Università degli Studi di Milano – Polo Didattico e di Ricerca di Crema
Corso di Laurea in Sicurezza dei Sistemi e delle Reti Informatiche

ANALISI DELLE VULNERABILITÀ DEL PROTOCOLLO WEP

Danilo Vizzarro
dvizzarro@crema.unimi.it
www.danilovizzarro.it
22 novembre 2005

INDICE

ABSTRACT

Lo Standard IEEE 802.11 adotta il protocollo di crittografia WEP (Wired Equivalent Privacy) per tutelare la sicurezza delle comunicazioni nelle reti wireless. È stato dimostrato che tale protocollo presenta delle gravi vulnerabilità che rendono possibile l'accesso abusivo alle reti che lo implementano. In questo documento verranno discusse le modalità che permettono violare il protocollo WEP.

1. INTRODUZIONE

Le reti wireless hanno guadagnato negli ultimi anni grande popolarità in quanto permettono di eseguire tutte le operazioni possibili nelle reti cablate, senza la necessità di collegare via cavo i client della rete. La trasmissione dei dati è in broadcast e utilizza onde radio ad una frequenza di 2.4 GHZ. Questo significa che chiunque disponga di un hardware in grado di ricevere onde radio della stessa frequenza, può intercettare le trasmissioni. Nasce da qui la necessità di proteggere le comunicazioni crittografandole ed è proprio di questo che si occupa il WEP. Sfruttando alcune vulnerabilità del protocollo, è però possibile portare a termine con successo attacchi di intercettazione e di tampering.

2. IL PROTOCOLLO WEP

Ci sono 2 tipi di implementazioni del WEP:

1. La **Standard Implementation** che utilizza una chiave di 40 bit. La lunghezza di questa chiave rende il protocollo vulnerabile ad attacchi di tipo Brute Force, anche utilizzando modeste risorse di computazione.
2. L'**Extended Implementation** che utilizza una chiave di 104 bit. Si ritiene che questa versione del protocollo non sia soggetta ad attacchi Brute Force, in quanto sarebbero necessari tempi enormi per provare tutte le possibili combinazioni.

2.1 **Obiettivi del WEP**

Il WEP è stato progettato per tutelare:

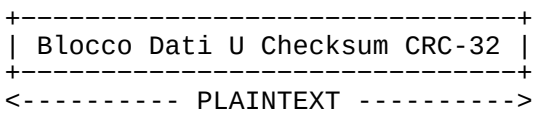
- La **confidenzialità** dei dati che viaggiano su un canale cifrato per prevenire le intercettazioni abusive;
- Il **controllo dell'accesso** ovvero proteggere l'infrastruttura wireless dagli accessi esterni;

- L'**integrità** dei dati per prevenire il tampering dei messaggi utilizzando la checksum CRC-32.

2.2 **Design del protocollo**

Il protocollo WEP utilizza una chiave 'K' detta Wep Key che viene condivisa tra le parti comunicanti. Il processo di crittografia è costituito da alcune fasi fondamentali.

1. **Checksumming**: I dati del messaggio M da cifrare vengono divisi in blocchi di lunghezza fissa e per ogni blocco viene calcolata una checksum CRC a 32 bit indicata con c(M) che viene concatenata al blocco da cifrare. Tale concatenazione costituisce il PlainText P= (M,c(M)). Si noti che c(M) non dipende dalla Wep Key.

