2023

Recon 4.2 BLE emo Living AI

IVAN ARBALIEV DIMITROV

Contents

Introduction

This report provides insights into a vulnerability assessment conducted on the EMO Robot, with a focus on its IoT capabilities. The assessment process followed a rigorous approach, examining the EMO Robot's IoT functionality, network connectivity, data transmission, and integration with smart home devices. By analyzing the robot's strengths, weaknesses, and potential risks, valuable insights can be gained to enhance the security of its IoT ecosystem.

This report discusses the tools utilized during the assessment, providing insights into their functionality and how they contributed to uncovering vulnerabilities. The process followed is outlined, detailing the steps taken to analyze the robot's functionality and identify potential risks. The report presents the results obtained from the assessment, including security strengths, identified weaknesses, and potential vulnerabilities.

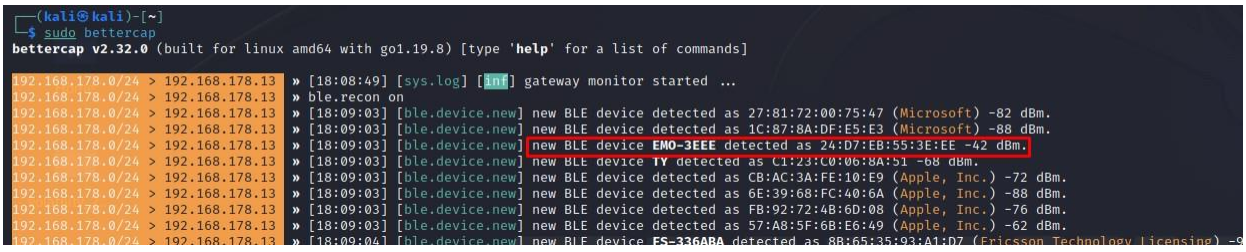
The findings presented in this report aim to improve the overall security posture of the EMO Robot's IoT capabilities, enabling the development of mitigation strategies and enhancing resilience. Understanding the vulnerabilities and risks associated with the EMO Robot empowers red teaming practitioners and security professionals to proactively address potential weaknesses and simulate real-world attack scenarios.

It is important to note that this report focuses solely on the vulnerability assessment of the EMO Robot's IoT capabilities and does not cover its integration with other platforms beyond this scope.

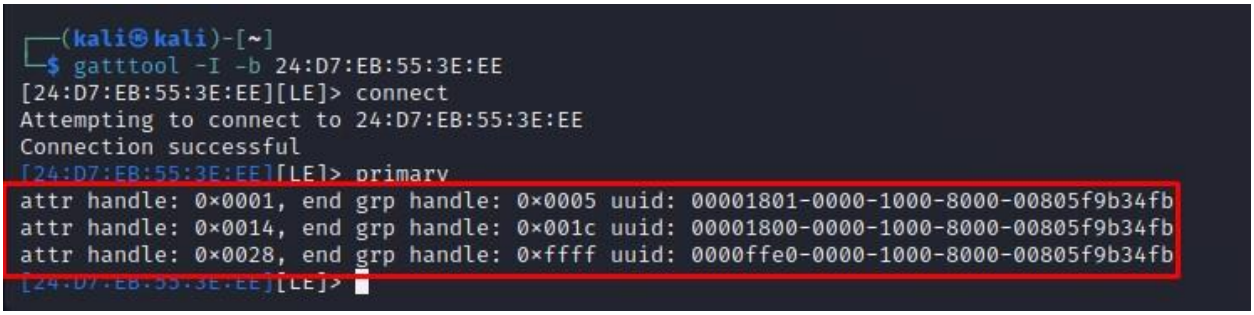
Tools used.

Bettercap is a powerful and versatile tool used for network monitoring, packet manipulation, and penetration testing. While it primarily focuses on Ethernet networks, it also provides robust functionality for Bluetooth network analysis. With its Bluetooth capabilities, Bettercap enables security professionals and enthusiasts to assess and manipulate Bluetooth devices and their communications. One of the key features of Bettercap's Bluetooth module is the ability to discover nearby Bluetooth devices, including smartphones, laptops, headphones, and IoT devices. It employs active scanning techniques to identify devices and collect information such as their MAC addresses, device names, and supported services.

