Polling the Smart Battery for Efficiency: Lifetime Optimization in Battery-Sensing Intrusion Protection Systems

Timothy K. Buennemeyer, Theresa M. Nelson, Randy C. Marchany, and Joseph G. Tront
Bradley Department of Electrical and Computer Engineering
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061
{timb, tnelson, marchany, jgtront}@vt.edu

Abstract

This paper introduces a supporting model for a unique Battery-Sensing Intrusion Protection System (B-SIPS) for mobile computers, which alerts when power changes are detected on small wireless devices. An analytical model is employed to examine smart battery characteristics to support the theoretical intrusion detection limits and capabilities of B-SIPS. Battery-based attack detections can be significantly increased by investigating variable smart battery polling rates, system management bus speeds, and attack execution times. This research explores the modification of smart battery polling rates in conjunction with the variance of malicious network activity. An optimum static polling rate for each of the selected illicit network attack densities was determined by altering these two parameters. These optimum static polling rates introduce minimum and maximum thresholds for the various scenarios mobile devices encounter on a daily basis. Future work will investigate dynamic solutions to optimize battery lifetime under a range of circumstances by encompassing the data results found in this study.

1. Introduction

The primary challenges in developing defensive applications such as intrusion detection systems (IDSs) for small, wireless computers are limited processing capability, memory, and battery resources. Traditionally, network and host-based IDSs employ rules to detect known malicious activity. Anomaly detection systems (ADSs) use statistical methods to establish a system profile and then trigger alerts when that normal profile is violated. This research initiative is developing a battery-based detection system that employs mobile devices as sensors that use an instantaneous current-based threshold algorithm to indicate anomalous activity and trigger alerts.

An indicator that a rogue process is being run on a device without the knowledge of the user is an unexplained increase in the instantaneous current drawn from a device's battery. This could indicate anomalous activity such as a worm spread, virus infection, network probing, flooding, or denial of service (DoS) attack. All of these malicious activities can cause the battery current to rise such that a well-designed system could detect the illicit activity. The *Battery-Sensing Intrusion Protection System* (B-SIPS) detection capability provides security administrators (SAs) with a complementary tool in a network environment as a nontraditional method to detect anomalous battery exhaustion, IEEE 802.11 (Wi-Fi), and IEEE 802.15.1 Bluetooth attack activity that standard IDSs are incapable of detecting [1].

This research examines various means to refine the B-SIPS detection capabilities. Smart battery polling rates, system management bus speeds, and attack execution times can be used to improve the theoretical accuracy of battery-based anomaly detection.

The rest of this paper is structured as follows. Section 2 presents related work. Section 3 discusses the smart battery polling model's design. Section 4 presents the testing and analysis of various battery characteristics during attacks against small mobile computers. Section 5 provides a conclusion and direction for future work.

2. Related Work

The security of power-constrained mobile hosts is generally considered as an afterthought in comparison to service availability. Battery power is an important resource in the wireless domain, especially for small, mobile devices. This presents designers with the perplexing problem of choosing more security at the expense of greater power usage and potentially less service availability. This is an unresolved tradeoff that continues to challenge network and system developers. Establishing secure communication channels through proper authentication could increase service accessibility from a user's perspective but may further increase the device's computational and transmission requirements, leading to faster battery drain.

An Advanced Power Management (APM) technical specification was developed to better manage device power usage to extend battery life [2]. APM is an application programming interface which allowed computer and Basic Input Output System (BIOS) manufacturers to include power management into their BIOS and operating systems (OSs), thus reducing energy consumption. The next evolution in power management was the Advanced Configuration and Power Interface (ACPI) that established an industry-standard for interfaces to OS directed

1-4244-1029-0/07/$25.00 ©2007 IEEE.

configuration and power management on laptops, desktops, and servers [3]. The ACPI specification enabled power management technology to evolve independently in OSs and hardware while ensuring that they continue to work together. The Smart Battery System Implementers Forum offered an open systems communication standard for industry-wide adoption that described data sharing directly between batteries and the devices they powered [4]. Their introduction of a Smart Battery Data (SBData) specification was used to monitor rechargeable battery packs and to report information to the System Management Bus (SMBus), which implemented a two-wire bus design to communicate battery data directly to the device [5] [6].

