# Battery-Sensing Intrusion Protection System Validation Using Enhanced Wi-Fi and Bluetooth Attack Correlation

Benjamin R. Moyers, John P. Dunning, Timothy K. Buennemeyer, Randolph C. Marchany, and Joseph G. Tront
Bradley Department of Electrical and Computer Engineering
Virginia Polytechnic Institute and State University
Blacksburg, Virginia
{bmoyers, jpvt40, timb, marchany, jgtront}@vt.edu

**Abstract—*This paper discusses mobile device security and extends the original Battery-Sensing Intrusion Protection System (B-SIPS) [1] design by introducing the Multi-Vector Portable - Intrusion Detection System (MVP-IDS). MVP-IDS validates reported anomalous battery depletion from B-SIPS clients with real-time Wi-Fi and Bluetooth traffic using attack signature detection modules. To correlate instantaneous current (IC) anomalies with Wi-Fi and Bluetooth attack traffic, MVP-IDS integrates B-SIPS anomaly detection with the signature-based matching systems of Snort [2] and a newly developed research system, Bluetooth Attack Detection and Signature System (BADSS).***

## I. INTRODUCTION

Personal Digital Assistants (PDAs) and smart phones, also known as Portable Information Devices (PIDs), are less computationally powerful than desktop and laptop Personal Computers (PCs), but possess many of the same features and functionalities. Two defining features of PIDs are IEEE 802.11 (Wi-Fi) and IEEE 802.15.1 (Bluetooth). Though they are similar, many basic security measures common in PCs are not present in PIDs, primarily because of limited power and computation cycle resources. This research shows that the addition of an Intrusion Detection System (IDS) on PIDs can greatly enhance its security.

## II. BACKGROUND AND RELATED WORK

Today's attackers are exploiting PIDs through Denial of Service (DoS) attacks on battery drain, known as sleep deprivation, or battery exhaustion attacks [3, 4]. To combat these attacks, B-SIPS [1] was developed as a client/server-based system for PIDs which detects anomalies in IC drain of smart batteries. The work presented a viable resource-conserving solution appropriate for detecting Wi-Fi and Bluetooth attacks that significantly drain battery power. This was accomplished using a Dynamic Threshold Calculation (DTC) [1] algorithm that continually monitored the PID's IC changes.

Bluetooth exploits have emerged as a new vector for attackers, mainly due to the increasing extent to which the technology is being deployed. O'Connor [5] developed a network-based Bluetooth IDS to discover portable devices under attack within Bluetooth piconets. Data collection has been facilitated by this IDS and because of this capability, many Bluetooth attacks now have attack signatures.

## III. MVP-IDS DESIGN

The main objective of this research is to hinder outside sources from negatively influencing the usability and lifetime, per battery charge, of PIDs. Since PIDs are dependent on mobile battery sources with limited lifetimes, attacks focused on draining battery life can, in effect, produce a DoS on these devices [1, 3, 4, 6].

The original B-SIPS design demonstrated that anomalous IC drains, representing attacks against PIDs, could be recognized and reported to a centralized server for forensic analysis. Moreover, in doing so, it also introduced and uncovered grounds for further research in the attempt to validate these IC drains with actual wireless attack traffic.

While the range of attacks against standard PCs varies greatly from DoS to SQL injections, the range of attacks targeting PIDs is far more specific and specialized. This research effort primarily focused on the need to protect PID and their battery lifetimes from the following classifications of attacks:

1. *Denial of Service:* DoS attacks are used to render some service or resource of a device unavailable.
2. *Device Discovery:* Attacks of this genre focused on discovering devices in range of attack and obtaining their Bluetooth device addresses.
3. *Information Theft:* This type of attack focuses on infiltrating the device to steal a user's confidential information.
4. *Reconnaissance:* These attacks are aimed at recovering device information that could possibly uncover device weaknesses and vulnerabilities.
5. *Service Discovery:* Attacks from this category are aimed at obtaining the types of services that a target device is capable of performing.

