Bluetooth Proxy

Final report

|  |  |
| --- | --- |
| Written by | |
| Avishay Shasha |  |
| Itay Zach | 301227963 |

# Contents

[1 Abstract 3](#_Toc430967837)

[2 Architecture 4](#_Toc430967838)

[2.1 Terms And Definitions 4](#_Toc430967839)

[2.2 Overview 4](#_Toc430967840)

[2.3 Hooking Wrapper Using MS Detours 6](#_Toc430967841)

[2.3.1 Blocks Diagram 6](#_Toc430967842)

[2.3.2 Windows Socket API (WSA) and Hooking 6](#_Toc430967843)

[2.3.3 MS Detours with DLL Injection 7](#_Toc430967844)

[2.4 Bluetooth Proxy Application 8](#_Toc430967845)

[2.4.1 Blocks Diagram 8](#_Toc430967846)

[3 Issues We Came Across During The Project 9](#_Toc430967847)

[4 Conclusions 10](#_Toc430967848)

[5 References 11](#_Toc430967849)

# Table of Figures

[Figure 1 - Given system 4](#_Toc430967763)

[Figure 2 - BT proxy complete system 5](#_Toc430967764)

[Figure 3 - Hooking wrapper blocks diagram 6](#_Toc430967765)

[Figure 4 - BT Proxy app blocks diagram 8](#_Toc430967766)

# Abstract

Today’s world is all about communication and instant activities.

Bluetooth is a well-known protocol many devices and application are based on. Bluetooth has many advantages that allows almost every day usage – data can be transferred between two stand-alone devices with no need of a mainframe or a large system.

But the main problem with Bluetooth is the short distance limitation. So what if you could run an already existing application without the distance limitation?

The “Bluetooth proxy” aims to solve the distance limitation problem by taking over the Windows Bluetooth API functions and send the data over TCP/IP to an Android device that will be used as the Bluetooth Proxy.

# Architecture

## Terms And Definitions

|  |  |
| --- | --- |
| Term | Definition |
| WSA | Windows Socket API |
| BT | Bluetooth |
| DLL | Dynamic Link Library |
|  |  |
|  |  |
|  |  |

## Overview

Bluetooth Proxy is meant to take control over a given Bluetooth system BT functions in order to transmit the data over TCP/IP.

Let us say a doctor has to configure a BT based hearing device for the patient, the hearing device (and the patient) have to come to the doctor’s clinic for the configuration to be done.



Figure 1 - Given system

The idea is to expand the given system to the following:



Figure 2 - BT proxy complete system

The system consists of two major building blocks:

* Hooking wrapper using Microsoft Detours
* BT Proxy Android Application

## Hooking Wrapper Using MS Detours

### Blocks Diagram



Figure 3 - Hooking wrapper (relies on Windows side) blocks diagram

### Windows Socket API (WSA) and Hooking

Windows Socket API allows different protocols (e.g. TCP, BT, etc.) to be used with the same API functions. In a Server-Client system, a Server will open a socket using socket() and will wait for clients to connect using accept(), while the client will connect to the server using connect() function, and will probably send data using send() function.

Regardless of the protocol being used, data will be transmitted and received with the exact same functions.

The idea behind API hooking is to “hijack” a given API function by committing a user-defined one that has the same signature. That way, because the WSA functions are used in the original application, we could apply the same function but with TCP arguments instead of BT arguments.

### MS Detours with DLL Injection

MS Detours is a tool that allows such hooking to be done easily. From the MS Detours overview:

"Detours is a library for intercepting arbitrary Win32 binary functions on x86 machines. Interception code is applied dynamically at runtime. Detours replaces the first few instructions of the target function with an unconditional jump to the user-provided detour function. Instructions from the target function are placed in a trampoline. The address of the trampoline is placed in a target pointer. The detour function can either replace the target function, or extend its semantics by invoking the target function as a subroutine through the target pointer to the trampoline."

There are several ways to use this tool (for example, changing the address of the original function to the hooked one), but we decided to use the DLL injection method.

Using DetourCreateProcessWithDllEx() function, the original process will be executed, and during run-time, the user-defined DLL containing the hooked functions will be injected to the running process. That way, when the original application runs the WSA with BT arguments will actually run the WSA with TCP arguments.

## Bluetooth Proxy Application

### Blocks Diagram



Figure 4 - BT Proxy app blocks diagram

### TCP Server and BT Client

The BT Proxy app purpose is to be a tube between the given Windows application and the BT device. The communication from the Hooking Wrapper is done by connecting as a client in the Windows side to a TCP server in the BT Proxy side.

After the TCP Server receives a TCP packet, it transmits it as a BT packet through the BT client. The packet will be received at the BT original application as it should function as a BT Server.

# Issues We Came Across During The Project

Sockets are controlled by the OS. A bug made the socket to remain opened what caused the program not to work

# Conclusions

1. Learning by “getting dirty”
   1. Detour a simple process (MessageBox)
2. Plan the system before you code
3. Plan the schedule before you code
   1. Have defined milestones
4. Work flow division to blocks

# References