Bluetooth Security

Mousa Al-kfairy Shannon Ryke
Email: {moual274,shary268}@student.liu.se
Supervisor: David Byers, davby@ida.liu.se
Project Report for Information Security Course
Linköpings universitetet, Sweden

Abstract

In this project we talk about WPAN security threats and issues especially Bluetooth security. We studied Bluetooth structure, its history, specifications, old security threats, and security mechanisms. We then used different hacking tools to attempt to break into Bluetooth with limited success.

1. Introduction

In this paper WPAN (Wireless Personal Area Network) is examined from the security perspective. With a focus on Bluetooth looking at its security and currently existing attack methods.

Some specific implementations of the Bluetooth protocols, and also the Bluetooth architecture are looked at. This will include several attacks on Bluetooth networks using various tools and then examination of the results.

1.1 Paper Overview

After the introduction the paper will cover the different tools and methods the project will use to create and attack Bluetooth networks. The next section will examine the results of these attacks. Then a general conclusion of our finding's regarding Bluetooth Security.

1.2 Bluetooth General Definition

First a general description of Bluetooth best described in Wikipedia as, "Bluetooth is an industrial specification for wireless personal area networks (PANs). Bluetooth provides a way to connect and exchange information between devices such as mobile phones, laptops, personal computers, printers, GPS receivers, digital cameras.

1.3 Bluetooth History

The first specification of Bluetooth was developed by Jaap Haartsen and Sven Mattisson, Ericson mobile platform in Lund, Sweden. After that the SIG (Bluetooth special interest group) was founded for Bluetooth specifications and development.

Since Bluetooth's release many security issues where found by academic research groups these where reported to security companies and over time a lot of concern has gone into possible threats of Bluetooth devices. However no major attacks has happened and as of 2006 no new major vulnerabilities have arisen.

2. Background

2.1 Bluetooth Classes and Specifications

In the following table you can see the different classes of Bluetooth and how they differentiate between the sets of classes by power and range:

Class	Range
Class 1	~ 100 M
Class 2	~10 M
Class 3	~1 M

Bluetooth as any new technology has many different versions as time progress when problems are fixed, and new ideas implemented. The following describes the progression of Bluetooth.

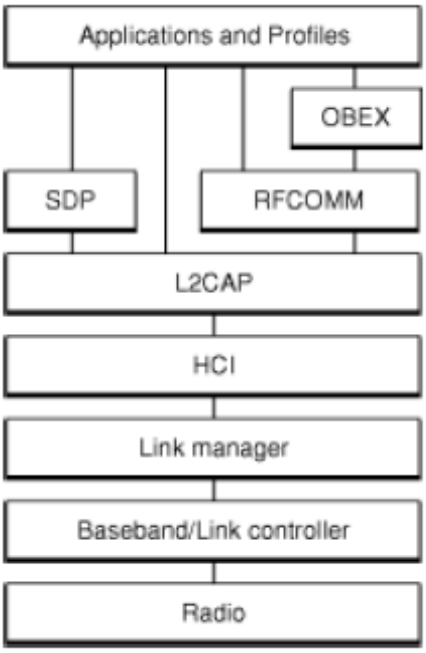
- Bluetooth 1.0 and 1.B: Is the first version of Bluetooth and it is considered as the base for Bluetooth, this version has many problems and mainly it is not compatible with a lot of manufactured products.
- Bluetooth 1.1: It the IEEE 802.15.1, many problems in the first version has been fixed and new specs were added for example the support for non-encrypted channels.
- Bluetooth 1.2: this is a version which is compatible with the first version and it adds some new features to the last one; one of the main new features is the speed of connection it becomes much faster compared to earlier versions. Another feature is a range extension.

- Bluetooth 2.0: it is also compatible with 1.1 and the main added features for this version are the Enhanced Data Rate (EDR) of 3.0 Mbit/s for both data (ACL) and voice (eSCO) packets.
- Bluetooth 2.1: this is the latest version of Bluetooth with a lot of functionality.

