Bluetooth Intrusion Techniques

Jesús Antonio Alvarez-Cedillo,
Parallel Computing Laboratory
CIDETEC IPN, México

Patricia Perez-Romero
Parallel Computing Laboratory
CIDETEC IPN, México

Miguel Hernandez-Bolaños,
Parallel Computing Laboratory
CIDETEC IPN, México

***Abstract*— Bluetooth technology has grown through the years, but because of the increase , security has become an issue. In all the world the people interacting and use this technology, when transferring data through devices or when using the computer or personal devices, null confidentiality and null authenticity are factors to remain secure. How this technology are very common, malicious exploits attack all time searching find access. In this paper we shown different exploits and vulnerabilities that are either in the design already, or have been maliciously exploited. Some of these exploits are Bluesnarfing, Bluejacking, Bluebugging, etc. By understanding these exploits, one can gain a general idea of how vulnerable Bluetooth possibly could be.**

Keywords – *B*luetooth – Bluesnarfing – Bluejacking – Bluebugging - SAFER+ - MAC address - DOS attack - Intrusion Techniques.

I. INTRODUCTION

Wireless technology has evolved through the years to facilitate the public with ease, convenience and efficiency in accessing data. Due to this growing demand, Bluetooth has advanced as an ubiquitous technology found in many everyday appliances.

Wireless technology allows use and share computing resources without physical connections between the client device and server of resources. This technology has been very important for companies primarily for communication, but its scope is much greater. The client devices are desktop computers, palmtops and cell phones, automobiles, refrigerators, and printers, the days of having a cluster of wires to interface into different devices are now the past.

With the growing popularity of this technology, malicious exploits also have grown respectively. One can easily locate an exploit through the Internet, which most of them are trivial to understand and use. There are many exploits to choose from to do one's bidding. Some of them can eavesdrop on information being transferred between two parties. Others can Spam a nearby mobile phone. While other software can virtually control all the processes of a mobile phone, such as editing contacts to making a call through the phone. The possibilities are astounding. Instead of searching individually for each exploit software that exists, we've decided to conveniently bundle them into a pack. A one stop shop for popular Bluetooth exploits.

II. BLUETOOTH

Bluetooth technology is designed and optimized for use in mobile devices, such as mobile computers, cellular handsets, network access points, printers [1,3], PDA's, desktops, keyboards, joysticks and virtually any other device. The technology is relatively robust and inexpensive. It operates in a short range 2.4GHz Industrial-Scientific-Medical (ISM) band, which can reach distances of 10 to 100 meters. It uses Frequency Hop (FH) spread spectrum, which divides the frequency band into a number of hop channels. A Time-Division Duplex scheme is used for full duplex transmission. There are tiny radio-frequency transmitters, no larger than 1.0 by 0.5 inches that can run off a watch battery for months. Power considerations are always important for battery-powered mobile devices, and Blue-tooth's low power modes meet those requirements with less than 0.1 W active power. Bluetooth is intended to be a standard that works at two levels:

It provides agreement at the physical level (radio-frequency standard).

It also provides agreement at the next level up, where products have to agree on when bits are sent, how many will be sent at a time and how the parties in a conversation can be sure that the message received is the same as the message sent.

The Bluetooth protocol uses a combination of circuit and packet switching to send/receive data [2, 4]. A frequency-hopping spread spectrum technique is used to make it difficult to track or intercept transmissions. Each Bluetooth device has a unique 48 bit hard-wired device address for identity, which allows for 2^{48} devices. Bluetooth devices basically form piconets to communicate. Each piconet [5] comprises of up to eight active devices where one is the 'master' and the rest are 'slaves'. The master searches for Bluetooth devices followed by invitations to join the piconet addressed to specific devices. The 'master' then assigns a member-address to each slave and controls their transmissions. Devices can belong to several piconets. Bluetooth also provides for easy integration of TCP/IP for networking.

A. *Bluetooth Protocol Architecture*

First, Bluetooth is designed for communications applications [6] It is designed to support high quality simultaneous voice and data transfers, with rates reaching up to 721 Kbps. It supports both synchronous and asynchronous services and easy integration of TCP/IP for networking purposes. The Bluetooth specification divides the Bluetooth protocol stack into three logical groups. They are the Transport Protocol group, the Middleware Protocol group and the Application group, as shown in Figure 1.

