Mobile Device Profiling and Intrusion Detection using Smart Batteries

Timothy K. Buennemeyer, Theresa M. Nelson, Lee M. Clagett, John P. Dunning,
Randy C. Marchany, and Joseph G. Tront
Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24061
{timb, tnelson, lclagett, jpvt40, marchany, jgtront}@vt.edu

Abstract

This paper introduces capabilities developed for a Battery-Sensing Intrusion Protection System (B-SIPS) for mobile computers, which alerts when abnormal current changes are detected. The intrusion detection system's (IDS's) IEEE 802.15.1 (Bluetooth) and 802.11 (Wi-Fi) capabilities are enhanced with iterative safe process checking, wireless connection determination, and an automated intrusion protection disconnect ability. The Correlation Intrusion Detection Engine (CIDE) provides power profiling for mobile devices and a correlated view of B-SIPS and Snort alerts. An examination of smart battery drain times was conducted to ascertain the optimal transmission rate for the B-SIPS client. A 10 second reporting rate was used to assess 9 device types, which were then compared with their corresponding baseline battery lifetime. Lastly, an extensive usability study was conducted to improve the B-SIPS client and CIDE features. The 31 expert participants provided feedback and data useful for validating the system's viability as a complementary IDS for mobile devices.

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.

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) in a network environment with a complementary IDS tool. This nontraditional method can detect anomalous battery exhaustion, IEEE 802.15.1 (Bluetooth) and IEEE 802.11 (Wi-Fi) attack activity that standard IDSs are incapable of detecting [1].

This research further examines various means to refine the B-SIPS detection capabilities. Smart battery diagnostic readings and system alerts are transmitted in a network centric environment and are used to establish unique power profiles for the mobile devices. Intrusion events are correlated by both network time and device IP address with B-SIPS and Snort IDS reports in order to identify illicit activity and the scope of network attacks. Additionally, smart battery drain times are compared to determine the optimal reporting rate for the Dell Axim X51, and then applied to other mobile devices to better understand the effect of running the B-SIPS client has on a particular system. Lastly, a usability study was conducted to validate the implemented IDS and to further focus efforts to refine the system's overall suite of capabilities.

The rest of this paper is structured as follows. Section 2 reviews background and related work. Section 3 presents the system design and implemented B-SIPS client and correlation engine capabilities. Section 4 discusses the smart battery drain testing results. Section 5 presents an in-depth expert usability study to assess the viability of the system. Section 6 provides a conclusion and direction for future work.

2. Related work

Battery power is an important resource in the wireless domain, especially for small mobile devices. This presents designers with the problem of choosing more security at the expense of greater power usage and potentially less service availability. Establishing

1530-1605/08 $25.00 © 2008 IEEE

secure communication channels through proper authentication could increase service accessibility from a user's perspective, but it may increase the device's computational and transmission requirements, leading to faster battery drain.

The Advanced Power Management (APM) specification is an application programming interface which allowed computer and Basic Input Output System (BIOS) manufacturers to include power management in their BIOS and operating systems (OSs), thus reducing energy consumption [2]. Subsequently, the Advanced Configuration and Power Interface (ACPI) established an industry-standard for interfaces to OS directed power management on laptops, desktops, and servers [3]. The Smart Battery System Implementers Forum offered an open systems communication standard for industry-wide adoption that described data sharing between batteries and the devices they powered [4]. Their Smart Battery Data (SBData) specification was used to monitor rechargeable battery packs and to report information to the System Management Bus (SMBus) [5] [6].

Stajano et al. [7] suggested the idea of energy depletion attacks in 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]. These attacks exploit the power management system by inhibiting the device's ability to shift into reduced power states.

Martin et al. [8] subdivided sleep deprivation attacks against laptop computers. *Service-requesting* attacks try to connect to the mobile device repeatedly with power draining service requests. *Benign* attacks attempt to start a power demanding process or component operation to drain the battery. *Malignant* attacks infiltrate the host and alter programs to devour more battery resources than are typically required.

Racic et al. [9] 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 attacks drained the device's battery, rendering it useless in a short period of time by keeping it 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 [10].

Nash et al. [11] 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 and then adapted this concept on a per-process basis as a method for indicating possible intrusions and rogue applications.

For personal digital assistants (PDAs), Jacoby [12] developed a host-centric *Battery-Based Intrusion Detection* solution. This system was comprised of three distinctive IDS applications. For low power devices, the *Host Intrusion Detection Engine* was a rules-based program tuned to determine battery behavior abnormalities in the busy and idle states using static threshold levels. A complementary *Source Port Intrusion Engine* was employed to capture network packet information during a suspected attack. For robust devices, the *Host Analysis Signature Trace Engin*e was used to capture and correlate spectrum signature patterns using periodogram analysis to determine the dominant frequency and magnitude (x,y) pairs. This system presented the first feasible battery-based IDS solution for PDAs to our knowledge.

B-SIPS research is developing an innovative battery power constraint-based model and system to help defend small mobile computers and smartphones. 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. System design

B-SIPS provides threshold monitoring and alert notification as a host application, which triggers during detected power changes on small wireless devices. These hosts are employed as sensors in a wireless network and form the basis of the *Canary-Net* IDS [13]. This detection capability is scalable and designed to complement existing commercial and open source network IDSs. B-SIPS monitors device power consumption with Bluetooth and Wi-Fi communication activity. Irregular and attack activity is detected and reported to the server for correlation with *Snort* alerts.

