Bluetooth for Decoy Systems: A Practical Study

Ken Wong
British Columbia Institute of Technology
Burnaby, BC
kkenwwong@gmail.com

Aaron Hunter
British Columbia Institute of Technology
Burnaby, BC
aaron_hunter@bcit.ca

***Abstract*—We present an approach to tracking the behaviour of an attacker on a decoy system, where the decoy communicates with the real system only through low energy bluetooth. The result is a low-cost solution that does not interrupt the live system, while limiting potential damage. The attacker has no way to detect that they are being monitored, while their actions are being logged for further investigation. The system has been physically implemented using Raspberry PI and Arduino boards to replicate practical performance.**

I. Introduction

We introduce a novel architecture to mitigate cybersecurity threats in a networked environment. We develop a solution in which a fictitious main systems interacts with a monitoring system through the use of Bluetooth Low Energy (BLE) devices. We show that using a BLE connection between a decoy system and a monitoring system actually prevents attackers from seeing the data travelling between servers. We also provide filters and applications that can be installed in practice to detect, trace and give an alert in the event of an attack. Signifcantly, our results have been tested in a real, physical networked environment.

II. Motivation

A fundamental step in preventing break-ins is to gather intelligence about adversaries and their methods [1]. One well-known technique is to deploy a honeypot, which is essentially a fake system that appears to contain real data [2]. However, it is well known that low-interaction honeypots alone cannot prevent break-ins because they do not track the actions of the attacker sufficiently [3]. Worse yet, attackers can often easily identify when a system is a honeypot, and choose not to explore it. In order to avoid this problem, we can implement a high-interaction honeypot that interacts with the real system; unfortunately this can give the attacker access to real system data. In fact, inadequate monitoring of honeypot traffic can actually increase the risk of theft of data from the real system [1].

It has been suggested that a better honeypot can be developed using wireless (Bluetooth) communication between a honeypot and a real server [4]. The idea is that the honeypot will have no hard-wired connections to the real server; as such, the attacker will not be able to access the main server through the honeypot. In effect, this idea is intended to produce a high-interaction honeypot without the inherent risks of system access. There are two problems with this idea. First, Bluetooth is expensive in terms of energy usage. Second, a Bluetooth signal can be detected from reasonably far away; this means the attacker might detect the communication. In this study, we address both of these issues through the use of Bluetooth Low Energy devices (BLEs).

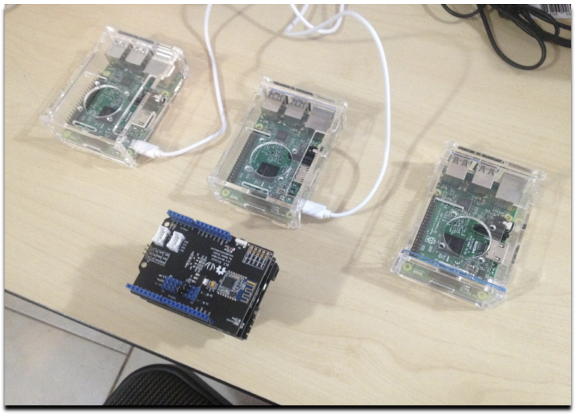


Fig. 1. Physical Setup

Our goal is to demonstrate, in a practical setting, a system architecture that uses BLEs to communicate between a decoy system and a real system in a manner that is hidden from attackers. We are already aware of the advantages of this approach in principle; it remarins to show that these advantages can actually be obtained in practice.

III. The System Design

Our experimental system involves three Raspberry PI devices acting as servers. One of these represents the attacker, and one represents the decoy system; these are on the same network. The third server is the actual main server, and it is in a different network enivironment. An Arduino electronic hardware device is is then used as an interpreter; it is the key to align the Bluetooth Low Energy devices such that they can communicate with each other on each server. The interpreter devices act like the master controller that can control each of the Raspberry PI electronic boards. The physical configuration is shown in Figure 1.

The devices needt to be relatively close; in testing we required less than 10 meters distance. Classical Bluetooth would allow greater distance, but the tradeoff would be increased power usage. We can place BLEs in locations where wireless access points would be difficult to power [5].

978-1-5386-0683-4/17/$31.00 ©2017 IEEE

The Arduino electronic board is placed in the middle, and acts as the master controller for the BLEs. The Arduino activates and deactivates the BLEs at any time they are needed. For example, if there is a need to send data between the Linux servers, this is enabled by the Arduino. The BLE devices communicate with each server using this design. The BLEs rely on the Interpreter device, because it is the master control key that activates the Bluetooth Low Energy devices, thereby permitting them to communicate with each other. When the Interpreter is disconnected, all communication between the servers stops. In this way, it secures the communication that is flowing through from the decoy system to the main system.

Several Python programs were written to detect, trace and alert the presence of an attack by communicating with the monitoring system through the Bluetooth Low Energy devices. In particular, the following main tasks are performed:

- All logins on decoy machine are logged.
- Session information is sent the monitoring machine through bluetooth.