The Transport group protocols allow Bluetooth devices to locate each other, and to manage physical and logical links with higher layer protocols and applications. It is important to note that the Transport protocol group does not indicate any coincidence with the Transport layer of the Open Systems Interconnection. Rather these protocols correspond to the Data-Link and Physical layers of the OSI model. The Radio, Baseband, Link Manager, Logical Link Control and Adaptation (L2CAP) layers and the Host Controller Interface (HCI) are included in the Transport Protocol group. These protocols support both asynchronous and synchronous transmission. All the protocols in this group are required to support communications between Bluetooth devices. A brief discussion of the layers in the Transport group follows.

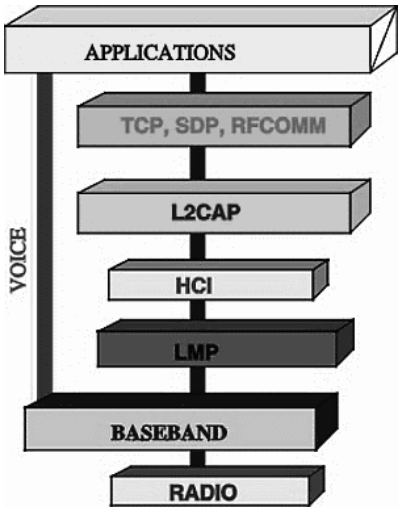

Figure 1 : Bluetooth Protocol Architecture

- *Radio Layer*

The specification of the Radio layer is primarily concerned with the design of the Bluetooth transceivers.

- *Baseband Layer*

This layer defines how Bluetooth devices search for and connect to other devices. The master and slave roles that a device may assume are defined here, as are the frequency-hopping sequences used by devices. The devices use a time division duplexing (TDD), packet-based polling scheme to share the air-interface. The master and slave each communicate only in their pre-assigned time slots. Also, defined here are the types of packets, packet processing procedures and the strategies for error detection and correction, signal scrambling (whitening), encryption, packet transmission and retransmissions. The Baseband layer supports two types of links: Synchronous Connection-Oriented (SCO) and Asynchronous Connection-Less (ACL). SCO links are characterized by a periodic, single-slot packet assignment, and are primarily used for voice transmissions that require fast, consistent data transfer. A device that has established a SCO link has, in essence, reserved certain time slots for its use. Its data packets are treated as priority packets, and will be serviced before any ACL packets. A device with an ACL link can send variable length packets of 1, 3 or 5 time-slot lengths. But it has no time slots reserved for it.

- *Link Manager Layer*

This layer implements the Link Manager Protocol (LMP), which manages the properties of the air interface link between devices. LMP manages bandwidth allocation for general data, bandwidth reservation for audio traffic, authentication using challenge response methods, and trust relationships between devices, encryption of data and control of power usage. Power usage control includes the negotiation of low power activity modes and the determination of transmission power levels.

- *L2CAP Layer*

The Logical Link Control and Adaptation Protocol (L2CAP) layer provides the interface between the higher-layer protocols and the lower-layer transport protocols. L2CAP supports multiplexing of several higher layer protocols, such as RFComm and SDP. This allows multiple protocols and applications to share the air-interface. L2CAP is also responsible for packet segmentation and reassembly, and for maintaining the negotiated service level between devices.

- *HCI Layer*

The Host Controller Interface (HCI) layer defines a standard interface for upper level applications to access the lower layers of the stack. This layer is not a required part of the specification. Its purpose is to enable interoperability among devices and the use of existing higher-level protocols and applications.

The Middleware Protocol group includes third-party and industry-standard protocols, as well as Bluetooth SIG developed protocols. These protocols allow existing and new applications to operate over Bluetooth links. Industry standard protocols include Point-to-Point Protocol (PPP), Internet Protocol (IP), Transmission Control Protocol (TCP), wireless application protocols (WAP), and object exchange (OBEX) protocols, adopted from Infrared Data Association (IrDA). Bluetooth SIG-developed protocols include

1) A serial port emulator (RFCOMM) that enables legacy applications to operate seamlessly over Bluetooth transport protocols.

2) A packet based telephony control signaling protocol (TCS) for managing telephony operations.
3) A service discovery protocol (SDP) that allows devices to obtain information about each other's available services.

Reuse of existing protocols and seamless interfacing to existing applications was a high priority in the development of the Bluetooth specifications, as shown in Figure 2.

The Application group consists of actual applications that use Bluetooth links. They can include legacy applications as well as Blue-tooth-aware applications.

