Bluetooth Command and Control Channel

Heloise Pieterse[a] and Martin S. Olivier[b]

[a]Defence, Peace, Safety and Security Unit, Council of Scientific and Industrial Research, Pretoria, South Africa, PO BOX 395, Pretoria 0001, South Africa, Email: hpieterse@csir.co.za, Tel: +27 12 841 2110.
[b]Department of Computer Science, University of Pretoria, Pretoria, South Africa, University of Pretoria, Private Bag X20, Hatfield, 0028, South, Email: molivier@cs.up.ac.za.

Abstract

Bluetooth is popular technology for short-range communications and is incorporated in mobile devices such as smartphones, tablet computers and laptops. Vulnerabilities associated with Bluetooth technology led to improved security measures surrounding Bluetooth connections. Besides the improvement in security features, Bluetooth technology is still plagued by vulnerability exploits. This paper explores the development of a physical Bluetooth C&C channel, moving beyond previous research that mostly relied on simulations. In order to develop a physical channel, certain requirements must be fulfilled and specific aspects regarding Bluetooth technology must be taken into consideration. To measure performance, the newly designed Bluetooth C&C channel is executed in a controlled environment using the Android operating system as a development platform. The results show that a physical Bluetooth C&C channel is indeed possible and the paper concludes by identifying potential strengths and weaknesses of the new channel.

Keywords: mobile; applications; command and control; Bluetooth; Android; malware

1. Introduction

Bluetooth is a radio frequency technology using the unlicensed 2.5 GHz industrial, scientific and medical (ISM) band (Sairam et al., 2002) It is an open standard for wireless connectivity, mostly found in mobile devices (smartphones, tablet computers, laptops) to enable short-range communications and replace proprietary cables. The aim and purpose of Bluetooth technology is to offer universal low cost and user friendly communication. Since the earlier developments of Bluetooth technology, vulnerability exploits have plagued this technology. Well-known vulnerabilities include eavesdropping and impersonation, both which led to various attacks such as Denial of Service (DoS), relay attacks and the creation of Backdoors (Potter 2004). These vulnerabilities led to the improvement of Bluetooth security by including authorisation, authentication and encryption (Sun et al., 2001). Besides the security improvements, Bluetooth technology is still being exploited.

Vulnerabilities in Bluetooth technology allow for the development of Bluetooth Command and Control (C&C) channels. A Bluetooth C&C channel is a communication channel responsible for delivery data across a Bluetooth connection between two Bluetooth-enabled devices. Such a channel can allow for faster communication by automating the authentication and authorisation process. Thus, when two devices come within range, a bond will be created between the devices and data transfer can begin without requiring any user involvement. This is ideal for devices that must quickly share information when coming within range, such as mobile botnets. Previous research by Singh et al. (2010) and Hua and Sakurai (2012) explored the potential of Bluetooth C&C channels but relied on simulations when constructing the C&C channels. This paper explores the development of a physical Bluetooth C&C channel, moving beyond previous research that relied on simulations.

In order to develop a physical Bluetooth C&C channel certain requirements must be fulfilled and specific aspects regarding Bluetooth technology must be taken into consideration. There will be a focus on eliminating user involvement and automating authentication processes such as the pairing process. To measure performance of the newly designed C&C channel, execution takes place in a controlled environment by using the Android operating system (OS) as the development platform. To verify that communication is indeed taking place across the Bluetooth C&C channel, the Bluetooth packets are captured by using the Ubertooth One tool and are viewed in Wireshark. The results show that a physical Bluetooth C&C channel is constructed and capable of supporting communication.

The rest of the paper is structured as follows. Section 2 gives an overview of research relating to the development of Bluetooth C&C channels. Section 3 provides a brief introduction of Bluetooth technology, focussing on the history, pairing process and vulnerabilities of Bluetooth. Section 4 describes the requirements necessary for the development of a physical Bluetooth C&C channel. Section 5 provides a description of the development and initialisation of the Bluetooth C&C channel while Section 6 focuses on the communication and execution of the channel. Section 7 provides a discussion on the potential strengths and weaknesses, and Section 8 concludes the paper.

2. Related Research

Few articles describe the development of Bluetooth C&C channels. Singh et al. (2010) evaluated the suitability of Bluetooth as a possible C&C channel in the design of a mobile botnet. With the Bluetooth C&C structure, each mobile bot acts as a peer in the mobile botnet, listening for new commands and forwarding the commands to the other discovered bots. During the initial infection, the mobile bots registers a Universal Unique Identifier (UUID) in the service register present in the mobile device. This allows the mobile bot to be discovered by the other bots as they come within range. The mobile bot then waits for new incoming connections and when such a connection arrives the mobile bot establishes a two-way Bluetooth connection to allow communication to occur. An advantage of this Bluetooth C&C is that it prevents a defender from easily taking down the mobile botnet since the defender needs to be in range of the mobile bots when communication takes place. This may not always be possible due to the changing topology of the network. Even though Bluetooth provides the botmaster with a stealthy C&C channel, it requires the mobile bots to be within range of one another to successfully propagate the commands (Singh et al. 2010).

Hua and Sakurai (2012) focused on designing two separate mobile botnets using Short Message Service (SMS) messages and Bluetooth technology for command propagation. The SMS-based mobile botnet uses SMS messages to propagate the C&C messages by using a simple flooding algorithm. The proximity-based mobile botnet uses Bluetooth to forward the C&C messages and form the communication channel around seed nodes, which are selected based on their contact frequency with other infected nodes. When the number of contacted nodes within a specific time period exceeds a threshold, the device will recommend itself for the seed role. These nodes encounter other devices more frequently, allowing for quicker dissemination of commands. After multiple simulations, the authors proved that a uniform random graph is the most efficient topology for the SMS-based mobile botnet and that human mobility features can improve the command propagation of proximity-based mobile botnets (Hua and Sakurai 2012).

