Bluetooth Security: Beyond Bluebug

Session 5
Adam Laurie, Marcel Holtmann, Martin Herfurt
Tuesday, November 15th, 2005 – 2:45 pm
Agenda

- Bluetooth technology overview
- The security mechanisms
- Known vulnerabilities
- Tools that are used
- Live demonstration
Who is investigating

● Adam Laurie
  • CSO of The Bunker Secure Hosting Ltd.
  • DEFCON staff and organizer

● Marcel Holtmann
  • Maintainer of the Linux Bluetooth stack

● Martin Herfurt
  • Security researcher
  • Founder of trifinite.org
What is this about

New device found: Airbus A310
Proceed with configuration?
What is Bluetooth

- Bluetooth SIG
  - Trade association
  - Founded 1998
  - Owns and licenses IP

- Bluetooth technology
  - A general cable replacement
  - Using the ISM band at 2.4 GHz
  - Protocol stack and application profiles
How it works

● Data and voice transmission
  • ACL data connections
  • SCO and eSCO voice channels

● Piconet and scatternet topology

● Frequency hopping
  • 79 channels
  • 1600 hops per second
Creating the topology

- Hopping sequence defines the piconet
  - Master defines the hopping sequence
  - Up to seven active slaves
  - Scatternet creation
Bluetooth architecture

- **Hardware layer**
  - Radio, Baseband and Link Manager
  - Access through the Host Controller Interface
    - Standards for USB and UART

- **Host protocols**
  - L2CAP, SDP, RFCOMM, BNEP, AVDTP etc.

- **Application profiles**
  - Serial Port Profile, Dialup, PAN, A2DP, HID etc.
Bluetooth stack

Application specific security mechanisms

Bluetooth host security mechanisms

Security mechanisms on the Bluetooth chip
Bluetooth security

- **Link manager security**
  - All security routines are on-chip
  - Nothing is transmitted in “plain text”

- **Host stack security**
  - Interface to the link manager security
  - Part of the HCI specification
  - Easy interface
  - No further encryption of pin codes or keys
Bluetooth link keys

● Needed for authentication

● Used for encryption
  • SAFER+ (128 bit block cipher)

● Generated by pairing process
  • Passkey (1-16 alphanumeric characters)
  • Random number (from device internal clock)
  • BD_ADDR of piconet master
Security modes

● Security mode 1
  • No active security enforcement

● Security mode 2
  • Service level security
  • On device level no difference to mode 1

● Security mode 3
  • Device level security
  • Enforce security for every low-level connection
Security commands

- **Settings**
  - HCI_{Read|Write|Delete}_Stored_Link_Key
  - HCI_{Read|Write}_Authentication_Enable
  - HCI_{Read|Write}_Encryption_Mode

- **Actions**
  - HCI_Authentication_Requested
  - HCI_Set_Connection_Encryption
  - HCI_Change_Connection_Link_Key
Pairing functions

● Events
  • HCI_Pin_Code_Request
  • HCI_Link_Key_Request
  • HCI_Link_Key_Notification

● Responses
  • HCI_Pin_Code_Request_[Negative_]Reply
  • HCI_Link_Key_Request_[Negative_]Reply
How pairing works

● First connection
  (1) > HCI_Pin_Code_Request
  (2) < HCI_Pin_Code_Request_Reply
  (3) > HCI_Link_Key_Notification

● Further connections
  (1) > HCI_Link_Key_Request
  (2) < HCI_Link_Key_Request_Reply
  (3) > HCI_Link_Key_Notification (optional)
How to avoid pairing

- vCard
- Contacts
- IrMC
- OBEX
  - Channel 3: OBEX Push Profile
  - Channel 4: Synchronization Profile
    - Security Manager
- RFCOMM
- L2CAP
BlueSnarf

● Trivial OBEX push attack
  • Pull knows objects instead of pushing
  • No authentication

● Discovered by Marcel Holtmann
  • Published in October 2003

● Also discovered by Adam Laurie
  • Published in November 2003
  • Field tests at London Underground etc.
BlueBug

- Issuing AT commands
  - Use hidden and unprotected channels
  - Full control over the phone
- Discovered by Martin Herfurt
  - Motivation from the BlueSnarf attack
  - Public field test a CeBIT 2004
- Possibility to cause extra costs
HeloMoto

