

# Denial of Sleep Attacks in Bluetooth Low Energy Wireless Sensor Networks

Jason Uher  
Johns Hopkins University  
Applied Physics Laboratory  
Laurel, MD  
Email: jason.uher@jhuapl.edu

Ryan G Mennecke III  
Johns Hopkins University  
Applied Physics Laboratory  
Laurel, MD  
Email: ryan.mennecke@jhuapl.edu

Bassam S Farroha\*  
Department of Defense  
Ft Meade, MD  
Email: farroha@ieee.org

**Abstract**—Many of the benefits of an Internet of Things sensor network model stem from the extremely long service life of its base sensing layer. When data from the base sensing layer is provided by very low power technologies, such as Bluetooth Low Energy, a class of vulnerabilities called Denial of Sleep attacks can be especially devastating to the network. These attacks can reduce the lifespan of the sensing nodes by several orders of magnitude, rendering the network largely unusable. This paper investigates a Denial of Sleep attack against the Bluetooth Low Energy protocol that allows a malicious actor to rapidly drain the battery of a targeted sensing node, including power analysis, simulation results, and an example implementation. The outcome will be utilized to build better defenses and more predictable environments.

## I. INTRODUCTION

While the Denial of Sleep (DoSL) concept in wireless sensor networks is a relatively new concept as a discrete goal, several of the methods and implementations rise from past concepts in wireless sensor network security and Denial of Service (DoS) attacks. In the past, the distinction between DoS and DoSL has largely been moot - popular sensor network protocols such as SMAC [1] and TEEN [2] are designed to allow for large nets of low power nodes to efficiently exchange information but still assume nodes with batteries and processing capabilities orders of magnitude higher than those found in a typical modern PAN [3]. The last several years have revealed a trend toward a much broader concept of Wireless Sensor Networks (WSN), the Internet of Things (IoT). One of the primary aspects of an IoT model is the idea of sensing through a large number of distributed element each providing small data points, such as temperature, humidity, vibration, etc. In the IoT model these nodes are not participating in a full network as they would be in a traditional WSN model [4]. These nodes simply report their collected data at set intervals and often do not even participate in the network in traditional terms: a more capable intermediary node collects their reports and aggregates the data for use. With their relatively limited capability, these types of nodes are designed with a life of years, or even decades, simply reporting a sensed value at a set interval. One potential implementation for these types of sensors is the Bluetooth Low Energy (BLE) specification within the Bluetooth 4.0 release [5]. BLE provides a mechanism for small sensing devices to report their sensed data at set intervals and spend most of their time in a low power sleep mode, with a few commands allowed for configuring the node and requesting additional information if necessary.

Bluetooth Low Energy sensing nodes will find a home in the IoT ecosystem focused on reporting back values that will eventually be integrated into larger decision making processes [6]. These processes will likely drive several high impact applications including security monitoring, safety of life networks such as self-driving cars, and critical infrastructure protection. With this type of IoT architecture, one particularly devastating attack is the Denial of Sleep attack. The nodes, designed to have a very long service life, will often be deployed in hard to reach places, making replacements of the batteries and nodes very difficult. A well timed attack could render a sensing network inoperable for days, or even weeks, as maintenance technicians replace the nodes or batteries across the network. In a highly sensitive low latency WSN, this attack can be detrimental. One simple example scenario that can have a dramatic effect on a national level involves sensing networks designed to detect prohibited materials at import/export locations. If smugglers were able to bring that network down for only a few hours by draining the battery of sensing nodes, shipments of contraband material could pass through the port undetected. Utilizing a slightly different method, attackers could analyze the network as a whole for weak spots, and drain only the nodes along their path of entry, passing undetected through a network that otherwise appears to be functioning normally. For a household, secure building, or automotive related sensors an attack can allow intruders to avoid detection when entering and leaving the premises or causing traffic jams or accidents.

In this paper, we will demonstrate a targeted power drain attack against BLE devices. Section II will present previous work, including analysis of generic power drain attacks against sensing networks and specific BLE attacks. In Section III, an overview Bluetooth Low Energy protocol will be given. Section IV provides an analysis of the attack, including worst and base case scenarios for an attacker. Section V outlines our attack in the abstract, and provides insight into the aspects of the BLE protocol that make the attack possible. These results are verified in Section VI, where the attack is demonstrated against commercial BLE devices, demonstrating the effectiveness in the real world. Potential strategies for preventing this attack are considered future work, and are presented in Section VIII. Finally, Section IX presents a summary of this work and provides a look at future work that can be done to help mitigate this type of attack in the future.

