Evaluating the Impact of Malicious Spoofing Attacks on Bluetooth Low Energy Based Occupancy Detection Systems

William Oliff
Computing and Information Systems
University of Greenwich, UK
Email: william.oliff@gre.ac.uk

Avgoustinos Filippoupolitis
Computing and Information Systems
University of Greenwich, UK
Email: a.filippoupolitis@gre.ac.uk

George Loukas
Computing and Information Systems
University of Greenwich, UK
Email: g.loukas@gre.ac.uk

Abstract—**Occupancy detection of a building has a wide range of applications. Areas such as emergency management, home automation and building energy management can benefit from the knowledge of occupants' locations to provide better results and improve their efficiency. Bluetooth Low Energy (BLE) beacons installed inside a building are able to provide information on an occupant's location. Since, however, their operation is based on broadcasting advertisements, they are vulnerable to network security breaches. In this work, we evaluate the effect of two types of spoofing attacks on a BLE based occupancy detection system. The system is composed of BLE beacons installed inside the building, a mobile application installed on occupants mobile phones and a remote control server. Occupancy detection is performed by a classifier installed on the remote server. We use our real-world experimental results to evaluate the impact of these attacks on the system's operation, particularly in terms of the accuracy with which it can provide location information.**

1. Introduction

Bluetooth Low Energy (BLE) beacons, which combine low cost, low power requirements and compatibility with the majority of smart phones, have been adopted by numerous solutions geared towards smart environments. Education [1], healthcare [2], tourism [3], transportation [4] and smart building sectors are only few of the examples where this technology has been applied. Especially in a building context, the knowledge of locations is beneficial for emergency management [5], [6], [7] and building energy management [8]. The beacons, however, base their operation on the broadcast of advertisement packets which results in security vulnerabilities. This work investigates the effect that a specific type of threat (spoofing) has on the operation of a BLE based occupancy detection system. The security provisions of the Bluetooth protocol specification and Bluetooth based communication system implementations have been studied extensively [9]. There are also several examples of novel attacks [10] and applications of known ones, such as man-in-the-middle [11] proposed for its different versions, each with varying degree of practicality and effectiveness. In particular for BLE, the focus of most security research is on the development of authentication or message authorisation mechanisms, e.g. based on one-time passwords [12]. What is typically overlooked in the literature is that BLE, beyond being a communication protocol, is also an enabler of high-accuracy indoor localisation. As a result, there is little work in terms of analysis as to how different security attacks affect this aspect of the application of BLE and more specifically its ability to provide accurate indoor localisation. The security challenges in using BLE for indoor localisation have been investigated in [13]. The authors have demonstrated a packet injection attack using a malicious BLE beacon placed inside an office area. The evaluation of the impact is limited to a location heatmap illustrating a change in the distributions of occupants inside the area. In [14], the authors have implemented a BLE beacon prototype that adds a secure signature to the advertisement packet. They evaluated their solution in a tram maintenance shed to provide navigation instructions in a public transport context. Their evaluation, however, does not include an investigation of potential attacks against the proposed system. Finally, the security vulnerabilities of indoor positioning systems based on Wi-Fi and BLE technology are studied in [15], but without addressing how a network attack would affect the system's operation.

Here, we address this gap in the landscape of BLE security research by evaluating the impact of two different spoofing attacks on a BLE based indoor occupancy detection system [16], [17]. The first attack assumes a basic adversary with physical access, while the second assumes an advanced adversary who can remotely reconfigure a beacon.

2. Description of the System

In this section, we provide the details of our system design, including an overview of the architecture and a description of the individual system components.

