A Honeypot System for Wearable Networks

A. M. Leonard,H. Cai, K. K. Venkatasubramanian
Dept. of Computer Science
Worcester Polytechnic Institute
Worcester, MA, 01609
Email: {amleonard,hcai,kven}@wpi.edu

M. Ali, T. Eisenbarth
Dept. of Electrical and Computer Engineering
Worcester Polytechnic Institute
Worcester, MA, 01609
Email: {mmali,teisenbarth}@wpi.edu

***Abstract*—Securing any information exchanged within a Body Area Network (BAN) from unauthorized tampering is essential to ensure that such systems are *safe*, and thus do no harm, to the people using them. Solutions for enabling information security in BANs require extensive use of cryptographic primitives that involve considerable performance overhead. Consequently, information security is typically not available in wearable technologies. We need adaptive security solutions that increase the level of security in the event of threats but otherwise impose minimal security overhead in order for them to viable for BANs. The first step in building adaptive security for BANs is to detect the threats. In this paper we propose a solution for detecting adversaries attacking the communication channel of a BAN called a *wearable honeypot system*. It works by communicating fake user health information between the base station and a set of designated *decoy* nodes in the BAN. Any alteration of this traffic, in content or arrival time, is considered adversarial tampering. A preliminary implementation of this wearable honeypot system demonstrates that it is effective in detecting a variety of communication attacks on a BAN.**

I. Introduction

Emerging Body Area Networks (BANs) have demonstrated great potential in a broad range of applications in healthcare and wellbeing. A BAN consists of a set of low-capability monitoring devices deployed on a user. These devices continuously monitor the user and provide sensor information to a sink entity called the *base station* for processing. As BANs deal with personal health data, ensuring information security, especially over its communication channel is critical. Implementing communication security solutions on BANs is often very expensive in terms of computation and energy cost [3]. One way of addressing this problem that has been proposed is to make security primitives adaptive, that is keeping the level of security low (e.g., use cost effective crypto with short keys) or bare minimal when the system is operating in relatively safe environments and increasing the level of security (e.g., use complex crypto with long keys) in the event of an attack on the system [3]. *However, in order to adapt to the level of security, it is essential to detect the presence of adversaries to the BAN.*

In this regard, our approach relies on the use of the idea of honeypots in the context of BANs to identify the presence of adversaries (and sometimes determine their capabilities). A *honeypot* is a trap set to detect attempts by adversaries to gain unauthorized access to information systems [7]. Traditional honeypots have been used in enterprise networks and consist of a group of machines that appear to be part of the enterprise's network, but are actually isolated and monitored. The honeypots are designed such that legitimate access to the enterprise network never leads to a honeypot machine. Therefore, any access attempt observed at the honeypot is, by definition, unauthorized. *Detecting adversaries targeting the communication channel in a BAN can be used to adapt the security primitives within the BAN, based on the threat surface.* For a honeypot system to be useful we need it to possess two properties: (1) *ability to attract:* we need to have a high probability of attracting the adversaries toward the honeypot system for their presence to be detected; and (2) *ability to detect:* we need the ability to detect adversaries that are trying to impede the communication channel within the BAN. In this paper, we focus on the second property where the goal is to detect adversaries who are trying to attack the communication channel within the BAN. In essence we are building a *low-interaction honeypot system* that can detect the presence of communication attacks in the network. We leave addressing the first property for future work[1].

Our approach, known as *wearable honeypot*, works by utilizing the base station and one or more dedicated honeypot nodes in the BAN called the *decoy nodes*. The decoy nodes and the base station are continuously having pre-decided fake communication between them at all times. This communication simulates the exchange of sensed data from a sensor device to the base station. Any modification of this fake data en-route either in the form of data tampering or delay in arrival at the base station is considered to be an indication of the presence of adversaries, for which an alarm is generated. An initial implementation of this approach using one decoy node demonstrates the detection capabilities of our approach for variety of attacks on the BAN communication.

