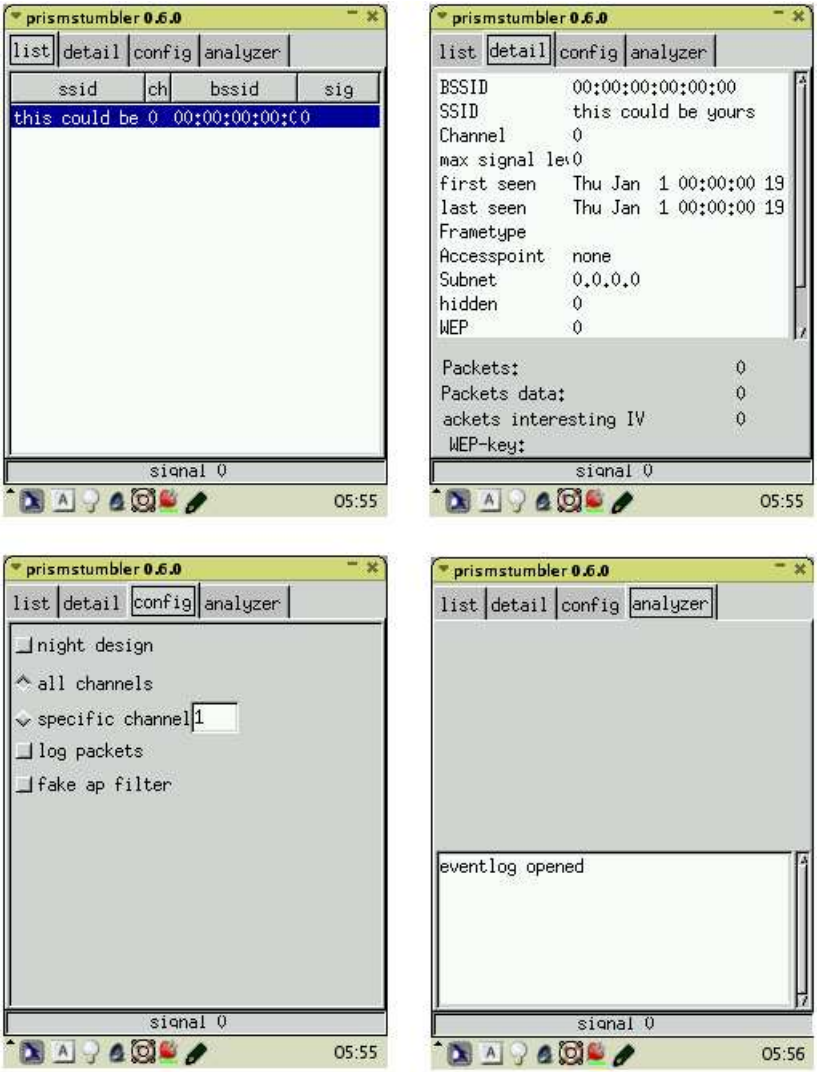


Figura 24: Interfaccia di PrismStumbler

- list: elenca tutte le reti con ssid, bssid, canale e qualità del segnale;
- detail: in cui vengono evidenziati i dettagli per una singola rete;
- config: il dettaglio della configurazione;

- analyzer: in cui c'è il log degli eventi.

2.4 Utilizzo e comparazione dei software

Airsnort e Airtraf sono due programmi molto intuitivi da utilizzare.

Una volta lanciato Airsnort, basta selezionare il tipo di scheda, l'interfaccia ed eventualmente il canale. Dopo di che basta cliccare su "Start" e il software inizierà l'acquisizione. Il contro di essere molto facile da utilizzare è nel fatto che le informazioni ottenute da questo programma non sono molto dettagliate. Infatti per ogni rete scoperta è in grado di visualizzare il bssid, nome della rete, l'informazione sul wep, l'IV, il canale, numero dei pacchetti, numero di pacchetti crittografati, il numero di pacchetti interessanti e, se è riuscito a rompere la chiave, la chiave in esadecimale e in ascii. Per quanto riguarda l'input di dati precedentemente acquisiti, Airsnort è in grado in importare il formato libpcap (tcpdump) e il suo formato proprietario, mentre in output può salvare solo nel suo formato interno.

L'uso di Airtraf è un pochino più articolato rispetto ad Airsnort, ma il vantaggio è che è in grado di fornire delle informazioni più dettagliate per ogni rete scoperta e sui protocolli che transitano sulle reti wireless. Già nella prima schermata (figura 16) di dettaglio delle reti, si hanno le stesse informazioni disponibili con Airsnort (a parte il crack della chiave wep) oltre alla qualità del segnale. Inoltre i dati sono ordinati per canale, informazione utile per elaborazioni successive. Passando al dettaglio per ogni rete (figura 18) si hanno informazioni su tutto il traffico che passa per l'access point, come il numero di pacchetti e la quantità di byte in input e in output e la banda disponibile. Inoltre si hanno dati dettagliati sui frame di management, di controllo, di dati e sui frame corrotti. Per ognuna di queste categorie si hanno nuovamente informazioni come il numero di pacchetti, la quantità di byte e la banda occupata.
Passando al dettaglio dei protocolli (figura 19) si hanno informazioni dettagliate sul numero di pacchetti e la quantità in byte in ingresso, in uscita e il totale. I dati sono divisi per:

- layer fisico: management, controllo, dati;
- layer di rete: IP, IP6, altri;
- layer di trasporto: TCP, UDP, ICMP, altri;

Per le stesse categorie, oltre alle informazioni dettagliate, si hanno i riepiloghi in percentuale. Nella schermata di riepilogo del tcp (figura 20) si hanno informazioni di riepilogo sul protocollo e dettagli sui collegamenti tra l'access point e i nodi della rete.
Per quanto riguarda il trattamento dei dati, è in grado di leggere e scrivere il formato libpcap.

I due programmi testati solo sull'iPaq, Wscan e PrismStumbler, sono entrambi molto intuitivi da utilizzare. Entrambi presentano una schermata principale di

riassunto delle reti riscontrate e una schermata successiva che aggiunge alcuni particolari sulle singole reti.

Kismet è il programma più completo. Può mostrare molte informazioni dettagliate (come descritto nel paragrafo 2.2.1) sulla rete, sull'access point, sulle schede di rete. I dati più importanti che visualizza (specialmente perchè è l'unico che lo può fare) sono relativi al produttore ed al modello delle schede e degli access point (figure 13,9) che intercetta. Inoltre è in grado di visualizzare statistiche di arrivo di pacchetti (figure 7, 8) e sulla potenza di segnale delle schede e dell'access point.
Per quanto riguarda il trattamento dei dati, può salvare i dati acquisiti un diversi modi (vedi Appendice A):

- tcpdump completo;
- weak, formato tcpdump ma solo dei pacchetti deboli;
- network, dove riporta in un formato leggibile le informazioni sugli access point;
- csv, da le stesse informazioni del formato network ma in una struttura comma separated;
- xml, il formato "proprietario", è la struttura più completa in cui oltre a descrivere la rete, come nel formato network, aggiunge i dettagli per ogni client di ciascuna rete.

2.5 Tabelle riassuntiva

	Kismet	Airtraf	Airsnort	Wscan	PrismStrubler
Sistemi operativi supportati					
Linux i386	SI	SI	SI	SI	SI
Linux Arm	SI	NO	NO	SI	SI
BSD	SI	NO	NO	SI	SI
Win32 (Cygwin)	SI	NO	NO	NO	NO
MacOS X	SI	NO	NO	NO	NO
Schede di rete supportate					
prism2	SI	SI	SI	SI	SI
Cisco Aironet	SI	SI	SI	NO	NO
Orinoco	SI	SI	SI	NO	NO
Trattamento dei dati					
input libpcap (tcpdump)	NO	SI	SI	NO	NO
output libpcap (tcpdump)	SI	SI	NO	NO	NO
input formato proprietario	NO	NO	SI	NO	NO
output formato proprietario	SI	NO	SI	NO	NO
Altre caratteristiche					

	Kismet	Airtraf	Airsnort	Wscan	PrismStrubler
Licenza	GPL	GPL	GPL	GPL like	GPL
Wep Crack	SI	NO	SI	NO	NO
Supporto GPS	SI	NO	SI	NO	NO
Documentazione	buona	buona	scarsa	scarsa	scarsa

Tabella 2: Comparazione software per pc

3 Possibili attacchi

3.1 WEP: introduzione

Il più grande problema del WEP è il riuso del cipher. Come detto nel paragrafo 1.2, il messaggio che viene mandato in rete è $C=P \oplus RC4(v,k)$. Per le proprietà dell'operatore xor, dati due messaggi crittografati con lo stesso vettore di inizializzazione ***iv*** e la stessa chiave ***k*** si ha:

$$C_1 = P_1 \oplus RC4(v,k)$$
$$C_2 = P_2 \oplus RC4(v,k)$$
$$C_1 \oplus C_2 = (P_1 \oplus RC4(v,k)) \oplus (P_2 \oplus RC4(v,k)) = P_1 \oplus P_2$$

In altre parole un xor tra due testi cifrati restituisce lo xor dei testi in chiaro [1, B].

Per portar a termine questo tipo di attacco sono necessarie due condizioni:

1. la disponibilità di testi cifrati con un keystream utilizzato molte volte
2. la conoscenza, anche parziale, dei dati originali.

Il keystream è generato a partire dalla chiave ***k*** e dal vettore di inizializzazione. Lo standard 802.11 prevede l'utilizzo di un vettore di quattro chiavi da utilizzare a seconda della postazione, in accordo con la configurazione impostata dal sistemista, ma praticamente nessuna implementazione utilizza questo metodo. Quindi la chiave è nota a tutti i membri della rete, cioè comune a tutti i pacchetti. Dunque l'unica cosa che resta da variare è il vettore di inizializzazione, che, tra l'altro, è trasmesso in chiaro nel pacchetto, in modo che il destinatario possa decodificarlo. La generazione dell'iv è una questione molto spinosa: infatti lo standard non definisce come debba essere gestito, nè impone l'unicità del vettore. A causa di questo molti produttori pongono a zero il vettore ogni volta che la scheda di rete viene reinizializzata (questo, per una scheda pcmcia, significa toglierla e rimetterla) poi utilizzano dei semplici incrementi di uno per i valori successivi. Come risultato si ha che i pacchetti con un basso valore del vettore di inizializzazione sono utilizzati molti volte.

