Wireless Network Esercitazioni

Alessandro Villani
avillani@science.unitn.it

Wireless Router WRT54G

LINKSYS

WRT54G

- È un router Wireless:
  - 1 porta ADSL
  - 4 porte ethernet
  - 802.11b e 802.11g
- La particolarità è che esegue un firmware "linux"
- La Linksys ha rilasciato i sorgenti del firmware

WRT54G: Aggiornamento Firmware

- Configurazione via WEB (login vuota e password admin)
- Assegnato un IP al router (l'ip di default è 192.168.1.1) si aggiorna il firmware via WEB
- A questo punto si possono abilitare le connessioni ssh
- La login è root, la password (di default) è admin

WRT54G: Aggiornamento Firmware

- Attualmente ci sono molti progetti per estendere le funzionalità di questo router
- Uno dei firmware più interessanti è scaricabile all'indirizzo:
  ftp://ftp.sveasoft.com/pub
- Il file attuale è:
  Firmware_Samadhi2_v2_2.00.8.6sv.bin

WRT54G: Comandi

- La documentazione è reperibile all'indirizzo:
  http://docs.sveasoft.com/
- Si possono eseguire molti comandi unix:
  - ls
  - cd
  - ifconfig
  - cat
  - ps

WRT54G: Filesystem

- La struttura del filesystem è quella di un sistema linux:
  - /etc
  - /bin
  - /sbin
  - /proc
- Ad esempio:
  cat /proc/cpuinfo
  cat /proc/net/wireless

WRT54G: Comandi Linksys

- wl è il comando generico per la gestione del router
  - wl ver → versione del sistema
  - wl radio → stato dell'802.11
  - wl radio on → attiva 802.11
  - wl radio off → spenge 802.11
  - wl chanlist → lista dei canali validi
  - wl channels_in_countr IT b → canali validi in Italia per 802.11b

WRT54G: Possibili Applicazioni

- Installare regole di instradamento, firewalling, traffic shaping direttamente sull'AP
- Installare un end-point VPN → non c'è più bisogno di WEP!
- Installare un captive portal direttamente sull'AP → il firmware sviluppato da PortLess Network implementa questa feature (http://www.portless.net/ewrt/index.html)

WEP Cracking

WEP: Wired Equivalent Privacy

- WEP si basa sull'algoritmo RC4 della RSA
- È un sistema di crittazione basato su una chiave condivisa
- La chiave condivisa è lunga 40 bit
- È concatenata a un vettore di inizializzazione (IV) lungo 24 bit
- Si ottiene cosi un seed di 64 bit per l'RC4

WEP: Wired Equivalent Privacy

- Molte schede generano i 24 bit dell'IV utilizzando un counter od un generatore di numeri pseudocasuali
- Alcune schede azzerano l'IV ogni volte che sono inizializzate e poi incrementano il counter di 1 → aumentano la probabilità che la chiave sia riusata (i valori bassi di IV sono più probabili)

WEP: Wired Equivalent Privacy

- Per inviare un pacchetto di dati:
  - Dato il paylod M, viene calcolato il CRC di 32 bit c(M) che viene concatenato ad M → M·c(M)
  - La chiave k è concatenata all'IV determianto per il pacchetto → IV·K
  - L'algoritmo RC4 è inizializzato usando questo pacchetto e viene generata una sequenza di bytes → RC4(IV·k)
  - M·c(M) a questo punto e messo in xor con RC4(IV·k) → C = (M·c(M))⊕RC4(IV·k)
  - I 3 byte dell'IV sono trasmessi in chiaro (insieme con l'indice della chiave WEP)

WEP: Wired Equivalent Privacy

- Il ricevente concatena l'IV ricevuto con la chiave WEP condivisa
- Decritta il payload e se il CRC coincide allora il pacchetto è valido

WEP: RC4

- Key Scheduling Algorithm
- RC4 utilizza un vettore di stato di 256 ottetti S[256] e due contatori i, j
- Inizializzazione dello stato:
  - S[n] = n, i=0, j=0
  - Il vettore temporaneo T di 256 ottetti si inserisce la chiave K ripetendola se corta
  - Si scorre S scambiando gli elementi del vettore

```
for i = 0 to 255
    j = (j + S[i] + T[j]) mod 256
    scambia (S[i], S[j])
```

WEP: RC4

- Pseudo Random Generation Alghoritm. Generazione del keystream:
  - Per generare un ottetto z del keystream dallo stato corrente (S, i, j):
    i = (i + 1) mod 256
    j = (j + S[i]) mod 256
    scambia (S[i], S[j])
    t = (S[i] + S[j]) mod 256
    z = S[t]
  - Inizialmente i=0, j=0 e si scarta T.
  - Il processo di generazione continuerà finche non ci sono più dati

WEP: Riutilizzo della codifica

- Se utilizziamo lo stesso IV, viene generata la stessa sequenza (keystream) di byte da RC4
- Crittando così due messaggi p1 e p2 abbiamo:
  - $C1 = P1 \oplus RC4(IV \cdot k)$
  - $C2 = P2 \oplus RC4(IV \cdot k)$
  - $C1 \oplus C2 = P1 \oplus RC4(IV \cdot k) \oplus P2 \oplus RC4(IV \cdot k)$
    $= P1 \oplus P2$
- Quindi con l'xor di due messaggi cifrati si ottiene l'xor dei due messaggi in chiaro

WEP: Riutilizzo della codifica

- Se si conosce uno dei due messaggi si ottiene l'altro
- Se si hanno molti messaggi codificati con lo stesso keystream è facile risalire ai messaggi originali
- I protocolli impongono molte similarità sui pacchetti!
- Non si devono riusare i keystream!