The proposed Bluetooth C&C channels by Singh et al. (2010) and Hua and Sakurai (2012) only used simulations and neither constructed a physical Bluetooth C&C channel to determine the feasibility of such a channel.

3. Bluetooth Technology

Bluetooth technology was invented in 1994 by L.M. Ericsson (Sairam et al., 2002). During the winter of 1998 Ericsson, Nokia, Intel, IBM and Toshiba further evolved the Bluetooth standard by establishing the Bluetooth Special Industry Group (SIG) (Bisdikian 2001). The last few years saw 3COM, Microsoft, Lucent and Motorola also started participating in SIG (Sairam et al., 2002). The goal behind SIG is to further improve Bluetooth technology as short-range, low cost and user-friendly connections among portable devices, allowing for ad-hoc connectivity (Bisdikian 2001}. These capabilities are possible due to the architectural design of the Bluetooth technology.

The Bluetooth architecture consists of a protocol stack that is divided, by the Bluetooth specification, into three distinct groups: transport protocol group, middleware protocol group and the application group (McDermott-Wells 2004). The transport protocol group consists of the following layers: radio, baseband, link manager and the Logical Link Control and Adaptation Protocol (L2CAP) (McDermott-Wells 2004). These layers allow Bluetooth devices to locate each other while managing the physical and logical links (McDermott-Wells 2004). The middleware protocol group includes third-party, industry-standard and Bluetooth-SIG protocols, allowing existing and new applications to operate over Bluetooth links (McDermott-Wells 2004). Some of the protocols found in the middleware protocol group are the Internet Protocol (IP), Transmission Control Protocol (TCP) and serial port emulator

(RFCOMM). The application group consists of the actual applications that use the Bluetooth links (McDermott-Wells 2004). The architecture of the Bluetooth technology leads to several advantages: publicly available, royalty free, replaces the use of cables, supports both voice and data, and uses an unregulated frequency band which is available around the world (McDermott-Wells 2004).

As with all other forms of technology, vulnerabilities are also found plaguing Bluetooth technology. Current security threats targeting Bluetooth technology are due to vulnerabilities allowing eavesdropping and impersonation. Eavesdropping allow an attacker to *listen* to messages being exchanged during the pairing of devices (Jakobsson and Wetzel, 2001). This is possible if there is no encryption on the application layer or if the attacker is able to impersonate a device (Jakobsson and Wetzel, 2001). Impersonation occurs when an attacker poses as a legitimate Bluetooth device, allowing access to unauthorised data. This often achieved by means of a relay attack. A relay attack, similar to a man-in-the-middle attack, occurs when adversary C communicates to victim A, posing as victim B and to B as victim A (Levi et al., 2004).

The above mentioned vulnerabilities called for an improvement of Bluetooth security. One of the steps taken by Bluetooth developers to improve the security surrounding Bluetooth connections is by including a process called pairing. The process of pairing refers to a trusted relationship between two devices which are formed by secret codes, also known as pins (Minar and Tarique, 2012). The purpose behind this process is to create a custom link key that will allow for secure communication between devices (Gehrmann and Nyber, 2001). The pairing process involves user interaction where the users are responsible for confirming the identity of the connecting devices. This level of user interaction also increases the security surrounding Bluetooth connections and can prevent unauthorised users from misusing Bluetooth technology.

The Secure Simple Pairing (SSP) protocol is the most widely used pairing protocol since the inclusion of the protocol in the Bluetooth Core specification version 2.1. The protocol specifies the necessary steps for two Bluetooth devices to establish a shared common link for subsequent secure communications (Phan and Mingard, 2012). The SSP protocol consists of the following six phases (Haataja and Toivanen, 2010):

- **Capabilities exchanged**: Devices that are pairing for the first time or are re-pairing, exchange their Input/Output (IO) capabilities in order to determine the appropriate association model to use for the pairing. In the case of mobile devices, which have access to displays and keyboards, the Numeric Comparison (NC) model is used.

- **Public key exchange**: The devices generate the public-private key pairs and the Diffie-Hellman key before exchanging the keys.

- **Authentication stage 1**: In the case of the NC model, a 6-digit number is displayed on both devices attempting to make a connection. The users are responsible for comparing and confirming the numbers before completing the pairing process. If the numbers are identical, the users select *yes* and the pairing can proceed.

- **Authentication stage 2**: The devices exchange the values, compare and verify their integrity.

- **Link key calculation**: The devices compute the link key using their own Bluetooth addresses, the previously create values, and the Diffie-Hellman key constructed during phase 2.

- **LMP authentication and encryption**: Creation of the encryption keys.

Once the devices successfully completed the pairing process, a bond is established between the devices and communication can proceed.

To develop a physical Bluetooth C&C channel, multiple aspects regarding the Bluetooth technology must be taken into consideration, which will be further explored in the following section.

4. Requirements for a Physical Bluetooth C&C Channel

To design a suitable Bluetooth C&C channel, there is an important aspect concerning Bluetooth that must be taken into consideration. Bluetooth, like any other electronic component, consumes battery power. If the Bluetooth is left on indefinitely, it will quickly consume the battery power of the mobile device. To minimize the consumption of the battery due to Bluetooth activities, the Bluetooth will only be active during specific periods of a day as well as only for a limited time period. These active periods are based on patterns of human mobility.

Human mobility patterns tend to have two useful properties: regularity and chaotic (Aviv et al., 2010). There are patterns that tend to be regular since people repeatedly visit the same places over and over again (Aviv et al., 2010). These patterns can become chaotic in the sense that an act of randomness can alter the regular pattern, causing a given person to contact different people over a specific time or visit different places (Aviv et al., 2010). The patterns that tend to be regular are divided into three periods of mobility: no mobility, low mobility and high mobility. These periods of mobility are defined according to Stability and Availability.