Stajano and Anderson [7] suggested the idea of energy depletion attacks as early as 1999, which they described as *sleep deprivation torture*. An emerging class of attacks, battery exhaustion and denial of sleep attacks represent malicious situations whereby the device's battery has been unknowingly discharged, and thus the user is deprived access to information [8] [9]. Since system designers of energy-constrained devices incorporate power management to monitor active processes and to shutdown unnecessary components, sleep deprivation and power exhaustion attacks seek to invade and exploit the power management system to inhibit the device's ability to shift into reduced power states.

In analyzing battery attacks against laptop computers, Martin et al. [8] further subdivided sleep deprivation attacks into three basic categories: service-requesting, benign, and malignant power attacks. A service-requesting power attack attempts to repeatedly connect to the mobile device with genuine service requests with the intent of draining power from the device's battery. A benign power attack attempts to start a power demanding process or component operation on the host to rapidly drain its battery. A malignant power attack successfully infiltrates the host and changes programs to devour much more power than is typically required.

As mobile computers become more widely adopted and deployed, they become viable targets for attackers. Racic et al. [10] demonstrated successful battery exhaustion attacks that transited commercial cellular phone networks to exploit vulnerabilities in an insecure multimedia messaging service, context retention in the packet data protocol, and the paging channel. These and other attacks could drain the battery power of target devices and render them useless in a short period of time by keeping them in a busy state. Most concerning is the fact that the cellular phone user and network administrator were unaware that the attack was ongoing. An attack of this nature will use more device power, and thus demonstrates the potential effectiveness of an integrated battery-sensing IDS [11].

Nash et al. [12] developed a battery constraints-based IDS for laptop computers aimed toward defending the system against various classes of battery exhaustion attacks. They leveraged the laptop's robust computational power to estimate power consumption of the overall system based on metrics which included CPU load, disk read and write access, and network transmissions and receptions by using a multiple linear regression model. This data was combined with performance data counters in the Windows NT OS environment. Using multiple linear regressions allowed them to find the correlation of coefficients for each of the measured metrics and a way to determine component power usage from the overall device's power consumption. Moreover, they adapted this concept of estimating system-wide power consumption on a per process basis as a method for indicating possible intrusions and identifying rogue processes on mobile devices. As with any trigger-based system, the challenge is in determining the proper thresholds. Unauthorized activity that falls below the settings may go undetected.

For mobile handheld devices, Jacoby [13] developed a Battery-Based Intrusion Detection (B-BID) approach as a purely host-centric IDS solution. This system was comprised of three distinctive IDS applications based on the power capabilities of the device regarding resources and processor clock speeds. At the low power end (fewer resources and slower clock speeds) was the *Host Intrusion Detection Engine*, which was a rules-based program tuned to determine battery behavior abnormalities based on static threshold levels in the busy and idle states. In the mid range, a complementary module called the *Source Port Intrusion Engine* was employed to capture network packet information during a suspected attack. At the high end, the *Host Analysis Signature Trace Engin*e was used to capture and correlate signature patterns using periodogram analysis in the frequency domain to determine the dominant frequency and magnitude (x,y) pairs.

To our knowledge, this system presented the first feasible working IDS solution for a small mobile device using battery constraints. However, this host-based system lacked reporting and correlation capabilities to warn upstream defensive resources in a net-centric environment. Another deficiency was its static threshold setting and that it allowed the user the option to monitor the system automatically or to manually invoke actions to impede an attack. Although the manual approach is possible, it is unlikely that the user would monitor the host continuously and be able to respond fast enough to prevent substantial power depletion on a regular basis. As a concept, the B-BID approach presents fertile ground for further development, scalability, and research extension.

B-SIPS research is developing an innovative battery power constraint-based model and system to help defend small mobile computers, smart cellular phones, and communication-enhanced Personal Digital Assistants (PDAs). Interoperability and low power design were inspired by the demand to significantly increase battery life and thus the usefulness of small mobile hosts. Battery constraint-based intrusion detection and this B-SIPS research endeavor would not be feasible without these technological advances in ACPI and smart batteries.

