Nomadic Comunication
AA 2008-2009

Report 2: Sicurezza in una rete 802.11 protetta con Wired Equivalent Privacy

Group N. 4

Marco Cattani, Matteo Chini

Sommario

Questo report vuole analizzare la sicurezza di una rete wireless che opera mediante protocollo WEP. In particolare si vuole valutare la robustezza al variare della lunghezza della chiave e delle tecniche di cracking. Verrà analizzata la quantità di traffico catturato necessaria per recuperare la chiave segreta e le percentuali di successo ottenute con diverse quantità di pacchetti. Nel report è inoltre presente una sezione riguardante iperf il cui scopo è quello di valutare la qualità dei dati prodotti, confrontandoli con quelli di Netperf: un altro tool per misurare il throughput ampliamente utilizzato.

Indice

1 Introduzione

A differenza delle reti cablate, il mezzo di trasmissione utilizzato dalle reti wireless è condiviso e accessibile da tutti. Questa caratteristica comporta svariati problemi. Tra questi, la sicurezza dei dati trasmessi è uno dei maggiori. Una rete wireless non protetta non può, infatti, garantire nessun requisito di privacy, sicurezza (il mezzo di trasmissione è accessibile da chiunque e chiunque può trasmettere) o di controllo degli accessi. Per questo motivo, sono stati sviluppati diversi protocolli che garantiscano questi aspetti. Con questa relazione vogliamo verificare la robustezza di uno di questi protocolli: WEP (Wired Equivalent Privacy) il cui nome promette di garantire un livello di privacy pari a quello delle reti cablate.

Negli esperimenti di laboratorio, un computer correttamente autenticato ed associato ad una rete protetta da WEP, genererà del traffico dati. Un altro computer *monitor* non associato alla rete registrerà il traffico e cercherà di recuperare la chiave sfruttando alcune falle presenti in WEP. Di seguito vengono descritti alcuni protocolli di sicurezza comunemente utilizzati.

1.1 Sistema aperto

Questa soluzione non utilizza nessun protocollo per la sicurezza. Il livello di sicurezza di questa soluzione è nullo in quanto permette di accedere alla rete a qualsiasi dispositivo. È molto semplice da implementare in quanto non richiede nessuno software o hardware. Il controllo degli accessi tramite sistema aperto è praticamente impossibile.

1.2 Sistema aperto + autenticazione MAC

Simile al sistema aperto, utilizza l'indirizzo MAC della NIC card come autenticazione. Dal punto di vista dell'amministrazione questo sistema è abbastanza complesso da gestire in quanto richiede il mantenimento di una lista di indirizzi MAC abilitati. L'indirizzo MAC inoltre è poco human-readable, la lista dei MAC può raggiungere dimensioni considerevoli e la gestione di dispositivi ospiti (abilitati temporaneamente) va fatta manualmente. Il livello di sicurezza di questa soluzione è debole in quanto le NIC card permettono di modificare il proprio indirizzo MAC. Il traffico inoltre viene trasmesso in chiaro (non criptato) ed è possibile intercettarlo senza essere autenticati alla rete.

1.3 Sistema aperto + gateway web

Simile ai due sistemi precedenti, utilizza un gateway di livello 3 per reindirizzare tutto il traffico ad un server web contenente una pagina di autenticazione. Una volta autenticato, il traffico dell'utente viene autorizzato. Questo sistema rispetto alla lista di MAC è più sicuro e permette una gestione semplice degli utenti e degli ospiti. Uno svantaggio di questo sistema è che richiede ai dispositivi collegati un browser web per accedere a qualsiasi servizio. Il livello di sicurezza è superiore alla soluzione basata su indirizzo MAC. Il traffico viene comunque trasmesso in chiaro è può essere intercettato.

1.4 WEP

Questo sistema cripta il traffico a livello 2 tra client e access point. Per accedere alla rete viene utilizzata una chiave condivisa. Questo sistema soffre, quindi, dei problemi tipici delle chiavi condivise come la distribuzione della chiave o la modifica di essa. Per criptare utilizza una chiave che può essere di 64 o 128 bit (in realtà 40 e 104 bit + una sequenza di 24 bit sempre diversa trasmessa in chiaro nel pacchetto). Il livello di sicurezza è superiore ai sistemi precedenti ed il traffico è trasmesso criptato. In pratica, come è stato dimostrato, WEP è molto fragile, non offre il livello di sicurezza promesso (Wired Equivalent Privacy) e non è pensato per fare il controllo degli accesso.

1.5 802.1x

Questa soluzione è uno standard IEEE che lavora a livello 2. L'autenticazione viene effettuata utilizzando il protocollo EAP (Extensible Authentication Protocol)[1]. Questo protocollo è molto flessibile e permette di utilizzare svariati meccanismi di autenticazione (EAP-MD5, EAP-TLS, EAP-TTLS, PEAP). É necessaria l'installazione di un software sul client (Supplicant) che gestisca questo protocollo. L'access-point (Authenticator) si appoggia su un server RADIUS (Remote Authentication Dial-In User Service) per autenticare il Supplicant. 802.1x definisce due tipi di connessione alla rete wireless. Una attraverso una porta *non-controllata*, che viene utilizzata dal Supplicant per richiedere l'autenticazione, l'altra, *controllata*, che viene utilizzata dal client per accedere alla rete. Inizialmente la porta controllata è chiusa e viene aperta al Supplicant solo dopo essere

[1] RFC 2284

stato autenticato dall'authentication server (RADIUS). Il traffico viene criptato utilizzando una chiave dinamica.

1.6 802.11i e WPA

IEEE 802.11i (conosciuto anche come WPA2) è uno standard sviluppato dalla IEEE per colmare le lacune di sicurezza di WEP. In attesa di 802.11i la Wi-Fi Alliance ha introdotto un protocollo transitorio per tamponare l'emergenza WEP: il Wi-Fi Protected Access (WPA). Il meccanismo di autenticazione si divide in due tipi: *consumer* ed *enterprise*. L'autenticazione *enterprise* utilizza dapprima un server RADIUS e produce una chiave privata PMK (Primary Master Key) utilizzata dal client per inizializzare l'algoritmo di criptazione. Le informazioni di autenticazione sono a loro volta criptate per proteggersi da possibili intercettazioni. La versione *consumer* utilizza invece una password segreta condivisa PSK (pre-shared key) per generare la chiave PMK.