The system was developed in Microsoft C# in the .NET Compact Embedded (CE) environment [14]. The client code was ported to run within Windows CE for Mobile 5.0, and the B-SIPS suite of tools is produced for Dell Axim X51v and other Pocket PC type devices. The detection tools were employed on Cingular 8125, Verizon XV6700, Palm Treo 700w and Samsung SCH-i730 smartphones running Mobile 5.0 Phone OS.

B-SIPS detection capability focuses on small mobile hosts that are Bluetooth and Wi-Fi enabled.

Thus, conservation of power is of paramount consideration in determining what information is captured, where the information is stored, when the attack signatures are transmitted, and how intrusion correlation is conducted. B-SIPS alert notification is done on the client device for the user and across the network by a server for the SA. Certain power-depleting attacks such as floods, buffer overflows, and various DoS attacks can be profiled by their pulsing patterns or continuous high drain characteristics, while other attacks merely create temporary spikes in power usage and are much more difficult to pattern. B-SIPS is an ADS and IDS hybrid because it attempts to correlate its alerts with signature-based intrusion reports from other network IDSs.

B-SIPS uses battery constraints and current thresholds to trigger device alerts in idle and busy states. The potential for false positives and false negatives is of great concern. The system strives to minimize both through dynamic threshold tuning. Also, the system attempts to correlate alerts with packet header information for forensic analysis. B-SIPS detects anomalous activity that exceeds the system's dynamic threshold value. The *Dynamic Threshold Calculation* (DTC) algorithm iteratively considers known device processes, backlighting, and system states [1]. Although false positives are a possibility with any detection system, B-SIPS is less prone to false positive alerts because the DTC considers normal device power draining activities and then only triggers an alert when the threshold is exceeded by the device's response to anomalous activity.

B-SIPS calculates the DTC value for comparison with the battery's instantaneous current reading. However, the smart battery only provides the instantaneous current reading once per second, at best, due to limitations in the smart battery chipset. When a threshold breach occurs, B-SIPS transmits reports to a server running the *Correlation Intrusion Detection Engine* (CIDE). The reporting continues while the DTC value is exceeded. Although rapid reporting has a strong potential benefit for early detection and corrective actions by the SA, there is a clear tradeoff in that the client device will expend additional energy to transmit a potentially high volume of reports that could reduce the useful battery life of the device. The increased PDA energy drain is graphically represented to alert the SA in near real-time. CIDE attempts to correlate the transmitted B-SIPS alerts with Snort IDS reports for attack identification and confirmation. Lastly, CIDE provides device power profiling with detailed information to the SA for monitoring purposes.

3.1. B-SIPS mobile device capabilities

The B-SIPS client was originally developed as a means to poll a PDA's smart battery and based on these data readings, then determine if an intrusion had taken place. Through the development of this research, the B-SIPS client evolved from an IDS into a capable mobile device intrusion protection system.

One feature designed to protect the device was the iterative checking of running processes against the *Safe and Unsafe Processes* lists. These lists are loaded by the B-SIPS client application at startup. The Safe Processes list includes all processes the user has determined to be valid to operate on the device. The Unsafe Processes list is a counterpart list, containing process names that are considered by the user to be illicit. All running processes are compared to the list of safe and unsafe processes each time a new process is started. If the process is matched in the unsafe list, then it is terminated. If a new process is not included in the Safe Processes list, the user is alerted with a pop-up notification. The alert gives the user an option to add the process to the Safe Processes list, to kill the process and add it to the list of unsafe processes, or to ignore the running of the process. The running processes are compared to the list of safe processes whenever the number of running processes changes on the mobile device as shown in Figure 1a. This was implemented to increase the code's efficiency and to reduce memory overhead. Because the B-SIPS client should be running continuously, one of the application's design goals was to minimize the resources used, such as device memory, processing power, and battery resources.

In the CE environment, there is a tradeoff between code efficiency and more robust capabilities. This is due to the minimalist design idea behind CE devices. With process list checking, B-SIPS cannot account for a process stopping or starting between comparison intervals. However, the chances of this occurring in a mobile computing environment are remote. Typically, 40-70 processes are running on a Windows XP system, while 14-20 processes are usually running on a Mobile 5.0 enabled device from our observations. In the CE environment, applications are minimized to conserve battery resources, memory, and processor usage. An underlying CE design decision allows applications to continue running in the background instead of closing because it is perceived to be more resource efficient than to terminate and restart the process. With so few active processes, the likelihood is low that a process will start while another process is simultaneously being terminated in the same one second interval. Windows Mobile 5.0 start-up is relatively austere compared to other operating systems, so it is feasible for the device user to know which programs they generally run and

when they are starting a new process.

B-SIPS was enhanced with Bluetooth detection abilities by incorporating *InTheHand.net* libraries from *32feet.net*. The system can display the device's Bluetooth name, which is commonly advertised by a device in discoverable mode. B-SIPS can detect other Bluetooth enabled devices within range that might pose a threat. By displaying the device name and 12-character hexadecimal address to the user, B-SIPS is able to identify an individual device. B-SIPS transmits the device name and Bluetooth address to CIDE, which aids the SA's ability to recognize specific devices.

Reporting Bluetooth enabled devices in range can provide the B-SIPS client user with a sense of plausible attackers in an Internet café scenario. If the user suspects that an attack is occurring against their device, they can discover the intruder's address using B-SIPS *Connections* tab as shown in Figure 1b. The Bluetooth device address uniquely identifies the radio, similar in purpose to a Wi-Fi MAC address for a network interface card. However, leaking the device address information can permit illicit OS or device fingerprinting. This information could prove invaluable when reacting to an intrusion alert and conducting forensic analysis of an attack.

