USBProxy

Building an Open and Affordable
USB Man in the Middle Device

Dominic Spill
dominicgs@gmail.com

Dominic Spill

USBProxy developer

Work on Ubertooth, BTBB, gr-bluetooth, Daisho, Unambiguous Encapsulation

Side projects include fcc.io, BeagleDancer, PS/2 tap

Adam Stasiak

USBProxy developer

.Net port of FaceDancer codebase

controllingxbox.blogspot.com

Background

2 billion USB devices sold each year (2008)

Most common device interface

Low / Full / High / Super speed

SuperSpeed Plus coming soon

Background

Huge surface for security assessment

- Host drivers
- Device firmware
- Connection between

I build tools for packet sniffing/injection

- Seems like USB might be fun to try

Note: Commercial Solutions are Available

Note: Commercial Solutions are Available

However, I'm going to ignore them because:

OPEN SOURCE!!!!!

FaceDancer

'It's not a bus, it's a network' - Sergey Bratus

Designed by Travis Goodspeed

- An extension to the GoodFET
- Prototype devices in Python
- Some great examples
- NCC Group 'umap' tool

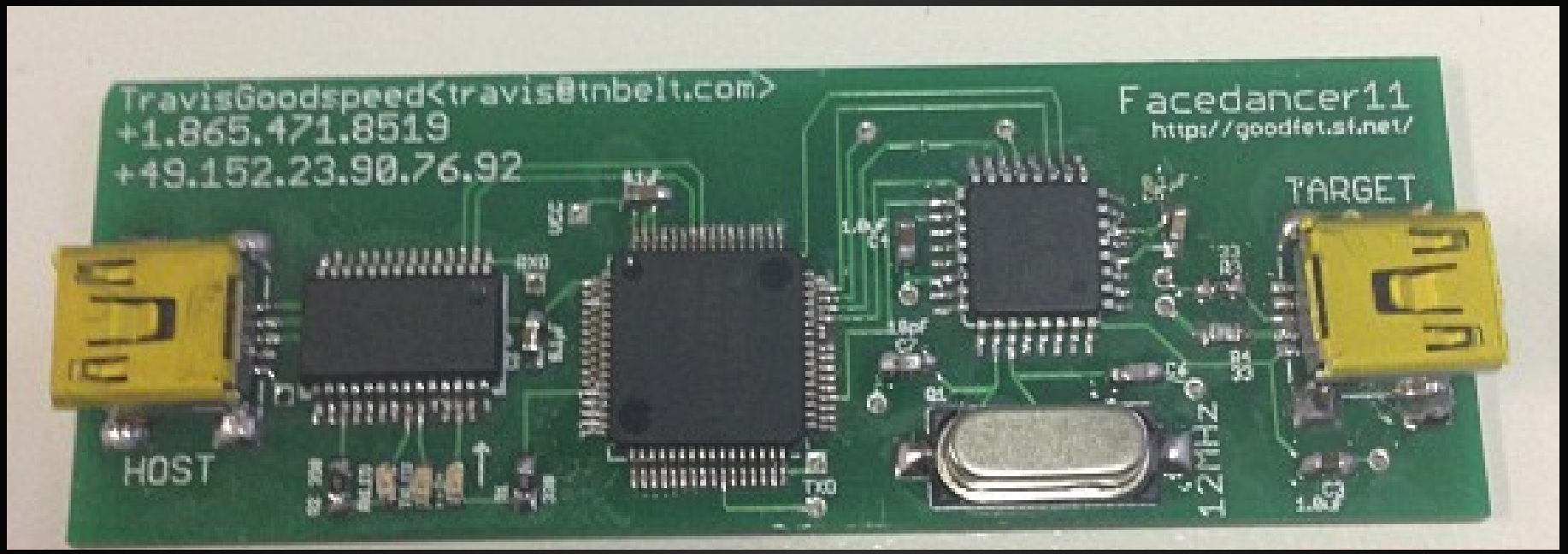
FaceDancer


Facedancer21
USB Device Emulator:
Mass Storage
FTDI Serial
HP49G+ Serial
Firmware Update (DFU)
Keyboard (HID)
http://goodfet.sf.net/
44CON
HOST
TARGET
12MHz
TravisGoodspeed<travis@tnbelt.com>
+1.865.471.8519
+49.152.23.90.76.92

FaceDancer Drawbacks

Speed / latency / cost

Soldering skills?