3. Model Design

When a small mobile device is kept in a high activity state for extended periods of time, the battery power is depleted faster than normal, decreasing its expected charge life. This research seeks to protect the device's battery life by detecting anomalous battery draining activities. The research goal is to optimize the static polling period mobile devices use to determine malicious power consumption. An assumption was made that once an attack is detected it is immediately eliminated, thus affording each attack with a maximum duration of one smart battery polling time interval.

With this assumption stated, an absolute maximum polling rate was established. To accomplish this, specifications associated with the device's processor, SMBus battery polling mechanism, intrusion detection algorithm, and wireless networking transmission were ascertained. The primary mobile device used in this B-SIPS research effort is the Dell Axim X51 PDA. Its specifications regarding the required time to transmit SBData to the processor exceeded that of the data information processing by more than an order of magnitude. The equations to determine the smart battery's maximum polling rate are presented in Fig. 1.

With present technology, smart batteries transmit 879 bits using the SMBus [6], thus requiring a maximum polling rate of 879 divided by the standard 100Kbps transmission rate [14]. This physical constraint is determined to be 8.79ms. With the maximum polling rate calculations complete, the next step in optimizing the communication between the device and its smart battery was to determine the specific rate at which the battery should be polled. In doing so, it was first necessary to consider flaws in the current polling mechanism, how those flaws could be leveraged by malicious users, and how the flaws might be mitigated.

From the attacker's perspective, precisely timed attacks have the potential to defeat the B-SIPS client detection capabilities. If the attacker knew the precise timing of the polling rate of the battery's chipset, then the attacker could attempt to craft intrusion packets to arrive within those limited time windows between the battery's polling intervals, as shown in Fig. 2. The present smart battery specification dictates instantaneous current sampling once per second, so this packet crafting is a possibility, although remote. B-SIPS' answer to this issue is that the attacker will

```
Single_byte_variables = ACLineStatus + BatteryFlag +BatteryLifePercent
    + Reserved1 + Reserved2 + Reserved3 + BackupBatteryFlag +
    BackupBatteryLifePercent + BatteryChemistry ;
Double_byte_variables = BatteryLifeTime + BatteryFullLifeTime +
    BackupBatteryLifeTime + BackupBatteryFullLifeTime +
    BatteryVoltage + BatteryCurrent + BatteryAverageCurrent +
    BatteryAverageInterval + BatterymAHourConsumed +
    BatteryTemperature + BackupBatteryVoltage ;
Transmit_bytes = Single_byte_variables + (Double_byte_variables * 2) ;
Maximum_polling_rate = (Transmit_bytes * 8) / SMBus_Tx_rate
```

Fig. 1. Maximum polling rate determination

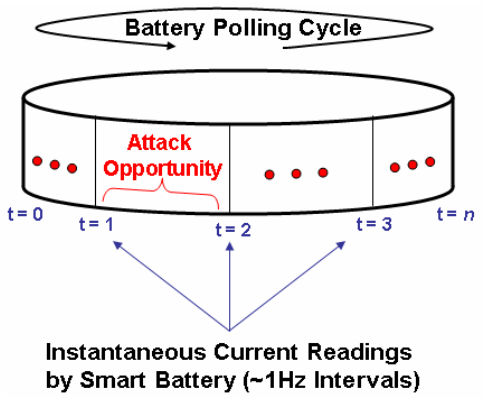

Fig. 2. Battery polling cycle timing attack window

most likely be unable to manipulate both his attack's timing and the energy usage of the targeted device simultaneously. Since the attack is transiting a wireless environment, the timing would be even more difficult, if not impossible to control. Alternatively, if the smart battery could be designed to randomly sample its instantaneous current within that one second interval and still provide comparable performance and diagnostic readings, then this precise timing attack manipulation would be exceedingly difficult to execute.

This leads to a noted limitation that the smart battery provides its diagnostic readings only once per second. At present, original equipment manufacturers (OEMs) have built this generation of smart batteries to provide a limited set of information to the OS for managing the device's power usage and recharging the battery. In the future, if OEMs could improve the smart battery's chipset to poll at a faster rate to accommodate the needs of battery-based IDSs, the timing attack window concern would be mitigated. This would provide the added benefit of potentially helping B-SIPS detect more attacks. The idea hinges on the fact that certain attacks could occur at speeds that exceed the battery's sampling speed, so those attacks could be missed. This research is being conducted to determine the typical speed of attack executions with regard to current device processing rates and bus speeds. Although B-SIPS cannot solve this issue, this research may suggest the appropriate sampling speed for next generation smart batteries to further enhance the detection system's capabilities.

