Proceedings of the 2004 IEEE
Workshop on Information Assurance
United States Military Academy, West Point, NY June 2004

Battery-Based Intrusion Detection: A First Line of Defense

Grant A. Jacoby, Randy Marchany, and Nathaniel J. Davis IV, *Senior Member, IEEE*

***ABSTRACT:* This paper proposes a first line of defense early warning system via a host-based form of intrusion detection that can alert security administrators to protect their corporate network(s). This innovative technique operates through the implementation of battery-based intrusion detection (B-bid) on mobile devices by correlating attacks with their impact on device power consumption using a rule-based host intrusion detection engine (HIDE). HIDE monitors power behavior to detect potential intrusions by noting irregularities of power consumption and works in conjunction with a host analysis signature trace engine (HASTE) to provide protection to both mobile hosts and, by extension, their affiliated network.**

Index terms – Intrusion Detection, Wireless Security, Power

I. INTRODUCTION

Battery power in mobile computing is a critical resource: no energy, no computing. Without it, all of the electronic functionality users seek in single handheld devices is simply impossible. As demonstrated by Martin, life for mobile batteries expected to last 30 days may be shortened to as little as one to two weeks when under constant direct attack [1]. Loss of battery power in the commercial and military sectors, particularly on this order of magnitude, equates to mission failure and a loss of revenue and life respectively. While many techniques are used to maximize power, none to date focus on its constraints to determine if an attack is present. This paper proposes how resident monitoring of demands placed on battery power can be used as an early warning form of host-based intrusion detection.

Existing intrusion detection methods are network-centric; however, with the wide-scale proliferation of wireless computing devices, there is a growing need for an efficient host-centric method. To our knowledge, there is nothing in the literature that has theorized and then built an efficient host-centric application for the sake of an Intrusion Detection System (IDS) for smaller mobile devices. Our *Host Intrusion Detection Engine* (HIDE) is a Fuzzy rule-based program that leverages sensing abnormal battery behavior as a means of detecting and then identifying a variety of attacks. This paper provides a brief analysis of the issues surrounding and the experimental work in progress on a viable *Battery-based Intrusion Detection* (B-bid) approach that can integrate relatively easily into the design of any host *and* network-based IDS to provide greater security.

B-bid measures energy use over time to determine if an attack is present. Energy (E) levels can be measured instantaneously or averaged over time (T) on an increasing number of mobile host platforms. Consequently, probabilistic bounds for energy and time can have confidence intervals and are therefore a good means in which to measure abnormal behavior of power dissipation. The technique and efficacy in which these variables of power are measured serves as a profound and viable means for providing additional value to IDS and virus programs as well as protecting battery life.

Moreover, this approach is particularly efficient and straightforward in comparison to present day IDSs which are based on multiple, complex probability theories over multiple variables (i.e., latency, traffic loads, encryption, etc). This approach also addresses a recognized difficulty in anomaly detection in knowing what features of input to monitor, i.e., an attack may alter time of execution and even energy consumption, but it is far more difficult for a hacker to manipulate both energy and time without detection with a B-bid approach integrated into the system. Though not all attacks can be detected, this research indicates an acceptable number of them can be by monitoring power variables and expected bounds of consumption.

The remainder of this paper briefly outlines successful techniques how this can be accomplished and is organized as follows: Section 2 adds some related background issues; Section 3 provides the methodology and design issues for the B-bid approach and Section 4 provides the preliminary testing and results in progress; Section 5 presents and extended analysis representation of feedback; and Sections 6 and 7 outline research work ahead as well as its potential widespread benefits.

II. RELATED ISSUES

Security and power are collectively the two most significant and frustrating issues presently facing wireless systems and network developers. Wireless networks are vulnerable to anyone who knows how to intercept radio waves at the proper frequencies. As noted by Brown, since data is sent

Grant Jacoby is a Ph.D. candidate in and Randy Marchany and Nat Davis are members of the Bradley Department of Electrical and Computer Engineering at Virginia Polytechnic Institute and State University. E-mail: {gjacoby, marchany, ndavis}@vt.edu

0-7803-8572-1/04/$20.00 ©2004 IEEE

through the air, many traditional "wired" network security measures are considerably less effective [2]. Authentication is the most important step for setting up a secure channel for administrators and data authenticity is the most prominent security risk from a user's point of view. Market pressure for authentication to be faster, transparent and more robust is at odds with constraints of small mobile computing. Computing power and bandwidth are scarce commodities. The use of a computationally intensive cryptosystem, such as RSA, may not be a palatable choice in such environments nor is the use of digital signatures to sign every packet with its private key entirely feasible since these measures are prohibitively inefficient. In short, authentication will continue to be a problem and intrusions will occur.