2 Wired Equivalent Privacy

2.1 Definizione nello standard 802.11

All'interno dello standard 802.11 viene definito il protocollo Wired Equivalent Privacy (WEP): un protocollo di sicurezza a livello MAC che provvede a garantire la stessa sicurezza di una comunicazione cablata. Lo standard definisce l'utilizzo di una chiave K di 40 bit (WEP-40) o di 104 (WEP-104) condivisa tra tutte le stazioni (STA) e l'Access Point (AP) a cui si connettono. Sia nel caso di chiave a 40 che a 104 bit viene utilizzato lo stesso algoritmo, sistema di incapsulamento e decapsulamento. Utilizzando WEP il frame viene incrementato di 8 byte. La fig.1 mostra i campi e le modifiche apportate che vengono aggiunti al MPDU:

- IV: questo campo di 4 byte contiene parte della chiave che servirà per criptare il messaggio. È composto da 3 sottocampi: l'Inizialization Vector che è la chiave vera e propria ed è composta da 24 bit, il campo PAD utilizzato come riempimento composto da 6 bit a 0, il campo Key ID composto da 2 bit che identifica la chiave utilizzata per la comunicazione (l'AP permette di impostarne 4 ma solo una viene utilizzata per la comunicazione)
- Data: il campo Data contiene l'MPDU criptata dall'algoritmo

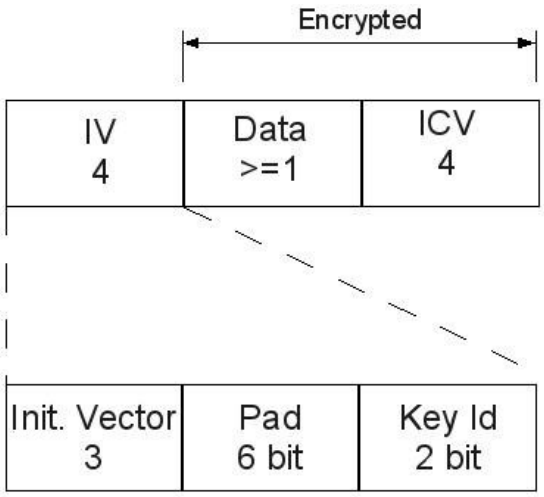

Figura 1: Campi della WEP MPDU

- ICV: Integrity Check Value è un CRC di 32 bit associato al campo data e viene protetto grazie alla crittografia. Viene calcolato sull'intero MPDU.

L'algoritmo definito dallo standard per cifrare i messaggi è l'Alleged RC4 (ARC4)[2] il quale prende in input l'IV concatenato con la chiave (IV || chiave = seed) e genera una sequenza di byte casuali (key stream) per criptare il messaggio. L'algoritmo di RC4 è suddiviso in due parti:

- KSA: Key Scheduling Algorithm è la prima parte dell'algoritmo e serve per inizializzare un array S di 256 elementi inizializzati da 0 a 255. Viene eseguita una permutazione di questi elementi in funzione della lunghezza e degli elementi che compongono il seed.

- PRGA: Pseudo Random Generation Algorithm è la seconda parte dell'algoritmo RC4 ed è un generatore di byte casuali il quale esegue un ciclo finchè non si ha raggiunto il numero di byte necessari per criptare il messaggio. Ad ogni ciclo restituisce un byte. Utilizza la permutazione precedentemente creata dal KSA per la generazione di numeri casuali.

Una volta ottenuta questa sequenza di byte casuali si passa alla criptazione dell'MPDU ottenuta mediante XOR del key stream con i byte che compongono il

[2]identico come comportamento all'algoritmo RC4, il quale, però, ha nome coperto da brevetto

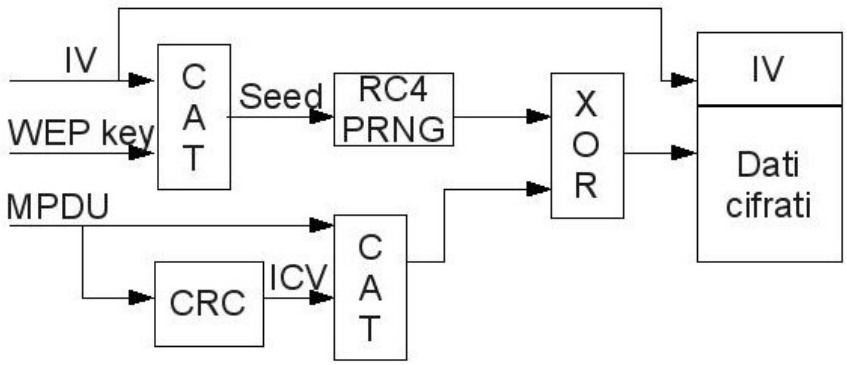

Figura 2: cifratura della MPDU tramite WEP

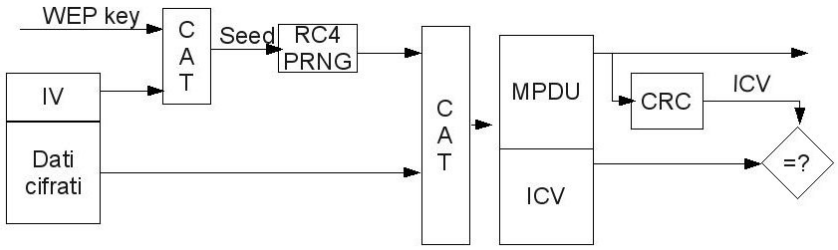

Figura 3: decifratura della MPDU tramite WEP

pacchetto concatenato con l'ICV. La fig. 2 vuole riassumere la sequenza finora descritta per ottenere il messaggio criptato, che verrà trasmesso concatenato al campo IV descritto in precedenza. La fase di decifratura è simile a quella di cifratura: poichè il pacchetto ricevuto è composto dall'IV concatenato con il messaggio criptato, la stazione ricevente potrà ricavare lo stesso seed utilizzato per la cifratura. In questo modo l'algoritmo RC4 genererà la stessa sequenza di byte prodotta in precedenza. Eseguendo nuovamente lo XOR tra il key stream e il messaggio si ottiene, quindi, il messaggio in chiaro composto da MPDU e ICV (che permette di verificare la correttezza del pacchetto). La fig. 3 riassume i passi per decifrare. Nel caso di utilizzo di WEP, lo standard definisce inoltre un sistema di autenticazione a chiave condivisa basato sull'invio di quattro frame tra STA e AP:

1. un primo frame di tipo Management (authentication) mandato dal richiedente all'AP nel quale si indica di voler utilizzare il sistema di autenticazione a

chiave condivisa.