Emo's Mac address: 24:D7:EB:55:3E:EE

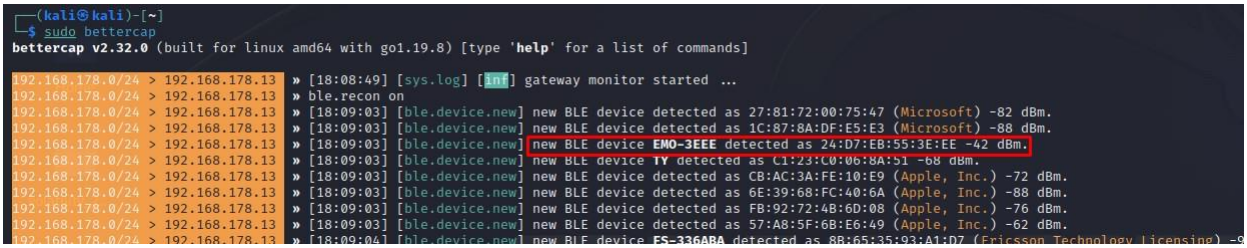


Gatttool is a command-line utility that is part of the BlueZ Bluetooth stack, which is commonly used on Linux-based systems. It provides a way to interact with Bluetooth Low Energy (BLE) devices, allowing users to perform various operations such as device discovery, connecting to devices, reading and writing characteristics, and enabling notifications. With gatttool, users can scan for nearby Bluetooth devices and obtain information such as the device's address, name, and supported services. It allows for establishing a connection to a specific device by specifying its address. Once connected, users can interact with the device's services and characteristics. The tool supports read and write operations on characteristics, enabling users to retrieve data from the device or modify its settings. Additionally, gatttool provides the ability to enable notifications, allowing the user to receive updates whenever a characteristic value changes on the device.

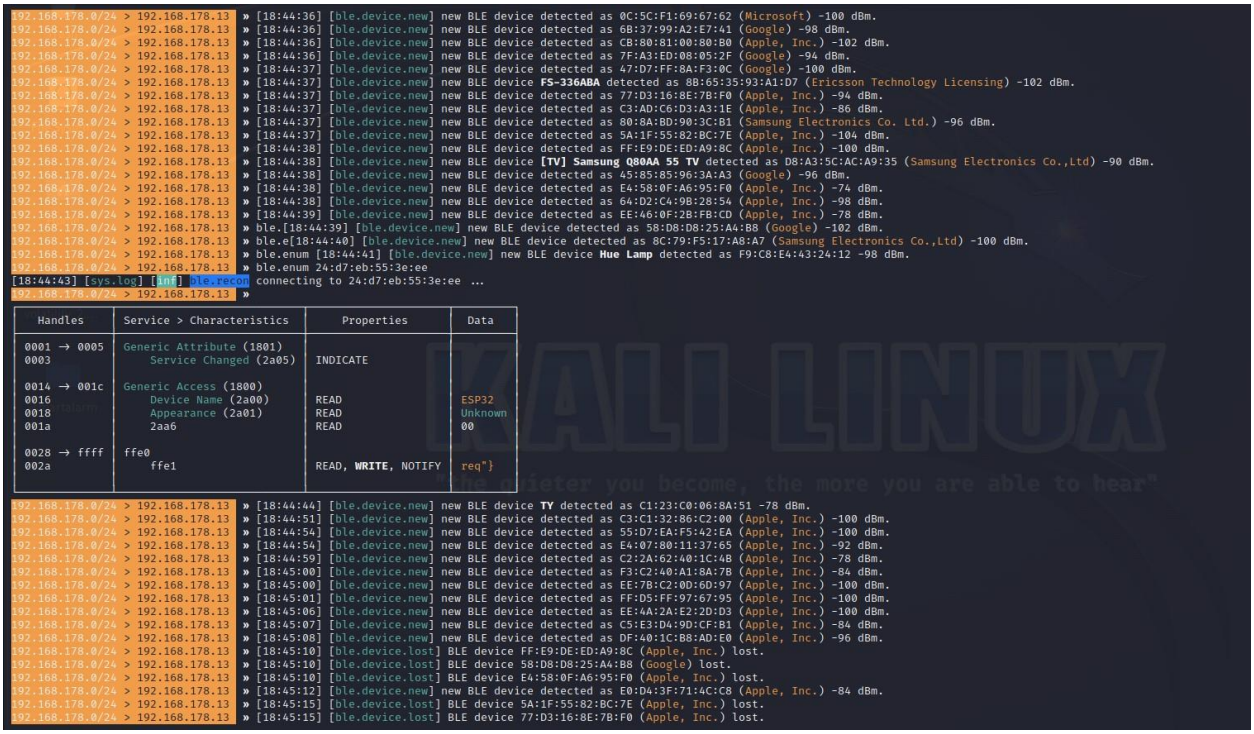


Process and Results

I started my Bluetooth research on Emo by trying to detect his Bluetooth Mac address. I used Bettercap to listen to Bluetooth devices near me. I used a TPlink BT dongle that allows me to monitor BT fields near me. In the screenshot below I am showing the scanning process.