- No Mobility:
    - Stability: Stability is high with no changes in geographical positioning.
    - Availability: Active for long periods, during nightfall and early morning hours, when people are sleeping.
- Low Mobility:
    - Stability: Stability is moderate with infrequent changes in geographical positioning.
    - Availability: Active for moderate periods, during daily hours, when people are actively working.
- High Mobility:
    - Stability: Stability is low with frequent changes in geographical positioning.
    - Availability: Active for short periods, during morning hours and late afternoons as people travel to their required destinations.

During the period of no mobility, no changes occur in geographical positioning since humans are at home and usually sleeping during this period. To minimize battery consumption, the mobile device will not activate the Bluetooth C&C channel during this period due to the low contact frequency with other devices.

As stated previously, humans tend to follow regular patterns and visit the same locations at regular intervals. These places are possibly offices, homes, schools, colleges, etc. Movement of humans at these locations tend to be moderately stable as they remain in their offices or classrooms for long periods, thus allowing for low mobility. During this period, a specific person comes in close contact with many other people regularly and for substantial time periods.

Then there are periods when the mobility of humans is quite high and they move around frequently. During these periods of high mobility, a person comes into close contact with many different people but only for very short time intervals. Mobile devices will not be always able to construct the Bluetooth C&C channel during these periods, making communication as these time periods impractical. This will in return minimize the impact of the Bluetooth C&C channel on the battery consumption.

From the three periods of mobility described above, the period of low mobility provides the most stable time period for the longest available time and therefore the Bluetooth C&C channel will only be active during this particular period.

Besides the consumption of battery power, Bluetooth technology requires user involvement to pair two mobile devices. Users are required to enable the Bluetooth, authorise the connection and authenticate the connecting device. For the Bluetooth C&C channel to be successful, the channel must require no user involvement to succeed. Therefore, to exclude any user involvement, the Bluetooth C&C channel performs the above mentioned activities on behalf of the user. Once the mobile devices are paired, communication can proceed via Bluetooth without requiring any user involvement.

The requirements for a physical Bluetooth C&C channel are:

- Enable and disable Bluetooth automatically.
- Have knowledge of the Bluetooth MAC addresses of connecting devices.
- Eliminate user involvement (the user must not be requested to complete the pairing request).
- The devices must not be required to be in discoverable mode when establishing the Bluetooth C&C channel.

Meeting the above listed requirements will lead to the development of an effective and efficient communication channel.

5. Development and Initialisation of the Bluetooth C&C Channel

The development platform for the Bluetooth C&C channel is the Android OS. The Android OS was selected because of the openness of the OS design and the ease by which the OS can be customised, thus allowing for the manipulation of the Bluetooth adapter. To create the physical Bluetooth C&C channel between two Android devices, additional private Bluetooth Application Programming Interfaces (APIs) are required. These private Bluetooth APIs, *IBluetooth.aidl* and *IBluetoothCallback.aidl*, allow for the direct manipulation of the Bluetooth adapter and specific processes, such as the pairing process.

