2009

Bluetooth Hacking: A Case Study

Dennis Browning
Champlain College

Gary C. Kessler
Champlain College - Burlington, kessleg1@erau.edu

Bluetooth Hacking: A Case Study

Dennis Browning
Champlain College Center for Digital Investigation
Burlington, Vermont
dennisbrowning@gmail.com

Gary C. Kessler
+1 802-865-6460
Champlain College Center for Digital Investigation
Burlington, Vermont
Edith Cowan University
Perth, Western Australia
gary.kessler@champlain.edu

ABSTRACT

This paper describes a student project examining mechanisms with which to attack Bluetooth-enabled devices. The paper briefly describes the protocol architecture of Bluetooth and the Java interface that programmers can use to connect to Bluetooth communication services. Several types of attacks are described, along with a detailed example of two attack tools, Bloover II and BT Info.

Keywords: Bluetooth hacking, mobile phone hacking, wireless hacking

1. INTRODUCTION

Bluetooth (BT) is one of the newer wireless technologies in use today. The name derives from that of Harald Blaatand, a tenth-century king of Denmark and Norway who united many independent Scandinavian tribes into a single kingdom. Bluetooth wireless communication technology is meant to be a universal, standard communications protocol for short-range communications, intended to replace the cables connecting portable and fixed electronic devices (Bluetooth SIG, 2008a). Operating in the 2.4 GHz range, Bluetooth is designed to allow wire-free communication over a range of short-haul distances in three power classes, namely, short range (10-100 cm), ordinary range (10 m), and long range (100 m) (Sridhar, 2008). Cell phones, personal digital assistants (PDAs), and smart phones are a few of the devices that commonly use Bluetooth for synchronizing email, sending messages, or connecting to a remote headset (Mahmoud, 2003a). What are less well known to users of Bluetooth devices are the risks that they incur due to various vulnerabilities of the technology. Bluehacking, bluejacking, marphing, bluesniping, and bluesnafting are just a few of the names given to the act of hacking a device via

Bluetooth (Laurie, Holtmann, & Herfurt, 2006). In this paper, we will discuss the technology needed to hack a cell phone, some of the tools, and precautions that users can take to help protect their Bluetooth devices.

2. TECHNOLOGY

Figure 1 shows a diagram of the Bluetooth protocol stack in order to show the various attack vectors. The protocol layers of particular interest in this paper are:

- Logical Link Control and Adaptation Protocol (L2CAP): Provides the data interface between higher layer data protocols and applications, and the lower layers of the device; multiplexes multiple data streams; and adapts between different packet sizes (Hole, 2008a, 2008d; Sridhar, 2008).
- Radio Frequency Communications Protocol (RFCOMM): Emulates the functions of a serial communications interface (e.g., EIA-RS-232) on a computer. As Figure 1 shows, RFCOMM can be accessed by a variety of higher layer schemes, including AT commands, the Wireless Application Protocol (WAP) over the Transmission Control Protocol/Internet Protocol (TCP/IP) stack, or the Object Exchange (OBEX) protocol (Hole, 2008a, 2008e; Sridhar, 2008).
- Object Exchange protocol: A vendor-independent protocol allowing devices to exchange standard file objects, such as data files, business cards (e.g., vCard files), and calendar information (e.g., vCal files). OBEX is a higher layer application and runs over different operating systems (e.g., PalmOS and Windows CE) and different communications protocols (e.g., Bluetooth and IrDA) (Gusev, n.d.).

Most of the tools that are being used to hack Bluetooth phones use the Java programming language. In order for the software to work, the phone that is used to initiate the attack needs to support JSR-82, which is the official Java Bluetooth Application Programming Interface (API) (JCP, 2009). If the attacker's phone does not support JSR-82, that phone cannot be used to attack other phones. This is an important note because although Bluetooth is widely available on cell phones, Java and JSR-82 support may not be.

JSR-82 consists of two packages, namely, javax.bluetooth, which is the core Bluetooth API, and javax.obex, which is independent of the Bluetooth stack and provides APIs to other protocols, such as OBEX. The capabilities of JSR-82 include the ability to (Hole, 2007; Mahmoud, 2003b):

- Register services
- Discover devices and services

- Establish L2CAP, RFCOMM, and OBEX connections between devices, using those connections to send and receive data (voice communication is not supported)
- Manage and control the communication connections
- Provide security for these activities

Hole (2008a, 2008f) and Mahmoud (2003b) provide good overviews of how this code functions.