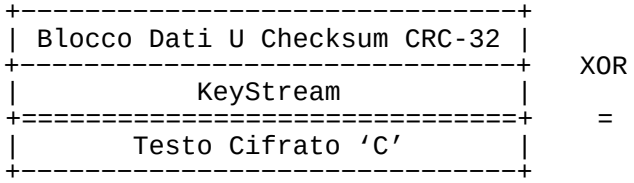


2. **Cifratura**: Si sceglie casualmente un vettore di inizializzazione 'IV' di 24 bit che viene concatenato alla Wep Key. Alla stringa ottenuta viene applicato l'algoritmo RC4 che genera una chiave di cifratura di lunghezza fissa detta 'KeyStream'.

$$KeyStream = RC4(IV \cup K)$$

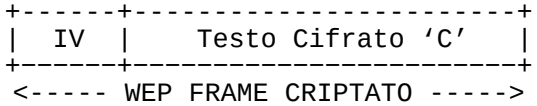
   Si esegue poi l'operazione di XOR tra il Plaintext e il KeyStream e il risultato ottenuto è il Testo Cifrato 'C'.

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



3. **Trasmissione**: Il pacchetto inviato nel network sarà composto dal vettore di inizializzazione 'IV' concatenato al Testo Cifrato 'C'.

$$A \rightarrow B : (IV, P \oplus RC4(IV,K)) \text{ con } P = (M,c(M))$$



4. **Decifratura**: avviene esattamente nel modo inverso. Il destinatario concatena l'IV che è contenuto nel pacchetto inviato e la Wep Key che conosce e cifra la stringa ottenuta utilizzando l'RC4 ottenendo cosi il KeyStream.. Potrà quindi eseguire l'operazione di XOR tra il KeyStream e il Testo Cifrato 'C' per calcolare il PlainText. È possibile verificare che il pacchetto sia giunto a destinazione senza essere modificato, controllando che la checksum sia corretta.

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

2.3 **Il vettore di inizializzazione IV**

Il Vettore di Inizializzazione IV ha una dimensione di 24 bit ed è trasmesso in chiaro per far in modo che il destinatario possa calcolare il rispettivo KeyStream concatenando lo stesso alla Wep Key. L'IV è una sequenza di bit generata dall'Access Point o dalla Wireless Card utilizzata dal client. Il WEP raccomanda che l'IV sia diverso per ogni pacchetto, ma non specifica in che modo debba variare. Se lo stesso IV fosse usato più di una volta con la stessa chiave WEP, il protocollo non sarebbe sicuro (si veda Cap. 3.3). Nonostante la variazione dell'IV sia alla base della sicurezza della comunicazione, molte implementazioni, in particolare quelle delle schede PCMCIA, si limitano ad incrementare il vettore di 1 ogni volta che un pacchetto viene trasmesso. Dato che l'IV ha dimensione 24 bit, è possibile creare circa 17 milioni di KeyStream diversi. Inviando pacchetti di 1500 Bytes a 11Mbps sono sufficienti poco più di 5 ore per esaurire tutti i vettori di inizializzazione. A ciò si aggiunge che molte schede PCMCIA sono resettate a 0 ogni qual volta vengono re-inizializzate (la re-inizializzazione avviene quando si inserisce una scheda nel notebook). Se poi consideriamo che in una rete wireless solitamente ci sono più utenti, è facile capire che la possibilità di trovare pacchetti aventi un vettore di inizializzazione con valore basso diventa molto frequente.

2.4 **WEP checksum CRC-32**

Il CRC-32 (Cyclical Redundancy Check) è un algoritmo di checksum che prende in input un blocco di dati di lunghezza variabile e restituisce un output di lunghezza 32 bit. Nel WEP la checksum viene calcolata prima della cifratura dei dati e viene poi allegata al pacchetto. A destinazione viene ricalcolata e confrontata con quella contenuta nel pacchetto per verificare che lo stesso non sia stato alterato abusivamente. I pacchetti con checksum non corretta vengono scartati. La CRC-32 applicata al WEP rispetta 2 proprietà fondamentali:

1. È una funzione lineare del messaggio. Questo significa che

$$c(x \oplus y) = c(x) \oplus c(y)$$

2. È una funzione non dipendente dalla Wep Key, ovvero, per calcolarla è sufficiente conoscere il PlainText.

Nel prossimo capitolo verrà analizzato in che modo è possibile utilizzare queste proprietà per manipolare i pacchetti.