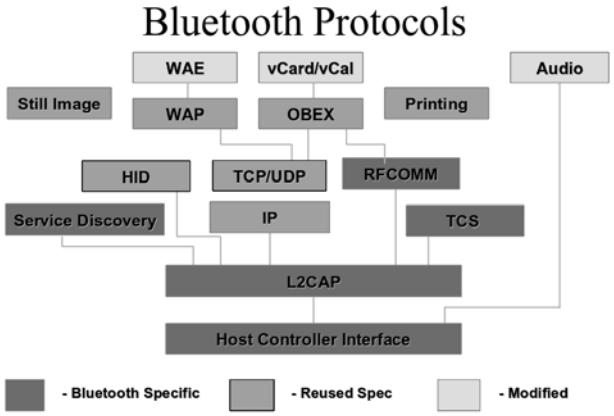

Figure 2: Interoperability with Existing Protocols and Applications

B. Hacking

Currently there are a few methods known for bypassing Blue-tooth's security measures.

One method of hacking Bluetooth has been named "Bluesnarfing", and as with most Bluetooth hacks, the reason for its existence is a fault of the way Bluetooth is implemented on certain mobile phones. In this case, is the way in which the object exchange (OBEX) protocol [4] has been implemented. What it does is, it silently access these mobile phones contacts, calendar and pictures without the owner's knowledge - a clear violation of the owner's security expectations. Nokia is one of a few mobile phone companies who have acknowledged that some of their devices have this fault, and have addressed it with updated firmware for their faulty products.

Another method is that of "back-door" hacking. This is where a device which is no longer trusted can still gain access to the mobile phone and gain access to data as with Bluesnarfing, or also use services like WAP, etc.

A third flaw in some mobile phones allows for a hacker to use a method called "Bluebugging" in order to hack into the owner's phone. It is possibly the most dangerous of the attacks, and allows hackers to send/read SMS, call numbers, monitor phone calls and also do everything that back-door and Bluesnarfing allows. This is a separate vulnerability from Bluesnarfing and does not affect all of the same phones as Bluesnarfing

The seemingly harmless "Bluejacking" is a different style of attack. It works on the fact that during the initialization process, when a device wishes to be paired with you, a message containing the device's name and whether you want to pair with this device is displayed. To many people this is just an innocent joke to get a reaction out of someone by renaming their phone and then sending them a clever anonymous message and watching their reaction. However, if a malicious individual names their phone something like "Click accept to win!!" then they can gain access to someone's Bluetooth device if an owner falls for the trick.

As with computers, there is also the risk of worms and viruses. One such worm is the Cabir worm, which tries to pair the Bluetooth device to any other Bluetooth device in the vicinity, and if successful it will install itself on the paired device. Once it is there, it will attempt to repeat this process, and also when the device is switched on, the worm will drain the battery by scanning for enabled Bluetooth devices.

There is also the possibility for Denial of Service (DOS) attacks on Bluetooth devices. This works exactly the same way that traditional DOS attacks work, with a hacker sending invalid Bluetooth requests and is occupying a device's Bluetooth channel so it cannot communicate with any other Bluetooth devices.

The first three of these issues are purely faults of the manufacturers of particular mobile phones, and firmware has been released since their discovery to correct any faulty models. These problems illustrate the dangers of using Bluetooth devices if they are not implemented properly. Indeed, they can all be solved, for most phones, by switching the phone into "invisible" mode so that it will not be recognized by other Bluetooth devices. Switching off the Bluetooth capability when you're not using it is another more extreme option. The Bluejacking and Cabir worm issues can only hack someone's phone if they agree to be paired with the device and in the case of the cabir worm if they agree then it also tries to install software. There are also security updates and anti-virus software readily available for users. These user security measures show that, as with any technology, there is responsibility on the user to take care of their devices.

III. EXPLOITS

There are many exploits that can be easily accessed through the Internet. Some of these exploits are trivial that just Spam other mobile phones nearby. While other exploits are advanced enough to edit mobile phone contacts or make a call through the phone. The following sections individually provide general information of existing exploits.