2.2 Theoretical Methods

2.2.1 Bluetooth Structure

The following figure shows the Bluetooth hierarchy stack which is considered as the heart of Bluetooth and the most important part within Bluetooth, its importance comes from the fact it gives Bluetooth compatibility and portability. In this part of the paper we will briefly discuss the main parts of this stack:



The layers from the bottom to the up:

Radio

It is simply like a physical layer in OSI, but with different functionality. Its main function are the modulation and demodulation of the data into RF, it also describes the physical characteristics in Bluetooth which are; modulation characteristics, radio frequency tolerance, and sensitivity level.

Baseband – link controller

Actually, there is no formal distinction between the baseband and link controller, but we will try to describe each one by the best description possible. The baseband portion is responsible for formatting the data from the Radio layer and the second responsibility for it is to handle the synchronization in the link. The second portion of this layer (Link controller) is responsible for carrying out the link manager's commands and establishing and maintaining the link stipulated by the link manager.

Link manager:

It is responsible for establishing and configuring links and managing power-change requests, among other tasks.

In Bluetooth there are two types of links which are:

- Connection oriented: it is a Synchronous communication that used for isochronous and voice communication.
- Connectionless oriented: Asynchronous communication which is used for exchanging data.

HCI (host controller interface layer)

This is the boundary layer which divides the lower and upper layers of the Bluetooth stack .and it is used to support Bluetooth systems that are that are implemented across two separate processors.

L2CAP (logical link control and adaptation protocol) layer:

This is the first layer in the upper layers portion of the Bluetooth hierarchy stack and it is responsible for :

- Establishing or requesting ACL (Synchronous connection oriented) connection if does not exist.
- Multiplexing different protocols in the higher levels to allow them use ACL.
- Repackaging the data packets it receives from the higher layers into the form expected by the lower layers.

SDP (service discovery protocol) :

Defines actions for both servers and clients of Bluetooth services.

RFCOMM layer

Emulates the serial cable line settings and status of an RS-232 serial port. RFCOMM connects to the lower layers of the Bluetooth protocol stack through the L2CAP layer.

OBEX (object exchange)

Is a transfer protocol that defines data objects and a communication protocol two devices can use to easily exchange those objects

2.2.2 Bluetooth security overview:

In any Bluetooth device there are always four main entities that manage the security these entities are:

- Bluetooth device address : a unique 48 bits address (every device has it is own address)
- Private authentication key: a 128 bit random key which is used in the authentication.
- Private encryption key: It varies between a 8 – 128 bit random key (there are two reasons why the key is not fixed .
- First, because of every country has its own requirement and cryptographic algorithm. Second, for flexibility in future upgrading to the algorithm).
- Random number: a 128 bit number generated by the Bluetooth device and frequently changes.

The encryption key in Bluetooth changes every time the encryption is activated, the authentication key depends on the running application to change the key or not. Another fact regarding the keys is that the encryption key is derived from the authentication key during the authentication process.

The time required to refresh the encryption key is 228 Bluetooth clocks which is equal to appr. 23 hours. RAND or the random number generator is used for generating the encryption and authentication key. Each device should have its own random number generator. It is used in pairing (the process of authentication by entering two PIN-codes) for passed keys in the authentication process.

Also, in Bluetooth there are three security modes which are:

- Mode 1: Non-secure.
- Mode 2: service level security.
    - Trusted device.
    - Un-trusted devices.
    - Unknown devices.
- Mode 3; link level.

The trusted device is a device that has been connected before, its link key is stored and it's flagged as a trusted device in the device database. The un-trusted devices are devices that have also previously connected and authenticated , link key is stored but they are not flagged as a trusted devices .The unknown devices are the devices that have not connected before .

In Bluetooth service level we have three type of service in regard to the security, these services are:

- Services that need authentication and authorization: this is automatically granted to the trusted devices but for the un-trusted devices manual authentication is required.
- Services that need authentication only: in this case the authorization process is not necessary.
- Open services.