RaspDancer / BeagleDancer

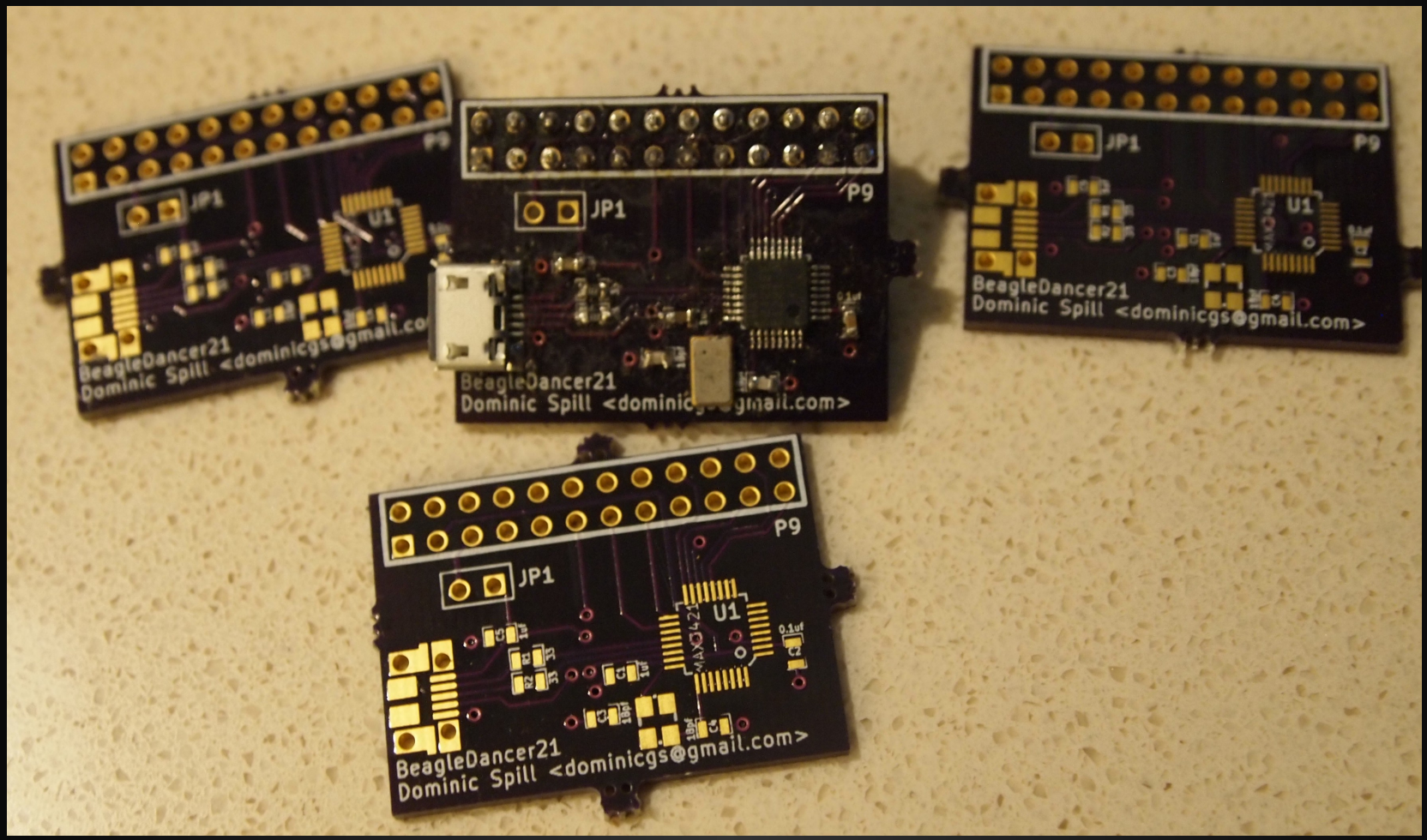
RaspDancer – Phillipe Teuwen

Increases SPI to 26MHz

Fewer parts – cheaper

BeagleDancer – Dominic Spill

BeagleDancer



BeagleDancer is pointless

MAX3421 USB OTG IC

BBB has OTG built in to TI am3359

<dominicgs> I made a FaceDancer for the BBB!

<mossmann> You know the BBB can probably do FaceDancer-y things anyway?

USB Sniffer

Nicholas Boichat

- GSOC 2010
- Kernel module to relay USB traffic
- Sniff packets with USBMon

Unmaintained

- 2.6.x kernel
- Want to do more than sniff

Should I write my own kernel module?