## II. OTHER DENIAL OF SLEEP WSN ATTACKS

Denial of Sleep attacks are classified as a subset of Denial of Service attacks that allow for nuanced effects within sensor networks, but have only recently become an area of interest in the research community. Targeted attacks in WSNs allow malicious users to alter routing, create blind spots in the sensing network; but there are many methods for eliminating that node. The relatively large number of options for eliminating a node, coupled with the assumption of large batteries, means that DoSL attacks have largely been ignored in the literature. This is beginning to change, however, as WSN nodes move towards more aggressive power budgets with the intent of increasing network life to decades or more [7]. In this section we present a summary of the previous work specifically focused on DoSL attacks, including the methodology and achievable effects.

Brownfield provides a good introduction to the DoSL attack and an overview of potential weaknesses in the popular media access control (MAC) protocols of the era [8]. This work is built on [9], providing a strong basis to build DoSL attacks by identifying the potential sources of energy consumption in the various functions that a MAC protocol must perform (channel assessment, collision detection, etc). Both [10] and [11] demonstrate an effective method of denying sleep to a full network when the S-MAC [1] protocol is utilized, but these schemes focus on large, interconnected sensor networks and make it difficult to selectively target individual nodes. In addition, the attacks are resource intensive in that they require the attacker to be constantly monitoring the channel and rely on strict packet timing to achieve results. In contrast, attacks may be extremely simple to implement, such as in [12]. In this scheme, the node is kept alive by repeatedly sending a request to send (RTS) message, which keeps all the nodes within range of the attacker listening for new messages. While this method can be effective, it is also extremely simple to detect and mitigate, as the authors show in [13]. In addition to being easily defeated, this attack shares the same drawback as the majority of DoSL attacks in the literature: all nodes are equally affected.

Overall, the bulk of DoSL attacks in the literature target the traditional sensor network model based on the assumption of central controllers and routers with large power sources. With new models for extremely low power, high latency IoT devices added to the infrastructure [14], new models for DoSL attacks will have to be considered. Within these new models, the potential for DoSL attacks will be significantly higher, and demonstrated in Section V.

## III. BRIEF PROTOCOL OVERVIEW

The Bluetooth Low Energy protocol uses master/slave roles to control how the BLE radio connection is managed. In this setup any device can be either the master or the slave depending on initiator contact and vendor implementation. The master device can connect up to seven slave devices to form what is referred to as BLE piconet. Slaved devices can connect to one master device at a time but they are allowed to switch between master devices to form ad-hoc interconnected piconets which

are referred to as BLE scatternets. A master device can poll any of the slave sensor devices for sensor data but the slave can only communicate to the master when initiated by the master device. The protocol is built for ultra-low power consumption and an extended range when compared to traditional Bluetooth by utilizing 40 of the 80 channels at the 2.4GHz ISM band using a GFSK modulation with 0.5 index. The 2 MHz guard intervals and the allocation of 3 advertising channels when compared to traditional Bluetooth offer faster connection times and lower power consumption.

BLE devices access data through the use of profiles, services, and characteristics that are derived from the General Attribute Profile (GATT) in an object oriented and server based structure. Profiles are definitions of possible applications and they can specify general behavior. The GATT profile utilizes a client/server which specifies storage and data flow. In order to share application specific data BLE devices must conform to the same profiles that can be generic and proprietary. A service is a collection of data and behavior that represents a specific function or aspect. Services are defined and accessed by either a 16 or 128 bit unique UUID. The collection of data in a service is represented by characteristics. Characteristics contain a single labelled, defined, and discrete value that represents a specific attribute of the sensor, e.g. temperature, step count, or battery life. Although the characteristic is a single value it can be made up of several sensor data points. For example, data from the accelerometer x, y, and z dimensions can form a single velocity value. Characteristics have a behavior component that states how the information can be accessed, i.e. read, read write-no-response, read write-response, and notify. The standard defines several profiles, but also allows vendors to define proprietary functions. Each characteristic is defined and accessed by either a 16 or 128 bit unique UUID.

