

Defenestration #1

BlueSecurity: quando il dente fa male!



(in)sicurezze del protocollo Bluetooth

Alfredo Morresi (aka Legolas)
<http://www.rainbowbreeze.it>
26/11/2006 - Monteurano



Bluetooth – qualche pettegolezzo

- C'era un volta un re vichingo danese, Harald Blatand (nel X sec ebbe l'idea di far comunicare, militarmente e commercialmente, la Danimarca con la Norvegia)...
- Nel 1994 Ericsson studia una soluzione per eliminare cavetteria varia nella connessione tra telefonino e auricolare o computer
- Nel 1998/09 si forma il SIG, (Bluetooth Special Interest Group - Nokia, IBM, Intel, Toshiba, Ericsson) allo scopo di trovare una tecnologia wireless aperta, universale, a basso costo ed accessibile a device di piccole dimensioni per trasmissioni dati e vocale
- Nel 1999/08 il SIG rilascia le specifiche per il Bluetooth 1.0
- Oggi il SIG conta più di 2000 membri (Microsoft, Mororola, Apple, Lucent, 3COM ecc) e siamo al Bluetooth 2.0, retrocompatibile con la 1.0 e la 1.2



Bluetooth – come funziona

- Sistema radio basato su un'architettura hardware e una pila protocollare software (analoga al modello OSI/ISO)
- Opera alla frequenza del 2.45Ghz (la stessa dell'amico 802.11)
- Uso del FHSS (Frequency Hopping Spread Spectrum) per evitare interferenze con altri apparecchi (salta 1600 volte/sec tra i 79 canali disponibili, ad intervalli di 1Mhz da 2.402 a 2.480 Ghz)
- Diverse potenze di emissione: classe 1 (100 mW di potenza, 20dBm di uscita, fino a 100 metri di copertura), classe 2 (2,5 mW, 4dBm, 10 metri), classe 3 (1 mW, 0dBm, 0,5 metri)
- Bassa velocità di trasferimento (da meno di 1 Mbps per la 1.0 ai 2-3 Mbps per la 2.0)
- Ridotto consumo energetico (minore di versione in versione)