All of these internal calculations require a particular amount of battery power and processor time to be performed. Advanced Power Management (APM) is a specification that defines a layered cooperative environment which allows applications, operating systems (OS), and the system BIOS to work together towards the goal of reducing power consumption in computers. Power management enables systems to conserve energy by using less power when idle and by shutting down completely when not in use, thereby extending the useful life of system batteries without degrading performance. Extensions to this convention, Dynamic Power Management (DPM) techniques, have been suggested by Benini *et al* [3] to take battery constraints into account. However, battery scheduling and management for multi-battery systems [3] [4] [5] do not address system power consumption, but optimize the battery subsystem to best satisfy power requirements. Another organized power-related effort is the Smart Battery System (SBS) forum [6], an emerging industry standard which aims to create open communication standards between batteries and the systems they power to improve battery efficiency, and facilitate interoperability between products from battery, software, semiconductor, and system vendors. Their development of the Smart Battery Data (SBData) Specification monitors rechargeable battery packs and reports information to the Systems Management Bus (SMBus), a simple two-wire bus used for communication with low-bandwidth and power related devices on a motherboard [7]. SBS specifications are the only open system level specifications available today that enable standardization of the electrical and data interfaces by defining the SMBus, the SBData, charger and multi-battery selector commands.

Interestingly, in every case above, low power design and interoperability has largely been motivated by the need to improve battery life by minimizing average power consumption [8]. Yet it is through these developments that B-bid is made possible because truly maximizing battery life requires an understanding of both the source of energy and the systems that consume it -- intended *and* malicious. Recognizing the issue of energy consumption in a mobile environment, power dissipation has rapidly become a first-order design constraint in virtually every type of computing mobile devices and workstation alike. It stands to reason then that it is only a matter of time before (more) attackers prey on battery life.

One possible side effect of attacks is a significant increase in power consumption of the target, thereby decreasing its expected life through excessive battery consumption. The attacks achieve this by keeping the target device busy, and preventing it from going into low power state modes. In some cases, these attacks can be prevented or detected using existing security techniques. Unfortunately, mechanisms to trigger these techniques are currently limited and much power loss and corruption can take place before the techniques are activated. An attack of any kind will consume power and that is why the attack's impact on battery behavior needs to be integrated into IDS and anti-virus programs as an additional layer of defense. Additionally, next generation intrusion detection tools will need to be able to perform correlation analysis of multiple inputs. This research is designing an additional Fuzzy system Host Analysis Signature Trace Engine (HASTE) as the correlation engine for signature identification of a specific sub-set *(dirty dozen)* of known hostile attack signatures using a Chi-Square Tests algorithm for standard deviation to ascertain goodness of fit between pattern matches.

To summarize our research position and work, we advocate the following points:

➢ Tools and mechanisms for efficient host-based intrusion detection are inadequate and require more research and development to fully integrate *B-bid* related resource monitoring of power properties.

➢ *HIDE* software, application or OS integrated, can have positive impacts on host power preservation under forms of high energy attacks and multiple tasks.

➢ *HASTE* software and monitoring needs to be integrated into the defense of mobile hosts as well as in larger corporate network defense strategies to provide an early warning defense system for other network hosts – mobile and wired.

III. METHODOLGY

A. IDS Algorithms

As Cannady has deduced, intrusion-detection systems use several types of algorithms to detect possible security breaches, including algorithms for statistical-anomaly detection, rule-based anomaly detection, and a hybrid of the two [9]. Statistical-anomaly detection systems try to detect security breaches by analyzing audit-log data for abnormal user and system behavior. They assume such behavior indicates an attack is taking place. Profiles of normal user and system behavior that serve as the statistical base for intrusion must be built. The main advantages of statistical anomaly detection are that it does not require prior

ISBN 0-7803-8572-1/04/$20.00 ©2004 IEEE

knowledge of security flaws in network systems to detect possible intrusions and it is able to detect many novel attacks. Rule-based detection can characterize most known attacks by tracing a sequence of events. These events can be modeled into high-level system state changes or audit-log events to form rules bases. Rule-based detection systems monitor system logs and resources, searching for models that match an attack profile or *signature.* Administrators must regularly update the rules to reflect newly discovered attack methods. Because rule-based systems monitor for known attack patterns, they generate very few false alarms.

As Schwartau asserts, an intrusion detection system should be fast enough to catch different types of intruders before harm is done [10]. The goal of HIDE is to alert the user when a suspected attack is underway before irreparable damage is caused while consuming less power than it saves in the process. The alert is a multi-tiered hybrid form of detection, combining the advantages of both rule-based and statistical-anomaly IDS while eliminating some of their disadvantages, such as their inability to detect new methods of attack and the amount by which behavior must deviate from a profile to detect an attack respectively. HIDE first monitors anomalous behavior of the battery when it fails to go into Suspend or Sleep modes. HIDE then, depending on the capabilities of the mobile device, performs one or three of the following operations: sends IDS alarm message to the nearest supporting proxy server for further analysis; then captures an energy signature of the attack and transmit it to same and/or, again depending on the processing and memory capabilities of the computing device, compares the attack signature to a resident *short-list* of known attack signatures. If a match is made, this information is also sent.