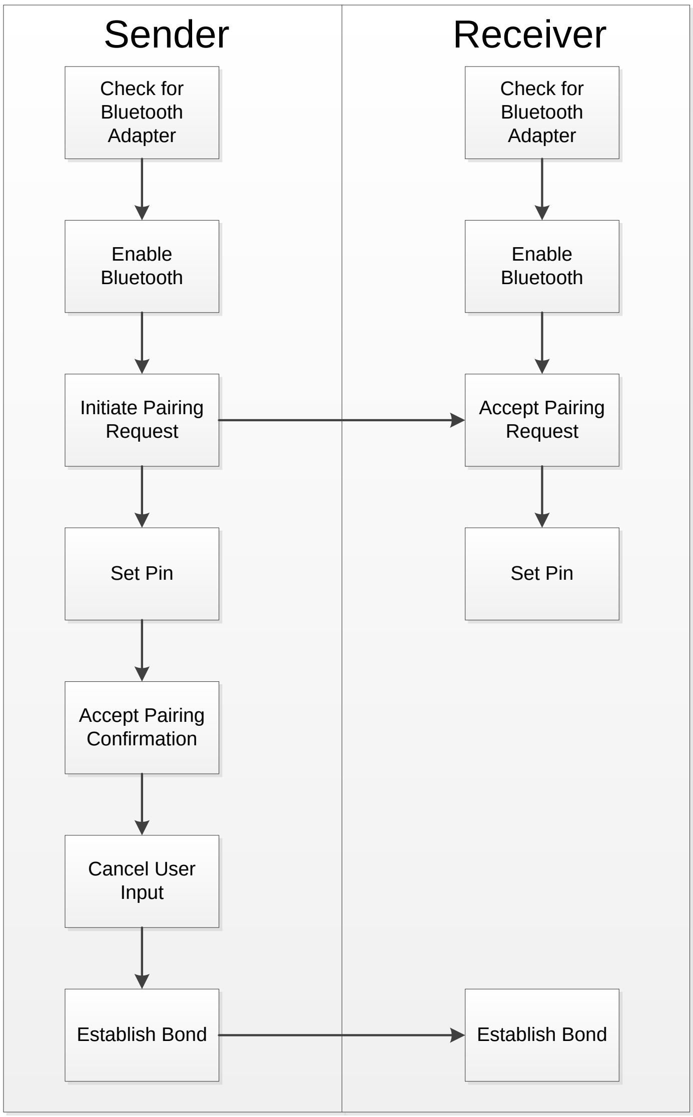


Figure 1: Pairing process as followed by the Bluetooth C&C channel

Figure 1 shows the pairing process as followed to establish the Bluetooth C&C channel. The pairing process must be automated, with only requiring the participation of two mobile devices and no user involvement. The first device, referred to as the Sender, will initialise the Bluetooth connection and perform the data transfer while the second device, referred to as the Receiver, will accept the Bluetooth connection and receive the data. The first step of initialising the Bluetooth C&C channel is to determine if both of the participating devices have a Bluetooth adapter. If both devices have access to a Bluetooth adapter, the devices enable the Bluetooth. To ensure a successful Bluetooth C&C is established, the Bluetooth of the Receiver must be enabled first before the Sender initiates the pairing process. As soon as the Sender comes within range of the Receiver, the pairing process begins. Both devices set the pin that will be used during this process. For the purpose of this demonstration the pin is set to 123456. The Sender is then responsible for the following two steps: setting the pairing confirmation to true and cancelling the requirement for user input. The first step allows the Sender to connect to the Receiver without requiring a user to press the confirmation button on either of the devices. The second step cancels the requirement for user input, remove the dialog from the screen and allow the pairing process to proceed without alerting the users of the devices. After successfully completing the pairing process, a bond will be created between the two devices.

By creating the bond between the two devices constructs the Bluetooth C&C channel. With the bond created, the devices will be able to start communicating.

6. Communication Across and Execution of the Bluetooth C&C Channel

This section focuses on the construction of communication that will allow data transfer to proceed and will also measure the execution of the channel, to determine if the physical construction of the Bluetooth C&C channel was successful.

6.1 Construction of Communication across the Channel

Once the bond between the devices is established, the initialisation of the Bluetooth C&C channel is completed and communication across the channel can occur. To allow for data transfer, the Radio Frequency Communication (RFCOMM) protocol is used. RFCOMM is a transport protocol that emulates serial connections and provides transport capabilities between Bluetooth-enabled devices (Panse and Kapoor, 2012). The RFCOMM protocol acts as a cable replacement protocol by emulating the RS-232 control and data signals over the Bluetooth baseband (Bruno et al., 2002). The Sender creates an insecure RFCOMM Bluetooth socket while the Receiver is listening on a previously created insecure RFCOMM Bluetooth socket. Insecure RFCOMM sockets are used to avoid any additional authentication. When the devices established the sockets, a communication channel is created via the RFCOMM protocol and data transfer can proceed. The Sender is responsible for sending the data to the Receiver and if required, the Receiver can respond to the received data. The data transfer can continue for as long as necessary but if the Bluetooth is left on indefinitely, the battery power of the two devices will quickly be consumed. Therefore, the bond created between the devices must be removed after completing the communication across the channel. By removing the bond dismantles the Bluetooth C&C channel between the two devices, leaving no traces of the communication on either of the two devices. If required further communication is required, the Bluetooth C&C can be constructed between the two devices again.

6.2 Measurement of Execution

To verify that the Bluetooth C&C channel is successfully constructed and that data transfer does indeed take place during the execution of the channel, the Ubertooth One development tool is used to capture the encoded packets. The Ubertooth One is an open source 2.4 GHz wireless development platform, designed for Bluetooth experiments (Project Ubertooth, 2013).

The architecture of the Ubertooth One is visualized in Figure 2. Once plugged into a computer (see Figure 3, the small and high powered USB dongle are ready to start capturing Bluetooth packets (Project Ubertooth, 2013).

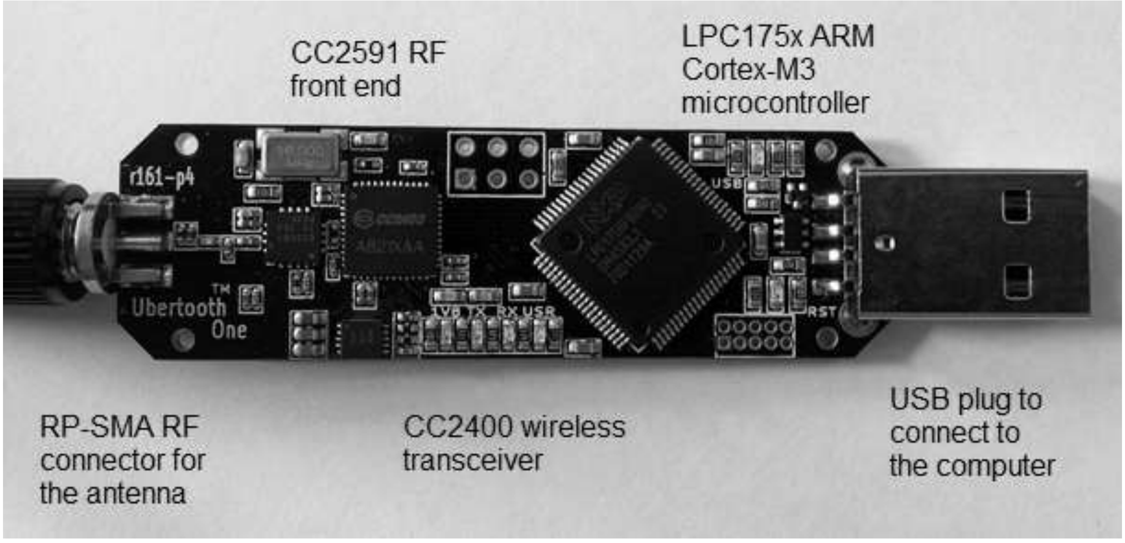

CC2591 RF front end
LPC175x ARM Cortex-M3 microcontroller
r161-p4
Ubertooth™ One
1V8 TX RX USR
USB
RST
RP-SMA RF connector for the antenna
CC2400 wireless transceiver
USB plug to connect to the computer

