Securing MANETs with BITSI: Danger Theory and Mission Continuity

Marco Carvalho[1], Richard Ford[2], William Allen[2] and Gerald Marin[2]

[1]Institute for Human and Machine Cognition, 40 S. Alcaniz St., Pensacola, FL, USA
[2]Florida Institute of Technology, 150 W. University Boulevard, Melbourne, FL, USA

ABSTRACT

MANET (Mobile Ad hoc Network) environments are becoming increasingly important as potential users recognize the benefits of being able to create a functional network using little or no fixed infrastructure. Unfortunately, the very properties that provide such flexibility also cause significant complications in terms of security. The collaborative nature of the system combined with its continual state of flux requires solutions that are highly dynamic, and that can adapt to massive changes in system resources, traffic patterns and network topology.

In this paper, we outline a new approach to MANET security called BITSI (the Biologically-Inspired Tactical Security Infrastructure). BITSI is based upon the concepts of Artificial Immune Systems and Danger Theory. After introducing the motivations for BITSI we provide a brief description of its underlying theories and proposed architecture. Two experiments conducted within our MANET simulator are described, and we demonstrate that BITSI can detect and respond to certain classes of Denial of Service attacks. Finally, we describe our future plans for BITSI, and how its approach can be combined with other, more traditional, security solutions.

Keywords: BITSI, MANET Security, Biologically-Inspired Security, Danger Theory, Artificial Immune Systems

1. INTRODUCTION

Mobile computing has revolutionized how stakeholders interact with information assets. Data is no longer confined to the desk or even the personal computer. Instead, convergence devices provide users with powerful platforms for accessing, leveraging and modifying any data from anywhere. However, with the boon promised by ubiquitous connectivity comes the challenge of servicing the bandwidth requirements of these nomadic devices, and securing flows of information that can extend well past the traditional boundaries of an organization. In order to overcome infrastructure-related issues and facility deployment in tactical environments, researchers have designed mobile ad hoc networks (MANETs) that consist of mobile nodes working together to provide for mesh connectivity creating a virtual dynamic network infrastructure for communication services. Such types of networks have broad applications, ranging from simple home-base or office deployments to tactical disaster relief and military operations.

Unfortunately, the dynamism of the MANET environment poses significant challenges for those tasked with securing such environments. In this paper, we provide a short overview of prior work in MANET security, and then describe our implementation of our own biologically-inspired MANET security solution. Finally, we present simulation results that demonstrate BITSI's ability to detect and respond to denial of service attacks even in the presence of false positives.

2. MANETS AND SECURITY

Computer security is hardly a solved problem even for a static environment such as a workstation protected by a firewall. Even when routes and endpoints are largely fixed, the dynamism of the network as a whole and the rapidly evolving threat makes static or threat-specific systems of limited value. In the MANET environment, however, the situation is considerably worse. RFC2051[1] defines a MANET as:

"A MANET consists of mobile platforms (e.g., a router with multiple hosts and wireless communications devices) – herein simply referred to as "nodes" – which are free to move about arbitrarily. The nodes may be located in or on airplanes, ships, trucks, cars, perhaps even on people or very small devices, and there may be

Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security 2008, edited by Belur V. Dasarathy, Proc. of SPIE Vol. 6973, 69730D, (2008) · 0277-786X/08/$18 · doi: 10.1117/12.777756

multiple hosts per router. A MANET is an autonomous system of mobile nodes. The system may operate in isolation, or may have gateways to and interface with a fixed network."

Using this definition as a basis, it is quickly apparent that ensuring the confidentiality, integrity and availability of information on a MANET is no small task. Aside from attacks susceptible to wired networks, MANET operators must also contend with:

1. Collaborative routing algorithms – in a MANET, nodes cooperate to route traffic and some nodes may be under an attacker's control.
2. Constrained bandwidth – broadcast media limitations restrict the available bandwidth and bandwidth must be shared by normal traffic, control traffic and any attack traffic.
3. Power constraints – battery life is often a concern for MANET designers; for example, nodes best positioned to route traffic may have to conserve power.
4. Node motility – because the observed traffic depends on the changing network topology, it is difficult to characterize normal versus attack traffic.
5. Dynamic network membership – nodes frequently enter and leave the network and may appear with no correlation to prior observations.
6. Lack of centralized organization – network monitoring, network management, network security must potentially be provided without recourse to more capable centralized nodes that can act on behalf of their neighbors.

These challenges have inspired a tremendous amount of research on MANET security – a full review would be a paper in itself! However, much of this material is devoted solely to detecting routing-level attacks, and is not necessarily extensible to more general threats. A solid overview of the problem space is provided by Sterne in Ref. 2, which outlines the specific threats MANETs usually face, as well as proposing a collaborative IDS scheme for the environment. Thus, instead of a comprehensive literature review, we will focus on those solutions aimed at detecting Denial of Service attacks within a network that initially begins in a secure state.

In tactical network environments, a realistic assumption is that as the network evolves, machines (i.e. nodes) accrue changes in the field. These changes may or may not introduce security vulnerabilities. The challenge with such an environment is that often the machines are isolated, and unable to query any central repository to determine if a detected change is benign or malicious. Furthermore, as rapidly-spreading malware can move extremely quickly (see, for example, the spread of SQL.Slammer[3]), systems need to be able to respond to threats autonomously, reconfiguring themselves to "stay ahead" of a rapidly shifting environment.

