La chiave? E' sotto lo zerbino...

Alessandro Barenghi

Dipartimento di Elettronica e Informazione
barenghi - at - elet.polimi.it

21 gennaio 2011

Missione Impossibile?

- Come vi comportereste se doveste parlare solo a una persona che avevate conosciuto, ma :
  - Siete in una stanza affollata
  - Potete solo parlare attraverso un megafono
  - e tutti quelli accanto a voi riescono a riprodurre la vostra voce
  - Non vi ricordate che viso avesse la persona...[1]
- Punti bonus se l' altra persona è in grado di essere sicura di aver capito cosa dite

[1] ... neppure come si chiamava, con che voce parlasse...

Wireless Lan

- Scopo: evitare la scocciatura del cavo fisico
- Nate come alternativa alla rete cablata per l' "ultimo metro"...
- In realtà lo sono per l' ultimo centinaio di metri :)
- Utili per condividere l' accesso a una rete con chiunque nel raggio d' azione

Chiunque?

- Di solito, “chiunque” è un po’ troppo vasto come insieme...
- Stessi problemi delle vecchie LAN basate su ripetitori
- Nessuna garanzia di confidenzialità per i client connessi
- Nessuna autenticazione di chi si connette
- Nessuna garanzia contro l’iniezione di pacchetti non integri

La cassetta degli attrezzi

- La crittografia fornisce una cassetta per attrezzi per garantire quello che vorremmo
- Cifrare il traffico consente di avere confidenzialità
- Autenticare gli utenti evita gli accessi non autorizzati
- Usare algoritmi di hashing consente di garantire l' integrità

Un primo tentativo ...

- 1997: IEEE introduce il protocollo Wired Equivalent Privacy
- Una chiave comune per cifrare il traffico, 40-108(-232) bit
- Un Initialization Vector (IV) da 24 bit, noto, che agisce da sale
- L' autenticazione è gestita tramite la chiave comune
- L' integrità è gestita tramite CRC32

... finito male

- Nel 1997 WEP non poteva essere rotto, ma ...
- Alcune scelte di progetto lasciano dei dubbi[2]:
  - 40 bit di chiave sono pochini
  - RC4 era noto dal 1995 per avere qualche falla
  - il CRC32 non è un algoritmo di hashing sicuro
- Il criterio guida per gli ultimi 2 punti è stato l' ottimizzazione delle prestazioni

[2] se siete sufficientemente paranoici

Confidentiality Fail

- la cifratura RC4 si basa su due fasi:
  - La generazione di un flusso di byte pseudo-casuali dipendenti da una chiave (keystream)
  - Lo XOR tra il flusso casuale e il testo in chiaro
- Riuso dello stesso keystream $\Rightarrow$ recupero del keystream
- L' IV (24 bit) dovrebbe prevenire questo problema...
- Birthday paradox : in pratica un pacchetto ogni 5000 ha la stessa chiave

Integrity Fail

- CRC32 è un meccanismo di hashing con molte proprietà utili...
- Qualche volta sono troppe proprietà per essere veramente utili
- Costruire un messaggio(collisione) diverso da uno dato con lo stesso CRC32 è banale
- posso costruire messaggi "sensati" con lo stesso CRC32

Authentication Fail

- Nel 2001 Scott Fluhrer, Itsik Mantin, and Adi Shamir scoprono una vulnerabilità in RC4
- Dati molti testi cifrati con chiavi "simili" in parte , posso dedurre tutta la chiave
- Ma chi può essere così disgraziato da cifrare con moltissime chiavi simili, di cui una relazione interna è nota....
- Primo attacco : chiave recuperata con $\sim 100k - 1M$ pacchetti
- Ultimo sviluppo (27th Chaos Computer Congress) $\sim 9k$ pacchetti

Morale

- Usi impropri di un cifrario (sale troppo corto, chiave reale troppo corta)
- Errori nella scelta dei meccanismi di integrità regalano la possibilità di inviare pacchetti "sporchi"
- Falle nei cifrari utilizzati hanno risultati catastrofici (RC4)
- Siamo ancora fermi al 1999 nel cipher design...

Un po' di soluzioni...

- In risposta alla rottura completa di WEP, è stato progettato WPA
- RC4 è stato mantenuto per retrocompatibilità ...
- Ma il sistema di gestione della chiave è cambiato :)
- E' stato aggiunto un contatore per evitare i replay attack
- Il CRC-32 è rimasto, che male ci sarà mai ...