3. ATTACCHI AL WEP

Per attacchi di tipo passivo è necessario disporre di una scheda wireless in grado di monitorare il traffico radio alla frequenza di 2.4GHZ, per attaccare attivamente tale scheda deve essere in grado non solo di monitorare il traffico, ma anche di trasmettere dati alla stessa frequenza.

3.1 **Known PlainText Attack**

Se facciamo lo XOR di due pacchetti cifrati con gli stessi IV e Wep Key si ottiene un risultato interessante:

$$C1 \oplus C2 = (P1 \oplus KeyStream) \oplus (P2 \oplus KeyStream) = P1 \oplus P2$$

Questo significa che dati due pacchetti cifrati, è possibile ottenere lo XOR dei corrispondenti pacchetti in charo. Di conseguenza se un attaccante conoscesse uno dei due PlainText, potrebbe conoscere il contenuto dell'altro in questo modo:

$$P1 \oplus (C1 \oplus C2) = P1 \oplus (P1 \oplus P2) = P2$$

Inoltre, conoscendo un PlainText e il relativo Testo Cifrato, è possibile ottenere il KeyStrem facendone lo XOR.

$$P1 \oplus C1 = P1 \oplus (P1 \oplus KeyStream) = KeyStream$$

Conoscendo alcuni KeyStream e i relativi vettori d'inizializzazione IV, è anche possibile sfruttare una debolezza dell'algoritmo RC4 e ottenere la Wep Key[1] utilizzando la criptoanalisi differenziale. Saranno ora esaminati alcuni metodi per ottenere una valida coppia PlainText e Testo Cifrato.

3.2 **Authentication Spoofing**

Gli Access Point per identificare i client wireless utilizzano un sistema di shared key authentication costituito da alcune fasi fondamentali[2]:

1. Il client wireless (detto Initiator) invia un authentication request management frame (una richiesta di autenticazione), non cifrato, dove viene specificato che l'Initiator ha intenzione di usare la shared key authentication;
2. L'Access Point (detto Responder) risponde inviando un authentication management frame di challenge, contenente una stringa random di 128byte di testo in chiaro;
3. L'Initiator riceve il messaggio e invia al Responder lo stesso challenge cifrato con il WEP;
4. Il Responder decifra il messaggio e verifica che la checksum CRC-32 sia valida.

Il challenge si ripete per ogni host che vuole autenticarsi e questo significa che intercettando una singola sequenza di autenticazione, è possibile avere una coppia PlainText e rispettivo Testo Cifrato. L'attaccante a questo punto ha la possibilità di calcolare il KeyStream e risalire da questo alla Wep Key.

3.3 **KeyStream Reuse**

È comune nelle infrastrutture wireless che la Wep Key venga cambiata molto raramente. Dato che i vettori di inizializzazione (si veda Cap. 2.3) vengono esauriti in un tempo relativamente breve, diventa facile per un attaccante trovare dei vettori duplicati. A vettori duplicati corrispondono KeyStream duplicati e ciò espone il network all'attacco KeyStream Reuse. Una volta trovati due pacchetti che utilizzano lo stesso IV ci sono vari metodi per ottenere il PlainText. Se il testo in chiaro di un pacchetto è noto, è facile trovare il PlainText dell'altro pacchetto. In alternativa c'è da considerare che molti campi dei pacchetti IP sono facilmente predicibili in quanto utilizzano strutture standard. Un altro modo per ottenere dei PlainText, è inviare pacchetti di qualsiasi tipo da un Host collegato ad Internet controllato dall'attaccante, con lo scopo di generare traffico e intercettare la corrispondente cifratura.