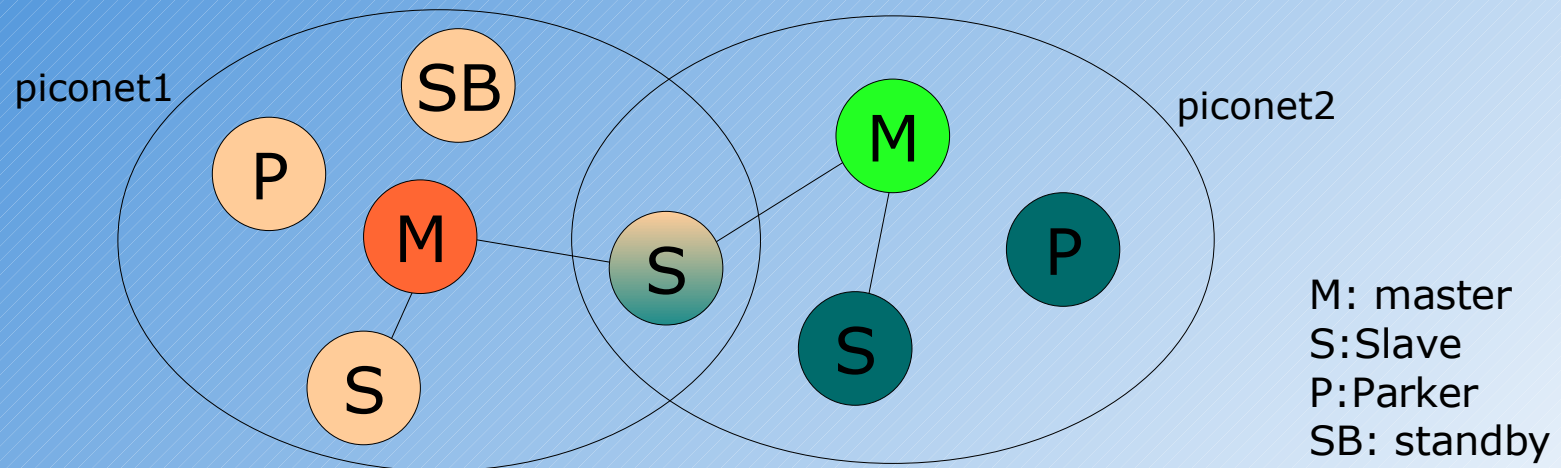


Bluetooth – come funziona

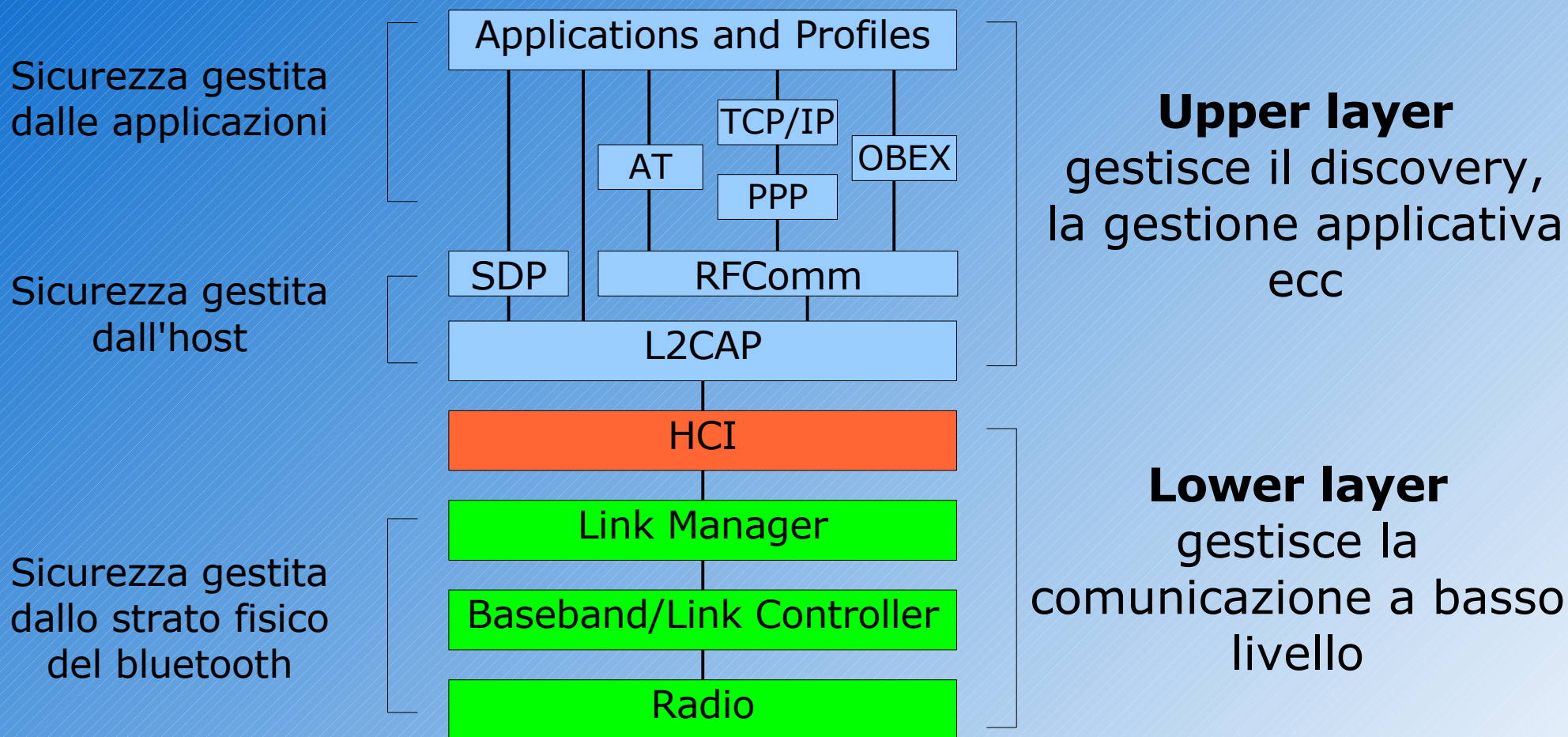
Una serie di device collegati assieme formano una PICONET

- per ogni piconet c'è un device che fa da master e gli altri sono tutti slave (attivi, parcheggiati o in standby)
- il master stabilisce la sequenza di hopping
- ci possono essere fino a 200 slave ma solo 7 possono essere attivi contemporaneamente