Se al posto di un incremento costante si utilizzasse un valore casuale, la situazione non migliorerebbe sensibilmente. Infatti lo standard prevede che il vettore di inizializzazione sia di 24 bit. Questo significa che utilizzando un valore casuale di iv per ogni pacchetto trasmesso si rischia il riutilizzo della chiave dopo solo 5000 pacchetti, che equivale a pochi minuti di trasmissione.

Tipo di scheda	Chiave	Num. pacchetti	Pacchetti deboli	Rottura chiave
Aironet 4800	40 bit	1355 milioni	96	No
Aironet 4800	104 bit	120 milioni	96	No
3Com 3CR-WE62092A	40 bit	18.5 milioni	Meno di 100	No

Tipo di scheda	Chiave	Num. pacchetti	Pacchetti deboli	Rottura chiave
3Com 3CR-WE62092A	104 bit	10 milioni	Meno di 100	No
Orinoco Silver	40 bit	2.7 milioni	Molti	Si
Compaq WL110	40 bit	24 milioni	Molti	Si
Compaq WL110	104 bit	13 milioni	Molti	Si
Nokia D211	40 bit	22 milioni	Molti	Si
Nokia D211	104 bit	28 milioni	Molti	Si

Tabella 3: Attacchi a forza bruta[2, B]

Una volta riscontrati numerosi pacchetti con lo stesso ***iv***, si possono utilizzare vari metodi per ottenere il testo in chiaro:

- per esempio si può partire ad analizzare i campi dei pacchetti, infatti nel protocollo IP molti campi sono predicibili (ad esempio la sequenza di login in un sistema Unix/Linux);
- un altro metodo sicuro per ottenere un testo determinato è spedire una mail di spam ad uno o più dei client wireless della rete;
- se l'access point accetta connessioni non crittografate, è possibile mandare dei pacchetti in broadcast a tutte le postazioni ed analizzare la loro risposta.

Nel caso "sfortunato" in cui non si riesca ad ottenere un testo predeterminato, è possibile tentare un attacco di tipo dizionario. É un attacco che richiede un lavoro di preparazione notevole e che, nel caso in cui la chiave venga cambiata, si deve ricominciare da capo. In pratica il cracker costruisce una tabella contenete tutti i keystream validi per ogni valore del vettore di inizializzazione. Come detto questo è un lavoro molto lungo, che genera un gran mole di dati: considerando una chiave da 40 byte e 1500 byte per ognuna delle 2^{24} possibilità si raggiunge qualcosa come 24 GB di dati.

3.2 Autenticazione del messaggio

Il protocollo WEP prevede un campo di checksum per verificare l'integrità del messaggio. Per questo viene utilizzato un checksum di tipo CRC-32. In realtà questo sistema non è sufficiente per lo scopo che si vorrebbe ottenere: infatti il CRC-32 è stato progettato per rilevare errori accidentali nella trasmissione dei dati e non per garantire l'impossibilità di modifiche effettuate da terze parti.

3.2.1 Modifica del messaggio

Il checksum utilizzato dal WEP è di tipo lineare, quindi vale la proprietà distributiva rispetto allo xor: dati x e y si ha: $c(x \oplus y) = c(x) \oplus c(y)$.

Questo significa che è possibile eseguire delle modifiche controllate al testo cifrato senza invalidare il checksum. Si supponga di intercettare un testo C inviato da A a B: $A \rightarrow (B):(v,C)$. Come già detto si ha $C=[M,c(M)] \oplus RC4(v,k)$. Ora si suppone che esista C' da decrittare in M' con $M' = M + \Delta$. Infine si trasmette in nuovo messaggio a B. Non resta che trovare C':

$$\begin{aligned} C' &= C \oplus [\Delta, c(\Delta)] \\ &= RC4(v,k) \oplus [M,c(M)] \oplus [\Delta,c(\Delta)] \\ &= RC4(v,k) \oplus [M \oplus \Delta, c(M) \oplus c(\Delta)] \\ &= RC4(v,k) \oplus [M', c(M \oplus \Delta)] \\ &= RC4(v,k) \oplus [M', c(M')] \end{aligned}$$

Questo significa che è possibile modificare a piacimento i dati inviati, fallendo nello scopo di garantirne l'autenticità. Inoltre questo tipo di attacco permette di disturbare notevolmente le trasmissioni senza conoscere il messaggio originale M.

3.2.2 Generazione di messaggi "autentici"

Il checksum del WEP è una funzione del messaggio indipendente dalla chiave. Questo significa che può essere calcolato da un "avversario" che conosce il testo originale.