2. un secondo frame di tipo Management (authentication) mandato dall'AP al richiedente e contenente un *challenge text* casuale che il richiedente dovrà provvedere a criptare con la propria chiave

3. un terzo frame di tipo Management (authentication) mandato dal richiedente all'AP contenente il *challenge text* incapsulato come WEP MPDU.

4. un quarto e ultimo frame di tipo Management (authentication) che confermerà o meno al richiedente l'autenticazione in base all'esito del confronto tra il frame ricevuto ed il proprio.

Oltre al sistema di autenticazione basato su chiave condivisa, le stazioni possono associarsi senza nessun tipo di autenticazione (Open System Authentication) e comunque utilizzare il protocollo WEP per criptare i pacchetti. In questo caso i frame scambiati sono solamente due:

1. un primo frame di tipo Management (authentication) mandato dal richiedente all'AP nel quale si indica di voler utilizzare l' open system authentication

2. un secondo frame di tipo Management (authentication) mandato dall'AP al richiedente nel quale si indica se il richiedente è stato autenticato oppure no.

In entrambi i casi, comunque, l'invio e la ricezione dei dati avviene in modo cifrato grazie all'utilizzo della WEP

2.2 Debolezze del WEP

Il riutilizzo del keystream è il problema maggiore in un sistema di crittografia come l'RC4. Quando due pacchetti sono criptati con la stessa chiave, eseguendo uno XOR su di essi si ottiene lo XOR dei pacchetti in chiaro. Se uno dei due pacchetti è conosciuto, è facile ottenere l'altro pacchetto.

- keystream $k = IV||chiave$
- Primo messaggio cifrato $C1 = M1 \oplus RC4(k)$
- Secondo messaggio cifrato $C2 = M2 \oplus RC4(k)$
- $C2 \oplus C1 = M1 \oplus RC4(k) \oplus M2 \oplus RC4(k) = M1 \oplus M2$

Molti dei pacchetti appartenenti ad una comunicazione sono di dimensione e contenuto conosciuti, quindi ottenendo uno di questi per un attaccante sarebbe possibile ottenere in chiaro il resto della comunicazione. Per risolvere questo problema, viene utilizzato nel protocollo WEP un IV casuale di 24 bit che permette di cifrare i messaggi con keystream diversi. Mantenendo la chiave sempre uguale, però, il numero di keystream ottenibili sono $2^{24} = 16777216$. Lo standard consiglia, ma non impone, di cambiare IV all'invio di ogni pacchetto. Inoltre molte case costruttrici di schede wireless generano l'IV impostandolo a 0 ogni volta che una comunicazione viene iniziata e incrementa il contatore di 1 ad ogni pacchetto. Questo tipo di generazione dell'IV fa riutilizzare spesso gli stessi keystream in una comunicazione.

La distribuzione delle chiavi in una rete protetta da WEP è un problema: lo standard non definisce un sistema per distribuirle in modo chiaro e comodo. L'inserimento viene fatto in modo manuale ed ogni volta che viene cambiata la chiave sull'AP, a tutte le stazioni bisognerà riassegnare la chiave.

Nell'agosto del 2001 S. Fluhrer, I. Mantin e A. Shamir pubblicano il paper con il titolo "Weakness in the Key Scheduling Algorithm of RC4" mostrando le debolezze delle chiavi generate dall'algoritmo RC4 e dimostrano che il tempo necessario ad un attacco basato sul loro metodo cresce in modo lineare alla lunghezza della chiave in bit. Il loro metodo si basa sul fatto che parte del seed utilizzato per generare la chiave (IV) viene mandato in chiaro all'interno del pacchetto. Questo permette di dedurre con un alta probabilità molti stati iniziali dell'algoritmo KSA. Loro identificano come "weak IV" gli IV del tipo (B+3:FF:x) dove B identifica il byte della chiave che si vuole trovare (B+3 identifica il primo byte della chiave segreta), FF indica che il secondo byte dell'IV deve essere fisso a FF mentre il terzo byte della chiave non ha prerequisiti ma è importante ai fini dell'algoritmo di cracking. Con questi valori possiamo identificare il fatto che per ogni byte che compone la chiave esistono 256 weak IV. In una rete protetta con WEP-40 in cui la chiave è composta da 5 byte segreti vi sono 5*1*256=1280 weak IV. Aumentando la lunghezza della chiave, quindi, aumentiamo il numero di weak IV che un attacker può sfruttare. La tabella 2.2 vuole descrivere il numero di weak IV in funzione della lunghezza della chiave.

Lunghezza della chiave	Numero di weak IV	percentuale dello spazio di IV
40	1280	0.008%
104	3328	0.020%

Questo sistema si basa sull'assunzione di conoscere il primo byte generato dall'algoritmo RC4. Nel caso di reti 802.11 a livello LLC viene utilizzata un'inte-

stazione di tipo SNAP il cui primo byte è conosciuto e sempre uguale a 0xAA. Questo fatto permette di applicare l'algoritmo di S. Fluhrer, I. Mantin e A. Shamir con successo.

3 Esperimenti di laboratorio

In seguito verranno spiegati nel dettaglio l'hardware utilizzato per effetture i test in laboratorio, il software coinvolto e la metodologia di test. Nell'ultima sezione vengono mostrati e valutati i risultati ottenuti nei test.

