Attacco

I love your phonebook - !Bluetooth hack!-

Matteo Valenza
Pierpaolo Palazzoli
Fabio Mostarda

Grado di difficoltà

L'aspetto preoccupante è che, a fronte di una notevole diffusione di apparati con supporto bluetooth (anche maggiore del wireless 802.11) e nonostante siano conosciuti numerosi bug che li affliggono, non vi sia consapevolezza dei rischi di sicurezza ad essi associati

Quando parliamo di bluetooth intendiamo quella tecnologia molto diffusa che permette uno scambio di dati via etere. Non un "dente blu" come indica il significato letterale del termine ... Moltissimi dispositivi dispongono del bluetooth, primi fra tutti i cellulari, i pda, i personal computer, i dispositivi di gioco e di controllo (come mouse, tastiera ecc...); il bluetooth è dunque largamente utilizzato e massicciamente impiegato nella tecnologia moderna.

L'aspetto preoccupante è che, a fronte di una notevole diffusione di apparati con supporto bluetooth (anche maggiore del wireless 802.11) e nonostante siano conosciuti numerosi bug che li affliggono, non vi sia consapevolezza dei rischi di sicurezza ad essi associati. Per comprendere meglio la portata di queste vulnerabilità, conviene portare un esempio concreto. E' noto che alcune regole poste dalla legislazione di vari paesi, come il divieto di parlare al cellulare in automobile, hanno favorito lo sviluppo di dispositivi complementari che permettono di parlare a mani libere: il vivavoce d'auto bluetooth o il diffuso auricolare bluetooth ne sono un esempio; i dati che questi dispositivi si scambiano attraversano l'aria e non sono crittografati: basterebbe quindi intercettare il traffico per ascoltare una conversazione! L'unica "sicurezza" esistente è l'utilizzo del codice di pairing per stabilire la prima connessione; questo metodo è però facilmente intercettabile e pressochè inutile ai fini della sicurezza, e si utilizza per stabilire se la connessione tra due dispositivi è legittima.

La tecnologia: standard e difetti strutturali del protocollo

Il protocollo wireless Bluetooth nacque su iniziativa del gruppo Ericcson, e dal 1999 è supportato dal consorzio SIG (Bluetooth Special Interest Group). Il nome non deriva da qualche razza ittica svedese, come si potrebbe pensare di primo acchito, bensì dal soprannome

Dall'articolo imparerai...

- Minima conoscenza di linux.
- Conoscenza dello stack bluetooth linux.

Cosa dovresti sapere...

- attaccare dispositivi bluetooth.
- imparare a proteggerti da potenziali aggressori.

Listado 1. *lsusb*

```
# lsusb
Bus 004 Device 001: ID 0000:0000
Bus 003 Device 002: ID 07b8:b02a
      D-Link Corp.
Bus 003 Device 001: ID 0000:0000
Bus 002 Device 001: ID 0000:0000
Bus 001 Device 001: ID 0000:0000
```

di un sovrano danese del nono secolo, molto conosciuto nella cultura nordica; il simbolo del Bluetooth è costituito in effetti dalle iniziali del suo nome, scritte in rune antiche. La tecnologia Bluetooth si basa su comunicazione radio (da 2.4 a 2.4835 GHz) a salto di frequenza (fino a 1600 volte al secondo); a seconda della potenza disponibile, possono essere determinate tre classi di dispositivi:

- Classe 1: potenza max di 100 mW (20 dBm), raggio max 100 metri
- Classe 2: potenza max di 2.5 mW (4 dBm), raggio max 10 metri
- Classe 3: potenza max di 1 mW (0 dBm), raggio max 1 metro

I dispositivi Bluetooth possono comunicare tra loro istituendo delle reti adhoc che prendono il nome di piconet; ogni rete può contenere sette dispositivi, dei quali uno è eletto Master.

Il trasferimento dati può avvenire in modo unidirezionale tra quest'ultimo e gli altri apparati; la bidirezionalità è garantita dalla rotazione della posizione di Master in modalità round robin. Reti di maggiori dimensioni potrebbero teoricamente essere realizzate aggregando più piconet mediante dispositivi in modalità bridge (slave su una piconet e master sull'altra), permettendo la creazione di scatternet. Le velocità di connessione sono passate dai 200 Kbit/s del Bluetooth 1.0 ai 2 Mbit/s dello standard versione 2.1, con la prospettiva di crescere ancora. Al momento della connessione, il dispositivo Bluetooth scambia varie informazioni: nome, classe, servizi e informazioni tecniche (produttore, ecc..); questa scambio avverrà o in risposta ad una ricerca effettuata da un altro device o in seguito ad una richiesta mirata al proprio indirizzo (di 48 bit). Di questi 48 bit, i primi 24 sono dipendenti dal produttore (e quindi statici) e solo gli ultimi 24 sono variabili: questo comporta la fattibilità di scovare l'indirizzo di un apparato anche in assenza di altre informazioni.