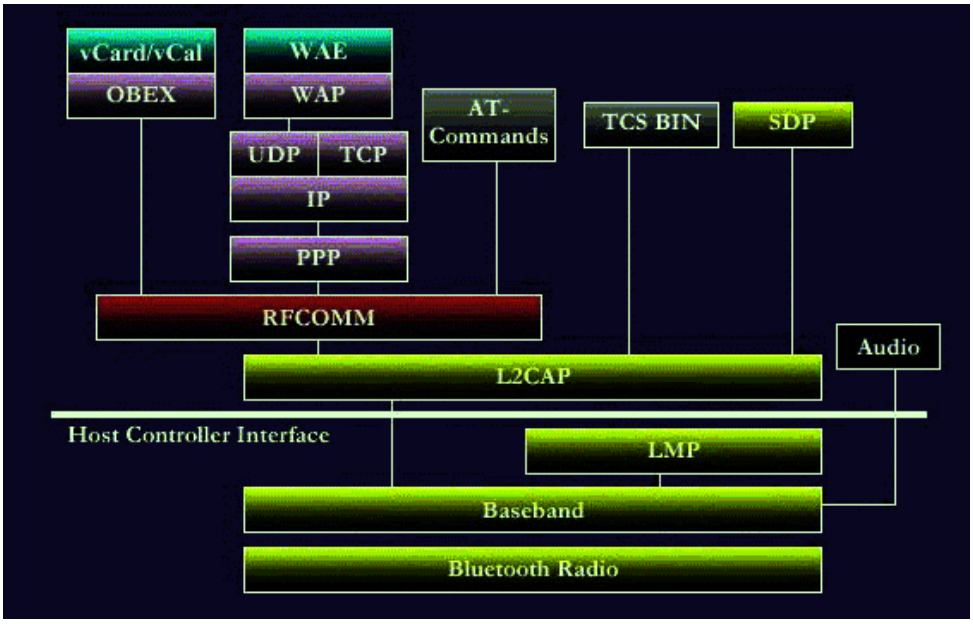


Figure 1: Bluetooth protocol stack (Source: *Tutorial-Reports.com, n.d.*)

3. BLUETOOTH SECURITY

Bluetooth defines three security modes. Security Mode 1 provides no security enforcement, meaning that the device is effectively taking no steps to protect itself. Security Mode 2 enforces security at the service level. In this mode, a particular application might be relatively safe but no additional device protection has been added. Security Mode 3 is the highest level of security, employing link level enforced security mechanisms. Security Mode 3 protects the device from certain intrusions and, therefore, all services and applications (Bluetooth SIG, 2008b; Hole, 2008b; Laurie et al., 2006).

All Bluetooth services have a default set level of security. Within the service level security, there are also three levels of security. Some services that require authorization and authentication in order to be used, some require

authentication only, and some are open to all devices (Bluetooth SIG, 2008b). Bluetooth devices themselves have two levels of security when describing other devices, namely trusted devices and untrusted devices.

4. TYPES OF ATTACKS

There are a variety of attacks that can be employed against Bluetooth devices, many with colorful names such as bluebugging, bluebumping, bluedumping, bluejacking, bluesmacking, bluesnarfing, bluespooofing [sic], bluestabbing, bluetoothing, and car whisperer. All take advantage of weaknesses in Bluetooth that allow an attacker unauthorized access to a victim's phone. It is imperative to note that while Bluetooth is commonly associated with networks limited in scope to 100 m, attacks on Bluetooth devices have been documented at ranges in excess of 1,500 m. using Bluetooone [sic] (Laurie, 2006).

One common approach to hacking Bluetooth devices is to employ malformed objects, which are legal files exchanged between BT devices that contain invalid information, thus causing unexpected results. When a Bluetooth device receives a malformed object, such as a vCard or vCal file, the device may become unstable or fail completely. Alternatively, an attacker might also use a vCard or vCal file to inject commands allowing the attacker to take control of the device. This kind of an attack can be very harmful to a phone (E-Stealth, 2008; Laurie et al., 2006).

