Joseph Paul Cohen
http://blucat.sf.net
The wireless future is here now!
Overview

- Streams (w/jokes)
- blucat inline netcat replacements
- blucat as Bluetooth nmap
- rfcomm and l2cap basics
- look at some devices
- how to prototype
- blucat architecture
- JSR-82 Basics
Get started!

Get blucat source:

```
git clone https://github.com/ieee8023/blucat
```

or to just run blucat on mac:

```
brew install blucat
```

Get demo android app:

```
https://github.com/ieee8023/blucat-android-remote
```
USB Adaptors that work great!

AirCable Host XR3
http://www.aircable.net/products/host-xr3.php

Plugable BT4LE
http://plugable.com/products/usb-bt4le
Questions for you

How many of you have:

- Used a Bluetooth API?
- Used netcat to talk to a webserver?
- Created outrageously complex Bash scripts that involved piping?
STREAMS==AWESOME

1001011010110100100100101010110101101001001001001001001
STREAMS==AWESOME

You can send files or data
They even connect us all to PalTalk!
STREAMS==AWESOME

And it's all abstracted so each side just sees bits
You can abstract a really complicated process this way.
And then ignore how complicated and dysfunctional they are.
This works great for the TCP/IP

Why?

Let's look at HTTP

- It's so simple
- It's human readable
- Documentation isn't really necessary
- Debugging is easy
- You can encapsulate it
- You can customize it
$ nc -v mit.edu 80
Connection to mit.edu port 80 [tcp/http] succeeded!
GET / HTTP/1.1
Host: mit.edu

HTTP/1.1 302 Moved Temporarily
Server: AkamaiGHost
Content-Length: 0
Location: http://web.mit.edu/
Date: Mon, 09 Mar 2015 19:43:03 GMT
Connection: keep-alive
What is Blucat?

1. debugging tool for bluetooth applications
   a. connect to service for testing/emulation

2. device exploration tool
   a. reverse engineer existing services
   b. record nearby devices using scripts

3. a component in building other applications
   a. build applications on top of Blucat
with netcat

nc -l 123 | nc machine1 123
with blucat

| blucat -url btsspp://000000000CAFE:4 |

blucat -l 4 |
$nmap somehost
Starting Nmap 5.21 ( http://nmap.org )
Nmap scan report
Not shown: 846 closed ports, 152 filtered
PORT    STATE SERVICE
22/tcp  open   ssh
80/tcp  open   http
$blucat devices
#Searching for devices
+000000000CAFE, "The Engineer", Trusted:true, Encrypted:false
+123456789000, "Nexus 7", Trusted:true, Encrypted:false, -2
+012345678900, "GT-P1010", Trusted:false, Encrypted:false,
+001234567890, "Android Dev Phone 1", Trusted:true, Encrypted:
#Found 3 device(s)
$blucat services
#Listing all services
+,00000000CAFE, "The Engineer", Trusted:true, Encrypted:false
-,"OBEX Message Access E-Mail Server", ",", btgoep://0000000
-,"AV Remote Control Target", ",", bt12cap://00000000CAFE:00
-,"OBEX Phonebook Access Server", ",", btgoep://00000000CAFE
-,"Advanced Audio", ",", bt12cap://00000000CAFE:0019
-,"OBEX Object Push", ",", btgoep://00000000CAFE:12
-,"Android Network Access Point", ",", bt12cap://00000000CAF
-,"Headset Gateway", ",", btspp://00000000CAFE:2
-,"OBEX Message Access SMS/MMS Server", ",", btgoep://0000000
-,"Android Network User", ",", bt12cap://00000000CAFE:00ff
-,"Handsfree Gateway", ",", btspp://00000000CAFE:3
Scanning