B. Skinning Signatures

As it is nearly impossible to capture and match all signature executions, we assert that the most efficacious method in matching is by referencing signatures from the "Top 10" known attack against either Windows or UNIX, operating systems, depending on which the device uses. These attacks are updated annually by the SANS Institute [11] who has determined that the vast majority of successful cyber attacks are made possible by vulnerabilities in a small number of common operating system services. Since most attackers are opportunistic, they take the easiest and most convenient route to exploit the best-known flaws with the most effective and widely available attack tools. Although there are thousands of security incidents each year affecting these operating systems, the overwhelming majority of successful attacks target one or more of these twenty vulnerable services [11]. All the same, if intruders have knowledge of the database of intrusion signatures in an IDS, they can easily attempt attacks that are not represented. Since the Top 10 list is public, we advocate the *Dirty Dozen*: a couple additional signatures from other popular attacks known to affect mobile device applications (i.e., by capturing an illicit rogue *fingerprint*) or which are commonly known in some hostile domains.

C. B-bid Architecture

The B-bid HIDE architecture consists of a software component that uses near real time data to indicate the device's power in Idle state, Busy state and transition between them to detect intrusions. For consistency and handling purposes, only one *software-based* monitoring unit is preferred. In contrast, no matter where or how many embedded monitoring units are placed in the system, final analysis focuses on measuring the rate of power consumption in each state during pre-determined time slices. The more locations and units there are to assist in this, the more heat generated [12], power consumed and chance for inaccuracies in data collection and analysis. Although the most energy efficient method is not necessarily the most effective at detecting attacks and vice versa, HIDE employs more power efficient rule-based Fuzzy techniques as part of an overall cost-benefit consideration in determining the best suited detection methodology and engine.

To conserve energy, HIDE runs periodically as a background process when the battery is not in Suspend or Sleep modes. Once a suspicious change in system power usage is identified, the program will run continuously until two or three threshold violations are captured for solely Idle and Busy states respectively. Determining *normal* thresholds for Idle and Busy states is not difficult, because the absolute minimum current of each state can be determined and calibrated accordingly for each mobile device. Where intrusions are not identified, these are called *false negatives.* Where normal data activities are identified as anomalous, they are called *false positives.* Ideally, an IDS minimizes damages of both true positives and performance impacts of false positives. Thus timing in how and when HIDE runs is a key factor for both power *spared* and performance *preserved* on mobile hosts as part of the cost of providing additional security. For example, if the HIDE program is suspended too long or too often, a damaging attack may take place. On the other hand, if HIDE runs continuously, resource costs may not be justified.

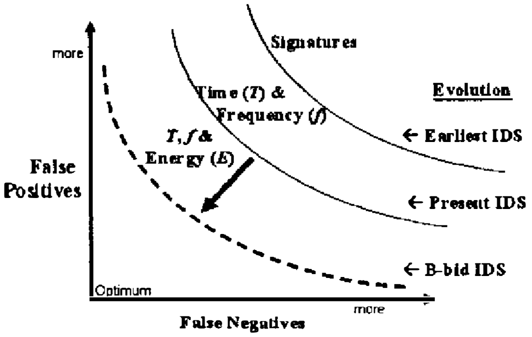

Figure 1 – Direction and Method of B-bid Research

If normal program behavior is not adequately captured, future unseen normal behavior will be classified as anomalous, thus contributing to the false positive rate. As Figure 1 illustrates, a logical and effective solution is to use a hybrid of statistical and rule-based analysis based on Fuzzy logic which handles a specific set of variables. Fuzzy rules allow us to easily construct if-then rules that reflect common ways of describing abnormal battery depletion activities (ABDA). These, in our case, are very specific due to the granularity of the data feeds and are founded on well known and measurable battery constraints. HIDE can be reasonably extended since Fuzzy logic can also adapt for some learning in the forms of weighting given to the Fuzzy input set's defined.

IV. PRELIMINARY TESTING

A. HIDE Design

Determining a practical threshold when the device transitions between power states is more challenging given the variety of configurations and actions possible. It is reasonable, however, to determine effective power consumption thresholds in proprietary devices that have a smaller and standard suite of applications and protocols in which both behavior and usage are well known. HIDE will not invoke a more effective and energy demanding virus scan or IDS program until it detects abnormal power consumption multiple times and has user consent. This leverages the human knowledge factor and essentially confronts the user's willingness to sacrifice expected battery life to protect battery life when the system is under attack. This factor also helps to mitigate false positives. Under normal usage and no attack, other security related software will not be automatically invoked by this pervasive style.

As the HIDE flowchart in Figure 2 depicts, only after a mobile device has a consecutively high rate of consumption in Idle and Busy states does it warrant the user's attention to take action. Juxtaposed along side the flowchart are present day processor and memory capabilities from low to high-end mobile devices capable of performing these functions. In very small devices, only an alarm warning may be possible. Modifications and testing to this flow design is on-going to determine the highest level of accuracy in a variety of user scenarios.

