Assessing Bluetooth Security Risks in Public Places

Betty Lauer
Urip Mangkusubroto
George Pipkov
Pat Steinhardt
Harvard Extension School

Abstract

Bluetooth enabled devices are rapidly increasing in both new device types and the total number of devices used. They hold data and information that all of us view as private. While the Bluetooth specification has made great effort to design security into all Bluetooth devices, we believed that the weak spot would be the fact that they need to be configured by their users for maximum security. With this weakness in mind, we set out to assess the real world occurrence of devices configured with the security of the device, and the data resident on the device, at risk of a Bluetooth attack. What we found using a variety of testing tools is that there are indeed many devices left vulnerable. Anyone mounting such an attack could easily and anonymously locate these devices. Several proof of concept tools are available that have demonstrated successful attacks. The combination of available tools to launch a Bluetooth attack and the easy access to vulnerable devices as a result of this research, indicate a very real Bluetooth security risk. Regardless of the built in theoretical security of Bluetooth devices there is a very real world risk of us seeing successful Bluetooth attacks in the future when user's do not configure their devices for maximum security or when manufacturers do not implement optimal security.

1. Introduction

At the end of 2005, over 500 million Bluetooth-enabled devices had been sold. Over five million new Bluetooth-enabled devices are sold every week [1]. Many of these devices contain personal information about the owner and/or confidential information intended for the owner's personal use. This information could include the owner's name, phone number, mailing address, and phone contact list. As more devices provide email connectivity, the information stored on the devices could also include personal and business email contacts and the information in the emails themselves. What security risks do the millions of users of Bluetooth-enabled devices face? What information is "easily" found, and what information is relatively secure? In order to answer these questions, we:

- Surveyed what type of information is easily determined about Bluetooth devices by running two Bluetooth scanning applications
- Developed an application to determine how difficult it would be to pair with other devices
- Researched known Bluetooth security vulnerabilities and best-practices for securing devices

2. Related Work

Fromits early beginnings in 1998, the Bluetooth Special Interest Group (SIG) [2] designed the Bluetooth specification with security in mind. Even with the specification's resolve to ensure the security of Bluetooth enabled devices, there have been several successfully demonstrated security attacks.

In 2003, Ollie Whitehouse coined the term "War-Nibbling" to map the location of Bluetooth devices within an organization [3]. He was able to show that discoverable devices within range were easily picked up by simply using commands for the Bluetooth interfaces. He was also able by brute force of the last six bytes of the Bluetooth address to pick up non-discoverable devices. This tool's source code is available to anyone and is named Redfang [4]. An actual attack of these devices at this point was hard to implement because one would have had to pick up data while being transmitted. It did however expose the fact that the devices were exposed if an attack was to take place.

Yakiv and Wool [5] went a step further than Whitehouse when they implemented a practical demonstration of a Bluetooth attack similar to Whitehouse's. Yakiv and Wood's did not require the attacker to listen in on the original connection but instead required the two devices once discovered to repeat the pairing process allowing the attacker to pick up the PIN. As Bluetooth devices frequently require the user to repeat the PIN code this was not thought to be unusual behavior for a typical Bluetooth device user. By forcing the pairing process and then using the information gathered Yakiv and Wool were able to crack a four digit PIN in 0.06-0.3 seconds.

The work closest to what we are doing was done by Adam Laurie [6]. He called it "bluestumbling". In his work he exposed vulnerable devices along with several additional Bluetooth attacks that have spawned the development of tools to mount a variety of Bluetooth attacks including Bluejacking [7], Bluesnarfing [8] and other possible attacks.

Most successful attacks have been proof of concept attacks demonstrating that the attack is possible. We believe this research to be unique as we are providing real world data on the prevalence of vulnerable Bluetooth devices in public places such as airports, malls and coffee shops.

3. Bluetooth Vulnerability Assessment

With the two goals in mind of assessing the prevalence of vulnerable Bluetooth devices in public venues and attempting to pair with any of these devices we employed several tools. An original application, a free tool available for download on the internet for use on Windows XP machines named Bluescanner [12] and an Apple Mac OS X "out of the box" utility application named *Bluetooth File Exchange* (BFE).

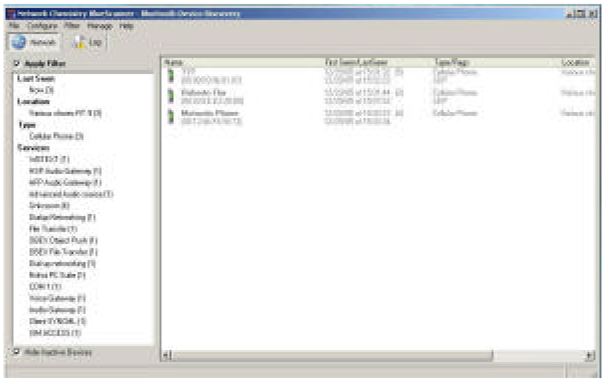
With these applications running on Bluetooth enabled laptops we were able to anonymously gather data in venues targeted as simply having a great number of people. Targeted venues included airports, mall, coffee shops and others where one would expect many people just going about their daily lives.