The main reason why most of these exploits can occur is because a Bluetooth device is left on discoverable mode, which allows it to be discovered by another Bluetooth device. Once

discovered, the exploit software will have retrieved the device's MAC address [7,8] which it can use to issue an attack. If the device was never in discoverable mode, this event would have never happened. Newer technologies now will only allow devices to remain in discoverable mode for only a limited time. For example, if a person wants his device to be discoverable, he would have to switch the mode to on, and then after a certain time has past it will automatically switch off. This is a good safety mechanism to make sure the device is never left on discoverable mode. In addition, discoverable mode is legitimately used for pairing of two Bluetooth devices. This process is not time consuming at all. [9] So, having an auto shutdown of discoverable mode in an arbitrary time will still leave ample time for the pairing process to complete.

A. Bluesnarfing

Bluesnarfing [10] is an unauthorized access of information. The unauthorized access allows the attacker to gain and edit information on calendar entries, contacts list, and emails.

Bluesnarf has been first identified by Marcel Holtmann in September 2003. Independently, Adam Laurie also discovered the same vulnerability in November 2003. To be able to perform a Blue snarf attack, the attacker's device needs to connect to the Object Exchange Protocol (OBEX) Push Profile (OPP). This protocol is primarily responsible for exchanging information between two devices, including business cards and other objects, and is very much similar to the known FTP protocol. The OBEX does not usually require authentication, and if it does require authentication, it will not be a problem as long as everything is implemented correctly. So to execute an attack, the attacker connects to an OBEX Push target and performs an OBEX Get request for files such as "telecom/cal.vcs" for the device's calendar or "telecom/pb.vcf" for the devices phone book. The OBEX process that is running does not provide file browsing, the names of the previously mentioned files can easily be known through the Infrared Mobile Communications, which they include specifications of many file names. So due to a device firmware problem, an attacker can easily access those files. Since this problem relates with a firmware problem, only certain mobile phones are susceptible to this attack. Currently Sony Ericsson and Nokia have a few models that are affected by Blue snarf [Table 3].

B. Bluebug

Bluebug [12] is the name of a Bluetooth security loophole that has been identified by Adam Laurie from A.L. Digital Ltd. on some Blue-tooth-enabled cell phones. Exploiting this loophole allows the unauthorized downloading phone books and call lists, the sending and reading of SMS messages, connection to the INTERNET, changing a service provider, initiating a call through the phone, and many more. Under ideal conditions, it is possible for a Bluebug attack to only take a few seconds. Due to the limited transmit power of class 2 Bluetooth radios; the distance of the victim's device to the attacker's device during the attack should not exceed 10-15 meters. A directional antenna can be attached to the radio in order to increase the range.

Since the Bluebug security loophole allows to issue AT (modem) commands via a covert channel to the vulnerable phones without prompting the owner of the phone, this security flaw does allow a vast number of things that may be done when the phone is attacked via Bluetooth. Some example follows.

Initiating Phone Calls

The Bluebug security loophole allows the attacker to initiate phone calls from the victim's device. Things that can be done with initiating phone calls include:

1) Eavesdropping: When the victim passes, a phone that is owned by the attacker (e.g. an anonymously used prepaid-card phone) is called. From this moment on, the attacker is able to listen to all the conversations that the victim does until the victim hangs up the phone
2) Causing financial damage: Since phone calls to any number can be established, it is also possible to call premium service numbers from the victim's device. If the victim does not realize that a phone call is connected to a premium service number, this would cause important financial damage to the victim.

Sending SMS from the victim's device for

1) Finding out the victim's phone number: The phone number of the respective device is not stored at a predefined location. The device's number can be gained by sending an SMS from the victim's device to a phone that is owned by the attacker.
2) Causing financial damage: There are quite a lot of SMS-based services that cost the client about 5 Dolars per SMS. Usually, these services are used to sell ring tones and logos. There are also news subscriptions that can be ordered by SMS that continuously cause costs to the victim.
3) Tracking the victim: As a location-based service, some providers allow other users to locate their customers by the GSM global cell id which their phone is connected to. According to the mode the respective GSM cells are configured, this information can be very detailed. In order to do this, the provider must get the permission from the customer. This permission is usually given via SMS (which is sent by the attacker).
4) Revealing secrets: Often SMS messages are used to silently communicate secret information with other people. *Reading SMS of the attacked device is often touching the victim's privacy. Paparazzi could use this attack in order to find out more about certain celebrities.*

Reading and writing phone book entries for

1) Finding out callers and called persons: In GSM handsets, phone books are also used for managing call lists. So the attacker may find out who the victim called last, who was trying to reach the victim's device and who reached the victim's device.
2) Modifying entries: A nasty phone book entry could be the name "Darling" and the international emergency number 112 :)
3) Obfuscating the abuse: After initiating phone calls, the list of dialed numbers could be overwritten.