Più piconet collegate tra loro formano una SCATTERNET

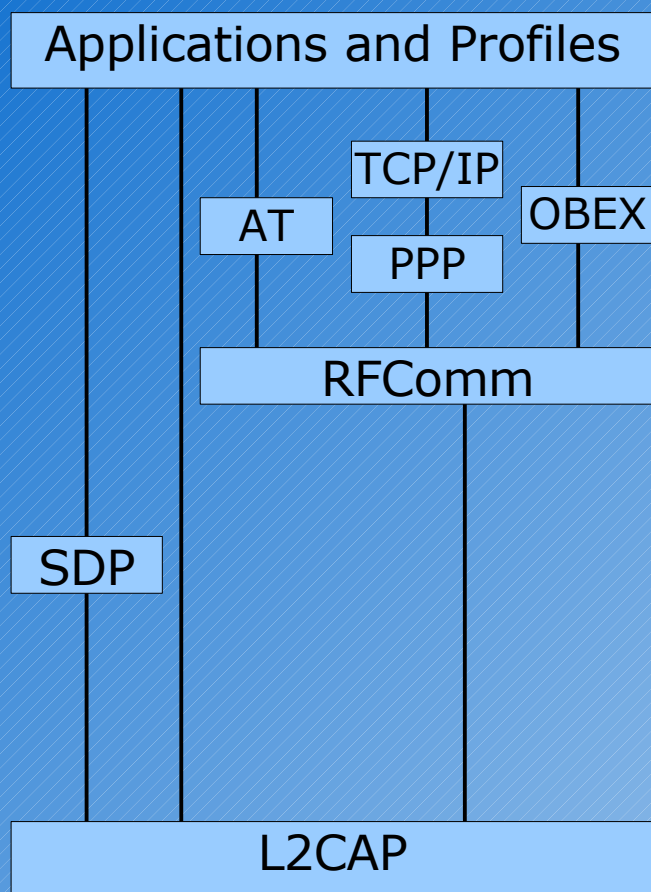


Bluetooth – lo stack



Bluetooth – lo stack

Upper layer



Object EXchange
Point 2 Point Protocol
Comandi AT

Emulazione di connessioni seriali RS-232

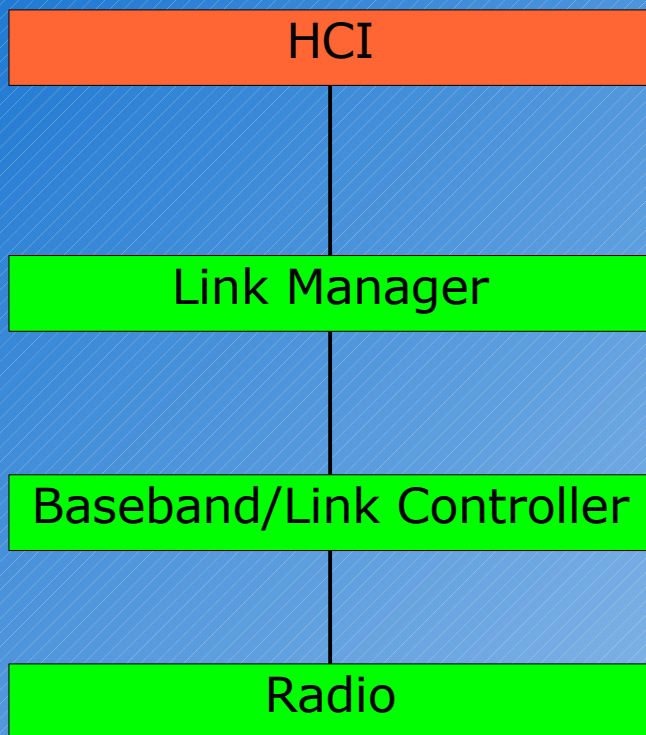
Service Discovery Protocol: pubblica i servizi offerti dal device e ricerca quelli presenti sugli altri device con cui si vuole comunicare. Per servizi si intendono singole funzionalità del dispositivo (Dial-up Networking, Fax, Serial Port, OBEX Object Push, IrMC Sync, ecc)

Logical Link Control and Adaptation Protocol: si occupa di incapsulare i pacchetti e fornire un meccanismo di astrazione simile a quello delle porte del protocollo TCP/IP; quando un device comunica con un altro utilizza una determinata porta, che può essere vista come un canale logico



Bluetooth – lo stack

Lower layer



Host Controller Interface: fornisce un'astrazione delle funzionalità di basso livello del sistema radio sottostante

Gestione dei pacchetti per i differenti link, gestione del canale SCO (Synchronous Connection Oriented – per comunicazione voce) e di quello ACL (Asynchronous ConnectionLess – per comunicazione dati)

Mantiene la sincronizzazione tra i link, gestione dei dati in tx/rx

Responsabile della modulazione/demodulazione RF



Bluetooth – gestione della sicurezza

L'applicazione della sicurezza si può avere su 2 livelli: per i servizi (scambio di un file, scambio di una vCard), per i collegamenti, cioè prima di stabilire un canale di comunicazione dove poi far passare i servizi (il classico pairing tra device)

Tre livelli di sicurezza predefiniti

- **Mode 1 – No security:** siamo tutti amiconi e ci fidiamo l'uno dell'altro, quindi niente sicurezza
- **Mode 2 – Application/Service based (L2CAP):** la sicurezza è attivata a livello di servizio, ma chiunque può connettersi con me (policy flessibili a secondo dei servizi)
- **Mode 3 – Link (device) level security:** la sicurezza è attivata a livello di collegamento fisico, comunicano solo dispositivi “trusted”