2.2.3 Key management:

Link key:

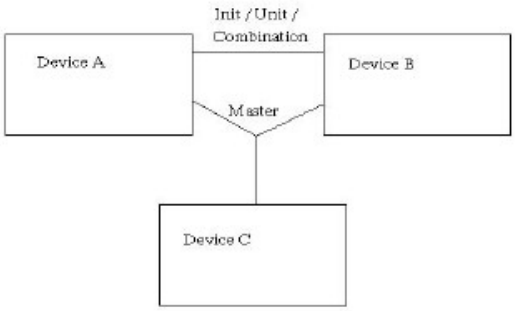
This is the main key used for authentication and it is a also an argument used for generating the encryption key which is a 128- bit random number shared between two or more devices .

To serve different type of applications different type of link key are used:

- Temporary key Kmaster : Used when the device wants to transmit to a number of devices (not a single device), it replaces the link key temporarily.
- Combination key Kab: generated by a pair of devices from the device information.
- Initialization key Kint: used when there is no unit key or combination key, replaces the link key and it is only used through the installation process.
- Unit key KA: generated when the single device is installed.

One of the main parts of the key management is the PIN code which varies from 1 to 16 octets. The PIN code requires a UI; in case of no UI a fixed PIN code is used like when your Bluetooth headset is connected to your mobile phone. The most secure method is to enter the PIN in both devices in order to establish a connection.

Key Generation process:



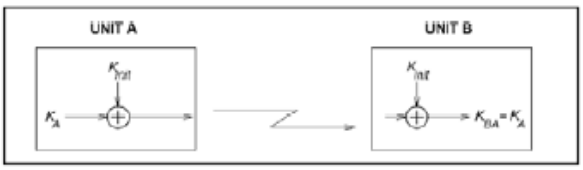
1

(This Picture is taken from http://web.archive.org/web/20060519034246/http://www.niksula.hut.fi/~jiitv/bluesec.html#chap5)

In order to establish a connection an initialization key should be generated, the initialization key is built by the E22 algorithm and uses Bluetooth device addresses, PIN codes and 128 – bit random number that is generated by the devices. In the case of one device having a fixed PIN code then the BD_ADDR of the other device shall be used instead. In the other case when the two devices both have fixed PINs then no communication is used. The initialization key is generated the first time that these devices are connected to each other.

Then, the initialization key is used through the process of exchanging the keys and generating the link key, then the initialization key deleted.

The unit key KA is generated once the device installed and operated in the first time , then the UK(unit key) is stored in a none – volatile memory and not changed . In case if UK is changed then a wrong LK (link key) is generated by the previously defined devices. There is only one device that uses its own key to generate the LK, typically and in most cases this is the device with restricted memory because it can only remember it is own key. The E21 is the algorithm used for generating the UK.



This picture is taken from (specification of Bluetooth system, velum 0)

From the above figure , notice that a device A , sends it own unit key to device B , device B uses this unit key as a link key for this connection A-B .

After exchanging the unit key, the initialization key will be discarded in both devices.

The combination key is generated through the initialization phase, and as mentioned before it is generated by the device information. The Combination key is generated by the following equations:

Device a: Keya=E21 (RANDa, BD_ADDRa)

Device b: Keyb=E22 (RANDb, BD_ADDRb)

Devices should securely exchange generated random numbers RANDa and RANDb, by the process done by LK using Xor function as the following:

$K \oplus RANDa$

$K \oplus RANDb$

After exchanging the keys the old link key shall be deleted from the two devices.

The E3 algorithm is used for generating encryption key from the 128-bit random number and 96-bit ciphering offset number (COF). There are two ways to generate COF. The first one is from the master BD_ADDR. The second one is when we have ACO (Authenticated ciphering offset, it is an auxiliary parameter that is generated when a successful authentication occurs). The encryption key changes every time either device enters the encryption mode.

Master key is a 128 – bit random number that is generated by two 128-bit random numbers and using E22 algorithm in this encryption. Then, a third random number is generated and sent to the slave device (it is a master-slave connection). Then the current LK overlay is computed by master and slave. The new link key is then sent to the slave, and calculated by the Xor function.