For the specific problem of Denial of Service, traceback of the originator of the spoofed traffic is difficult. Kim et al.[4] propose a solution based upon overlaying a small world network on top of the nodes, and looking for traffic correlations. This approach seems well-suited to dense networks, and should be usable within the BITSI framework if required. However, as the BITSI environment is relatively well-controlled, spoofing of network traffic can be detected more easily; thus, approaches currently explored within BITSI are more focused on non-spoofed DoS attacks.

The work described in this paper is different from prior work in its application of both danger theory and online distributed learning strategies for system defense. An even more fundamental difference in our proposed approach is the notion of mission survivability, as opposed to global defense. Analogous to complex biological systems, BITSI proposes a model where individual losses are acceptable as long as they can be detected, controlled and isolated for the preservation of the mission.

3. DANGER THEORY

It is our belief that a MANET security solution must be decentralized, adaptive and resilient. Because of these requirements, a biologically-inspired approach is attractive, as natural systems often display these qualities. In particular, computer scientists have often been tantalized by the concept of building an Artificial Immune System, which can dynamically detect and adapt to new threats.

Artificial Immune Systems (AIS) have held great promise in the security field. Early work by IBM[5] and the University of New Mexico[6] all focused on systems that could detect "non-self" elements and respond to them. Despite being demonstrated at the Virus Bulletin Conference in San Francisco in 1997,[7] commercial implementations of these concepts are generally weak at best.

Part of the challenge with the AIS model is that the human immune system seems to be far more complex than simple self/non-self discrimination. For example, many non-self entities are accepted by the body (e.g. parenterally-administered drugs) without provoking an immune response. Clearly, there is more at work than just discriminating between the body and "everything else".

In order to address this, Matzinger proposed that natural immune systems respond not to just self/non-self, but also detect damage.[8] When a cell dies via natural causes, the concept is that well-regulated biological pathways are followed. Conversely, when a cell undergoes stress or traumatic destruction, certain danger signals are generated. This is known as cellular necrosis, and is in contrast to pre-programmed cellular death via apoptosis. While this theory is somewhat controversial in biological systems,[9] the paradigm does turn out to be surprisingly helpful when constructing artificial immune systems.

Since this insight, there has been significant effort applied to exploring this paradigm in the context of an AIS. In Ref. 10, Aikelin et al. propose the use of Danger Theory as a missing component of traditional IDS/AIS systems.

The application of Danger Theory (DT) to attacks within MANETs is generally focused on the notion of damage to the system. Events that may lead, for instance, to performance degradation (i.e. damage to the task) are considered to be dangerous, regardless of their possibly malicious or benevolent nature. Note that in many cases, it is not clear if damage is occurring simply due to the relative position between nodes (for example, two nodes may share a poor link and therefore have lossy communication) or due to malicious activities. Nevertheless, we note that DT is an effective moderator of our immune system model only when damage is discovered does the system attempt to discern the cause. The following list outlines some common attack classes and our triggers within DT:

- Denial of Service attack (resource consumption): When a mission objective is not met, the system checks the node for resource constraints, which can include CPU load, memory utilization or network usage. Because it may be difficult to relate mission objectives to node resources, it is also possible to establish thresholds at which the system will sense "damage" based on per-request or per-client consumption.
- Routing Attacks: The system notes that packet loss is occurring. Note that this will commonly occur due to environmental conditions. However, when routing errors are suspected (and the connectivity damage is discovered) the system can begin the process of determining the likely cause of problems.
- Worms/Viruses: The system should be able to note the creation of new processes and/or files, plus new outbound requests. However, from a pure DT perspective, detection will only begin if the worm/virus consumes too many resources or triggers outbound traffic that is deemed to be suspicious, or detrimental to the performance of the system.

Of course, there are many classes of attack that would not trigger a purely-DT moderated system. For example, a user whose password had been compromised and then used maliciously would not be detected. Similarly, attacks where the damage is not immediately critical to the mission (such as data exfiltration) will not be detected using a system wholly based upon DT. As such, we argue that DT should be just one component in a larger system. This larger system is discussed below.

4. BITSI OVERVIEW

Given the security challenges of the MANET environment, our work has focused on applying theoretical concepts to real-world attacks. In particular, we have begun development of BITSI, which leverages different aspects of biological systems.

The underlying architecture of BITSI is quite straightforward. Within this environment, we envision a system where each node has a BITSI reference monitor on it. This agent resides in a local trusted component (not modifiable in the field) at each system and monitors the behavior of the node, as well as the traffic which is forwarded on the local network. From such a vantage point, BITSI collaboratively works to respond to different attacks.

In terms of attacks, our vision for BITSI is one of mission enablement. That is, BITSI accepts that some attacks will succeed on the network, but aims to mitigate their overall impact to ensure mission continuity. This approach is different from more traditional remediation attempts, whose goal is to stop all attacks.