3.1 Bluescanner

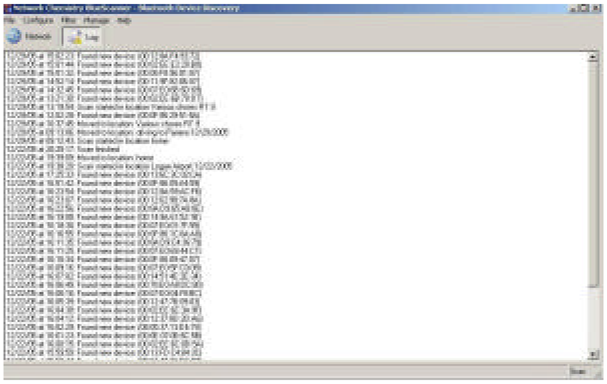
Bluescanner is a free tool available from Network Chemistry [9] that runs on Windows XP. This utility was designed with the objective of allowing organizations to expose any Bluetooth vulnerabilities within their organization. The tool identifies all discoverable devices within the range of the Bluetooth device and records the information it can gather without pairing with them. The data it collects includes the device's human friendly name, unique address, type, time of discovery, time last seen and any Bluetooth Service Discovery Protocol (SDP) information the device provides.

There are two main windows during the operation of the application that allow you to see either the detail view of the results or the log of the session. A data file is also recorded that allows you to keep track of monitoring sessions over time.

The detail view:

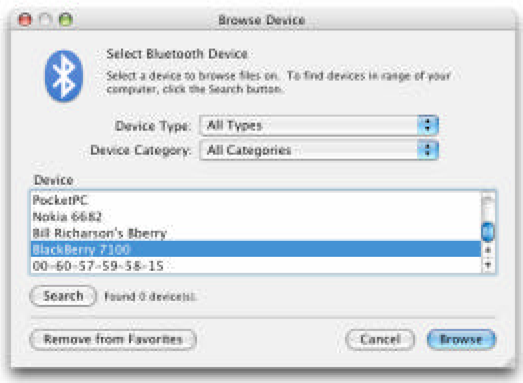


The log view:



3.2 Bluetooth File Exchange

The Orange group used an Apple Mac OS X "out of the box" utility application named *Bluetooth File Exchange* (BFE). This application was use to gather Bluetooth data in public places. The data that can be gathered using *Bluetooth File Exchange* are: the device name; address; type; and available services. BFE does not report the major and minor class for each device like Bluescanner does. The main window for BFE is shown here:



Select Bluetooth Device

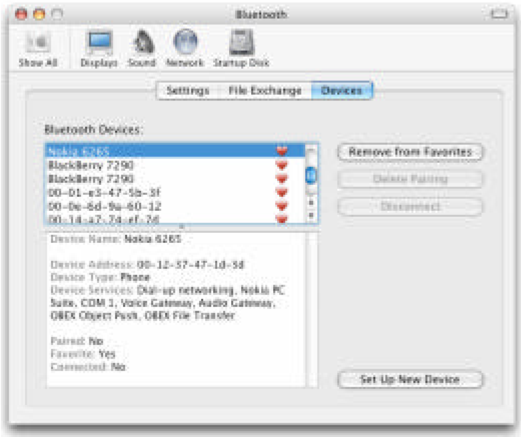
Select a device to browse files on. To find devices in range of your computer, click the Search button.

Device Type: All Types
Device Category: All Categories

Device
PocketPC
Nokia 6682
Bill Richarson's Bberry
BlackBerry 7100
00-60-57-59-58-15

Found 0 device(s).

The single purpose of this application is to detect any Bluetooth devices in the area and to connect to the other device. If the application detects a discoverable device, to see detailed information about the device, the user must add (save) the detected device as a favorite in the operating system Bluetooth preference window:



Bluetooth Devices:
Nokia 6265
BlackBerry 7290
BlackBerry 7290
00-01-e3-47-5b-3f
00-0e-6d-9a-60-12
00-14-a7-74-ef-7d

Device Name: Nokia 6265

Device Address: 00-12-37-47-1d-3d
Device Type: Phone
Device Services: Dial-up networking, Nokia PC Suite, COM 1, Voice Gateway, Audio Gateway, OBEX Object Push, OBEX File Transfer

Paired: No
Favorite: Yes
Connected: No

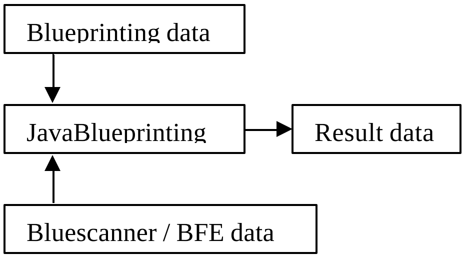
In our experiments, consisting of three executions of BFE in two different locations, 44 devices were detected. Of that total, BFE was able to gather detailed information on three devices (two computers and one phone). The other devices were not verified by BFE; probably because these devices had already moved out of range.