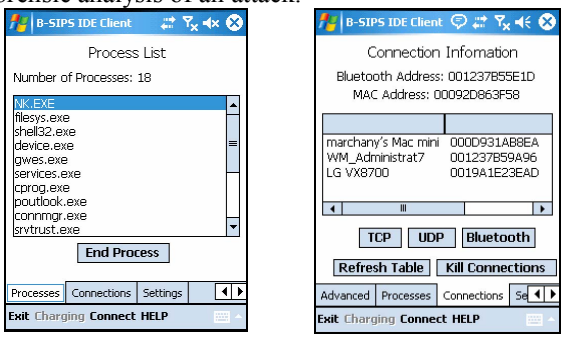

Figure 1. a) Process list and b) connection tab

The last line of defense against an attack is to disconnect the B-SIPS enabled device from the network or paired Bluetooth devices. The user can temporarily disable Bluetooth and Wi-Fi radios using the Connections tab. This ability allows a user to react quickly to an intrusion. B-SIPS also has an automated disconnect capability for when the device is left unattended. After reaching a preset user specified time period tolerance, the B-SIPS client will shutoff both radios to impede an attack with no user intervention.

The B-SIPS Advanced tab displays smart battery values that include: voltage, current, battery life, temperature, battery flag and AC status, which are standard smart battery calls. This information is displayed to the user and sent to the server. The values list is refreshed in one-minute intervals to conserve system memory resources. The memory capacity of many mobile devices is limited compared with notebook and desktop computer systems. B-SIPS is designed to run in the background, so conserving memory resources by reducing displayed and temporarily stored data is a necessary tradeoff for better operating efficiency. The information is still maintained because it is offloaded to the server at regular intervals; hence there is no reason to make more than 60 seconds of diagnostic readings viewable to the device user.

The B-SIPS client is designed with customizable features to accommodate varying user skill levels. Users with advanced computer skills can configure the application to provide more refined detection and alert information, while basic users can effectively operate the system with default settings. B-SIPS includes several personalized setting options such as the ability to enable automatic disconnection upon attack, time until connections are disabled, use of Safe Processes and Unsafe Processes lists, and activation of alerts and context notifications of suspected intrusions.

The B-SIPS client supports mobile device profiling at the server-based CIDE by providing pertinent OS, device specific, and smart battery information. This information includes diagnostic smart battery data, device name, Bluetooth and Wi-Fi MAC addresses, as well as the current number of running processes. With limited resources available on the mobile device, key data is uploaded to the CIDE and stored in the system's backend MySQL database. CIDE provides robust data views, graphical interfaces, intrusion report correlation, and detailed mobile device profiling for the SA. An in-depth examination of how CIDE uses the B-SIPS client provided data, as well as an overview of enhancements to the existing CIDE capabilities suite, will be discussed in Section 3.2.

3.2. Server-based CIDE capabilities

Each device that runs the B-SIPS client has particular energy consumption characteristics from its instantaneous smart battery current fluctuations to the average current trends over time. CIDE tracks this data and calculates each device's average battery current and its associated standard deviation. This allows for a unique profile to be created for each device. As more data is sent to CIDE, a better profile or more mature operating range can be developed. CIDE creates a profile to track mobile device battery current and the number of running processes while the device is not under attack. An average and standard deviation is calculated for each, so if the device reports more than a standard deviation from its mean; CIDE alerts the SA.

This notification is displayed in the form of a list view. Each device sending data to the server has a row

in the list view where the most recent values used in its profile are shown. When a device is operating outside of its profile range, the row is highlighted in red, otherwise the row is green to signify normal operation. Below the list view, a whisker plot is drawn for each device. A black dot indicates the battery current average, vertical black lines show how far the standard deviation extends, and a green dot shows the most recent battery current value. Each whisker plot is set to a different vertical scaling, making it easy to view. Profile graphs are not labeled; instead they utilize brushing and linking. Clicking on a graph will highlight the row in the list view associated with the whisker graph as shown in Figure 2. Conversely, clicking on a row in the list view will highlight the whisker graph associated with it, and selecting multiple rows in the list view will highlight all associated whisker graphs.

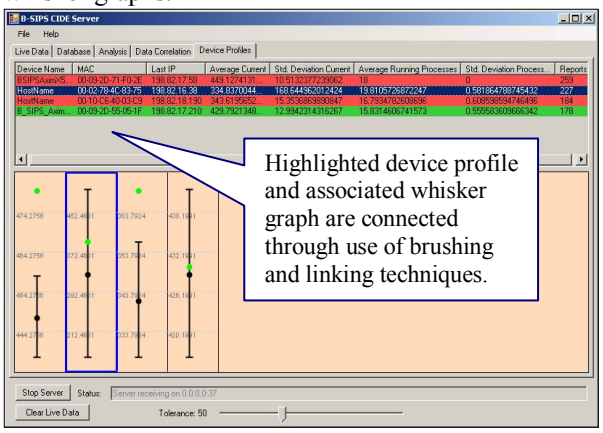

Figure 2. Device power profiling on CIDE

Initially, device profiling was linked to a network allocated IP address. However, these addresses are dynamically assigned within a wireless environment, so this created an issue where different devices' data could be inadvertently merged into the same profile. In the latest revision, each B-SIPS client reports its Wi-Fi MAC address to uniquely identify the device to overcome the identification problem. Using the MAC address ensures that the same device is being profiled each time, since IP addresses are not static in most wireless environments. CIDE stores device profile data in the system's backend MySQL database, including the averages and standard deviations used to calculate a device profile. The list view in the Profile tab can only show a limited amount of information before becoming cluttered. Alleviating this problem, a row in the Profile view can be double clicked to show all available information about a particular device. This action opens a new window, displaying the most recent data received. The detailed profile information window is shown in Figure 3. Any number of windows can be opened at once, allowing a SA to examine the information from multiple devices simultaneously. Furthermore, the information window updates in real-time as new data is received from B-SIPS clients, so the window is never outdated.