Thus, we envision a system where some level of attack is tolerable, provided it does not compromise the mission objectives defined by QoS-policies specified a priori for mission-critical applications. Softer security responses move away from binary "go/no-go" decisions toward responses which represent more of a continuum, such as rate-limiting traffic or selectively blocking connections from a particular application. By dynamically identifying and monitoring critical operations and performance requirements for specific contexts and missions, BITSI can focus on securing the core operation of the system, as opposed to trying to address the possibly unbounded space of all possible attacks.

The challenge with such a "live and let live" approach is that it ignores the underlying sensitivity of computer data. Clearly, some information in a military environment has long term value and high criticality; other information may have little or no long term value, but is, at the short time scale, critical (an example of this might be a session key for a temporary encrypted connection). Given that this information could be extremely small in comparison to its importance (such as a 128-bit encryption key), it is very difficult to use biological techniques to prevent data exfiltration attacks, as there is no obvious biological analogy. However, this is not necessarily a fatal flaw in our approach; firstly, it seems unlikely that BITSI would be the only protective measure on a system; secondly, given the size of the problem space, a robust solution for even part of the space is of value. BITSI has been designed with this in mind, and is capable of being integrated with other content-management/IDS tools.

One interesting issue within the DT framework is that of classification errors specifically, where non-malicious traffic is classified as bad ("Type I" errors/false positives), and attack traffic is classified as good ("Type II" errors/false negatives). If we momentarily limit our discussion to damage, and use damage as a unique and reliable indicator of an attack, we could argue that an attack that produces no damage is not a successful attack, even if the attacker intended to cause harm. As such, while it would be beneficial to be able to reliably-detect failed attacks, one can coherently argue that this is not necessary for mission continuity.

Conversely, there is some probability P_{fp} that a legitimate interaction will be misclassified as damaging. Such an occurrence could result from a number of different scenarios. For example, a legitimate request could cause an exceptional load on the server during the normal course of operation. Such a load is not an indicator of an attack, though one may argue that if it should occur to the point of affecting critical services, some remediation is required regardless of intent. Similarly, imagine a request R0 that causes a server to unload its entire cache of pre-calculated values. When another client issues a request R1, the server may experience very high workloads as these values are recalculated. Thus, from the perspective of the server, the "attack" is contained within R1, not R0.

This type of scenario is very difficult to detect in real time. However, we believe that a promising avenue of exploration is a formal "cause and effect" analysis. Statistical causal inference from observational data has been effectively applied and demonstrated in numerous research areas and applications. The approach essentially consists on determining causal structures (in the form of a Markov equivalent graph of a causal network)[11] that includes the variables of interest (critical system metrics). Carefully chosen conditional independence tests between variables (often represented as time-series) and pruning strategies provide the basis of operation of most of the algorithms available for the task, which also include more specific algorithms for Markov Blanket discovery.

5. PRELIMINARY EXPERIMENTAL RESULTS

Carrying out experiments on MANET security is no small task for a number of different reasons. As with all security-related experiments, real-world data can be hard to obtain. Furthermore, there is a dearth of test networks on which solutions can be executed. As such, emulation and simulation become useful tools for the experimenter. In this paper, we have opted to leverage our MANET simulator Hephaestus. For detailed implementation issues, the reader is referred to Ref. 12; however, at the most abstract level, the simulator is a mixed-level discrete event simulator, capable of simulating large wireless networks at a messaging level. Although Hephaestus contains a robust implementation of HSLS[13] for routing, for the purposes of simplicity, we chose to assume that routing was perfect within the network that is, if a path exists between two nodes, this path is known and taken. While this is clearly not a realistic assumption, it is our sense that this simplifying assumption does not significantly degrade our results and provides for significant performance enhancements (we explore this issue in detail in an upcoming publication). Rigorous application of our methods under HSLS routing is future work, though simple experiments do indeed show results similar to those obtained here.

As the simulation consists of many random factors, we chose to fix our topology between runs, thereby lowering computational costs. The topology chosen for each test was generated randomly, uniformly distributing the nodes over a squared area and allowing their fixed communications range to set the initial topology. There were no guarantees for full connectivity; however packets dropped due to lack of routing path between client and server were discounted in performance calculations. Movement of the clients is based upon a random walk, with a constant speed of 1 meter/second. The size of the simulation grid is 1000 by 1000 meters and the number of simulated nodes in all tests is 25. At the beginning of the simulation, nodes are randomly distributed in the grid (uniform distribution) and connectivity is estimated based on each node's transmission power. All nodes are assumed to have identical radios and antennas, with fixed transmission power and connectivity radius of approximately 400 meters. While this approach is not likely to be representative of an actual tactical MANET, our sense is that it represents a reasonable backdrop for proving the validity of the techniques employed.

Simulation of different scenarios is important, as the performance of the system is likely to be different for different runs, based upon stochastic events (such as false positives/negatives) and topology changes or churn rates. However, given the preliminary nature of our experiments, we have limited our simulations to 25 nodes, with 1 server and a total of 15 clients. There is 1 "malicious" client randomly chosen at each run, which issues attack packets aimed at the server. The choice of target node and attacker was randomly chosen at the beginning of each run. In order to obtain statistical validity, each run was executed 50 times and averaged.