A causa di questo è possibile aggirare le misure per il controllo di accesso. Se un attaccante riuscisse ad ottenere il testo in chiaro di un frame sarebbe in grado trasmettere traffico arbitrario sulla rete. Inoltre, come già detto, conoscere il testo in chiaro e quello cifrato, significa conoscere il keystream che può essere usato per creare nuovi pacchetti con lo stesso ***iv***. Dato un testo cifrato C e il corrispondente testo in chiaro P, si può ricavare il keystream in questo modo:

$$P \oplus C = P \oplus (P \oplus RC4(v,k)) = RC4(v,k)$$

É quindi possibile spedire un nuovo messaggio crittografato M' da A a B $A \rightarrow B :[v,C']$ dove $C' = [M',c(M')] \oplus RC4(v,k)$. Questo nuovo messaggio utilizza lo stesso ***iv*** del messaggio precedente, ma questo non viola lo standard, quindi non viene segnalato alla vittima dell'attacco.

3.2.3 Spoofing dell'autenticazione

Questo è un caso particolare del precedente. Per autenticare una stazione mobile si utilizza un meccanismo a sfida. L'access point invia alla stazione un messaggio in chiaro, la stazione lo crittografa con la chiave comune e lo rimanda all'access point. Generare pacchetti crittografati correttamente è considerato una prova sufficiente del possesso della chiave.

Come detto sopra, un attaccante che conosce sia il testo in chiaro che quello crittografato è in grado ricavare il keystream e quindi autenticarsi sull'access point.

3.2.4 Decodifica del messaggio

Esistono due tecniche che possono essere utilizzate per decodificare i messaggi: IP redirection e Reaction attacks.

La tecnica dell'IP redirection viene utilizzata quando l'access point viene anche utilizzato come router per la connessione ad internet. Si utilizza la tecnica del paragrafo 3.2.2 per modificare l'indirizzo IP di destinazione di un pacchetto autentico, sostituendolo con l'IP di una macchina di cui l'attaccante ha il controllo. Quindi l'access point decodifica il pacchetto e lo manda al computer dell'attaccante, che, in questo modo, possiede sia il testo in chiaro che quello crittografato. Supponendo che $\boldsymbol{D}_H$ e $\boldsymbol{D}_L$ siano la parte alta e bassa di un una word a 16 bit contenete l'indirizzo IP di destinazione originario, $\boldsymbol{D}_H$**'** e $\boldsymbol{D}_L$**'** di quello nuovo, $\boldsymbol{K}$ sia il checksum dell'indirizzo vecchio e $\boldsymbol{K}$**'** di quello nuovo. Allora: $\boldsymbol{K' = K + D_H + D_L - D_H' - D_L'}$.
Si possono verificare tre casi:

- $\boldsymbol{K}$ noto: si può calcolare $\boldsymbol{K'}$ semplicemente come detto sopra e correggendo il pacchetto con uno xor tra $\boldsymbol{K}$ e $\boldsymbol{K'}$, che cambierà il checksum dell'IP con il giusto valore di $\boldsymbol{K'}$

- $\boldsymbol{K}$ non è noto: dato $\boldsymbol{G = K' - K}$ si deve calcolare $\Delta = \boldsymbol{K'} \oplus \boldsymbol{K}$. Dato $\boldsymbol{G}$ ci sono 2^{16} possibili combinazioni di Δ.

- $\boldsymbol{K = K'}$: in pratica si compensa il cambio di destinazione cambiando un altro campo del pacchetto, per esempio l'IP del pacchetto sorgente. Questo modifica potrebbe non essere valida, in quanto ci potrebbero essere delle regole nel firewall che scartano i pacchetti con indirizzi di partenza non noti. Una soluzione migliore consiste nel modificare l'indirizzo nel modo seguente: $\boldsymbol{D_L' = D_H + D_L - D_H'}$. Per esempio se l'indirizzo di partenza fosse 10.20.30.40 e l'attaccante può controllare una macchina sulla sottorete 192.168.0.0/16 sceglierà 192.168.103.147, con il risultato che il checksum del pacchetto non cambia.

La tecnica del Reaction attacks viene utilizzata quando non è possibile utilizzare l'IP redirection, ma funziona solo col il traffico TCP. L'attacco avviene in questo modo: l'attaccante intercetta un pacchetto crittografato: $\boldsymbol{A \rightarrow B:[v,C]}$. Dopo di che crea un nuovo pacchetto $\boldsymbol{C'}$ scambiando alcuni bit del pacchetto originale (secondo una tecnica che verrà spiegata più avanti), corregge il checksum e lo invia sulla rete. Se la modifica ha prodotto un pacchetto valido, il destinatario risponde con un pacchetto di ACK (facilmente riconoscibile anche senza decodificarlo a causa della lunghezza), altrimenti il pacchetto viene scartato. Se il pacchetto viene accettato si è scoperto un bit del messaggio originale. Ripetendo n volte l'attacco è possibile ottenere buona parte del testo originale.
Creazione del pacchetto: $\boldsymbol{C' = C \oplus D}$, dove $\boldsymbol{D}$ specifica la posizione dei bit da scambiare. Per scegliere $\boldsymbol{D}$ si sceglie un i arbitrario, si settano i bit con indice i e i+16 di $\boldsymbol{D}$ a 1 e tutti gli altri a 0. Per la proprietà dell'addizione in modulo $2^{16} - 1$

si ha $\boldsymbol{P \oplus D = P \bmod 2^{16} - 1}$ quando $P_i \oplus P_{i+16} = 1$. Assumendo che il checksum del pacchetto originale sia corretto allora lo è anche quello del pacchetto nuovo. In questo modo si è ottenuta l'informazione di un bit.