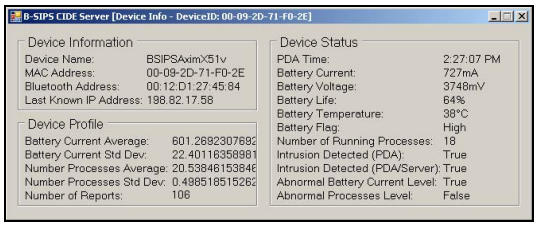

Figure 3. Detailed profile report on CIDE

In addition to the specific device profiling, CIDE correlates B-SIPS client data received from multiple devices with Snort intrusion reports. This is a multi-step process that ensures the proper actions are taken. Correlation analysis method is shown in Algorithm 1.

Algorithm 1. Correlation analysis

New Report Received from Device

```
1:  If New_Report is from device under attack
2:      If Current_Attack does not exist or if
          Current_Attack_Timestamp is older than one minute
3:          Create new Attack and make it
              Current_Attack
4:          Add Current_attack to Attack_List
5:          Set Current_Attack_Timestamp equal to
              New_Report_Timestamp
6:          Add New_Report to Current_Attack
7:      Else
8:          Add New_Report to Current_Attack
```

User Clicked Timestamp to View Attack

```
1:  Set Current_Timestamp equal to timestamp associated with
      user click
2:  Set Clicked_Attack equal to attack in Attack_List that matches
      Current_Timestamp
3:  Get all Reports in Clicked_Attack and add to
      B-SIPS_Data_List
4:  Get all snort data 30 seconds before Current_Timestamp and
      add to Snort_Data_List
5:  Get all snort data 30 seconds after Current_Timestamp and
      add to Snort_Data_List
6:  If Snort_Data_List is empty
7:      Mark Clicked_Attack as Plausible
8:  Else if Snort_Data_List has a IP Destination that matches a
      Client IP in B-SIPS_Data_List
9:      Mark Clicked_Attack as Confirmed
10: Else
11:     Mark Clicked_Attack as Likely
```

When a B-SIPS client is under attack, it will alert the CIDE server of the intrusion event. CIDE will then check a tolerance set by a SA that can be used to filter out B-SIPS clients that are set too sensitively for typical battery current fluctuations. If this threshold is exceeded, the server will flag that particular device as being under attack. If there are no other devices under attack, CIDE will create a new attack group in the *Attack List* in the Correlation tab. For the next minute, every B-SIPS client flagged as being attacked will have its data saved in that attack grouping in the Attack List. This timeframe was chosen because the B-SIPS

client can hold its data for up to a minute before transmitting. If the user sets reporting to 60 seconds, then data from the same attack would be delayed accordingly.

CIDE attempts to correlate Snort data with B-SIPS detected attacks. Since Snort can potentially identify an attack faster by monitoring the network packet data, its timestamp for the attack may differ with CIDE. Correlating with Snort is done by looking up all attack data thirty seconds before and after the first B-SIPS client was flagged as being attacked. In this way CIDE accounts for time differences between the Snort system and the B-SIPS clients. Viewing this data is done by clicking on a timestamp in the Attack List. All B-SIPS client and Snort data that was correlated to the intrusion is displayed to the SA for analysis as shown in Figure 4. Clicking on a timestamp in the Attack List will also display the number of correlated B-SIPS and Snort reports and the assessed significance of the event. There are three attack significance levels: *Plausible* is when Snort reports are not correlated with the attack; *Likely* is when Snort reports were correlated with the attack, but the destination IP address did not match any B-SIPS reports; and *Confirmed* is when Snort reports were correlated with the attack and the destination IP address of both B-SIPS and Snort reports matched.

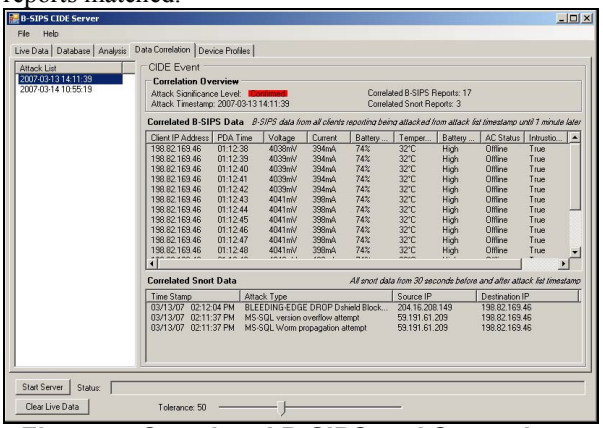

Figure 4. Correlated B-SIPS and Snort alerts

CIDE has a wide range of capabilities, including correlating with a Snort IDS and device profiling. These tools allow the SA to quickly and efficiently determine the attack type and occurrences. CIDE has these capabilities because of its inter-linking with a device running the B-SIPS client. While the server-based CIDE does not suffer from power limitations, B-SIPS clients operating on handheld devices do. Ensuring the effectiveness of the B-SIPS client, smart battery drain testing was conducted to optimize transmission rates for maximum client efficiency and performance while minimizing battery resource drain on mobile devices. Detailed results pertaining to this testing are presented in Section 4.