One of the challenges is trying to measure the performance of the system as a whole, as there are many possible metrics. For the purpose of the experiments outlined here, it should be possible to measure the quality of service by averaging over the number of legitimate clients except that the graph may not be fully connected during the simulation. Thus, even in the absence of attacks, traffic may be dropped by the system due to temporary lack of available routes. In order to offset this problem, we have separately accounted for all packet drops due to lack of available routes or blocking during the simulations. The number of legitimate packets that were not dropped due to routing constraints constitutes the baseline for maximum performance. At every simulation step, the quality of service provided by the server was calculated as the ratio between the number legitimate packets serviced at that time step and the total number of legitimate requests not dropped by lack of route to destination. This model is appropriate for the simple preliminary experiments outlined in this paper. However, when considering more complex experiments with background traffic and network congestion, it is not clear how system performance will be evaluated. Thus, this is left as an open and potentially difficult problem to be addressed as part of our future work.

In the following experiments, we limit the attack carried out to simple denial of service attacks. Clients make legitimate requests to a server which is randomly pre-selected for each simulation run; after a few simulation steps (50), a number of attacker nodes, which were originally sending legitimate requests, start making malicious requests. The reason for the fixed delay for attacks in our simulation tests is simply to separate routing stabilization effects (which are artifacts often observed at the beginning of simulations[14]) from the patterns of interest. During each time step, a server that receives an attack packet will serve no more requests until the next time-step. This service gap is detected by an agent on the machine (as damage, using a DT framework) and the attacking packet flagged as a potential cause.

In the first experiment, the node response is very simplistic. When an attack is detected the attacking node is blocked for the duration of the experiment. When more traffic from this node is encountered, the block is propagated to the neighbor that sent the traffic. Thus, as time progresses, blocks eventually surround the attacking node, and the attack is nullified.

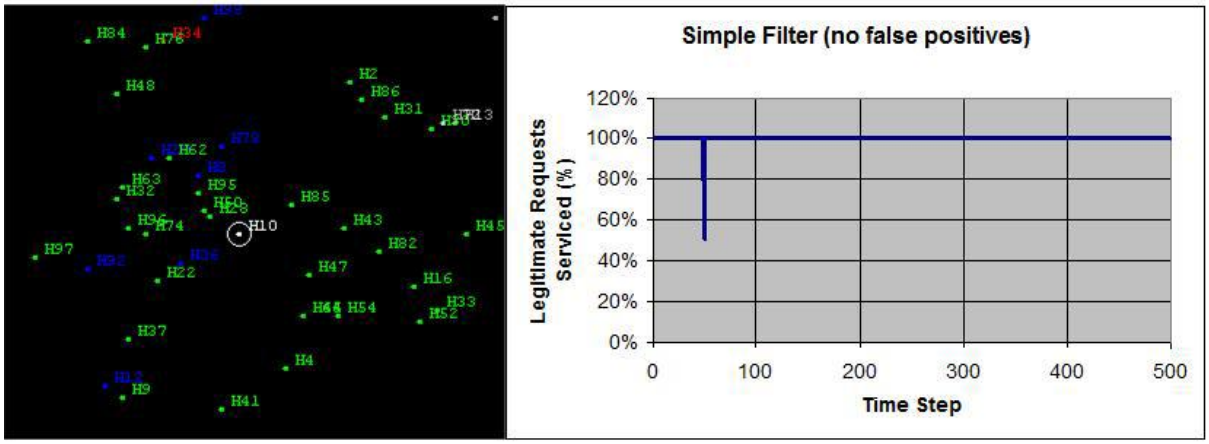

Figure 1. a) Sample Simulation showing the target node (H10) and the attacker. All nodes in blue have received filter propagation from H10 and are currently acting as active blockers (pushback). b) Average percentage of requests serviced.

Figure 1 shows the results of this experiment averaged over 50 runs. At time 50, one attacker starts sending attack packets to the server. When the first packet is received, damage is detected as the drop in the number of legitimate requests serviced (quality of service). The attacking node is permanently blocked and the block is propagated to nodes in the data path to pushback the filter, eventually isolating the attacker. Because the block is permanent after the first attack, once the filter is put in place the performance is never again degraded, remaining constant at the maximum level (100%) for all subsequent time steps. As it can be seen, the results show that BITSI is capable of responding well to the attack, and the remote firewalling of traffic prevents the attack, as well as lowers the overall network load. While this is a nice proof of concept, the approach is too simplistic.

The first experiment's primary drawback is that it is very brittle; making the assumption that traffic is never misclassified is not realistic. Instead, any system we design must take into account the possibility of both Type I and Type II classification errors. Furthermore, in Experiment 1, the system has no real learning component, and so never "learns" either locally or systemically from past experiences.

One of the potential benefits of a biologically-inspired approach is its resilience to false positives. Given the precept that no damage detection system is likely to be perfect (creating both Type I and II errors), one important characteristic of the system is how it handles errors of either type. Thus, we created an experiment where legitimate traffic have a probability P_{fp} of being classed as malicious, and where malicious traffic had a probability P_{fn} of being classed as legitimate.