## IV. POWER DRAIN ANALYSIS

Based on the analysis from [9], denial of sleep attacks can be modeled effectively by utilizing a basic average of the power draw during different operating states of the nodes.

$$T_{life} = \frac{C_{bat}}{\sum_{n=1}^N T_n * P_n} \quad (1)$$

In this model, the total sensor life,  $T_{life}$ , is the sum of the charge drawn by each of the  $N$  operating modes of the sensor. This draw is calculated by multiplying the power draw of each state,  $P_n$  by the amount of time that state is active,  $T_n$ . In the simplest case, and that analyzed in [9], is the case where there are only two modes: processing and sleeping. In this case, the characterization of  $T_{life}$  can be reduced to a single dependent variable based on the amount of time the sensor spends sleeping ( $T_s$ ) and active ( $T_a$ ).

$$R_s = \frac{T_s}{T_s + T_a} \quad (2)$$

This reduces our model for the total sensor life to

$$T_{life} = \frac{C_{bat}}{(R_s)(P_s) + (1 - R_s)(P_a)} \quad (3)$$

While this model would be considered low fidelity for most WSNs, it fits perfectly for a protocol such as BLE for two primary reasons. First, the design of the BLE protocol closely matches this model in that there are really only two modes of operation: actively sensing and reporting or asleep. Second, due to the relative efficiency of modern ASICs, transmit and receive functionalities in BLE consume similar amounts of current [15]. When nodes are receiving commands, processing data, or actively transmitting they consume approximately the same amount of current making the concept of a single  $P_{active}$  very realistic.

If we assume that a sensor is reading in sufficient intervals then the ratio of sleep and active time will remain constant, we can rearrange equation 3 and take the derivative to find the consumption over time.

$$\frac{d}{dt}C_{bat} = \frac{d}{dt}T_{life} \cdot (R_sP_s + (1 - R_s)P_a) \quad (4)$$

$$= R_sP_s + (1 - R_s)P_a \quad (5)$$

With this discharge rate, we can find the estimated battery capacity at time  $t$  as

$$C_{bat}(t) = C_{bat=0} - R_sP_s + (1 - R_s)P_a \quad (6)$$

Finally, because BLE requests allow for simultaneous read and write requests on different hopsets, equation 6 can be modified to allow for  $N_c$  simultaneous commands. In this case, the amount of power drawn while actively decoding commands is increased linearly with  $N_c$ , as shown in

$$C_{bat}(t) = C_{bat=0} - R_sP_s + (1 - R_s)(P_aN_c) \quad (7)$$

This final equation will provide us with an estimate of the battery at any time  $t$  within the discharge cycle of the node. Section VII demonstrates the efficacy of this predictor with respect to measured data during the attack.

## V. OUR EXPERIMENTAL ATTACK

Our implementation of an experimental basic Denial of Sleep exploits the ubiquitous connection methods utilized by BLE devices. These methods allow BLE devices to connect to other, unknown devices without proper verification even when proper BLE authentication is otherwise utilized.

Because sensor node BLE devices operate in a ZeroInteraction Authentication (ZIA) and are strictly M2M connections, it is very difficult to determine if the interacting device is friend or foe. This ZIA model vulnerability is exacerbated by the ad-hoc meshes, star piconets, and scatternet networks implemented by BLE WSNs. While these network models allow for robustness, flexibility and power savings, while also simultaneously providing fast connection times with multiple master devices they lack sufficient authentication

procedures to prevent malicious data requests. During the creation and operation of massive BLE WSNs, BLE sensor devices could potentially have to connect to hundreds of different nodes in order to extend the sensor network, especially in a mobility application. The lack of proper device identification can lead to unwanted connections that drain the power resources of nodes by making multiple fast connections which can drain resources not considered in the original network's power budget. This type of repeated connection attack can be multiplied by several orders of magnitude by utilizing resources on the BLE target sensor node. The resources available on BLE sensor nodes can be transmission and reception of BLE sensor information by accessing the BLE services and characteristics that are available. These attributes of the sensor node can have read, write and notify behaviors that all have valuable resource utilization processing. By manipulating these attributes of the BLE sensor node a maximum power drain can be established and used against all sensor node devices. These services and characteristics can be characterized by monitoring or sniffing the WSN interconnections. This type of passive collection will not only allow the characterization of vendor specific proprietary BLE profiles but will also provide multiple BLE sensor MAC addresses and connection intervals. Once the system has been characterized a simple attack can be programmed and orchestrated using open source BLE stack code that is widely available for Linux and which can be distributed on portable cheap disposable hardware platforms that require low power use.

