A THREE-TIER DAMAGE-DRIVEN SECURITY INFRASTRUCTURE FOR MISSION CONTINUITY

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

[1]Institute for Human and Machine Cognition (IHMC)
40 South Alcaniz St., Pensacola, FL, 32502

[2]Florida Institute of Technology
Dept. of Computer Sciences
150 University Blvd., Melbourne, FL.

mcarvalho@ihmc.us, {rford,wallen,gmarin}@fit.edu

ABSTRACT

Mobile Ad Hoc Networks are one of the core enabling technologies for tactical environments such as those envisioned in future military operations and disaster relief scenarios. While a significant amount of research has focused on technology interoperability and protocol design, there has been less work addressing the security and information assurance requirements for such environments. Properly identifying and satisfying these requirements will be mandatory for successful deployment in a hostile cyberspace.

This paper describes BITSI (Biologically-Inspired Tactical Security Infrastructure), our approach to provide MANET security with focus on mission continuity. Our measure of success is the completion of the mission (or missions) within pre-defined QoS requirements, as opposed to the protection of a single machine. After a brief discussion of the theoretical principals involved in BITSI, we describe the proposed security mechanism and system architecture.

INTRODUCTION

Mobile Ad Hoc Networks (MANETs) are widely accepted as a key technology enabler for tactical operational environments such as those often found in Network Centric Warfare and disaster relief operations. While still a relatively young technology, MANETs have been the focus of intensive research efforts for the last few years and continue to pose both great promise and new challenges to the research community. Security is arguably one of the most difficult of these challenges, and the focus of a growing number of research efforts in the domain.

Addressing MANET security requires not only coping with typical network and host security issues (which is hardly a solved problem), but also a number of new challenges related to the unusual characteristics of the environment. Such characteristics include the shared nature of the communications medium, highly dynamic topologies and the lack of fixed infrastructure components and services. To date, most security technologies and methods for MANETs have been ported from the wired environment. While a few of these strategies have shown some level of effectiveness for specific scenarios, the growing consensus is that new approaches and methods are necessary for the domain.

In this paper we describe a three-tier security mechanism utilized in BITSI, the Biologically-Inspired Tactical Security Infrastructure, first introduced in [1] and [2]. BITSI combines the concepts of Artificial Immune Systems and Danger Theory with social reputation to create a damage-driven security infrastructure whose sole focus is mission continuity.

BITSI leverages the fact that tactical MANETs (especially those used within military domains and disaster relief missions) are created for a well-defined purpose. In our case, we define the mission objective(s) as the primary core purpose of the system (much like survival is the core, primary objective of biological systems). From such an analogy, BITSI introduces a new approach for system security focused on survivability and mission continuity.

After a brief overview of our target scenario and motivations, we will introduce and discuss the multi-tier architecture proposed for BITSI, as well as some preliminary results of the proposed approach.

This work was sponsored by the U.S. Army Research Laboratory - Cooperative Agreement W911NF-080200023.

THE ARMY'S TACTICAL MANET ENVIRONMENT

Our scenario of interest is the tactical military environment that will constitute the edge networks envisioned as part of the United States Army's Future Combat Systems (FCS) program [3]. The relevant key characteristics of these types of networks are a) non-reliance on a fixed centralized support infrastructure, and b) a shared communications medium, which not only raises a number of security concerns, but also introduces complexities associated with resource contention and interference. Another very important (and arguably, unfortunate) "feature" of FCS is the use of multi-purpose systems and machines to host mission-critical applications, as well as less relevant, "convenience" applications. This futuristic view of a single portable device that handles all electronic needs of the soldier in the field will unavoidably have to accommodate the notion of general-purpose applications such as email clients and web-browsers co-existing in a system that may also contain mission-critical components. Thus, it becomes difficult to adequately separate important from desired functions whether under attack or simply resource-constrained.