$ ./blucat scan 00000000CAFE

#Scanning RFCOMM Channels 1-30

btspp://00000000CAFE:2 -> Open Channel!!! BluetoothRFComm

btspp://00000000CAFE:3 -> Open Channel!!! BluetoothRFComm

btspp://00000000CAFE:12 -> Open Channel!!! BluetoothRFComm

btspp://00000000CAFE:16 -> Open Channel!!! BluetoothRFComm

btspp://00000000CAFE:17 -> Open Channel!!! BluetoothRFComm

btspp://00000000CAFE:19 -> Open Channel!!! BluetoothRFComm

#Scanning L2CAP Channels 0-65000

bt12cap://00000000CAFE:1 -> Open Channel!!! BluetoothL2CAP

bt12cap://00000000CAFE:3 -> Open Channel!!! BluetoothL2CAP

bt12cap://00000000CAFE:17 -> Open Channel!!! BluetoothL2CAP

bt12cap://00000000CAFE:19 -> Open Channel!!! BluetoothL2CAP
Bluetooth URI Monikers

ex: btspp://10643FC98386:17
Bluetooth URI Monikers

btspp - Bluetooth serial port profile RFCOMM

btl2cap - Logical link control and adaptation protocol

btgoep - OBEX Generic Object Exchange profile
serial port profile (SPP)

- designed to emulate RS-232 serial ports
- same major attributes of TCP sockets
  - in order, retry,
- only allows ~30 ports
  - depends on stack
  - assigned dynamically like portmap (TCP/111)
link layer common access protocol (L2CAP)

- can make unreliable similar to UDP
- default maximum packet size is 672 bytes
- RFCOMM uses L2CAP as a transport
  - connects over L2CAP PSM #3
- more port numbers
  - aka PSM (Protocol Service Multiplexer) number
L2CAP in Bluetooth Protocol Architecture
<table>
<thead>
<tr>
<th>protocol</th>
<th>terminology</th>
<th>reserved/well-known ports</th>
<th>dynamically assigned ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>port</td>
<td>1-1024</td>
<td>1025-65535</td>
</tr>
<tr>
<td>UDP</td>
<td>port</td>
<td>1-1024</td>
<td>1025-65535</td>
</tr>
<tr>
<td>RFCOMM</td>
<td>channel</td>
<td>none</td>
<td>1-30</td>
</tr>
<tr>
<td>L2CAP</td>
<td>PSM</td>
<td>odd numbered 1-4095</td>
<td>odd numbered 4097 - 32765</td>
</tr>
</tbody>
</table>
MAC addresses can be looked up as normal!

http://standards.ieee.org/develop/regauth/oui/oui.txt
$.blucat -v -l -e /bin/bash
#Listening at btspp://002608AAAAA:4

$.blucat services
"BlueCatPipe","",btspp://002608AAAAA:4

$.blucat -url btspp://002608AAAAA:4 -v
#Connected
Hi
/bin/bash: line 1: Hi: command not found
Bluetooth pipefitting for -e
Inspection devices

Bluetooth has “profiles”

Identified by UUID and device class

Implemented by one or more services which may be RFCOMM or L2CAP
000C55F8FBEE, "Officejet 6300 series"

00-0C-55 (hex)  Microlink Communications Inc.
000C55 (base 16) Microlink Communications Inc.

8F, 31, Hsintai Road
Chupei City
Hsinchu 302
TAIWAN, PROVINCE OF CHINA
30F306AAAAAA, "Officejet 6300 series", Trusted:false, ...
"OBEX Object Push", "", btgoep://30F306598203:2
"Serial Port", "", btspp://30F306598203:1
"Basic Printing", "", btgoep://30F306598203:4
"Basic Imaging", "", btgoep://30F306598203:3
$./blucat -url btspp://30F306598203:1
Dear Sir,

Your serial port is showing.

# Connected

$ ./blucat -v -url btspp://30F306598203:1
"Serial Port0", "", btspp://9471ACDBACAD:11

9471ACAAAAAA, "Alcatel one touch 665A", ...
"AUDIO Gateway", "", btspp://9471ACDBACAD:1
"OBEX Object Push", "", btgoep://9471ACDBACAD:4
"Serial Port0", "", btspp://9471ACDBACAD:11
"Dial-up Networking", "", btspp://9471ACDBACAD:9
"Voice gateway", "", btspp://9471ACDBACAD:2
$ ./blucat -url btspp://9471ACAAAAAA:11
AT+CGMI
+CGMI: Alcatel
OK

AT+CGMM
+CGMM: one touch 665A
OK

AT+CGMR
+CGMR: Alcatel 010 04, 2012/03/05 14:56
OK
More AT Hayes Commands?