In a controlled experiment between 2 computers, BFE was able to connect to the other computer and browse files on that computer, without any pairing or acknowledgment. This did require that the Bluetooth settings on the target system have all security features turned off. However, this does show that malicious attacks and/or theft of data can occur when Bluetooth is enabled and improperly configured on a device.

3.3 Java Blueprinting Analysis Tool

Blueprinting [10] is an application developed by trifinite.org to "blueprint" Bluetooth device using the device address and services. The group tried using this tool, but found it required technical expertise beyond that available to the group. Another application, similar to the Blueprinting application, is @stake's RedFang [4]; a tool that determines undiscovered device type using only the device address. The Orange group developed an application combining much of the functionality of both of these applications but that used the output from the Bluescanner application and BFE as it's input.

Our blueprinting application uses data from RedFang to determine detailed device type information from a Bluetooth address.



The reason this application was developed was to help refine the type of devices being detected. This allowed us to identify specific device models that may be vulnerable to the SNARF attack. A list of Bluetooth devices susceptible to such attacks was found in an article from thebunker.net.

The application analyzes data provided from Bluescanner and BFE, and determines manufacturer and model information. It does not use the device name to do this; although many device names do contain the manufacturer information, most devices allow the name to be changed. Rather, the device address is used. There are specific characteristics to the Bluetooth device address which allow manufacturer information to be determined. A sample of executing the Blueprinting Analysis Tool (BAT) is shown here:

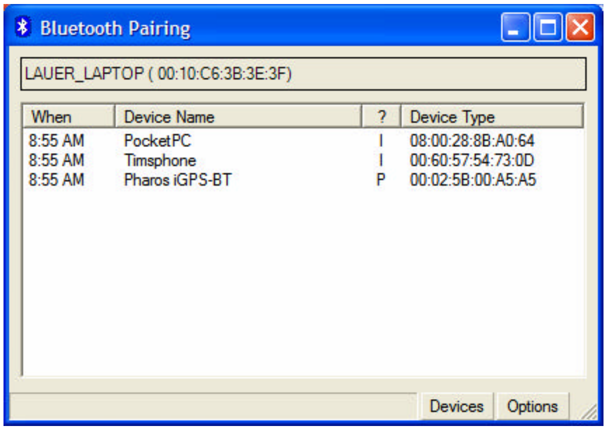
```
$ java JavaBp filename.dat
1. 00:0A:D9:65:5F:5D
        >sony ericsson p800
        >sony ericsson p900
        >sony ericsson t610
        >sony ericsson z600
        >ericsson t68i
2. 00:0F:86:13:D8:E4
        >blackberry 7290
```

3.4 BtPairing Application

The Orange group developed a Windows application in Visual Basic .NET 2003, utilizing the .NET framework. This application, named BtPairing, is intended to provide the user with a tool to:

1. Track up to 100 Bluetooth-enabled devices per execution, displaying when the device was last seen, the name of the device, the Bluetooth address of the device, whether the device has been paired or not, and additional device information. This information is displayed in the application window, in a log file, and can be exported to a .csv file.
2. For each new device found, use the rules stored in an application-specific database to determine how to attempt to pair with the device.
3. Provide a simple way to specify a series of PIN values to use to attempt to pair with the device. This gives the user the ability to specify some common PIN values to use before attempting a "brute-force" pairing described next.
4. Potentially attempt a "brute-force" pairing with the device. This means to attempt all possible 4-digit PIN values, and then all possible 8-digit PIN values, until pairing is accomplished.

When started, the application window looks like this:


Bluetooth Pairing

LAUER_LAPTOP (00:10:C6:3B:3E:3F)

When	Device Name	?	Device Type
8:55 AM	PocketPC	I	08:00:28:8B:A0:64
8:55 AM	Timsphone	I	00:60:57:54:73:0D
8:55 AM	Pharos iGPS-BT	P	00:02:5B:00:A5:A5

The main components of the window are:

1. The name and Bluetooth address for the device running the application
2. A list of all of the discovered Bluetooth devices. The list has the most-recently discovered devices at the top. The columns contain:

Column	Contains
When	The time the device was last seen (note: if you run the application for more than a day, this will be the date if the device was not found today)
Device Name	The name as presented by the Bluetooth device.
?	The "status" of this device within the application. This is currently a one-character value and will be: • " " (space) – the application has not done any pairing with this device • I – based on rules, pairing for this device was Ignored • P – successful pairing was completed • X – a rule gave a PIN that was not accepted
Device Type	The Bluetooth address of the device, and any additional information about the device that can be determined.

3. A status bar, where there is a status message that is periodically updated, a "Devices" button, which displays a dialog box where device pairing rules are maintained, and an "Options" button, which displays a dialog box where various other information is displayed and actions may be performed.