In the military domain, one approach to this problem is to create uniform system images that are thoroughly tested for a specific operation. In preparation for a mission, these images are loaded and are deemed immutable during the mission. Thus, the underlying assumptions in this case are a) that systems are, *a priori*, safe and trusted, and b) that systems are not modified during the mission. Based on these axioms, one is led to the conclusion that any subsequent communication and interaction between authenticated systems is safe.

In practice, however, not only are the standard images created for different systems not guaranteed to be safe, but most importantly systems *do* change during the course of a mission. Changes in priority, targets of opportunity and broader changes in the operational theater and battle conditions are just some examples of cases that may require a commander (or even the soldier in the field, at the most local level) to modify the configuration and actual applications running on their systems. More often than not, such changes are innocuous to other critical applications (i.e. the application of a local update on a web-client or a media plug-in), but just as often, they may introduce vulnerabilities that may ultimately compromise the mission as whole.

Under these premises, the US Army Research Laboratory is investigating novel security strategies that will provide the flexibility necessary to support quick changes and reconfigurations in the field, while maintaining the necessary operational requirements to ensure mission continuity and completion. The assumptions posited for this research effort are:

a) Systems start from a known base image, which we will define as the base state of the system. This maintains the original assumption that a "standard" image is created and tested for mission devices.

b) Systems can be reconfigured, modified and extended in the field (thus departing from their original base state). This assumption relaxes the (unrealistic) constraint that standard system images are static throughout the mission.

c) A trusted security component is available at each node. This read-only trusted component is capable of monitoring all communications and processes running on the system. It can also monitor and modify system calls and network traffic; thus it is capable of preventing traffic spoofing (a simplifying assumption that will be relaxed in subsequent phases of this research project).

d) The main objective of the macroscopic system is that of mission survivability, not necessarily system protection. It is assumed that other conventional security tools will remain in place (or be further developed) to provide other security properties.

In this paper we will introduce and discuss the design of a three-tier distributed security mechanism that will provide a novel, self-regulating and self-healing mechanism to protect mission continuity while supporting in-field system changes and reconfigurations.

BITSI: A BIOLOGICALLY-INSPIRED TACTICAL SECURITY INFRASTRUCTURE

Artificial Immune Systems have been previously proposed for computer network security. Such immune systems are typically inspired by the ability of white cells in the human body to identify antigens and quickly respond to a potential attack (important early works in the field are [4] and [5]). While certainly an interesting idea, the concept has been primarily applied in the context of self/non-self (SNS) discrimination. That is, as a series of techniques and algorithms for identifying anomalies in some system (i.e. non-self) and, under the assumption that they constitute a potential threat, isolating and responding to the anomaly. SNS discrimination is plagued by very high level of false positives, and high computational

requirements due to the large scale and complexity of most practical systems.

More recently, biologists [6] have proposed a new model for the triggers that would lead to immune response in biological systems. The notion of Danger Theory (DT) was then introduced to associate immune response with the concept of damage to a biological system. These concepts were quickly adopted by the AIS-community [7] as a grounding, or trigger, component for artificial immune systems. The approach directly addressed the issues of high false positive rates and complexity because only events that correlate with perceived damage are considered for anomaly-detection investigation. However, two major questions remain to be answered: (1) What constitutes damage in a computer system? (2) How can such damage be detected? BITSI associates these concepts with the mission-critical nature of military tactical networks. Damage, in that context, can be identified as any event that deviates from the mission objectives, possibly degrading its performance or compromising its completion. From that perspective, BITSI proposes a security infrastructure that monitors local system tasks that are critical to the broader "mission objectives" to prioritize its anomaly detection and immune response tasks.

At the local (host) level, the concept of a mission is simplified by a set of well-defined QoS requirements for critical applications. The assumption is that if critical applications are performing within QoS specifications the overall mission objectives are likely (although not guaranteed) to be met. Furthermore, in addition to treating obligations, our model identifies actions that are disallowed by policy. This complementary policy model should be capable of detecting several additional classes of attack plus associated damage.

