TCP/IP has tools such as nmap and netcat to explore devices and create socket connections. Bluetooth has sockets but doesn't have the same tools. Blucat fills this need for the Bluetooth realm. Blucat can be thought of as a:

1. debugging tool for bluetooth applications
2. device exploration tool
3. a component in building other applications

Blucat is designed to run on many different platforms (including Raspberry Pi) by abstracting core logic from native code using the Bluecove library to interact with a variety of Bluetooth stacks. This talk will go over the objectives, designs, and current results of the project. More information is at http://blucat.sourceforge.net/
Joseph Paul Cohen (ieee8023)

Joseph is a Ph.D. student at the University of Massachusetts Boston. He has worked for large finance, IT consulting, and startup software companies. He now focuses on computer science research in areas of machine learning and cyber security education.
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?
Overview

- Streams
- blucat inline netcat replacements
- blucat as Bluetooth nmap
- rfcomm and l2cap basics
- look at some devices
- how to prototype
- scanning stats
- blucat architecture
STREAMS==AWESOME

100101101011010010010010101001001001001010
You can send files or data
They even connect us all to PalTalk!
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
GET / HTTP/1.1
Host: defcon.org

HTTP/1.1 200 OK
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
X-Content-Security-Policy: default-src 'self'
Strict-Transport-Security: max-age=16070400; includeSubDomains
Server: lighttpd
Cache-Control: public, max-age=600
Content-Language: en
Connection: keep-alive
Date: Mon, 15 Jul 2013 02:53:06 GMT
Last-Modified: Mon, 15 Jul 2013 01:36:50 GMT
Content-Type: text/html
Vary: Accept-Encoding
Transfer-Encoding: chunked

...site
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 btsp://00000000CAFE:4

blucat -l 4 |
with nmap

$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
+00000000CAFE, "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://00000000CAFE:01
-,"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://00000000CAFE
-,"Headset Gateway", ",", btsp://00000000CAFE:2
-,"OBEX Message Access SMS/MMS Server", ",", btgoep://00000000CAFE
-,"Android Network User", ",", bt12cap://00000000CAFE:000f
-,"Handsfree Gateway", ",", btsp://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
btl2cap://00000000CAFE:1 -> Open Channel!!! BluetoothL2CAP
btl2cap://00000000CAFE:3 -> Open Channel!!! BluetoothL2CAP
btl2cap://00000000CAFE:17 -> Open Channel!!! BluetoothL2CAP
btl2cap://00000000CAFE:19 -> Open Channel!!! BluetoothL2CAP
Defcon 2013 (Thursday-Saturday) visible bluetooth devices
Defcon 2013 Bluetooth Statistics

$sort names | uniq | wc -l
  92

$ cat bdaddr | sort | uniq | wc -l
  126

$ cat pairingrequests | wc -l
  1367
Best Bluetooth Device Names @DC13

hackbook
-INFECTED
HyperNerd-Mobile
DOD
SensordroneE344
cybertron
tOuch-mE-5G
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
L2CAP in Bluetooth Protocol Architecture
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
<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
On connect execution!

```
$./blucat -v -l -e /bin/bash
#Listening at btspp://002608AAAAAA:4
```

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

```
$./blucat -url btspp://002608AAAAAA:4 -v
#Connected
Hi
/bin/bash: line 1: Hi: command not found
```
Bluetooth

Plumbing
Bluetooth pipefitting for -e
Inspecting devices

Bluetooth has “profiles”

Identified by UUID and device class

Implemented by one or more services which may be RFCOMM or L2CAP
30F306AAAAAAA, "Officejet 6300 series", Trusted:false, ...
"OBEX Object Push", ",", btgoep://30F306598203:2
"Serial Port", ",", btsspp://30F306598203:1
"Basic Printing", ",", btgoep://30F306598203:4
"Basic Imaging", ",", btgoep://30F306598203:3
Dear Sir,

Your serial port is showing.
Serial Port0", "", btspp://9471ACDBACAD:11
$ ./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 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
+,001B7A2879AA, "Nintendo RVL-CNT-01", Trusted:false, Encrypted:false, NA
-,",", ",", null
-,"Nintendo RVL-CNT-01", ",", btl2cap://001B7A2879AA:0011
-,",", ",", null

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", btssp://00000000CAFE:3

$ ./blucat -url btssp://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;n.nn</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))
IhPone iAP service

- "Wireless iAP", "", btspp://34C059AAAAAAAA:1

Explored with Alex Whittemore @DC13

Goal to play/stop/control audio and tracks

Should be the same as interacting with standard UART in wire Apple connector
## iPod Accessory Protocol

<table>
<thead>
<tr>
<th>Field</th>
<th>Size</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>2</td>
<td>0xff 0x55</td>
</tr>
<tr>
<td>Length</td>
<td>1</td>
<td>Size of Mode + Command + Parameter</td>
</tr>
<tr>
<td>Mode</td>
<td>1</td>
<td>The mode the command is referring to.</td>
</tr>
<tr>
<td>Command</td>
<td>2</td>
<td>The two bite command.</td>
</tr>
<tr>
<td>Parameter</td>
<td>0..n</td>
<td>Optional parameter, depending on the command.</td>
</tr>
<tr>
<td>Checksum</td>
<td>1</td>
<td>0x100 - ( (sum of all length/mode/command/parameter bytes) &amp; 0xFF)</td>
</tr>
</tbody>
</table>

[Source](https://nuxx.net/wiki/Apple_Accessory_Protocol)
Ihpone iAP service
iPod Accessory Protocol

Speaking the protocol only made the ihpone say “This accessory is not supported”

“...establishing Bluetooth data connections with Apple devices requires a unique discovery/pairing sequence and negotiation with the Apple authentication co-processor”

Soo, this service requires a special chip from apple
Rapid prototyping with Blucat
How to prototype

Current presentation is using blucat

- 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
Launch blucat and pipe to dispatcher

blucat -k -v -url btspp://00000000CAFE: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

...
Scanning every 5 minutes from fixed location

Bluetooth devices are set to visible

blucat outputs in csv format
file = "logs.csv"
data = read.csv(file=file, header=T, row.names=NULL);

library(zoo)
dailyscans = as.Date(as.POSIXct(data[,2]/1000, origin="1970-01-01"))

hist(dailyscans, breaks=100, freq=T)
So now we have some data...
march = dailyscans
[dailyscans>="2013-03-01"] [dailyscans<"2013-4-01"]
Histogram of Joseph's scans

Frequency

Feb 27  Mar 05  Mar 11  Mar 17  Mar 23  Mar 29  Apr 04  Apr 10  Apr 16

daily scans
Java based

Uses BlueCove Java Libraries

Tested on Mac and many Linux versions using Bluez
State of the code
svn co http://svn.code.sf.net/p/blucat/code/trunk/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

java -cp $COMMONLIBS:$LIBS blucat.Main $@ 2> >(grep --line-buffered -v NSAutoreleaseNoPool >&2)
Java Native Interface

==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){...}
Thanks!

http://josephpcohen.com
http://blucat.sf.net