Figure 2: The Ubertooth One

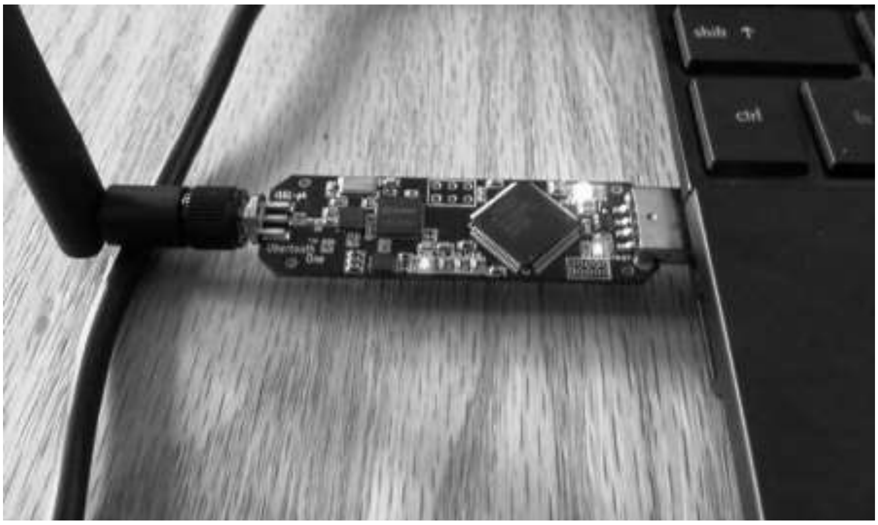
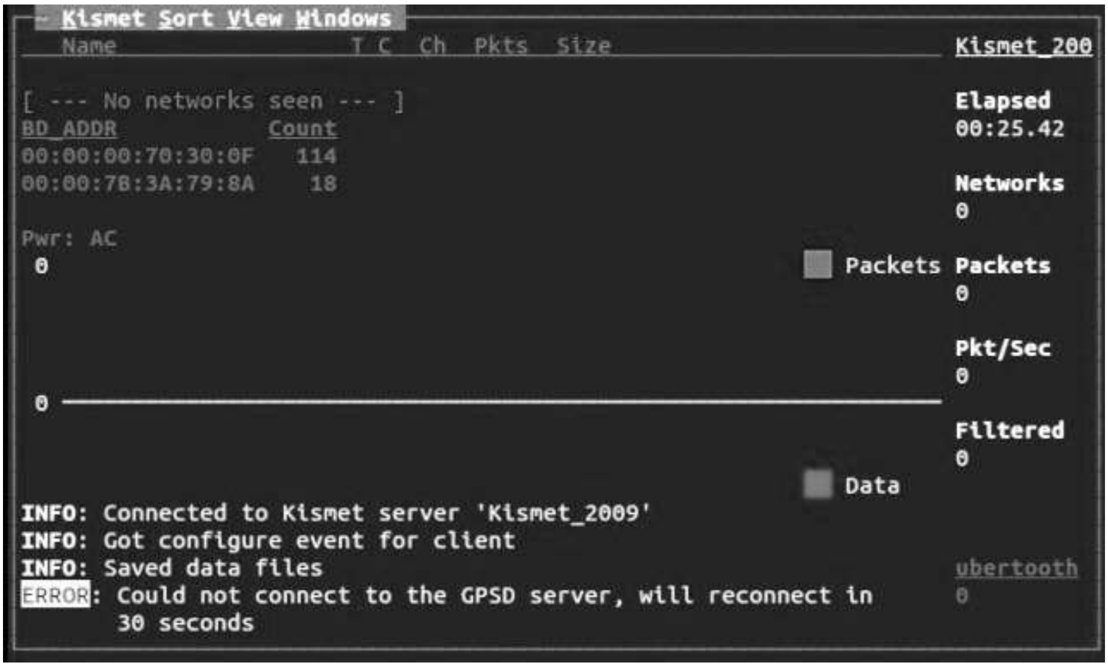


Figure 3: The Ubertooth One plugged into a computer

The Ubertooth One works closely with various wireless monitoring tools in order to visualise the captured packets. For the purpose of verifying the constructed Bluetooth C&C channel, the Kismet software is used. Kismet is an 802.11 layer 2 wireless network detector, sniffer and intrusion detection system (Kismet, 2013). Both the Ubertooth One tool and Kismet software were active while the Bluetooth C&C channel was executing. The Bluetooth packets, as captured using the Ubertooth One tool and the Kismet software, are shown in Figure 4.