2.1. Overall Architecture

The primary concept of our approach is based on the use of BLE beacons that have been installed inside a building. The beacons communicate with a back-end server via a mobile application running on a smartphone, allowing our

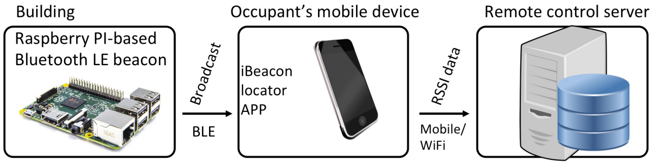


Figure 1. System Architecture

system to infer a user's location inside the building, as illustrated in Figure 1. The use of BLE beacons has been adopted in commercial deployments in order to provide micro-localisation services. The beacons operate in a non-connectible mode and only transmit BLE advertising data packets which contain information such as the beacons' identity, as shown in Figure 3. A mobile phone in the vicinity of the beacons receives the data packets and stores their received signal strength indicator (RSSI) values along with the beacon identities using a mobile application. Finally, the mobile application sends all the collected beacon data to the remote server where a trained classifier processes the data to derive the user's location. Additionally, in our approach:

- There is no assumed knowledge of the location of the beacons inside the building
- The smartphone does not process the received BLE advertising packets to calculate the user's location
- The server does not send location information back to the mobile device.

As the system does not perform any localisation processing on the mobile device, the computational and memory requirements of the mobile device are significantly reduced. It is the server's responsibility to process the raw beacon identity and RSSI information to calculate a building's occupancy. This is achieved via an initial data gathering phase which is used to train a classifier running on the server. Section 3.2 elaborates on the data gathering phase and data processing. Also, processing the raw data on the server allows for the use of more computationally expensive classification algorithms.

2.2. BLE Beacons

Our beacons are based on a Raspberry Pi 2 Model B with an attached Bluetooth 4 LE module via a USB interface. The BlueZ package is used to emulate beacons on the Raspberry Pis and customise the BLE advertising data. The beacons are purely transmitters and broadcast BLE advertising packets. Figure 2 shows one of our Raspberry Pi based prototype BLE beacons.

To identify each beacon individually, we make use of the three different IDs (UUID, Major & Minor) available within the beacon packet structure, as shown in Figure 3. Using these different identifies, we create a small hierarchy where, the Universally Unique Identifier (UUID) is used to define a universal group. Hence, all beacons used in our experiments have the same UUID, allowing them to be separated from other unassociated Bluetooth devices. Then, the major number is used to define the local groups of beacons. For example, beacons belonging to certain floors within a building will share the same major number. Lastly, the minor number is used to define each individual beacon within its local group.

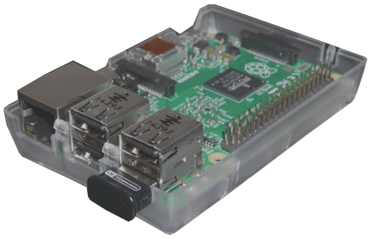


Figure 2. Our Raspberry Pi based Beacon

Prefix	UUID	Major	Major	TX
[9 Bytes]	[16 Bytes]	[2 Bytes]	[2 Bytes]	[1 Byte]

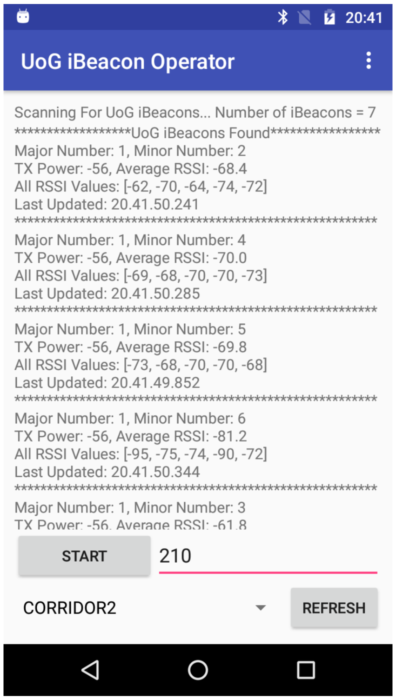
Figure 3. BLE beacon Advertising Packet Data Structure

2.3. Mobile Application

We have developed an Android mobile application using Android version 6 (Marshmallow) that has two modes of operation: data gathering mode and normal mode. In data gathering mode, the application would be used by a trained operator to collect and label the incoming data from beacons to train a machine learning classification algorithm running on the remote control server. In this mode, the application does not connect to the remote control server. In normal mode, however, the application is used by a client of the system, which is collecting the data from installed beacons and reporting to the remote control server, where a trained classification algorithm determines occupancy.