In BITSI, nodes can be changed and reconfigured in the field. All system changes and the short-term history of communications patterns and application calls are tracked. If damage to a critical application is detected (for example, a violation of its QoS requirements), BITSI will, at run time, search for causally correlated events in the past history of changes and events. Continuously monitoring the feedback from critical systems, BITSI acts upon the correlated anomalies (e.g. terminating processes, blocking connection to a specific port, or from a specific node, etc.). The process allows BITSI to identify the offending event and block it accordingly through a process of trial and error.

As illustrated in Figure 1, the trusted kernel (or BITSI kernel) is composed by a set of components that monitor the local environment to identify and respond to potential damage. The Mission parameters constitute the QoS requirement for critical applications. They are, at the higher level, represented in a semantically rich language for disambiguation and distribution. For efficiency, once disambiguated and distributed, only a compact representation of the relevant policies is converted as Mission Parameters. In BITSI, the definition of Mission Parameter requirements in the form of QoS for critical applications is provided by the KAoS policy services [8].

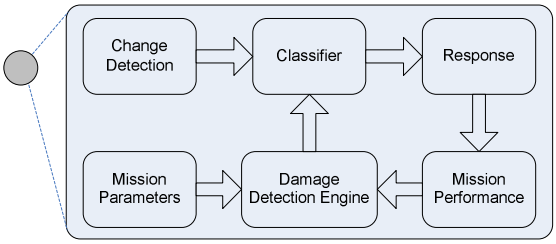

Figure 1. Damage detection and immune response feedback loop

The damage detection component (or Damage Detection Engine – DDE) is responsible for monitoring the performance of mission-critical applications (i.e. Mission Performance box) to identify deviations that could be flagged as "damage". Such damage could, for instance, be defined as exceeding the maximum allowed request/response delays of a critical application.

Once damage is detected, a classifier in the system (Classifier box) will compare the sequence of recent detection events with application calls and network activity. The analysis will allow the classifier to infer possible causal (or at least correlated) events potentially associated with the damage. The inferred cause will be used as basis for the response of the system.

Conceptually, this information can be used to learn new policies that will speed up the immune reaction in subsequent, similar events. Furthermore, learned patterns can be shared with peer nodes to enable some level of collective knowledge that may allow nodes to be immunized even before they are exposed to a previously unknown threat.

The core components of the BITSI kernel that enable this capability are illustrated in Figure 2. The kernel is installed at each node, and is the component responsible for the monitoring and control tasks required for the security infrastructure.

In BITSI, we propose a three-tier defense system that supports a local detection and immune response mechanism at the local (lower) level, the sharing of information amongst peer nodes to improve local learning

at the (second) group level, and the collaborative action of multiple nodes against a common threat (i.e. system level response). We now outline these levels, and the benefits they provide.

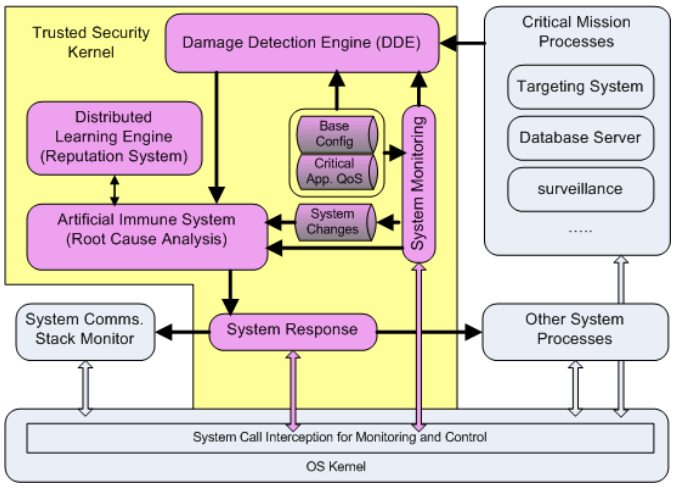

Figure 2. A simplified view of the BITSI kernel and core components

A THREE-TIER SECURITY ARCHITECTURE

As a distributed infrastructure, BITSI provides a three-tier defense system that spans from a local detection/response mechanism to distributed group learning, and finally to a collaborative, coordinated immune response to an attacker.