The experiment was again run 50 times, and the average Quality of Service measured for legitimate clients, still using the simple naïve blocking filter. As illustrated in Figure 2, even if a relatively low percentage of false positives is added to the simulation (in this case, 5% for both P_{fp} and P_{fn}) , the system fails to adapt, eventually blocking all nodes in the networks and completely degrading its capacity to service requests.

In order to address the problem a more adaptive and robust node behavior was created. Instead of issuing a permanent block when malicious traffic was encountered, each node was given a memory that stored information about previous attacks from other nodes. This memory was used to determine which potential attacker nodes should be blocked and for how long. Note that this learning is local, in that node A's conclusions about an attacker does not modify the opinion of other nodes in the system.

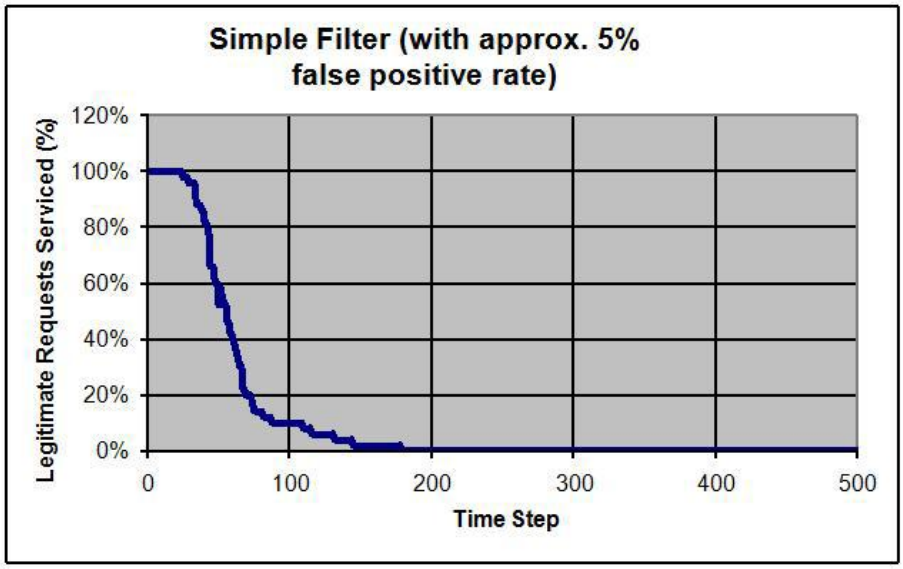

Figure 2. Simple blocking filter performance under 5% false positive classification rate.

Each node maintains a history for each other node it encounters over the network. The size of this history is fixed at 10 records for the purposes of this experiment, though the optimal history size is likely to depend on Type I and Type II error rate, traffic patterns and network size and density. By way of illustration, if node A had interacted with 5 other nodes, it would store 5 circular historical logs, each containing 10 events.

Events stored in the log are used to measure the ratio of packets classified as attacking packets and packets classified as legitimate for each node. Thus, when damage is detected by a node, the intent is to log the event (and potentially, the severity/type of the damage). This is most clearly described by an example: when a node first detects the presence of another node (by receiving data packets or route advertisements from it) a circular queue is created and stored in association with the node. The size of the queue will be influenced by the scale of the network and traffic volume. Currently, the system uses a fixed arbitrary size of 10, capable of holding a very short term history of traffic per node.

The queue for each node will hold two types of entries, a "zero" value for each packet received from that node that is classified as a legitimate packet (i.e. not associated to damage), and a "one" value for each attacking (or damaging) packet received from that node. At first, the queue is initialized with "zero" values, which means that nodes are a priori assumed to be legitimate.

If damage to one of the system's critical processes is then perceived, a brief causal analysis process takes place to try identifying the cause of the damage. In this case, if a packet received from another node is believed to be the cause, the packet is classified as an attacking packet*. At this point, the node's history will log an attack entry. Conversely, packets received from that (or other nodes) that have not been classified as damaging will result in a "zero" entry in the queue.

At any given time, the ratio between the number of attack and legitimate entries for a given node can be used to determine if the node is an attacker and if it should be blocked (block-decision function). Historical blocking data can be used to estimate the duration of each block (block-duration function). Nodes that are frequently classified as attackers will be blocker for longer periods than others. The challenge, however, is to identify a block-decision function that provides both a quick reaction time to potential attackers (early blocking) and, at the same time, tolerance for false positives and good adaptation properties to support changes in the system.

*For the purpose of this simulation, damage is directly perceived as a random classification of the packet, which may have an error rate of P_{fp} and P_{fn}. The classification errors, illustrated here as a fixed ratio, are directly mapped in actual deployment to mis-identification of causes or mis-interpretation of damage.