When determining appropriate smart battery sampling speeds, the density of malicious network traffic must be accounted for. B-SIPS' instantaneous current-based threshold algorithm will detect the total number of attacks (num_attacks) against the mobile system throughout its battery discharge period. The system's polling rate refers to the time period, in seconds, between each SBData query, the total battery lifetime refers to the maximum charge life, in mA, dictated by the manufacturer and battery model, and the battery voltage (vcc_battery) represents the standard power drain associated with the device. The parameters used to determine the lifetime of a mobile device are shown in the pseudocode in Fig. 3.

```
for (1/10 sec; 1/10 sec; 600 sec)
  battery_attack_time(sec) = polling_rate(sec) * num_attacks
  battery_attack_mW = battery_attack_time(sec) *
        processor_active_rate(mW/sec)
  battery_attack_mA = battery_attack_mW / vcc_battery
  battery_remaining_mA = total_battery_lifetime(mA)  -
        battery_attack_mA
  battery_remaining_mW = battery_remaining_mA * vcc_battery
  polls_per_min = 60 (sec/min) / polling_rate(sec)
  time_spent_polling = polls_per_min * battery_poll_time
  battery_mW_per_min = (time_spent_polling * processor_active_rate)
        + (time_not_spent_polling * processor_idle_rate)
  battery_remaining_mins = battery_remaining_mW /
                  battery_mW_per_min
  lifetime_mins = ( battery_attack_time(sec) / 60 (sec/hr)) +
        battery_remaining_mins
  if lifetime_mins > max_lifetime
        max_lifetime = lifetime_mins
```

Fig. 3. Static optimum polling rate determination pseudocode

4. Testing and Analysis

The modeled experiment introduces and explores an approach to optimizing the effect of smart battery polling in mobile devices equipped with B-SIPS by changing the static sampling rate. The study intends to increase the lifetime of mobile devices, while decreasing the likelihood of device and application malfunction due to malicious network traffic. To accomplish this, steps were taken to determine the optimum polling rates for various network attack densities, as shown in Section 4.1. Next, Section 4.2 shows how each of the optimum polling rates performs in comparison to the currently implemented polling rate. This section also draws conclusions about the benefits and drawbacks of varying the static polling rates.

4.1. Optimum Polling Rate Determination

Two relevant parameters should be considered when constructing a testing platform and optimization mechanism for smart battery polling schemes. The first of these is the polling rate. By modifying the polling rate, the device obtains an opportunity to improve one of two mutually exclusive lifetime increasing advantages: overhead reduction and rapid attack detection. The first opportunity involves overhead reduction by refraining from polling the battery more often than is necessary for a given set of network characteristics. Devices in low attack density environments benefit by reducing their smart battery polling interval.

The term *attack density* describes the volume of network attacks experienced by a mobile device over a single smart battery discharge period, where attacks represent B-SIPS identified threshold violations. While overhead reduction focuses on *low network attack densities*, rapid attack detection provides a second lifetime increasing advantage. This arises in malicious networks where devices are constantly being bombarded with attacks. *High network attack densities* obtain an advantage by polling the device's smart battery more frequently, due to the assumption that

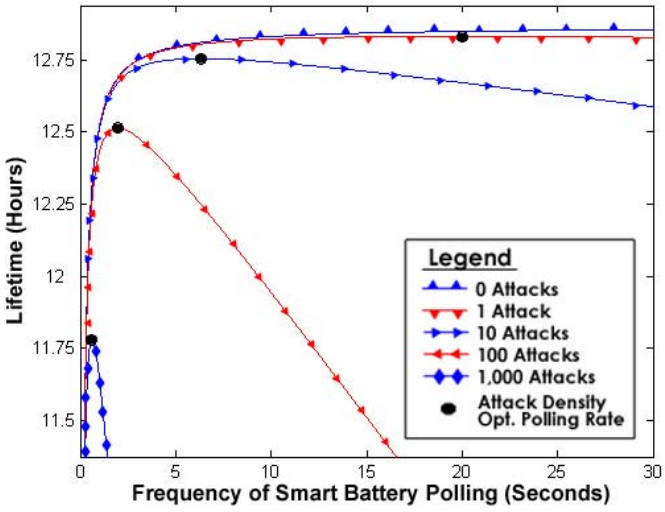

Fig. 4. Effect of battery polling on PDA lifetime

malicious network attacks, once detected, are immediately disarmed. This assumption provides the device with an effective way to save energy by reducing its polling rate, and thus, capturing and disabling malicious traffic in a timely fashion.