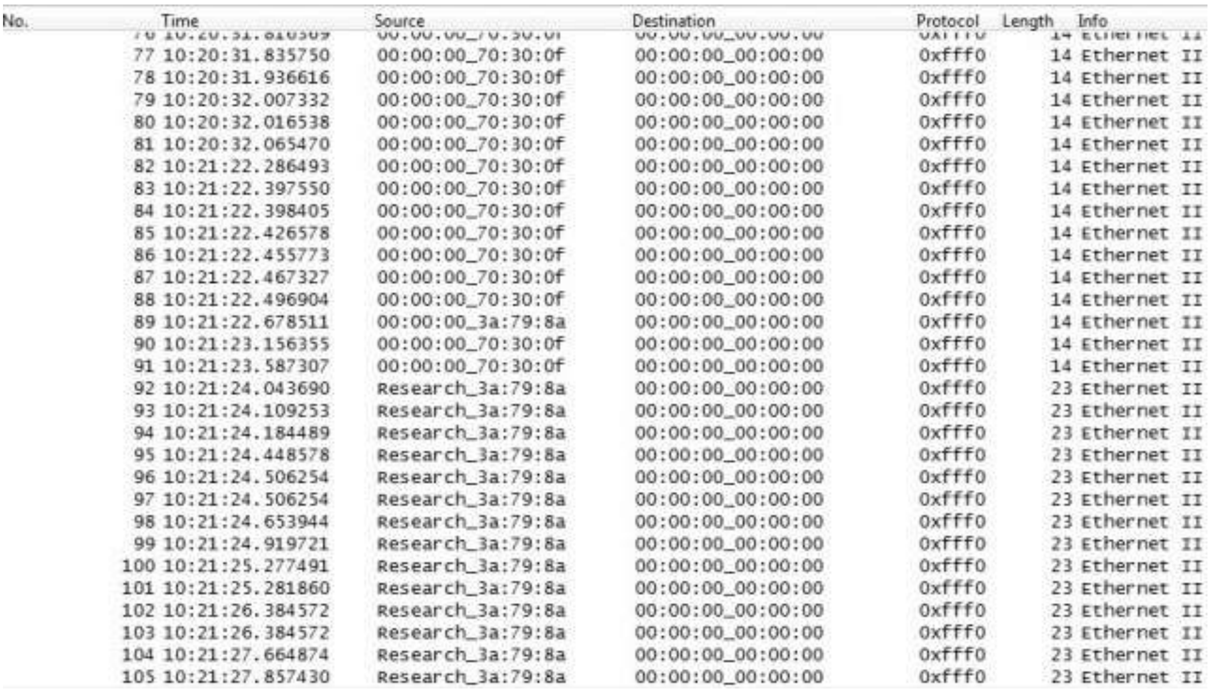
```
 ~ Kismet Sort View Windows
   Name                    T C  Ch  Pkts  Size                Kismet_200

[ --- No networks seen --- ]                                  Elapsed
BD_ADDR                Count                                  00:25.42
00:00:00:70:30:0F        114
00:00:7B:3A:79:8A         18                                  Networks
                                                              0
Pwr: AC
 0                                                  ▨ Packets Packets
                                                              0

                                                              Pkt/Sec
                                                              0
 0 ───────────────────────────────────────────────────────────
                                                              Filtered
                                                              0
                                                    ▨ Data
INFO: Connected to Kismet server 'Kismet_2009'
INFO: Got configure event for client
INFO: Saved data files                                        ubertooth
ERROR: Could not connect to the GPSD server, will reconnect in 0
       30 seconds
```

Figure 4: Screenshot of the Kismet software showing the captured Bluetooth packets

Figure 4 also shows that the following Bluetooth MAC addresses were participating while communication took place: 00:00:00:70:30:0F and 00:00:7B:3A:79:8A. The Bluetooth MAC address which is similar to a MAC address of a computer, consist of 48 bits (Hager and Midkiff, 2003). The 48 bits can further be divided into a 16 bit Non-significant Address Portion (NAP), a 8 bit Upper Address Portion (UAP) and a 24 bit Lower Address Portion (LAP) (Tariq et al., 2000). Both the Ubertooth One and Kismet are only capable of identifying the LAP address portions but in some instances can also identify the UAP address portions by analysing the timing and additional characteristics of multiple packets (Project Ubertooth, 2013).



No.	Time	Source	Destination	Protocol	Length	Info
76	10:20:31.816309	00:00:00_70:30:0f	00:00:00_00:00:00	0xfff0	14	Ethernet II
77	10:20:31.835750	00:00:00_70:30:0f	00:00:00_00:00:00	0xfff0	14	Ethernet II
78	10:20:31.936616	00:00:00_70:30:0f	00:00:00_00:00:00	0xfff0	14	Ethernet II
79	10:20:32.007332	00:00:00_70:30:0f	00:00:00_00:00:00	0xfff0	14	Ethernet II
80	10:20:32.016538	00:00:00_70:30:0f	00:00:00_00:00:00	0xfff0	14	Ethernet II
81	10:20:32.065470	00:00:00_70:30:0f	00:00:00_00:00:00	0xfff0	14	Ethernet II
82	10:21:22.286493	00:00:00_70:30:0f	00:00:00_00:00:00	0xfff0	14	Ethernet II
83	10:21:22.397550	00:00:00_70:30:0f	00:00:00_00:00:00	0xfff0	14	Ethernet II
84	10:21:22.398405	00:00:00_70:30:0f	00:00:00_00:00:00	0xfff0	14	Ethernet II
85	10:21:22.426578	00:00:00_70:30:0f	00:00:00_00:00:00	0xfff0	14	Ethernet II
86	10:21:22.455773	00:00:00_70:30:0f	00:00:00_00:00:00	0xfff0	14	Ethernet II
87	10:21:22.467327	00:00:00_70:30:0f	00:00:00_00:00:00	0xfff0	14	Ethernet II
88	10:21:22.496904	00:00:00_70:30:0f	00:00:00_00:00:00	0xfff0	14	Ethernet II
89	10:21:22.678511	00:00:00_3a:79:8a	00:00:00_00:00:00	0xfff0	14	Ethernet II
90	10:21:23.156355	00:00:00_70:30:0f	00:00:00_00:00:00	0xfff0	14	Ethernet II
91	10:21:23.587307	00:00:00_70:30:0f	00:00:00_00:00:00	0xfff0	14	Ethernet II
92	10:21:24.043690	Research_3a:79:8a	00:00:00_00:00:00	0xfff0	23	Ethernet II
93	10:21:24.109253	Research_3a:79:8a	00:00:00_00:00:00	0xfff0	23	Ethernet II
94	10:21:24.184489	Research_3a:79:8a	00:00:00_00:00:00	0xfff0	23	Ethernet II
95	10:21:24.448578	Research_3a:79:8a	00:00:00_00:00:00	0xfff0	23	Ethernet II
96	10:21:24.506254	Research_3a:79:8a	00:00:00_00:00:00	0xfff0	23	Ethernet II
97	10:21:24.506254	Research_3a:79:8a	00:00:00_00:00:00	0xfff0	23	Ethernet II
98	10:21:24.653944	Research_3a:79:8a	00:00:00_00:00:00	0xfff0	23	Ethernet II
99	10:21:24.919721	Research_3a:79:8a	00:00:00_00:00:00	0xfff0	23	Ethernet II
100	10:21:25.277491	Research_3a:79:8a	00:00:00_00:00:00	0xfff0	23	Ethernet II
101	10:21:25.281860	Research_3a:79:8a	00:00:00_00:00:00	0xfff0	23	Ethernet II
102	10:21:26.384572	Research_3a:79:8a	00:00:00_00:00:00	0xfff0	23	Ethernet II
103	10:21:26.384572	Research_3a:79:8a	00:00:00_00:00:00	0xfff0	23	Ethernet II
104	10:21:27.664874	Research_3a:79:8a	00:00:00_00:00:00	0xfff0	23	Ethernet II
105	10:21:27.857430	Research_3a:79:8a	00:00:00_00:00:00	0xfff0	23	Ethernet II