The methodology and goal behind MVP-IDS is simple: recognize a significant change in IC on a PID and correlate the change with malicious Wi-Fi or Bluetooth traffic. To accomplish this, MVP-IDS is divided into four distinct modules: *B-SIPS Client* for IC anomaly triggering, *Snort-Based Wi-Fi Module* for Wi-Fi attack detection, *BADSS Module* for Bluetooth attack detection, and

978-1-4244-2515-0/09/$25.00 ©2009 IEEE

*Correlation Intrusion Detection Engine [1]* (CIDE) server for attack correlation and response, as shown in Fig. 1.

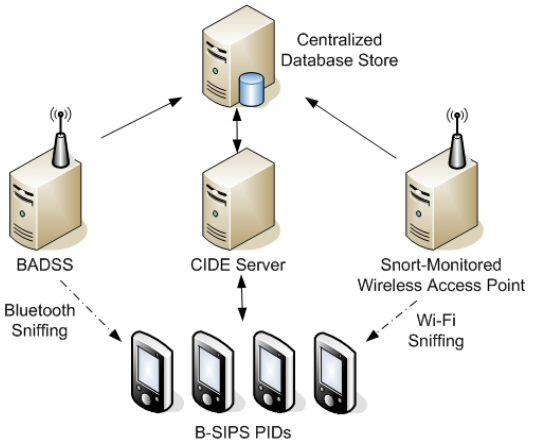


FIG. 1. MVP-IDS system overview.

### a. *B-SIPS Client*

B-SIPS client attack detection is based on irregularities in device IC changes. B-SIPS clients poll the smart battery for voltage, current, temperature, percent battery life, battery flag, and AC line status to determine battery consumption status. The DTC algorithm was developed to deter the system from falsely identifying IC changes as actual attacks by considering known reasons for battery drain, including the starting of known processes and changing in device back lighting. Based on this detection, alert thresholds are adjusted every second to reduce false positives [1].

### b. *Snort-Based Wi-Fi Module*

Since Snort is highly used in industry and recognized as a quality IDS, it was chosen to analyze Wi-Fi network streams between B-SIPS clients and other computing devices. The main objective of the Snort-based module was to ensure that all Wi-Fi traffic was monitored and analyzed.

The design begins with an active Internet connection from an ISP switch to the MVP-IDS monitored subnet. The Snort sensor was then placed between the MVP-IDS subnet switch and a WAP. To allow the Snort sensor to be transparent and still effectively detect Wi-Fi attacks, a network tap was used to passively sniff all transmitted Wi-Fi traffic. Fig. 2 details the Snort-Based Wi-Fi Module and the interactions with each of its components.

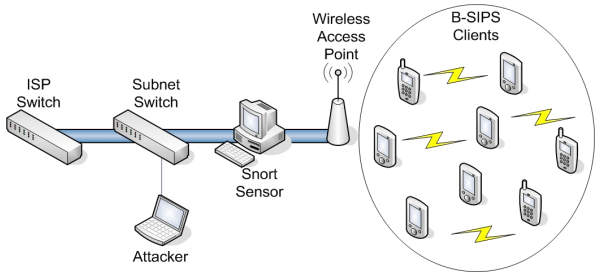


Fig. 2. Snort-Based Wi-Fi module.

From the previously mentioned classifications, a list of 14 different network-based attacks was assembled to gain insight on PID attack response and to test MVP-IDS. Table I shows the attacks and their associated classifications.