Call Forwards

Setting call forwards on the victim's phone could cause a lot of confusion. So instead of calling the victim, the caller reaches the device connected to a random number that has been set.

Internet Abuse

The attacker can use the Bluebug loophole to establish an Internet connection that could for example be used for the illegal injection of Mail-Worms like Sasser, Phatbot or NetSky.

Network Provider Pre-selection

In locations like airports, where many users arrive with their cell phones, service providers could use the Bluebug loophole in order to register these phones with their networks.

C. Bluejack

Bluejacking [12] is a trivial exploit that utilizes a design flaw in the OBEX layer. Any random person can easily Spam another person's mobile phone by sending a text message. The original designs intentions were for a person to easily send another person their business card via their mobile phone. However, that technology has been exploited by people now sending random messages to another person when the device is close enough to detect the other Bluetooth device.

D. Bluesmack

Bluesmack is a DOS (Denial of Service) attack [13]. This attack exploits the L2CAP layer, which is responsible for echo requests similar to ICMP (Internet Control Message Protocol) ping. Bluesmack sends an enormous amount of ping requests to another device, which makes the victim's device unresponsive to any services. This attack is similar to "Ping of Death" and "Smurf Attack". Since a standard Bluetooth device cannot handle large amounts of ping requests, this will result in a buffer overflow and cause the device to be unavailable. This method can be tweaked to create a buffer overflow by injecting malicious code into the device.

E. Cracking Bluetooth PIN

Apart from all the other Bluetooth exploits mentioned in this section, cracking the Bluetooth PIN is not software that can be downloaded. Instead, it's a theoretical process of how one might be able to crack a Bluetooth PIN. It's still considered an exploit of course, and a very interesting one indeed, so that's why it's being included in this section.

Some reasons why cracking the PIN is possible is because the PIN number is usually much too short to provide a secure access code. Most people do not want to hassle with a long number to have to remember. Another reason is Bluetooth has incorporated new cryptographic primitives, which possibly has not been tested thoroughly and may contain hidden flaws. Finally, Bluetooth originally was restricted to a range of only 10m. This was a main factor into security, since most attackers can't commit to an attack, unless they are within range of the victim. However, as wireless technology advanced through the years, range extenders are possible now and can be built very inexpensively.

When two Bluetooth devices establish a channel for communication, the process where the channel is being established is called the pairing or bonding process. This process requires the use of a PIN, so it's during the paring process of two Bluetooth devices when a PIN could be cracked. However, before the paring process begins, a PIN code should have been entered into both Bluetooth devices.

There are some devices that have a fixed PIN which cannot be changed (e.g. wireless headphones). In this situation, the fixed PIN is entered into the peer device. This will be a problem if both devices have fixed PIN, which those two devices cannot pair with one another.

The Bluetooth pairing process consists of three steps.

1. Creation of an initialization key (K_{init})
2. Creation of a link key (K_{ab})
3. Authentication

After the above steps are completed, the devices have an additional option to derive an encryption key to hide all future communications.

The creation of an initialization key (K_{init}) step requires three parameters, a MAC address (BD_ADDR), PIN code and its length and a 128 bit random number (IN_RAND). This step

generates the K_{init} which is a 128 bit word using the E_{22} algorithm (Figure 3).

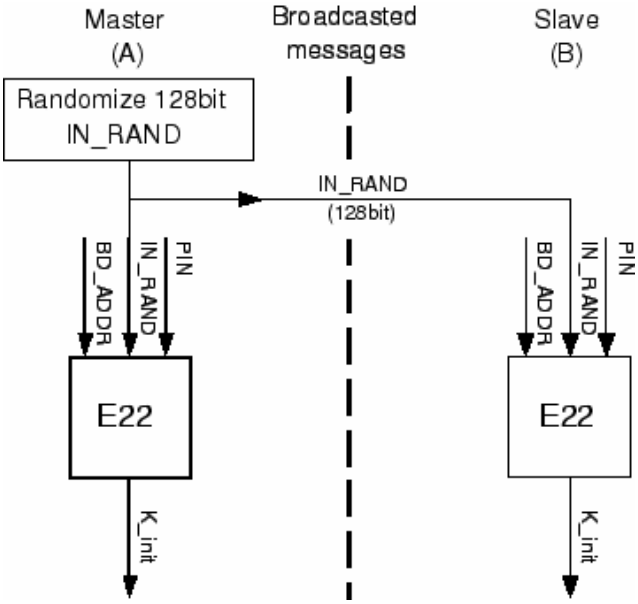

Figure 3 : Generating K_{init} using the E_{22} Algorithm

The BD_ADDR used in the E_{22} algorithm [14] will be the device that does not have a fixed PIN. If both devices do not have a fixed PIN, then the BD_ADDR used will be the slave device that receives the IN_RAND. The initialization key is only used during the pairing process and is discarded upon creation of the link key.