For that purpose, BITSI uses an exponential function that depends on the difference between the number of good and bad packets received from each other node (Equation 1). Because the size of the queue is fixed and the decision (in this example) is binary, the function can be simplified to track only the number of "good" packets remaining in the queue. However, the same strategy can be easily extended to account for multiple levels of damage and more complex systems. In such case, the function will be similar to the Gibbs (or Boltzmann) distribution, commonly used in SoftMax[15] online learning strategies to evaluate multiple competing strategies for a given problem.

$$Index_{damage} = \frac{\epsilon^{\eta \cdot \sum(X_{bad})}}{\epsilon^{\eta \cdot \sum(X_{bad}+X_{good})}} = \frac{1}{\epsilon^{\eta \cdot \sum(X_{good})}} \quad (1)$$

Equation 1 provides a damage index that basically describes a non-linear relationship between "bad" and "good" events in the short-term past behavior of one specific node. Given a threshold (thresh), nodes whose damage index value is higher than the threshold, are classified as causing damage and consequently as attackers. Another parameter of importance to the system is η, which essentially provides a scaling factor to the system. Following the conventional online learning notation, η is the inverse of the temperature (τ) of the system (i.e. $\eta = 1/\tau$).

The exponential nature of the function is of great importance to enable fast reactions to possible attacks while supporting system adaptability. The temperature (τ) defines how reactive the system is, accentuating the differences between (perceived) good and bad packets. At the end of each simulated time step the target node evaluates all nodes that it received a packet from in the previous time step. Equation 1 is used to determine, for each queue, if the sender should be blocked or not based on that information. If a decision for blocking is made, a counter maintaining the number of blocking decisions for each node is incremented and then used to calculate the duration of the block. Nodes that are often classified as attackers (high blocking count) will be blocked for longer period of times. Once again, in this case, the duration of the blocking follows an exponential function for the number of blocks.

This simple heuristic rule to determine the duration of a block ensures that repeat offenders are penalized more heavily than low-count ones, effectively separating noise due to misclassification from actual attacks. Both the blocking-decision function and the block-duration heuristic have been implemented in BITSI and tested with the Hephaestus simulator and other tools. Figure 3 shows the simulation results for 500 time steps of the test scenario.

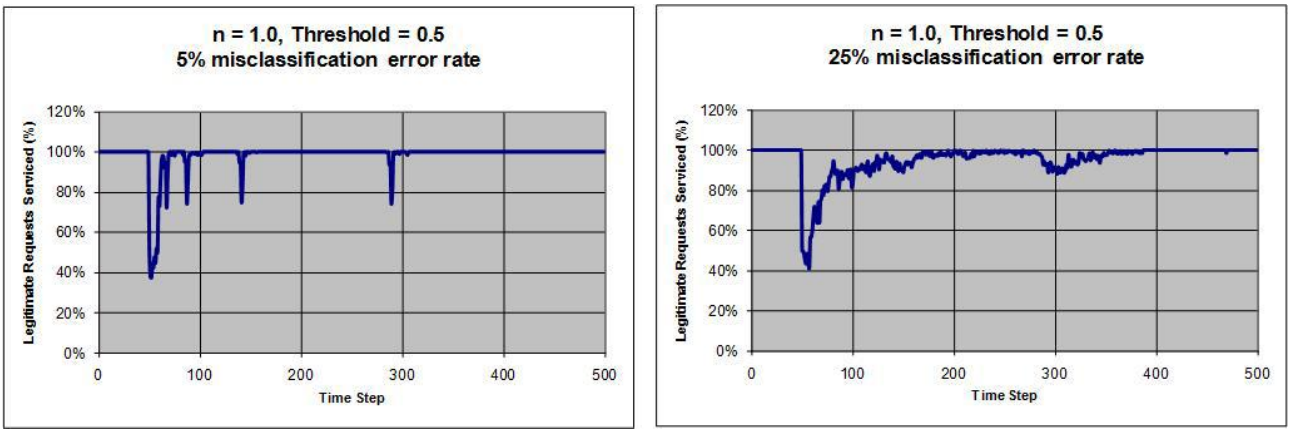

Figure 3. Average Quality of Service (50 runs) as a function of time. The misclassification rates are 5% (left – a) and 25% (right – b) for both Types I and II error.

For these tests, the parameters of the equations (η and the threshold) were fixed. In Figure 3a, the probability of misclassifying a packet is 5%. This probability refers to both type I and type II errors. In the first case

(Figure 3a), the mapping between damage and the true 'attack packet' is relatively accurate (95%), which enables BITSI to quickly identify and react to the attack that is initiated at time step 50. When a potential attacker is first identified, the duration of the blocking requested by the server grows exponentially (block-duration function), which can be seen in Figure 3a by the increasingly spaced drops in Quality of Service (i.e. Legitimate Requests Serviced). These drops in QoS are relatively frequent at the beginning (between time steps 50 and 150) due to the expiration of the first short-lived blocking filter. As the attack persists, the system's confidence in its decision about the block increases and the duration grows exponentially, leading to the eventual isolation of the attacker for very long periods of time.

In Figure 3b, the same case is shown for higher probability of misclassification (25%). In this case, for the fixed function parameters chosen for this example ($\eta = 1.0$ and threshold = 0.5) the system wrongly reacts to misclassified attacks, which affects its ability to identify and isolate the true attacker. However, even under such conditions, as time progresses, BITSI builds enough statistical evidence about the attacker to properly define and propagate the correct blocks.