3.1 Hardware utilizzato

I test svolti hanno coinvolto l'utilizzo dell' Access Point Cisco Aironet 1230B, il quale fa parte degli AP di Cisco denomitati 1200 Series[3]. Tale AP supporta una comunicazione su standard 802.11b e molti parametri possono essere configurati da parte dell'utente. La configurazione iniziale su cui sono stati effettuati i test è la seguente:

- Radio Channel= Channel 13 (2472Mhz)
- Radio Preamble=Short
- Beacon Period=100 Kμs
- Fragmentation Threshold=2346
- RTS Threshold=2347

Sull'access point era attivo il protocollo WEP a 64 e 128 bit. La chiave WEP condivisa è stata generata in modo casuale così che non fosse possibile effettuare attacchi *brute-force* con *dizionario*[4].

3.2 Software utilizzato

Di seguito vengono riportate le caratteristiche dei software utilizzati per testare la robustezza di WEP.

[3] http://www.cisco.com/en/US/products/hw/wireless/ps430/

[4] In un attacco con dizionario vengono utilizzate come chiave tutte le parole appartenenti ad un certo dizionario (dizionario inglese, nomi propri, nomi di città, ecc..)

3.2.1 Airodump-ng

Airodump-ng è stato utilizzato per catturare i frame 802.11. A differenza di Wireshark, airodump permette di visualizzare, durante la registrazione, il numero di IV univoci (Initialization Vector) catturati.
Di seguito viene visualizzato un esempio di schermata di Airodump-ng:

```
CH  9 ][ Elapsed: 1 min ][ 2007-04-26 17:41 ][ WPA handshake: 00:14:6C:7E:40:80

BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB  ENC  CIPHER AUTH ESSID

00:09:5B:1C:AA:1D   11  16       10        0    0  11  54. OPN              NETGEAR
00:14:6C:7A:41:81   34 100       57       14    1   9  11  WEP  WEP         bigbear
00:14:6C:7E:40:80   32 100      752       73    2   9  54  WPA  TKIP   PSK  teddy

BSSID              STATION            PWR  Lost  Packets  Probes

00:14:6C:7A:41:81  00:0F:B5:32:31:31   51     2       14
(not associated)   00:14:A4:3F:8D:13   19     0        4  mossy
00:14:6C:7A:41:81  00:0C:41:52:D1:D1   -1     0        5
00:14:6C:7E:40:80  00:0F:B5:FD:FB:C2   35     0       99  teddy
```

Airodump-ng visualizza una lista degli access point visibili dalla scheda di rete assieme ad una lista dei client connessi (station). La prima linea mostra il canale wireless utilizzato, il tempo trascorso dall'avvio, la data e, opzionalmente, se è stato catturato un handshake WPA/WPA2. Di seguito viene chiarito il significato di alcune sigle presenti nella schermata 3.2.1:

- # Data: numero di pacchetti catturati (numero di IV univoci nel caso di WEP), compresi i pacchetti di broadcast.
- #/s: numero di pacchetti al secondo
- ENC: protocollo di criptazione utilizzato.
  OPN = nessuna criptazione
  WEP? = non sono stati catturati abbastanza pacchetti per decidere se WEB o WPA, WEP, WPA o WPA2
- CIPHER: L'algoritmo utilizzato per criptare (CCMP, WRAP, TKIP, WEP, WEP40, WEP104)
- AUTH: Il protocollo di autenticazione utilizzato. Il valore del campo può esseree MGT (WPA/WPA2 con server di autenticazione separato), SKA (WEP con chiave segreta condivisa), PSK (WPA/WPA2 con chiave condivisa), o OPN (sistema aperto)

3.2.2 Aircrack-ng

Aircrack-ng è stato utilizzato per recuperare le chiavi WEP utilizzate nelle comunicazioni registrate da Airodump. Per fare ciò è necessario un certo numero di pacchetti. Aircrack-ng utilizza diverse tecniche per individuare la chiave WEP (FMS, KOREK e PTW):

- FMS: Fluhrer, Mantin, Shamir è un attacco basato su uno studio di vulnerabilità dello stream cypher RC4 (vedi sezione 2.2) da parte dei tre ricercatori che poi hanno dato il nome a questo attacco. L'attacco si basa su due vulnerabilità di RS4. La prima vulnerabilità è dovuta al fatto che una piccola parte della chiave segreta determina un gran numero di bit della permutazione iniziale fornita dal KSA (key scheduling algorithm). La seconda vulnerabilità è causata dal fatto che parte della chiave utilizzata da KSA è trasmessa in chiaro (l'IV). Con questo algoritmo la chiave segreta può essere recuperata con un numero di pacchetti che varia da 4,000,000 a 6,000,000.

- KOREK: durante una discussione su di un forum[5], una persona sotto lo pseudonimo di Korek pubblicò un codice che descriveva 17 attacchi su protocollo WEP suddivisi in tre gruppi: il primo gruppo cercava di trovare la chiave dal primo byte generato dal PRGA, il secondo includeva la conoscenza dei primi due byte generati ed il terzo gruppo consisteva in un metodo in grado di ridurre lo spazio di ricerca chiamato *inverted attacks*. Con questo set di algoritmi la chiave segreta (104 bit) può essere recuperata con un numero di pacchetti che varia da 500,000 a 2,000,000.

- PTW: è un attacco basato su un precedente lavoro di Andreas Klein. Similmente al Korek attack in cui si utilizzano i primi due byte del PRGA, questo tipo di attacco utilizza il k-esimo byte generato per identificare il k+1-esimo byte della chiave. Con questo algoritmo la chiave segreta (104 bit) può essere recuperata utilizzando 85,000 pacchetti con una probabilità del 95%.