Sono previsti 3 livelli di sicurezza:

- Security Mode 1: in chiaro.
- Security Mode 2: sicurezza a livello applicativo, senza concorso del protocollo Bluetooth.
- Security Mode 3: crittografia affidata agli apparati.

Per garantire una certa autenticazione tra gli apparati, si può ricorrere alla tecnica di pairing; questo consiste nello scambio di una credenziale condivisa (PIN), che viene conservata ed usata in tutte le comunicazioni successive tra i due apparati. Dal punto di vista della confidenzialità, alcuni apparati Bluetooth supportano due algoritmi crittografici:

- E21/E22 per la generazione delle chiavi di inizializzazione della connessione
- E0 per la crittografia dei pacchetti.

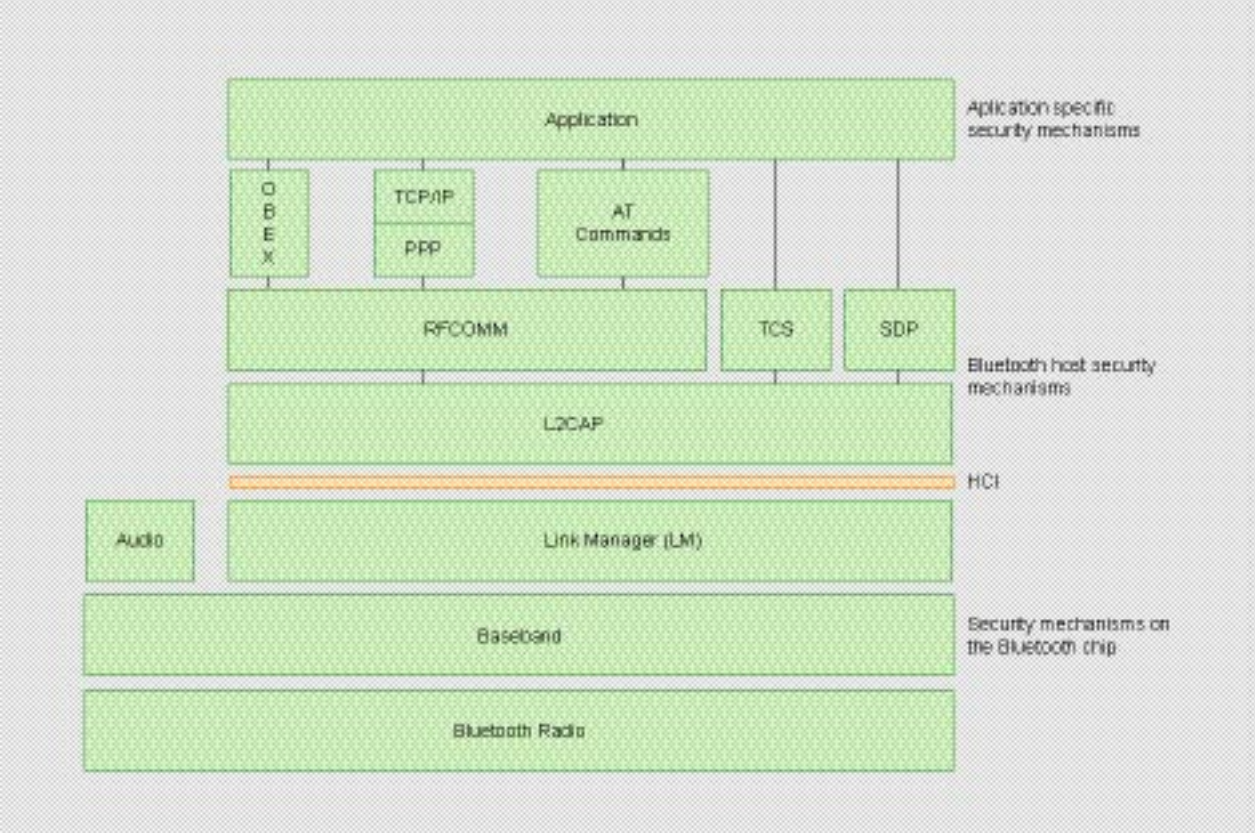

Figura 1. *Piconet e Scatternet*

I primi due sono cifrari a blocchi della famiglia SAFER+, e permettono di generare una chiave di 128 bit a partire da un numero PIN e da un random di 128 bit; la maggiore debolezza in questa parte della tecnologia Bluetooth sta nel lassismo di certe implementazioni, che accettano PIN di sole 4 cifre. L'algoritmo E0 è invece un cifrario a stream (come RC4, per intenderci) basato su chiavi a 128 bit, e si è rivelato vulnerabile ad attacchi statistici: il numero di operazioni necessarie ad ottenerne la chiave è 2^38 (anzichè 2^128).