In data gathering mode, the application detects all BLE traffic in range, filtering out our beacon advertising packets and stores information related to that individual beacon, along with RSSI data for each BLE packet received. To filter out other unwanted Bluetooth traffic, the application first attempts to identify the UUID of our beacons within the received payload. If the UUID is found then, extract the other identifies with the RSSI and store the beacon, else reject the packet. In this mode, the operator simply needs to set what sector they are in and specify the amount of the time they wish to collect data for, as illustrated by Figure 4. Once data gathering has been started, the operator only needs to move around in the specified sector, as the rest of the data gathering process is automated. Additionally, the application creates a set of data logs when in data gathering mode, which can then be used to generate machine learning examples to train a classifier, as can be seen in Section 3.2.

When the mobile application is in normal operation, again the application will detect all BLE traffic it is in range of, storing our beacon packets and filtering out any unwanted Bluetooth packets. Also in this mode, once the application has collected an adequate amount of beacon advertising



```
Scanning For UoG iBeacons... Number of iBeacons = 7
*****************UoG iBeacons Found*****************
Major Number: 1, Minor Number: 2
TX Power: -56, Average RSSI: -68.4
All RSSI Values: [-62, -70, -64, -74, -72]
Last Updated: 20.41.50.241
********************************************************
Major Number: 1, Minor Number: 4
TX Power: -56, Average RSSI: -70.0
All RSSI Values: [-69, -68, -70, -70, -73]
Last Updated: 20.41.50.285
********************************************************
Major Number: 1, Minor Number: 5
TX Power: -56, Average RSSI: -69.8
All RSSI Values: [-73, -68, -70, -70, -68]
Last Updated: 20.41.49.852
********************************************************
Major Number: 1, Minor Number: 6
TX Power: -56, Average RSSI: -81.2
All RSSI Values: [-95, -75, -74, -90, -72]
Last Updated: 20.41.50.344
********************************************************
Major Number: 1, Minor Number: 3
TX Power: -56, Average RSSI: -61.8
```

Figure 4. UI of the Mobile Application in Data Gathering Mode

packets and corresponding RSSI data over a short period, such as the window size used for the classifier, it will send this raw data to the back-end server for processing. This approach of only periodically sending data to the server reduces the traffic over the wireless or mobile network interfaces and results in lower energy consumption.

2.4. Server

The role of the server is to process the beacon identities and corresponding RSSI data sent from mobile devices, and then estimate the building's occupancy. Initially the classifiers that run on the server need to be trained using the data gathered during the data gathering phase. In normal operation mode, the server uses the trained classifiers to update building occupancy estimation based on the data coming from the mobile devices.

2.5. Description of the BLE Spoofing Attacks

We have produced two BLE attack scenarios, one considering a basic and one considering an advanced attacker model. Both involve spoofing the Universally Unique Identifiers (UUIDs) of the beacons.

2.5.1. Basic Attacker Model - Evil Twin Attack. Here, the assumption is that the attacker has physical access to the area of deployment of the BLE beacons and has sniffed their UUIDs. The attacker then adds a beacon in a new location mimicking one of the existing beacons by transmitting with its UUID.

2.5.2. Advanced Attacker Model - Beacon Swap Attack. This scenario assumes that the attacker has no physical access, yet is able to remotely hijack beacons and change their configurations. This can be emulated by swapping the identities of two existing beacons, while keeping their locations unchanged.

3. Experimental Evaluation

This section elaborates on the experimental methodology we adopted and the experimental results we produced. We first give a description of the method we followed for our experiments and we continue by presenting our classification approach and related results.