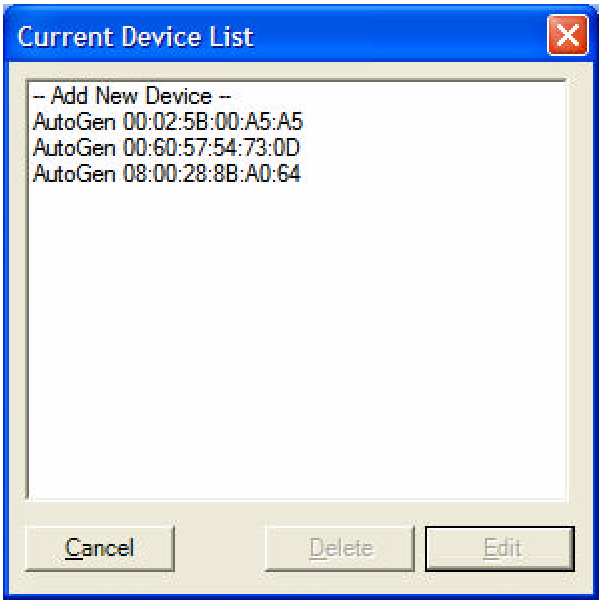
The application will scan for new Bluetooth devices every 30 seconds. When a new device is discovered, the following steps will be taken:

1. All rules will be scanned for a match in the following order:
   a. Address rules
   b. Device-Type rules
   c. Device-Name rules

   The first rule, and only the first rule, that matches will be used.
2. If no rule is found, then the "Quick PIN" list will be used for pairing. If there is no "Quick PIN" list, this step is skipped. If there is a "Quick PIN" list, then each PIN found in the list, in the order found in the list, will be used to attempt pairing.
3. If no rule is found, and Quick PIN pairing was not successful, and the "Try All PINs" option is checked, then each PIN is tried, starting at 0000 and going to 9999, then all 8-digit PINs are tried.
4. If a rule is found, then the PIN associated with that rule is used for pairing.

More information on defining rules is presented later in this document.

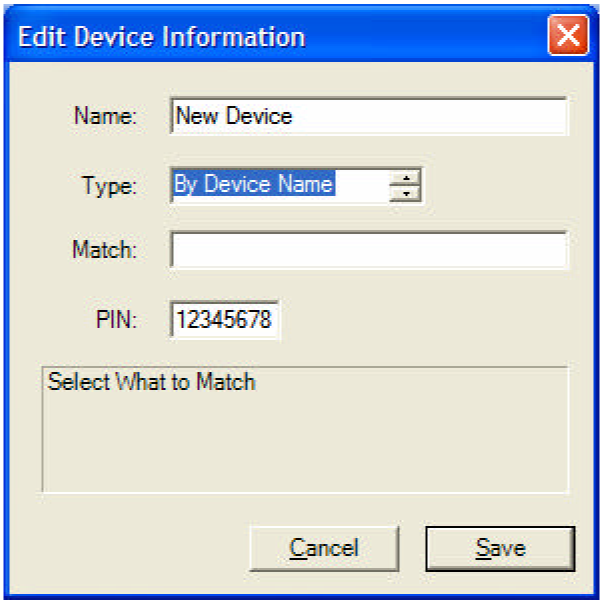
The "Devices" button brings up the following dialog box:


Current Device List

-- Add New Device --
AutoGen 00:02:5B:00:A5:A5
AutoGen 00:60:57:54:73:0D
AutoGen 08:00:28:8B:A0:64

This is a list of all of the rules currently defined to the application. Rules are stored in an external file and are therefore "remembered" across application executions.

To create a new rule, select the "—Add New Device –" item, then click "Add". This will cause the following dialog to be displayed:


Edit Device Information

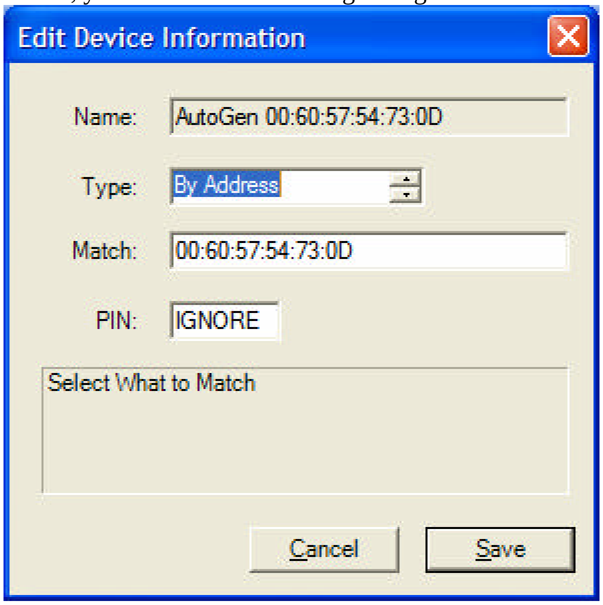
Name: New Device
Type: By Device Name
Match:
PIN: 12345678

The information you need to enter is:

Label	Description
Name	The unique name you wish to use to identify this entry. It is the name that will be displayed in the "Current Device List".
Type	Defines what will be matched – either the Device Name, Device Class, or Device Address. This determines what the next value will contain
Match	A regular-expression pattern used to match specific information on a Bluetooth device that has been discovered
PIN	The 1-8 digit PIN code that will be used to attempt pairing, or "IGNORE" to indicate that pairing should not be done on matching devices

After entering all information, select the "Save" button to save this entry into the database. You will return to the "Current Device List". The new device entry will be used the next time Bluetooth scanning is done.