Una volta lanciato, Aircrack cerca di individuare, un byte alla volta, la chiave WEP. Per ogni i-esimo byte della chiave WEP, vengono eseguiti dei test statistici (FMS, KOREK e PTW) che assegnano dei punti ai byte che con più probabilità corrispondono al byte della chiave WEP. Di seguito viene visualizzato un esempio di schermata di Aircrack (fudge factor = 2, bruteforce dell'ultimo byte abilitato):

[5] NetStumbler

```
[00:00:10] Tested 77 keys (got 684002 IVs)

KB    depth   byte(vote)
 0    0/  1   AE( 199) 29(  27) 2D(  13) 7C(  12) FE(  12)
 1    0/  3   66(  41) F1(  33) 4C(  23) 00(  19) 9F(  19)
 2    0/  2   5C(  89) 52(  60) E3(  22) 10(  20) F3(  18)
 3    0/  1   FD( 375) 81(  40) 1D(  26) 99(  26) D2(  23)
 4    0/  2   24( 130) 87( 110) 7B(  32) 4F(  25) D7(  20)
 5    0/  1   E3( 222) 4F(  46) 40(  45) 7F(  28) DB(  27)
 6    0/  1   92( 208) 63(  58) 54(  51) 64(  35) 51(  26)
 7    0/  1   A9( 220) B8(  51) 4B(  41) 1B(  39) 3B(  23)
 8    0/  1   14(1106) C1( 118) 04(  41) 13(  30) 43(  28)
 9    0/  1   39( 540) 08(  95) E4(  87) E2(  79) E5(  59)
10    0/  1   D4( 372) 9E(  68) A0(  64) 9F(  55) DB(  51)
11    0/  1   27( 334) BC(  58) F1(  44) BE(  42) 79(  39)

KEY FOUND! [ AE:66:5C:FD:24:E3:92:A9:14:39:D4:27:4B ]
```

Nell'esempio visualizzato, ogni riga corrisponde ad un byte della chiave WEP. La prima colonna indica quale byte della chiave si sta cercando di recuperare, la seconda colonna indica invece il numero di possibili soluzioni che il programma testerà in quella posizione. Nelle altre colonne sono indicati i vari byte candidati come soluzione con, fra parentesi, il numero di voti ricevuti dai test statistici. Il numero di candidati da testare è deciso per ogni byte da un parametro chiamato *fudge factor*. Se Aircrack è stato lanciato con *fudge factor* uguale a 2, il programma prenderà in considerazione tutte le soluzioni con un numero di voti maggiore o uguale al numero di voti della soluzione più votata diviso per 2 (il valore di *fudge factor*). Aumentare il valore di *fudge factor* aumenta la probabilità di trovare la soluzione ma rallenta notevolmente l'esecuzione del programma. Nell'esempio possiamo notare come l'ultimo byte della chiave non compaia nella schermata. Questo è dovuto al fatto che Aircrack cerca di recuperare quest'ultimo byte attraverso bruteforce[6]. Vediamo di seguito una lista dei parametri più significativi con cui Aircrack può essere lanciato:

- -c : restringe la ricerca a caratteri alfanumerici (h20 - h7F)
- -h : restringe la ricerca a caratteri che rappresentano numeri (h30 - h39)
- -f 2 : setta il fudge factor a 2 (default)
- -x1 : abilita il bruteforce dell'ultimo byte (default)
- -n 128 : specifica la lunghezza della chiave da trovare pari a 128 bit (default)

[6] prova tutte le possibili combinazioni

3.2.3 Iperf

Iperf è un tool di test creato per misurare il throughput di una rete tramite la creazione di data stream ed il loro invio. Nell'esperimento è stato utilizzato per generare traffico WEP che potesse essere intercettato. Iperf può essere lanciato in due modalità:

- server: è la modalità di "ascolto" di iperf. Il programma viene lanciato definendo il tipo di protocollo adibito al trasporto (TCP o UDP) e opzionalmente la porta di ascolto del programma[7].
- client: è la modalità di "invio" di iperf: viene scelto l'ip del server a cui mandare i dati, il protocollo di trasporto, la banda utilizzata per il collegamento, il tipo di test da effetture (unidirezionale, bidirezionale o più thread in parallelo), il tempo di trasmissione dei dati e il numero di secondi dopo cui si vuole visualizzare un report.

3.3 Metodologia utilizzata

3.3.1 Tipologia di test

Per eseguire i test necessari a raccogliere i dati sperimentali, è stato eseguito iperf su di una STA autenticata ad una rete wireless protetta da WEP. Iperf è stato settato per trasmettere ad una velocità di 10 Mb/s (così da essere sicuri che trasmetta alla massima velocità). La soglia di frammentazione della STA è stata impostata a 256 byte. Siccome siamo interessati a misurare il numero di pacchetti necessari a recuperare la chiave segreta, l'unico scopo della configurazione della STA è quello di generare il maggior numero di pacchetti al secondo (così da poter catturare il numero di IV univoci desiderato nel minor tempo possibile). La configurazione dell'access point utilizzata è descritta in 3.1. La chiave WEP è stata inizialmente settata a 40 bit, scelti a caso. Abbiamo poi eseguito Aircrack con soglie di 10.000, 20.000, 30.000, 40.000 e 50.000 pacchetti catturati. Aircrack è stato eseguito in modalità standard (fudge factor = 2). In caso di un fallimento nel recupero della chiave, è stato incrementato il *fudge factor* a 4 e successivamente a 6. Per ogni soglia il test (cattura dei pacchetti + tentativo di cracking) è stato eseguito 5 volte. Per ogni test abbiamo registrato il numero di chiavi provate ed il tempo necessario al recupero della chiave segreta (in realtà siamo solo interessati a sapere se il recupero ha avuto successo oppure no).

[7] lanciato senza definire una porta specifica viene utilizzata di default la 5001

# IV univoci	fudge factor	test 1 secondi - chiavi	test 2 secondi - chiavi	test 3 secondi - chiavi	test 4 secondi - chiavi	test 5 secondi - chiavi
50.000	2	8s - 567k	8s - 530k	9s - 582k	9s - 566k	9s - 498k
40.000	2	8s - 456k	8s - 555k	9s - 10	1s - 1	1s - 1
30.000	2	1s - 214	5s - 1370	1s - 1	1s - 6	1s - 1
20.000	2	17s - 717	∞s - 138k	1s - 1	∞s - 134k	1s - 1
	4	-	9s - 169k	-	∞s - 272k	-
	6	-	-	-	∞s - 402k	-
	6*	-	-	-	∞s - 419k	-
10.000	2	∞s - 165k	∞s - 134k	∞s - 173k	∞s - 140k	∞s - 174k
	4	∞s - 346k	∞s - 278k	∞s - 342k	∞s - 346k	∞s - 342k
	6	∞s - 443k	∞s - 397k	∞s - 512k	∞s - 462k	∞s - 522k
	6*	∞s - 5152k	7s - 1886k	∞s - 4241k	∞s - 5227k	∞s - 3375k

Tabella 1: Tabella riassuntiva dei risultati dei test effettuati con chiave WEP a 40 bit. (6* indica che Aircrack è stato lanciato specificando la lunghezza della chiave WEP e con fudge factor = 6)

In modo analogo l'insieme di test è stato ripetuto con una chiave WEP di 104 bit utilizzando però soglie di 40.000, 50.000, 60.000, 70.000 e 80.000 pacchetti catturati.