https://github.com/boos/bluesnarfer/blob/master/src/bluesnarfer.c

http://www.forensicswiki.org/wiki/AT_Commands

http://www.anotherurl.com/library/at_test.htm

http://gatling.ikk.sztaki.hu/~kissg/gsm/at+c.html
$ blucat services
#Listing all services
+,"001B7A2879AA", "Nintendo RVL-CNT-01", Trusted:false, Encrypted:false, NA
-,"", ",", null
-,"Nintendo RVL-CNT-01", ",", btl2cap://001B7A2879AA:0011
-,"", ",", null

$ blucat scan 001B7A2879AA
#Scanning RFCOMM Channels 1-30
#Scanning L2CAP Channels 0-65000
btl2cap://001B7A2879AA:1 -> Open Channel!!!
btl2cap://001B7A2879AA:11 -> Open Channel!!!
btl2cap://001B7A2879AA:13 -> Open Channel!!!
$ ./blucat services
#Listing all services
+,00000000CAFE, "The Engineer", Trusted:true, Encrypted:false, NA
-,"OBEX Message Access SMS/MMS Server", ",", btgoep://00000000CAFE:16
-,"OBEX Phonebook Access Server", ",", btgoep://00000000CAFE:19
-,"OBEX Object Push", ",", btgoep://00000000CAFE:12
-,"Headset Gateway", ",", btspp://00000000CAFE:2
-,"OBEX Message Access E-Mail Server", ",", btgoep://00000000CAFE:17
-,"Handsfree Gateway", ",", btspp://00000000CAFE:3
Handsfree Gateway, btspp://00000000CAFE:3

$ ./blucat -url btspp://00000000CAFE:3 -v

# Waiting for connection
# Connected
AT
AT+

ERROR
AT*

# Error: Connection is closed
## Hands-Free Profile

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+BLDN</td>
<td>Redials the previously dialed number.</td>
</tr>
<tr>
<td>AT+BRSF</td>
<td>Retrieves the supported features.</td>
</tr>
<tr>
<td>AT+BVRA</td>
<td>Enables or disables voice recognition in the AG.</td>
</tr>
<tr>
<td>AT+CCWA</td>
<td>Enables call waiting notification in the AG.</td>
</tr>
<tr>
<td>AT+CHUP</td>
<td>Rejects an incoming call.</td>
</tr>
<tr>
<td>AT+CIND?</td>
<td>Reads the current status of the AG indicators.</td>
</tr>
<tr>
<td>AT+CIND=?</td>
<td>Retrieves the indicator mappings for the AG.</td>
</tr>
<tr>
<td>AT+CLIP</td>
<td>Enables the call line identification.</td>
</tr>
<tr>
<td>AT+CMER</td>
<td>Registers or unregisters status updates.</td>
</tr>
<tr>
<td>AT+VGM=&lt;gain&gt;</td>
<td>Notifies the AG service when the microphone volume on the headset is changed to the specified gain value.</td>
</tr>
<tr>
<td>AT+VGS=&lt;gain&gt;</td>
<td>Notifies the AG service when the speaker volume on the headset is changed to the specified gain value.</td>
</tr>
<tr>
<td>AT+VTS</td>
<td>Transmits DTMF codes to the network.</td>
</tr>
<tr>
<td>ATA</td>
<td>Receives an incoming call.</td>
</tr>
<tr>
<td>ATD&gt;nnn</td>
<td>Dials a number in memory.</td>
</tr>
<tr>
<td>ATDdd...dd</td>
<td>Dials a number.</td>
</tr>
</tbody>
</table>
What works

AT+CNUM
"16175555555",129,,4

AT+CIND=?
("call", (0,1)), ("callsetup", (0-3)), ("service", (0-1)), ("signal", (0-5)), ("roam", (0,1)), ("battchg", (0-5)), ("callheld", (0-2))

ATD55555555555 // dials number
Rapid prototyping with Blucat
How to prototype

Current presentation is using blucat (maybe)

- Android app creates service
  - sends strings to whoever connects
  - “f” and “b” are wired to buttons
- Laptop runs blucat and pipes it into script
- Script dispatches “f” and “b” to press left and right keys
- https://github.com/ieee8023/blucat-android-remote
Launch blucat and pipe to dispatcher

blucat -k -v -url btspp://0000000000CAFE:4 -e "/bin/bash $(pwd)/dispatcher.sh"
Dispatcher reads input

while read input
do
    if [[ "$input" == *"f"* ]]; then
        echo "Forward"
        sh key-mac.sh 124
    fi
fi