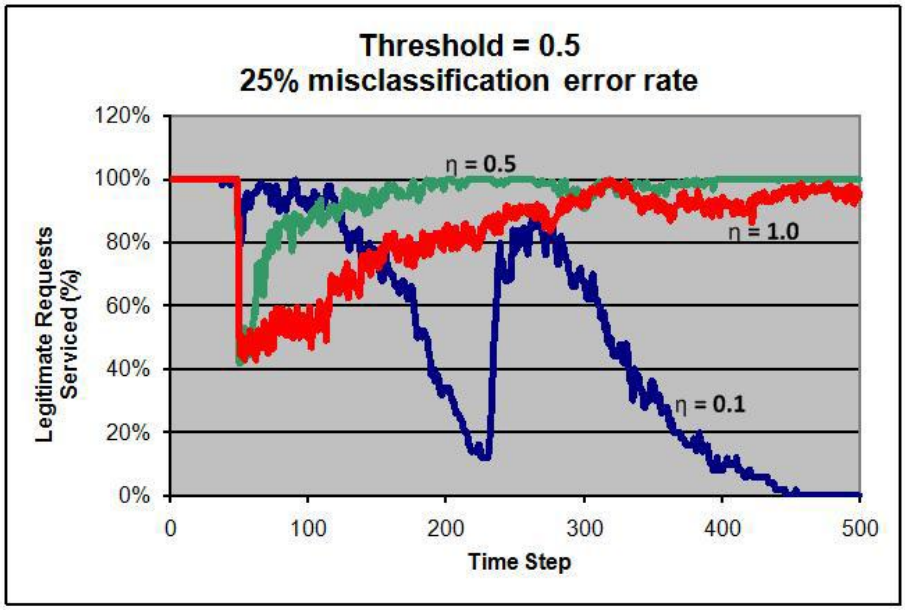

Figure 4. Effects of temperature ($1/\eta$ parameter) on the reactiveness of the system.

As illustrated in this simple example, the attributes of the blocking-decision function (Equation 1) are very important to balance the system in terms of reactiveness and stability. Figure 4 shows the previous example (25% misclassification error rate) for different values of temperature ($1/\eta$). The higher the temperature, the lower the value of η, and more reactive the system. In uncertain environments with high probability for misclassifications, very reactive systems tend to perform poorly (as indicated in Figure 4), which is essentially the opposite behavior noted in systems with low levels of false-positives.

Consequently, the appropriate choice of temperature is not independent of other system parameters. As illustrated in the smoothed functions shown in Figures 5 and 6, higher η values (low temperatures) may also be detrimental for higher values of threshold in environments with higher misclassification error rates.

As part of the BITSI effort, these parameters will be adjustable and adaptive, at run time, to accommodate changes in mission objectives, system complexity (which may lead to higher error rates) and operation tempo. This work is an initial proof of concept, designed to show how sensitive a damage classification technique, such as one proposed in BITSI, may be to Type I and Type II errors. Much further work remains. The two most important changes we envision are building a learning environment which retains memory of past decisions and their outcome – both locally and globally, and the expansion of the work to handle more types of attack.

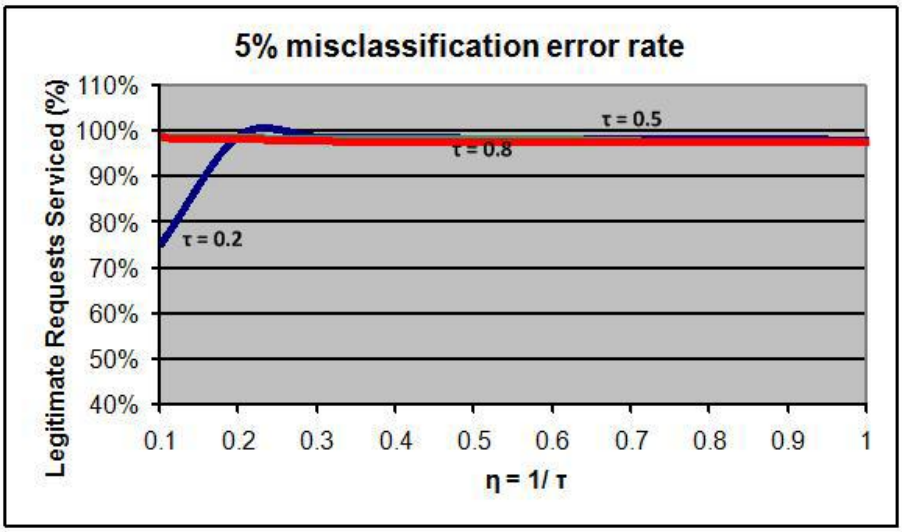

Figure 5. Smoothed curves showing the relationship between temperature and classification threshold for 5% misclassification error rates