Related Work: Honeypots have been proposed for mobile environments before, particularly with smartphones One major area of mobile honeypot development is the smartphone [1], [2], [4], [6], [8]. These approaches are *not* applicable in our case because: (1) they are particularly designed for capable device such as smartphones and are considerably difficult to implement on low capability devices such as nodes and (2) some of these techniques, such as [2], [4], utilize a larger network of other smartphones to determine the presence of attackers which is not applicable for a BAN context. To the best of our knowledge this is first work describing a honeypot system for a network of wearable devices.

The rest of the paper is organized as follows. Section II and III present the system model and wearable honeypot system, respectively. Section IV presents the security analysis for our system. Finally, Section V concludes the paper.

[1]Honeypot systems often also require the *ability to interact with the adversary* to understand their intensions, capabilities. However, such honeypots, often referred to as high-interaction honeypots are expensive and not suitable for the BAN environments as of yet.

II. Problem Statement and Threat Model

A BAN consists of several wireless medical sensing devices that are worn by the user. These sensors continuously collect patient data and send them over a *single-hop* wireless network to a *base station* for further processing. *The goal of the adversary is assumed to harm the user's immediate safety by attacking the communication channel in the BAN.* We make several assumptions regarding their capabilities (i.e., *threat model*). (1) The base station is not compromised and cannot be used to attack the BAN. (2) The user on whom the BAN is deployed is assumed to be the legitimate owner of the BAN and therefore non-malicious. (3) Any adversary on the system is assumed to be stealthy and the goal is to harm the user safety through communication attacks, such as eavesdropping, injection, spoofing, or man-in-the-middle attacks. This means they can compromise the communication channels in the network, unless a high-security channel is used. (4) We assume the adversary cannot access the individual devices in the BAN.

III. Wearable Honeypot

Figure 1 shows the current architecture of wearable honeypot system. It consists of two classes of entities: (1) the base station and (2) several *decoy nodes*. The decoy nodes are specially designated nodes in the network whose only task is to be part of the honeypot system. These are different from the rest of the nodes in the BAN that actually monitor the user' health. We call these user health-monitoring nodes as *legitimate nodes*. To simplify the discussion we will focus on a honeypot with one decoy node. The idea can be easily extended to multiple decoy nodes. Essentially the base station tells the decoy node to send it (the base station) fake sensor data as if the decoy node were a legitimate node in the BAN. The base station already knows what data will be sent and therefore when it receives the fake data from the decoy nodes, checks to see if the received data is same as what it expects. Any discrepancy is detected as an attempted compromise leading to the detection of the adversary's presence.

Communication channels: There are two logical communication channels between the decoy nodes and the base station: a *high-security* channel and an *adaptive-security* channel. Over the high-security channel, the base station sends a *coordination message* to the decoy nodes to inform them of four things: (1) what type of fake data to send to the base station, (2) number of fake data points to be sent, (3) at what frequency, and (4) a seed for a pseudorandom number generator (which we will describe later). Once the coordination message has been received, each decoy node can then send out a set number of what we refer to as *honeypot messages* (i.e., fake sensor data) to the base station. The high-security channel, as its name suggests, should be difficult to compromise and therefore quite expensive on the nodes. We use AES block cipher in CBC mode with 128 bit pre-deployed keys. However, it is used rarely and hence has an acceptable overhead. The honeypot messages are then sent over the adaptive-security channel between the decoy node and the base station. We define adaptive-security channel as the having the level of security that is similar to what the legitimate nodes in the network use. This makes sure that an adversary cannot tell the difference between legitimate node and decoy node traffic. For the purposes of this work, our adaptive-security channel had no security. This means all honeypot messages and messages from legitimate nodes were transmitted in the open. Once all the honeypot messages have been sent, a new

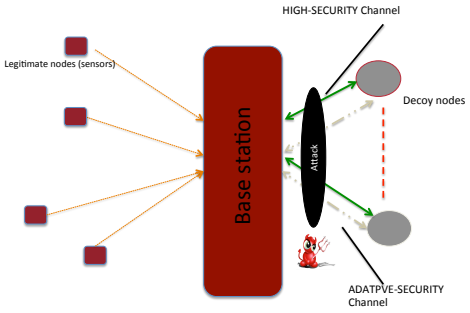

Fig. 1: **Wearable Honeypot System**

coordination message can be issued by the base station that asks the decoy node to sent a new set of honeypot messages. If the honeypot messages are altered in any way, the base station will know that an attacker is actively manipulating the messages. This model has additional utility in that if an adversary passively monitors the channel, it will be unable to distinguish real messages from the false honeypot messages, leading to misinformation.