If you select an existing Device entry, and click the Edit button, you will see the following dialog box:


Edit Device Information

Name: AutoGen 00:60:57:54:73:0D
Type: By Address
Match: 00:60:57:54:73:0D
PIN: IGNORE

Note that the name of the entry is not editable; this is by design. Once you have named an entry, it will always have that name. However, you may change any or all of the other information for this entry, and then select "Save" to cause the database to be updated.

If the application either successfully pairs with a device, or cannot pair with a device after trying to, it will create a rule on that device address. The name of the entry will be “AutoGen” followed by the device address. The entry will be a “By Address” entry, and the PIN will either be the PIN found or “IGNORE” if no PIN worked. This will prevent the application from repeating work already done.

The device entries define in the application database may be thought of as “pairing rules”, in that each device entry defines if pairing should be done on a group of devices, and whether pairing should be done or what PIN should be used.

Some examples of entries that may be useful:

To Completely Disable Pairing	
Name	Never Pair
Type	By Device Address
Match	.*
PIN	IGNORE

Note: this will effectively change the behavior of the application to be a Bluetooth scanner.

To Pair with the Pharos GPS	
Name	Pharos GPS
Type	By Device Name
Match	Pharos iGPS-BT
PIN	12345678

This is a specific device (although it would match ALL of these GPS devices, as the name cannot be changed).

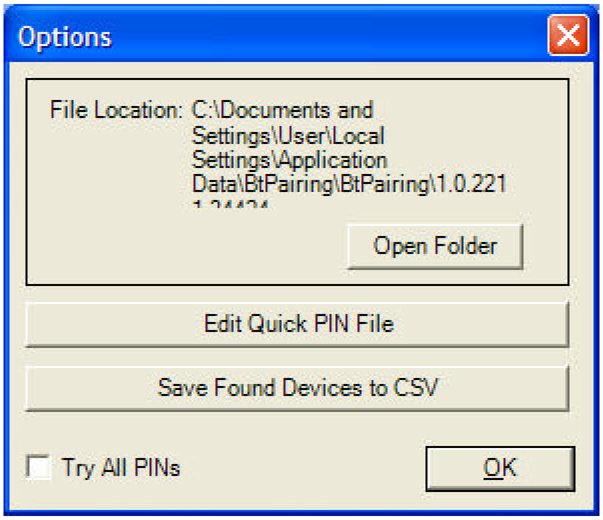
To Ignore Cell Phones	
Name	Ignore Cell Phones
Type	By Device Class
Match	CellPhonePhone
PIN	IGNORE

CellPhonePhone is part of the “class” of a device. The values available vary based upon the device; some values may be seen in the Bluetooth log file.

To Try To Pair With PDA’s	
Name	PDA’s
Type	By Device Class
Match	Handheld Computer
PIN	1111

Note that the regular expression pattern in the “Match” value is not case sensitive and may contain any characters valid in a regular expression. For examples of regular expressions, see the online manual at http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpguide/html/cpconregularexpressionsaslanguage.asp.

From the application window, clicking the “Options” button brings up the following dialog box:



Options

File Location: C:\Documents and Settings\User\Local Settings\Application Data\BtPairing\BtPairing\1.0.221

□ Try All PINs

The first box of information, labeled “File Location”, displays the folder in which the files used by the application are stored. The application will use the following files:

File	Purpose
bluetooth.log	Contains information on devices as they are found and as pairing is attempted.
bluetooth.xml	Contains the device database stored in XML format.
bluetooth.csv	If the list of found devices is stored in CSV format, then this file exists, and contains the information in CSV format. The columns in the CSV file are: Datetime, DeviceAddress, DeviceName, DeviceClass. Each value is enclosed in quotes.
pins.txt	A text file containing a series of PIN values to be used in pairing. The PINs listed are attempted in the order found. There may be multiple PINs per line, separated by spaces or commas, and there may be as many lines as needed

The “Edit Quick PIN File” button will cause the Quick PIN text file to be opened in the application associated with .txt files (more than likely, Notepad). As soon as the file is saved, the Quick PIN values are loaded from the file. The Quick PIN file format is described above, in the description of each file. Note that there is no Quick PIN file unless you create one.