Analizzando lo stack Bluetooth, è facile notare la presenza di un layer di separazione (HCI, Host controller Interface) tra servizi di basso livello (muxing/demuxing radio, sincronizzazione dei link, streaming, etc..) e servizi con maggior grado di astrazione.

I più interessanti dal punto di vista della sicurezza sono quelli posti sopra il layer HCI e sotto i protocolli di comunicazione standard (TCP, etc...), ovvero:

- L2CAP (Logical link control and adaptation protocol): effettua il multiplexing di servizi superiori, l'incapsulamento dei pacchetti e la gestione del Channel Identifier; supporta comunicazioni sincrone e asincrone.
- RFCOMM: un emulatore di seriale RS232.
- SDP: gestione/discovery di servizi.
- TCS (Telephony Control protocol Specification)
- OBEX (OBject EXchange); questo servizio permette di scambiare oggetti tra diversi apparati mediante azioni di "push"; sfortunatamente, alcune implementazioni scadenti permettono di effettuare operazioni di "pull" (reperimento di oggetti dal nome conosciuto) senza autenticazione, con conseguenze facilmente immaginabili.

I maggiori esperti di sicurezza informatica sono concordi nel ritenere che l'insicurezza dei dispositivi Bluetooth non derivi tanto dal protocollo, quanto dalle manchevolezze delle sue implementazioni; in effetti, gli attacchi ad oggi conosciuti contro il protocollo Bluetooth vero e proprio sono:

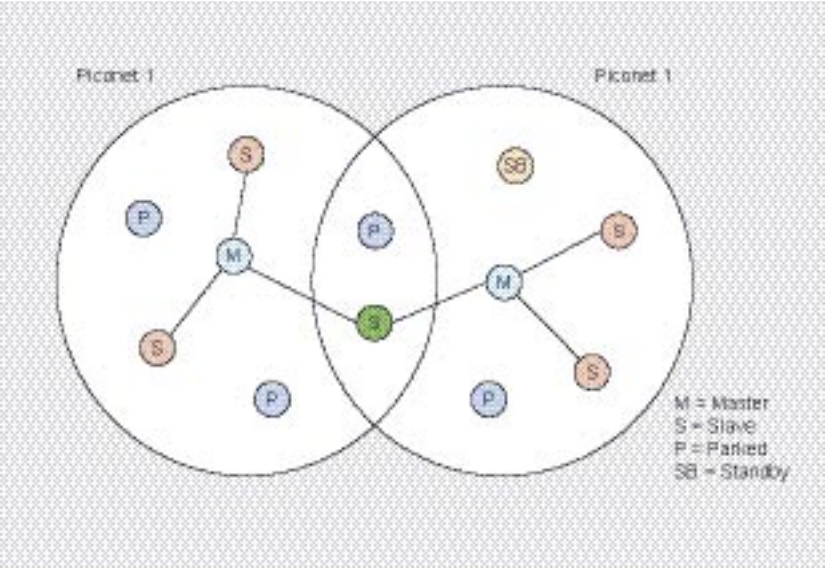

Figura 2. *Bluetooth Stack*

- Bruteforcing delle chiavi di E21/22 (con la complicità di password corte...)
- BlueDump, un attacco che sfrutta il meccanismo di pairing. Data una coppia di device che si sono già autenticati mediante pairing, conoscendo l'indirizzo fisico di uno di essi è possibile inviare con tale indirizzo una richiesta di Reset del pairing; questa, se accettata, compromette la sicurezza.
- Blueprinting, ovvero la raccolta di informazioni tecniche (produttore, versione, OS..) da apparati Bluetooth in raggio di rilevazione; utile sia a fini di auditing di sicurezza che per creare un database di device da attaccare.
- Bluejacking, una tecnica che sfrutta i messaggi di pairing per inviare comunicazioni non gradite che, se il pairing va comunque a buon fine in quanto ingenuamente accettato dall'utente bersaglio, possono portare alla compromissione dell'apparato.