...
Java based

Uses BlueCove Java Libraries

Works on Mac and Linux
State of the code
https://github.com/ieee8023/blucat

blucat
- BlucatClient.java
- BlucatConnection.java
- BlucatServer.java
- BlucatState.java
- BlucatStreams.java
- BluCatUtil.java
- ListServices.java
- Main.java
- PairUtil.java
- PrintUtil.java
- RemoteDeviceDiscovery.java
- ScanServices.java

com.intel.bluetooth
- MicroeditionConnector.java
- PairUtil.java

compression
- CompressedBlockInputStream.java
- CompressedBlockOutputStream.java

> lib

- github-machaval
  - bluecove-2.1.0.jar
  - bluecove-2.1.1-SNAPSHOT-jarias.jar
  - bluecove-2.1.1-SNAPSHOT-ma-ku.jar
  - bluecove-2.1.1-SNAPSHOT-r63-sources-all.zip
  - bluecove-2.1.1-SNAPSHOT-r63-sources.zip
  - bluecove-2.1.1-SNAPSHOT-r63.jar
  - bluecove-2.1.1-SNAPSHOT-r64-sources.jar
  - bluecove-2.1.1-SNAPSHOT-r64.jar
  - bluecove-bluez-2.1.1-SNAPSHOT-r63-sources.tar.gz
  - bluecove-bluez-2.1.1-SNAPSHOT-r63.jar
  - bluecove-emu-2.1.1-SNAPSHOT-r63-sources.tar.gz
  - bluecove-emu-2.1.1-SNAPSHOT-r63.jar
  - bluecove-gpl-2.1.0.jar
  - bluecove-gpl-2.1.1-SNAPSHOT-r63-sources.tar.gz
  - bluecove-gpl-2.1.1-SNAPSHOT-r63.jar
  - bluecove.zip
  - bluecovegpl.zip
- commons-io-2.4-sources.jar
- commons-io-2.4.jar
- IOBluetooth
Running blucat

```
$./blucat

if [[ $OSTYPE == *darwin* ]]; then
    LIBS=build/blucat.jar:lib/bluecove-2.1.1-SNAPSHOT.jar
...
elif [[ $OSTYPE == *linux* ]]; then
    if [[ $MACH == *arm* ]]; then
        LIBS=$DIR/...
    else
        LIBS=$DIR/...
    fi
fi
fi
java -cp $COMMONLIBS:$LIBS blucat.Main $@ 2> >(grep --line-buffered -v NSAutoreleaseNoPool >&2)
```
==Somewhere in the program:
System.loadLibrary("bluecove");
// Searched for file
// libbluecove.so
// in LD_LIBRARY_PATH

==BluetoothStackBlueZ.java:
private native
int rfServerGetChannelIDImpl(long handle) throws
IOException;

==Some C file
JNIEXPORT void JNICALL
Java_bluecove_rfServerGetChannelIDImpl(JNIEnv *env, jobject obj, jlong handle){ ....
}
### JSR-82 Basics

**LocalDevice**
Portal to adaptor

**RemoteDevice**
Represents paired, discovered, and connected devices

**Connection**
Access to Streams. "Notifier" version is server.

**UUID**
Identifies service and "profile."

**ServiceDiscovery**
Access to run SDP on nearby devices.

**ServiceRecord**
Advertisement of service. UUID, Channel, Name.
Example

1. Make a connection to a device in Java
2. Echo messages to console
3. Send messages to the device
String url = "btspp://50B0FA2C2AAE:4";
StreamConnection con =
    Connector.open(url,
    Connector.READ_WRITE,
    true);
InputStream is =
    con.openDataInputStream();
OutputStream os =
    con.openDataOutputStream();
BufferedReader in =
    new BufferedReader(
    new InputStreamReader(is));

String line = null;
while((line = in.readLine()) != null) {
    System.out.println(line);
}
Speaker

Joseph Paul Cohen
Email: jeocohen@cs.umb.edu
National Science Foundation Graduate Fellow
Ph.D Candidate - Computer Science
University of Massachusetts Boston

blucat
http://blucat.sf.net
https://github.com/ieee8023/blucat