The "Save Found Devices to CSV" button will cause the list of all devices currently found to be written, in CSV format, to the bluetooth.csv file. Any existing file will be overwritten. This is intended to be used in case the information needs to be imported into another application.

The "Try All PINs" checkbox controls whether or not every possible PIN value should be used if there are no matching Device Entry rules and none of the Quick PIN values worked. If this is checked, then the application will go through all PIN values starting with 0000 and going to 9999 and then 00000000 to 99999999, attempting to pair with the device with each PIN. If a PIN is found with which pairing works, no further PIN values are tried.

Checking this box can cause a large amount of time to be used pairing to a device. Also, in the case of devices where pairing attempts must be confirmed, the user of the device being paired will be prompted to accept or reject each PIN. However, if you have a device like a GPS, this option can be useful in determining what the default PIN is. This option is not checked by default.

The application is available for downloading at http://btpairing.home.comcast.net/. The application is provided "as is". There is no warranty or guarantee. If you download the application and run it, you do so at your own risk. It is possible that running this application may be illegal. As the authors discovered during development, it may be very annoying to those around you with Bluetooth devices if you run this device with "Try All PINs" enabled.

There are several improvements already slated for the application. If you have other suggestions, or any other comments about the application, please email btpairing@comcast.net. The suggestions already under review are:

1. Allow the frequency of discovery to be configurable. It is currently set for every 30 seconds after the last one completes.
2. Change the "?" column character to be an icon more representative of what the status is.
3. Add more information on the list of found devices page, specifically in the Device Type column.
4. Make the names of things more consistent and representative; for example, change "Device Entry" to "Pairing Rule" in all places.
5. Provide some default rules and/or Quick PIN codes in the deployment.
6. Allow for a more extensible pairing rule mechanism; perhaps involving the ability to use other .NET classes that follow a defined API.

4. Results

4.1 Bluescanner

This section describes the data that the team collected using Bluescanner. It then presents statistical analysis that seeks to uncover correlations between the level of security of the Bluetooth devices detected by the team and the devices' type and brand.

Team members made trips to busy public places, such as airports, shopping malls and coffee shops, and scanned for detectable Bluetooth devices. A team member then put together a data set with 89 observations. For each observation the following binary variables were coded: level of security (0 for no link level security, and 1 for existing link level security), device type (cell phone, smart phone, laptop), device brand (Nokia, Sony Ericsson), and user friendly device name.

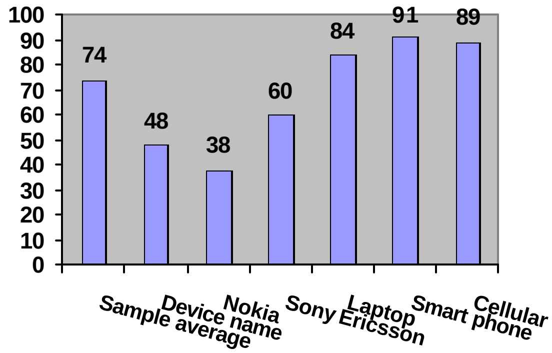
To identify which device characteristics are associated with the presence of link level security, a maximum likelihood logit estimator was used, with "level of security" as a binary dependent variable. The model results are summarized below:

Independent Variable	Coefficient (Standard Error)
Device name	-1.90*** (.58)
Cellular phone	2.14* (1.17)
Smart phone	2.21** (1.11)
Laptop computer	1.09 (1.09)
Nokia	-1.88** (0.78)
Sony Ericsson	-.68 (.98)
Constant	.47 (.99)
N=98, pseudo R^2=.24, Log-likelihood= -42.25, $?^2$=26.34	

Note: *Significant at <.1 one-tailed test
** Significant at <.05 one-tailed test
*** Significant at <.01 one-tailed test

The model suggests that the existence of a user-friendly device name is associated with a lower probability of link level security. It also suggests that smart phones are more likely than other devices to have link level security. Finally, the data raises the possibility that Nokia devices are less likely than other devices to have link level security.

To assess the magnitude of these effects, the predicted probability of link level security is calculated for different types of devices, using the free Clarify 2.1 software, created by Gary King, Michael Tomz, and Jason Wittenberg (http://gking.harvard.edu/stats.shtml). The following chart compares the probability of having link level security for different devices.



The conclusions that we can draw from this model are only tentative, because of the small sample size, and because the included independent variables can explain only about a quarter of the variation in the dependent variable. However, these results point to avenues for further research. For example, scholars can further examine the purported vulnerability of Nokia devices compared to other brands (in our sample Nokias are only half as likely, on average, to display link level security).

It would be especially interesting to investigate the association between a user friendly device name and a lower level of security. The statistical model, of course, shows correlation, rather than causation, and it would be nonsensical to claim that the existence of a device name itself *leads* to lower security. However, it seems plausible to argue that device name is a proxy for less technically savvy users. In other words, our model confirms the hypothesis that users who tinker with the settings of their devices, usually end up compromising the device's security. Thus, our model results have implications for the important debate in security studies on user education.