Molto più insidiosi sono gli attacchi alle implementazioni dello stack Bluetooth, in particolare ai moduli soprastanti l'HCI; i più famosi sono:

- BlueSnarf: utilizza una richiesta ("pull") di un oggetto OBEX per accedere a servizi sul device bersaglio; a causa dell'incredibile superficialità degli sviluppatori, tale device supporrà che, per invocare un oggetto, il richiedente debba averlo conosciuto in precedenza: ritendendo inutile l'autenticazione, sarà quindi sufficiente invocare tali servizi...ed essi risponderanno!
- HELOMoto: sfrutta gravi bug OBEX nello stack Motorola, ovvero il mantenimento di una connessione trusted dopo che un qualunque push OBEX viene interrotto; forni-

Listato 2. *hcitool scan*

```
# hcitool scan
Scanning ...
    00:12:D1:XX:XX:XX FlO
Con il comando sdptool browse 00:12:D1:
      XX:XX:XX visualizziamo i servizi resi
      disponibili dal dispositivo 00:12:D1:XX:XX:XX
# sdptool browse 00:12:D1:XX:XX:XX
```

sce pieno accesso AT, permettendo quindi di effettuare chiamate, mandare sms, etc...

- BlueBug: sfruttando implementazioni RFCOMM che non annunciano tutti i servizi su SDP, permette di sfruttare tali servizi nascosti per lanciare comandi AT, così da effettuare chiamate, mandare sms, etc...
- BlueBump: un attacco che, partendo da una autenticazione legittima, ad esempio con vCard, cerca di ottenere il reset della chiavi di pairing del bersaglio al fine di autenticarvisi in un secondo momento; quest'attacco presuppone quindi un approccio sotto "mentite spoglie", sfruttando tecniche di social engineering o di altro tipo, che poco hanno a che fare con il protocollo Bluetooth.
- BlueSmack: è un attacco che sfrutta pacchetti echo L2CAP malformati per causare buffer overflow e DoS.
- CarWhisperer: sfruttando le chiavi di default di molti dispositivi vivavoce per auto, permette di autenticarsi ad essi ed ascoltare conversazioni o addirittura iniettare contenuti audio all'interno del veicolo.

E' facile notare come gli attacchi ai livelli superiori dello stack siano in grado di causare con poco sforzo effetti notevoli, grazie alla complicità di implementazioni Bluetooth affette da bug; forse la causa di questi problemi diffusi è una ingegnerizzazione più focalizzata sulla semplicità di utilizzo dell'interfaccia uomo-macchina dell'apparato che non sulla sua sicurezza di esercizio.

Primi passi

Munirsi di un nokia 7650 o nokia 8310i o di un motorola razor v3 ...

Prenderemo come esempio un nokia 8310i.

Chiamiamo il cellulare "FIO", accendete il dispositivo e attivate il bluetooth. In questo momento il cellulare, in modalità "visibile a tutti", manda nell'etere un messaggio contenente il nome (FLO) e attende un collegamento senza fili. Assicuriamoci di aver installato correttamente il dispositivo bluetooth sul pc, assicurarsi di disporre di un kernel con i moduli necessari per il collegamento bluetooth. Nel caso in cui non siano presenti i moduli necessari (fare riferimento a kernel.org), ricompilare il kernel e installare il servizio bluetooth. Su una distribuzione come Debian, l'installazione del servizio bluetooth si esegue tramite il comando apt-get install bluetooth. Tramite lo script di init /etc/init.d/bluetooth restart è possibile avviare il servizio che gestisce il demone bluetooth. Inseriamo il device e osserviamo tramite un tail -f /var/log/message il riconoscimento automatico del device come usb bluetooth (esempio di tail -f message). Il dispositivo che abbiamo utilizzato (usb bluetooth 2.0) ci indica il suo corretto funzionamento, tramite una luce rossa intermittente.

Assicuriamoci che il dispositivo sia installato:

Con il comando hcitool scan scansioniamo l'aria alla ricerca di dispositivi bluetooth.