Honeypot Messages: Obviously, the data in the honeypot messages exchanged are pre-loaded both at the base station and the decoy node. The honeypot messages have to be chosen such that they are fake but seem real enough for the adversary to not be able to tell it apart from real user data. We therefore decided to pre-load the decoy node and the base station with real (triaxial) accelerometer data collected from the UCI Machine Learning Repository (https://archive.ics.uci.edu/). The reason for choosing accelerometer data is two fold: (1) individual values of accelerometer data is difficult to predict, (2) there is higher likelihood of having multiple accelerometer sensors in a BAN compared to other data types (e.g., electrocardiogram (ECG)) and thus it would be reasonable to see multiple streams of accelerometer data within a BAN. The reason we pre-load the accelerometer data is because mathematically synthesizing the data is very computationally intensive. In order to capture the diversity of user activities the accelerometer data pre-loaded on the nodes and the base station are of several types: such as walking, sitting, standing, lying etc. In addition we also pre-loaded data that captures activity transitions such as sitting to standing, standing to walking etc. These provide a couple seconds of realistic transition.

Given that the nodes in a BAN have very limited storage capabilities we do not have the ability to store large quantities of accelerometer data. Consequently, our pre-loaded accelerometer values may repeat after a while, which may result in the data being identified as fake by the adversary who is eavesdropping. Our approach to addressing this issue is to add a small amount of variable random noise to the actual accelerometer data before it is transmitted. For this we implemented a PRNG at both the base station and the decoy nodes. The seed for the PRNG is transmitted as part of the coordination message. Thus for any accelerometer data sent by the node a very particular value of noise (or offset) is also appended to the value, both of which are known to the base station. For this work we utilized TinyMT PRNG [5]. We used TinyMT because it is specifically optimized for low-capability devices such as sensing platforms. Further, it has a period of 2127, and the floating-point numbers are based upon evenly distributed 32 bit integers. In our implementation TinyMT returns a floating point r such that $0 \leq r < 1$. Given

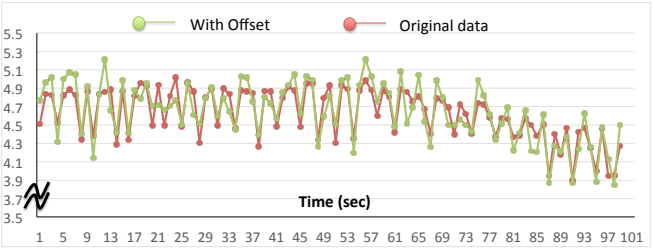

Fig. 2: **Honeypot Messages**

this value we compute an offset $n = (r - 0.5) * std * 2$ to calculate the offset that we want to add to the accelerometer data. Here, std is the standard deviation of the accelerometer data. As our accelerometer data is in three axes, the value n is calculated three times, once for each axis. The value of n will be different for each axis as the std value for the accelerometer data in each axis would be different. Figure 2 shows an example accelerometer data for walking (represented as the magnitude of the x, y, and z axis accelerometer data) and the resultant randomized data (i.e., with offset). It can be seen that randomized data does not repeat and stays in range within the actual activity.

Dealing with Packet Loss: Once we have the honeypot messages being exchanged between the base station and the decoy node, we can use it to detect adversaries. In an ideal communication environment, the detection would be straightforward. A data value (i.e., accelerometer data) X is the expected at time t and data value Y shows up at time t (assuming all HMACs and other error detection/integrity checking codes do not throw an exception), this means the data has been tampered. The attacker has a 1/4906 chance of sending an expected packet within a window, given the nodes use a 12-bit ADC for reading from the accelerometer. Given the low probability of success, it is difficult to inject packets consistently. However, as the BANs use a wireless network, it is possible for the packets to be dropped. In which case when data value X is the expected at time t, nothing appears[2]. In such situations raising an alarm would be premature as we could be just experiencing packet loss. Our approach to dealing with packet loss is to maintain a window of expected data values of size n. As long as the number of data values missing in this window is less than or equal to k, such that $k < n$, we do not raise an alarm. For this work we chose $n = 10$ and $k = 4$. If the number of missing data values increases above k then we raise an alarm that there is a potential adversary in the midst. This message window also protects from replay attacks, as the expected data value is always known, the accelerometer data points seldom repeat, and so an attacker cannot resend an old value. Further, within the message window the average inter-packet latency is tracked. If, within a window, the average inter-packet latency is higher than the inverse of the frequency specified in the coordination packet an alarm is raised.