Some of the common attacks on Bluetooth devices include:

- Bluebugging: An extraordinarily powerful attack mechanism, bluebugging allows an attacker to take control of a victim's phone using the AT command parser. Bluebug allows an attacker to access a victim's phone in order to make phone calls, send short message service (SMS) messages, read SMS messages stored on the phone, read and write contact list entries, alter phone service parameters, connect to the Internet, set call forwarding, and more (Bluebugging, n.d.; Laurie et al., 2006).
- Bluejacking: The sending of unsolicited messages to open Bluetooth devices by sending a vCard with a message in the name field and exploiting the OBEX protocol (Bluejacking, 2009).
- Bluesmack: A Bluetooth analog of the Ping-of-Death denial-of-service attack. This is a buffer overflow attack using L2CAP echo messages (Bluesmack, n.d.; Laurie, 2006).

- Bluesnarf and Bluesnarf++: Attacks allowing for the theft of information from a Bluetooth device using the OBEX Push Profile. The attacker needs only find a phone that has Bluetooth in discoverable mode. Bluesnarf works by a connection to most of the Object Push Profile services and the attacker retrieves the file names of known files from the Infrared Mobile Communications (IrMC) list instead of sending vCard information as expected. With these attacks the hacker can retrieve items such as the phonebook, calendar, and other personal information. With Bluesnarf++, the attacker has full read and write access to the file system of the phone. When an attacker is connected via the OBEX Push Profile, he/she has full access to the victim's phone without having to pair the two devices. The biggest risk with this function is that an attacker can delete crucial file system files, rendering the victim's device useless. In addition, the attacker can access any memory cards that are attached to the device (BlueSnarf, n.d.; Bluesnarfing, n.d.; Laurie et al., 2006).
- Helomoto: Helomoto is functionally similar to the Bluebug attack but takes advantage of poor implementations of "trusted device" handling on some phones. As in bluebug attacks, the attacker pretends to send a vCard to an unauthenticated OBEX Push Profile on the victim's phone. Once started, the attacker interrupts the transfer process and the victim then lists the attacker's phone as a trusted device. The attacker can then connect to the victim's phone and take control of the device by issuing AT commands. This attack is so-named because it was first discovered on Motorola phones (Helomoto, n.d.; Laurie et al., 2006).

These attacks are only a few that can be launched against Bluetooth interfaces in phones, laptops, and even automobiles. E-Stealth (2008) and Laurie et al. (2006) offer information about a wide range of attacks that can be launched via Bluetooth vulnerabilities.

5. TOOLS FOR ATTACK

There are many options that a user can choose from when looking to attack a Bluetooth phone. Web sites such as E-Stealth (http://www.e-stealth.com/) and FlexiSPY (http://www.flexispy.com/) offer commercial products to allow one party to eavesdrop or attack another party's Bluetooth device, ostensibly to trap an unfaithful spouse, catch an unscrupulous employee, or monitor a teenage child. These are merely commercial versions of hacker tools that include Bloover, Bloover II, BT Info, BT_File_Explorer, ISeeYourFiles, MiyuX, and STMBlueS (D3scene, 2008; E-Stealth, 2008; Getjar, 2008; Laurie et al., 2006; SE-NSE, 2006). Many of these programs (like so many hacker tools such as Back Orifice and SubSeven), are distributed as "management tools" but what differentiates them from bona fide management tools is that the managed party may not be aware that the program is running. And, like any "management"

tool, these programs are often platform-dependent so that they work best on certain brands of devices and may not work on all devices; MiyuX, for example, works best on Sony Ericsson phones. A nice collection of all of these tools in one package can be found at tradebit (http://www.tradebit.com/filedetail.php/5006527-basic-bluetooth-spy-software).

5.1 Testing the Software

The first author experimented with the feasibility of actually using this software in a real environment, employing Bloover II (which allows an attacker to obtain information from a victim's phone) and BT Info (which allows an attacker to control the victim's phone). Both were part of the Ultimate Bluetooth Mobile Phone Spy Software New Edition 2008 available from E-Stealth (http://www.e-stealth.com/).

It is worth noting that this software claims to be useable on any Bluetooth phone to hack any other Bluetooth phone but, like so many software claims, this one was overstated. Initial attempts to use the software on a Sanyo SCP-7050 failed because the software could not be installed. Later, the first author purchased a BlackBerry Curve. Although the software user guide provided instructions on how to install the software on a BlackBerry, the install failed when an error stated that the phone did not support the correct Java API.