3.4 **Packet Tampering**

Una conseguenza della prima proprietà della WEP checksum, (si veda Cap. 2.4) è la possibilità di apportare delle modifiche controllate al Testo Cifrato facendo in modo che la checksum resti valida. Supponiamo che si voglia inviare un Testo Cifrato 'C'. Risulterà che:

$$C = RC4(IV,K) \oplus (M,c(M))$$

Se applico ad M una variazione Δ eseguendo lo XOR (M⊕Δ), è possibile mantenere valida la checksum applicando la stessa variazione Δ anche a c(M) eseguendo lo XOR (c(M)⊕Δ).

Quindi se:

$$C = RC4(IV,K) \oplus (M,c(M))$$

Applicando la variazione ottengo un nuovo testo cifrato C1 dove:

$$C1 = RC4(IV,K) \oplus ((M \oplus \Delta),(c(M) \oplus \Delta))$$

Che per le proprietà dello XOR equivale a:

$$C1 = RC4(IV,K) \oplus (M,c(M)) \oplus (\Delta + c(\Delta))$$

Ma dato che $C = RC4(IV,K) \oplus (M,c(M))$ andando a sostituire si ha che:

$$C1 = C \oplus (\Delta + c(\Delta))$$

Questo significa che facendo variare in modo controllato i bit del Testo Cifrato 'C' e della rispettiva checksum, è possibile alterare il contenuto del messaggio.

3.5 **Packet Injection**

Una conseguenza della seconda proprietà della checksum (si veda Cap. 2.4) è che la stringa di controllo può essere calcolata anche da un avversario che possiede una valida coppia PlainText/Testo Cifrato, ma che non conosce la Wep Key. Infatti (si veda Cap. 3.1) è noto che P1⊕C1 = KeyStream. Si conosce anche l'IV associato al Testo Cifrato C1 (si veda Cap. 2.2), di conseguenza è sufficiente scegliere un PlainText qualsiasi, eseguire lo XOR con il Keystream ed ottenere un valido Testo Cifrato. Basterà poi concatenare l'IV al Testo Cifrato ottenuto per creare un pacchetto che l'Access Point considererà valido a tutti gli effetti. Questo attacco è noto come Packet Injection in quanto permette di inviare pacchetti di qualsiasi tipo nella Wireless Lan attaccata.

3.6 **Ip Redirection**

Condizione necessaria per effettuare un attacco di questo tipo è disporre di una connessione ad Internet. L'idea fondamentale consiste nell'intercettare un pacchetto crittografato generato dalla rete wireless e modificarlo utilizzando le tecniche illustrate nel Cap. 3.4, facendo in modo che l'indirizzo di destinazione del pacchetto sia un Host

di Internet controllato dall'attaccante. L'Access Point riceverà il pacchetto cifrato, provvederà a decifrarlo e ad inviarlo al nuovo indirizzo. L'attaccante potrà quindi disporre del Testo Cifrato e del rispettivo PlainText e potrà risalire alla Wep Key utilizzando un Known PlainText Attack. La modifica dell'indirizzo può apparentemente sembrare un'operazione complessa. In realtà i pacchetti intercettati del traffico in entrata saranno destinati ad indirizzi ip della rete facili da determinare. Questo significa che dato il Testo Cifrato di un indirizzo ip e ottenuto il PlainText dello stesso, sarà sufficiente calcolare la variazione Δ (si veda Cap. 3.4) che c'è tra il PlainText dell'indirizzo ip noto e il Δ dell'indirizzo ip dell'host di Internet controllato dall'attaccante ed eseguire lo XOR del Testo Cifrato dell'indirizzo ip noto, con la variazione Δ considerata[3]. Arrivato a destinazione il pacchetto, l'attaccante potrà disporre di una valida coppia di PlainText/Testo Cifrato e potrà procedere con la decodifica della Wep Key.