WEP: Attacchi di forza bruta

- Bastano due pacchetti in generale (per essere sicuri che il CRC non coincida per caso anche con una chiave WEP sbagliata)
- Può utilizzare una lista di chiavi “facili”
- Analizzando l’intero spazio di ricerca dato dai 40 bit, ci possono volere circa 45 giorni → Non pratico per chiavi a 104 bit

WEP: Attacchi basati su Weak IV

- S. Fluhrer, I. Mantin, A. Shamir hanno dimostrato che esistono delle debolezze nell'algoritmo di generazione delle chiavi in RC4 → "*Weakness in the Key Scheduling Algorithm of RC4"*
- L'attacco descritto nel loro articolo, oltre ad essere estremamente veloce, richiede un tempo che cresce linearmente con la lunghezza della chiave WEP!

WEP: Attacchi basati su Weak IV

- Il fatto che un larga parte della chiave (3 byte) sia trasmessa in chiaro aumenta la facilità di cracking:
  - Le prime tre iterazioni del KSA sono facilmente deducibili per il fatto che le prime tre cifre della chiave sono note!
- Si può vedere che c'è una probabilità del 5% che i valori in S[0] - S[3] non cambino dopo le prime 3 iterazioni del KSA

WEP: Attacchi basati su Weak IV

- È stato dimostrato che un IV di un certo tipo sono soggetti ad essere crackati:

  (B+3, 255, x)

  dove B è il byte della chiave segreta che stiamo crackando
- Quindi per ogni byte della chiave ci sono 256 Weak IV

WEP: Attacchi basati su Weak IV

- I primi valori dei dati crittati è l'header SNAP (*Sub Network Attachment Point*). È uno standard (di livello 2) per la trasmissione di datagram IP su reti IEEE 802
- L'header non crittato è AA in esadecimale
- Xor dei primi dati crittati con AA ci da il primo byte del PRGA
- Questa informazione ci può consentire di ricostruire la prima cifra della chiave WEP se ho un Weak IV del tipo (3, 255, x)

WEP: Attacchi basati su Weak IV

- Esistono anche altre famiglie di Weak IV
- Oltre il primo byte della chiave l'operazione si complica perché richiede di ciclare sul PRGA per più passi e quindi potremmo non essere più in grado di dedurre con una ragionevole probabilità le permutazioni di S

Airsnort

- Esistono vari tools che consentono di determinare in modo automatico una chiave WEP
- Uno di questi è Airsnort, scaricabile all'indirizzo:
  http://airsnort.shmoo.com/
- È un programma linux
- Richiede che la scheda wireless sia in modalità monitor
- Funziona ad esempio con le schede Prism2, Orinoco e Cisco