4. Smart battery drain testing results

Tests were conducted to determine an advantageous reporting rate for the system. The premise was to determine a breakeven point between adequate reporting for timely attack detection and balancing the amount of energy used to transmit the B-SIPS client reports. There is a clear tradeoff with this issue. If the system reports too infrequently, then attack detections can be inadvertently missed by the SA. Additionally, infrequent reports might be occluded in a flooded network environment and never reach the CIDE server for correlation. Alternatively, frequent reporting ensures that more reports are sent and thus logically an improved chance of being received in this system's design. However, excessive reporting can waste battery power because Wi-Fi transmissions are expensive in terms of the device's power usage. Finding a balance between B-SIPS reporting and energy usage supports an aspect of our research premise, the design goal is to use enough battery charge life to detect and protect the mobile device while not wasting it unnecessarily.

For consistency, all of these tests were run under the same conditions in the Virginia Tech Information Security Lab. Ten Dell Axim X51 PDAs were fully charged, and the device settings were standardized with maximum backlight, the processor set at maximum performance, both Bluetooth and Wi-Fi radios enabled, and no running programs other than boot-up processes. The lab temperature remained within 20^0-25^0 Celsius, during testing. This kept the Li-Ion batteries within acceptable operating tolerances to mitigate device recalibration due to temperature effects on the battery power output [15]. The discharge observations were measured using a stopwatch and then recorded. These drain times were converted to seconds and then charted for comparison and analysis.

An early observation with the raw data indicated that the battery drain readings fell within two standard deviations of the mean drain time. Fortunately, this consistency amongst battery discharge readings signified that the device drain characteristics could be used to help determine the optimal B-SIPS code reporting rate. If the device's battery readings were widely scattered or inconsistent, then this optimization method would not have been feasible. The test data indicated that 95% of the population of mobile device batteries will consistently discharge in a relatively short time window, as anticipated by various smart battery manufacturers' documentation and specifications [15].

The assessment of 10 trials with 10 different Axim X51 PDAs was conducted with the goal of minimizing battery power use while maximizing the small mobile computer's ability to detect Bluetooth and Wi-Fi propagated attacks and illicit activities. Figure 5 demonstrates the individual device's battery drain impact within the normalized distribution, and it indicates that the majority of readings fall within 101 seconds (roughly 1.7 minutes) of each other.

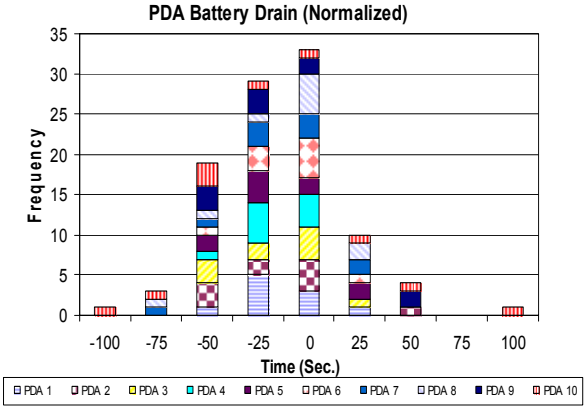

Figure 5. Normalized Axim X51 battery drain

This charting signifies that the battery drain characterization follows a normal distribution, which is important because this observation allows for the determination of an optimized transmission rate for assessing other device models, as well as for comparative statistical analysis in selecting the optimal report transmission rate to sustain the system and while under attack. Following the same 10x10 testing methodology, drain tests were conducted at the following B-SIPS transmission rates: 1, 5, 10, 20, 40 and 60 seconds, using the Axim X51. This in-depth testing determined that B-SIPS transmission rates at 10 second intervals offered the best performance in terms of report rate while conserving device battery resources as shown in Figure 6.

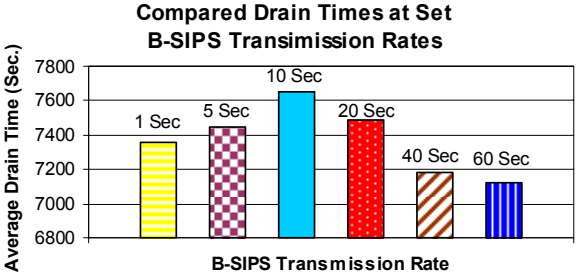

Figure 6. Compared B-SIPS transmission rates

Additional testing was then conducted for 5 PDAs and 4 smartphones. The devices tested were the Dell Axim X51v, X50v, and X30, HP iPAQ 4150 and hx2795, Verizon XV6700, Cingular 8125, Palm Treo 700w, and Samsung SCH-i730. In this testing, each device was observed during 5 trials with its startup processes running, Wi-Fi, and Bluetooth radios operating to establish baseline drain rates. Then each device was tested under the same lab conditions while running the B-SIPS client, using the determined optimal 10 second transmission rate. The drain times for the device baselines versus with B-SIPS client code are compared in Figure 7. The B-SIPS client uses less than 2% battery resources on most devices.