Bluetooth – gestione della sicurezza

A livello di sicurezza per il collegamento, ogni device può essere

- “untrusted device”: hanno accesso solo ai servizi che non richiedono autenticazione
- “trusted device”: hanno accesso anche ai servizi che richiedono l'autenticazione

A livello di servizi invece, ogni servizio può richiedere

- autorizzazione e autenticazione: la richiesta deve essere autorizzata manualmente e può provenire solo da un trusted device
- autenticazione: basta che sia un trusted device
- niente: tutti i device possono accedervi



Bluetooth – il pairing tra dispositivi

Grazie al pairing, due device passano dallo stato untrusted allo stato trusted

Vengono coinvolti nella procedura di pairing il BT_ADDR (indirizzo fisico del device, 48bit, univoco, paragonabile al MAC address dei dispositivi di rete), una chiave di cifratura e una chiave di collegamento dei numeri pseudocasuali e degli algoritmi per la generazione delle chiavi (E0, E21, E22 ecc)

Per una descrizione dettagliata della procedura di pairing consultare <http://www.niksula.hut.fi/%7Ejiiitv/bluesec.html>

Il pin che viene richiesto in questa fase diventa il segreto condiviso con cui poi vengono generate le successive chiavi e che assicura la riservatezza della comunicazione



Tecniche di attacco

Premessa

il **Bluetooth**, inteso come standard, è **sicuro**
I problemi sono a livello applicativo e di implementazione

Tre tipologie di attacchi

- Attacchi contro il protocollo (lower layer)
Attacco a E22, Spoofing, BlueDump, DOS
- Attacchi contro le funzionalità di discovery (upper layer)
Identificazione dei dispositivi, abuso delle info di discovery: Bluejacking, Discovery Mode Abuse, Blueprinting
- Attacchi contro i servizi offerti dai dispositivi (upper layer)
Abuso dei servizi, prelievo delle informazioni, controllo completo: BlueSnarf, HELOMoto, Bluebug, BlueBump, BlueSmack, CarWhisperer



Attacco liv protocollo: brute force del PIN

Ci sono dispositivi che hanno un PIN preimpostato come auricolari, antenne GPS, kit vivavoce, e in genere questo PIN è di 4-5 caratteri numerici. Basta attuare un brute force e in poco tempo si ottiene il pairing con il device (spesso poi sono 0000, 1234, 1111, 9999 o si trovano su Internet a secondo del prodotto)



Attacco liv SDP: Discovery mode abuse

Il bluetooth ha tre stadi: acceso, spento, invisibile

Quando un device è acceso risponde alle richieste di inquiry inviate in broadcast da altri device, comunicando la sua presenza (C'è nessuno? Sì, ci sono io che mi chiamo ...). Quando invece è invisibile, non risponde, ma rimane comunque attivo

Ogni device è identificato dal BT_ADDR

48 bit suddivisi in 6 ottetti (00:A0:12:43:1F:52), dove i primi 3 sono fissi e dipendenti dal costruttore, gli altri 3 sono variabili a secondo del dispositivo. Praticamente il MAC address del chip bluetooth

Posso, interrogando in maniera diretta un BT_ADDR, vedere se c'è un device attivo che mi risponde a questa richiesta

Posso fare lo scanning di un certo range di BT_ADDR per conoscere i device attivi in quel range. Con più dongle bt a disposizione, posso anche lavorare in parallelo, assegno ad ognuno un intervallo del range e controllare in minuti invece che in ore

Software utili: RegFang, BTScanner, Bluesniff



Attacco liv upper: Bluesmack



BlueSmack è un tipico attacco DOS (Denial of Service) che permette di far diventare instabile un sistema operativo sino a fargli generare delle eccezioni critiche (rivisitazione del classico Ping of Death che affligge Windows 95 in ambiente Bluetooth)

Si incrementa oltre misura la dimensione di un pacchetto echo request (L2CAP ping) che verrà poi spedito al device vittima

Alcuni device oltre ad un certa dimensione del pacchetto (Compaq IPAQ con pacchetti più grandi di 600 byte ad esempio), ricevono il dato ma generano degli errori che fanno bloccare completamente il sistema operativo



Attacco liv SDP: Bluejacking (buono)

Per semplificare il pairing tra device, viene scambiato l'identificativo del device, una stringa più lunga di 248 caratteri

Il bluejacking consiste nell'usare questa stringa per comunicare, senza voler realmente associare i device, tipo un device che si chiama "Piacere, sono quello con la maglietta blu e i capelli corti, mi chiamo Marco :)"

Altra forma di bluejacking è quella di inviare una vCard con, al posto del nome del contatto, un messaggio. Moltissimi device accettano le vCard senza chiedere conferma, mostrano sul display nome e cognome del contatto ricevuto (in realtà il nostro messaggio) e chiedono di aggiungerlo alla rubrica

Molto in voga, soprattutto tra i ragazzi, per divertirsi un po' in posti affollati (metro, sale d'attesa o altro)



Attacco liv SDP: Bluejacking (cattivo)

C'è sempre un "dark side of the moon"!

