Performance Measure for Security of Bluetooth Enabled Devices Using BBD

Monu Kumar, Bhagvan Krishna Gupta
Computer science & DIT University
Uttarakhand, India

Abstract—The use of Bluetooth is increasing rapidly because of wireless communication which has many advantages like cable replacement, fast and growing. In this paper we are using BBD to protect Bluetooth enabled devices against the attacks that are possible on such devices. With the help of BBD we can enhance the performance of Bluetooth enabled devices communication between two devices. This paper gives the performance analysis and evolution to execute the programs implemented which are used for securing the communication of Bluetooth enabled devices and we also discuss about the memory required and time to lead and execute the methods, threads and classes. This paper describes some java APIs and packages which are used during the implementation of proposed approach and J2ME provides the Bluetooth java APIs.

Keywords—Bluetooth, Pairing, BBD, java APIs, J2ME,

I. INTRODUCTION

Bluetooth is a short range data transmission and communication technology and now it become more popular because of its use. Now a day it is used by most of the mobile phones for short range communication and for data sharing from one mobile to other without any cost. So many applications has been developed which uses Bluetooth on mobile phones. Apart from the mobile it is used by other electronic devices like car, hand watch, locks etc. By means of using Bluetooth we have a lot of advantages like there is no cost of using the services provided by it and low power consumption. Apart from these we also have some other services which makes easy of the use of mobile phones. There is no requirement of cable to transfer the data with high speed and in less time.

BBD is basically a Bluetooth detector sensor device stand for BlipTrack Bluetooth Detector. It is used to detect the Bluetooth devices in the passing vehicles to calculate the average speed of vehicle and for traffic analysis. It also has an inbuilt GPS receiver to give the exact location of BBD [2].

J2ME stand for Java to Mobile Edition provides some Bluetooth java APIs and packages which help to implement the Bluetooth functionality. It also provides the packages for Bluetooth protocols for data sharing and communication. The APIs and packages are discussed in detail in implementation section.

II. RELATED WORK

Most of the work has already been done on Bluetooth technology. Some technology has many problems with the Bluetooth communication.

In [1], the author publishes a methodology against Men-In-The-Middle attack that is possible on Bluetooth devices implemented with the help of IDS and IPS.

In [13], the author proposed an Out Of Band (OOB) channelling with enhanced security. The benefit of using this method is remove all type of Men-In-The-Middle attacks against the Just Work association model (there are four association models:-Just Work, Numerical Comparison, Pass key Entry and Out-Of-Band).

A. Short description of existing reports related to Bluetooth technology

After the introduction of Bluetooth technology the use of it has been increases and for that the security is needed to keep the information secret. There are so many security methods are introduced with an improved methods then previous one that means each method has some advantages and some problems are also there in that. So regarding the security of Bluetooth devices there are some reports has been published [3]. Some reports are listed below:

Table-1 Description of existing reports

<table>
<thead>
<tr>
<th>Year</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2003</td>
<td>Regarding the protocol used by Bluetooth technology, Adam and Bend published some security laws. Their research results that the Bluetooth technology user losses their private information,[4]</td>
</tr>
<tr>
<td>2004</td>
<td>In that year, the Bluetooth virus was introduced. It was the first Bluetooth virus which was a threat to the Bluetooth technology and it was harmful for Bluetooth user’s private</td>
</tr>
<tr>
<td>Year</td>
<td>Event</td>
</tr>
<tr>
<td>-------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>Jan-2005</td>
<td>Lasco - A mobile malware was introduced which was a self replication worm.[6]</td>
</tr>
<tr>
<td>Apr-2005</td>
<td>Passive attacks on Bluetooth with the help of off-line PIN was introduced.[7]</td>
</tr>
<tr>
<td>Aug-2005</td>
<td>Bluetooth enabled devices like mobile phones was introduced which were used to track other mobile devices inside the vehicles.[8]</td>
</tr>
<tr>
<td>Apr-2006</td>
<td>A large number of devices were left in a visible state that posed the possibility of spread of a Bluetooth worm [9].</td>
</tr>
<tr>
<td>Oct-2007</td>
<td>A technique based on link key cracking was introduced [10].</td>
</tr>
<tr>
<td>2008</td>
<td>Attacks based on pairing protocols on Bluetooth v2.1 was introduced.[11]</td>
</tr>
<tr>
<td>2010</td>
<td>Some security enhancement on the pairing and authentication procedures was introduced on the security of Bluetooth. [12][13].</td>
</tr>
<tr>
<td>2011-2013</td>
<td>There are so many attacks and their countermeasures were introduced during this period. Many security procedures and techniques has been introduces in this period.</td>
</tr>
<tr>
<td>2013-2014</td>
<td>OOB as NFC become popular [13].</td>
</tr>
</tbody>
</table>

**III. HOW PROPOSED APPROACH WORK**

Suppose there are two users having Bluetooth enables devices wants to communicate and also assume that they are paired before. When they open the visibility of Bluetooth and try to connect with each other but they are disabled to connect with each other because of a third Bluetooth devices which are the attaching devices. So they delete the previously exchange keys and try to pair again, at that time attacker detect the common shared keys and connect with both of communicating devices but user has no idea about that they have attacked.