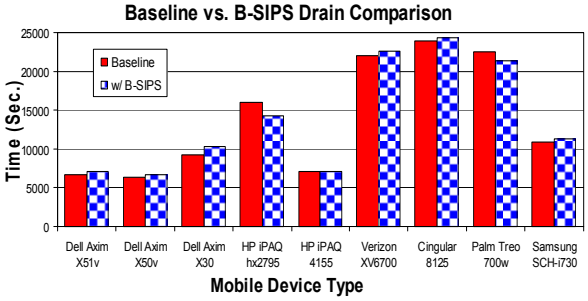

Figure 7. Baseline vs. B-SIPS Drain Time

As a corollary, device smart battery polling rates impact the B-SIPS capability to detect attacks. The theoretical limits of the system's detection capabilities were examined in our related publications [16] and [17]. With today's implemented smart battery technologies, polling rates are significantly slower than those explored in our analytical models. The polling rates for the devices studied in the battery drain testing are presented in Table 1. This table shows device polling rate disparity and indicates which devices are more vulnerable to attack. Devices with slower smart battery polling rates and running older CE operating systems are more susceptible to timing attacks whereby an attack can be executed within the polling cycle of the smart battery. Thus, an attack has a greater likelihood of going undetected.

Table 1. Mobile device battery polling rates

Make	Model	Device Operating System	Polling Rate (Sec.)	Timing Attack Vulnerable
Dell	Axim X51v	Mobile 5.x	9	Medium
Dell	Axim X51	Mobile 5.x	9	Medium
Dell	Axim X50v	Mobile 2003	9	High
Dell	Axim X30	Mobile 2003	2	Medium
HP	iPAQ hx2795	Mobile 5.x	< 1	Low
HP	iPAQ 4155	Mobile 2003	9	High
Verizon	XV6700	Mobile 5.x	9	Medium
Cingular	8125	Mobile 5.x	28	High
Palm	Treo 700w	Mobile 5.x	9	Medium
Samsung	SCH-i730	Mobile 2003	*	High

* OEM not following standard smart battery function calls.

With the device reporting rate optimized, and enhancements made to both the client application and CIDE, the next reasonable step in this research endeavor was to determine how well B-SIPS performed with human-computer interaction. To do so,

a usability study was constructed to evaluate the system's interfaces and capabilities. The subsequent expert feedback and data was then used to refine the hybrid IDS solution to improve network security for mobile devices. This usability study's premise, goals, and outcomes are explained in Section 5.

5. Expert usability study results

A usability study was conducted to validate the functionality and usefulness of the B-SIPS client and server-base CIDE, as well as to determine any areas in which user interaction could be improved. This study adhered to the guidelines and approval requirements of the Virginia Tech Institutional Review Board, as well as following the *Usability Engineering Process* [18], as applicable to this research effort. Prior to participation, candidates were given a brief background primer pertaining to the B-SIPS research area, an outline of what the study entailed and expected of them, and an opportunity to rescind their participation. Those proceeding were given the resources and brief device tutorials necessary to complete a six part automated usability study application. The sections included an entrance interview, which allowed the background knowledge and experience of the users to be gauged, B-SIPS client benchmark tasks, B-SIPS client survey, B-SIPS CIDE server benchmark tasks, B-SIPS CIDE server survey, and an area to voice comments.

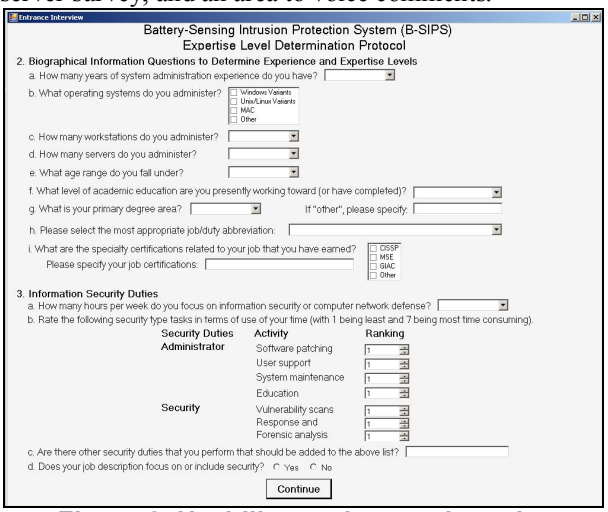

Figure 8. Usability study questionnaire

The usability study focused on user reactions, questions, and comments made by 31 participants performing various benchmark tasks on both the B-SIPS client and CIDE. Experience in terms of security and educational purposes was varied, however 45% had served as system administrators for more than 6 years, as shown in Figure 9, and 25% pursued or possessed Doctoral degrees in Computer Engineering, as shown in Figure 10.

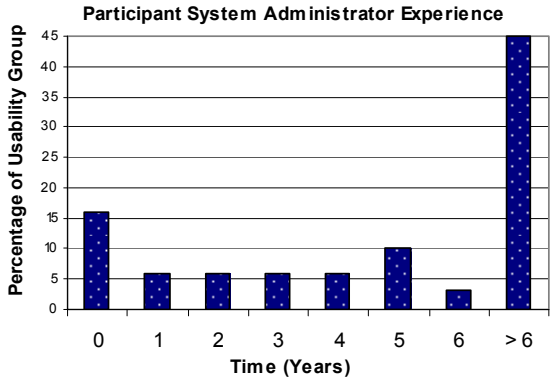

Figure 9. Participant experience level

Although the group was fairly knowledgeable of the fact that network security is a major threat area for small mobile devices, few participants were able to correctly name any Wi-Fi or Bluetooth attacks. This highlights the point that security applications for both general users and system administrators need to be fairly intuitive and must not make assumptions on knowledge that their users may not possess.