The phones that were used successfully for testing throughout this project were United Kingdom versions of a Sony Ericsson W550i and a W800i. Theses phone both support JSR-82 enabling them to run the software. In order to actually use the phones, a Subscriber Identity Module (SIM) card was needed for each phone. The SIM card does not actually need to be active if the attacker is only going to be probing and manipulating the target phone and not making calls. Throughout the testing for this project both phones used inactive SIM cards.

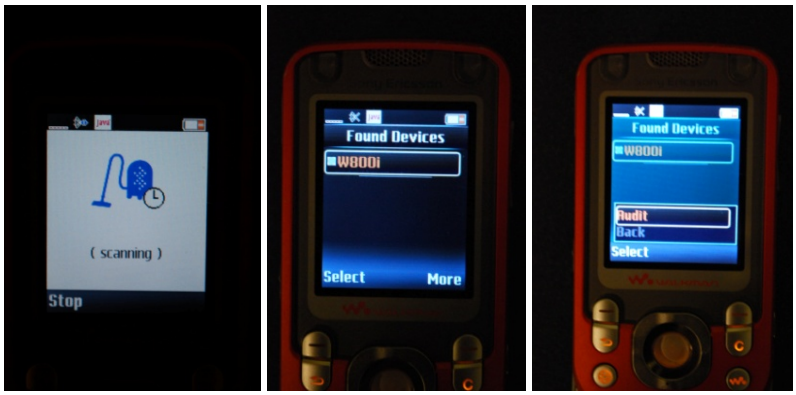
5.2 Bloover II

Bloover (also known as Blooover), standing for Bluetooth Wireless Technology Hoover, is a proof-of-concept application. Bloover II is a second-generation version of a program that consists of several different types of attacks, including Bluebug, Bluesnarf, Helomoto, and the use of malformed objects. Breeder is a related program that propagates Bloover II clients (Laurie et al., 2006).

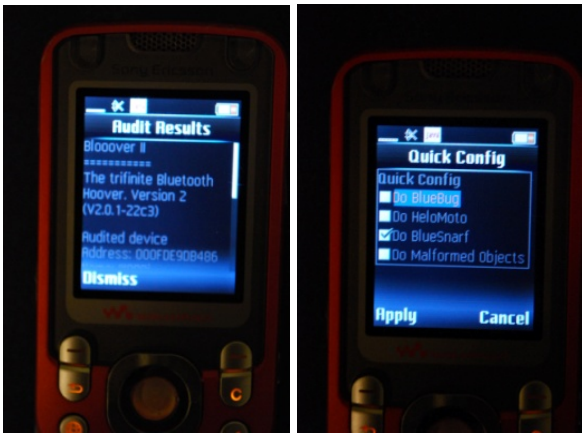
The attack software package that was purchased included a program called Bloover II. Once a JSR-82 enabled phone was found, the program installed easily. As for running the program, it seemed to always halt on one of the processes. One of the processes that the software kept halting on was when the program was running the "HeloMoto" attack. During this attack, the hacking phone tries to "plant" an entry into the victim's phonebook. Within the options

of the Bloover II program, the hacker can chose which attacks they would like to use on the victim's phone. When going through and trying each attack by itself, the software would always halt on some process. The only operation that could be conducted was the initial audit of the phone to get basic information about the phone.

Figure 2 shows a series of screen shots using Bloover II from a W550i phone to access a W800i phone. Figure 2a shows the attacker's phone scanning for another Bluetooth phone; in Figure 2b, a device named W800i is found. The audit feature of Bloover is initiated (Figure 2c) and results (Figure 2d) include the target device's address, communications channel for communication with the headset and other functional profiles, the RFCOMM channel, and phone contact information. A specific attack type (Bluebug in this case) is selected from the Quick Config menu (Figure 2d).



2a) 2b) 2c)