IV. Security Analysis

The wearable honeypot system has the capability to detect a variety of attacks that adversaries can mount on the BAN to harm the user. We describe each of these attacks and also show how the wearable honeypot system can detect their presence: **MITM attack:** The link-layer protocol (e.g., Bluetooth) often yields several attacks involving disconnecting the base station from the nodes and inserting the adversary in the middle. Doing this would limit the amount of communication and leave the nodes vulnerable and able to be completely hijacked, i.e., disconnected from base station. The adversary now has the ability to pair with the node and become its new master. However, any disconnection of the decoy node from the base station would immediately be deemed as an attack. **Spoofing the node:** Here, the adversary is pretending to be a node already in the BAN. One reason an adversary may want to do this is to confuse the base station and send false information around. This may cause behavior in the BAN that would be detrimental to the user. However, pretending to be the decoy node requires being able to predict the fake accelerometer data accurately in the first attempt. This is extremely difficult to achieve without detection. Further, if the base station receives any packets from decoy nodes before the coordination message was sent, then this also allows adversaries to be detected. **Spoofing base station:** Here, the adversary is a spoofed base station. However, in our approach the decoy node should receive only coordination messages and nothing else. If a decoy node receives any other message form the spoofed base station, the adversary will be detected.

[2]As we are dealing with a single hop network, the delayed packets are not considered here.

V. Conclusions

Preserving information security is essential for BANs in order to ensure user safety. Current information security solutions are computationally expensive and often not available. One way of addressing this problem is to build adaptive security solutions that change the level of security within the BAN depending on the threat present. In this paper we present wearable honeypot system that can detect adversaries and enable adaptive security. In the future we will extend this work by (1) performing a more formal evaluation of the approach, in the presence adversaries and larger number of (potentially virtualized) honeypot nodes; and (2) characterizing the signaling overhead of the honeypot setup over the cost of maintaining only a high-security channel.

References

[1] A. Galante, A. Kokos, and S. Zanero. Bluebat: Towards practical bluetooth honeypots. In *IEEE Int. Conf. on Communications*, 2009.

[2] E. Gelenbe, G. Gorbil, D. Tzovaras, S. Liebergeld, D. Garcia, M. Baltatu, and G. Lyberopoulos. Security for smart mobile networks: The nemesys approach. In *International Conference on Privacy and Security in Mobile Systems (PRISMS)*, 2013.

[3] S. Mare, J. Sorber, M. Shin, C. Cornelius, and D. Kotz. Hide-n-Sense: Privacy-aware secure mHealth sensing. Technical Report TR2011-702, Dartmouth College, Computer Science, Hanover, NH, September 2011.

[4] C. Mulliner, S. Liebergeld, and M. Lange. Poster: Honeydroid-creating a smartphone honeypot. In *Proceedings of the IEEE Symposium on Security and Privacy*, May 2001.

[5] M. Saito and M. Matsumoto. Tiny mersenne twister (tinymt): A small-sized variant of mersenne twiste, January 2011.

[6] E. Vasilomanolakis, S. Karuppayah, M. Muhlhauser, and M. Fischer. 'hostage' a mobile honeypot for collaborative defense. In *7th International Conference on Security of Information and Networks*, 2014.

[7] F. Zhang, S. Zhou, Z. Qin, and J. Liu. Honeypot: a supplemented active defense system for network security. In *Parallel and Distributed Computing, Applications and Technologies, 2003. PDCAT'2003. Proceedings of the Fourth International Conference on*, pages 231–235, Aug 2003.

[8] K. Zolfaghar and S. Mohammadi. Securing bluetooth-based payment system using honeypot. In *International Conference on Innovations in Information Technology*, 2009.