B. HIDE Tool Kit

Although HIDE in practice is not exclusively a host-based detection system, our experimental results focus on attacks against a Dell and NEC PDAs running PocketPC 2003 and CE NET 4.2 OSs respectively. Our methodology and testing has been designed with two additional proof-of-concept goals: to use readily available software and hardware as much as possible and to be a tool readily accessible to users and system/security administrators. To this end, we are using latest versions of VisualStudio.NET 2003 along with the .NET Compact Framework. Given this programming environment, we take a variety of code -- to include the power related structures provided, API member function calls and a few of our own creation -- convert it into C# and then port it over into a variety of mobile device platforms through an emulator. This capability is relatively new and greatly simplifies and empowers the process of developing an application to run on multiple devices.

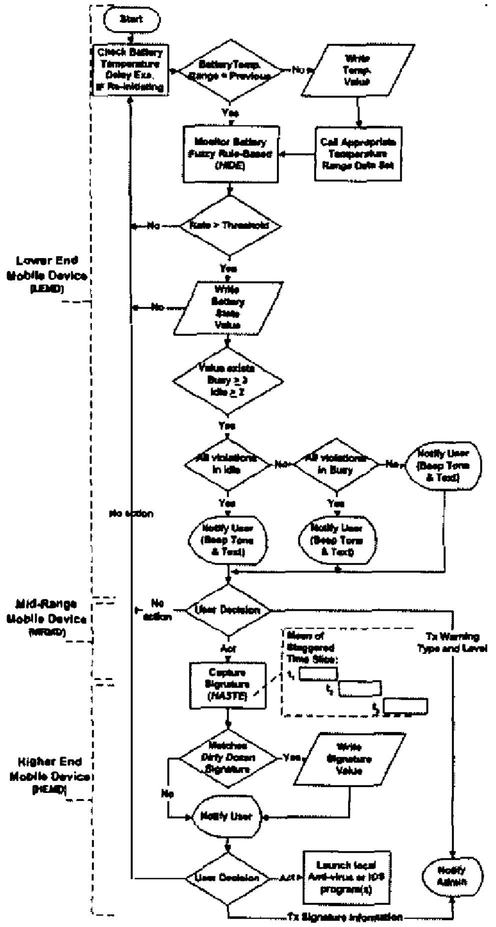
Start
Check Battery Temperature Delay Exe. if Re-initiating
Battery Temp. Range = Previous
No
Write Temp. Value
Yes
Monitor Battery Fuzzy Rule-Based (HIDE)
Call Appropriate Temperature Range Data Set
No
Rate > Threshold
Yes
Lower End Mobile Device (LEMD)
No
Write Battery State Value
Value exists Busy ≥ 3 Idle ≥ 2
Yes
All violations in Idle
No
All violations in Busy
No
Notify User (Beep Tone & Text)
Yes
Yes
No action
Notify User (Beep Tone & Text)
Notify User (Beep Tone & Text)
Mid-Range Mobile Device (MRMD)
No action
User Decision
Act
Mean of Staggered Time Slice:
t1
t2
t3
Tx Warning Type and Level
Capture Signature (HASTE)
Matches Dirty Dozen Signature
Yes
Write Signature Value
Higher End Mobile Device (HEMD)
No
Notify User
User Decision
Act
Launch local Anti-virus or IDS program(s)
Notify Admin
Tx Signature Information

Figure 2 – HIDE Flowchart Design

Despite HIDE being *portable* in this fashion to different mobile platforms, power characteristics of the battery must be calibrated and locally stored, preferably in EEPROM (using EEPROM adds an extra layer of protection for sensitive data if security is compromised). The user or developer must also know which devices are not capable of achieving all four states defined by APM and which are not fully supported for battery readings by OEMs who choose the interfacing chipset and the subsequent function calls they will support. In many cases, if these calls are not required by the operating system, OEMs choose not to do the extra work.

C. Device States and Opportunities

For accurate intrusion detection, we must correctly classify intrusions by state. APM defines four power states: Ready,

0-7803-8572-1/04/$20.00 ©2004 IEEE

Idle, Suspend, and Off. Ready or Busy is when the system or device is fully powered up and ready for use. Idle is an intermediate system-dependent state which attempts to conserve power. Idle is entered when the CPU is idle and no device activity is known to have occurred within a machine-defined period of time. The machine will not return to a Busy state until a device raises a hardware interrupt or any controlled device is accessed. The Idle state is the lowest level of power consumption available in which all data and operational parameters are still preserved [13]. Computation will not be performed until normal activity is resumed. Resumption of activity will not occur until signaled by an external event such as a button press, timer alarm, receipt of request, etc. When in the Off state, the device is powered down and inactive. Operational and data parameters may or may not be preserved in Off state.