2.2.4 Encryption

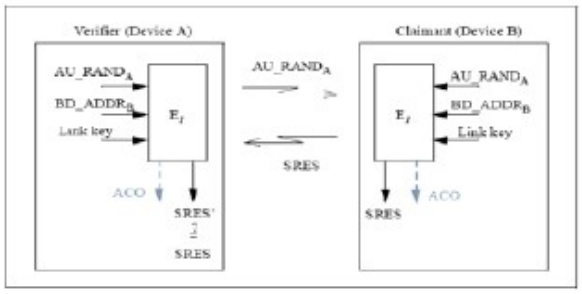
In the encryption of the Bluetooth traffic, Bluetooth uses the E0 encryption method. There are several encryption modes. The first one when we use unit key or combination key ,then the broadcast traffic is not encrypted , the addressed is either encrypted or not . The second one is when the master key used; in this case we have three modes which are:

- .No-encryption.
- Broadcast traffic is not encrypted, but the addresses are encrypted.
- All traffic is encrypted .

The size of encryption keys must be defined by a negotiation between devices. It varies from 8 to 128 bits.

2.2.5 Authentication

The authentication process is based on symmetric keys, so the two devices share the same key and use it in authentication process.



The figure you can see above shows how the authentication process takes place.

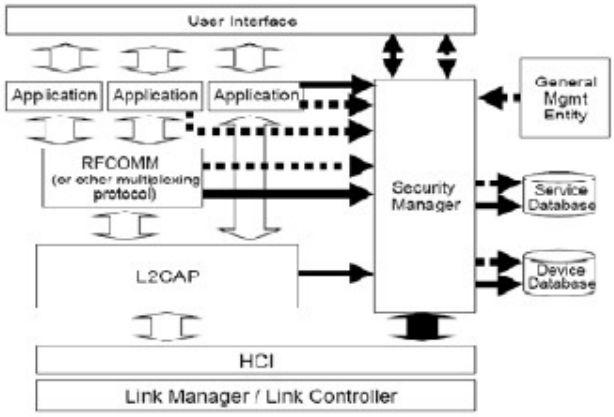
- The verifier sends a random number to the claimant to be authenticated.
- The claimant and the verifier use the E1 authentication function with a random number from BD_ADDR (claimant) and the current link key to get a response.
- The claimant sends the response to the verifier, who then makes sure the responses match.

This is in general how the authentication process occurs between two devices. But in some case there is only one way authentication, so only one device is authenticated.

In case of no-authentication occurred or fails, there is a time interval should pass till the next time and this time doubled in each try.

2.2.6 Security layers

In this part, we will discuss the function of each layer or stack party of Bluetooth stack architecture. We will use the following graph as a base of our discussion. Note: this graph is taken from "Bluetooth security architecture version 1.0, Thomas Muller ".



Bluetooth security architecture resides in L2CAP layer and above it. The only Bluetooth specific is RFCOMM, all other are not Bluetooth specific.

L2CAP: enforce security for cordless telephony.

RFCOMM: enforce security for Dial-up networking.

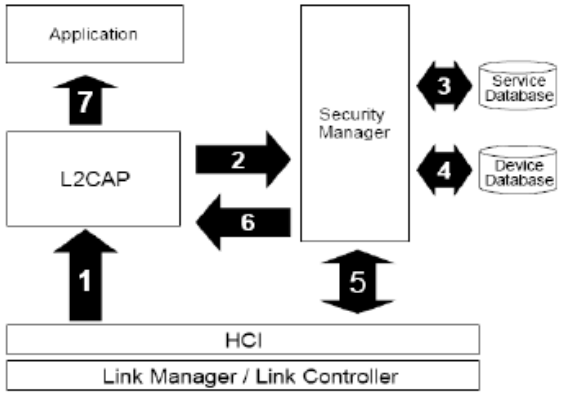
OBEX: files transfer and synchronization.