3.1. Design and Methodology

We conducted our experiments inside a computer laboratory of the University of Greenwich. The laboratory is essentially an office space furnished with objects such as desks, benches and chairs. We have used a virtual grid, with each grid block representing a 1m x 1m space, to map the experimental area and illustrate the different beacon deployment patterns. The laboratory has been divided into seven sectors. Sectors one to five are computer bays while sectors six and seven represent an open corridor through the centre of the laboratory.

Three different deployment patterns were used. Firstly, the ground truth (see Figure 5) deployment pattern was used to collect the training data for the machine learning classifier. Second, the evil twin (see Figure 6) deployment pattern was used to emulate an attacker mimicking one of the existing beacons, as discussed in Section 2.5.1. In this deployment, beacon six has been duplicated. Lastly, the swap (see Figure 7) deployment pattern was used to emulate the scenario of an attacker remotely hijacking beacons and changing their configuration, as discussed in Section 2.5.2. In this deployment, the software of beacons B1 and B2 has been modified to transmit the swapped identities.

For the data gathering phase, we used the mobile application while randomly walking within the boundaries of each sector for all three different deployment patterns. When gathering the data, we used the mobile application to label the incoming BLE advertising packets with the current sector. Moreover, for each received BLE packet, we ensured that the source of the packet was one of our BLE beacons by checking the UUID. Then, we logged the respective identifiers and RSSI with the assigned sector label. The beacons' transmission frequency was set at 7 Hz throughout the experiment. Additionally, we used a timer within the mobile application to only log incoming BLE advertising packets for 210 seconds for each sector.

3.2. Classification Based Occupancy Detection

Using our data set from the ground truth deployment (see Figure 5) to train a Logistic Regression (LR) classifier. We have chosen to use a LR classifier as it a fairly