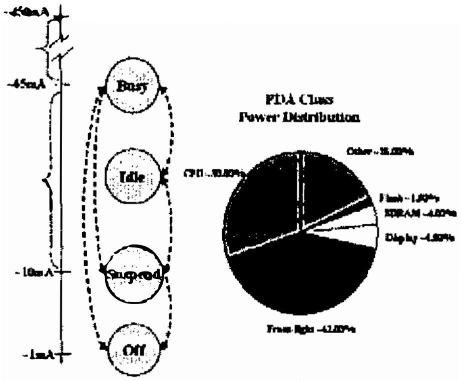

Figure 3 – Idle States and Power Distribution. PDA class power distribution chart taken from [14].

These states and the manner in which power management works in most mobile devices are an opportunity for the attacker as well as for HIDE success. For example, when a PDA such as an iPaq goes into Idle, many of its devices are still receiving power. Figure 3 shows the general current ranges for each state as well as the power distribution for a PDA class of devices [14] in which the CPU accounts for 30% of power and the screen 42% when backlit. In Idle the CPU looses nearly all current and the backlight is turned off, equating to about 64% reduction in power. This can be deceiving however, if the wireless LAN card picks up a network request and transmits an acknowledgement. Worse yet, once on, the card may pick up multiple requests, and unless its communication protocol has been altered it will try to send back an acknowledgement every time and more than once. In addition, the power required to transmit is greater than it is to receive. Even if the mobile device is set not to continue to respond to the same IP address, this defense will fail in the case of a distributed denial of service (DDOS) attack directed at it. All the while, a user may have no knowledge this is happening and the battery is being exhausted in a *higher state of idle*. As Benini *et al* observe, batteries, like those used in many PDAs, are considered exhausted when their output voltage falls below 80% of the nominal voltage (energy that can be obtained from a cell when it is discharged at a specific constant current) [3]. Thus, a user may discover a "dead battery" if this activity is not checked.

Most recent APM features in PDAs affect battery usage time through adjusting the standby period [15]. Nevertheless, this too does not prevent the system from remaining in Idle under DDOS. Similarly, the default setting for a PocketPC is that it will shut off automatically after 3 minutes of inactivity. However, some mobile devices with PocketPC turn on at midnight every night to roll over the calendar for the next day [14]. When the mobile host wakes up, it sends a query to the base station to see if the base station has any data to send. If the WLAN card is inserted in the CF slot, the Pocket PC could possibly remain on until the battery is drained if a DDOS attack is accordingly timed to strike.

D. Initial Testing

Since the range of Idle is known, a reasonably accurate estimate of power consumption can be made for when the device remains in this state. When sufficiently high (abnormal), previously unknown and unmonitored activity levels in Idle are discovered by the B-bid approach. This also holds true if the device remains in an elevated high consumption rate in Busy. Detecting ABDA takes into account that abnormally high power consumption can be a directed attack against the system or battery as well as probable unacceptable rates for conceivably normal activity, in affect protecting the user from both malicious outsiders and himself. With the exception of some proprietary devices, detecting abnormal behavior is more challenging when the device fluctuates between states or the attack remains just under the threshold alarm set by HIDE for the various states.

A *ping flooding* attack is detectable by our iPaq using HIDE in lab settings and we will soon apply all of our *dirty dozen* attacks to determine the fidelity of settings while in Idle and Busy states. Martin recently researched that a website can be power unfriendly by displaying an undetectable animated gif, increasing the average power as much as 80% without the user knowing [1] (for laptops and assuming the animation option is user activated). B-bid using HIDE can easily detect and alert the user to such increases. In fact, HIDE can also be executed on command to monitor websites in the event users believe they may be viewing hostile pages or if they wish to ensure it is running due to a prior alarm sounding or as an extra measure of security.

Since chemical states in batteries are altered as a result of time and environmental conditions, HIDE allows for *relearning* of capacity settings -- provided this is supported by the chipset placed in by the OEM -- to try to offset the effects of aging and temperature (temperature having the greatest impact on discharge rate). As outlined in Figure 2, HIDE adapts to different temperature fluctuations over time (currently set when the temperature reaches 10^0 Celsius change from the last written temperature range).

Fortunately, temperature affects on lithium ion batteries, which make up the bulk of power supplies for small computing device are near linear and flat for "office" temperature of 20-25^0 Celsius [16], meaning there is no need for frequent recalibrations (about once every three months should suffice [17]). We conducted our tests accordingly in this temperature range.

Figure 4 illustrates how the significance of time and energy lost during an attack is both shortened and enhanced by the employment of HIDE. For example, the time lost to (and then salvaged from) attacks during the lifetime of one battery charge should not exceed the time host-based IDS saves. In the event we had allowed the first attack to run until the battery was exhausted, total usable energy would have lasted to *t5*.