The creation of the link key (K_{ab}) requires the use of the K_{init} to exchange two new random 128 bit words, known as LK_RAND_A and LK_RAND_B. The two new random words are used in the E_{21} algorithm [14] (Figure 4) to generate the K_{ab}.

Once the link key has been generated, mutual authentication can be performed, which relies on a challenge-response scheme. The two devices have separate roles, where one is labeled the verifier and the other is the claimant. The verifier's role is to send a random 128 bit word called the AU_RAND_A to the claimant, which the claimant will use to generate a 32 bit word called the SRES using the algorithm E_1 (Algorithm 16). The claimant will then send the SRES to the verifier, who verifies the response word by performing the same calculation. If the response word is successful, the verifier and claimant will exchange roles, and the process is repeated. During the mutual authentication process, a 96 bit word called the ACO will be generated as a side effect. This word is optional to be used to create an encryption key.

Now that we understand how the pairing process works, we can go over the basics to actually crack the PIN. Assume that an attacker has eavesdropped on an entire pairing and authentication process, and have saved all the messages (Table 1). With the above information obtained, the attacker can use a brute force algorithm to find the PIN used. By iterating through all possible values of PIN, the attacker can find a hypothesis for K_{init}, since IN_RAND and BD_ADDR is known and is applied to the E_{22} algorithm. Now use the initialization key to decode the messages in 2 and 3 from the Table 1.

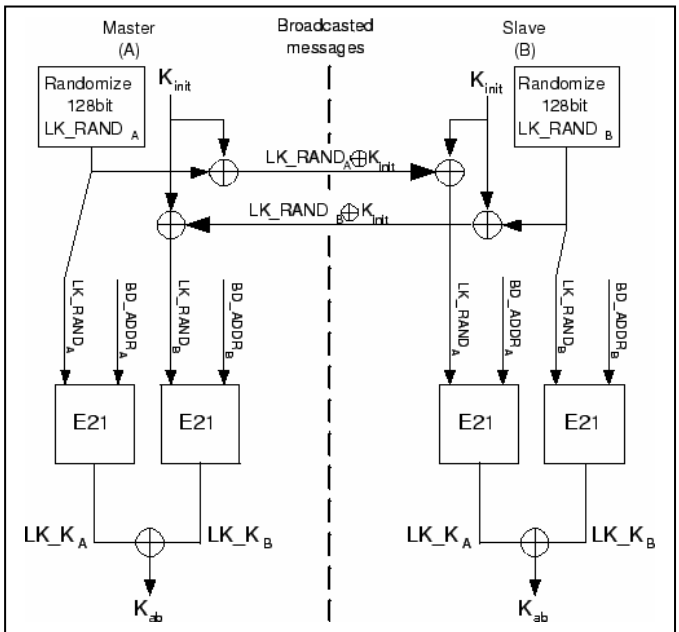

Figure 4: Generating the K_{ab} using the E21 Algorithm

messages 2 and 3 contain enough information to form a hypothesis for K_{ab}. With the hypothesis Kab the attacker can test it with the last four messages in Table 1. This whole process is described in a flow layout in Figure 5.

TABLE I. LIST OF MESSAGES SENT DURING THE PAIRING AND AUTHENTICATION PROCESS OF BLUETOOTH DEVICES A AND B

#	Src	Dst	Data	Length	Notes
1	A	B	IN_RAND	128 bit	plaintext
2	A	B	LK_RANDA	128 bit	XOR with Kinit
3	B	A	LK_RANDB	128 bit	XOR with Kinit
4	A	B	AU_RANDA	128 bit	plaintext
5	B	A	SRES	32 bit	plaintext
6	B	A	AU_RANDB	128 bit	plaintext
7	A	B	SRES	32 bit	plaintext

This Bluetooth pin crack method is not taking advantage of the design in the Bluetooth pairing process. Instead, as long as an attacker can eavesdrop and save all the messages of the pairing process, and be able to apply guessed Pin's to the E_{22}, E_{21} and E_1 algorithm, this brute force attack is then definitely feasible. There are of course computational enhancements that can be done in this basic method to speed up the PIN cracking, which is described further in "Cracking the Bluetooth Pin " [6]. Table 2 will give an idea of the performance of this procedure upon different PIN lengths with an advanced method mentioned in the above paper.