Establishing a connection (from the layers)

This part discusses how Bluetooth connection is established and how the operation passed from Bluetooth layers.

The first thing is defining the accessed service and which security level is related to this service, and then an authentication process will occur. The authentication process takes place only when a request to a service submitted.

We can summarize the authentication process as; first, a connection request to L2CAP, and L2CAP request access from the security manager. Then, the security manager looks in service and device DBs to determine if an authentication and encryption is needed or not. After granting the access by the security manager L2CAP continue to set up a connection.



Regarding the protocol stack, for any new connection request, the request submitted to L2CAP, in some cases also in RFCOMM for multiplexing, and then the protocol parameters are passed to the security manager for decision making. These parameters enter as query values to the security manager. Finally, the security manager according to it is query results; may either grant access or reject the access.

2.2.7 Security interfaces:

We have different security interfaces in different protocol stacks of Bluetooth, each one of these interfaces serve the security of Bluetooth with different functions.

We will start by briefly studying the databases provided by the security manager. In Bluetooth security manager there should be two types of databases,

In the first services database, each service should be stored in none-volatile memory or at the register at start up. For each service the database should store a set of

parameters such as, if it is a mandatory authorization or authentication, and if it is allowed to broadcast or not.

In Device database, for each device it stores, the trust level, LK, BD_ADDR, and the device name. If the entry is deleted a device will become unknown.

L2CAP should ask for the security manager for access rights for incoming and outgoing connections for a specific set of services.

In the security manager passed parameters are ProtocolоIdentification which identifies the protocol passed to the query ,PSM (the channel), BD_ADDR which is the Bluetooth device address of the remote computer, incoming connection (true in case of incoming and false for outgoing),connection handle. After performing the query, the security manager returns the value of access which is either granted or denied.

Other interfaces are connected to Bluetooth security structure, and we will stop by these interfaces, because almost all the other interfaces have the same procedure and structure.

3. Connection in practice

3.1 Connecting Devices:

This section will talk about connecting devices with each other via Bluetooth from the security view.

3.1.1 Connecting a Headset:

The headset architecture consists of two main devices the headset (HS) and the audio gateway (AG) . AG could be any other type of devices like laptop, cellular or any other device.

In order to build a secure connection between HS and AG, it is recommended to use the baseband authentication, encryption and to store the passkey and link key . Since the HS normally does not have UI , then the other device should control some settings of HS and take care of the security .

In HS it is feasible to keep a fixed passkey (PIN) , since it is hard by none-interface to change the passkey and it should be physically .

Connection procedure :

This is the steps of how the connection occurs:

- The user presses the HS button preparing it for pairing process.
- Open his mobile phone , turn on Bluetooth and start searching for Bluetooth devices .
- Mobile finds the HS by the querying process and getting response from HS.
- HS asks for authenticating the phone.
- If there is no previous connection between these two devices, a pairing process should occur.
- User enters the passkey, information exchange between devices, link key is generated, shared between devices and should be stored on a none-volatile memory.
- Authentication process, each device authenticates the other one.
- Exchanging encryption key so that all the traffic between two devices is encrypted.
- User switch the pairing process off on HS, so no new pairing process will be accepted.

From now on, the authentication process is based on exchanging the link key between devices.

4. Attack

4.1 Tools & Programs

- Hardware Used:Dell XPS, Nokia N95, Nokia 6150, Hp IPAQ HX2790b.
- Operating Systems: Ubuntu, Backtrack, Windows Vista, Symbian OS, windows mobile.
- Software Used:Bluebugger, Bluediving, Bluescanner,Bluesnarfer,BTscanner, Redfang, Blooover2, ftp_bt.
- Dell laptop with windows vista to be broken into and for scanning then with Linux to attempt attacks. Pocket pc for being attacked, and one mobile for attacking one for being attacked.

4.1.1 Types of Attack

Discovery

Scanning: Using tools to find the MAC address of nearby devices to attack. This generally finds devices set to discoverable although programs exist with a brute force approach that detects them when hidden. These programs also provide other basic information such as device classes and names. BTscanner can also suggest attacks the device will be vulnerable to.