Mettiamo che la stringa identificativa del mio device sia:
"Promozione TIM: digita 1234 e, connettendoti alla rete, riceverai
5 euro di ricarica omaggio dopo il messaggio di conferma"

A questo punto provo ad associarmi ad un device, iniziando la
procedura di pairing. Nel device vittima apparirà la richiesta di un
pin per connettersi con "Promozione TIM: digita 1234 ecc ecc"

Se qualcuno cade nel tranello, digiterà 1234, lo ridigito anche io sul
mio device, completando la procedura di pairing e ottenendo
accesso trusted al device della vittima, potendo accedere senza
ulteriori controlli a molti servizi

Tecnica di social engineering, come il phishing per i siti web. Non
sfrutta una reale vulnerabilità del protocollo



Attacco liv servizio: RFCOMM

RFCOMM è un protocollo in grado di emulare le comuni porte seriali dei computer RS-232 via bluetooth. Pensato per evitare di riscrivere molte applicazioni esistenti in quanto già funzionanti con comunicazione seriale. Dotato di SDP (Service Discovery Protocol) che permette di richiedere i servizi attivi disponibili sul device

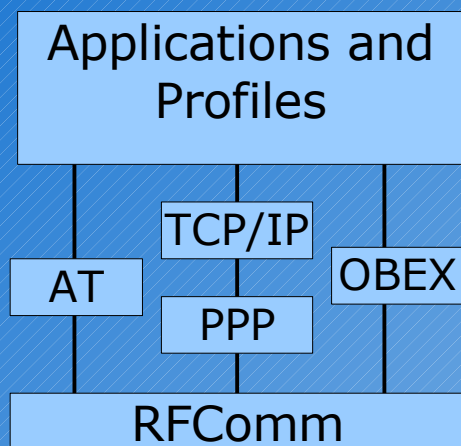
Il problema maggiore è che nelle applicazioni su protocollo seriale non c'era bisogno di implementare delle funzioni di sicurezza in quanto per utilizzarle si presupponeva il collegamento fisico con un cavo seriale (non facile da nascondere per un eventuale aggressore)

Ovviamente con il bluetooth questa garanzia di sicurezza fisica viene a cadere: basta collegarsi al device, tramite SDP chiedere quali servizi sono disponibili, e se non ci sono protezioni particolari stabilite dal produttore (e quindi variabili da device a device), si possono iniziare le danze



Attacco livello servizio: OBEX

Il protocollo OBEX (OBject EXchange) viene usato per lo scambio di dati (business card, file e altro) ed è un servizio invocabile per tramite di RFCOMM (presente anche sull'IrDa)



Esiste l'OBEX Push Profile per inviare dati al device e l'OBEX Pull Profile per prendere dei dati dal dispositivo, specificando, nel caso di un file, posizione e nome del file da prelevare

In genere, per rendere più fruibile l'OBEX Push, non è prevista l'autorizzazione (scambio di Business Card ad esempio), mentre ci vuole per l'OBEX Pull



Attacco liv servizio: BlueSnarf



In alcuni device, l'OBEX Pull non è protetto da autenticazione (se lo saranno scordato)

In alcuni device, informazioni specifiche vengono salvate in file di testo predefiniti, come ad esempio la rubrica (telecom/pb.vcf) o il calendario (telecom/calc.vcs)

Quando si verificano entrambe le precedenti condizioni su uno stesso device, posso tentare con successo un attacco di tipo BlueSnarf e prelevare dati *molto* sensibili dal device vittima senza che questi se ne accorga minimamente

Non pochi cellulari vulnerabili (Nokia 6310, 6310i, 8910, 8910i; Ericsson R520m, T39m, T68, Sony Ericsson T68i, T610, Z1010)



Attacco liv servizio: BlueSnarf++



Derivato dal BlueSnarf, questo attacco è ancora più aggressivo: ottiene infatti l'accesso in lettura/scrittura dell'intero file system del device

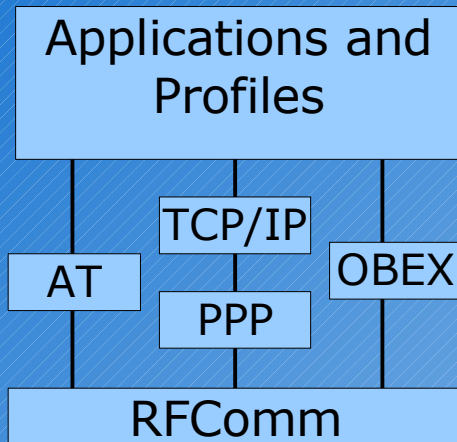
In alcuni device, invece di implementare un semplice OBEX Push daemon, è stato realizzato un vero OBEX ftp server a cui ci si può connettere con la stessa security dell'OBEX Push, ovvero nessuna!

