



Alessio Marziali  
(alessio.marziali@cyphersec.com)  
www.cyphersec.com  
Marzo 21 - 2007

Note : Prima di leggere questo articolo e' necessario capire che *fare breccia in networks altrui e' reato* ed è perseguibile a norma di legge. Lo scopo di questo articolo e' quello di dimostrare quanto sia vulnerabile la cifratura WEP e WPA e come sia da evitarne l'utilizzo in ambienti enterprise. Portare a termine un crack di una rete WEP/WPA e' necessario essere in possesso dei seguenti strumenti.

#### Strumenti Software:

- Kismet (<http://www.kismetwireless.net>)
- Aircrack Suite (<http://www.aircrack-ng.org/doku.php>)

#### Strumenti Hardware:

- Una qualunque scheda wifi

#### La Teoria

Molte aziende subiscono attacchi di tipo informatico e spesso non se ne rendono conto, gli utenti domestici percepiscono un sensibile calo di velocità della propria connessione e maledicono le compagnie telefoniche. Il protocollo WEP (Wired Equivalent Protocol) E' un protocollo che definisce un insieme d'istruzioni e regole tramite i dati possono viaggiare nell'aria con un minimo di sicurezza. Per la codifica dei dati WEP utilizza l'algoritmo RC4 nel momento in cui i dati "escono" dal AP. Il problema della sicurezza nel mondo WEP deriva dall'errata implementazione dell'algoritmo RC4, non l'algoritmo in se stesso. RC4 è, e credo lo sarà ancora per diverso tempo, un algoritmo sicuro, basti pensare che viene utilizzato pesantemente all'interno di SSL (Secure Socket Layers) e quindi nella maggior parte degli applicativi legati al commercio elettronico. Ma come è possibile sfruttare una implementazione errata per fare breccia all'interno di una rete privata? Non è una cosa che richiede molto tempo, una stima recentemente condotta ha dimostrato che una rete con un traffico costante può essere bucata nel giro dalle 2 alle 6 ore. Per capire come effettuare il cracking di una rete wifi è necessario comprendere dove risiede l'errore. Come abbiamo detto WEP utilizza l'algoritmo RC4. RC4 utilizza una cifra di emissione che crea una Unique Key (denominata Packet Key) per ogni pacchetto che viene codificato. Questa operazione viene eseguita grazie alla combinazione di varie caratteristiche di una password condivisa, un valore di stato ed un vettore d'inizializzazione (IV) il quale viene usato per "offuscare" i dati. Tutti questi componenti e passaggi vengono definiti "Key Scheduling Algorithm" (KSA). Il risultato di queste operazioni è un array il quale viene utilizzato per generare un algoritmo pseudocasuale (PRGA) che a sua volta produce un flusso di dati messo in XOR con il plain text (messaggio) il quale alla fine produce il cypher text che viaggia sopra le nostre teste. Questo messaggio è sicuramente differente dal messaggio originale, di fatto contiene un valore che viene identificato come checksum. Il checksum è un valore unico computato da dati contenuti nel pacchetto, il quale viene utilizzato per garantire l'integrità dei dati. Quando il pacchetto viene ricevuto dall'host che ne ha fatto richiesta, viene ricalcolato il checksum e confrontato con quello originale, se i due checksum corrispondono il pacchetto viene accettato e processato, altrimenti viene scartato. Molti sistemi di IDS verificano ogni pacchetto che viaggia sulla rete ed effettuano questa operazione proprio per rilevare presenze di AP rogue i quali possono effettuare operazioni di MITM. Una volta che il dato viene decifrato, l'IV viene anteposto ai dati, insieme ad un bit che segna il pacchetto come cifrato. Tutto questo viene

trasmesso nell'atmosfera, da dove viene poi catturato e decifrato dal legittimo destinatario. Il processo di decifrazione è l'esatto opposto del processo di cifratura. Come prima cosa, l'IV viene rimosso dal pacchetto, e relazionato con la password condivisa. Questo valore viene utilizzato per ricreare la KSA, che viene conseguentemente utilizzata per ricreare il key-stream. Il flusso e il pacchetto di dati cifrati vengono quindi messi in XOR, e la risultante è il testo in chiaro. Alla fine, il CRC viene rimosso dal testo in chiaro e comparato con il CRC ricalcolato: a questo punto, il pacchetto viene accettato oppure scartato.