## VI. ATTACK IMPLEMENTATION

The Fitbit Charge HR [16] was chosen as the target BLE sensor node for this test because of its new release and 120 hour extended operation. This device was a good candidate because it collects sensor data and periodically communicates the eHealth metrics to the master device via BLE. The system was modeled to collect data for an average person and periodically pass that data when polled at deterministic intervals from the master device so as to have the battery last 120 hours before recharge. The first step was to characterize the Fitbit Charge HR system in a normal working environment. This was completed using an Ellisys Bluetooth Explorer which passively sniffs all traffic for Bluetooth and BLE. The characterization of the services and characteristics are shown below. Fitbit Charge HR Service, Characteristic Profile:

Battery Service

- Battery Level
- Client Config Descriptor

Device Information

- Manufacturer Name String
- Descriptor - Characteristic Format
- Characteristic Unknown

Vendor specific

- Vendor specific char0019

- - Descriptor
- Vendor specific
- Characteristic char000d
- - Descriptor
- Characteristic char0010
- Characteristic char0012
- - Descriptor
- Characteristic char0015
- - Descriptor

From the characterization profile, the sensor's data dumping algorithm was recorded along with other information. Other commands were also observed, where one would send out notifications and another exposed writeable characteristic fields that allow up to 20 bytes to be written at a time. Finally, a readable Battery Level characteristic was discovered that makes profiling power consumption significantly easier. Utilizing these commands, a DoSL attack was created by modifying the BlueZ open source Bluetooth stack that is available on Linux operating systems. The implementation was run on a commodity dual core laptop, but has also been tested on a raspberry Pi running from a battery source. Code modification for BlueZ supported the automation of the test to retrieve deterministic battery read intervals during testing.

The sensor data collection process is characterized by the following steps:

- Turn notify on to the characteristic char0015
- Establish Fitbit Airlink active connection
- Write sensor dump command
- Read the battery level by setting notify on on characteristic char0023.

Different tests were created in a python script that started an instance of the bluetoothctl command line application and piped data to the application using Linux named pipes and the modified BlueZ code. An example of the initialization of the bluetoothctl application with multiple 10 byte writes for 60 seconds is in the following code snippet.

```
#!/ env python os.system(mkfifo fifo) os.system(cat
> fifo &) os.system(blueoothctl < fifo)
os.system(echo power on > fifo) os.system(echo
connect <MAC > fifo) os.system(echo select-
attribute char0010)
while(1):
    os.system(echo write 0xFF 0xFF 0xFF \
                0xFF 0xFF 0xFF \
                0xFF 0xFF 0xFF \
                0xFF)
    if(time > 60): break
```

An attack utilizing writes could consistently write 1 to 20 random bytes to sensor node characteristic char0010 which would in theory drain the power at a constant rate. In order to calculate the power consumption in relation to time, the throughput of the sensor node had to be tested. Writing results

from the Ellisys showed the throughput to be around 1KB/s. The maximum packet size for BLE is 23 bytes therefore 44 packets would approximately represent one second.

In order to estimate the sensor life using the model from Section IV, the power consumption for each of the device states is required. Based on a combination of device disassembly, the processor used [16] and BLE chipset datasheet [15], the parameters for the model can be estimated. With a system voltage of 3.3V, the active and sleeping states draw approximately 12.5mA and 2μA respectively, against a 180mAh battery.

Utilizing this system, a series of 5 tests were run varying the read/write characteristics and time spent sleeping or writing. Section VII provides a summary of the results for each of these tests.

## VII. RESULTS

In order to test the battery draw effects we ran 5 tests varying both the sleep ratio ( $R_s$ ) and the number of simultaneous accesses ( $N_c$ ).

Figure 1 shows both the expected theoretical lifetimes and the experimental battery readings for each of the 5 different test cases. As expected, this DoSL shows a significant reduction in battery life using only a single attack sensor. With the introduction of additional sensors and the corresponding increases in  $N_c$  it is likely that batteries may be drained in a matter of minutes, rendering the entire network unusable until it can be serviced.