FIRST LEVEL: THE LOCAL DEFENSE SYSTEM

The basic operating mode of BITSI is at the local level. There are two types of detection/response mechanism that occur. One is the innate response of the system, that is, the response to known types of attacks and system actions. The innate response of the system requires no learning and is not triggered by direct damage to critical application. The detection and response mechanism at this level are very similar to conventional signature-based IDS mechanisms (and are entirely synergistic with such approaches).

As illustrated in Figure 3, the attacker (black node on the left) is launching a known attack to the victim (gray node on the right). The known signature of the attack is immediately detected by the BITSI kernel at the gray node, and a corrective action is taken. The response is not re-evaluated based on system feedback. A policy encoding the known attack and the appropriate response is simply activated for the specific event.

The second type of detection/response that happens at the local level is the adaptive response mechanism. In contrast with the innate response mechanisms, the adaptive component targets unknown attacks. It focuses on critical application performance degradation or damage to infer that an attack (or at least an undesirable operational condition) is taking place.

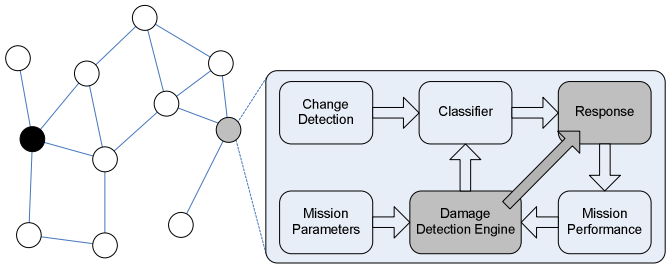

Figure 3. Local innate response to known attacks

Armed with information about the attack and with a recent history of events generated by application and network activities, the local kernel attempts to infer a potential cause for the damage and take corrective actions. Corrective actions may or may not affect the measured performance of the critical applications and, through the feedback loop illustrated in Figure 4, will be either penalized or reinforced accordingly.

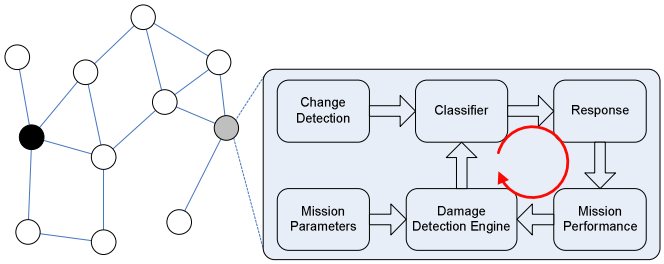

Figure 4. Feedback loop for local detection and response mechanism

The classifier also takes into account information stored in the change detection component. The component stores a history of all changes in binary or system configuration that deviate from the base, standard image. Loosely speaking, correlation (or causal correlation) between an event and damage are likely indicators of an attack source, while temporal correlations between damage notification events and system changes are good indicators of a vulnerability that may have been introduced and is now being exploited. These changes are related not to application performance in relation to QoS requirements, but to changes in system binaries or configurations.

Inferences about the potential cause of perceived damage also takes this information into account, as well as the history of binary and configuration changes. Systems that have been modified from their base-state are, *a priori*, a potential source of vulnerabilities and a possible source of

attacks. Information about deviations from the base state is also very important in order to properly allocate blame and identify nodes that have been compromised before they can launch widespread attacks. For instance, a node that has strongly-correlated damage to one of its web-services with a CRC change to one of its shared libraries may choose to reverse the change, and tag this modification as a potential vulnerability. This information may be used later by the innate local system to proactively prevent any subsequent changes to the library.

Consider, for instance, the example where damage being reported by a data translation service (a critical application in this example) is highly correlated with requests received for that service from a peer node *N*. In this case the kernel may initially identify *N* as a possible source of damage[2]. A naïve response strategy would consider blocking further requests from node *N* and monitoring if the performance of the application for other clients returns to the required levels. If the action is ineffective, a second strategy can be tried – for instance addressing the next highest correlated event.