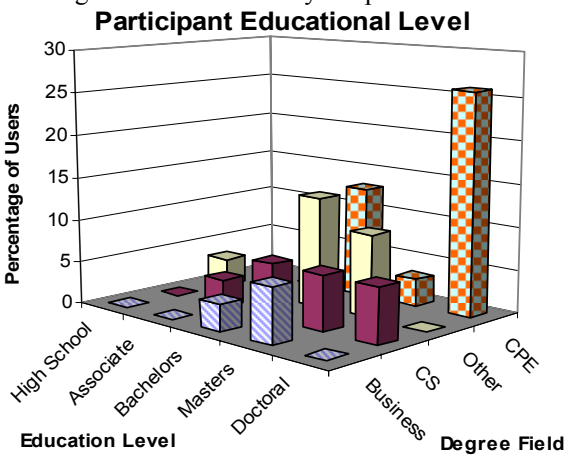

Figure 10. Participant education level

Users were extremely satisfied with the B-SIPS client and server, finding the future relevance of the project to be 4.68 on a scale from 0 to 5 as shown in Figure 11. This indicates B-SIPS fills a security void.

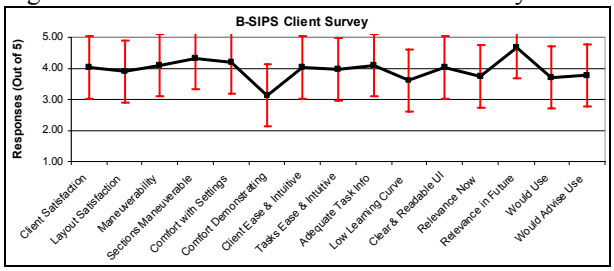

Figure 11. B-SIPS client survey results

Great interest was expressed in requiring the use of such an IDS tool for corporate business devices. Users felt that maneuverability was easy, but that they would not feel comfortable demonstrating it to a new user; this indicates that improvements could be made to decrease the learning curve.

The study participants were even more comfortable using the CIDE environment with standard Windows interfaces. They also felt strongly that the system was intuitively easy to navigate through and manipulate, relevant to security specialists and SAs, and that CIDE technologically enhanced the state-of-the-art in the security field by providing a hybrid IDS solution for monitoring mobile devices. The CIDE usability questionnaire responses are shown in Figure 12.

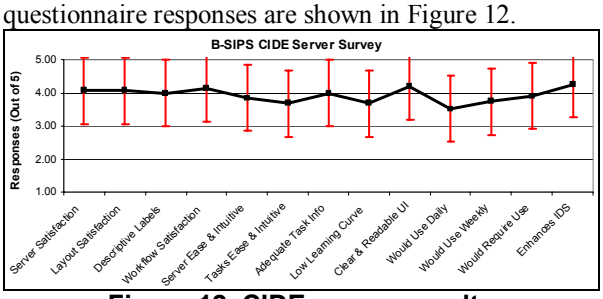

Figure 12. CIDE survey results

Encompassing suggestions made by participants, as well as notes taken by facilitators based on facial reactions during user interactions and statistics gathered from benchmark tasks, a list of usability problems was devised. These problems were evaluated for importance, after which appropriate solutions and man-hour costs were developed. Once this was accomplished, the usability problems were placed in cost-importance tables, one each for the client and CIDE server [18]. A summary of each the B-SIPS client cost-importance issues is shown in Table 2.

Table 2. Client usability issues addressed

B-SIPS Client Issues	Solutions	Resolution
Connections tab unclear	Make labels more clear	Fix
Device Calibration causes user alarm	Add splash screen to explain Calibration	Fix
Set button provides no notification of success	Add pop-ups for Set buttons	Fix
Frequent pop-up alerts can become annoying	Add Turn-Off-Alerts option in Settings tab	Fix, if time
Users unsure of buttons; did not use help file	Hyperlink labels / buttons to help file	Fix, if time
Attack identification in Advanced tab needed	Add Attack column and highlight rows	Fix, if time
Misspellings found	Fix typos	Fix, if time
Process names do not match .EXE names	Add process display name to Process list	Fix in next version
Attack notification not easily accessible	Simplify for easier access	Fix in next version
Keyboard disappears after recalibration	Investigate and mitigate code issue	Fix in next version

Each usability issue is assigned a resolution priority, ranging between *Fix* and *Fix in next version*, which will allow the developers of B-SIPS to focus on correcting the most crucial usability issues first. Priority of effort is placed on improving human interface issues, providing displayed feedback when buttons are selected, improving existing features, and clarifying buttons.

The same usability issue resolution methodology was applied to assessing the CIDE server. Developers focused on correcting necessary usability issues first. Priority of effort is placed on improving the help file, adding data column sorting, incorporating explanation tool tips for buttons and the tolerance bar. Server date and timestamps were added to logs because the CIDE time is more reliable, since it synchronized with a time server. Lastly, summary counters for groups of attack reports are being incorporated. A summary of CIDE server cost-importance issues is shown in Table 3,

Table 3. CIDE usability issues addressed

CIDE Server Issues	Solutions	Resolution
Improve help file	Update help file	Fix
Data sorting required	Allow sorting	Fix
Tolerance feature confusing	Explain tolerance feature	Fix
Right-click not intuitive	Add explanation labels	Fix
Correlation tab unclear	Label display panel as B-SIPS Data	Fix
No server time given	Add time to logs	Fix
Fields do not allow for copy / paste	Allow all fields to be copy / paste capable	Fix, if time
No Live Data date given	Add Date column to Live Data tab	Fix, if time
Correlation does not group attack data	Display attack time range and # of rows	Fix, if time
Correlations cannot be viewed together	Allow selection of multiple correlations	Fix in next version
Data representations are not connected	Add brushing / linking	Fix in next version
Graphs are unclear	Enhance data; use larger text brushing / linking	Fix in next version
Tolerance setting not retroactive for database	Make tolerance retroactive	Fix in next version

The B-SIPS usability study determined that this research endeavor is usable, relevant, and an application that participants would like to see widely used in the future. Many subjects had not previously considered the severity of the lack of network security tools for mobile devices, but after an introduction to B-SIPS, the users indicated an interest in making the deployment of such a system mandatory in corporate settings. Additionally, participants were able to aid the B-SIPS team in pinpointing usability problems and gaining an insight into useful features for inclusion in the next B-SIPS version. Overall, users were comfortable maneuvering with both the B-SIPS client and server-based CIDE, confident that the system was

viable and necessary for the future security of wireless networks and mobile devices, and complimentary of the system design and research progression.