TABLE I.
NETWORK-BASED ATTACKS DEPLOYED AGAINST PIDS

| # | Attack Name | Classification |
|---|---|---|
| 1 | Ping Flood | DoS |
| 2 | ACK Flood | DoS |
| 3 | FIN Flood | DoS |
| 4 | PUSH Flood | DoS |
| 5 | RST Flood | DoS |
| 6 | SYN Flood | DoS |
| 7 | URG Flood | DoS |
| 8 | XMAS Flood | DoS |
| 9 | YMAS Flood | DoS |
| 10 | Nessus Default Scan | Reconnaissance |
| 11 | Nmap Intense Scan | Reconnaissance |
| 12 | Nmap OS Scan | Reconnaissance |
| 13 | Nmap Quick Scan | Reconnaissance |
| 14 | Unicorn Scan | Reconnaissance |

### c. *BADSS Module*

The BADSS Module, shown in Fig. 3, was built to recognize Bluetooth attacks and was designed consisting of two main components. First, the Merlin II Bluetooth protocol analyzer [7] was used for Bluetooth packet capturing and exporting of those captures to text files. Next, the *BADSS Intrusion Detection Engine* (IDE) processes each textual packet capture file, attempting to match the file's Bluetooth traffic patterns with attack signatures contained in its signature database. If a match is discovered, the BADSS IDE inserts an attack record into the BADSS attack database.

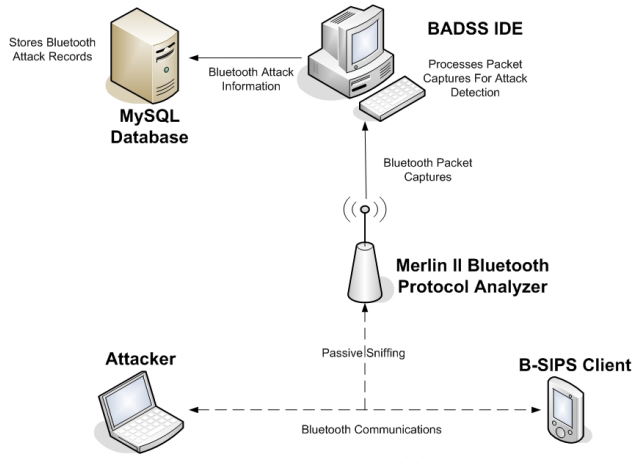


Fig. 3. BADSS module.

This research has assembled a list of 21 Bluetooth attacks, as shown in Table II, and classified each for better understanding. To comprehend Bluetooth attacks, this research utilized the Merlin II Bluetooth Protocol Analyzer, which allowed for viewing of packet-level Bluetooth communications to aid in this understanding. Not only does the Merlin II capture Bluetooth transmissions on the packet level, but it also has a preprocessor engine that assembles multiple low-level packets into a single high-level protocol packet.

TABLE II.

BLUETOOTH ATTACK CLASSIFICATIONS

| # | Attack | Classification |
|---|---|---|
| 1 | RedFang | Device Discovery |
| 2 | BTScanner | Device Discovery |
| 3 | Tbear | Device Discovery |
| 4 | BluePrint | Service Discovery |
| 5 | PSM Scan | Service Discovery |
| 6 | RFCOMM Scan | Service Discovery |
| 7 | BlueBug | Information Theft |
| 8 | BlueSnarf | Information Theft |
| 9 | BTCrack | Information Theft |
| 10 | CarWhisperer | Information Theft |
| 11 | Helomoto | Information Theft |
| 12 | BlueSmack | Denial of Service |
| 13 | Nasty vCard | Denial of Service |
| 14 | L2CAP Header Overflow | Denial of Service |
| 15 | HCIDumpCrash | Denial of Service |
| 16 | Nokia N70 DoS | Denial of Service |
| 17 | Bluetooth Stack Smasher | Denial of Service |
| 18 | Ping of Death | Denial of Service |
| 19 | Tanya | Denial of Service |
| 20 | BlueSpam | Denial of Service |
| 21 | Blueper | Denial of Service |

Signatures for common attacks had to be created in order to develop a signature-based IDS for recognizing Bluetooth attacks. Previously, [5] developed packet-based attack signatures for all but two attacks listed in Table II, those being *RedFang* and *Blueper*. These two new attacks, as well as their associated attack signatures, are described below.