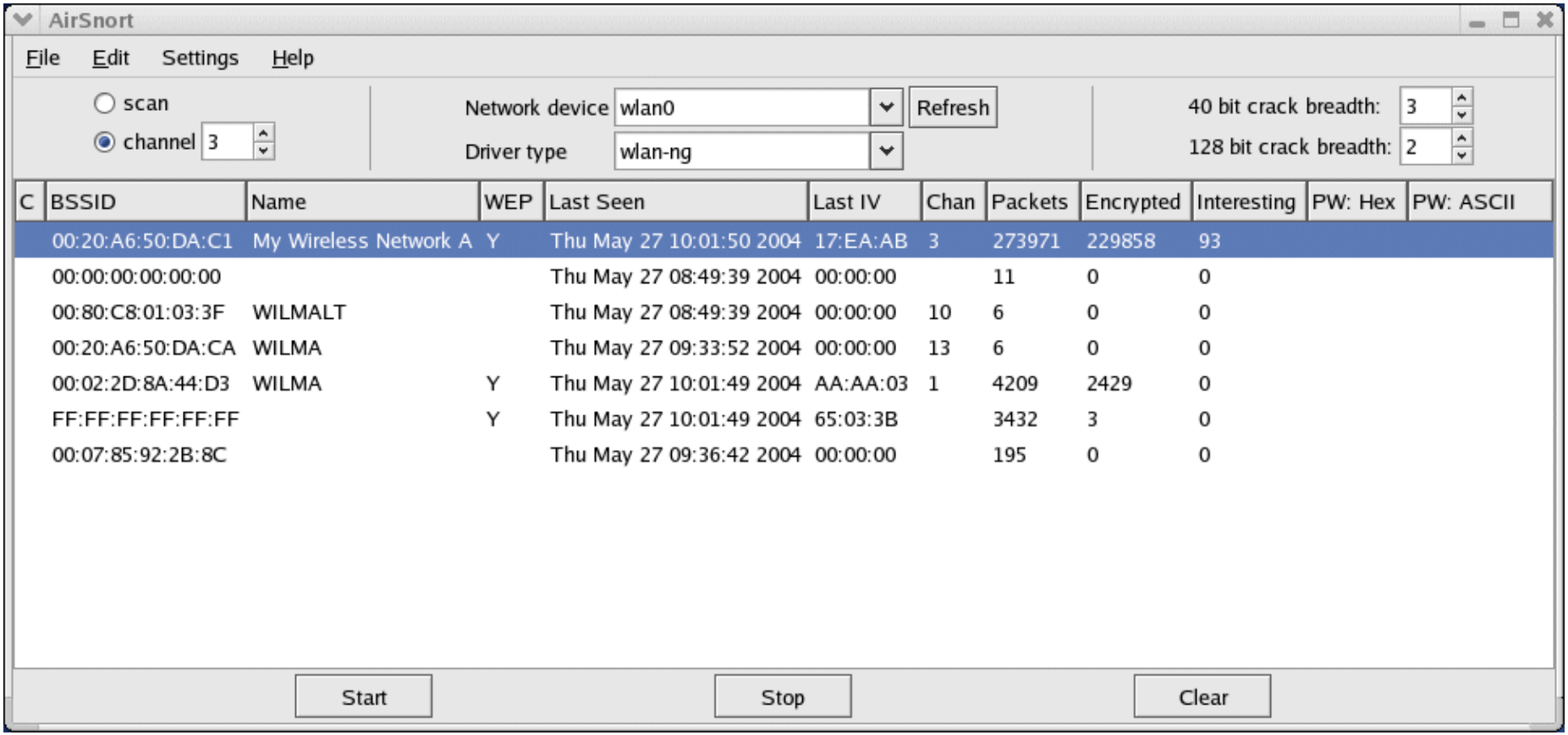
Airsnort

- Una volta attivato, il programma cattura i pacchetti ed in contemporanea cerca di crackare la chiave WEP:
  - Tutti i pacchetti non data (eccetto i beacon) sono scartati
  - I pacchetti non crittati sono scartati
  - I pacchetti crittati sono selezionati e quelli ritenuti non interessanti sono scartati
- I pacchetti ritenuti interessanti sono i Weak IV individuati da Fluhrer, Mantin e Shamir (più Weak IV individuati successivamente)

Airsnort

- Ogni 10 weak IV acquisiti, airsnort utilizza un attacco probabilistico
- Si può controllare quanto profondamente analizzare l'albero delle diverse possibilità
- Un valore *n* del parametro "breadth" indica che verranno provate gli n valori più probabili per ciascuna posizione della chiave
- Sono richiesti circa 1500 weak IV per una chiave a 64 bit e circa 3000 per una chiave a 128 bit

Airsnort


AirSnort
File
Edit
Settings
Help
scan
channel 3
Network device wlan0
Refresh
Driver type wlan-ng
40 bit crack breadth: 3
128 bit crack breadth: 2
C | BSSID | Name | WEP | Last Seen | Last IV | Chan | Packets | Encrypted | Interesting | PW: Hex | PW: ASCII
00:20:A6:50:DA:C1 | My Wireless Network A | Y | Thu May 27 10:01:50 2004 | 17:EA:AB | 3 | 273971 | 229858 | 93
00:00:00:00:00:00 | | | Thu May 27 08:49:39 2004 | 00:00:00 | | 11 | 0 | 0
00:80:C8:01:03:3F | WILMALT | | Thu May 27 08:49:39 2004 | 00:00:00 | 10 | 6 | 0 | 0
00:20:A6:50:DA:CA | WILMA | | Thu May 27 09:33:52 2004 | 00:00:00 | 13 | 6 | 0 | 0
00:02:2D:8A:44:D3 | WILMA | Y | Thu May 27 10:01:49 2004 | AA:AA:03 | 1 | 4209 | 2429 | 0
FF:FF:FF:FF:FF:FF | | Y | Thu May 27 10:01:49 2004 | 65:03:3B | | 3432 | 3 | 0
00:07:85:92:2B:8C | | | Thu May 27 09:36:42 2004 | 00:00:00 | | 195 | 0 | 0
Start
Stop
Clear

Airsnort

- Test di attacco effettuato utilizzando:
  - L'Access Point dell'Avaya
  - Due laptop per generare traffico
  - Un laptop con una scheda Netgear ed Airsnort
- Impostata una chiave WEP a 64 bit, ovvero 40 bit di chiave, ovvero 5 caratteri → .SL04
- Dopo circa 2.5 ore di acquisizione, circa 1.300.000 pacchetti acquisiti (di cui 1.200.000 criptati) e 1127 Weak IV, la chiave è stata determinata!