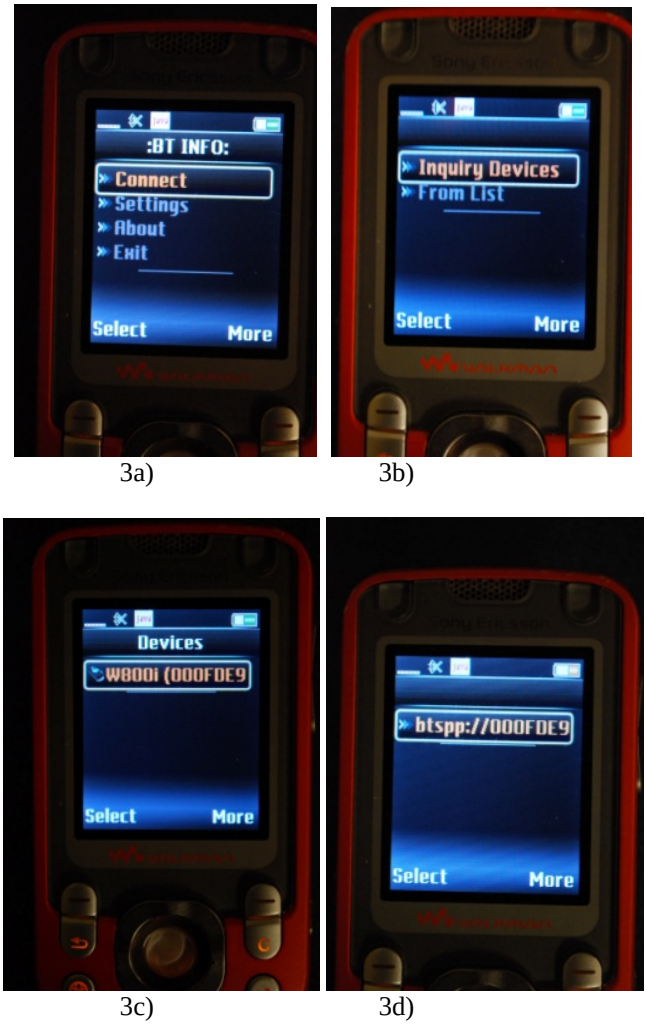


2d) 2e)

Figure 2. Bloover II screen shots.

5.3 BT Info

Because of increased functionality, a larger amount of time was spent using a program called BT Info. With this program, the attacker can completely control the target device *if* the attacker can become paired with the target. Once the Bluetooth pairing takes places, the attacker can perform a broad set of functions on the target phone, ranging from placing a phone call or sending an SMS message to turning the phone off or performing a master reset. The hardest part for the attacker, in fact, is finding a device with an open Bluetooth connection or tricking someone into pairing his or her phone.



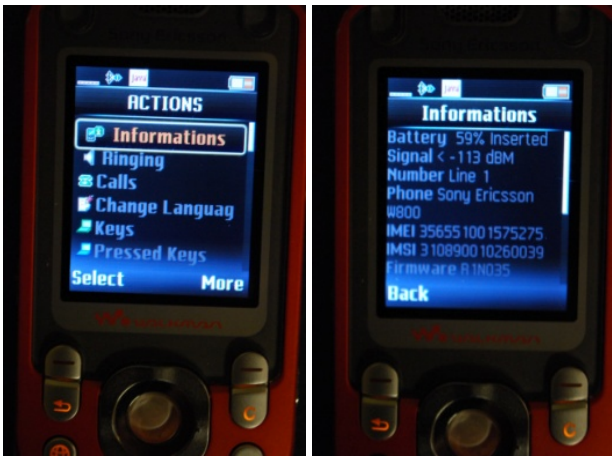
3a) 3b)

3c) 3d)

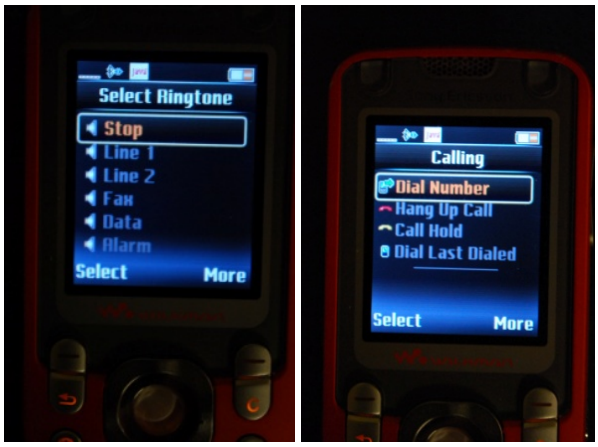
Figure 3. BT Info screen shots (device pairing).

Figure 3 shows a series of screen shots of an attacker's phone (W550i) pairing up with a target phone (W800i). Once pairing has been successfully accomplished, BT Info displays a menu of possible actions (Figure 4a). The Informations screen (Figure 4b) allows the attacker to retrieve basic information about the target phone, such as the phone manufacturer and model, firmware version, battery level, signal level, International Mobile Equipment Identity (IMEI), and International Mobile Subscriber Identity (IMSI).