*RedFang:* This device discovery tool is used to find Bluetooth-enabled devices that are operating in non-discoverable mode in close proximity of an attacker. It is a brute force tool that sequentially scans Bluetooth device addresses repeatedly. The signature for a RedFang attack is described below and shown in Fig. 4:

| Step | Device | Action |
|---|---|---|
| 1: | Attacker (Master) | Sends name request to target. |
| 2: | Target (Slave) | Replies with name response. |
| 3: | Attacker | Sends detach command terminating the connection. |
| 4: | Attacker | Sends version request. |
| 5: | Target | Replies with version response. |
| 6: | Attacker | Sends feature request. |
| 7: | Target | Replies with feature response. |

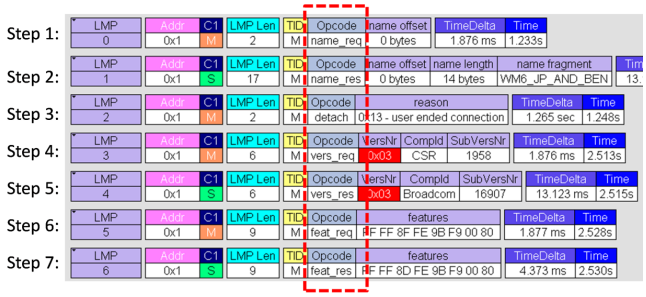


Fig. 4. RedFang attack signature.

All seven communication steps between the attacker and target are accomplished in 1.6 seconds or less. This time was determined by repeated testing and analysis of RedFang packet captures. Many benign Bluetooth communication captures legitimately contain these packet transactions for valid communications. Therefore, in order to recognize a RedFang attack, timing is critical.

*Blueper:* A newly developed attack specifically for this research effort, Blueper aims to exhaust a target PID's battery and memory resources. To do so, it uses the *USSP-Push* tool to flood the target with incoming files. The signature for a Blueper attack is described below and shown in Fig. 5:

| Step | Device | Action |
|---|---|---|
| 1: | Attacker | Sends connection request to target. |
| 2: | Target | Replies with connection response. |
| 3: | Attacker | Sends configure request. |
| 4: | Target | Replies with configure response. |
| 5: | Attacker | Sends OBEX service search pattern. |
| 6: | Attacker | Sends file through RFCOMM protocol. |
| 7: | Attacker | Repeat Step 6. |

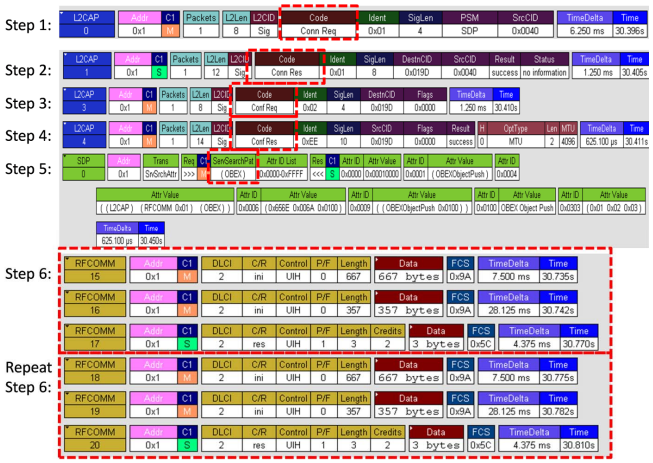


Fig. 5. Blueper attack signature.

Steps 1 through 5 are simply connection setup, thus they are easy to recognize. However, steps 6 and 7 prove to be significantly more difficult to detect. Since files sent from the attacker can be any size, the signature for the Blueper attack is very generic. It operates on the basis of pattern matching, which attempts to look for repeated sets of packet transmissions between the attacker and target devices that are identical.