The results show significant increases in power consumption by the varying DoSL attacks implemented. The 120 hour sensor node uptime was dramatically decreased to a maximum power drain time of approximately 6 hours. Once the characterization of the system in normal operation was completed, the sensor data dumping procedure outlined in Section V was profiled and recorded. The procedure was used to dump the sensor data at different intervals to keep the device out of the sleep state for varying values of  $R_s$ .

*Test Case 1* consisted of periodically connecting to the device every 300 seconds, holding it in a read state for 60 seconds, and dumping all available sensor data. While the amount of data collected during the 300 second interval was not deterministic there was, on average, 100 seconds of data transfer after the 60 second read state. Based on this test we can find  $R_s$  as approximately  $140/300 = 0.46$ . Because we are only reading from the device, the total number of connections  $N_c$  is only 1. Under these conditions, it took approximately 29 hours to successfully drain the target device, for a total reduction of about 75%.

*Test Case 2* reduces the read interval down to 100 seconds, but also eliminates the final data dump. This provides us with

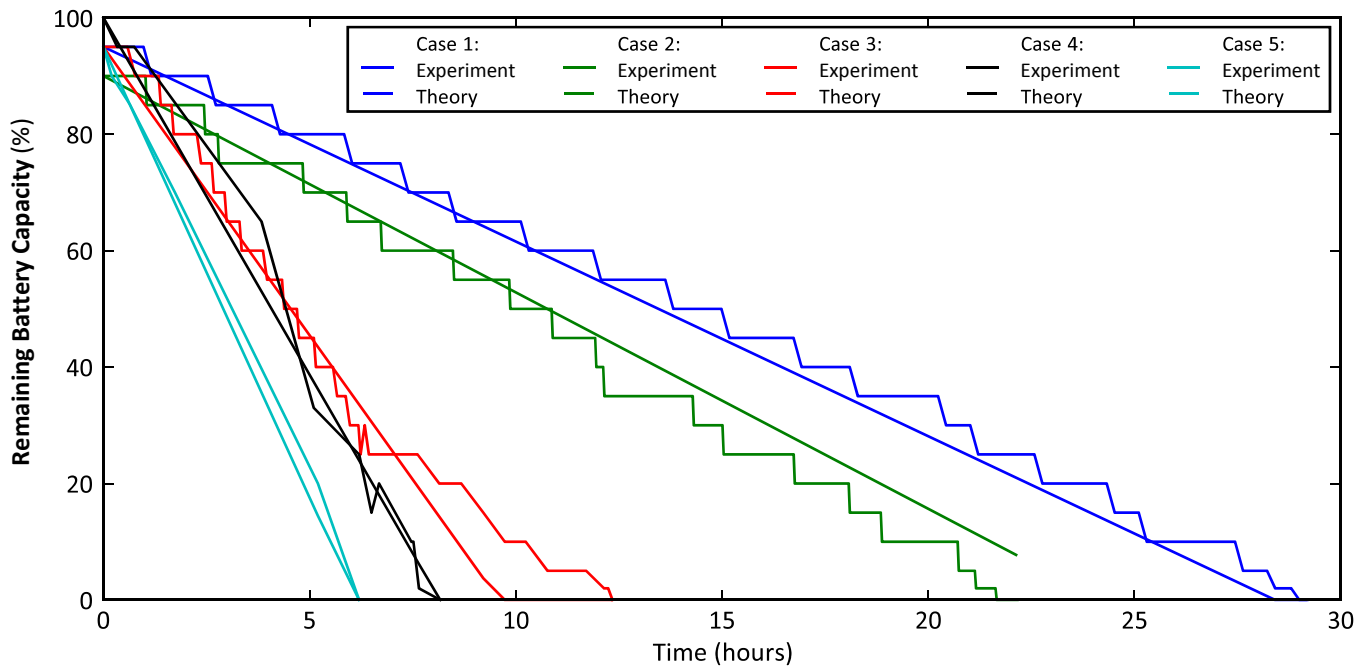


Figure 1: Comparison of Theoretical and Experimental Battery Drain.

a more robust estimate of the total time spent transmitting, and provides a strict  $R_s$  of 0.6. With these parameters a total sensor life of approximately 22 hours was achieved.