4.2 Java Blueprinting Analysis Tool

The results from using the BAT program are shown here:

Device Type	Bluescanner	BFE
Sony Ericsson p800/p900/t610/ Z100/t68i	8	1
Sony Ericsson k700i/t630	6	3
Blackberry 7290	9	14
Motorola A1000	3	0
Nokia 6820/7650/ 6630/7820/6620	7	3
Nokia 6230	1	0
Nokia 3230	1	2
Nokia 6230/6630/ 7610	2	0
Nokia 6320/7610	1	0
Nokia 6310/3650/ 7600/ngage/8910/ ngageV3/6600	1	2
Siemens sx1/s65/ S55/ Comneon h1	0	1
Unknown	27	16
Known	**57%**	**62%**

4.3 BtPairing Application

If we disable the active pairing features of the BtPairing device, it will generate results very similar to the Bluescanner application, in that it will create information on the address, name, and type of every Bluetooth-enabled device it discovers. A sample of the information stored by the BtPairing application is shown here:

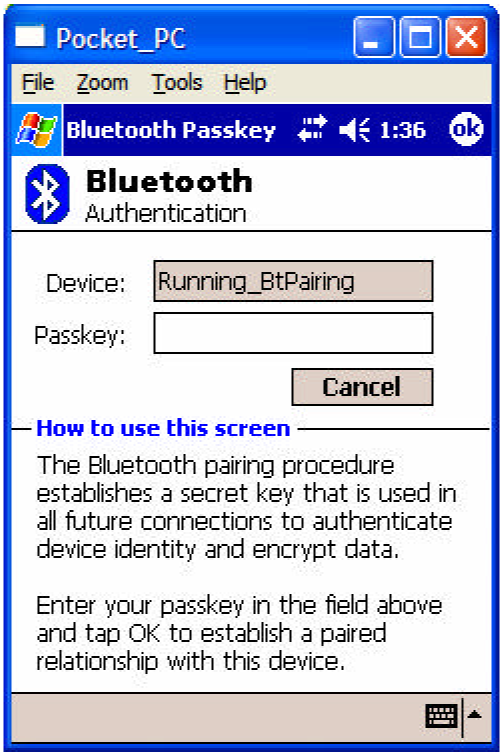
```
"2006-01-20
20:22:46","00:02:5B:00:A5:A5","Pharos
iGPS-BT","UnclassifiedDevice",
"2006-01-20
20:22:46","00:60:57:54:73:0D","Myphone","C
ellPhonePhone, ObexService, TelephonySer-
vice",
"2006-01-20
20:22:46","08:00:28:8B:A0:64","08:00:28:8B
:A0:64","HandheldComputer, NetworkService,
ObexService",
```

The columns are the date and time, the device address, the device name, and any additional device characteristics discovered.

Where the BtPairing application would have provided additional information would be if it had been able to pair with any of the devices. However, the team was not able to gather any statistically relevant information on pairing.

5. Discussion of the Results

One of the first results we found with using the BtPairing application is that the vast majority of communication devices (cellular phones and PDA's) by default require the phone or PDA user to validate a pair request. Typically, the request is a prompt to enter the same PIN as the device attempting to pair has used. A sample of such a screen is shown below



Bluetooth Authentication

Device: Running_BtPairing

Passkey:

The Bluetooth pairing procedure establishes a secret key that is used in all future connections to authenticate device identity and encrypt data.

Enter your passkey in the field above and tap OK to establish a paired relationship with this device.

Note that the screen shot is from a PDA running Windows Mobile being viewed on an XP system; the actual PDA screen is within the window titled "Pocket_PC".

This means that the BtPairing device will have to wait until either the device goes out of range or the user selects "Cancel"; it is unlikely that the user would know the PIN being used by the application. Also, this means that "brute force" pairing with these types of devices is, practically, impossible, since the user would need to select "Cancel" many times before a pairing would take place. Although we were not able to test on a wide range of devices, we did not find any cell phones or PDA's that did not have this level of security enabled by default.

One broad category of Bluetooth devices, though, would be vulnerable to "brute force" pairing attacks. This category is the set of devices that do not provide a user interface, but are rather used as "tools". Devices in this category would include GPS's, headsets, microphones, speakers, and Bluetooth-enabled printers. Although the BtPairing application will in fact "break" in to these devices, it may take such a long amount of time to do so as to be practically impossible. Repeated observations show that about one pairing per second is obtainable between the BtPairing application and an enabled, in-range device. This means that to just go through all of the 4-digit PIN pairing would take almost three hours! (10,000 PINs at 3600 PIN attempts per hour). The likelihood of a publicly available device being stationary for the time period required for a brute-force PIN pairing seems fairly low.

As the device database increases with device information and default PIN information, though, it will be possible for more and more devices to be paired "in the wild". Gathering this database will be done over time. The database that the team collects will be available on the team web site, updated when appropriate. Note that any entries users of the application find useful should be emailed to the team email address for inclusion in the database.