6. Conclusion and future work

This paper discussed enhanced B-SIPS capabilities developed for mobile computers, which included the implementation of iterative safe process checking, wireless connection determination, and an automated intrusion protection disconnect ability. Configurable settings empower the user to adapt B-SIPS to their comfort level. This combined with existing capabilities allows B-SIPS to be a practical and effective IDS tool. CIDE provides mobile device power profiling and an integrated view of correlated B-SIPS and Snort alerts to enable rapid SA analysis of ongoing or past attacks.

In a parallel testing effort, this research examined device smart battery drain results to develop an improved B-SIPS detection capability by balancing timely SA alert notification with device energy consumption. This in-depth testing addressed some of the tradeoffs of operating B-SIPS in terms of battery charge life and device performance. Using Axim X51 PDAs, it was determined that a 10 second reporting rate was the optimal setting for the B-SIPS client. Baseline battery drain testing was then conducted for 9 PDA and smartphone models, and those results indicated that B-SIPS client code used less than 2% of battery resources for most of the test devices compared with their corresponding baseline battery lifetime. Moreover, an assessment of the mobile devices was made about their vulnerability to battery polling timing attacks. The devices were rated from high to low based on their smart battery polling rate and age of OS.

An extensive usability study was conducted to improve the B-SIPS client and server-based CIDE capabilities and features. The 31 expert participants provided feedback and data useful for validating the system's viability as a complementary IDS for mobile devices. The participants overwhelmingly found the system useful and relevant for protecting mobile computing devices. Combining host-based anomaly detection, device profiling, and intrusion correlation with Snort's signature-based identification within a net-centric environment provided a unique hybrid IDS solution where little else currently exists. This was cited by the participants as the research's key strength.

One aspect of future work will be a large scale network simulation using *ns-2*, which will provide optimizations and thresholds pertaining to the network throughput costs of B-SIPS client and server communications. The simulation will also provide a service capacity in the form of a capped number of B-SIPS clients allowable per CIDE server. Finally, an optimal transmission period will be devised.

7. References

[1] T. Buennemeyer, F. Munshi, et al., "Battery-sensing intrusion protection for wireless handheld computers using a dynamic threshold calculation algorithm for attack detection," in *40th Annual Hawaii Int'l Conf on System Sciences (HICSS-40)*, IEEE Computer Soc., Waikoloa, Hawaii, 2007.
[2] Microsoft, "Advanced power management v1.2," http://microsoft.com/whdc/archive/amp_12.mspx, 2001.
[3] "Advanced configuration and power interface," http://www.acpi.info, 2005.
[4] "Smart battery system implementers forum," http://www.sbs-forum.org, 2005.
[5] "System management bus," http://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 *7th Int'l Workshop on Security Protocols*, Cambridge, UK, 1999.
[8] T. Martin, M. Hsiao, et al., "Denial-of-service attacks on battery-powered mobile computers," in *2nd Annual IEEE Conf on Pervasive Computing and Communications*, Orlando, FL, 2004.
[9] R. Racic, D. Ma, et al., "Exploiting MMS vulnerabilities to stealthily exhaust mobile phone's battery," in *15th USENIX Security Symposium*, Vancouver, BC, 2006.
[10] T. Buennemeyer, M. Gora, et al., "Battery exhaustion attack detection with small handheld mobile computers," in *IEEE Int'l Conf on Portable Information Devices (Portable '07)*, Orlando, FL, 2007.
[11] D. Nash, T. Martin, et al., "Towards an intrusion detection system for battery exhaustion attacks on mobile computing devices," in *3rd IEEE Int'l Conf on Pervasive Computing and Communications Workshops (PerCom '05), Kauai Island, HI*, 2005.
[12] G. Jacoby, R. Marchany, et al., "Using battery constraints within mobile hosts to improve network security," *Security & Privacy Magazine, IEEE,* vol. 4, pp. 40-49, 2006.
[13] T. Buennemeyer, G. Jacoby, et al., "Battery-sensing intrusion protection system," in *7th Annual IEEE SMC Information Assurance Workshop*, West Point, NY, 2006.
[14] MSDN, "Microsoft .NET framework developer center," http://msdn.microsoft.com/netframework/, 2006.
[15] Dallas Semiconductor, "Lithium-ion cell fuel gauging with Dallas Semiconductor battery monitor ICs," 2001.
[16] T. Buennemeyer, T. Nelson, et al., "Polling the smart battery for efficiency: lifetime optimization in battery-sensing intrusion protection systems," in *IEEE Southeast Conf*, Richmond, VA, 2007.
[17] T. Buennemeyer, T. Nelson, et al., "Battery polling and trace determination for bluetooth attack detection in mobile devices," in *8th Annual IEEE SMC Information Assurance Workshop*, West Point, NY, 2007.
[18] H. Hartson, "Usability engineering process," http://courses.cs.vt.edu/~cs5714/fall2006/, 2006.