Tramite comandi come ls, get, put e rm si puo' fare il bello e il cattivo tempo nel filesystem device vittima



Attacco livello servizio: AT

I comandi AT sono un set di comandi, utilizzate inizialmente nei modem, per impartire diverse istruzioni (dalla composizione di un numero di telefono, ai settaggi di alcuni profili di chiamata ecc). Quando usiamo un modem, lo facciamo per mezzo dei comandi AT



Anche nei cellulari sono stati integrati i comandi AT (da qui la possibilità di usare il cell come modem esterno) ed estesi con nuovi set di istruzioni per la gestione delle caratteristiche peculiari del device (gestione della rubrica, degli sms, delle configurazioni ecc)

I produttori di cellulari rilasciano dei documenti con tutti i comandi AT implementati sui propri device, per favorire il lavoro degli sviluppatori



Attacco liv servizio: BlueBug



Su alcuni device, per i soliti errori di implementazione dello stack bluetooth, quando viene fatta una richiesta dei servizi RFCOMM disponibili non viene mostrato il servizio AT, però è ugualmente accessibile, e sempre senza nessun pairing o altro controllo di sicurezza (una pessima security thru obscurity)

Con la guida dei comandi AT del device sottomano, a questo punto ho PIENO controllo (posso leggere-ricevere-spedire sms, leggere-modificare-cancellare elementi della rubrica, impostare deviazioni di chiamate, ottenere il numero della sim vittima, ottenere l'IMEI per clonare la sim vittima, connettermi ad Internet sfruttando la connessione dal device vittima, poi usando un po' di fantasia chissà)

Non pochi cellulari vulnerabili (Nokia 6310, 6310i, 8910, 8910i; Sony Ericsson T68i, T610, Z1010)



Attacco liv servizio: HeloMoto

Alcuni Motorola (V-Series) hanno un'incorretta implementazione della gestione dei device trusted

Si inizia con il device attaccante che tenta una connessione OBEX Push (ad esempio mandando una vCard) per la quale non viene richiesta nessuna autenticazione

Viene interrotto il trasferimento e come per magia il device attaccante ora risulta tra i trusted device di quello vittima (Merlino a questi gli faceva un baffo!)

Posso quindi connettermi ad un servizio headset (auricolare, in genere il servizio RFCOMM 7), e da lì usare tutto il set di comandi AT del device

Device vulnerabili: Motorola v80, v5xxx, v6xxx, E398 (e poi mi chiedono perchè li chiamo MotoSola)



Attacco liv servizio: Car Whisperer

Quante macchine ci sono oggi Bluetooth-enabled? E' interessante sapere che spesso i PIN per fare il pairing sono fissi e dipendono dal produttore dell'auto

Come tutti i chip bluetooth, anche quelli di queste macchine hanno i primi 3 ottetti del BT_ADDR impostati sul codice del produttore

Con queste informazioni e avendo a disposizione una macchina "blu", posso effettuare con successo un pairing e connettermi al servizio RFCOMM sul canale 1 e aprire una connessione di tipo SCO (per traffico voce) con un auricolare in mio possesso. A questo punto posso sentire/parlare nella macchina vittima



Bluetooth - Gli optional a corredo



Il massimo d'azione dei dispositivi bluetooth è di 100 metri per quelli di classe 1

Nel 2004 Il gruppo Trifinite hanno modificato un dongle bluetooth di classe 1 sostituendo l'antenna omnidirezionale del dongle con una direzionale e hanno eseguito un attacco di BlueSnarfing e BlueBug su un Nokia 6310i a 1,78 km di distanza.

Info su http://trifinite.org/trifinite_stuff_bluetooone.html



Bluetooth - Gli optional a corredo



La risposta ottenuta ad una richiesta SDP di inquiry per i servizi disponibili via RFCOMM si compone di 24 bit: 11 bit di Service Class, 5 bit di Major device class e 6 bit di Minor device class.

I significati dei valori sono standard e si trovano su <https://www.bluetooth.org/foundry/assignnumb/document/baseband>

Ed esempio 0x140680: indica 'Rendering and Object Transfer' con una MajorClass 'Imaging' e MinorClass 'Printer'.