At this time when they want to share something BBD detected all the available devices within the range of BBD. BBD detect the device name, time of detection and MAC addresses of all available devices. This information collected by the BBD within a specified period of time and sends to the cloud server. Cloud server further sends the notification to the communicating devices that there is new device in the coverage. By getting this notification the Bluetooth user get aware and stop the communication. Below figure shows the working of proposed.

![Working of proposed approach](image)

**IV. IMPLEMENTATION OF PROPOSED APPROACH**

The simulation of proposed approach has been implemented using Java technology. Java provides Bluetooth APIs for mobile application and J2ME. With the help of Blip Track Bluetooth Detector we can detect the Bluetooth devices within the coverage area of BBD. We have implemented a program for BBD which can detect the name, time and MAC addresses of available Bluetooth devices

**A. Bluetooth System Requirement**

The Bluetooth enabled devices must satisfy a set of requirements and they also support the Bluetooth protocols used during the communication and data transmission and they also support the Bluetooth profiles [14], they are as follows:

- **RFCOMM**: Radio frequency communication it require Generic Access Profile.
- **SDP**: Service delivery protocol requires Service Discovery Application Profile.
- **L2CAP**: Logical link control and adaptation protocol require Serial Port Profile.
B. Java APIs for Bluetooth Connection:-
Java packages used for master slave connection are as follows:-

1) javax.microedition.io: In short awe call it MID. MID is a profiler which includes networking based on GenericConnection framework from the Connected Limited Device Configuration(CLDC). The CLDC is a framework specification for Java ME applications which describes the basic libraries and some virtual machine features [16]. The class hierarchy of this package is given below [15].

```
java.lang Object
  o javax.microedition.io.Connector
  o javax.microedition.io.PushRegistry
  o java.lang.Throwable
    o java.io.IOException
      o javax.microedition.io.ConnectionNotFoundException
```

Fig. 2. javax.microedition.io class hierarchy

2) javax.bluetooth:- It is a JSR-82 specification implementation for J2SE. It provides Bluecore and Avetana packages. Bluecore is an open-source JSR-82 implementation and Avetana is a commercial JSR-82 implementation. The Class hierarchy is given below [15].

```
java.lang.Object
  o javax.bluetooth.DataElement
  o javax.bluetooth.DeviceClass
  o javax.bluetooth.DiscoveryAgent
  o javax.bluetooth.LocalDevice
  o javax.bluetooth.RemoteDevice
  o java.lang.Throwable
    o java.io.IOException
      o javax.bluetooth.DiscoveryAgentException
      o javax.bluetooth.DeviceException
      o javax.bluetooth.ServiceRegistrationException
  o javax.bluetooth.UUID
```

Fig. 3. javax.bluetooth Class hierarchy

3) javax.obex- Object Exchange Protocol os in short OBEX provides the classes and interfaces required to implement Bluetooth authentication.

C. Java APIs for Bluetooth Connection:-
The simulation of BBD is implemented by using java swing APIs. The most common package used to implement BBD is “javax.swing.” which provide pure java component. Below figure shows simulation implementation of BBD:-

When we click on the start button it will start to detect the Bluetooth name and their MAC addresses. After detecting the Bluetooth when we click on send button it will send the information to the server.

![Fig. 4. Java based simulation of BBD](image-url)
V. IMPLEMENTATION OF PROPOSED APPROACH

The performance of BBD simulation is given based on memory size and with respect to time and it also given based on number on threads and classes loading time.

![Fig. 5. Memory requirement to detect Bluetooth by BBD](image)

Above diagram shows the memory requirement in megabytes with respect to time of detection of Bluetooth devices. When time increases, the memory size increases and the shaded area shows the heap size required and heap size used.

![Fig. 6. Threads, methods and Loaded Classes](image)

Above figure shows the threads, methods and loaded classes at different time instances. The values of threads remain somewhat constant from the very beginning and the value of methods increases slightly while the loaded classes increase more rapidly than the methods along with the time.

VI. CONCLUSIONS & FUTURE WORK

The simulation of BBD is somewhat easy to implement in java and it detect the Bluetooth devices in a fraction of time. So the real time implementation of this approach is helpful for Bluetooth security. It can be further improve the performance of this approach by implementing for real world. After the analysis and evaluation of this simulation approach we can say that it will going helpful for Bluetooth user. It is difficult to install the BBD at each street because of cost but not impossible. The collected information send be the server is store in database for future use. For future work this approach can be implemented for enhance version of Bluetooth used in mobile or other devices.

REFERENCES


[14] Developing Applications with the Java APIs for Bluetooth™(JSR-82).