The Ringing screen (Figure 4c) allows the attacker to control the ringing on the target phone. This option allows the attacker to force the target phone to start ringing and not stop until the target phone is turned off or the attacker issues the *Stop* command. Within the Ringing option, the attacker is able to select the type of ringtone to start.



4a) 4b)



4c) 4d)

Figure 4. BT Info screen shots (initial menu functions).

The Calling menu (Figure 4d) offers four options, allowing the attacker to dial any number, hang up a call, place a current call on hold, or redial the last number. An attacker can use the Calling option, for example, to call a second phone owned by the attacker in order to listen in on the victim's conversations. If the target phone has a speaker function that operates when the phone is closed, the attacker can still be able to establish a call and listen in. From the main Actions menu, the attacker can also change the display language that the phone uses.

5a) 5b)

5c) 5d)

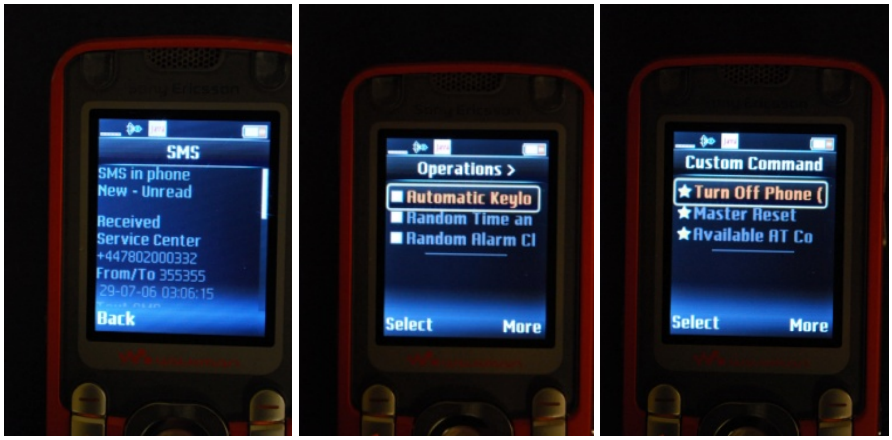
Figure 5. BT Info screen shots (Keys functions).

The Keys function (Figure 5a) is a feature of BT Info that allows an attacker to

watch the keys that the victim pushes as they are being pushed or allows an attacker to remotely press keys on the victim's phone. For the latter function, the attacker can access the target phone's "joystick" keys (Figure 5b) or individual keypad keys (Figure 5c). The control function of BT Info (Figure 5d) allows the attacker to remotely access the target's control keys, including volume control, media player, and camera.

BT Info also gives an attacker access to the target phone's text messages. The SMS action (Figure 6a), for example, allows the attacker to select a mailbox on the victim's phone and retrieve the complete contents of all SMS messages. Some of the other actions are simply informational, including the temperature of the phone, what Bluetooth devices are trusted on the victim's phone, what sound, if any, the phone makes when a button is pressed, the memory status, and what action forces a keylock.

The Operations action (Figure 6b) has several options. Automatic Keylock gives an attacker the ability to automatically lock the victim's when it is unlocked; i.e., when the victim unlocks the phone, it will automatically relock itself. The Random Time and Date Change option randomly changes the date and time on the victim's roughly a hundred times per minute. Similarly, the Random Alarm option randomly sets the victim phone's alarm settings.



6a) 6b) 6c)

Figure 6. BT Info screen shots (miscellaneous).

The Custom Command function (Figure 6c) allows an attacker to power down or force a master reset on a victim's phone. This function can also be used to

execute whatever AT commands are available on the target phone. BT Info also has a Phonebook function that allows an attacker to read the victim's phonebook and recent call history.

BT Info was tested using several different Bluetooth phones and was employed most successfully between the two Sony Ericsson phones mentioned above. The first author was able to use one of the Sony Ericsson phones to connect with a Motorola Razr, although the functionality of BT Info was somewhat limited, only allowing call initiation and access to SMS messages. Functionality of BT Info will vary by the model of both attacker and target phone (E-Stealth, 2008).

A video of the first author using BT Info between the two Sony Ericsson phones can be found at http://c3di.champlain.edu/TR/BTInfo_Browning.m4v (11 minutes, 350 MB).