6. Reducing Vulnerability

The following guidelines should be followed by all users of Bluetooth-enabled devices. It may be that some user education on Bluetooth and how it works is also required. Also, many device manufacturers now set the default device settings to those that are most secure; more manufacturers should do the same.

Bluetooth User's Security Guidelines
Whenever possible, disable the ability for your device to be "discovered" by other devices. This will reduce the vulnerability to bluesnarfing, bluebugging, and bluejacking attacks. Note that is may reduce the ability to receive business contacts through the device, since "invisible" devices cannot share information.
Whenever possible, make the name of the device something that cannot be used to identify the device, the owner, or the device location. The name of the device is visible if the device is discoverable.
Make sure the device has the latest version of the oper-

ating system, firmware, etc. installed. Vendors often have many vulnerabilities already solved in the latest versions of the software, but this software may not already be on the device
Do not pair with unknown devices
Do not accept files or other data transmitted from unknown devices.
Test your assumptions. Run an application like BlueScanner or BtPairing and see what type of information your device is making available.

7. Conclusions

Aside from Bluetooth devices with the non-secure security mode, the remaining devices with either service or link level security modes have a real risk of being violated if not configured as non-discoverable if an attacker had the motive and currently available proof of concept tools. The number of devices we were able to pick up indicates that many owners are leaving their devices open to a possible attack. With Bluetooth devices using radio waves there is always the risk of a security breach and users should always be aware of that. Combine this with the fact that the only secret when using Bluetooth devices is the PIN code that the four digit PIN has been shown to be quite quickly cracked and the fact that 50% of used PINS are “0000” [12], the potential for a successful attack we believe is something Bluetooth device owners should be aware of.

The number of Bluetooth devices in use is rapidly increasing. In 2005, the market for Bluetooth enabled devices grew to more than 272 million units, twice the number of devices shipped in 2004 [13]. This rapid increase in the number of Bluetooth enabled devices, combined with what we believed to be the device owner’s naïve understanding of the security risks of leaving their devices in discoverable mode, provided the motive for this research.

As a result of this research we cannot make any estimation as to the percentage of devices being used by individuals that are configured insecurely. We were only able to gather information from the discoverable devices. What we can say is that wherever we went where there were a sufficient number of people, there were always a considerable number of devices that were discoverable. All of these were open to the possibly of having their security violated. We can also state as a result of this research that if someone was trying to launch an attack they could easily target locations for successful attacks.

8. References

[1] “Bluetooth Technology Benefits”, [Online] Available at: http://www.bluetooth.com/Bluetooth/Learn/Benefits/. Accessed [January 15, 2006].

[2] Bluetooth SIG, [Online] Available at: https://www.bluetooth.org/ . Accessed [January 20, 2006].

[3] Ollie Whitehouse, “War Nibbling: Bluetooth Insecurity”, [Online] Available at: http://www.rootsecure.net/content/downloads/pdf/atstake_war_nibbling.pdf . Accessed [January 20, 2006].

[4] Redfang, “Bluetooth Discovery Tool”, [Online] Available at: http://www.securiteam.com/tools/5JP0I1FAAE.html . Accessed [January 21, 2006].

[5] Yaniv Shaked, Avishai Wool,“Cracking the Bluetooth Pin”, [Online] Available at: http://delivery.acm.org.ezp1.harvard.edu/10.1145/1070000/1067176/p39-

shaked.pdf?key1=1067176&key2=4476344311&coll=ACM&dl=ACM&CFID=62797063&CFTOKEN=72880313 . Accessed [January 20, 2006]

[6] Laurie, Adam et al., “Serious flaws in bluetooth security lead to disclosure of personal data”, [Online] Available at: http://www.thebunker.net/security/bluetooth.htm. Accessed [January 4, 2006].

[7] “Bluejacking”, [Online] Available at: http://en.wikipedia.org/wiki/Bluejacking . Accessed [January 20, 2006]

[8] “Bluesnarfing”, [Online] Available at: http://en.wikipedia.org/wiki/Bluesnarfing . Accessed [January 20, 2006]

[9] Network Chemistry, “Bluescanner”, [Online] Available at: http://www.networkchemistry.com/products/bluescanner.php . Accessed [January 20, 2006]

[10] Trifinite Group, “Blueprint”, [Online] Available at: http://trifinite.org/trifinite_stuff_blueprinting.html . Accessed [December 20, 2005].

[11] Security Briefs, [Online] http://www.thebunker.net/security/bluetooth.htm. Accessed [January 20, 2006]

[12] Juha T.Vainio, “Bluetooth Security, [Online] Available at: http://www.niksula.cs.hut.fi/~jiitv/bluesec.html . Accessed [January 20, 2006]

[13] Bluetooth Device Shipments Double in 2005 [Online] Available at: http://www.itnewsonline.com/showstory.php?storyid=2417&scatid=3&contid=3 . Accessed [January 20, 2006]