3.3 Possibili rimedi

Analizzando i dati di tabella 3 si vede che la difficoltà di un attacco dipende in buona parte dall'hardware utilizzato. Non avendo la certezza della qualità con cui un produttore ha implementato il WEP non rimane che cercare altre soluzioni.
Una di queste consiste nel posizionare l'access point della rete wireless al di fuori del firewall e considerare tutti i membri della rete come "ostili". In questo caso per garantire un accesso alla rete sicura si dovrà utilizzare una VPN.
Inoltre sarebbe auspicabile una miglior gestione delle chiavi: utilizzare il vettore di chiavi, anzichè solo la prima e sostituire le chiavi di frequente.
Da un punto di vista puramente hardware, è opportuno utilizzare degli access point il più direzionale possibile e con un raggio di azione ridotto, in modo da evitare fisicamente la possibilità di un'intrusione.

La soluzione migliore, sarebbe non utilizzare il WEP, ma passare a sistemi autenticazione più avanzati, come l'EAP, anche se, purtroppo, sono ancora pochi i produtttori di schede wireless che lo supportano.

4 Appendice A: formati di cattura di Kismet

In questa sezione verranno riportati alcuni esempi di cattura nei vari formati di Kismet.

4.1 Cattura in formato network

```
Network 1: "provaqprova" BSSID: "00:02:A5:2D:A2:DD"
    Type     : infrastructure
    Carrier  : 802.11b
    Info     : "None"
    Channel  : 07
    WEP      : "Yes"
    Maxrate  : 0.0
    LLC      : 38220
    Data     : 368341
    Crypt    : 368336
    Weak     : 60
    Total    : 406561
    First    : "Tue Feb 11 15:42:47 2003"
    Last     : "Tue Feb 11 16:48:00 2003"

Network 2: "provaprovaeap" BSSID: "00:40:96:55:25:0B"
    Type     : infrastructure
    Carrier  : 802.11b
    Info     : "sl180"
    Channel  : 07
    WEP      : "No"
    Maxrate  : 11.0
    LLC      : 37896
    Data     : 724
    Crypt    : 0
    Weak     : 0
    Total    : 38620
    First    : "Tue Feb 11 15:42:47 2003"
    Last     : "Tue Feb 11 16:47:59 2003"
    Address found via TCP 130.192.1.0
```

4.2 Cattura in formato csv

```
Network;NetType;ESSID;BSSID;Info;Channel;Maxrate;WEP;LLC;Data;\
Crypt;Weak;Total;First;Last;BestQuality;BestSignal;\
BestNoise;GPSMinLat;GPSMinLon;GPSMinAlt;GPSMinSpd;\
GPSMaxLat;GPSMaxLon;GPSMaxAlt;GPSMaxSpd;DHCP;ARP;UDP;TCP;
```

```
1;infrastructure;provaqprova;00:02:A5:2D:A2:DD;None;07;0.0;Yes;\
38220;368341;368336;60;406561;Tue Feb 11 15:42:47 2003;\
Tue Feb 11 16:48:00 2003;44;235;0;;;;;;;;;;;;;;;
2;infrastructure;provaprovaeap;00:40:96:55:25:0B;sl180;07;11.0;\
No;37896;724;0;0;38620;Tue Feb 11 15:42:47 2003;\
Tue Feb 11 16:47:59 2003;44;235;0;;;;;;;;;;;;;;;130.192.1.0;
```

4.3 Cattura in formato xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE detection-run SYSTEM
"http://kismetwireless.net/kismet-1.6.2.dtd">
<detection-run kismet-version="2.8.1" start-time="Tue Feb 11 15:42:47 2003"
end-time="Tue Feb 11 16:48:00 2003">
  <wireless-network number="1" type="infrastructure" wep="true"
  cloaked="false" carrier="802.11b"
  first-time="Tue Feb 11 15:42:47 2003"
  last-time="Tue Feb 11 16:48:00 2003">
    <SSID>provaqprova</SSID>
    <BSSID>00:02:A5:2D:A2:DD</BSSID>
    <channel>7</channel>
    <maxrate>0.0</maxrate>
    <packets>
      <LLC>38220</LLC>
      <data>368341</data>
      <crypt>368336</crypt>
      <weak>60</weak>
      <total>406561</total>
    </packets>
    <datasize>377041113</datasize>
    <wireless-client number="1" type="fromds" wep="false"
    first-time="Tue Feb 11 15:42:47 2003"
    last-time="Tue Feb 11 16:48:00 2003">
      <client-mac>00:A0:C9:44:B9:1E</client-mac>
      <client-packets>
        <client-data>246360</client-data>
        <client-crypt>246357</client-crypt>
        <client-weak>60</client-weak>
      </client-packets>
      <client-datasize>368794782</client-datasize>
      <client-maxrate>0.0</client-maxrate>
    </wireless-client>
    <wireless-client number="2" type="established" wep="false"
    first-time="Tue Feb 11 15:42:47 2003"
```