Su <http://www.betaversion.net/btdsd/> c'è un database con la lista di molti device, con i servizi a disposizione e le eventuali vulnerabilità riscontrate, con chiave BT_ADDR (univoco per ogni chip bluetooth)



Bluetooth sw: Linux



I mattoncini di base

- **BlueZ**

implementazione dello stack bluetooth sotto GNU/Linux

hcitool – sdptool - rfcmm

<http://www.bluez.org/>

- **OpenObex**

client per le operazione con il servizio OBEX



Bluetooth sw: Linux

```
Shell to hell
-----
[MAIN MENU]

[1] Scan
[2] Scan and attack
[3] Scan and attack (endless loop)
[4] Info Menu
[5] Action Menu
[6] Change preferences
[7] Show preferences
[8] Show logfile
[9] Exit
-----

>>> 5

-----
[ACTION MENU]

[1] Choose a target
[2] Blue Snarf
[3] Blue Snarf++
[4] Blue Snarf Ericsson
[5] Blue Bug
[6] Blue Bug AT shell
[7] Helo Moto
[8] Blue Smack
[9] Stop Blue Smack
[10] Nasty VCard
[11] Symbian Remote Restart
[12] Try all attacks
[13] Automatic attack
[14] Launch redfang
[15] Change your bluetooth address
[16] Send soundfile
[17] Scan RFCOMM channels
[18] Launch RFCOMM shell
[19] RFCOMM Connection
[20] HCI Connection
[21] Request new link key
[22] Info menu
```

Bluediving

effettua il discovery e molte tipologie di attacco su device bluetooth (tutti quelle fino a qui descritti tranne il CarWhisperer)

<http://bluediving.sourceforge.net/>



Bluetooth sw: Linux

```
uxterm
Base Address: 00:02:72:B1:D5:DA
RSSI: +0 Link q: 000
Profile Descriptor List:
  "OBEX File Transfer" (0x1106)
  Version: 0x0100

Service Name:      Bluetooth Serial Port
Service Description: Bluetooth Serial Port
Service Provider:  Symbian Ltd.
Service RecHandle: 0x10003
Service Class ID List:
  "Serial Port" (0x1101)
Protocol Descriptor List:
  "L2CAP" (0x0100)
  "RFCOMM" (0x0003)
  Channel: 2
Language Base Attr List:
  code_ISO639: 0x656e
  encoding: 0x6a
  base_offset: 0x100

Service Name:      Handsfree Audio Gateway
Service RecHandle: 0x10004
Service Class ID List:
  "" (0x111f)
  "Generic Audio" (0x1203)
Protocol Descriptor List:
  "L2CAP" (0x0100)
  "RFCOMM" (0x0003)
  Channel: 3
Language Base Attr List:
  code_ISO639: 0x656e
  encoding: 0x6a
  base_offset: 0x100
Profile Descriptor List:
  "" (0x111e)
  Version: 0x0100
```

```
uxterm
Base Address: 00:02:72:B1:D5:DA
LS Address      C1k Off  Class  Name
1 00:60:57:D9:30:2D 0x3aed 0x500204 Nokia 6600
0 00:60:57:67:79:36 0x2cbf 0x520204 Nokia 6310i
0 00:0A:D9:F4:67:59 0x3b4e 0x520204 T610
0 00:02:EE:57:35:70 0x2449 0x502204 Nokia7650
0 00:80:37:16:A5:75 0x1d02 0x200404 n/a
0 00:0E:07:26:D8:93 0x1f99 0x520204 n/a
0 00:60:57:B0:4C:DC 0x1dd9 0x500204 Nokia N-Gage
1 00:60:57:BC:C1:EE 0x6dc1 0x500204 n/a
1 00:0A:D9:E3:B2:F9 0x65be 0x520204 n/a
0 00:02:EE:0D:A6:FB 0x3f35 0x502204 n/a

Devices found: 10
```

BTScanner

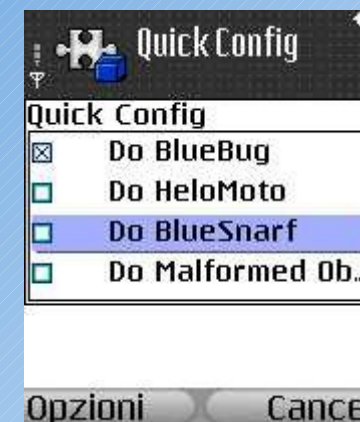
simile a kismet, ma per le reti bluetooth

ottiene informazioni sui servizi RFCOMM

ha un device con molti codici produttore (per i BT_ADDR)



Bluetooth sw: Cellulari



Bloover II

Realizzato dalla crew di trifinite, funziona su cellulari con MIDP 2.0 e JSR-82 (Bluetooth API) (praticamente molti Nokia e SonyEriccson)