*Test Case 3* introduces an additional variable,  $N_c$ . By writing to the sensor's command buffer while simultaneously dumping data the rate of discharge can be increased linearly. With a single sensor writing and reading,  $N_c$  is equal to 2. Additionally, the total read cycle was reduced to 75 seconds, for an  $R_s$  of 0.2. As seen in Figure 1, the multiplicative draw from the additional  $N_c$  provides a greatly reduced sensor life of 12 hours, or a 90% reduction in sensor life. It should be noted that during the experimental run the sensor began rebooting as the battery life was reduced, slowing down the attacker's ability to write data effectively. This is reflected in the graph as a change in slope around the 6.5 hour mark where the power draw drops off.

*Test Case 4* uses the same set up as case 3, but eliminates the sleep period all together effectively reducing  $R_s$  to 0. In this case an 8 hour power drain is achieved, or 93% reduction.

*Test Case 5* takes the attack to its theoretical limit using a single attacker node. By appending 17 erroneous bytes to the end of the Large Dump Command, the command packet can be increased to the maximum size allowed by the BLE protocol. Interestingly, this particular device will receive and process the command despite the trailing bytes. This additional 17 bytes per command effectively increases  $N_c$  to 2.5, resulting in approximately 6 hours of total sensor life.

## VIII. POTENTIAL ATTACK MITIGATION

By design BLE establishes ubiquitous connections in order to authenticate devices. This ubiquitous connection methodology,

by default, is what makes the DoSL attack so effective on resource limited WSNs if the appropriate privacy mechanisms in BLE are not implemented. The following subsections will describe different mitigation methods to reduce the impact of DoSL attacks on BLE resource constrained devices.

### A. Resolvable Private Addressing

In WSNs that use static public device addressing, simple device white listing applications will not be effective because of the ability to sniff and spoof. In the Bluetooth specification Version 4.2 [5], a mechanism for creating and using Resolvable Private Addressing (RPA) is defined. Using either a Local Identity Resolving Key (IRK) or Peer Identity Resolving Key (IRP), a RPA can be generated periodically and used to establish connections to host or slave. The RPA key generation algorithm creates a random RPA that can only be resolved by the IRK internal to the system. The RPA address is resolved and authenticated at the Link Layer using Link Layer Device Filtering. If the RPA cannot be resolved, or the address received is public or random, the packet will not be processed and the host will remain in sleep state. In order to maintain device address privacy, the Link Layer must process the RPAs appropriately using Link Layer Device Filtering.

### B. Link Layer Device Filtering

Changes to the protocol that would mitigate the DoSL attack could also cause degradation in the ad hoc extensible networks are formed.

1) *White List*: Device filtering at the Link Layer is accomplished by the Host creating a White List Record with allowed device address. As mentioned in the previous section, RPAs should be used and regenerated instead of using a public address. The RPA will be resolved and compared against the White List. The same White List should be used by the Link Layer to filter Advertisers, Scanners, and Initiators. White List filtering allows the Link Layer to process requests without waking the Host which preserves resources.

2) *Advertising Filter Policy*: When in the Advertising State, the Advertising Filter Policy should be set to the following mode. Process Scan and Connection requests from devices having Resolvable Private Addresses that are on the White List.

3) *Scanner Filter Policy*: The BLE device implementing the Scanner State should utilize the Extended Scanner Filter Policies with the following mode. Only devices with RPAs on the White List should be processed.

4) *Initiator Filter Policy*: The BLE Initiator Filter Policy should be set to the following mode. Only devices with RPAs on the White List should be processed.

5) *Black List*: In addition to properly utilizing all the privacy features that are germane to the Bluetooth Specification Version 4.2, the following Black List should be implemented in the Link Layer. In order to mitigate spoofing attacks between the generation times of each RPA, every RPA should only be used once per connection. After its use the RPA should be placed on Black List that the Link Layer can use for filtering. When a new connection event occurs, the RPA is first compared to the Black List. If the device isn't on the Black List the RPA is then resolved and compared against the White List for processing and added to the Black List. If the device is already on the Black List or the RPA cannot be resolved to a private address on the White List, the connection event will not be processed and the Host will remain in the sleep state.