```
last-time="Tue Feb 11 16:48:00 2003">
  <client-mac>00:02:A5:6F:A0:F2</client-mac>
  <client-packets>
    <client-data>120551</client-data>
    <client-crypt>120549</client-crypt>
    <client-weak>0</client-weak>
  </client-packets>
  <client-datasize>7543091</client-datasize>
  <client-maxrate>0.0</client-maxrate>
</wireless-client>
<wireless-client number="3" type="fromds" wep="false"
first-time="Tue Feb 11 15:42:51 2003"
last-time="Tue Feb 11 16:47:52 2003">
  <client-mac>00:40:96:55:25:0B</client-mac>
  <client-packets>
    <client-data>988</client-data>
    <client-crypt>988</client-crypt>
    <client-weak>0</client-weak>
  </client-packets>
  <client-datasize>545456</client-datasize>
  <client-maxrate>0.0</client-maxrate>
</wireless-client>
<wireless-client number="4" type="fromds" wep="false"
first-time="Tue Feb 11 15:43:35 2003"
last-time="Tue Feb 11 16:46:35 2003">
  <client-mac>00:02:2D:1A:91:74</client-mac>
  <client-packets>
    <client-data>22</client-data>
    <client-crypt>22</client-crypt>
    <client-weak>0</client-weak>
  </client-packets>
  <client-datasize>1188</client-datasize>
  <client-maxrate>0.0</client-maxrate>
</wireless-client>
<wireless-client number="5" type="fromds" wep="false"
first-time="Tue Feb 11 15:44:14 2003"
last-time="Tue Feb 11 16:44:55 2003">
  <client-mac>00:01:02:1D:55:B1</client-mac>
  <client-packets>
    <client-data>244</client-data>
    <client-crypt>244</client-crypt>
    <client-weak>0</client-weak>
  </client-packets>
  <client-datasize>136073</client-datasize>
```

```
  <client-maxrate>0.0</client-maxrate>
</wireless-client>
<wireless-client number="6" type="fromds" wep="false"
first-time="Tue Feb 11 15:50:37 2003"
last-time="Tue Feb 11 16:45:57 2003">
  <client-mac>00:08:21:31:86:D5</client-mac>
  <client-packets>
    <client-data>10</client-data>
    <client-crypt>10</client-crypt>
    <client-weak>0</client-weak>
  </client-packets>
  <client-datasize>432</client-datasize>
  <client-maxrate>0.0</client-maxrate>
</wireless-client>
<wireless-client number="7" type="fromds" wep="false"
first-time="Tue Feb 11 15:52:26 2003"
last-time="Tue Feb 11 16:43:35 2003">
  <client-mac>00:01:02:9E:80:25</client-mac>
  <client-packets>
    <client-data>161</client-data>
    <client-crypt>161</client-crypt>
    <client-weak>0</client-weak>
  </client-packets>
  <client-datasize>18761</client-datasize>
  <client-maxrate>0.0</client-maxrate>
</wireless-client>
<wireless-client number="8" type="fromds" wep="false"
first-time="Tue Feb 11 16:43:47 2003"
last-time="Tue Feb 11 16:43:48 2003">
  <client-mac>00:01:02:2E:CE:76</client-mac>
  <client-packets>
    <client-data>2</client-data>
    <client-crypt>2</client-crypt>
    <client-weak>0</client-weak>
  </client-packets>
  <client-datasize>638</client-datasize>
  <client-maxrate>0.0</client-maxrate>
</wireless-client>
<wireless-client number="9" type="fromds" wep="false"
first-time="Tue Feb 11 16:44:07 2003"
last-time="Tue Feb 11 16:44:07 2003">
  <client-mac>00:01:02:9E:7D:F0</client-mac>
  <client-packets>
    <client-data>3</client-data>
```