Installazione security bluetooth tools

L'attacco e il download di informazioni sensibili, al bluetooth di alcuni cellulari, è possibile nel caso in cui questi sono vulnerabili; per sapere quali servizi e quali telefoni trovati sono vulnerabili utilizziamo il programma btscanner. Questo, abbastanza datato, permette con una semplice interfaccia Ncurses di sapere con esattezza se il telefono è suscettibile ad alcuni attacchi. L'installazione è molto semplice, esistono i pacchetti per moltissime distribuzioni: nel caso di Debian, è possibile installare il programma tramite apt-get install btscanner. Eseguiamo btscanner in terminale con permessi di root, lanciamo un inquiry scan (tramite la pressione del tasto 'i') e rapidamente ci verrà fornito un elenco di dispositivi disponibili nel raggio d'azione dell'antenna bluetooth, indicando la vulnerabilità se nota. Il Gain dell'antenna bluetooth è molto importante, più è alto il Gain più il nostro segnale si propagherà nell'aria. Per aumentare il range d'azione dell'antenna bluetooth, se l'antenna non è sostituibile, è possibile seguire questo how to che si adatta a qualsiasi antenna usb bluetooth. (link link link antenna trinfinite. immagine.) Proseguiamo installando una suite di programmi di audit e hack di dispositivi bluetooth. BlueDivingNG è uno script in perl che racchiude in una interfaccia semplice e testuale tutti gli attacchi documentati fino ad oggi. Tramite l'utilizzo combinato di più programmi e script con BluedivingNG è possibile effettuare un test di sicurezza di alto livello, su dispositivi bluetooth di diverso tipo. BlueDivingNG non è di facile comprensione e installazione, cercheremo durante questa lettura di informarvi riguardo la prima installazione. Scaricare ed

Listado 3. *make-tools.sh*

```
# vi tools/make-tools.sh
#!/bin/bash
make
echo -en "\n<<< Compiling bccmd\n"
cd bccmd_src
make
mv bccmd ..
cd ..
echo -en "\n<<< Compiling btftp\n"
cd btftp_src
make
mv btftp ..
cd ..
echo -en "\n<<< Compiling btobex\
                    n"
cd btobex_src
make
mv btobex ..
cd ..
echo -en "\n<<< Compiling bss\n"
cd bss-0.8
make
mv bss ..
cd ..
echo -en "\n<<< Compiling
      carwhisperer\n"
cd carwhisperer-0.2
make
mv carwhisperer ..
cd ..
echo -en "\n<<< Compiling
      greenplaque\n"
cd greenplaque_src/
make
mv src/greenplaque ..
cd ..
echo -en "\n<<< Compiling
      redfang\n"
#tar xfvz redfang.tar.gz
#mv redfang redfang_src
cp redfang_src/fang redfang
```

estrarre BlueDivingNG0.8, all'interno della cartella troviamo un utile README che ci avverte di alcune dipendenze utili all'esecuzione corretta del programma, come ad esempio un kernel 2.4 / 2.6, bluez, sox, obexftp, libreadline, expat / XML::simple. All'interno del documento ci sono istruzioni per installare in maniera automatica tutti i programmi tramite lo script in bash make-tools.sh situato all'interno della cartella tools/. Editiamolo e cambiamo la path di bss-0.6 a bss-0.8 , commentiamo 2 righe relative al processo di installazione di redfang, tramite un cancelletto avanti a #tar xfvz redfang.tar.gz e #mv redfang redfang_src.

Apriamo la cartella tools/btftp_src/ ed editiamo il file folder.c. Nelle prime righe notiamo alcuni include delle librerie xml da verificare. Usciamo senza salvare e cerchiamo l'esatta posizione di xmlmemory.h e di parser.h tramite il comando locate.

Per la distribuzione utilizzata in esempio, la posizione è gnome-xml/xmlmemory.c e gnome-xml/parser.h Editiamo il file tools/btftp_src/folder.c e correggiamo la posizione delle librerie xml, usciamo e salviamo.

Torniamo nella directory principale del programma e lanciamo make-tools.sh, se tutto è corretto non dovrebbero esservi errori, i vari tools dovrebbero essere compilati e funzionanti. Lanciamo il programma tramite il comando perl bluedivingNG.pl. Nel caso in cui ci fossero errori bloccanti il programma esce e genera in output a display un messaggio contenente l'errore trovato. Fate riferimento al README per altre delucidazioni.

Qui sotto, la schermata principale di BlueDivingNG.pl

Attacco a dispositivi Bluetooth

L'utilizzo di BTscanner può anche non essere necessario inquanto BluedivingNG è anche un abile scanner di device bluetooth. Importante precisare anche che BluedivingNG implementa una scansione tramite più device bluetooth la cosiddetta greenplaque. Questa scansione risulta più precisa non perchè il range di attività aumenti, ma in quanto esistono delle incompatibilità tra i vari device, ed avere diverse antenne bluetooth da poter utilizzare è sicuramente più indicato per una attività come la nostra. Lanciamo la procedura di scan tramite la pressione del tasto 1; alla presenza di un dispositivo il computer emetterà un suono simile ad una esplosione, non è esploso il pc ... tranquilli. Il programma scansionerà l'aria alla ricerca di dispositivi visibili e farà un elenco.

Copiamo il bdaddr del dispositivo e aggiungiamolo ai device conosciuti (add known device) tramite la pressione del tasto 4.