Figure 5: A snapshot of the captured packets viewed in Wireshark

A snapshot of the captured packets is shown in Figure 5, which were further analysed in Wireshark. Analysis of the captured packets shows that the devices participating in the Bluetooth C&C channel were active between 10:20 and 10:22. Most of the communication occurs between 10:21:22 and 10:21:27, showing that the construction, initialisation and communication occur within one minute. This short set-up of the Bluetooth C&C channel makes this channel very efficient and feasible where quick data transfer is required.

This snapshot of the captured Bluetooth packets verifies that the Bluetooth C&C channel was successfully constructed and active during a very short time period. Furthermore, of the two mobile devices participating in the communication it can be derived that the mobile device with the LAP of 70:30:0F is indeed a Receiver since this device activated its Bluetooth first. Thus the mobile device with the LAP of 3A:79:8A is the Sender that forwarded the command to the Receiver. The complete execution of the Bluetooth C&C are shown in Table 1.

Table 1: Log file captured during the execution of the Bluetooth C&C Channel

Timestamp	Participant	Data logged
07-17 12:42:48.328	Receiver	Received Bluetooth Event that can be ignored
07-17 12:43:08.063	Receiver	Enable Bluetooth
07-17 12:43:12.570	Sender	Enable Bluetooth
07-17 12:43:14.228	Receiver	Received Bluetooth Event that can be ignored
07-17 12:43:17.103	Receiver	Start Bluetooth Connection
07-17 12:43:17.128	Receiver	Wait for Pairing Request to Start
07-17 12:43:17.173	Receiver	Received Bluetooth ON Event
07-17 12:43:20.367	Sender	Start Bluetooth Connection
07-17 12:43:20.367	Sender	Wait for Pairing Request to Start
07-17 12:43:25.164	Sender	Create Bond with Receiver
07-17 12:43:25.164	Sender	Set Pins
07-17 12:43:25.171	Sender	Set Pairing Confirmation to True
07-17 12:43:25.171	Sender	Cancel User Input for Pairing
07-17 12:43:25.783	Receiver	Set Pins
07-17 12:43:26.898	Sender	Send Data
07-17 12:43:26.937	Sender	Response From Receiver: Data Received
07-17 12:43:27.906	Sender	Removed Paired Device

The use of the Ubertooth One tool and the Kismet software allows for the analysis of the Bluetooth C&C channel, showing that this channel is indeed established and that data transfer can occur across it quickly and efficiently.

7. Discussion

The physical construction of the Bluetooth C&C channel was proved feasible and showed that efficient communication across this channel is possible. By using Bluetooth allows for the creation of a stealthy C&C that can easily evade detection, making the discovery of the channel difficult for defenders. Destruction of the physical Bluetooth C&C channel is challenging as it requires the defenders to be within close range of the devices to effectively detect and destroy this channel. This is indeed not always feasible, improving the secrecy by which the channel can operate. Also allowing for a stealthy implementation is the automation of the pairing process, eliminating user involvement and removing the requirement of being in discoverable mode. To effectively avoid detection after transferring data across the Bluetooth C&C channel, the bond between the two participating devices can be removed without the requiring any user involvement. The removal of the bond also removes any traces from the devices, hiding their participation in the Bluetooth C&C and so avoiding detection by defenders and the users of the devices. The physical constructed Bluetooth C&C provides thus a stealthy channel which can operate in secrecy without alerting the users of the devices.

Besides the secrecy offered by the Bluetooth C&C channel, other additional advantages are also provided by the channel. Firstly, Bluetooth is a popular technology and is found on most mobile devices available today. Thus the initialisation and construction of a Bluetooth C&C will always be a possibility. Secondly, unlike other C&C channels using SMS messages or the Internet, no cost is involved in the initialisation or implementation of the Bluetooth C&C channel. Finally, data transfer across the Bluetooth C&C channel is quick and also cost-free. These advantages show that physical Bluetooth C&C channels are feasible, capable of supporting communication and can effectively avoid detection.

The physical Bluetooth C&C channel is not without any weaknesses. Firstly, the initialisation and construction of the channel requires close proximity of the devices. If the devices are not within range or move out of range while attempting to construct the channel, the construction and communication of the channel will fail. Construction of the channel is thus only possible when the devices are closely clustered together for a certain time period. Secondly, due to the consumption of battery power by Bluetooth technology, the Bluetooth C&C channel is only capable of supporting short data transfers. Should the Bluetooth be left on indefinitely, the battery power of the device will completely drain, causing the device to shut down and destroy the C&C channel.

The weaknesses of the Bluetooth C&C channel however do not overshadow the strengths provided by this channel. Should devices be closely clustered together for lengthy time periods, the Bluetooth C&C channel can be initialised and constructed within seconds, allowing communication across the channel to proceed. Due to the secrecy and cost-effectiveness provided by the Bluetooth C&C channel, the channel is ideally suited for command dissemination within a mobile botnet. The Bluetooth C&C channel is however not limited to mobile botnets only, but can also be utilised in other environments where communication must quickly occur with minimal user involvement.