- Requires entry in “My Devices”
- Use OBEX push to create entry
  - No full OBEX exchange needed
- Connect to headset/handsfree channel
  - No authentication required
  - Full access with AT command
- Discovered by Adam Laurie
Authentication abuse

● Create pairing
  • Authenticate for benign task
  • Force authentication
  • Use security mode 3 if needed

● Connect to unauthorized channels
  • Serial Port Profile
  • Dialup Networking
  • OBEX File Transfer
BlueSmack

- Using L2CAP echo feature
  - Signal channel request and response
  - L2CAP signal MTU is unknown
  - No open L2CAP channel needed
- Causing buffer overflows
- Denial of service attack
BlueStab

- Denial of service attack
  - Bluetooth device name is UTF-8 encoded
  - Friendly name with control characters
  - Crashes some phones
  - Can cause weird behaviors
  - Name caches can be very problematic

- Credits to Q-Nix and Collin R. Mulliner
BlueBump

- Forced re-keying
  - Authenticate for benign task (vCard exchange)
  - Force authentication

- Tell partner to delete pairing
  - Hold connection open
  - Request change of connection link key

- Connect to unauthorized channels
BlueSnarf++

- OBEX push channel attack, again
  - Connect with Sync, FTP or BIP target UUID
  - No authentication
  - Contents are browseable
  - Full read and write access
  - Access to external media storage

- Manufacturers have been informed
BlueSpooof

- Clone a trusted device
  - Device address
  - Service records
  - Emulate protocols and profiles
- Disable encryption
- Force re-pairing
BlueDump

- Yanic Shaked and Avishai Wool
  - [http://www.eng.tau.ac.il/~yash/Bluetooth/](http://www.eng.tau.ac.il/~yash/Bluetooth/)
  - Expands PIN attack from Ollie Whitehouse
  - Requires special hardware or firmware

- Destroy trust relationship
  - Use the BlueSpooof methods

- User interaction for pairing still needed
Blueprinting

- Fingerprinting for Bluetooth
- Work started by Collin R. Mulliner and Martin Herfurth
- Based on the SDP records and OUI
- Important for security audits
- Paper with more information available
Bluetooone

- Enhancing the range of Bluetooth dongle with a directional antenna
- Long distance attack after DEFCON 12
- Snarfing over 1,78 km
Blooover

- Bluetooth Wireless Technology Hoover
  - Proof-of-Concept application
  - Educational purpose only
  - Phone auditing tool
  - J2ME MIDP 2.0 and JSR-82

- Written by Martin Herfurt

- Announced at 21C3 in Berlin
Blooover II

- Successor of the popular Blooover
  - Auditing tool for professionals
  - More attacks than only BlueBug
  - Not all of the attacks are feasible on J2ME

- To be developed later
Blooonix

- Linux distribution for Bluetooth audits
  - LiveCD based on Morphix
  - Latest official Linux 2.6 kernel
  - Contains all latest BlueZ utilities
  - Includes also special hacker scripts
  - Graphical interface
  - Report generation

- Not available at the moment
BluePot

- Bluetooth HoneyPot
  - Runs on J2ME phones
  - Imitates vulnerable phone
  - Logs incoming attacks and device information
  - Strikeback capable

- Written by Martin Herfurt
- Not released yet
The Car Whisperer

- Use default pin codes to connect to carkits
- Inject audio
- Record audio
- Don't whisper and drive!
The Car Whisperer

● Stationary directional antenna
  • 15 seconds visibility
  • Average speed of 120 km/h (range 500 m)
Conclusions

- Bluetooth is secure standard (per se)
  - Problems are at the application level
- Cooperation with the Bluetooth SIG
  - Pre-release testing at UPF (UnPlugFests)
  - Better communication channels
  - Clear user interface and interaction
  - Mandatory security at application level
  - Using a policy manager
Further information

- trifinite.org
  - Loose association of security experts
  - Public information about Bluetooth security
  - Individual testings and trainings
  - TRUST = trifinite unified security testing

- Contact us via mws05@trifinite.org
Questions or feedback