### Analizzare i dati Codificati

Evitare che i dati vengano intercettati durante il loro viaggio è impossibile, l'unica cosa che può fare il WEP è proteggerli dall'interpretazione una volta catturati. Partendo da questo dato di fatto iniziamo di nuovo dalla teoria. Come è giusto supporre, dopo aver determinato che i dati vengono codificati prima dell'invio il testo catturato ed un testo in chiaro conosciuto possono essere messi in XOR per produrre il keystream generato dal PRGA. La ragione di questo è che il WEP produce il testo cifrato unendone solamente due variabili e mettendole in XOR

```
+=====+
                CyperText =
          (PlainText XOR keyStream)
+=====+
```

Come si può vedere l'unico valore che maschera il testo in chiaro è il keystream. Se invertiamo il processo, vedremo che l'unico valore che maschera il keystream è il testo in chiaro.

```
+=====+
                Keystream =
          (CyperText XOR PlainText)
+=====+
```

Per estrarre il testo tutto quello che serve è uno sniffer wireless, e state tranquilli che in una rete moderna possiamo raccogliere MB e MB di dati da confrontare.

Come precedentemente descritto, il WEP usa un IV per cifrare ogni pacchetto. Se il mittente usa un IV per cifrare, il ricevente deve conoscere il valore corrispondente per decodificare. Un IV è composto da 3 bytes per ogni pacchetto WLAN, quando questo dato viene inviato, IV viene anteposto al pacchetto cifrato. Come tutti sappiamo un byte è composto da 8 bit, pertanto con una serie di calcoli è possibile affermare che la chiave di decodifica può assumere un valore di 16.777.216. Tanto eh? Per niente..

1 byte = 8 bits  
3 byte = 24 bits

1 bit = 0 / 1  
IV Keys =  $2^{24} = 16.777.216$

Questo non significa propriamente che dobbiamo aspettarci una collisione IV dopo il trasferimento di 16 milioni di pacchetti, nella maggior parte dei casi s'iniziano a vedere collisioni dopo i primi 300.000 pacchetti. Quindi

$2000 * 1500 = 4500000$   
 $4500000 / 1000000 = 4.5\text{MB}$

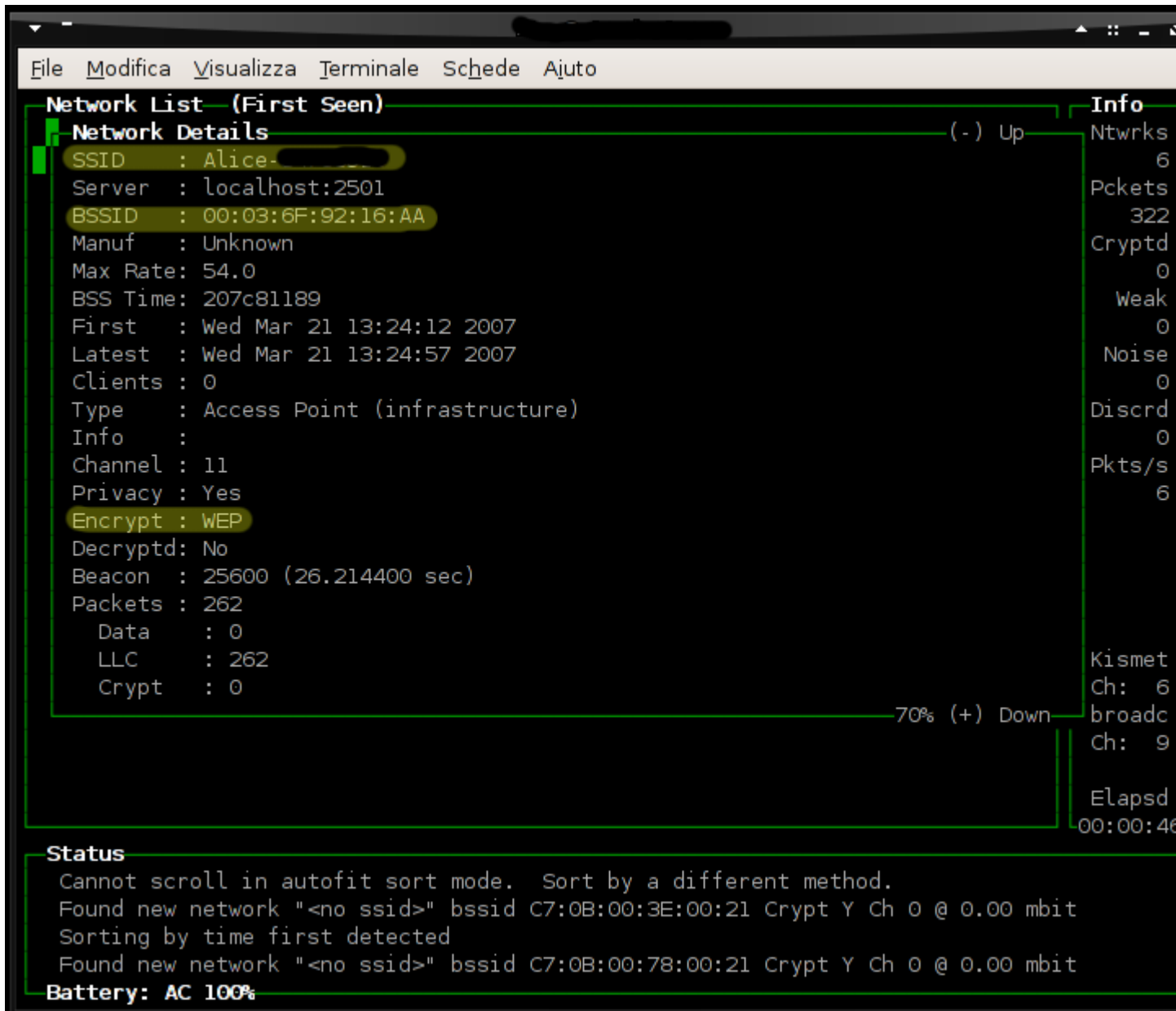
Il keystream viene calcolato da una serie di proprietà della password e di IV. Nel caso in cui si verifichi una collisione IV è disponibile sotto forma di tre caratteri "1:2:3".