6. PRECAUTIONS

As with so many aspects of security, user awareness and vigilance is the best defense against the kinds of attacks described here. The best way to protect a device, obviously, is to simply turn Bluetooth off. A device cannot be hacked via a Bluetooth attack vector if other Bluetooth devices cannot see it. Some devices come with Bluetooth turned on by default so users need to check this setting.

If Bluetooth must be enabled, the user can set the device to be hidden (analogous to not broadcasting the network name on a wireless network). Setting a device to be invisible will still allow Bluetooth communications to function but will only allow connections to trusted devices that have been previously configured. This protection is not perfect, however; if an attacker finds out that a particular device is trusted, they can use their own Bluetooth device to masquerade as the trusted device and will then be able to connect to the target phone (this is a common spoofing attack).

If a user must use Bluetooth, they should also only turn it on as needed. In addition, users should change their Bluetooth personal identification number (PIN) every month or so. Changing the PIN requires that any Bluetooth devices that the user regularly employs will need to be re-paired, but it also makes it a bit harder for attackers. Attacks succeed because many users will balk at constantly turning their Bluetooth port on and off, or changing the PIN, but at the very least users should change the default PIN when they first get their Bluetooth enabled device (Jansen & Scarfone, 2008).

7. CONCLUSION

The intent of this project was to determine how real the threat is of attacks to Bluetooth-enabled devices and how easy such attacks are to launch. After spending a relatively short amount of time and a few dollars, it is clear how

vulnerable Bluetooth technology really is. The idea that someone could listen to all conversations a victim is having without them even knowing, or have their text messages read, are key examples of the dangers of Bluetooth. Even worse, an attacker can initiate a call to someone or text someone without the victim ever knowing. The only way a user would be able to catch this activity is if they were to look through their call log or look at the sent messages on their phone. Even that might be insufficient, as the attacker can delete the records of their nefarious activity and the victim would never know until their bill comes out. The victim would only know about unusual behavior if they carefully look at their bill, which is increasingly problematic since many people do not even look at their detailed call records. And even if someone complains that they "did not make a call on this date and time," the mobile service carrier has proof that the call was made from this device because, indeed, it was.

Users need to be made aware of the vulnerabilities of these devices so that they can employ them more effectively, safely, and confidently.

ACKNOWLEDGEMENTS

This work was partially supported by Grant No. 2006-DD-BX-0282 awarded by the Bureau of Justice Assistance. The Bureau of Justice Assistance is a component of the Office of Justice Programs, which also includes the Bureau of Justice Statistics, the National Institute of Justice, the Office of Juvenile Justice and Delinquency Prevention, and the Office for Victims of Crime. Points of view or opinions in this document are those of the author and do not represent the official position or policies of the United State Department of Justice.

AUTHOR INFORMATION

Dennis Browning received his B.S. degree in Computer & Digital Forensics from Champlain College in May 2009 and currently works in the Information Technology Department at Fletcher Allen Health Care in Burlington, Vermont.

Gary C. Kessler, Ed.S., CCE, CISSP, is an Associate Professor, director of the M.S. in Digital Investigation Management program, and principle investigator at the Center for Digital Investigation at Champlain College. He is also an adjunct associate professor at Edith Cowan University in Perth, Western Australia.

REFERENCES

Bluebugging. (n.d.). trifinite.stuff Web site. Retrieved January 27, 2009, from http://trifinite.org /trifinite_stuff_bluebug.html

Bluejacking. (2009, January 6). Wikipedia. Retrieved January 27, 20909, from http://en.wikipedia.org/wiki/Bluejacking

Bluesmack. (n.d.). trifinite.stuff Web site. Retrieved January 27, 2009, from

http://trifinite.org /trifinite_stuff_bluesmack.html

BlueSnarf. (n.d.). trifinite.stuff Web site. Retrieved January 27, 2009, from http://trifinite.org /trifinite_stuff_bluesnarf.html

Bluesnarfing. (n.d.). Bluejacking Tools: The Biggest Collection of Bluetooth Tools on the Internet Web site. Retrieved January 27, 2009, from http://www.bluejackingtools.com /bluesnarfing/

Bluetooth SIG. (2008a). How Bluetooth Technology Works. Bluetooth.com Web site. Retrieved January 6, 2009, from http://www.bluetooth.com/Bluetooth/Technology/Works/

Bluetooth SIG. (2008b). Security. Bluetooth.com Web site. Retrieved January 6, 2009, from http://www.bluetooth.com/Bluetooth/Technology/Works/Security/