Inseriamo il bdaddr e un nome identificativo. Andiamo nel menù 'a' che sta per attack e lanciamo l'attacco referito. Molte volte non conosciamo il tipo di attacco che può andare a buon fine così lanciamo 'Try all attack' tramite la pressione del tasto '3'. Il programma tenterà di attaccare con tutti i mezzi disponibili. Alcuni errori dello script in perl fanno bloccare il programma. Nel caso in cui l'attacco di tipo bluesnarfer non riesca, Bluediving esce, in tal caso rilanciatelo e ripetere l'operazione, riprendendo dall'attacco seguente. Olte alla comoda interfaccia di attacco è presente anche un'area dove sono disponibili gli exploit conosciuti 'e'. Da questa, è possibile lanciare exploit direttamente al dispositivo aggiunto ai known device. Alcuni degli exploit lanciati

Listado 4. *xml libraries*

```
# locate xmlmemory.h
/usr/share/doc/libxml-1.8.17-r2/html/gnome-xml-xmlmemory.html
/usr/include/gnome-xml/xmlmemory.h
/usr/include/libxml2/libxml/xmlmemory.h
```

Listado 5. *folder.c*

```
# vi tools/btftp_src/folder.c
/*
 *
Bluetooth Generic Object Exchange Profile
 *
Copyright (C) 2003 Marcel Holtmann <marcel@holtmann.org>
 *
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License version 2 as
 * published by the Free Software Foundation;
 *
 * Software distributed under the License is distributed on an "AS
 * IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or
 * implied. See the License for the specific language governing
 * rights and limitations under the License.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
 *
 */
#ifdef HAVE_CONFIG_H
#include <config.h>
#endif
#include <stdio.h>
#include <errno.h>
#include <time.h>
#include <sys/param.h>
#include <gnome-xml/xmlmemory.h>
#include <gnome-xml/parser.h>
#include "goep.h"
#include "ftp.h"
```

potrebbero far connettere il nostro pc al dispositivo bluetooth, con o senza lo scambio di una chiave. Una volta lanciato l'exploit siamo effettivamente connessi. A questo punto è possibile lanciare un attacco: prendiamo in considerazione bluesnarfer. Con questo attacco è possibile copiare le entry della rubrica telefonica, cancellarle e inserire di nuove.

Tramite BluedivingNG è anche possibile fare una injection di file audio, mandare un file audio direttamente al cellulare, il quale immediatamente provvederà ad eseguirlo nelle cuffie o nello speaker del telefono target. E' divertente osservare la reazione del povero malcapitato. Per effettuare questi attacchi non è necessario obbligatoriamente l'utilizzo di un computer, basta anche un cellulare. I ragazzi di trifinite, hanno sviluppato un'applicazione chiamata Bloover e BlooverII (breeder) che fa un po' il lavoro di bluediving, racchiude diversi attacchi e funzionalità utili al nostro scopo...

Io personalmente testai la versione breeder (Speciale, edizione 22c3) su un Nokia N80, sono compatibili tutti i telefoni ingrado di far girare api java 2.0. Prima di installarlo verificatene la compatibilità. In questo caso il programma è molto versatile, è possibile tramite il menù options apportare modifiche di comportamento

Figura 3. *BlooverII*

del programma, ad esempio dato un target è possibile selezionare quali attacchi lanciare, che tipo di azioni intraprendere ecc... Fondalmentalmente svolge egregiamente il suo lavoro, anche se alcune volte si blocca inaspettatamente dopo l'attacco BlueBug. L'esecuzione del programma non compromette le normali funzionalità del telefono target in quanto è fatto solo per generare reportistica, anche se risulta possibile copiare e inserire entry della rubrica del bersaglio. Tornando a BluedivingNG, è possibile anche sniffare il codice di pairing durante un legittimo scambio di chiavi e attaccare direttamente un dispositivo mandando un ping tramite l2ping; tramite quest'ultimo il dispositivo target sarà notevolmente impegnato a rispondere al nostro ping e l'esecuzione del sistema operativo (tipicamente symbian) risulterà appesantito se non bloccato completamente. Questo attacco ricorda il ping of the death che ha afflitto per anni il sistema windows98; precisiamo che l'attacco dos è ancora uno dei più infallibili e dannosi per un dispositivo, che si tratti di un apparato telefonico o di un servizio web. Un altro attacco molto utile è il deauth: quando un dispositivo è associato ad un altro è possibile lanciare un comando di deautenticazione che scollega uno dei due dispositivi e li riobbliga allo scambio delle chiavi. Questo attacco è lo stesso che potremmo trovare in un articolo sulla sicurezza di reti wireless, molto utile per generare traffico e individuare così più facilmente la chiave scambiata ad inizio pairing. Altri tipi di attacchi che si possono effettuare sono ad esempio una connessione diretta rfcomm shell, dove avendo permessi di root è possibile eseguire comandi direttamente dal cellulare. Effettuare una chiamata o utilizzare il servizio modem... in questo caso però verrà visualizzata sul dispositivo la chiamata in corso e quindi vi è anche la possibilità di essere scoperti o di chiudere la connessione direttamente dal dispositivo.