CAUSAL ANALYSIS

In the previous example, the approach chosen for identifying the potential cause of damage was essentially based on a search strategy with system feedback that could have been implemented through a reinforcement learning mechanism. The initial state (or assumption) made in that process was based on the strongest simple correlation between system variables and the reported damage.

A better strategy for selecting potential candidates for the search process is to identify (whenever possible) causally correlated variables. Under certain assumptions about the data and the system, conditional correlation tests between variables may be used as indicators of a possible causal structure for the system. That information can be extremely valuable in selecting the variables that are potential causes for the damage.

There have been several proposed algorithms for the discovery of causal structures from observational data. Loosely speaking they can be classified as constraint-based methods and score-based methods.

Constraint-based methods generally start from a fully connected Bayesian network, using a number of statistical tests (conditional correlation) to determine if a given edge should be removed or remain in the graph. The numerous, potentially multi-dimensional conditional correlation tests used in methods of this kind generally require large amounts of data, and other statistical conditions to be satisfied such as causal sufficiency and faithfulness.

Score-based methods start from a possible Bayesian network representing the underlying causal structure of the system and use the structure to calculate a score (i.e. a Bayesian or a Minimum Description Length measures) to evaluate the candidate structure. Networks that score a higher metric value are preferred. Score-based methods generally perform better with less data but they may suffer from a local minima problem.

A SECOND LEVEL OF DEFENSE THROUGH DISTRIBUTED GROUP LEARNING

The second level of defense proposed for BITSI relies upon allowing nodes to share collected information and local inferences with peer nodes (Figure 5).

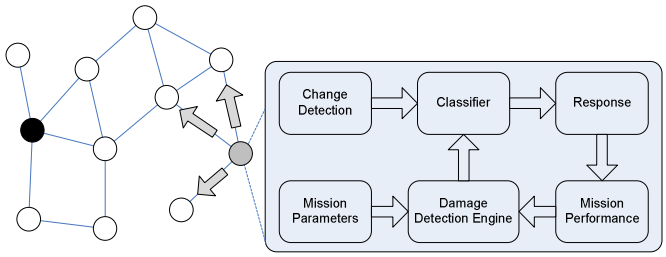

Figure 5. Reputation enables distributed group learning

TRUST IN A NON-TRUSTED ENVIRONMENT

The notion of sharing any kind of state information in an untrustworthy environment immediately raises the question of whether to trust the source of the information received or, for that matter, the information itself. In our target environment, a distributed reputation mechanism is necessary to assign different levels of trust to different nodes, and consequently to the information provided by these nodes.

Information provided by other nodes plays an important role in effective group learning. It allows knowledge created by one node to be shared proactively with other nodes before they are subject to similar attacks, reducing the learning-curve or maybe even completely immunizing the node from similar attacks before it is ever exposed to them. In BITSI, the reputation of other nodes in the network (and the information provided by them) is built

[2] It is important to note that BITSI does not need to consider the motive behind damage – the system does not need to determine if damage is due to an unfortunate confluence of circumstances or a deliberate attack in order to provide mission continuity.

based on previous interactions, as well as notion of context and node similarity.

CONTEXT DEPENDENT TRUST

One of the strategies leverages some of the previous research in context dependent trust to create a model that supports different levels of inter-node trust depending on the task at hand. For instance, a node that has been identified as a potential attacker of a specific web-service is likely to be blocked from accessing that service but may still be acceptable for data routing.

This approach fits well with the concepts proposed in BITSI where damage is the focus of attention. From that perspective, an "attacker" may not have any malicious intent but should still be blocked from having access to a service that it is currently damaging. Thus, a misbehaving node may still be a good provider of other services that are unaffected by whatever component is causing the damage encountered.

TRUSTING SIMILAR NODES

Another criterion that can be used to weight state and threat information provided by other nodes in the network is a measure of similarity. Recall that network nodes in our target environment start with a standard (and *a priori* trusted) system image. Changes to the system binaries or configurations are tracked by the BITSI kernel, and that information can be used locally to identify how different the current state of the system is from its base state.