### *d. CIDE Server*

The CIDE server functions as the supervisor for the system, performing attack correlation and developing grounds for administrative action [1]. The correlation and administrative analysis is done external to the PID by design due to limited memory, battery power, and processing constraints of PIDs. The CIDE server communicates with the Snort and BADSS modules to correlate which attack vector(s) triggered a DTC breach by a B-SIPS client.

Once the CIDE server has information regarding the correlation of an IC anomaly with an associated attack signature, it then sends administrative responses back to the attacked PID. Administrative responses not only tell the PID user that they are being attacked, but also supply details about the attack and administrative actions taken.

Blended attacks are a combination of attacks from both Wi-Fi and Bluetooth vectors simultaneously. An example of

this type of attack is BlueSYN [1]. It was created by combining a Wi-Fi-based *hping*, SYN flood, with a Bluetooth-based *l2ping*, BlueSmack flood. Using the combination of Snort, BADSS, and CIDE server; blended attack recognition and correlation is now possible with MVP-IDS.

## IV. MVP-IDS Limitations

### a. Theoretical MVP-IDS Approach

Conventional IDSs, continually monitor and analyze packet streams to successfully perform attack detection. However, in the mobile environment, resources for this continual task are not available when extended battery lifetimes are required. Due to PID hardware constraints, it was theorized to only have wirelessly transmitted packet streams analyzed once an IC threshold was breached, essentially using anomalies in IC as a triggering mechanism. Raw packet transmissions from Bluetooth and Wi-Fi would be logged to the PID while waiting on the IC trigger, with logs replacing themselves every *X* number of seconds or *Y* number of bytes. Internet domain raw sockets allow raw data, or actual byte transmissions from 802.11 frames, to be forwarded and used by an application [8]. The logs, however, would only be examined by the CIDE server if an IC threshold was breached.

Once an IC anomaly is detected, the B-SIPS client would then send its Bluetooth and Wi-Fi packet logs, along with battery data, to the CIDE server for further attack detection and correlation. The hypothesis of this theoretical approach is that IC anomalies could be validated through the correlation of malicious packet streams occurring in the same time interval and directed at the same PID.

### b. Implemented MVP-IDS Capabilities

Windows Mobile's lack of raw socket access to wireless packet steams forces the theorized approach to be simulated in the attempt to seek validation and predict feasibility. Packet streams still had to be monitored to simulate the theoretical idea, but by a full-featured operating system. The Snort-based Wi-Fi module and the BADSS module were developed because of this constraint. Both modules continually monitor packet streams in real-time and log attack discoveries into databases that can be queried by the CIDE server. Therefore, when the CIDE server receives status reports from B-SIPS clients that contain IC anomalies, it can then correlate the device status report with attacks logged by the Snort and BADSS attack databases.

## V. Results

### a. B-SIPS Client

Sudden changes in battery consumption proved to be a successful triggering mechanism for attack detection in B-SIPS clients [1]. In test bench scenarios and simulations, many attacks preformed against PIDs triggered the DTC, thus forcing complete analysis of Wi-Fi and Bluetooth traffic to and from the device. Using IC dynamic thresholding as a triggering mechanism improves battery life by avoiding unnecessary client/server packet capture traffic for legitimate communications.

### b. Snort-Based Wi-Fi Module

Using network-based attacks from Table I, the goal of this test set was to determine Snort's ability to recognize malicious packet streams and create alerts that were usable by security administrators. Also examined in this module was the theory of Snort's recognition of attacks, solely based on analyzing packet headers.