Listado 6. *Esecuzione di Bluediving*

```
bluediving-0.8 # perl bluedivingNG.pl
<<< Setting device name to phone of doom
<<< Setting device type to phone
<<< Setting device to non-visible mode
<<< Parsing vendor map... Done.
 _  _       _ _   _
                     | |__| |_ _ ___ __| (_)_ _(_)_ _ __ _
 | '_ \ | || / -_) _` | \ V / | ' \/ _` |
 |_.__/_|\_,_\___\__,_|_|\_/|_|_||_\__, |
           version 0.8 |___/
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
[MAIN MENU] menu: [a] Action [e] Exploit [i] Info [t] Tools
[1] Scan
[2] Scan and attack
[3] Scan and info
[4] Add Known Device
[5] Change preferences
[6] Show preferences
[7] Show logfile
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-= [x] Exit -
```

Bluetooth tips

Ho deciso di dedicare qualche minuto ai tips da me riscontrati durante

Listado 7. *Scansione con Bluediving*

```
<<< Start scanning for bluetooth devices...
<<< Greenplaque scanning mode need two or more hci devices
<<< Switching to hcitool scanning mode.
<<< Start scanning for bluetooth devices...
<<< Thu Jul 26 19:16:25 2007 Found host FlO addr
      00:12:D1:XX:XX:XX class unkown
```

Listado 8. *Aggiunta manuale di un device*

```
>>> 4
<<< Manually add a known bluetooth device...
Enter device address: 00:12:D1:XX:XX:XX
Enter a nickname for this device: FlO
Device FlO (00:12:D1:XX:XX:XX) registered.
```

questi giorni di studio del fenomeno attacco al bluetooth... Molti lasciano inconsciamente o volontariamente il bluetooth acceso e visibile 24 ore su 24 o per la durata dell'accensione del dispositivo, perchè come detto prima il fatto di aver a disposizione un auricolare o un vivavoce in macchina ne preclude l'accensione e la visibilità. Non tutti i dispositivi mobili sono vulnerabili e quindi è utile trovarsi in posti molto affollati tipo scuole, grandi aziende o locali notturni. Molto importante è la quantità di dispositivi antenna disponibili all'attaccante perchè, come accennavo, esistono diverse incompatibilità tra i device, quindi importante utilizzare antenne di tipo diverso bluetooth 1 e bluetooth 2 di marche differenti. BluedivingNG gestisce più di una antenna bluetooth: io ho testato fino a 4 antenne bluetooth contemporaneamente tramite un hub usb alimentato. Attenzione al nome del dispositivo bluetooth, in quanto il vostro pc potrebbe chiamarsi: "pc di adam" ... meglio mascherare il nome con uno + attribuibile ad un dispositivo di telefonia ... quindi un nome qualsiasi tipo : NokiaN80_giulia Il nome non l'ho scelto a caso ... NokiaN80 perchè mi maschero da nokia e giulia ... mi maschero da donna ... Come in un recente articolo pubblicato su questa rivista (tema: SocialEngineering), è utile mascherare la propria identità per il seguente motivo: quando con BluedivingNG lanciamo scan and attack o automatic attack, su alcuni dispositivi verrà visualizzato il classico messaggio: "Vuoi ricevere un messaggio da ..." questo perchè non tutti i telefoni sono vulnerabili e quindi nel sondare la vulnerabilità (nel caso in cui il disp. non fosse vulnerabile) viene visualizzato il messaggio e sicuramente è meglio essere mascherati per non incorrere in brutte sorprese. Mascherarsi con nomi femminili è utile perchè pochi ragazzi risponderebbero "no" rifiutando il messaggio, altrimenti si potrebbe utilizzare il nome "Tim - Ricarica" così sarà visualizzato : "Ricevere un messaggio da Tim - Ricarica ?". Se trovate dispositivi mobili non vulnerabili il massimo danno che si può fare è continuare a tentare di connettersi e lanciare un ping of the death, la connessione può essere stabilita anche senza scambio di chiavi, magari mascherandosi da headset... (cuffie) accettando il messaggio ... quindi a questo punto lanciamo tutti gli attacchi possibili, anche se non vanno a buon fine il dispositivo sarà oberato di richieste e personalmente ho notato che telefonini non vulnerabili "svarionavano", ossia mostravano una estrema lentezza ed un uso della batteria massiccio. Alcuni Modelli della Nokia dopo un attacco al bluetooth non terminavano l'illuminazione del display, quindi consumavano la batteria molto + velocemente.