In principle, nodes that are still in their base state are more worthy of trust than nodes that have been heavily modified during the mission. When a node receives inferences about a possible attacker or vulnerabilities identified from another node, it will consider both the context, and the reputation of the node providing the information. Nodes that have been independently modified in similar ways are likely to be exposed to the same kinds of vulnerabilities, and should give more weight to information provided by "similar" nodes.

For that purpose, the encrypted state information shared among BITSI kernels also includes a summarized history of the changes and variations from the base state. State similarity is used to weight the information provided by similar (or dissimilar) nodes (much like context information is used to weight the opinion of other nodes based on current network conditions and environmental properties).

In [9] we have already investigated different ways of combining reputation scores based on similarity between nodes. For example, in a danger-driven system, the goal is to provide for mission continuity. As such, preempting damage is of great value. Thus, when calculating reputation scores, it is helpful if the input of each other node is considered based upon its similarity to the node carrying out the evaluation. That is, if node *A* is a web server, it makes sense to weight the input of other web servers more highly when calculating a global reputation (especially if one thinks of reputation as a measure of how likely another node is to damage *you*).

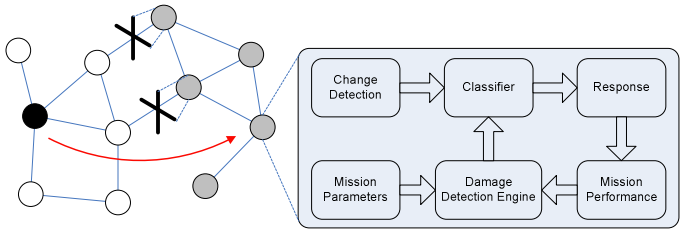

Figure 6. Collaborative immune response

A THIRD LEVEL OF DEFENSE: COLLABORATIVE IMMUNE RESPONSE

The highest level of the BITSI system is designed to provide a collaborative immune response. While Level I detects damage locally, and Level II shares this damage information, Level III is designed to collaboratively work to reduce network impact. For example, in Figure 6 we see the system determining that the attacking node should be blocked collaboratively, not locally. This saves network resources and limits the scope of nodes that are macroscopically detected as "bad".

In another example, the Level III system helps established groups determine how to handle a node that is new to the system. If the node has a series of changes that have previously been detected as damaging, the system may disallow service to the node. This type of learned response is extremely powerful but also dangerous if an attacker can exploit the immune reaction to his own ends. Further research is needed on this third layer; however, by focusing on damage, and by accepting the assumption that attackers are less prevalent than defenders we are confident that auto-immune responses can be prevented or at least minimized.

BITSI IN ACTION: ROUTING

Understanding the potential of BITSI is perhaps easiest when walking though a simple scenario that shows the system in action. Consider the snapshot of a MANET illustrated in Figure 7 where a node A is sending mission-relevant messages to a node B. The messages generated at node A are forwarded through multiple hops from A to B.

The route from A to B is dynamically chosen by the routing protocol (HSLS [10] in this example) in response to the topology changes caused by node mobility.

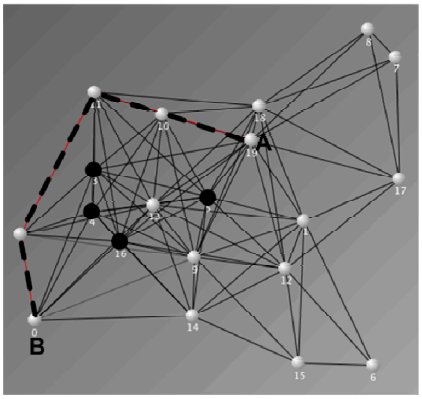

Figure 7. BITSI in action: Example of an immune response influencing the routing algorithm to isolate malicious nodes (NS-2 simulation)

Consider now the case where some of the nodes in the network (black nodes) have been attacked and compromised to deliberately capture mission-relevant packets and selectively corrupt traffic at the application before retransmission when forwarding traffic from node *A* to *B*. Note that this packet corruption happens at a higher level and may not be easy to detect at intermediate nodes in the path simply acting as relays – thus it is very difficult for conventional IDS techniques to correctly detect this attack through traffic analysis.