To produce results for this module, all attacks were launched from an attack laptop and directed at another laptop using Wireshark [9] for traffic capture. As the attacks were deployed, Wireshark recorded them as if they were being seen from a target device's point of view. Once the attack was stopped, the target PC then created a binary dump file that could be sent to Snort for analysis. Snort then parsed the file and returned its results for attack verification. Wireshark was used to capture the packet streams because of its ability to vary the number of packet bytes recorded during each capture section. With this ability, Wireshark first analyzed the attacks with complete packet lengths. Each attack was then repeated with captures of the smallest possible 802.11 frame (68 bytes). This was done so that the capture file would only contain packet headers and exclude payload data.

This research hypothesized that since most of the attacks listed in Table III were based solely on the manipulation of packet header fields; the results would show no difference between full and partial packet analysis. As Table III shows, the prediction was only partially correct. Some attack signatures did not trigger Snort because they apparently relied on payload data analysis.

TABLE III.
SNORT ATTACK RECOGNITION USING VARIED PACKET SIZES

| # | Attack Name | Snort Signature Recognition (Packet Headers Only) | Snort Signature Recognition (Entire Packets) |
|---|---|---|---|
| 1 | Ping Flood | 1/1 | 1/1 |
| 2 | ACK Flood | 1/1 | 1/1 |
| 3 | FIN Flood | 2/2 | 2/2 |
| 4 | PUSH Flood | 1/1 | 1/1 |
| 5 | RST Flood | 1/1 | 1/1 |
| 6 | SYN Flood | 1/1 | 1/1 |
| 7 | URG Flood | 1/1 | 1/1 |
| 8 | XMAS Flood | 1/1 | 1/1 |
| 9 | YMAS Flood | 1/1 | 1/1 |
| 10 | Nessus Default Scan | 12/26 | 26/26 |
| 11 | Nmap Intense Scan | 5/6 | 6/6 |
| 12 | Nmap OS Scan | 4/6 | 6/6 |
| 13 | Nmap Quick Scan | 4/4 | 4/4 |
| 14 | Unicorn Scan | 0/3 | 3/3 |

### c. BADSS Module

As with most software systems, BADSS was tested incrementally during development. As each attack signature was added to the signature database, tests were conducted to determine proper functionality. To perform these tests, Bluetooth packet captures were obtained from [5] and the

Merlin II analyzer. For an attack signature to be effective, it must not only be able to recognize attacks, but also not produce too many false positives to legitimate traffic.

Once all attack signatures were added to the BADSS IDE attack signature database, a comprehensive test was performed. This involved assembling a group of 104 capture files containing legitimate Bluetooth communications, as well as Bluetooth traffic recorded from common attacks listed in the BADSS IDE attack signature database. As Table IV shows, the BADSS IDE has a 100% attack detection rate, while only producing a 2.97% false positive rating.

TABLE IV.
BADSS ATTACK DETECTION RATES FROM PACKET CAPTURE FILES

| Attack | Detection Rate | False Positive Rate |
|---|---|---|
| RedFang | 3/3 | 0/101 |
| BTScanner | 3/3 | 0/101 |
| Tbear | 3/3 | 0/101 |
| BluePrint | 3/3 | 0/101 |
| PSM Scan | 3/3 | 0/101 |
| RFCOMM Scan | 3/3 | 0/101 |
| BlueBug | 3/3 | 0/101 |
| BlueSnarf | 3/3 | 0/101 |
| BTCrack | 3/3 | 0/101 |
| CarWhisperer | 3/3 | 3/101 |
| Helomoto | 3/3 | 0/101 |
| BlueSmack | 3/3 | 0/101 |
| Nasty vCard | 3/3 | 0/101 |
| L2CAP Header Overflow | 3/3 | 0/101 |
| HCIDumpCrash | 3/3 | 0/101 |
| Nokia N70 DoS | 3/3 | 0/101 |
| Bluetooth Stack Smasher | 6/6 | 0/101 |
| Ping of Death | 3/3 | 0/101 |
| Tanya | 3/3 | 0/101 |
| BlueSpam | 3/3 | 0/101 |
| Blueper | 3/3 | 0/101 |
| **Total** | **66/66 = 100%** | **3/101 = 2.97%** |

### *d. CIDE Server*

One of the significant additions that MVP-IDS has incorporated into the CIDE server's functionality is the ability to correlate IC anomalies with real-time Wi-Fi and Bluetooth attack traffic. Testing this functionality was broken into three separate categories, one for each attack medium.