D3scene. (2008, April 30). BTInfo. Retrieved January 29, 2009, from http://www.d3scene.com /forum/general-mp/13279-btinfo.html

E-Stealth.com. (2008). Ultimate Bluetooth Mobile Phone Spy Software User Manual. Retrieved January 29, 2009, from http://www.jamsa.us/inventory/UltimateMobilePhoneSpyManual.pdf

Getjar. (2008, March 10). STM Bluetooth Software and Tools. Retrieved January 29, 2009, from http://www.getjar.com/products/8042/STMBlueS

Gusev, A. (n.d.). Object Exchange (OBEX) Protocol Primer. Developer.com Web site. Retrieved January 29, 2009, from http://www.developer.com/ws/article.php/3573636

Helomoto. (n.d.). trifinite.stuff Web site. Retrieved January 27, 2009, from http://trifinite.org /trifinite_stuff_helomoto.html

Hole, K.J. (2007, March 2). Bluetooth -- Part 3: Link Controller and JSR-82 API Architecture. Retrieved January 29, 2009, from http://www.kjhole.com/Standards/BT/BT-PDF /Bluetooth3alt.pdf

Hole, K.J. (2008a, February 24). Bluetooth -- Part 1: Overview. Retrieved January 29, 2009, from http://www.kjhole.com/Standards/BT/BT-PDF/Bluetooth1alt.pdf

Hole, K.J. (2008b, March 8). Bluetooth -- Part 10: Introduction to Wireless Security. Retrieved January 29, 2009, from http://www.kjhole.com/Standards/BT/BT-PDF/Bluetooth10alt.pdf

Hole, K.J. (2008c, March 8). Bluetooth -- Part 4: Link Manager and J2ME Programming. Retrieved January 29, 2009, from http://www.kjhole.com/Standards/BT/BT-PDF /Bluetooth4alt.pdf

Hole, K.J. (2008d, March 11). Bluetooth -- Part 6: Logical Link Control and Adaptation Protocol. Retrieved January 29, 2009, from

http://www.kjhole.com/Standards/BT/BT-PDF /Bluetooth6alt.pdf

Hole, K.J. (2008e, March 23). Bluetooth -- Part 7: RFCOMM. Retrieved January 29, 2009, from http://www.kjhole.com/Standards/BT/BT-PDF/Bluetooth7alt.pdf

Hole, K.J. (2008f, March 29). Bluetooth -- Part 8: The JSR-82 API for Device Discovery. Retrieved January 29, 2009, from http://www.kjhole.com/Standards/BT/BT-PDF /Bluetooth8alt.pdf

Java Community Process (JCP). (2009). JSR 82: Java APIs for Bluetooth. Community Development of Java Technology Specifications Web site. Retrieved January 27, 2009, from http://jcp.org/en/jsr/detail?id=82

Jansen, W., & Scarfone, K. (2008, October). Guidelines on Cell Phone and PDA Security. National Institute of Standards and Technology Special Publication 800-124. Retrieved February 24, 2009, from http://csrc.nist.gov/publications/nistpubs/800-124/SP800-124.pdf

Laurie, A., Holtmann, M., & Herfurt, M. (2006, March 30). Bluetooth Hacking. WEBSEC 2006, London, U.K. Retrieved January 27, 2009, from http://trifinite.org/Downloads /trifinite.presentation_websec2006.pdf

Mahmoud, Q.H. (2003a, February). Wireless Application Programming With J2ME and Bluetooth. Sun Developer Network (SDN) Web site. Retrieved January 27, 2009, from http://developers.sun.com/mobility/midp/articles/bluetooth1/

Mahmoud, Q.H. (2003b, April). Part II: The Java APIs for Bluetooth Wireless Technology. Sun Developer Network (SDN) Web site. Retrieved January 7, 2009, from http://developers.sun.com /mobility/midp/articles/bluetooth2/

SE-NSE. (2006, November 5). MiyuX. se-nse v5 Web site. Retrieved January 29, 2009, from http://forums.se-nse.net/index.php?showtopic=5653

Sridhar, T. (2008, December). Wi-Fi, Bluetooth, and WiMAX. The IP Journal, 11(4), 2-17.

Tutorial-Reports.com. (n.d.). Bluetooth Tutorial: Protocol Stack. Retrieved January 28, 2009, from http://www.tutorial-reports.com/wireless/bluetooth/protocolstack.php