In BITSI, we address the problem at three levels. At the lower level, local application damage is detected and correlated with corrupted data packets and with a list of potentially malicious nodes. At a second level, reputation information is shared amongst nodes, enabling each node to weight its own reputation estimates. At a third level, a collaborative response emerges to isolate the attacker at the routing level.

Although packet corruption is not directly detectable at the BITSI kernel level, the mission-critical application running in node B *can* detect that traffic is arriving in an unusable state at the application level. That information is relayed as damage to the BITSI kernel, triggering an immune reaction.

Although unable to positively identify the nodes responsible for corrupting the packet, the BITSI kernel at node B can estimate the likely path followed from source (A) to destination (B). It does that based on local topology information available from HSLS. The kernel at node B then reports the local damage to the BITSI kernel at each node in the path (including the source), slightly reducing their reputation and trust worthiness regarding the forwarding of such critical messages to node B. The notification happens repeatedly while "bad" (i.e. corrupted) packets continue to reach node B.

The reported reputation is cumulative but it decays over time – that is, a node may be "forgiven" and recover its reputation if it stops receiving bad reports from a peer BITSI kernel. The reputation is used by the BITSI kernel at each node as a moderator of link cost. Routes that use nodes that have a poor reputation will have a higher cost and will be disfavored. Traffic will tend to avoid such nodes if lower-cost routes are available *but will still use them as a route of last resort.*

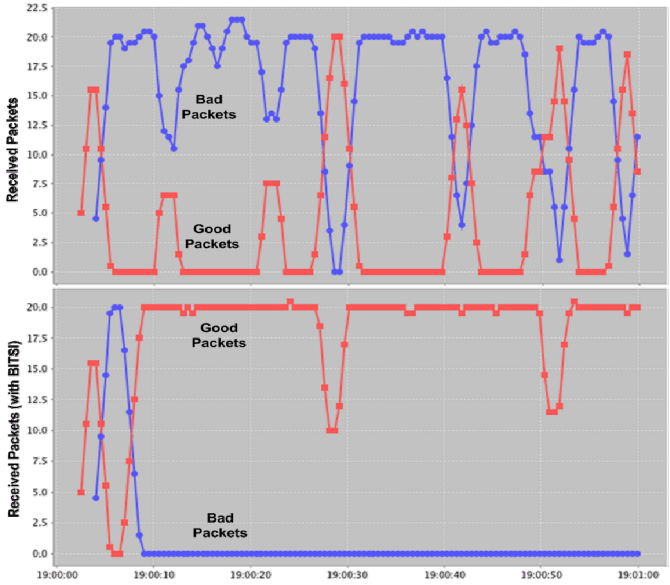

Figure 8. Number of Good versus Bad (corrupted) packets/sec. received without and with BITSI. Results from one of the several topologies simulated with NS-2.

Figure 7 illustrates a snapshot of a 20-mobile node scenario simulated in NS-2. Nodes are moving at 1m/s in a 400x400 meters grid. There is a single data flow in this scenario, with packets going from node A to a "critical-application" in node B. The messages are 512 bytes-long, sent 20 times a second. The goal of BITSI in this example is to increase, as much as possible, the number of "good" (i.e. non-corrupted) packets arriving at B through changes in the routing protocol. Only node B is capable of identifying a corrupted packet thus packet signature based detection at the intermediate nodes is not available.

Our preliminary simulation results are illustrated in Figure 8. The top graph shows the average number of good packets (i.e. not-corrupted) and bad packets received per second. In this example, because of the position of the malicious nodes (participating in most of the routes between A and B) the number of bad packets is generally

higher than the number of good packets in time. The second graph of the same figure shows the same scenario with BITSI enabled. In that case, when corrupted packets reach node B, damage is detected and the reputation of all nodes in the path (as estimated by B) is decreased – which directly affects the routing costs on links associated with those nodes, leading to a change in the data path to avoid the potentially malicious nodes, as illustrated in Figure 7.