4.1.2 Attacking Tools

- Bluejacking: Sending an unsolicited message over Bluetooth generally harmless but can be considered annoying at worst. Generally done by sending a vcard (electronic business card) to the phone and using the name field as the message.
- OBEX Push: A way of bypassing authentication by sending a file designed to be automatically accepted such as a vcard and instead using OBEX to forward a request for data or in some cases control. Used in the below attacks.

- Bluesnarfing: access to data on a device via Bluetooth such as text messages, contact lists, calendar, emails etc. This uses the OBEX push profile to attempt to send an OBEX GET command to retrieve known filenames such as telecom/pb.vcf. The enhancement to this Bluesnarf++ connects to the OBEX FTP server to transfer the files.
- HeloMoto: Full control of a device using AT commands. Either OBEX is used to create a connection is a Bluesnarf or a vcard card is sent and then the request is automatically cancelled leaving the attacking device as a trusted device in the target. This allows AT commands to be used.
- Bluebugging: Take control of the phone, make calls, and listen to calls etc anything a user can do. This attacks gains access to the mobile through the RFCOMM channel 17 which on certain phones is unsecured and can be used as a backdoor. Once connected AT commands are used to take control of the mobile.
- DOS Attacks: there are various attacks such as Bluesmack, Bluestab and in some cases Bluejacking that can be used to cause a DOS attack. This can range from using Bluejacking to repeatedly send messages to a phone that requires them to be accepted to using AT commands to crash to phone or malformed packets (ping of death). This can cause strange behavior in devices or they simply crash.

5. Bluetooth social engineering

Bluetooth is used by people daily so it is possible to use social engineering techniques to attack devices. One of the most common uses of Bluetooth is with Mobile Phone can be an interesting part of social engineering to examine.

According to Marek Bialoglowy in his research of Bluetooth security he mentioned these results by trying social engineering techniques “; I named my laptop Bluetooth dongle to PIN1234, 1234 or PASS1234 (in several different tests) and simply tried to connect to any discovered Bluetooth devices within the food court of one of the biggest malls in Jakarta. Benefiting on the 200m range of my equipment, I was able to discover from 3 to 11 Bluetooth devices during lunch time, and had tried to connect to each of them. Surprisingly, an average of 1 in 10 tries had my connection accepted. The phone users simply read "PIN1234" as the name of device which was trying to connect to his/her hand phone, and so the user types the 1234 PIN (passkey) to accept the connection”. The experiment is an excellent example of social engineering in Bluetooth. Some users tend to accept incoming connections leaving themselves at risk to outside attack. More a lack of education than anything else causes people not to recognize a threat when they see one and accept incoming connections. An interesting way of using social engineering to break into devices.

6. Results

6.1 Effectiveness of Attacks

Laptop

The attacks here where a resounding failure with all devices being attacked requiring user input to function. Bluebugging and Bluesnarfing where both attempted several times with trial and error the correct channels for these attacks where found and used to successfully contact the phone but failed to work without authentication. Attacks that succeeded without authentication gathered no data whatsoever.

Vs Mobiles

Attacks made against the Nokia N95 and Nokia 6250 both connected to the phone but required the user to accept to continue and thus where considered a failure. Attacks were also made against other nearby mobiles with either the same result or in a single case a successful transfer with Bluesnarfing but no data gathered. (Unusual filenames where assumed).

Vs Laptops

A single laptop with Bluetooth came into range and after asking the owner attacks where performed without success even when he decided to accept the connection.

Pocket PC

The software for this device was not compatible with its Bluetooth stack so no tests where possible.

Mobile

Vs Mobiles

The primary success was through this device and a program called blooover2. An auditing tool blooover2 tests the possible effect of various attacks and did a few minor attacks of its own. While the test devices required authentication for this audit to function passing devices showed several vulnerabilities and after hunting down owners and asking permission successful attacks where performed. The software inserted phonebook entry’s, copied phone books and changed call forwarding effectively taking phones off the network. The other program that had a single successful attack was called “Super Bluetooth attack” while the majority of phones required authentication a Sony Eriksson (model unknown) allowed access without. Phonebook, messages where accessible while calls could also be made and general settings changed (display, sounds etc).