Library of gadgets with kernel

- Ethernet, storage, serial

Surely it can't be too hard to write one

Maintenance may be an issue

GadgetFS

Filesystem interface to OTG device

Written by David Brownell in 2003

- Stable / in mainline kernel
- Maintenance not my problem

GadgetFS

Filesystem interface to OTG device

Written by David Brownell in 2003

- Stable / in mainline kernel
- Maintenance not my problem

… apart from our patch to GadgetFS

usb-mitm was born

Simple C tool

- Using libusb and gadgetfs
- Only worked for Ubertooth

Based on examples from linux-usb.org

~1000 hacky lines

USBProxy

Open source C++ framework

Flexible / extensible architecture

Built upon:

- GadgetFS
- libUSB
- BeagleBone Black

BeagleBone Black

Cheap / powerful

Built in USB OTG interface

Open source hardware

Demo

USB Packet Sniffing

USBProxy Structure

Relayer moves packets around

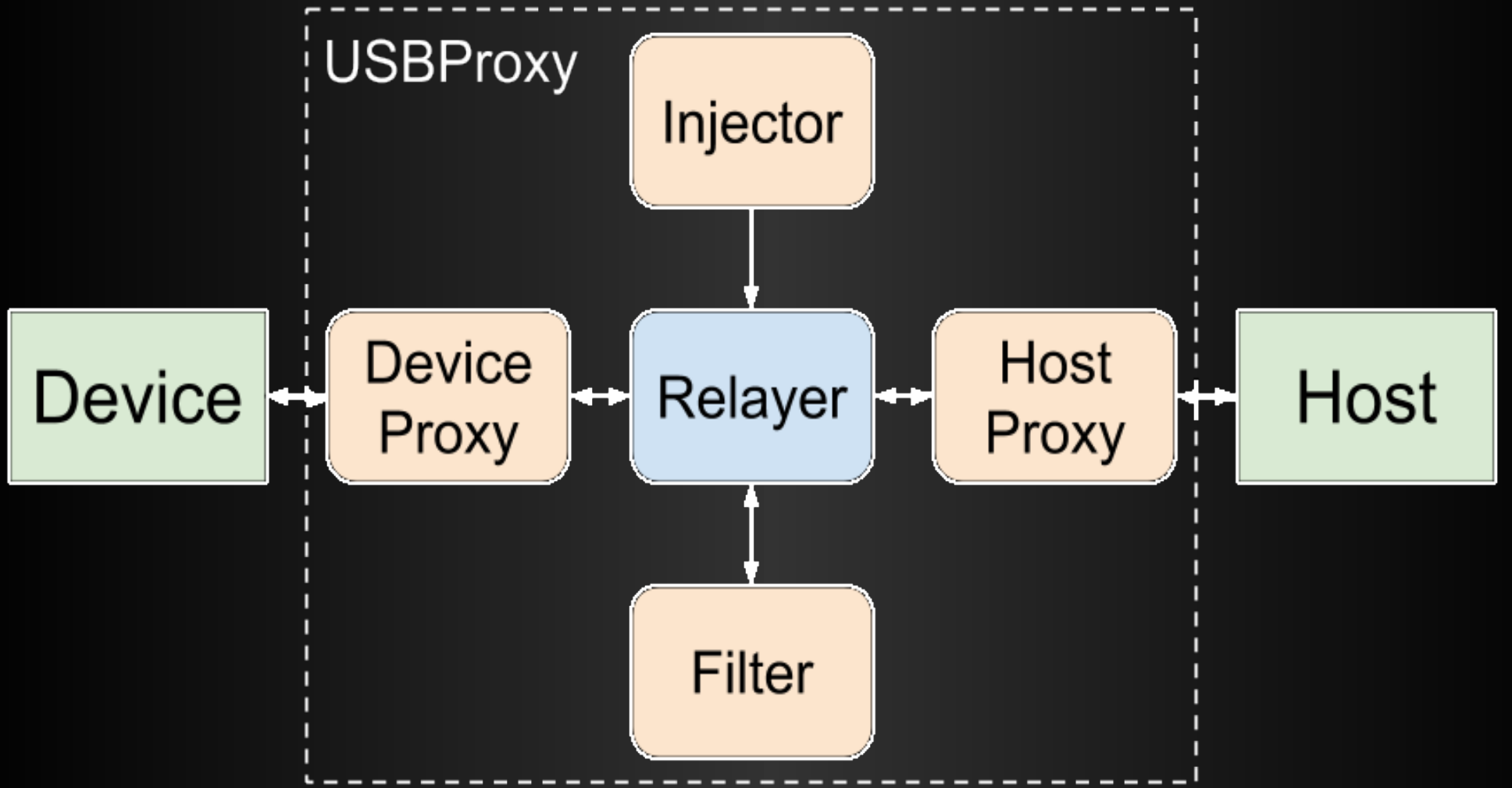
Manager handles setup and teardown

Proxies for talking to devices and hosts

Filters for modifying packets

Injectors for injecting arbitrary packets

USBProxy Structure



Filters

Log packets in transit

Modify packet data

Drop packets entirely

Demo

Packet Filters

Writing Filters

PacketFilter()

filter_packet(Packet* packet)

filter_setup_packet(SetupPacket* packet, bool direction_out)

PacketFilter_ROT13

```
void PacketFilter_ROT13::filter_packet(Packet* packet) {
	int i;
	for (i=2;i<8;i++) {
		if (packet->data[i]<=0x1d && packet->data[i]>=0x04) {
			if(packet->data[i]<=0x10)
				packet->data[i]=packet->data[i]+13;
			else
				packet->data[i]=packet->data[i]-13;
		}
	}
}
```

Writing Injectors

Injector()

void start_injector()

void stop_injector()

int* get_pollable_fds()

void full_pipe(Packet* p)

void get_packets(Packet, SetupPacket, timeout)

Demo

Injection

Proxies

Two flavours

- Host – connects to host (GadgetFS)
- Device – connects to device (libUSB)

Can use other libraries or technologies

- TCP host and device proxies

Demo

TCP Host/Device Proxies
(almost certain to fail)

Writing Proxies

int connect(timeout)

void disconnect()

void reset()

bool is_connected()

bool is_highspeed()

int control_request(setup_packet, nbytes, dataptr, timeout)