This experiment was repeated 100 times with and without BITSI, using different, randomly generated initial topologies at each run. The results show that without BITSI the average of good packets received at B on a 60-second run is approximately 660.5 with a sample std. dev of approx. 355.2. The same topologies tested with BITSI on resulted in good packet average of 879.2, (sample std. dev is 311.6). These results show, with 95% confidence, a statistically significant improvement on the number of good packets received with BITSI enabled.

CONCLUSIONS AND FUTURE WORK

In this paper we presented an overview of a three-tier Danger Theory-inspired security system for MANETs. Unlike other MANET security solutions, our goal is not, at the highest level, to identify misbehaving nodes. Instead, the focus of BITSI is to provide for mission continuity by detecting damage and reconfiguring nodes to mitigate it. The three-layer approach is distributed and self-organizing. As such, it is capable of functioning when the network becomes disjoint. Furthermore, by focusing on a mission-centric attribute such as damage, the system should be able to handle previously unconsidered attacks.

While BITSI represents an interesting and novel approach, it is certainly not a replacement for existing security solutions, nor is it a universal panacea. For example, attacks that do not cause detectable damage to the mission will not be detected by BITSI. Thus, an attack that results only in data exfiltration may not be detected by the system. Similarly, existing signature-based detection methods are still needed by BITSI, both to prevent predictable damage and to trigger changes in reputation.

Our future work will involve building a fully functional three-layer prototype. This work is particularly challenging due to the wide variety of mission scenarios and lack of large test environments in which experiments can be constructed. However, we remain confident that this approach can provide insight into new techniques for protecting critical mission functions.

REFERENCES

[1] Carvalho M., Ford R., Allen W.H., and Marin G. (2008), "Securing MANETs with BITSI: Danger Theory and Mission Continuity", SPIE Defense and Security Conference, Orlando, 17-20 March 2008.

[2] Ford R., Carvalho M., Allen W. (2007), "BITSI: A Biologically-Inspired Adaptive Defense Framework", Adaptive and Resilient Computer Security Workshop, Santa Fe Institute, 2007.

[3] Dietterle, R., The Future Combat Systems (FCS), in: Military Communications Conference, volume 5, pages 3269-3273, 2005.

[4] Kephart, J.O., Sorkin, G., Swimmer, M., and White, S.R. (1997), "Blueprint for a Computer Immune System", in the Proceedings of the International Virus Bulletin Conference, Virus Bulletin PLC, San Francisco, CA, 1997.

[5] Forrest, S., Hofmeyr, S., Somayaji, A., and Longstaff, T. (1996), "A Sense of Self for Unix Processes", in the Proceedings of the 1996 IEEE Symposium on Research in Security and Privacy, pages 120-128, IEEE Computer Society Press, 1996.

[6] Matzinger, P, (1994) "Tolerance, Danger and the Extended Family", Annual Review of Immunology, volume 12, pages 991-1045, 1994.

[7] Aickelin U., Bentley P., Cayzer S., Kim J., and McLeod J., Danger theory: The link between AIS and IDS?, 2nd International Conference in Artificial Immune Systems (ICARIS 2003), pages 147-155, Edinburgh, UK, 2003

[8] Bradshaw, J.M. et al.: "KAoS: Toward an Industrial-Strength Generic Agent Architecture." Software Agents, AAAI Press/MIT Press, Cambridge, Mass. 1997, pp. 375-418.

[9] Hoffman K., Ondi A., Ford R., Carvalho M., Brown D., Allen W., Marin G. (2008), "One of these things is not like the Others: Collaborative filtering in MANETs", in the Proceedings of the EICAR Conference 2008, Laval, France

[10] Hazy Sighted Link State (HSLS) Routing: A Scalable Link State Algorithm, BBN Technical Memorandum No. 1301. August 2001.