... quasi funzionanti :)

- 2008: Martin Beck and Erik Tews replicano l' attacco al sistema di integrità di WEP
- E' possibile decifrare una manciata di bytes del traffico in volo
- ... e quindi ricavare una manciata di bytes di chiavi validi :)
- A questo punto si possono iniettare un po' di *innocui* pacchetti di piccole dimensioni
- Come dei pacchetti ARP, per esempio :P

Three is the magic number

- Altra rottura, altra patch, nasce WPA2
- RC4 è stato sostituito da AES-256-CCMP
  - solido, non combina linearmente con il testo
- CRC32 è stato abbandonato a favore di HMAC
  - solo chi ha la chiave può calcolare il digest
- La chiave usata non è quella direttamente data dall' utente
  - tra il dire e il fare ci sono 4000 run di SHA-1 da far girare
  - non impedisce il bruteforcing, ma lo rallenta di molto

Hole in 196

- Quindi WPA2-AES-CCMP è perfettamente sicuro? Quasi...
- Lo standard 802.11 consente l' uso di una chiave di gruppo (GTK)
- La group key è nota a tutti i client connessi alla rete
- Chiunque la può usare per fingersi l' AP
- Chiamate gratis verso tutti :)
  - ARP Poisoning, Port scanning, Whatever^{TM}
- Chi è fuori dalla rete, continua a restare fuori e a non vedere nulla del traffico

Secure by design, insecure by application

- WPA2-AES è finalmente una soluzione ragionevolmente solida[3]
- Preconfiguriamo tutti gli AP in vendita per usare solo quello ed è fatta, no?
- No: le chiavi sono scelte dall' utente / amministratore della rete
- *Garantire la sicurezza è una gara tra gli ingegneri, che si sforzano di costruire sistemi a prova di idiota, e l'Universo, che cerca di produrre idioti migliori. Al momento sta vincendo l'Universo.*

[3] se vi fidate degli altri nella stessa rete

PEBKAC

- Spesso la scelta di una password da parte dell' utente ha risultati catastrofici
- Problem Exists Between Keyboard And Chair
- Esempi:
  - Una parola da dizionario $\Leftarrow \log_2(10^5) = 17$ bit
  - Una parola da dizionario + anno di nascita $\Leftarrow \log_2(10^5 * 120) = 24$ bit
  - Una parola da dizionario + camelcase $\Leftarrow\sim 30$ bit
  - Due parole da dizionario concatenate $\Leftarrow log_2(10^5)x2 = 34$ bit
  - Vecchio codice Poliself $\Leftarrow log_2(10^6) = 20$ bit

Defective by [hardware] design

- Di consueto, le password di default sono pessime
- Idea: perchè non generare delle default password forti e forzare l' utente a usarle?
- Alcuni produttori di modem/router WiFi enabled ci hanno pensato
- Problema: come generarle in modo sano?

Brilliant Ideas

- Serve ottenere un valore casuale e diverso per ogni oggetto
- Idee di rot13(Gryfrl) e rot13(Cveryyv):
  - Usiamo il numero di serie (non casuale)
  - Usiamo il MAC Address della scheda di rete
  - Usiamo un accrocchione a caso al posto di un algoritmo di hash
- Ovviamente, perchè pubblicare gli algoritmi di generazione delle chiavi? A chi interessano?
- Snfgjro [IT], Gryrpbz Vgnyvn [IT], Sevgm [DE]

Epic Fail.

- Il sistema precedente è sicuro[4], non fosse che:
  - Il numero di serie viene usato per comporre il nome dell ESSID pubblico della rete
  - Il MAC address può essere reperito ε-time ascoltando il traffico dell' access point
  - L' algoritmo-accrocchione è stato lasciato nel firmware del router durante la commercializzazione
- Ovvero, la password è sicura, nonchè sicuramente nota by default
- Accidentalmente, Snfgjro [IT], Gryrpbz Vgnyvn [IT], Sevgm [DE] hanno deciso di usare come password di default esattamente quelle originali.

[4] ROTFL

Password, 3rd edition style

- Come ottenere password forti[5]?
- E' sufficiente garantire che sia presa da un insieme con un minimo quantificato
- Scelta in modo casuale dall' insieme, p.e. con un dado
- Diceware password:
  1. Procurarsi un dizionario di almeno 100K parole
  2. Tirare 5d6 e comporre le cifre in un solo numero n
  3. Lookup nel dizionario alla n-esima parola
  4. Fino a quando non avete raccolto abbastanza parole, GOTO 1
- $log_2(6^5) = 12,9$ bit garantiti per parola
- Es. "PrematurataPerDueComeFosseVicesindaco" = 77.4 bit

[5] almeno quanto una buona palla di fuoco

Final points

- Uso della crittografia != Automagical Security
- Se non ho un segreto noto solo alle persone che parlano, è solo scrambling di dati, non crittografia
- Usare algoritmi noti e standard aiuta[6]

[6] sperando che gli standard siano solidi :P