Realizza attacchi di tipo BlueBug, HeloMoto, BlueSnarf e Malformed Object

http://trifinite.org/trifinite_stuff_blooverii.html



Bluetooth sw: Cellulari



EasyJack 2

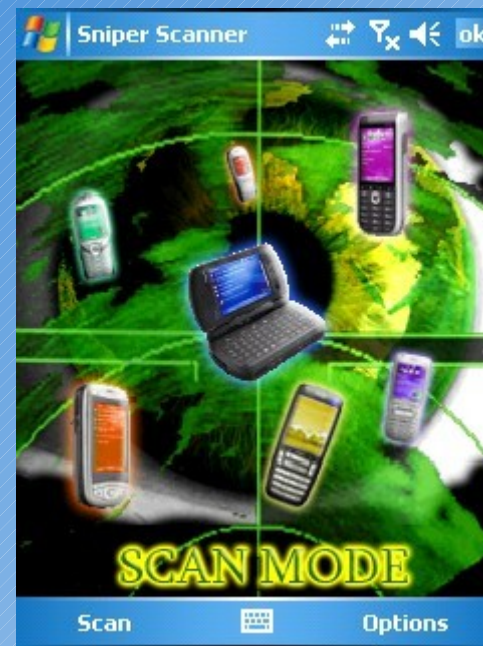
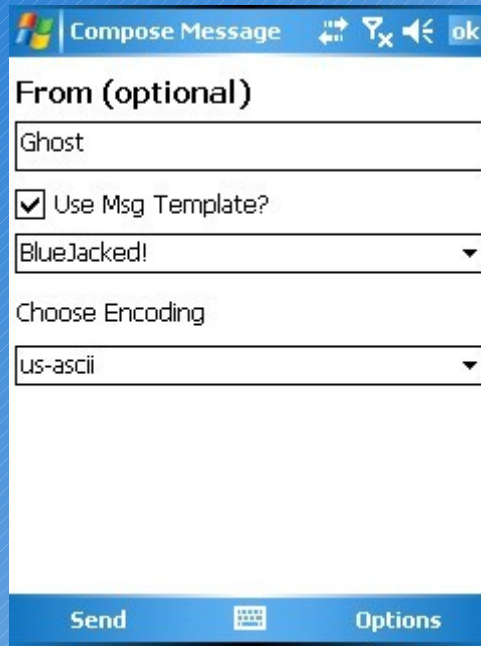
Ha una modalità di discovery per cui avverte vibrando della presenza di un nuovo device bluetooth nel range

Utile per il bluejacking (tramite vCard) o per mandare file di testo

<http://www.easy-jack.co.uk/index.html>



Bluetooth sw: PocketPC&SmarthPhone



Sniper

Nato per il bluejacking, spedisce messaggi o file in maniera anonima (via bluetooth), una specie di chat via bluetooth

<http://www.gadgetapps.net/info/6/Sniper%20for%20PocketPC%201.2.368>



La pratica: attacco RedFang e Bluebug

Eseguo una scansione dei dispositivi bluetooth ottenendo un BT_ADDR (sia visibile, sia invisibile tramite RedFang)

- hcitool scan
- ./fang -r 00803761A900-00803761A9FF -d

creo un bind tra un determinato servizio nascosto (17, il servizio AT appunto) un device RFCOMM del sistema aggressore

- rfcomm bind 1 00:0A:D9:xx:xx:xx 17

lancio minicom e lancio i miei comandi AT

- ATDT+39340xxxxxxx (effettua una chiamata al numero specificato)



La pratica: attacco BlueSnarf

Eseguo una scansione dei dispositivi bluetooth ottenendo un BT_ADDR (sia visibile, sia invisibile tramite RedFang)

- hcitool scan
- ./fang -r 00803761A900-00803761A9FF -d

Faccio una ricerca dei servizi RFCOMM disponibili e trovo che al canale 10 mi risponde il servizio OBEX Object Push

- sdptool browse 00:80:37:61:A9:22

A questo punto tento di prendere il file della rubrica (-g = GET)

- obexftp -b 00:80:37:61:A9:22 -B 10 -g telecom/pb.vcf



La pratica: BlueBag



Design italiano per non dare nell'occhio

<http://www.ikkisoft.com/bluetooth.html>



Linkografia

<http://it.wikipedia.org/wiki/Bluetooth>

<http://trifinite.org/>

<http://www.betaversion.net/btdsd/> - Database per blueprinting

<http://sicurezza.html.it/articoli/leggi/1664/bluetooth-tecniche-di-attacco-e-difesa/>

<http://www.bluetoothjacking.com/>

<http://www.bluejackq.com/>

<http://student.vub.ac.be/~sijansse/2e%20lic/BT/welcome.html>

<http://www.ikkisoft.com/bluetooth.html>

...e molti altri...



Questo documento è rilasciato sotto licenza Creative Commons Attribuzione-Non commerciale-Condividi allo stesso modo 2.5 Italia License