*Real-time correlation with Wi-Fi attacks:* With all Wi-Fi attacks logged to a MySQL database, when a B-SIPS client status report contained an IC anomaly, the Snort attack database was simply queried for matches. To sufficiently correlate an attack, matches were considered confirmed if the Snort attack database contained an attack record 30 seconds before or after the IC anomaly and that the IP address of the B-SIPS client was the same as was contained in the Snort attack record. A 30 second window was chosen as an acceptable time interval because of variable smart battery polling rates [1].

*Real-time correlation with Bluetooth attacks:* Much in the same fashion as the real-time correlation for Wi-Fi attacks, Bluetooth attacks were successfully correlated. When the CIDE server received an IC anomaly from a B-SIPS client, it queried the Bluetooth attack database for an attack record containing a matching Bluetooth device address and timestamp corresponding to 30 seconds before or after the B-SIPS client status report.

*Real-time correlation with blended attacks:* The CIDE server's real-time correlation of a blended attack was merely a check to see if it had already correlated the IC anomaly with both a Wi-Fi attack and a Bluetooth attack from their respective attack databases. If it had, it then categorized the attack as blended. This form of real-time correlation was intrinsically successful based upon the success of the real-time correlation with each of the two previous routines.

## VI. CONCLUSION AND FUTURE WORK

Mobile devices have an inherent need to function under stringent hardware constraints, causing the securing of these devices to often be done as an afterthought in the design process. To mitigate this design weakness and greatly enhance the security of PIDs, MVP-IDS was created. Using a hybrid approach to intrusion detection, our work confirms that PIDs can be secured in malicious environments by integrating IC anomaly triggers with attack signature correlation for Wi-Fi and Bluetooth traffic.

The lack of raw sockets in the Windows Mobile environment has greatly ill-affected the design of this system. A future project could implement a driver-level implementation of the pcap library to allow access to raw socket transmissions for Wi-Fi and Bluetooth, alike. This would eliminate the need for outside wireless traffic monitoring sources, such as the Snort and BADSS modules, and also test the feasibility of this research's theoretical approach to packet capturing.

## REFERENCES


[1] T.K. Buennemeyer, "Battery-Sensing Intrusion Protection System (B-SIPS)," Doctoral Dissertation, Bradley Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Blacksburg,VA, 2008.

[2] Snort, http://www.snort.org/, 2008.

[3] F. Stajano and R. Anderson, "The Resurrecting Duckling: Security Issues For Ubiquitous Computing," *Computer,* vol. 35, pp. 22-26, 2002.

[4] T. Martin, M. Hsiao, Ha Dong, and J. Krishnaswami, "Denial-of-service attacks on battery-powered mobile computers," in *Pervasive Computing and Communications (PerCom '04)*, pp. 309-318, 2004.

[5] T.J. O'Connor, "Bluetooth Intrusion Detection," http://www.lib.ncsu.edu/theses/available/etd-03212008-135411/unrestricted/etd.pdf, 2008.

[6] G. A. Jacoby, R. Marchany, and N. Davis, "How Mobile Host Batteries Can Improve Network Security," *Security & Privacy, IEEE,* vol. 4, pp. 40-49, 2006.

[7] LeCroy, "Merlin II Analyzers," http://www.lecroy.com/tm/products/ProtocolAnalyzers/MerlinII.asp?menuid=60, 2008.

[8] MSDN, "TCP/IP Raw Sockets," http://msdn.microsoft.com/en-us/library/ms740548.aspx, 2008.

[9] CACE Technologies, "Wireshark," 2009.