6.2 Security Effectiveness

The standard security method for Bluetooth is to simple have the device hidden or turned off and many devices require user input for any incoming message or connection.

This is surprisingly effective as when a device requires authentication for even a vcard it is difficult to find a way in without an unsecured channel. The biggest security risk seems to be the users themselves several attacks succeeded simple because the users accepted the incoming connection (many harmless audits where performed on bypassers) allowing access on their device (we considered this a failure of the attack). No amount of security can prevent a user opening the door so to speak. No additional security software was found for Bluetooth.

6.3 Overall Result and Comments

The process started with attempts to find and use Bluetooth programs on the Pocket PC this eventually failed despite attempt to find a compatible functioning stack and software.

The next stage was to use the Mobile phone with an old version of Blueooover1 this was met with limited success as some tests causes it to crash. The true problems began when Linux was installed and used starting with Ubuntu and the days spent trying to find dependencies, compile, and run software began. After repeated problems which were solved in turn leading to more complex problems. David later suggested another build of Linux called backtrack with all the programs already installed. This after some trial and error worked to a degree in finding and accessing devices although with no results. We believe with more time successful attacks would be possible while we learned the programs better. During this last backtrack stage blooover2 was found as was several other programs notably "Bluetooth Super Hack" and at least the practical side had some success at this late point.

7. Conclusions

At the end of this paper we should mention that there is no actual threat in Bluetooth and most of the hacking tools were developed by academic researchers or companies in order to test its security, but after 2007 no new attacks or vulnerabilities are encountered for Bluetooth. While these tools exist online they do not seem widely used or distributed and any potential wide scale threat seems unlikely in the near future.

The simple conclusion for most devices is the security lies in the hands of the user. To keep your device's Bluetooth turned off or hidden if it isn't needed. And to never accept any incoming connection or file you don't recognise. While some devices did have unsecured channels and could be directly attacked, the majority of attacks that succeeded where simply let through by the users.

As we have learnt from studying Bluetooth security, that there are two types of PIN-code a fixed one and a dynamic one (which should be entered by the user). This issue leads us to many different security mechanisms as every device uses Bluetooth has it is own security mechanism and technique. In this document we talked about the general Blue security mechanisms. Such as Authentication methods and encryption techniques used in Bluetooth. Also Security interfaces and associated databases.

So in summary while the tests met only with limited success we have learned a lot about Bluetooth and its security (or lack thereof). It seems that due to a range of factors no one really cares enough about Bluetooth to cause any real problems other than a few "hackers" who attack single users for amusement.

References

[1] http://trifinite.org/, , Accessed last May 9, 2008.

[2] http://en.wikipedia.org/wiki/Bluetooth,Accessed last May 9, 2008.

[3] AppleConnection, http://developer.apple.com/documentation/DeviceDrivers/Conceptual/Bluetooth/BT_Bluetooth_Basics/chapter_2_section_4.html. Accessed last May,9 2008.

[4] Dieter Gollmann, "Computer Security".

[5] Marek Bialoglowy," Bluetooth Security Review", http://www.securityfocus.com/infocus/1830,last access may,9 2008.

[6] Andreas Becker,"Bluetooth Security and Hacks", Ruhr-University Bochum, 2007.

[7] Thomas Muller,"Bluetooth Security Architecture", 1999.

[8] Bluetooth SIG, "Bluetooth Security White Paper", 2002.

[9] Essential Bluetooth hacking tools, http://www.security-hacks.com/2007/05/25/essential-bluetooth-hacking-tools, Accessed last May 9, 2008.

[10] Juha T. Vainio," Bluetooth Security", 2000 http://web.archive.org/web/20060519034246/www.niksula.hut.fi/~jiitv/bluesec.html, last aceesed may 9,2008.