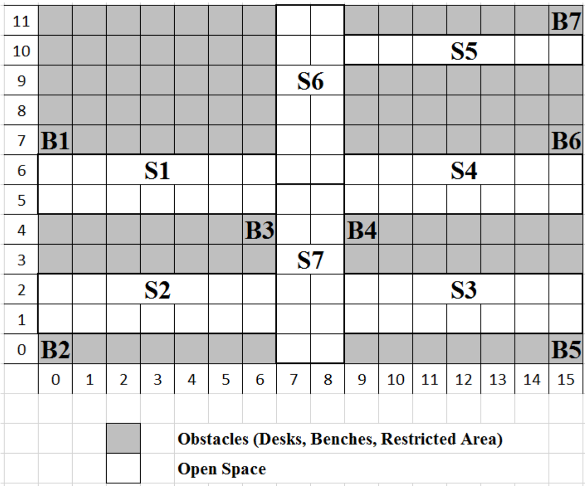


Figure 5. Beacon Deployment Ground Truth

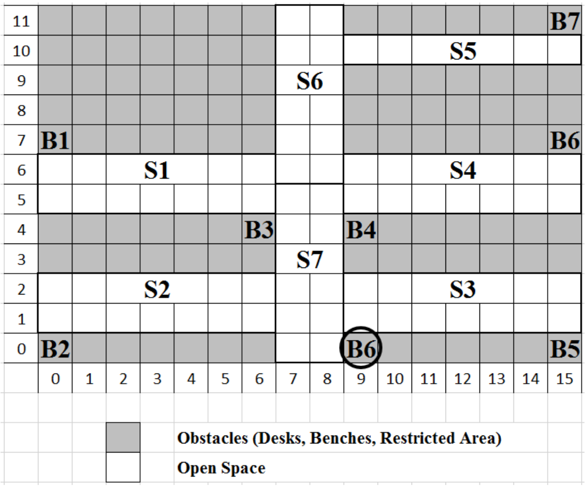


Figure 6. Beacon Deployment Evil Twin

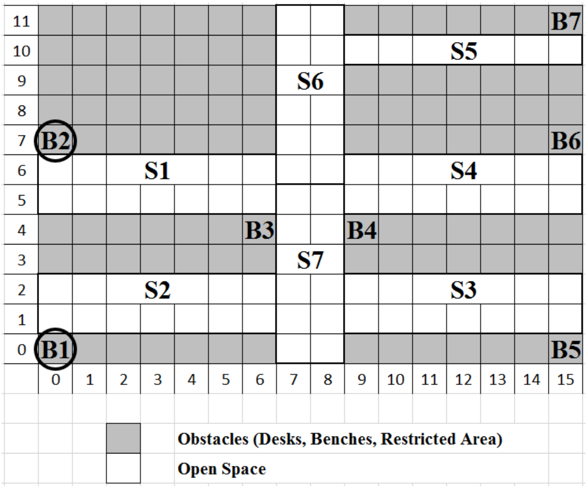


Figure 7. Beacon Deployment Swap

computationally inexpensive classification algorithm with a reasonable level of accuracy. As previously mentioned in Section 3.1, our raw data set contains the individual BLE advertising packets from specific beacons, with beacon identifiers, respective RSSI value and a sector label. To create the data points for training the LR classifier, we used a windowing approach and for each beacon we calculated the mean and standard deviation of its RSSI values within the window. This resulted in 14 features for each data point. We used three different window sizes (1s, 2s and 3s) with no overlap. As the size of the windows increases, the more raw BLE packets are used within the window and hence, lower amount of data points produced.

When training the LR classifier, the data sets were randomised and then partitioned into a 80% training set and a 20% testing set. 10-fold cross-validation was used to select a suitable λ regularisation parameter for the LR classifier for the three different window sizes. After performing 10-fold cross-validation, 0.3 was found to be the most suitable for the λ regularisation parameter for all window sizes. Moreover, as we use different window sizes, this results in our data sets sizes varying. More specifically, the training data set sizes was 1171, 583 and 387 data points for window sizes of 1s, 2s and 3s respectively. When testing the LR classifier, the unseen 20% test set was used to measure the performance of the LR classifier under normal system operation. This was the basis for our ground truth results, as shown in Section 3.2.1. The trained classifier was then tested with the evil twin and swap data sets.This informed our evil twin and swap results, as discussed in Sections 3.2.2 and 3.2.3 respectively. The test data set sizes were 1463, 728 and 483 data points for window sizes of 1s, 2s and 3s respectively for both evil twin and swap scenarios.

For evaluating the performance of the LR classifier, the metric used was its accuracy, which is defined as

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions}$$

However, accuracy generally does not optimally define the performance of the LR classifier, since it is negatively affected by class imbalance. Although our data gathering methodology inherently minimises class imbalance, as we spent an equal amount of time in each of the sectors, we decided to use a confusion matrix to better describe the performance for the normal operation and attack scenarios.

In our confusion matrix, each row represents the instances in an actual class, while each column represents the instance in an predicted class. The elements along the diagonal represent the number of instances when the predicted sector label is the same as the actual sector label. Whereas, the off-diagonal elements represent the number of instances that are misclassified. The confusion matrices have been normalised by the number of elements in each class. Additionally, we have colour coded the elements of the matrices by assigning white to 0.0 (0%) and black to 1.0 (100%), to improve the visualisation of the data.

3.2.1. Ground Truth Results. Tables 1, 2 and 3 show the experimental results belonging to the ground truth beacon deployment (see Figure 5), for the different window size values. These results show how our LR classification algorithm is performing when the system is in normal operation. By inspecting the confusion matrices we observe that the accuracy of the classification does generally improve as the window size increases, with a widow size of 3s performing the best overall. As discussed in Section 3.2, as the size of window increases in time, the more RSSI values are averaged in the window. However, increasing the window size results in less data points being available to train the classifier which can affect the performance of the system. We can observe that the two corridor sectors (S6 & S7) are being misclassified over the different window sizes, especially in instance of S7 with values ranging from 44% to 76%. This can be explained by looking at the floor plan in Figure 5. As we can see, sectors S1 to S5 are each primarily associated one beacon that isn't associated with any other sector (B1 with S1, B2 with S2, B5 with S3, B6 with S4 and B7 with S5). Whereas, sectors S6 and S7 are between multiple beacons and not have a single beacon primarily associated with them. Moreover, Logistic Regression is a linear classifier and therefore, when the data isn't linearly separable, the predictions are less accurate.