3.3.2 Risultati

Dalla tabella 1 possiamo notare come la chiave da 40 bit sia stata recuperata con successo in tutti i test con almeno 30.000 IV univoci. L'incremendo del *fudge factor* permette inoltre di recuperare la chiave anche in situazioni in cui gli IV sono solo 10.000. Dai grafici 6 e 7 possiamo notare come l'utilizzo di un *fudge factor* maggiore permetta di migliorare la percentuale di successo di Aircrack indipendentemente dalla dimensione della chiave.

L'incremento della dimensione della chiave da 40 a 104 bit permette di migliorare, anche se di poco, la robustezza di WEP. Come si può notare confrontando la tabella 2 con la tabella 1 con 40.000 IV la percentuale di successo è del 20% con una chiave di 104 bit mentre è del 100% con una chiave di 40 bit. Aumentando però di poco il numero di pacchetti catturati con IV univoci (60.000), anche la chiave da 104 bit si dimostra alquanto debole con una percentuale di successo del 100%.

È interessante notare come in alcuni test Aicrack sia riuscito a trovare la chiave quasi istantaneamente con un numero di tentativi inferiore alla decina. Molto probabilmete questo è dovuto al fatto che sono stati raccolti un gran numero di weak IV, il che ha portato il programma ad identificare in pochi passaggi la chiave.

# IV univoci	fudge factor	test 1 secondi - chiavi	test 2 secondi - chiavi	test 3 secondi - chiavi	test 4 secondi - chiavi	test 5 secondi - chiavi
80.000	2	1s - 30k	∞s - 169k	1s - 30k	2s - 566k	1s - 30k
	4	-	∞s - 341k	-	-	-
	6	-	11s - 2782k	-	-	-
70.000	2	1s - 95k	1s - 30k	1s - 131	∞s - 162k	1s - 30
	4	-	-	-	∞s - 322k	-
	6	-	-	-	14s - 3765k	-
60.000	2	∞s - 150k	∞s - 138k	1s - 30k	2s - 423k	1s - 30k
	4	∞s - 308k	6s -1632k	-	-	-
	6	11s - 2913k	-	-	-	-
50.000	2	∞s - 151k	1s - 30k	∞s - 164k	∞s - 154k	1s - 30k
	4	∞s - 324k	-	∞s - 327k	∞s - 289k	-
	6	∞s - 489k	-	∞s - 399k	∞s - 464k	-
40.000	2	∞s - 157k	1s - 30k	∞s - 169k	∞s - 145k	∞s - 109k
	4	∞s - 219k	-	∞s - 251k	∞s - 330k	∞s - 318k
	6	∞s - 411k	-	∞s - 523k	∞s - 512k	∞s - 505k

Tabella 2: Tabella riassuntiva dei risultati dei test effettuati con chiave WEP a 104 bit

Parametro	Distribuzione uniforme	Insieme degli IV catturati
Valore minimo	385	0
Valore massimo	16777117	16777215
Media	8388751	8301102
Deviazione standard	4843025,37	4833928,52

Tabella 3: Confronto tra i valori degli IV raccolti ed i valori di una variabile casuale con distribuzione uniforme

3.3.3 Distribuzione degli IV

È stata verificata se la sequenza degli IV generata dalla NIC card utilizzata fosse casuale o seguiva, invece, una sequenzialità. Per verificare quest'ipotesi abbiamo calcolato media e deviazione standard di 52.485 IV e li abbiamo confrontati con i parametri di una variabile casuale discreta con distribuzione uniforme (n=52.485).

Dai dati della tabella 3 può sembrare che i valori siano molto simili a quelli di una variabile con distribuzione uniforme. Per verificare questa intuizione abbiamo raggruppato gli IV in 20 classi discrete (vedi figura 4) ed eseguito un test C^2 (chi quadro). Lo scopo del test C^2 è quello di conoscere se le frequenze osservate O differiscono significativamente dalle frequenze attese E (che sono quelle di una variabile con distribuzione uniforme). Abbiamo quindi calcolato $\chi^2 = \frac{(O-E)^2}{E}$. Essendo diviso in 20 classi dobbiamo confrontare questo valore con i valori della distribuzione C^2 con grado di libertà 19. Con una confidenza dell'1% il valore

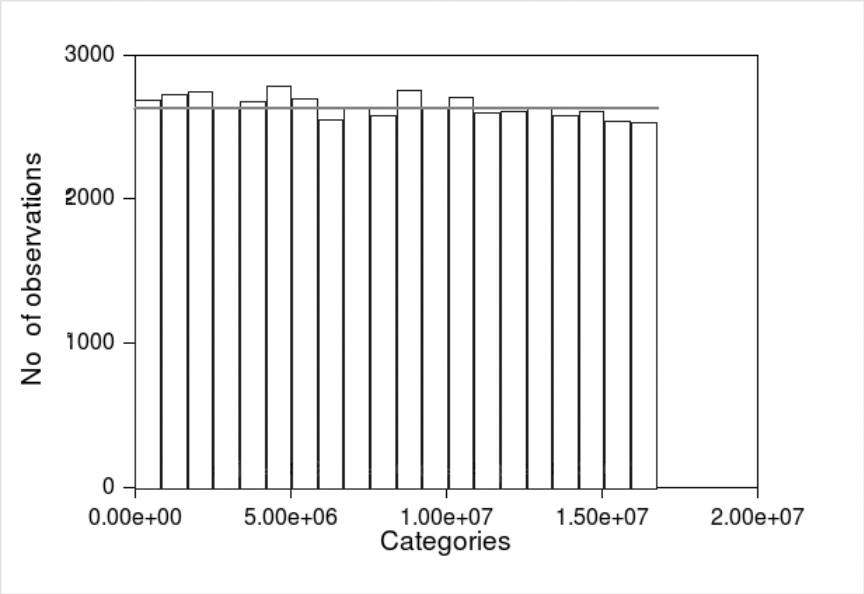

Figura 4: Frequenze discrete delle 20 classi di IV utilizzate per il test del χ^2

critico riportato nella tabella di C^2 è di 36,19. Dai calcoli, χ^2 risulta uguale a 39,61. Possiamo quindi osservare che con una confidenza dell'1% i dati raccolti non seguono una distribuzione uniforme (questi dati posso essere prodotti da una distribuzione uniforme con una probabilità inferiore all'1%).