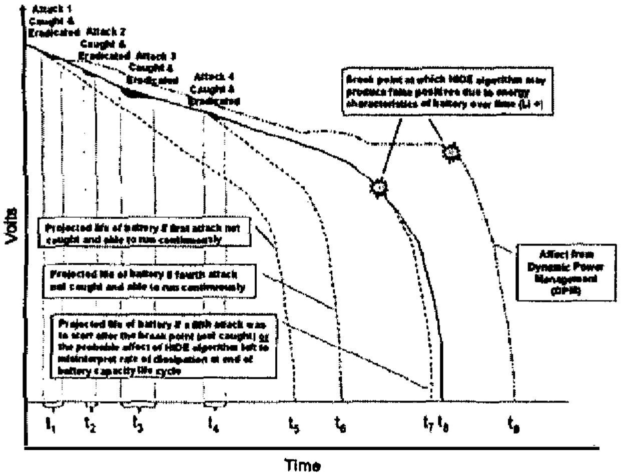

Figure 4 – Li+ Power Dissipation. DPM based off [18]

Power protection is not free and the user will need to decide if HIDE is worth the associated costs. In our initial lab results, HIDE consumed 8% power under normal usage with no attacks and saved 10% power when the system came under four separate attacks. This savings would be less if not all attacks were detected; however, the percentage of power lost (t8-t5/t8) would be greater regardless if attacks did go undeterred. The initial results are promising and more work is currently underway to confirm results under all *dirty dozen* attacks.

V. EXTENDING ANALYSIS

A. HASTE

Though the need for pattern recognition is addressed, the next generation of intrusion detection tools will need to be able to perform correlation analysis of multiple inputs from multiple locations. This research is designing an additional Fuzzy system Host Analysis Signature Trace Engine (HASTE) as the correlation engine for signature identification of the *dirty dozen* using a Chi-Square Tests algorithm for standard deviation. Signature results currently being taken by multi-meter of each attack variety (on a small number of mobile devices) will be used to test if the standard deviation of a population is equal to a pre-specified value to predict the relative frequency of outcomes in each possible attack category.

This goodness of fit or confidence interval improves as the sample size becomes larger. It is certainly the intended case with mobile devices using HIDE to report back power anomalies and, if possible, pattern matching using HASTE. B-bid's combination of statistical anomaly (HIDE) and rules-based (HASTE) detection algorithms to create an improved hybrid over either will increase the amount of system resources required. Additional disk space will be required for the storage of the profiles, and increased memory requirements will be encountered as the engines compare user activities or signatures with information in the two knowledge bases. Nevertheless, users can also execute HIDE and HASTE while plugged into a standard alternate current (AC) power supply. This will almost certainly be the case for users with mid-range mobile devices who do not have a higher authority to report HIDE detected anomalies. In this circumstance, users could put their device in an active AC cradle, capture the signature and then send it to a signature analysis program residing in their own workstation or to one possibly offered on-line. In either case, a viable and essential mobile *triage* of sorts is offered to mobile hosts that does not exist today.

To facilitate accurate capturing of signatures, Figure 2 shows that a mean average from three staggered times slices will be taken to mitigate irregularities. This procedure is conducted since, as Cannady notes, slight variations in an attack sequence can affect the comparison of the activity in the audit record to the existing rule to a degree that the intrusion is not detected by the intrusion detection mechanism [9]. Kumar notes that matching is also problematic in that due to all the intervening events several partial matches can occur which requires overhead time to track each partial match [19]. In addition, the time of these slices is device-dependent and will need to be limited to save resources. If a widespread (distributed) denial of service attack is underway, our multi-tier strategy harnesses and capitalizes from feedback provided from the most vulnerable and weakest processors members in a network to serve as a first line of defense early warning system for other stronger and more protected members of the network (behind the firewall). This conceivably would provide security administrators precious extra hours of response time to analyze network traffic, offering a golden opportunity to recognize and thwart attacks before they spread to the inner corporate network.

B. Correlating Perimeter Feedback

An assertion of this paper is that detection efforts can be more effective by correlating the outputs of diverse sensors and obtaining information from multiple locations. Although the B-bid approach may appear to only serve the protection interests of the mobile host, when abnormal battery depletion activities are detected and captured signatures are sent to a security administrator their collective threat analysis can be a significant visual enhancement to attack graphs. As Jha concludes, attack graphs can enhance both heuristic and probabilistic correlation approaches as well as legitimize the potential effectiveness of the intrusion detection system by the combined capability to identify patterns which indicate intrusive behavior [20]. Prior research papers by Erbacher on visualizing network intrusion data [21] [22] declare little prior work has been done in this area, particularly real-time network intrusion data. Moreover, there has been no discussion as to how to effectively collect and correlate relevant information from mobile host systems.

Given that the data is available and can be sent to the appropriate administrator or analysis server, the information determined by HIDE and HASTE can add and extend a layer of defense. This can be accomplished by adding thresholds from mobile host reports to existing network monitoring tools. Like Porras states, the goal of threshold detection (also known as summary statistics) is to record each occurrence of a specific event and detect when the number of occurrences of that event surpass a reasonable amount that one might expect to occur within a specified time period [23]. These events are then graphically projected to clearly highlight unnaturally high number of occurrences within a short period of time. Thus, integrating HASTE and HIDE feedback can be an effective early warning system that should benefit other segments of the corporate network.