Rendi sicuro il tuo cellulare

Installare software tipo antivirus sul cellulare è abbastanza inutile, in quanto questi non possono fare nulla contro connessioni o vulnerabilità note, possono solo proteggere da esecuzioni involontarie di codice malevolo, sotto forma di file di installazione ecc... I virus per i dispositivi mobili si comportano come un virus per pc, tentano di replicarsi inviandosi automaticamente ai destinatari della rubrica o tramite bluetooth attivo ai dispositivi compresi nel raggio d'azione bluetooth, tipicamente 10 Mt.

Tempo fa era possbile far andare in crash i dispositivi tramite l'utilizzo di spazi e caratteri speciali all'interno del nome di riferimento del dispositivo bluetooth ... esempio : "C3llulare Di Marc0 --- ZxZ XXX ZZZ xxx VVVFFFGGGGTYTYTYTY$%&/ &%$£!"£$%&/(" Durante una scansione con un altro dispositivo mobile era classico il repentino crash e riavvio del sistema operativo o addirittura l'impossibilità di collegamento con il dispositivo. La chiave di pairing non lasciatela memorizzata all'interno delle opzioni bluetooth del dispositivo, in quanto è possibile ricavarla facilmente e velocemente tramite un attacco di tipo bruteforce. Utilizzate chiavi di pairing complesse, faccio questa precisazione perchè le chiavi di pairing di default dei comuni dispositivi mobili sono: "1234" "0000" "1111". In alcuni telefoni, Nokia ad esempio (N80), nelle opzioni Bluetooth è specificato se il dispositivo può richiedere un dato in rubrica sim: inutile dire che è meglio limitarne l'accesso da remoto ai soli dispositivi conosciuti o ancora meglio a nessuno. Resta comunque una cosa molto importante da fare, tenere aggiornato il firmware del dispositivo. Purtroppo le maggiori case produttrici di

In Rete

http://trifinite.org Bluesnarf e vari altri tools
http://bluediving.sourceforge.net/ Bluediving
http://bluetooth-pentest.narod.ru/ Sito per approfondimenti vari sul Bluetooth
http://video.google.com/videoplay?docid=2254620871308597290 bluesnarfer video

dispositivi mobili aggiornano solo i modelli in produzione, e gli aggiornamenti del firmware non sono fattibili dagli utenti: occorre recarsi nei centri specializzati e pagare una piccola somma; personalmente ho fatto aggiornare il Nokia 7650 con l'ultimo firmware disponibile a 25 Euro.

Conclusione

Tenere acceso il bluetooth inutilmente è uno spreco di energia e ed è pericoloso per tutti i motivi che abbiamo spiegato sopra: usate il bluetooth solo in caso di necessità. Nessun telefono è esente dal problema degli attacchi al bluetooth (tranne quei telefoni che non ne dispongono) perchè come abbiamo notato l'apparato rimane in attesa di un collegamento e continuamente manda in etere il proprio nome; analogamente al wireless, il bluetooth risulta essere una tecnologia senza un "sicuro" futuro. Occorre rivedere completamente il protocollo invece di continuare ad aumentarne la capacità elaborativa... Da ultimi studi e ricerche il Bluetooth è stato notevolmente velocizzato ma questo aumento non porterà giovamento alla sicurezza, anzi renderà ancora più accessibili i dispositivi mobili e garantirà una velocità di risposta maggiore rispetto a quella attuale, con il pericolo che le informazioni spostate prima in un minuto si sposteranno in 6 secondi.. ●

Sull'autore

Snortattack.org, Portale orientato alla sicurezza informatica, è il risultato della fusione di conoscenze e collaborazione del team. Le tipologie di argomentazione trattate coprono 360 gradi tutte le tematiche relative alla sicurezza: attacco/difesa. Grande punto di forza è l'uso di Snort come soluzione alle inummerevoli problematiche di intrusione. Per tenere gli utenti aggiornati sulle nuove problematiche sono a disposizione un forum e una mailinglist. Con Snortattack.org, si intende creare uno Snort User Group finalizzato alla collaborazione per l'Italia e tutto il resto del Mondo, per l'uso di Snort e la trattazione di problematiche di security.