IV. BLUETOOTH EXPLOITS TEST

To give a better understanding of how the Bluetooth exploits work, a practical example always works best. This section will give a description of the process that was developed to test out different Bluetooth exploits, which are mainly found in the "Exploits" section of this paper. Unfortunately there are certain Bluetooth exploits that are only vulnerable on certain devices, which can be referenced in Table 3.

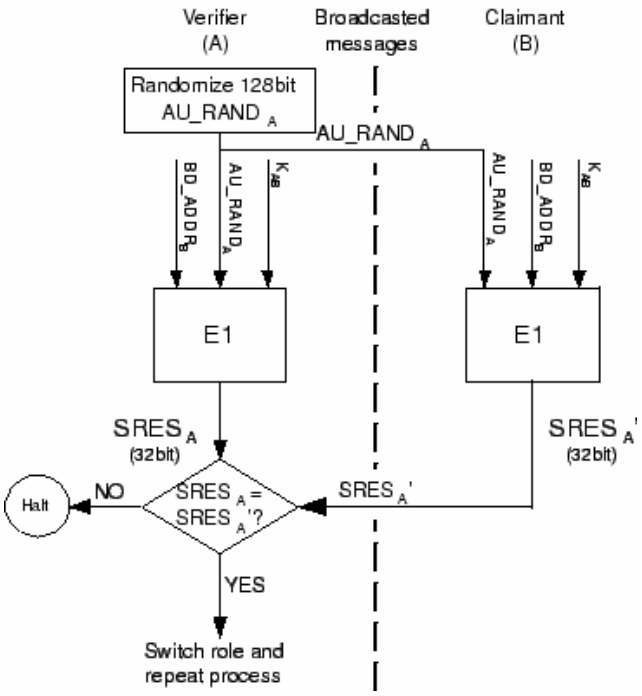

Figure 2 : Authentication using the E1 Algorithm

V. FUTURE WORK

It would be ideal if the exploits would work for any current mobile phone. The Blue snarf and Bluebug attacks are among the most interesting. However, as stated in the paper, only certain mobile phones are vulnerable to them, and that's a result of the old OBEX design in those mobile phones. Possibly for a future work would be to obtain a better understanding of the OBEX layer and see if there is any vulnerability that might have propagated through from its previous versions.

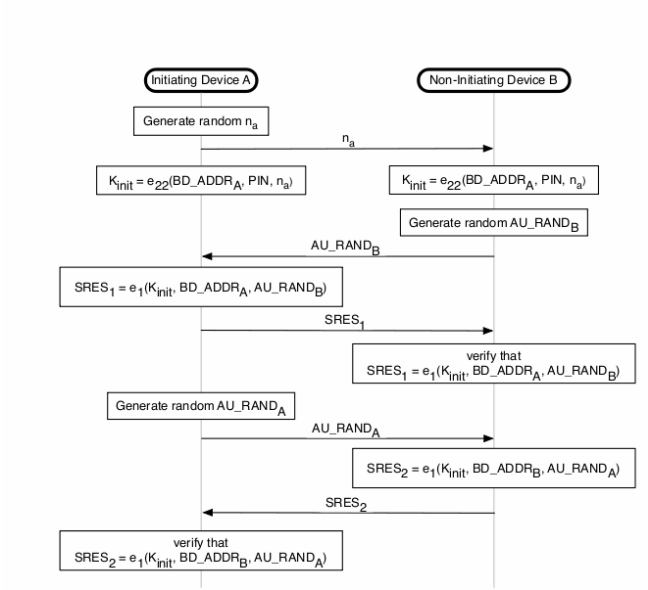

Figure 5: Flow of messages sent during the pairing and authentication process of Bluetooth devices A and B

Algorithm 1: Brute force attack

Input: Pin[0]
Output: HackValue.