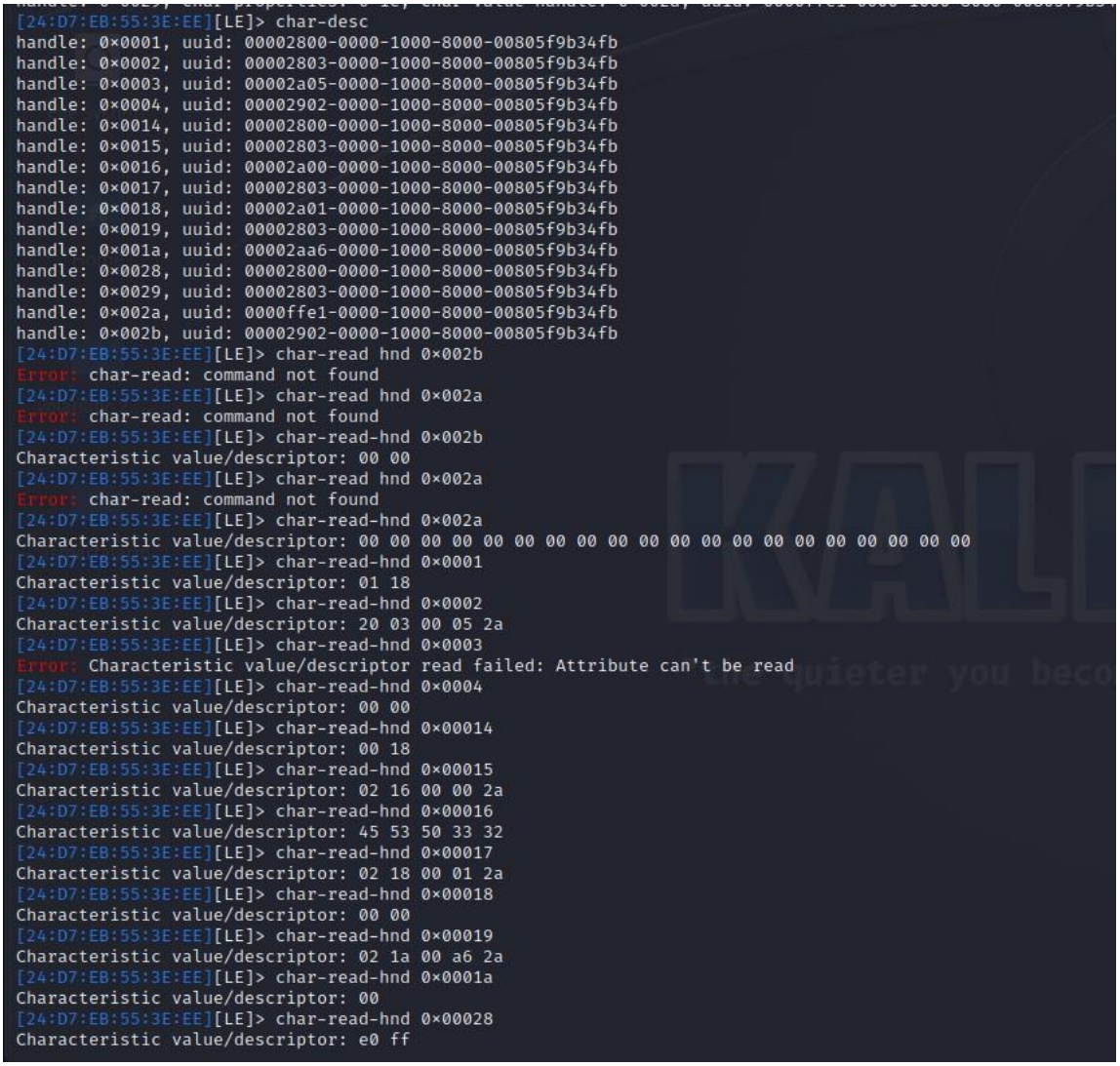
Once I knew EMO's mac address I could enumerate him, showing his handles and services running via BT. In the screenshot below I am showing the enumerated services that I can detect from EMO.