	S1	S2	S3	S4	S5	S6	S7
S1	0.78	0.02	0.02	0	0	0.05	0.13
S2	0.02	0.75	0.02	0	0.02	0.06	0.13
S3	0	0	0.86	0.02	0	0.06	0.06
S4	0.02	0	0.02	0.83	0.02	0.05	0.06
S5	0	0	0	0	0.95	0.05	0
S6	0.02	0	0.02	0	0.23	0.68	0.05
S7	0.13	0.08	0.17	0.08	0	0.1	0.44

TABLE 1. Exp = Ground Truth, Window = 1s

	S1	S2	S3	S4	S5	S6	S7
S1	0.9	0	0	0	0	0.05	0.05
S2	0	0.95	0	0	0	0	0.05
S3	0	0	0.76	0	0	0.04	0.2
S4	0	0	0	0.95	0	0	0.05
S5	0	0	0	0	0.82	0.18	0
S6	0.05	0	0	0	0	0.77	0.18
S7	0.23	0.09	0.09	0.04	0	0	0.55

TABLE 2. Exp = Ground Truth, Window = 2s

3.2.2. Evil Twin Attack Results. Tables 4, 5 and 6 illustrate the experimental results belonging to the evil twin beacon deployment (see Figure 6), for the different window size values. We can clearly note that there is an overall deterioration in the classification results. More specifically, the sectors that are mostly affected are the ones near the location of the evil twin beacon. For example, for the case of a window of 3s the classification accuracy for Sector 2 (which is close to the evil twin) drops from 100% to 75%. This is not, however, the case for Sector 5 (which is the area furthest away from the location of the evil twin) as the reduction to its classification accuracy is far less (drops from 93% to 86%). A potential application of this attack scenario could be the disruption of an indoor localisation system by an opponent who has sniffed the identities of the beacons and has physical access to the deployment area. As the experimental results indicate, the impact of the disruption can be spatially contained, depending on where the location of the evil twin beacon.

	S1	S2	S3	S4	S5	S6	S7
S1	0.93	0	0	0	0	0	0.07
S2	0	1	0	0	0	0	0
S3	0	0	0.74	0	0	0	0.26
S4	0	0	0	1	0	0	0
S5	0	0	0	0	0.93	0.07	0
S6	0	0	0	0	0	0.92	0.08
S7	0	0	0.08	0.08	0	0.08	0.76

TABLE 3. Exp = Ground Truth, Window = 3s

	S1	S2	S3	S4	S5	S6	S7
S1	0.8	0.02	0.01	0	0	0.02	0.15
S2	0.07	0.69	0.01	0.08	0.01	0.01	0.13
S3	0	0	0.55	0.43	0	0	0.02
S4	0.02	0	0.04	0.82	0.03	0.03	0.06
S5	0.02	0.01	0	0.01	0.83	0.11	0.02
S6	0.12	0.01	0.02	0.07	0.23	0.34	0.21
S7	0.09	0.04	0.03	0.43	0	0.04	0.37

TABLE 4. Exp = Evil Twin, Window = 1s

	S1	S2	S3	S4	S5	S6	S7
S1	0.88	0	0	0	0	0.02	0.1
S2	0.04	0.76	0	0.01	0.01	0	0.18
S3	0	0	0.62	0.32	0	0	0.06
S4	0.03	0	0.01	0.84	0.03	0.02	0.07
S5	0.02	0	0	0	0.89	0.09	0
S6	0.11	0	0.03	0.07	0.16	0.48	0.15
S7	0.1	0.02	0	0.3	0	0.01	0.57