As stated previously, these two methods of energy conservation are mutually exclusive, and are dependant on the state of the network attack density. Calculating the most efficient use of the smart battery, while maximizing the security provided by B-SIPS, requires varying the battery's polling rate and the network attack density.

To ensure that various network scenarios were depicted in the study, a *MATLAB* implementation of the pseudocode shown in Fig. 3 was constructed to calculate the optimum polling rate for each of the following network attack densities: 0, 1, 10, 100, 1,000, and 10,000 – 100,000 (in increments of 10,000). For each of these network attack densities, the lifetime was calculated with 60,000 polling rates, ranging from the battery being polled once every 600 seconds (0.00167Hz) to the battery being polled once every 0.01 seconds (100Hz). Once all calculations were complete, the lifetimes associated with each of the polling rates were graphed and the optimum lifetime was denoted with a black circle. The visual representation associated with the Dell Axim X51 specific hardware characteristics is depicted in Fig. 4. Optimum polling rates for each of the tested network attack densities are presented in Table 1.

Table 1.
Network Density Optimum Polling Rates

Number of Attacks	Optimum Polling Rate		Number of Attacks	Optimum Polling Rate	
	Sec	Hz		Sec	Hz
0	600.00	0.0017	40,000	0.08	12.5000
1	20.02	0.0500	50,000	0.07	14.2857
10	6.32	0.1582	60,000	0.06	16.6667
100	1.98	0.5051	70,000	0.06	16.6667
1,000	0.61	1.6393	80,000	0.05	20.0000
10,000	0.18	5.5556	90,000	0.05	20.0000
20,000	0.12	8.3333	100,000	0.04	25.0000
30,000	0.10	10.0000	*	*	*

4.2. Lifetime Calculations

These optimum polling rates were calculated for each of the network attack densities to determine the battery lifetime for each rate under various attack scenarios. A second *MATLAB* function used the polling rate to produce an array containing the lifetimes associated with each of the 15 network attack densities specified in Section 4.1. Running this program sequentially, with various input values, affords users with a useful graphical representation of how modifying the static polling rates affect the discharge time of the device under varying network attack densities.

The dashed lines in the Fig. 5 depict the Dell Axim X51 lifetime spectrum available for each polling rate determined in Section 4.1. The bold hashed line indicates the lifetime associated with the 1Hz polling rate currently used in OEM hardware, and the bold cross-hashed line indicates the lifetime of the fastest polling rate currently permitted by device hardware.

The *x*-axis represents the number of attacks the system undergoes during its battery charge lifetime. For the current polling rate, the smart battery is depleted in less than an hour for network attack densities greater than 20,000; in this research, speedy depletion is referred to as *battery lifetime grounding*. On the opposite end of the spectrum, the network attack density barely affects the lifetime of devices utilizing the minimum polling rate. This minimum polling rate, however, does not have the capability of supplying the device with even half of the lifetime that the current polling rate provides low network attack densities. Several of the optimized polling rates, specifically those above 10,000 attacks, provide a compromise between the lifetimes associated with the currently implemented and the minimum polling rates. These polling rates offer lifetimes slightly shorter than the current polling rate under low network attack densities, while drastically increasing the number of attacks required to ground the lifetime of the smart battery enabled device.