```
      <client-crypt>3</client-crypt>
      <client-weak>0</client-weak>
    </client-packets>
    <client-datasize>692</client-datasize>
    <client-maxrate>0.0</client-maxrate>
  </wireless-client>
</wireless-network>
<wireless-network number="2" type="infrastructure" wep="false"
cloaked="false" carrier="802.11b"
first-time="Tue Feb 11 15:42:47 2003"
last-time="Tue Feb 11 16:47:59 2003">
  <SSID>provaprovaeap</SSID>
  <BSSID>00:40:96:55:25:0B</BSSID>
  <info>sl180</info>
  <channel>7</channel>
  <maxrate>11.0</maxrate>
  <packets>
    <LLC>37896</LLC>
    <data>724</data>
    <crypt>0</crypt>
    <weak>0</weak>
    <total>38620</total>
  </packets>
  <datasize>77145</datasize>
  <ip-address type="tcp">
    <ip-range>130.192.1.0</ip-range>
  </ip-address>
  <wireless-client number="1" type="fromds" wep="false"
  first-time="Tue Feb 11 15:42:51 2003"
  last-time="Tue Feb 11 16:47:52 2003">
    <client-mac>00:40:96:55:25:0B</client-mac>
    <client-packets>
      <client-data>408</client-data>
      <client-crypt>0</client-crypt>
      <client-weak>0</client-weak>
    </client-packets>
    <client-datasize>30129</client-datasize>
    <client-maxrate>0.0</client-maxrate>
  </wireless-client>
  <wireless-client number="2" type="fromds" wep="false"
  first-time="Tue Feb 11 15:43:35 2003"
  last-time="Tue Feb 11 16:46:35 2003">
    <client-mac>00:02:2D:1A:91:74</client-mac>
    <client-packets>
```

```
    <client-data>22</client-data>
    <client-crypt>0</client-crypt>
    <client-weak>0</client-weak>
  </client-packets>
  <client-datasize>1012</client-datasize>
  <client-maxrate>0.0</client-maxrate>
  <client-ip-address type="arp">130.192.1.185
  </client-ip-address>
</wireless-client>
<wireless-client number="3" type="fromds" wep="false"
first-time="Tue Feb 11 15:44:14 2003"
last-time="Tue Feb 11 16:44:35 2003">
  <client-mac>00:01:02:1D:55:B1</client-mac>
  <client-packets>
    <client-data>30</client-data>
    <client-crypt>0</client-crypt>
    <client-weak>0</client-weak>
  </client-packets>
  <client-datasize>7635</client-datasize>
  <client-maxrate>0.0</client-maxrate>
  <client-ip-address type="tcp">130.192.1.145
  </client-ip-address>
</wireless-client>
<wireless-client number="4" type="established" wep="false"
first-time="Tue Feb 11 15:44:28 2003"
last-time="Tue Feb 11 16:46:25 2003">
  <client-mac>00:08:21:31:86:D5</client-mac>
  <client-packets>
    <client-data>125</client-data>
    <client-crypt>0</client-crypt>
    <client-weak>0</client-weak>
  </client-packets>
  <client-datasize>23070</client-datasize>
  <client-maxrate>0.0</client-maxrate>
  <client-ip-address type="arp">130.192.1.138
  </client-ip-address>
</wireless-client>
<wireless-client number="5" type="fromds" wep="false"
first-time="Tue Feb 11 15:45:35 2003"
last-time="Tue Feb 11 16:38:08 2003">
  <client-mac>00:A0:C9:44:B9:1E</client-mac>
  <client-packets>
    <client-data>51</client-data>
    <client-crypt>0</client-crypt>
```

```
    <client-weak>0</client-weak>
  </client-packets>
  <client-datasize>2346</client-datasize>
  <client-maxrate>0.0</client-maxrate>
  <client-ip-address type="arp">130.192.1.190
  </client-ip-address>
</wireless-client>
<wireless-client number="6" type="fromds" wep="false"
first-time="Tue Feb 11 15:45:49 2003"
last-time="Tue Feb 11 16:47:28 2003">
  <client-mac>00:02:A5:6F:A0:F2</client-mac>
  <client-packets>
    <client-data>54</client-data>
    <client-crypt>0</client-crypt>
    <client-weak>0</client-weak>
  </client-packets>
  <client-datasize>6669</client-datasize>
  <client-maxrate>0.0</client-maxrate>
  <client-ip-address type="tcp">130.192.1.146
  </client-ip-address>
</wireless-client>
<wireless-client number="7" type="fromds" wep="false"
first-time="Tue Feb 11 15:52:26 2003"
last-time="Tue Feb 11 16:43:35 2003">
  <client-mac>00:01:02:9E:80:25</client-mac>
  <client-packets>
    <client-data>29</client-data>
    <client-crypt>0</client-crypt>
    <client-weak>0</client-weak>
  </client-packets>
  <client-datasize>4994</client-datasize>
  <client-maxrate>0.0</client-maxrate>
  <client-ip-address type="tcp">130.192.1.136
  </client-ip-address>
</wireless-client>
<wireless-client number="8" type="fromds" wep="false"
first-time="Tue Feb 11 16:43:48 2003"
last-time="Tue Feb 11 16:43:48 2003">
  <client-mac>00:01:02:2E:CE:76</client-mac>
  <client-packets>
    <client-data>2</client-data>
    <client-crypt>0</client-crypt>
    <client-weak>0</client-weak>
  </client-packets>
```