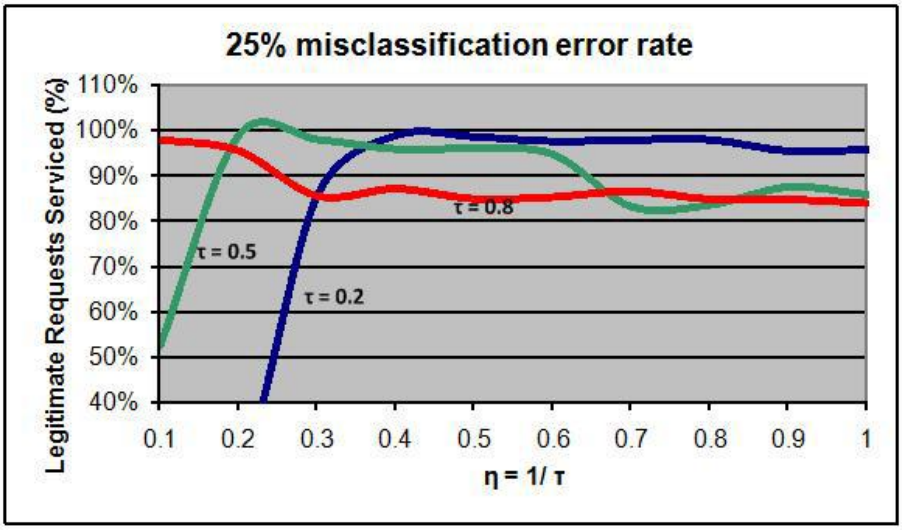

Figure 6. Smoothed curves showing the relationship between temperature and classification threshold for 25% misclassification error rates

Our initial foray into the field shows that the metaphor is particularly suitable for responding to both Denial of Service by flooding or by directly damaging the system. However, the Danger Theory metaphor can be equally well applied to any resource consumption problem in the MANET, and perhaps other security issues.

6. CONCLUSIONS

In this paper we introduce a biologically-inspired security infrastructure for military (or civilian) tactical network environments. Departing from the conventional notion of perimeter and system defense against attackers, BITSI proposes an organic defense framework where all computational devices in the battlefield work together to protect the mission and its critical processes.

We presented and discussed preliminary simulation results showing the utility of the proposed infrastructure for the target environment. Although we have only demonstrated BITSI's functionality with respect to a class

of previously-unseen Denial of Service attacks, our intuition is that the approach is more-widely applicable. Our future work aims to extend the technique to routing attacks and malicious dropping of packets.

ACKNOWLEDGMENTS

This work is part of a multi-institutional effort, under sponsorship of the Army Research Laboratory via Cooperative Agreement No. W911NF-07-2-0022, CFDA No. 12.630.

REFERENCES

1. S. Corson and J. Macker, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," Tech. Rep. 2501, IEEE, January 1999.
2. D. Sterne, P. Balasubramanyam, D. Carman, B. Wilson, R. Talpade, C. Ko, R. Balapari, C.-Y. Tseng, T. Bowen, K. Levitt, and J. Rowe, "A General Cooperative Intrusion Detection Architecture for MANETs," in *Proceedings of the Third IEEE International Workshop on Information Assurance (IWIA'05)*, pp. 57–70, EEE Computer Society, 2005.
3. D. Moore, V. Paxson, S. Savage, S. Shannon, C. Staniford, and N. Weaver, "The Spread of the Sapphire/Slammer Worm." CAIDA – Available online at http://www.caida.org/outreach/papers/2003/sapphire/sapphire.html, 2003.
4. Y. Kim, V. Sankhla, and A. Helmy, "Efficient Traceback of DoS Attacks using Small Worlds in MANETs," in *Vehicular Technology Conference*, **6**, pp. 3979–3983, September 2004.
5. S. R. White, M. Swimmer, E. J. Pring, W. C. Arnold, D. M. Chess, and J. F. Morar, "Anatomy of a Commercial-Grade Immune System," in *International Virus Bulletin Conference*, pp. 28–30, September 1999.
6. S. Forrest and T. Longstaff, "A Sense of Self for Unix Processes," in *Symposium on Security and Privacy*, pp. 120–128, IEEE Computer Society, IEEE Computer Society Press, 1996.
7. J. O. Kephart, G. B. Sorkin, M. Swimmer, and S. R. White, "Blueprint for a Computer Immune System," in *International Virus Bulletin Conference*, October 1997.
8. P. Matzinger, "Tolerance, Danger, and the Extended Family," *Annual Review of Immunology* **12**, pp. 991–1045, 1994.
9. P. Matzinger, "Essay 1: The Danger Model in Its Historical Context," *Scandinavian Journal of Immunology* **54**(1-2), pp. 4–9, 2001.
10. U. Aickelin, P. Bentley, S. Cayzer, J. Kim, and J. McLeod, "Danger Theory: The Link Between AIS and IDS?," in *2nd International Conference in Artificial Immune Systems (ICARIS 2003)*, 2003.
11. P. Spirtes, C. Glymour, and R. Scheines, *Causation, Prediction, and Search*, MIT Press, Cambridge, 2 ed., 2000.
12. A. Ondi, *Malicious Code Related Experiments with an Extensible Network Simulator*. PhD thesis, Florida Institute of Technology, 2007.
13. BBN, "Hazy Sighted Link State (HSLS) Routing: A Scalable Link State Algorithm," BBN Technical Memorandum 1301, BBN, August 2001.
14. L. F. Perrone, Y. Yaun, and D. Nico, "Modeling and Simulation Best Practices for Wireless Ad hoc Networks," in *Proceedings of the 2003 Winter Simulation Conference*, 2005.
15. R. Sutton and A. Barto, *Reinforcement Learning*, MIT Press, 1998.