1: Set PIN=0
2: Calculate a hypothesis for Kinit
3: decode $LKRAND_A$ and $LKRAND_B$
4: Calculate a hypothesis for Kab
5: SET $SRES=AUNRAND_A$
6: IF SRES=SRES' THEN
7: SET $SRES'=AUNRAND_B$
8: ELSE TRY ANOTHER PIN
9: IF $SRES_B$ = SRES' THEN
10: HACKVALUE=PIN
11: ELSE TRY ANOTHER PIN

Algorithm 1: Brute force attack on Bluetooth PIN algorithm

The part on cracking Bluetooth PIN is another very interesting topic which would be educational to understand further and possibly implement. Some problems that hindered this project from doing such an exploit are the necessary appliances/software needed to do the exploit. The exploit requires an eavesdrop device, which is not available. Nonetheless, this exploit possibly could be applicable for a future work.

TABLE 1 : RESULTS RUNNING AN ENHANCED VERSION ON A PENTIUM CELERON , CORE DUO USING UBUNTU 10.04

PIN Length (digits)	Time (seconds)
4	0.063
5	0.75
6	7.609
7	76.127

TABLE 2 : LIST OF MOBILE PHONES THAT MIGHT BE VULNERABLE TO BLUE SNARF AND BLUEBUG ATTACKS

Make	Model	Bluesnarf	Bluebug
Ericsson	T68	Yes	NO
Sony Ericsson	R520m	Yes	?
	T68i	Yes	?
	T610	Yes	Yes
	Z1010	Yes	?
	Z600	Yes	?
Nokia	6310	Yes	?
	6310i	Yes	Yes
	8910	Yes	?
	8910i	Yes	?
Motorola	V600	No	Yes
	V80	No	Yes

REFERENCES

[1] [1] Bluetooth SIG. Specification of the Bluetooth System. Version 1.1, Feb. 2001.

[2] [2] Bluetooth SIG. Bluetooth Network Encapsulation Protocol (BNEP) Specification. Technical report, Revision 0.95a, June 2001.

[3] [3] Bluetooth Special Interest Group, "Specification of the Bluetooth System 1.0b, Volume 1: Core," http://www.bluetooth.com, Dec. 1999.

[4] [4] J. Haartsen, "The Bluetooth Radio System," IEEE Personal Communications, Vol. 7, No. 1, pp. 28-36, Feb. 2000.

[5] [5] Vojislav B. Mišić, Eric W. S. Ko, Jelena Mišić, "Load and QoS-Adaptive Scheduling in Bluetooth Piconets," hicss, vol. 9, pp.90294c, Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 9, 2004

[6] [6] Rudi Latuske, ARS Software GmbH, OBEX performance evaluation and parameter optimization for high speed IrDA,2004

[7] [7] A. Das, A. Ghose, A. Razdan, H. Saran, and R. Shorey. Enhancing performance of asynchronous data traffic over the Bluetooth wireless ad-hoc network. In Proceedings Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies IEEE INFOCOM 2001., volume 1, pages 591–600, Anchorage, AK, Apr. 2001.

[8] [8] M. Kalia, D. Bansal, and R. Shorey. Data scheduling and SAR for Bluetooth MAC. In Proceedings VTC2000-Spring IEEE 51st Vehicular Technology Conference, volume 2, pages 716–720, Tokyo, Japan, May 2000.

[9] [9] B. A. Miller and C. Bisdikian. Bluetooth Revealed: The Insider's Guide to an Open Specification for Global Wireless Communications. Prentice-Hall, Upper Saddle River, NJ, 2000.

[10] [10] Martin Herfurt, Bluesnarfing, @ CeBIT 2004–Detecting and Attacking bluetoothenabled Cellphones at the Hannover Fairground

[11] [11] Adam Laurie, Marcel Holtmann , Martin Herfurt, Hacking Bluetooth enabled mobile phones and beyond , 21st Chaos Communication Congress December 27th to 29th, 2004, Berliner Congress Center, Berlin, Germany.

[12] [12] L Owens, First Bluejacking, Now Bluesnarfing, 2004.

[13] [13] Timothy K. Buennemeyer, Battery Polling and Trace Determination for

[14] bluetooth Attack Detection in Mobile Devices, proceedings of the 2007 IEEE.

[15] [14] Yanik Shaked, Avishai,Battery Polling and Trace Determination for bluetooth Attack Detection in Mobile Devices, Proceedings of the 2007 IEEE, Workshop on Information Assurance United States Military Academy, West Point, NY 20-22 June 2007.