void send_data(endpoint, attributes, maxPacketSize, dataptr, length)

void receive_data(endpoint, attributes, maxPacketSize, dataptr, length, timeout)

void setConfig(fs_cfg, hs_cfg, hs)

Device proxies can be devices

Define descriptors

Implement read and write functions

That's it!

Loopback Device

Simple test device

- Write data and read it back

send_data() adds packet to buffer

receive_data() func returns first packet from buffer

Loopback device

```
void send_data(__u8 endpoint,__u8 attributes, __u16 maxPacketSize, __u8* dataptr, int length) {
   if(head==tail && full)
      return; // Buffer is full, silently drop data

   struct pkt next = buffer[tail];
   next.length = length;
   next.data = (__u8 *) malloc(length);
   memcpy(next.data, dataptr, length);
   free(buffer[tail].data);
   buffer[tail].data=NULL;
   tail = (tail + 1) % BUF_LEN;
   full = (head==tail);
}

void receive_data(__u8 endpoint,__u8 attributes, __u16 maxPacketSize, __u8** dataptr, int* length, int timeout) {
   if(head==tail && !full) {
      // No packet data (could wait for timeout to see if data arrives)
      *length = 0;
   } else {
      struct pkt next = buffer[head];
      *dataptr = next.data;
      *length = next.length;
      head = (head + 1) % BUF_LEN;
      full = !(head==tail);
   }
}
```

Drawbacks of Curret Setup

- BBB ports are USB 2.0 (480Mb/s)
  - Have yet to test throughput of USBProxy
  - Assume it is lower than line speed
- One device and one host port
  - Both support OTG
  - Support alternative network setups
- 100Mb/s ethernet
  - Chip supports 1000Mb/s

Future Work

- USB 3
  - Requires suitable device interface
  - Most likely Daisho
- Shared library
- Configuration parser

Language Bindings

- Starting with Python
  - Some initial filter code exists
  - Nothing to demo yet
- Plan for more languages
  - Looking for volunteers to help write them

FaceDancer Compatibility

Waiting on Python device proxies

Hope to work with existing software tools

Some slight differences in design, but should work

Thanks

Adam Stasiak

Travis Goodspeed & Sergey Bratus

Phillipe Teuwen & Nicholas Boichat

Michael Ossmann

Questions

github.com/dominicgs/USBProxy

#USBProxy on freenode

@dominicgs / dominicgs@gmail.com

Read the paper!