## IX. CONCLUSIONS AND FUTURE WORK

The system presented in this paper provides an interesting proof of concept for real-world DoSL vulnerability, demonstrating that BLE sensors can be rendered unusable by attackers using a much unsophisticated devices. We have shown that our model for power consumption in DoSL attacks is closely replicated by a real world sensor. These sensing elements designed to last years could be drained in a matter of days, leaving our critical national infrastructure unprotected. There are several large research questions that must be answered about these types of DoSL attacks in the near future. First, a better characterization of the attack in large groups is required. In theory, a BLE device can be connected with up to 7 devices at once. A single Software Defined Radio attack platform could emulate the 6 other devices and drain the sensor even faster. Second, potential defenses against these types of attacks must be investigated. The naive solution, placing a hard limit on the communication time, does not work because critical sensing windows may be missed if the nodes are limited to

arbitrary restrictions on the timing and rate of sensing messages. A combination of research into the true effectiveness of the attack and the potential range of solutions will lead to a better security model for these sensing nodes in the future.

## REFERENCES

- [1] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient mac protocol for wireless sensor networks," in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3. IEEE, 2002, pp. 1567–1576.
- [2] A. Manjeshwar and D. P. Agrawal, "Teen: a routing protocol for enhanced efficiency in wireless sensor networks," in *null. IEEE*, 2001, p. 30189a.
- [3] M. Patel and J. Wang, "Applications, challenges, and prospective in emerging body area networking technologies," *IEEE Wireless Communications Magazine*, vol. 17, no. 1, pp. 80–88, 2010.
- [4] P. Bellavista, G. Cardone, A. Corradi, and L. Foschini, "Convergence of manet and wsn in iot urban scenarios," *Sensors Journal, IEEE*, vol. 13, no. 10, pp. 3558–3567, 2013.
- [5] I. Bluetooth SIG. (2016) Bluetooth smart (low energy) technology. [Online]. Available: <https://developer.bluetooth.org/TechnologyOverview/Pages/BLE.aspx>
- [6] K. Nair, J. Kulkarni, M. Warde, Z. Dave, V. Rawalgaonkar, G. Gore, and J. Joshi, "Optimizing power consumption in iot based wireless sensor networks using bluetooth low energy," in *Green Computing and Internet of Things (ICGCIoT), 2015 International Conference on. IEEE*, 2015, pp. 589–593.
- [7] I. T. Union, "Future technology trends of terrestrial imt systems," *ITU M.2320-0*, 2014.
- [8] M. Brownfield, Y. Gupta, and N. Davis IV, "Wireless sensor network denial of sleep attack," in *Information Assurance Workshop, 2005. IAW'05. Proceedings from the Sixth Annual IEEE SMC. IEEE*, 2005, pp. 356–364.
- [9] D. R. Raymond, R. C. Marchany, M. I. Brownfield, and S. F. Midkiff, "Effects of denial-of-sleep attacks on wireless sensor network mac protocols," *Vehicular Technology, IEEE Transactions on*, vol. 58, no. 1, pp. 367–380, 2009.
- [10] Y. W. Law, P. Hartel, J. D. Hartog, and P. Havinga, "Link-layer jamming attacks on s-mac," in *Wireless Sensor Networks, 2005. Proceedings of the Second European Workshop on. IEEE*, 2005, pp. 217–225.
- [11] C. Chen, L. Hui, Q. Pei, L. Ning, and P. Qingquan, "An effective scheme for defending denial-of-sleep attack in wireless sensor networks," in *Information Assurance and Security, 2009. IAS'09. Fifth International Conference on*, vol. 2. IEEE, 2009, pp. 446–449.
- [12] T. Bhattasali and R. Chaki, "Amc model for denial of sleep attack detection," *arXiv preprint arXiv:1203.1777*, 2012.
- [13] D. R. Raymond and S. F. Midkiff, "Clustered adaptive rate limiting: Defeating denial-of-sleep attacks in wireless sensor networks," in *Military Communications Conference, 2007. MILCOM 2007. IEEE. IEEE*, 2007, pp. 1–7.
- [14] 3GPP. (2015) Narrowband iot. [Online]. Available: <ftp://ftp.3gp> [15] N. Semiconductor. (2012) Nordic 8001 ble chipset datasheet. [Online]. Available: <https://www.nordicsemi.com/eng/Products/Bluetooth-SmartBluetooth-low-energy/nRF8001>
- [15] Fitbit. (2016) Fitbit charge hr. [Online]. Available: <https://www.fitbit.com/chargehr>

Prepublication review/authorization: PP-16-0646M

\*Contact Author: Dr. Sam Farroha