This wireless communication between machines is undetectable at a distance. The main logging functions are actually performed by two different programs for redundancy.

IV. RESULTS

The basic design of the system is intended to limit an attacker's opportunity to compromise the syste, while simultaneiously allowing us to gather data about the attacker. One practical outcome worth noting is that we have actually produced a working demo that uses BLEs for communication. This is, of course, not a true result; but it was a technical challenge. But there are several features that must be tested and validated more precisely.

Once the system was completely configured, we were able to simulate an attack in which the attacker accesses the decoy system. This involves three working components:

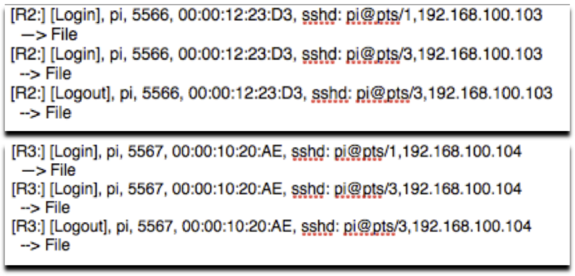
- The decoy system ran successfully, with the Python event logger in place.
- The attacker system ran successfully, and it was able to access the decoy system.
- The main system ran successfully, and it was able to receive the log file from the decoy system.

In the interest of space, we do not include screen captures of specific test results here. However, we can confirm that all of the programs ran successfully in the test enviroment. We were able to retrieve the attacker's IP address, SSHD Session, password, and Session ID. This data is useful for future investigation into a breach, and it was captured correctly on every test run.

It is also worth noting that the log files containing the attacker data were exchanged in good order from the decoy system to the monitoring system. Figure 2 shows the data that was in the log files on the main server after an example test.

V. DISCUSSION

This preliminary study set out to demonstrate that a decoy system communicating with Bluetooth would be able to capture information about an attacker, and communicate it to a monitoring system discreetly. We were able to show that this is actually the case. We successfully built a system that will detect an attack, trace the attack and give an alert if there is an attack by using Bluetooth Low Energy devices to data transfer between servers. All features were demonstrated in a practical system, built from low-cost Raspberry Pi and Arduino devices.



```
[R2:] [Login], pi, 5566, 00:00:12:23:D3, sshd: pi@pts/1,192.168.100.103
 —> File
[R2:] [Login], pi, 5566, 00:00:12:23:D3, sshd: pi@pts/3,192.168.100.103
 --> File
[R2:] [Logout], pi, 5566, 00:00:12:23:D3, sshd: pi@pts/3,192.168.100.103
 --> File

[R3:] [Login], pi, 5567, 00:00:10:20:AE, sshd: pi@pts/1,192.168.100.104
 —> File
[R3:] [Login], pi, 5567, 00:00:10:20:AE, sshd: pi@pts/3,192.168.100.104
 --> File
[R3:] [Logout], pi, 5567, 00:00:10:20:AE, sshd: pi@pts/3,192.168.100.104
 --> File
```

Fig. 2. Information Exchange

Based on the results of the present research, future work will enlist stronger filters such as adding additional authentication keys between each of the BLE devices that can strengthen security in the system. For instance, setting up an authentication key combination code that is required to be validated before entering the BLE master controller (Arduino), which is between the Raspberry PI Linux servers. Not only will this strengthen the security of the system, but it will also prevent the attacker from being able to control the Arduino electronic board, which is the master Bluetooth key controller. Once the attacker compromise the Arduino electronic board, they can have full accessibility to control the Bluetooth Low Energy devices and that will allow the attacker to control the main system too.

It is worth noting that we were not truly able to verify is the undetectability of the BLE communication between the decoy and the main server. We know that the signal is weak and it can not be directly detected at a distance. However, we can not state as an objective fact that the communication is undetectable by an intruder. First of all, an intruder might actually gain physical access to a nearby sensor. But a more difficult problem to address is the fact that an attacker may actually be able to use non-physical means to detect the communication. We leave this problem for future work.

REFERENCES

[1] A. Brown and T. Andel, "What's in your honeypot?" in *Proceedings of the 11th International Conference on Cyber Warfare and Security*, 2016, pp. 370–377.

[2] S. Padda, S. Gupta, G. Apoorva, S. Lofty, and A. Kaur, "Honeypot: A security tool in intrusion detection," *International Journal of Advanced Engineering, Management and Science*, vol. 2, no. 5, pp. 311–316, 2016.

[3] J. Rutherford and G. White, "Using as improved cybersecurity kill chain to develop an improved honey community," *IEEE Computer Society*, pp. 2624–2632, 2016.

[4] K. Fawaz, K. Kim, and K. Shin, "Protecting privacy of BLE device users," in *Proceedings of the 25th USENIX Security Symposium*, 2016, pp. 1205–1221.

[5] P. Kriz, F. Maly, and T. Kozel, "Improving indoor localization using bluetooth low energy beacons," *Mobile Information Systems*, vol. 62, pp. 1–11, 2016.