```
      <client-datasize>622</client-datasize>
      <client-maxrate>0.0</client-maxrate>
      <client-ip-address type="arp">130.192.1.141
      </client-ip-address>
    </wireless-client>
    <wireless-client number="9" type="fromds" wep="false"
    first-time="Tue Feb 11 16:44:07 2003"
    last-time="Tue Feb 11 16:44:07 2003">
      <client-mac>00:01:02:9E:7D:F0</client-mac>
      <client-packets>
        <client-data>3</client-data>
        <client-crypt>0</client-crypt>
        <client-weak>0</client-weak>
      </client-packets>
      <client-datasize>668</client-datasize>
      <client-maxrate>0.0</client-maxrate>
      <client-ip-address type="arp">130.192.1.134
      </client-ip-address>
    </wireless-client>
  </wireless-network>
</detection-run>
```

5 Appendice B: installazione Ipaq

L'installazione è articolata nelle seguenti fasi:[1, W]

1. installazione del bootloader;
2. installazione della root del filesystem;
3. configurazione del palmare.

Per iniziare l'installazione è necessario un software di sincronizzazione, come ActiveSync in ambiente WIndows. Il primo passo consiste nel trasferimento dei file "BootBlaster_1.18.exe" e "bootldr-2.18.01.bin". Successivamente sull'Ipaq si deve eseguire Bootblaster, dal menù "Flash" selezionare "Program" e quindi l'immagine del boot loader scaricata. Questa operazione richiedere circa 15 secondi. Successivamente si seleziona "Verify", sempre dal menù "Flash". Se la verifica è positiva si può riavviare il palmare. Il nuovo boot loader è installato. Durante la pressione del pulsante di reset, premere il joypad. In questo modo il palmare non avvierà nessun sistema operativo, ma rimarrà in attesa di comandi.

Per poter interagire con il palmare è necessario configurare un emulatore di terminale con i parametri **115200 8N1 no flow control**. Una volta collegato il palmare alla seriale e riavviato come descritto sopra, si otterrà un prompt dei comandi *boot>*.

Nel caso si debba reinstallare il boot loader, si devono usare il comando *load bootldr* e quindi dal terminale mandare il file "bootldr-2.18.01.bin" con il protocollo xmodem. Il palmare mostrerà qualcosa del tipo:

```
boot> load bootldr
loading flash region bootldr
using xmodem
ready for xmodem download..
BSD sum value is: 00000000
programming flash...
unlocking boot sector of flash
Protect=00000000
erasing ...
Erasing sector 00000000
writing flash..
addr: 00000000 data: EA00008E
addr: 00010000 data: E1A0C00D
verifying ... done.
startAddress :00000000
limitAddress :00018980
Protecting sector 00000000
Protect=00010001
```

Il passo successivo consiste nel configurare la partizione con il comando “partition reset”:

```
boot> partition reset
argv[1]=reset
defining partiton: bootldr
defining partiton: root
```

Ora si procede con l’installazione del sistema operativo. In questo caso è stata scelta l’immagine “bootgpe2-v0.7-pre7-h3600.jffs2”, basata su gtk2.2. Il comando da impartire è “load root” e quindi si deve mandare l’immagine al palmare, sempre con il protocollo xmodem.

```
boot> load root
loading flash region root
using xmodem
ready for xmodem download..
Erasing sector 00140000
Erasing sector 00180000
Erasing sector 001C0000
Erasing sector 00200000
  .
  .
  .
addr: 00360000 data: 781590DB
addr: 00370000 data: 642637AE
addr: 00380000 data: E0021985
addr: 00390000 data: 15DA97EC
Erasing sector 00FC0000
writing flash..
addr: 00100000 data: E0021985
addr: 00110000 data: E3BAD617
addr: 00120000 data: 0FA1F57B
addr: 00130000 data: 9343AEEB
  .
  .
  .
addr: 00600000 data: E0021985
addr: 00610000 data: FFFFFFFF
addr: 00620000 data: FFFFFFFF
addr: 00630000 data: FFFFFFFF
verifying ... formatting ... done.
```

Quest’ultima operazione è molto dispendiosa in termini di tempo, a causa della limitata velocità della seriale. Il tempo previsto è di circa 50 minuti.

Con il comando “boot” si riavvia il palmare. Al primo boot, deve inizializzare la chiave per il server ssh.

La versione di immagine installata è configurata per utilizzare il dhcp per le impostazioni della rete, quindi non è necessario configurare nulla, in quanto la rete del laboratorio dispone di un dhcp server.

6 Riferimenti

I riferimenti con B sono relativi alla bibliografia, mentre quelli con W corrispondono alla sezione web.

Riferimenti bibliografici

[1] L. Nielsen E. Dawson. Automated cryptanalysis of xor plaintext strings. Cryptologia, (2): 165-181, Apr. 1996.

[2] Claudio Ferrero. Test di resistenza ad airsnort. (CSP - uso interno).

Riferimenti web

[1] Jamey Hicks <jamey@handhelds.org> Alexander Guy <a7r@handhelds.org>. Familiar v0.5.2 installation instructions. http://familiar.handhelds.org/familiar/releases/latest/install/H3600/install.html.

[2] David Wagner <daw@cs.berkeley.edu> Nikita Borisov <nikitab@cs.berkeley.edu>, Ian Goldberg <ian@zeroknowledge.com>. Intercepting mobile communications: The insecutiry of 802.11. http://www.isaac.cs.berkeley.edu/isaac/web-faq.html.