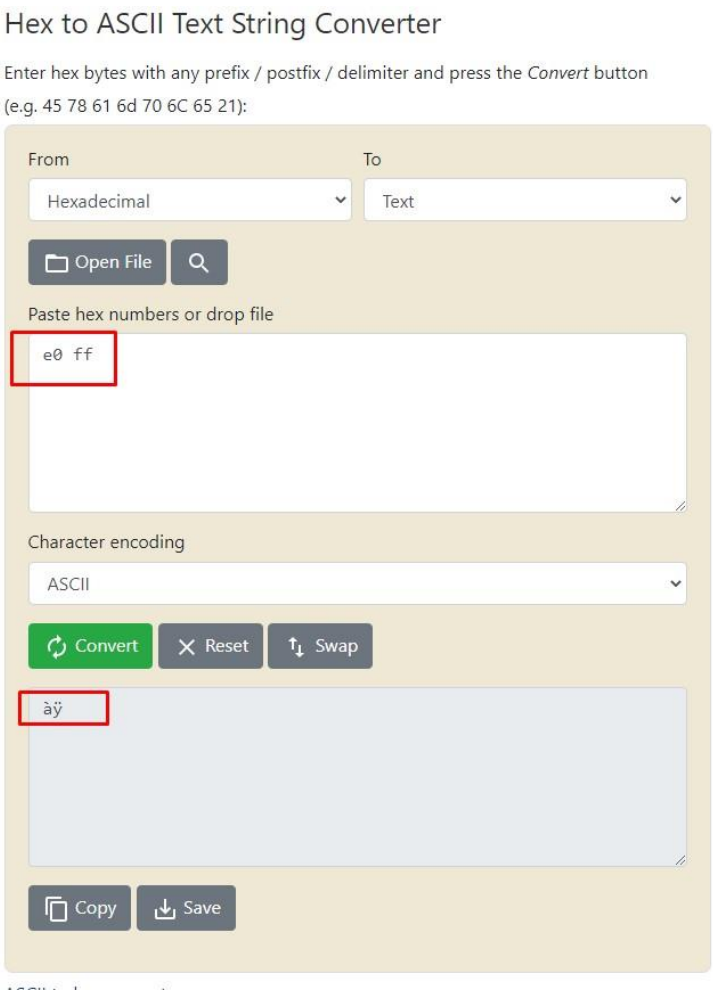
I found 5 services using EMO's BT capabilities. There were generic services like "Generic access", "Device name", "Appearance" that were Read only values. This means that I am unable to modify the values that they are returning. The last handle "002a" appeared to be running a proprietary service that had the "Write" property. I ddint know what it was doing because the service had no name but it had a description written in HEX. After I googled the Hex value (ffe0) I came to the conclusion that this was a temperature service.

After I was done with the reconnaissance I connected to EMO using gatttool. In the screenshot below I am showing the connection being made.

BT LE is very tricky communication method to go through and find useful information. Usually, the only thing you can find is a couple of bytes of information at a time. I went trough every single handle I could find, read the data, and then ran it through a Hex converter.



Most of the handles returned no readable data at all except one (0x00028). After putting the hex value (e0 ff) in the hex converter I found out that it was encrypted.



Conclusion

After researching EMO living AI Bluetooth connections I can conclude that the connection is secure. I explored all attack vectors I could detect. Almost all the BT services running on EMO were read only which makes it impossible to use as vulnerabilities for deeper exploration. I could find only one temperature service that was returning values different than 0 but the data I captured was encrypted.

Overall EMO's Bluetooth connection is very secure, and it doesn't leave backdoors or unpatched vulnerabilities, for which as a Blue teaming member I acknowledge that.

References

https://allabouttesting.org/tools-for-recon-bluetooth-devices-by-using-kali-linux/

https://stackoverflow.com/questions/27914223/corebluetooth-what-is-the-service-uuid-ffe0-shorten-for

https://stackoverflow.com/questions/66926055/ble-where-does-the-handle-come-from

https://null-byte.wonderhowto.com/how-to/bt-recon-snoop-bluetooth-devices-using-kali-linux-0165049/