8. Conclusion

This paper explored the initialisation and construction of the Bluetooth C&C channel, focussing circumventing the pairing process and the establishment of communication between the mobile devices. The pairing process, as followed by the Bluetooth C&C channel, requires no user involvement and allows mobile devices to pair without being in a discoverable mode. By removing the bond between the mobile devices after completing the communication ensures that the establishment and the execution of the Bluetooth C&C channel remain hidden to the users of the mobile devices. To show that the design of the Bluetooth C&C channel is feasible as an actual C&C channel, a physical implementation of the channel was done by using the Android OS as the development platform. By using the Ubertooth One tool, the Bluetooth packets could be captured and visualised in Wireshark. The captured packets show that data transfer is indeed taking place across the channel and that it can efficiently support communication. The final results showed that the physical implementation of a Bluetooth C&C channel is possible, forms a stealthy channel and is capable of effectively avoiding detection. Future work will focus on implementing the Bluetooth C&C channel on other mobile platforms. In addition, other potential uses of the channel can also be evaluated. With Bluetooth being a well-established and popular technology, there is definite growth potential for Bluetooth C&C channels.

References

K.V.S.S.S.S. Sairam, N. Gunasekaran and S.R. Redd, "Bluetooth in wireless communication," Communications Magazine, IEEE, vol. 40, no. 6, pp. 90-96, 2002.

B. Potter, "Bluetooth vulnerabilities," Network security, vol. 2004, no. 3, pp. 4-5, 2004.

J. Sun, D. Howie, A. Koivisto and J.A.A.K.K.O Sauvola, "Design, implementation and evaluation of Bluetooth security," In Proceedings of the IEEE International Conference on Wireless LANs and Home Networks, pp. 121-130, 2001.

K. Singh, S. Sangal, N. Jain, P. Traynor and W. Lee, "Evaluating bluetooth as a medium for botnet command and control," In 7th International Conference on Detection of Intrusions, Malware and Vulnerability Assessment, pp. 61-80, 2010.

J. Hua and K. Sakurai, "Botnet Command and Control Based on Short Message Service and Human Mobility," Computer Networks: The International Journal of Computer and Telecommunications Networking, vol. 57, no. 2, pp. 579-597, 2012.

C. Bisdikian, "An overview of the Bluetooth wireless technology," Communications Magazine, IEEE, vol. 39, no. 12, pp. 86-94, 2001.

P. McDermott-Wells, "What is Bluetooth?," Potentials, IEEE, vol. 23, no. 5, pp. 33-35, 2004.

M. Jakobsson and S. Wetzel, "Security weaknesses in Bluetooth," In Proceedings of the 2001 Conference on Topics in Cryptology: The Cryptographer's Track at RSA, pp. 176-191, 2001.

A. Levi, E. Çetintaş, M. Aydos, Ç.K. Koç, and M.U. Çaglayan, "Relay attacks on Bluetooth authentication and solutions," In Computer and Information Sciences ISCIS, pp. 278-288, 2004.

N.B.I. Minar and M. Tarique, "Bluetooth Security Threats and Solutions: A Survey," International Journal of Distributed and Parallel Systems, vol. 3, no. 1, pp. 86-94, 2012.

C. Gehrmann and K. Nyber, "Enhancements of Bluetooth baseband security," Proceedings of Nordsec 2001, pp. 191-230, 2001.

R.C. Phan and P. Mingard, "Analyzing the Secure Simple Pairing in Bluetooth v4.0," Wireless Personal Communications, vol. 64, no. 4, pp. 719-737, 2012.

A.J. Aviv, M. Sherr, M. Blaze and J.M. Smith, "Moving Targets: Geographically routed human movement networks," Department of Computer & Information Science, University of Pennsylvania, 2010.

T. Panse and V. Kapoor, "A review on security mechanism of Bluetooth communications," International Journal of Computer Science and Information Technologies, vol. 3, no. 2, pp. 3419-3422, 2012.

R. Bruno, M. Conti and E. Gregori, "Bluetooth: Architecture, protocols and scheduling algorithms," Cluster Computing, vol. 5, no. 2, pp. 117-181, 2002.

Project Ubertooth, available at http://http://ubertooth.sourceforge.net/, June 2013.

Kismet, available at http://http://www.kismetwireless.net/, June 2013.

C.T. Hager and S.F. Midkiff, "Demonstrating vulnerabilities in Bluetooth Security," In Global Telecommunications Conference, pp. 1420-1424, 2003.

M.F. Tariq, P. Czerepinski, A. Nix, D. Bull and N. Canagarajah, "Robust and scalable marching pursuits video transmission using the Bluetooth air interface standard," IEEE Transactions on Consumer Electronics, vol. 45, no. 3, pp. 673-681, 2000.

K. Haataja and P. Toivanen, "Two Practical Man-In-The-Middle Attacks on Bluetooth Secure Simple Pairing and Countermeasures," IEEE Transactions on Wireless Communications, vol. 9, no. 1, 2010.

Vitae

Heloise Pieterse is an MSc student in the department of Computer Science at the University of Pretoria, South Africa. She is currently on a Studentship program at the Council of Scientific and Industrial Research and works within the Command, Control and Information Warfare research group. Research interests include information security and mobile devices.

Martin S. Olivier is a professor at the Department of Computer Science in the School of Information Technology at the University of Pretoria. In addition to normal teaching and research duties, he is the research coordinator of the School of Information Technology. His current research interests include privacy and digital forensics as well as database, application and system security. Research Activities are carried out in the Information and Computer Security Architecture Research Group at the University of Pretoria.