### La Pratica

Dopo aver configurato tutti gli strumenti, avviamo Kismet e verifichiamo le reti disponibili. Kismet è uno strumento molto potente, ma decisamente ostico da controllare. Il sistema di navigazione è basato su

shortcuts, quindi è bene ricordarsi per lo meno i comandi base. E' comunque sempre disponibile un help semplicemente digitando "h".

Appena avviato Kismet si presenta con una interfaccia verde su sfondo nero, dopo aver ordinato la lista delle reti disponibili (comando s) selezionare la rete che si vuole attaccare e confermare tramite il tasto Enter. Si riceverà un output di questo tipo



```
File Modifica Visualizza Terminale Schede Ajuto
Network List (First Seen)
Network Details (-) Up
SSID : Alice-
Server : localhost:2501
BSSID : 00:03:6F:92:16:AA
Manuf : Unknown
Max Rate: 54.0
BSS Time: 207c81189
First : Wed Mar 21 13:24:12 2007
Latest : Wed Mar 21 13:24:57 2007
Clients : 0
Type : Access Point (infrastructure)
Info :
Channel : 11
Privacy : Yes
Encrypt : WEP
Decryptd: No
Beacon : 25600 (26.214400 sec)
Packets : 262
  Data : 0
  LLC : 262
  Crypt : 0
70% (+) Down
Info
Ntwrks 6
Pckets 322
Cryptd 0
Weak 0
Noise 0
Discrd 0
Pkts/s 6
Kismet
Ch: 6
broadc
Ch: 9
Elapsd
00:00:40
Status
Cannot scroll in autofit sort mode. Sort by a different method.
Found new network "<no ssid>" bssid C7:0B:00:3E:00:21 Crypt Y Ch 0 @ 0.00 mbit
Sorting by time first detected
Found new network "<no ssid>" bssid C7:0B:00:78:00:21 Crypt Y Ch 0 @ 0.00 mbit
Battery: AC 100%
```

### Intercettare i dati in volo

Ora che si è accertata la presenza di una rete wifi con cifratura WEP è necessario iniziare ad effettuare il dump dei pacchetti. Se la rete identificata è utilizzata non ci dovrebbe voler molto. Da terminale digitiamo il comando

```
airodump-ng <channel> --ivs --write <file output> <interfaccia wireless>
```

La quantità di pacchetti necessaria è variabile, diciamo che con un valore di 300.000 si possono già iniziare ad osservare gli IVS in collisione.

### Recuperare la chiave

Ora che i pacchetti sono disponibili, si passa ad aircrack-ng. Da terminale si lancia

```
aircrack-ng -a 1 -e <ssid> -b <bssid> <nome file dump>
```

dove <nome file dump> identifica il file precedentemente creato.

Attendere fino al termine del calcolo, recuperare la chiave ed accedere alla rete.

Nel caso in cui l'algoritmo utilizzato sia WPA, allora è necessario eseguire

```
aircrack-ng -a 2 -e <ssid> -b <bssid> -w <file dizionario> <nome file dump>
```

Come si può notare, il comando varia leggermente. Questo è dovuto dal fatto che WPA non è vulnerabile ad attacchi statistici, pertanto è necessario forzare la chiave con un attacco di tipo BruteForce. I file dizionario possono essere recuperati con molta facilità su internet.