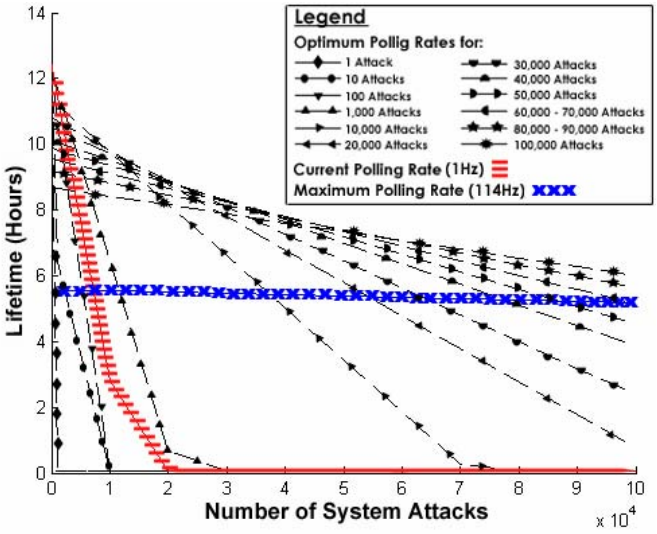

Fig. 5. Effect of polling approaches on smart battery lifetime

Table 2.
Network Optimal Polling Rate Lifetimes

Attack Density	Polling Rate	Lifetime in Hours For # Attacks			
		0	**1**	**10**	**100**
1	**0.05Hz**	12.678	**12.659**	12.482	10.711
100	**0.50Hz**	12.551	12.549	12.532	**12.358**
n/a	**1Hz †**	12.416	12.415	12.406	12.320
10,000	**5.56Hz**	11.342	11.342	11.340	11.326
100,000	**25Hz**	8.6889	8.6888	8.6886	8.6862
		1,000	**10,000**	**20,000**	**30,000**
1	**0.05Hz**	0.0595	0.0595	0.0595	0.0595
100	**0.50Hz**	10.625	0.0595	0.0595	0.0595
n/a	**1Hz †**	11.454	2.7951	0.0595	0.0595
10,000)	**5.56Hz**	11.184	**9.7608**	8.1795	6.5983
100,000	**25Hz**	8.6620	8.4201	8.1513	7.8826
		40,000	**50,000**	**60,000**	**70,000**
1	**0.05Hz**	0.0595	0.0595	0.0595	0.0595
100	**0.50Hz**	0.0595	0.0595	0.0595	0.0595
n/a	**1Hz †**	0.0595	0.0595	0.0595	0.0595
10,000	**5.56Hz**	5.0171	3.4358	1.8546	0.2733
100,000	**25Hz**	7.6138	7.3451	7.0763	6.8076
		80,000	**90,000**	**100,000**	*
1	**0.05Hz**	0.0595	0.0595	0.0595	*
100	**0.50Hz**	0.0595	0.0595	0.0595	*
n/a	**1Hz †**	0.0595	0.0595	0.0595	*
10,000)	**5.56Hz**	0.0595	0.0595	0.0595	*
100,000	**25Hz**	6.5388	6.2701	**6.0013**	*

† Denotes OEM standard smart battery polling rate.

Finally, Table 2 displays the lifetime, in hours, for the optimum polling rate in several representative network scenarios, where attack densities ranged from 1 – 100,000; optimal rate lifetimes are bolded in the table. This data confirms the research hypothesis that overhead reduction and rapid attack detection are mutually exclusive energy conserving methods. While the battery lifetime of a 0.05Hz polling rate is phenomenal when there are no, or few, network attacks, as indicated by overhead reduction, the device resources are rapidly depleted upon entering a network with an attack density of more than 1,000. Using a polling rate of 25Hz, on the other hand, offers the device a minimum of six hours of usage, even in extremely high network attack densities. The drawback to using this polling rate, however, is the overhead associated with the constant transmissions between the battery and device. This overhead causes a decrease in maximum smart battery lifetime to a degree that may not be a feasible expectation for user acceptance.

5. Conclusion and Future Work

The concept of employing battery constraints as a means of intrusion detection is a relatively new capability that was only recently made possible by developments in smart battery and ACPI technologies. The B-SIPS design offers a hybrid intrusion detection method that can serve to protect small mobile computers from anomalous activity which

seeks to drain battery power excessively. This research asserts that small mobile hosts in a net-centric environment can be protected by B-SIPS, which triggers alerts based on power utilization threshold breaches detected by an innovative instantaneous current-based threshold algorithm.