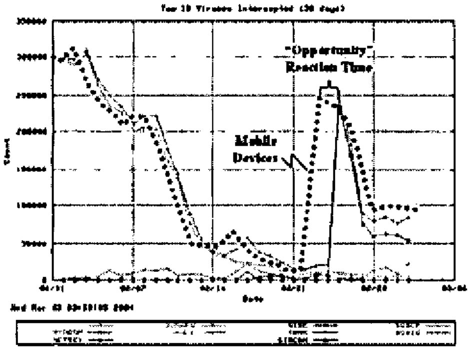

Figure 3 – Viruses Intercepted. Graph background from [24]

For example, the HASTE/HIDE devices are similar to the simple weather reporting substations found all over the country. These simple and relatively cheap systems provide basic data to more sophisticated analysis centers. This weather reporting hierarchy can be adapted to IDS containing HASTE/HIDE sensors. To illustrate this, Figure 3 is taken from a present day monitoring system of network attacks, showing how attacks will rise initially before being brought down to less paralyzing levels. Superimposed on this graph is a theoretical graphing of attacks on mobile devices, showing how such attacks would most likely occur on this front prior to it appearing in larger numbers against machines on the corporate network behind a firewall. The affect is an early warning system indicating the probability of oncoming attacks against the network and from which domain or segment.

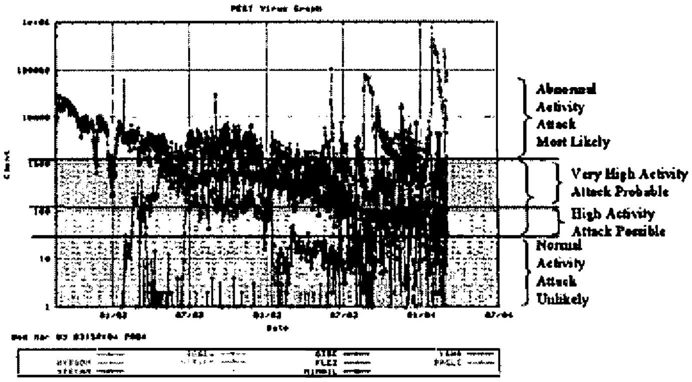

Figure 4 – Directed Attacks Thresholds. Graph background from [24]

Figure 4 compliments Figure 3 as it shows the log inverse of the same information, effectively illustrating thresholds of *directed* attacks against the network. The Y-axis represents the threat severity of a particular attack. Anything in the lowest band is considered to be "ground clutter" and not indicative of an attack. Anything between the middle and top bands indicates the attack is severe and widespread, requiring an immediate response to contain the attack. While one expects the curves to slowly diminish with time, a positive slope in the curve shows a secondary recurrence of the attack. This representation of attacks on and reported by mobile devices serves as a means to filter out noise and provide a level of intensity, indicating the severity of the attack and, consequently, the likelihood that it may occur against other segments of the network. This helps to mitigate or account for an anticipated number of false positives affiliated with anomaly-based IDS. Clearly these graphical instruments are enhanced by the inclusion of mobile host-based feedback. Administrators would be more enabled to capitalize from benefits drawn over such an integrated multi-layer and multi-tier defense strategy.

Although this type of monitoring is *reactive*, the goal is to identify an attempted break-in or attack before the attack is successful on a wider scale as part of a damage prevention and containment strategy. Das notes that as sophisticated attackers use more techniques to disguise their attacks, it is therefore necessary for researchers to improve their network-based systems to be able to better detect stealthy attacks or combine them with host-based methods [25]. Our approach is based on specifying security-relevant thresholds from reports generated from HIDE and HASTE as well as those presently in use for the network intrusion. Comparative performance results with other visual analysis approaches will be difficult because of the lack of standardize benchmarking and tools for such a system, however, we hope that our prototype implementation and benchmarking results will provide the necessary first step in this direction.

VI. FUTURE RESEARCH

Success of B-bid methodology ultimately relies on our ability to identify the correct thresholds of energy expended over time. Because the main variable is power consumption, we are confident that identifying essential information for battery exhaustion-related attacks will be more straightforward and foolproof than other forms of intrusion detection techniques, which have many other variables to consider.

As we move forward in this work, our primary goals in order are:

- to detect novel attacks against systems.
- to reduce the false negative and false positive rates as much as possible.
- to consume less power than we save in identifying attacks.
- to protect the program from attack itself.
- to maintain an architecture that can be easily integrated into more powerful IDS methods.
- to integrate feedback from mobile hosts into the overall security monitoring of the larger network.
- to integrate this architecture into wired hosts.

Intrusion Detection Systems themselves have become primary targets for attackers, especially those not wily enough to overcome the challenges related to power-related detection in trying to modify both energy and time to stay within a power consumption bound considered acceptable. Construction of a B-bid sensor infrastructure similar to that of US Weather Service stations will help responders identify and correlate a wide variety of input data in a timely fashion.