Bisogna notare però che la distribuzione degli IV non deve essere necessariamente uniforme e che un'approssimazione di questa distribuzione è più che sufficiente allo scopo di differenziare gli IV utilizzati. I dati della tabella 3 ed il grafico 5 mostrano come questa distribuzione approssimi un'uniforme.

4 Conclusioni

Come abbiamo potuto osservare, una rete 802.11 che utilizza come sistema il protocollo WEP è facilmente soggetta ad attacchi che permettono ad utenti non autorizzati di penetrare all'interno della rete. L'utilizzo di chiavi a 104 bit, invece che a 40, mantiene inalterata la semplicità di recuperare la chiave, a costo di un maggior numero di pacchetti da catturare. Il fatto che lo standard non specifichi nessun vincolo sulla scelta dell'IV scelto dalla NIC card ha fatto si che molte case produttrici rilasciassero card che utilizzano IV sequenziali e prevedibili il che

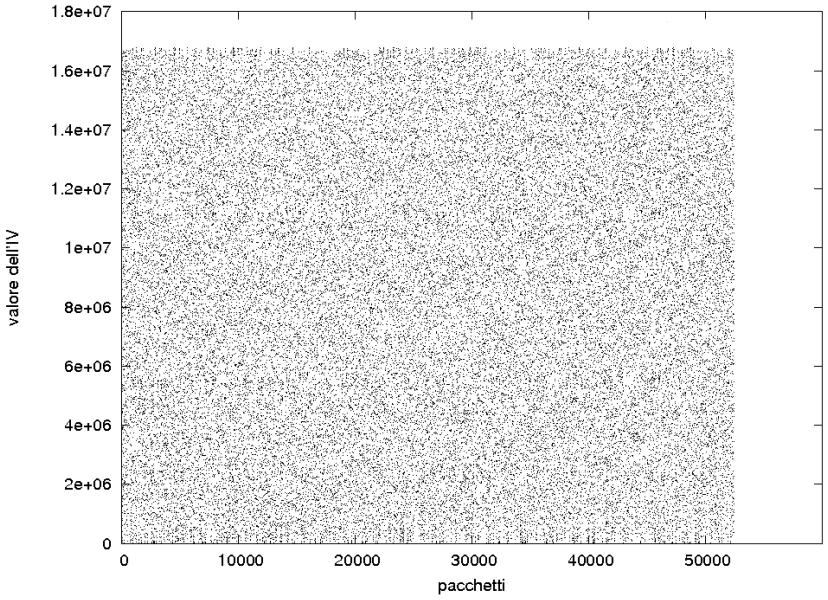

Figura 5: Distribuzione degli IV nel tempo

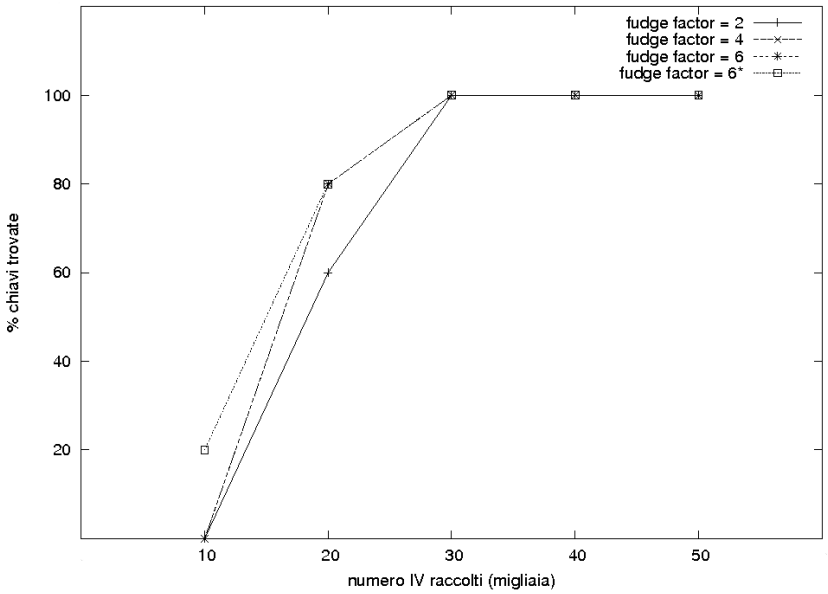

Figura 6: percentuale di riuscita nella decriptazione della chiave WEP a 40 bit

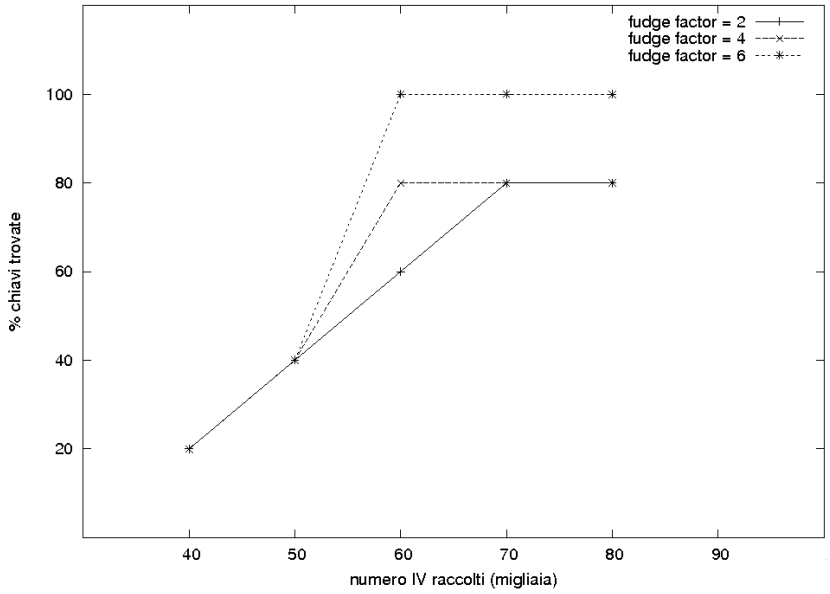

Figura 7: percentuale di riuscita nella decriptazione della chiave WEP a 104 bit

comporta una maggiore facilità di ricavare la chiave WEP. D'altro canto esistono altri vendor che impediscono l'utilizzo di weak IV scelti dalla scheda per comunicare.
Oramai, però' le reti wi-fi protette da questo algoritmo sono poche: grazie alla creazione di nuovi standard di sicurezza (802.1x) le reti wifi sono molto più protette e sicure e solo l'utilizzo di password semplici e banali permette ad attaccanti di accedere ad essa.

