Fuzz Testing Bluetooth Devices

Paul Shen

March 26, 2012

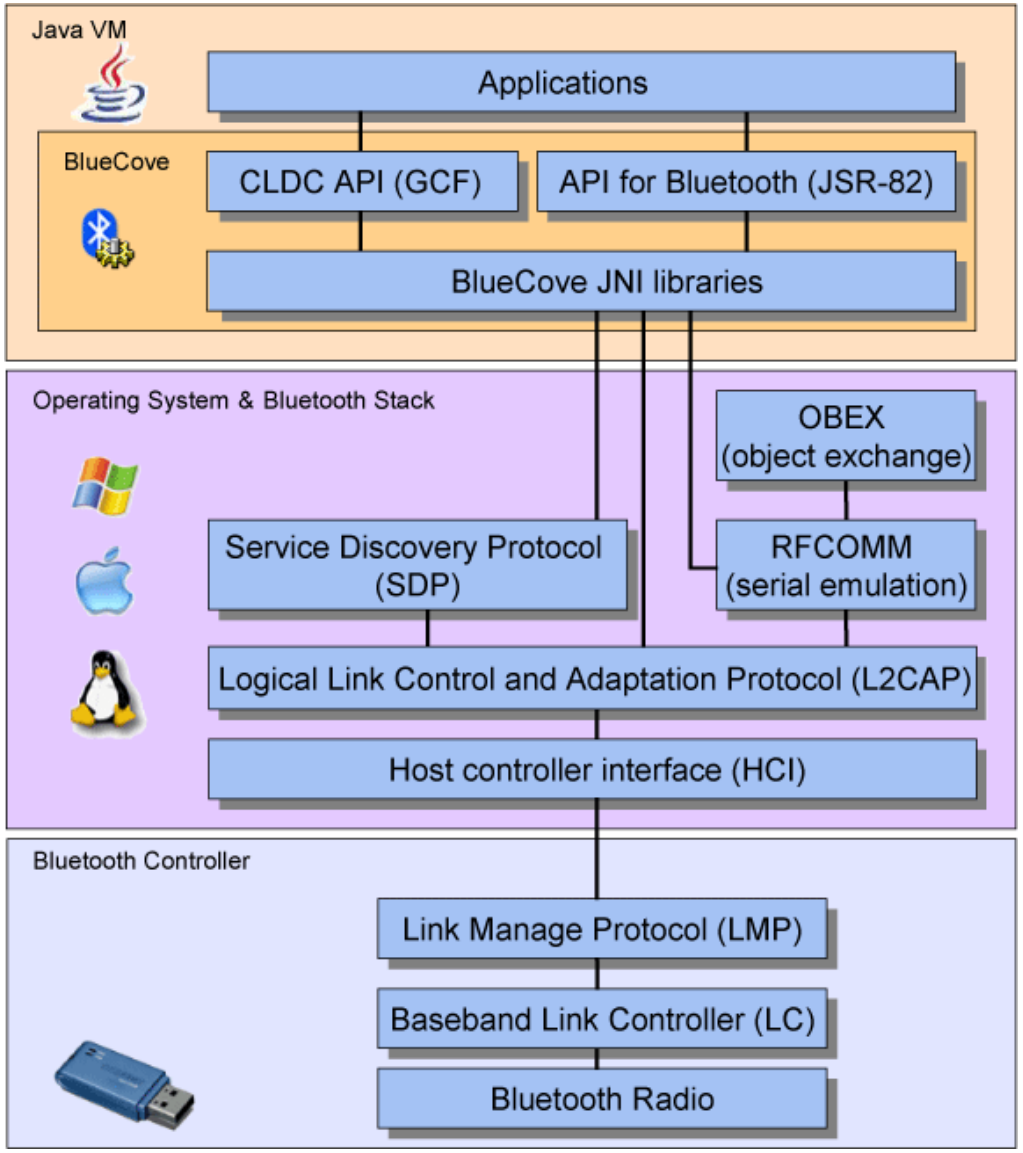
What is at stake?

- Bluetooth used in computers, mobile phones, handsfree equipment, and car audio systems, etc.
- Computers and mobile phones house a lot of personal information
- Bluebug exploit on handsets allowed remote access to text messages, call records and address books
- July 2011: Microsoft patched a Bluetooth vulnerability in Windows 7 and Vista that allowed an attacker to transmit packets to remotely execute code which allowed them to install programs; view, change, or delete data; or create new accounts with full user rights

Pairing Authentication in Bluetooth

- Pairing: a device receives a connection request from another unpaired device, user accepts request
- Requires and thus exposes the Service Discovery Protocol (SDP), in turn exposing core protocol L2CAP
- Bluetooth 2.0 and older: 4 digit pin used for verification, handsfree devices have hardcoded pin (usually 0000)
- Lacks robustness in protecting the protocol level against fuzzing – sending malformed packets to cause a crash – after pairing with a badly implemented device

Bluetooth stack



- A few Bluetooth stacks are used for many different Bluetooth products, so vulnerabilities in a specific stack apply to many different devices
- Stacks are usually either already known or can be easily learned

Fuzz Testing

- When a Bluetooth device receives an invalid message, software vulnerabilities often cause it to give an abnormal response:
  - crash
  - stop requiring pairing process
  - allow the installing and running of malware
- Invalid messages come from pairing with a non-conforming device or outside attacker
- Fuzz testing: invalid messages are fed to a system on purpose and the system's behavior is monitored

Crash-testing Bluetooth Devices

- Codenomicon, a Finnish data security company, tested 15 car kits, 5 mobile phones, 3 headsets and a picture frame with intelligent fuzz testing tools
- Bluetooth profiles specify general behavior that devices use to communication with each other
  - HFP: ability to conduct phone calls
  - A2DP: ability to play music located on a mobile device
- Device under test (DUT) setup: L2CAP – turn Bluetooth on and put into discoverable or pairing mode

Test Results and Conclusions

- All devices tested failed at one point; Codenomicon also claims that about 80% of devices in their plugfests have crashed
- L2CAP layer unreliable: most of the devices tested crashed within the first 100 cases of L2CAP protocol tests
- L2CAP doesn't require pairing, bypasses user acceptance and user may not even notice

The End

Any Questions?

http://arstechnica.com/business/news/2011/09/lousy-code-opens-up-bluetooth-hands-free-kits-smartphones-ars

http://bluecove.org/images/stack-diagram.png

http://www.codenomicon.com/resources/whitepapers/codenomicon_wp_Fuzzing_Bluetooth_20110919.pdf