VII. CONCLUSION

An effective intrusion detection strategy implements several layers of defenses. Battery-based intrusion detection should be an integral part of this, serving as a viable approach to significantly increase the probability of detecting an attack and, by virtue of this, protecting the battery life and security of mobile devices as well as providing an early attack warning to larger segments of the network. Currently there are no host-centric forms of IDS for small mobile devices. Given that the percentage of detected and reported attacks against wired systems is believed to be less than 10% by Durst [26], it seems reasonable to suspect that the number of detected attacks on mobile systems *is* considerably less without host-based IDS.

VIII. REFERENCES

[1] T. Martin, M. Hsiao, D. Ha, J. Krishnaswami, "Denial-of-Service Attacks on Battery-powered Mobile Computers," *Second IEEE International Conf. on Pervasive Computing and Communications*, 14-17 Mar. 2004.

[2] B. Brown, "The Security Difference Between b and I," *IEEE Potentials*, pp. 23-27, Oct-Nov. 2003.

[3] L. Benini *et.al.*, "Battery-driven dynamic power management," *IEEE Design & Test of Computers*, vol. 18, pp. 53–60, Apr. 2001.

[4] L. Benini *et.al.*, "Extending lifetime of portable systems by battery scheduling," in *Proceedings. Design Automation & Test Europe Conference.*, pp. 197–201, Mar. 2001.

[5] C.F. Chiasserini and R. R. Rao, "Energy Efficient Battery Management," *IEEE J. on Selected Areas in Comms.*, vol. 19, pp. 1235–1245, Jul. 2001.

[6] Smart Battery System Implementers Forum (http://www.sbs-forum.org).

[7] Systems Management Bus Organization (http://www.smbus.org/ specs/smbdef.htm).

[8] K. Lahiri, A. Raghunathan, and S. Dey, "Communication architecture based power management for battery efficient system design", *Proc. ACM/IEEE DAC*, pp. 691--696, 2002.

[9] J. Cannady, J. R. Harrell, J.R. "A Comparative Analysis of Current Intrusion Detection Technologies". *Proc. of Technology in Information Security Conference (TISC)*, pp. 212-218, 1996.

[10] W. Schwartau, "Time-Based Security," Interpact Press, pp.1-192, 1999.

[11] "The Twenty Most Critical Internet Security Vulnerabilities", SANS Institute, (http://www.sans.org/top20/).

[12] Dallas Semiconductor, "App. Note 197 Sense Resistor Power Dissipation," pp. 1-2, (http://www.maxim-ic.com).

[13] Microsoft Corporation, Advanced Power Management The Next Generation, Version 1.0, (http://www.microsoft.com/ whdc/hwdev/archive/busbios/amp_12.mspx).

[14] C. Brake, "Power Management in Portable ARM Based Systems," Accelent Systems, pp. 1-5, 12 Nov. 2001.

[156] Hewlett Packard, hp iPAQ Pocket PC h5550, (http://www.hp.ca/products/static/ipaq/fa107a/features.php).

[16] Dallas Semiconductor, "App. Note 131 Lithium-Ion Cell Fuel Gauging with Dallas Semiconductors," pp. 1-10, (http://www. maxim-ic.com).

[17] D. Friel, The Importance of Full Smart Battery Data Specification Implementation, pp. 1-8, (www.powersmart.com).

[18] L. Benini *et al*, "A discrete-time battery model for high-level power estimation," *Proc. Design Automation & Test Europe Conf.*, pp. 35-39, Mar. 2000.

[19] S. Kumar, E. Spafford, "A Pattern Matching Model for Misuse Intrusion Detection", *Proceedings of the Seventeenth National Computer Security Conference*, pp. 1-11, 2000.

[20] S. Jha, O. Sheyner, J. Wing, "Two Formal Analyses of Attack Graphs", *Computer Security Foundations Workshop, Proceedings. 15th IEEE*, pp. 49-63, 24-26 Jun. 2002.

[21] R. Erbacher, D. Frincke, "Visualization in Detection of Intrusions and Misuse in Large Scale Networks," *Proceedings. of the International Conference on Information Visualization '2000*, London, UK, pp. 294-299.Jul. 2000.

[22] R. Erbacher, D. Frincke, "Visual Behavior Characterization for Intrusion and Misuse Detection", U. of Idaho, pp. 1-9, 2001. [23] P.A. Porras and R.A. Kemmerer, "Penetration State Transition Analysis A Rule-Based Intrusion Detection Approach," *Proc. of the Eighth Annual Computer Security Applications* Conference, Texas, pp. 220-229, Dec. 1992.

[24] R. Jarrell, Virginia Tech, Mar. 2004, (http://babylon5.cc.vt.edu/ ~jarrell/esewgraph/).

[25] K. Das, "Attack Development for Intrusion Detection Evaluation", Masters Thesis, M.I.T., Dept of Electrical Engineering and Computer Science, Boston, pp. 1-97, 2002.

[26] R. Durst, T. Champion, B. Witten, E. Miller, E. Luigi; "Testing and Evaluating Computer IDS," *ACM: Digital Library: Communications of the ACM*, Volume 42, No. 7, 1999.