3.7 **Reaction Attack**

Anche in questo attacco la checksum svolge un ruolo fondamentale. Quando un host invia ad un Access Point (AP) un pacchetto TCP, viene esaminata la checksum: se essa è valida l'AP invierà all'host un pacchetto di acknowledgement. C'è da notare che i pacchetti di acknowledgement sono facilmente identificabili dalla loro dimensione, quindi, anche se sono cifrati, si potrà comunque capire (senza conoscere il l'esatto contenuto dell'acknowledgement) se la checksum inviata nel primo pacchetto TCP era valida o meno. Viene poi sfruttata una vulnerabilità della checksum CRC-32: dato un blocco in input e la sua corrispettiva checksum è possibile generare, in modo controllato, un altro blocco che abbia la stessa checksum del primo a condizione che sia verificata questa uguaglianza $P[i] \oplus P[i+16] = 1$, con 'P' che è il PlainText del blocco di input e 'i' la posizione di un bit. L'attacco viene eseguito in questo modo: dato un Testo Cifrato 'C' si esegue lo XOR $C \oplus \Delta$ dove Δ ha valore 1 nella posizione 'i' scelta casualmente e nella posizione 'i+16' e avrà valore 0 in tutte le altre posizioni. In questo modo si ottiene:

$$C1 = C \oplus \Delta$$

Esiste una proprietà dell'addizione modulo (2^16 -1) che afferma che:

$$P[i] \oplus \Delta \equiv P \bmod (2^{16} - 1) \text{ se } P[i] \oplus P[i+16] = 1$$

Dato che assumiamo che la checksum è valida per il pacchetto originale, questo significa che risulterà valida per il nuovo pacchetto solo quando $P[i] \oplus P[i+16] = 1$. Questo ci dà un'importante informazione sulla struttura del PlainText.

4. CONCLUSIONE

Il WEP era stato progettato per garantire la confidenzialità dei dati, l'integrità degli stessi e per controllare gli accessi alle reti wireless, ma è stato dimostrato che effettivamente non tutela nessuno di questo aspetti a causa di gravi vulnerabilità. Purtroppo non è cosi semplice introdurre nuovi protocolli in quanto già milioni di schede wireless sono in uso e adottando nuovi protocolli bisogna fare in modo, per ovvie ragioni di mercato, che le nuove funzionalità siano compatibili con l'hardware esistente che presenta risorse molto limitate. Và tuttavia sottolineato che se non fosse stata scoperta la falla nell'RC4 che permette di determinare la Wep Key dato il KeyStream, il WEP sarebbe molto più sicuro. Risulta comunque conveniente adottare delle contromisure per avere una minima tutela dei propri dati.

4.1 **Contromisure pratiche**

C'è da notare che la maggior parte delle reti wireless non utilizza neanche il WEP per proteggere i propri dati e sono quindi vulnerabili ad attacchi di qualsiasi tipo. Attivando il WEP l'attaccante non avrà accesso istantaneo alla rete e probabilmente deciderà di attaccare un'altra rete non protetta. Un consiglio è utilizzare la versione Extended del protocollo con una Wep Key di 104 bit per prevenire attacchi di tipo Brute Force. C'è anche da considerare che, non essendo previsto alcun protocollo per l'aggiornamento delle Wep Key, la configurazione degli Access Point e delle schede wireless è manuale. È preferibile che gli amministratori di rete provvedano personalmente al setting delle schede, aggiornando frequentemente la Wep Key e mantenendola segreta senza rivelarla agli utenti della rete. Ultima contromisura, probabilmente la più efficace, è il Mac Filtering. È conveniente configurare gli Access Point in modo tale che accettino solo pacchetti provenienti dagl'indirizzi Mac degli utenti della rete.

5. REFERENCES

1. A. Pasquinucci. Un'analisi dei problemi del wep. Apr. 2003.
2. W. A. Arbaugh, N. Shankar, and Y. J. Wan. Your 802.11 wireless network has no clothes. http://www.cs.umd.edu/~waa/wireless.pdf, Mar. 2001.
3. N. Borisov, I. Goldberg, D. Wagner. Intercepting mobile communication: the insecurity of 802.11. http://www.isaac.cs.berkeley.edu/isaac/wep-draft.pdf.