TABLE 5. Exp = Evil Twin, Window = 2s

	S1	S2	S3	S4	S5	S6	S7
S1	0.9	0	0	0	0	0.01	0.09
S2	0.04	0.75	0	0.03	0	0	0.18
S3	0	0	0.59	0.38	0	0	0.03
S4	0.02	0	0	0.9	0	0.04	0.04
S5	0	0	0	0	0.86	0.14	0
S6	0.09	0	0.01	0.03	0.1	0.58	0.19
S7	0.15	0.03	0	0.3	0	0.01	0.51

TABLE 6. Exp = Evil Twin, Window = 3s

3.2.3. Beacon Swap Attack Results. Tables 7, 8 and 9 show the experimental results belonging to the swap beacon deployment (see Figure 7), for the different window size values. Similarly to the evil twin case, we can observe a decrease in the overall classification accuracy. We must note, however, an interesting effect on the sectors for which the beacons have been swapped. For all window values, the classifier consistently misclassifies sector 1 for sector 2 and vice versa. In particular, the normalised values of missclassified data points exceeds 60% and in some cases approaches 80% (e.g. sector 2 in Tables 8 and 9. If we interpret this result in terms of the classifier's performance, it is evident that the deterioration is far worse than that of the evil twin case. However, in this case we observe a targeted manipulation of the classification results: occupants that are located in sector 1 will be confidently reported by the system as being in sector 2 and vice versa. This attack approach could be adopted in order to manipulate an indoor occupancy system and alter the information provided to the users. For example, in the case of a terrorist attack the opponent can implement a beacon swap to mislead the emergency responders and disrupt the rescue operation. Similarly to the evil twin scenario, the opponent has to sniff the identities of the beacons. However, he also has to gain access to the beacons' software to alter the information they transmit.

	S1	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**
S1	0.17	0.58	0	0.01	0.01	0.07	0.16
S2	0.73	0.12	0.01	0.01	0	0.11	0.02
S3	0.02	0	0.68	0.14	0.01	0.07	0.08
S4	0	0.01	0.01	0.85	0.01	0.06	0.06
S5	0.02	0	0	0.01	0.84	0.13	0
S6	0.09	0.01	0.01	0.05	0.22	0.48	0.14
S7	0.16	0.02	0.09	0.07	0.01	0.23	0.42

TABLE 7. Exp = Swap, Window = 1s

	S1	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**
S1	0.12	0.66	0	0	0.01	0.09	0.12
S2	0.77	0.08	0.01	0	0	0.13	0.01
S3	0.02	0	0.8	0.05	0	0.05	0.08
S4	0	0	0	0.86	0.03	0.06	0.05
S5	0.02	0	0	0	0.8	0.18	0
S6	0.11	0	0.01	0.03	0.14	0.61	0.1
S7	0.1	0.01	0.08	0.06	0.02	0.25	0.48

TABLE 8. Exp = Swap, Window = 2s

	S1	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**
S1	0.1	0.67	0	0	0	0.06	0.17
S2	0.78	0.1	0	0	0	0.12	0
S3	0	0	0.8	0.06	0	0.04	0.1
S4	0	0	0	0.93	0.01	0.03	0.03
S5	0.03	0	0	0	0.8	0.17	0
S6	0.1	0	0	0.03	0.12	0.64	0.11
S7	0.15	0	0.03	0.04	0	0.23	0.55

TABLE 9. Exp = Swap, Window = 3s

4. Conclusions and Future Work

We have evaluated the effect of two attack scenarios against an occupancy detection system based on BLE beacons. The attack scenarios address a basic opponent with physical access to the area and an advanced opponent who is able to remotely access the software of the deployed beacons and modify the information they transmit. Our real world experiments indicate that although both attacks result in a deterioration of our system's performance in terms of classification accuracy, the beacon swap scenario can alter the system's operation in a more sophisticated manner to manipulate the information provided to the users.

Looking at areas of future work, we will investigate approaches that can help us detect these types of attacks as well as well as response mechanisms that will improve the performance of our system. We will also extend the range of machine learning algorithms that we use, to investigate the effect of malicious spoofing attacks on different classification approaches.