4.1 Confronto tra Iperf e Netper

In questa sezione vogliamo confrontare la precisione del tool Iperf utilizzato in laboratorio per eseguire i test della prima esperienza di laboratorio, con un altro tool chiamato Netperf. Il confronto è stato fatto dopo aver notato alcuni comportamenti non corretti del programma. In particolare è stato notato che alcune volte i valori del throughput potevano supearare quello teorico, il quale poteva influire erroneamente sul valore medio della misurazione.
La Figura 8 descrive la configurazione della rete utilizzata per il confronto ed è gia stata descritta nella prima relazione: un laptop su cui sono eseguiti entrambi i programmi a lato server, connessione cablata tramite switch all'access point (un

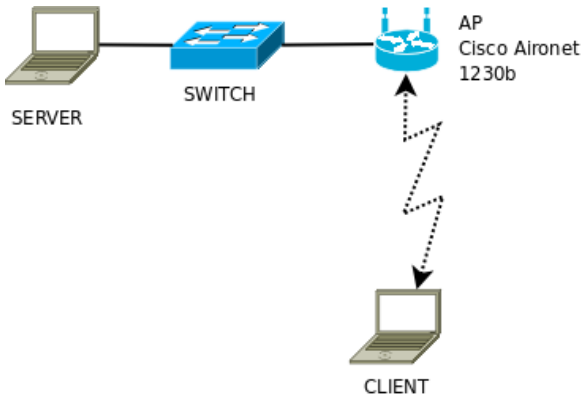

Figura 8: configurazione della rete utilizzata per il test di confronto

Cisco 1230b che supporta lo standard 802.11b) ed infine una connessione wifi al client che esegue i tool per misurare il throughput. L'AP è stato impostato per gestire traffico con velocità a 11 Mb/s, mentre sulla NIC card del client è stata disabilitata la frammentazione settando il fragmentation threshold a 2346. Per confrontare i due tool, è stato scritto uno script che eseguiva per 20 volte prima iperf e successivamente netperf. La durata di ogni sigola esecuzione è stata di 20 secondi. Nel grafico 9 è possibile vedere il troughput misurato dai due tool e notare come dopo una prima fase di linearità delle prestazioni della rete in cui i due tool registrano i valori molto simili, vi sono alcuni picchi dovuti probabilmente ad una congestione della rete o disturbi presenti sul canale. Il comportamento dei valori registrati dai due tool in questa seconda fase è comunque coerente: possiamo osservare come l'andamento dei valori sia simile anche se scostato nel tempo a causa dell' esecuzione dei test non concorrente. La tabella 4 mostra i dati ottenuti: la media del throughput misurato non è molto differente con un valore di 5.66 Mb/s per netperf e di 5 per iperf. Questi valori sono comunque contenuti all'interno del range definito da $media \pm dev.std$. Osservando poi i valori massimi registrati, le misurazioni sono molto simili facendo registrare un picco di 6.31 Mb/s per netperf e di 6.21 Mb/s per iperf. I valori minimi sono, però, molto differenti: nel caso di netperf il minimo è di 2.04 Mb/s mentre per iperf è di 0.77 Mb/s. Questi ultimi dati non mostrano nessun tipo di risultato evidente, poichè questi valori sono probabilmente dovuti a disturbi nella rete e che possono rovinare la bontà delle misurazione.
Possiamo concludere che i valori segnalati da Netstat possono confermare la correttezza delle misurazioni di Iperf.

	netperf	iperf
Media (Mb/s)	5.66	5
Dev. Std. (Mb/s)	0.95	1.55
Max (Mb/s)	6.31	6.21
Min (Mb/s)	2.04	0.77

Tabella 4: Tabella riassuntiva dei risultati dei test effettuati con chiave WEP a 104 bit

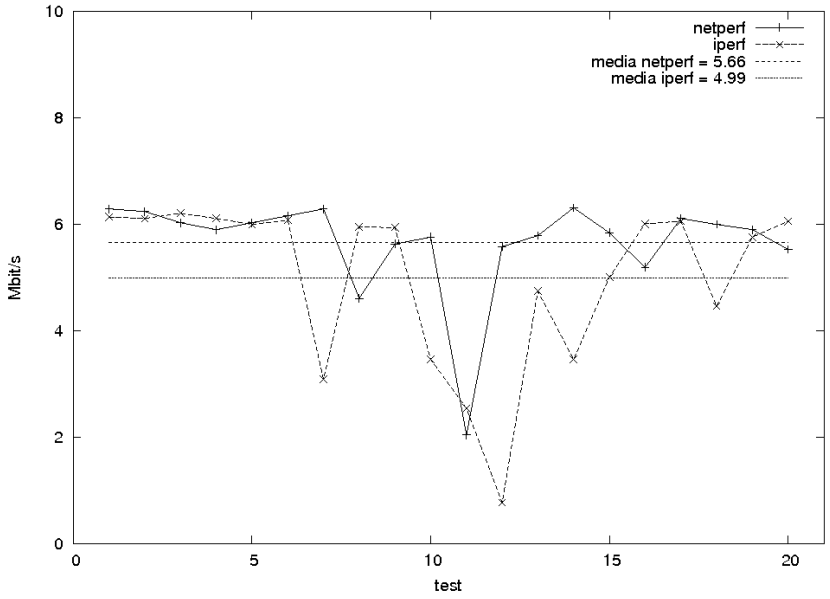

Figura 9: Throughput dei due tool a confronto