This endeavor explores the optimization of smart battery polling, with an aim to increase device lifetime and also increase the number of anomalous attacks that devices can both detect and disable. The study first determined the maximum polling rate that the Dell Axim X51 PDA could theoretically support. It then calculated optimum polling rates for a variety of network scenarios and used the lifetimes those polling rates provided to compare the effectiveness with that of the currently implemented 1Hz polling rate. As shown in Fig. 5, the optimized polling rates serve as a compromise, both in terms of benefits and disadvantages, between the currently implemented polling rate and the device maximum polling rate. Optimized polling rates prevented the device lifetime from being quickly grounded but were not able to provide the device with lifetimes as long as the currently implemented polling rates during periods of minimal network attack densities.

As an extension of the research presented in this paper, a dynamic polling rate analytical model is being developed and examined. Due to data collected and conclusions drawn from the static simulations of this research, a dynamic implementation of the smart battery polling mechanism is likely to present the best possible solution for B-SIPS detection functionality. This solution will allow devices to maximize their lifetimes for the vast majority of network attack densities, rather than having to select one of two mutually exclusive energy saving schemes, as their static counterparts must do. The dynamic polling rate will inherit its minimum polling rate from the optimum polling rate for a network attack density of 1, or 0.05Hz, and its maximum polling rate from the optimum polling rate for a network attack density of 100,000, or 25Hz. The dynamic polling algorithm will inform the device and its battery when the next poll will occur and will be highly dependent on the state of the current network attack density. The dynamic solution will allow future smart batteries to have the capability to learn about the present state of the device's network and to adapt their polling intervals accordingly. This will, in turn, protect the battery from malicious charge depletion and could also help B-SIPS defend running device applications from being altered, corrupted, or eavesdropped upon.

6. References

[1] T. K. Buennemeyer, F. Munshi, et al., "Battery-sensing intrusion protection for wireless handheld computers using a dynamic threshold calculation algorithm for attack detection," in *the 40th Annual Hawaii International Conference on System Sciences (HICSS-40).* Waikoloa, Hawaii, 2007.

[2] Microsoft, "Advanced power management v1.2," http://www.microsoft.com/whdc/archive/amp_12.mspx, 2001.

[3] ACPI, "Advanced configuration and power interface," http://www.acpi.info, 2005.

[4] SBS_Forum, "Smart battery system implementers forum," http://www.sbs-forum.org, 2005.

[5] SMBus, "System management bus," http://www.smbus.org, 2005.

[6] E. Thompson, "Smart batteries to the rescue," http://www.mcc-us.com/SBSRescue.pdf, 2000.

[7] F. Stajano and R. Anderson, "The resurrecting duckling: security issues for ad-hoc wireless networks," in *the 7th International Workshop on Security Protocols.* Cambridge, UK, 1999.

[8] T. Martin, M. Hsiao, et al., "Denial-of-service attacks on battery-powered mobile computers," in *the Second IEEE Annual Conference on Pervasive Computing and Communications.* Orlando, FL, 2004.

[9] M. Brownfield, G. Yatharth, et al., "Wireless sensor network denial of sleep attack," in *the Sixth Annual IEEE Systems, Man and Cybernetics (SMC) Information Assurance Workshop.* West Point, NY, 2005.

[10] R. Racic, D. Ma, et al., "Exploiting MMS vulnerabilities to stealthily exhaust mobile phone's battery," in *the 15th Annual USENIX Security Symposium.* Vancouver, BC, 2006.

[11] T. K. Buennemeyer, M. Gora, et al., "Battery exhaustion attack detection with small handheld mobile computers," in *the IEEE International Conference on Portable Information Devices (Portable '07).* Orlando, FL, 2007.

[12] D. C. Nash, T. L. Martin, et al., "Towards an intrusion detection system for battery exhaustion attacks on mobile computing devices," in *the Third IEEE International Conference on Pervasive Computing And Communications Workshops (PerCom '05).* Kauai Island, HI, 2005.

[13] G. A. Jacoby and N. J. Davis, "Battery-based intrusion detection," in *the IEEE Global Telecommunications Conference (GLOBECOM '04).* Dallas, TX, 2004.

[14] Epanorama.net, "Serial buses information page," http://www.epanorama.net/links/serialbus.html, 2006.