References

[1] M.-Y. Bae and D.-J. Cho, "Design and implementation of automatic attendance check system using ble beacon," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 10, no. 10, pp. 177–186, 2015.

[2] K.-Y. Lam, N. W.-H. Tsang, S. Han, W. Zhang, J. K.-Y. Ng, and A. Nath, "Activity tracking and monitoring of patients with alzheimers disease," *Multimedia Tools and Applications*, vol. 76, no. 1, pp. 489–521, 2017.

[3] Z. He, B. Cui, W. Zhou, and S. Yokoi, "A proposal of interaction system between visitor and collection in museum hall by ibeacon," in *2015 10th International Conference on Computer Science Education (ICCSE)*, July 2015, pp. 427–430.

[4] P. K. Volam, A. R. Kamath, and S. S. Bagi, "A system and method for transmission of traffic sign board information to vehicles and relevance determination," in *Advances in Electronics, Computers and Communications (ICAECC), 2014 International Conference on*. IEEE, 2014, pp. 1–6.

[5] N. Dimakis, A. Filippoupolitis, and E. Gelenbe, "Distributed building evacuation simulator for smart emergency management," *The Computer Journal*, vol. 53, no. 9, pp. 1384–1400, 2010.

[6] S. Timotheou and G. Loukas, "Autonomous networked robots for the establishment of wireless communication in uncertain emergency response scenarios," in *Proceedings of the 2009 ACM symposium on Applied Computing*. ACM, 2009, pp. 1171–1175.

[7] A. Filippoupolitis, G. Gorbil, and E. Gelenbe, "Spatial computers for emergency support," *The Computer Journal*, vol. 56, no. 12, pp. 1399–1416, 2012.

[8] M. Choi, W.-K. Park, and I. Lee, "Smart office energy management system using bluetooth low energy based beacons and a mobile app," in *Consumer Electronics (ICCE), 2015 IEEE International Conference on*. IEEE, 2015, pp. 501–502.

[9] C. T. Hager and S. F. MidKiff, "An analysis of bluetooth security vulnerabilities," in *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, vol. 3. IEEE, 2003, pp. 1825–1831.

[10] K. Haataja, K. Hyppönen, S. Pasanen, and P. Toivanen, *Bluetooth Security Attacks: Comparative Analysis, Attacks, and Countermeasures*. Springer Science & Business Media, 2013.

[11] M. A. Albahar, K. Haataja, and P. Toivanen, “Bluetooth mitm vulnerabilities: A literature review, novel attack scenarios, novel countermeasures, and lessons learned.” *International Journal on Information Technologies & Security*, vol. 8, no. 4, 2016.

[12] H. Jung, D. Shin, K. Cho, and C. Nam, “Ble-otp authorization mechanism for ibeacon network security,” *Journal of KIISE*, vol. 42, no. 8, pp. 979–989, 2015.

[13] P. Misra, S. Raza, V. Rajaraman, J. Warrior, and T. Voigt, “Security challenges in indoor location sensing using bluetooth le broadcast,” *ewsn 2015*, p. 11, 2015.

[14] T. Schulz, F. Golatowski, and D. Timmermann, “Secure privacy preserving information beacons for public transportation systems,” in *Pervasive Computing and Communication Workshops (PerCom Workshops), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1–6.

[15] S. Kim, S. Ha, A. Saad, and J. Kim, “Indoor positioning system techniques and security,” in *e-Technologies and Networks for Development (ICeND), 2015 Forth International Conference on*. IEEE, 2015, pp. 1–4.

[16] A. Filippoupolitis, W. Oliff, and G. Loukas, *Occupancy Detection for Building Emergency Management Using BLE Beacons*. Cham: Springer International Publishing, 2016, pp. 233–240. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-47217-1_25

[17] ——, “Bluetooth low energy based occupancy detection for emergency management,” in *2016 15th International Conference on Ubiquitous Computing and Communications and 2016 International Symposium on Cyberspace and Security (IUCC-CSS)*, Dec 2016, pp. 31–38.