Intrusion and Anomaly Detection Model Exchange for Mobile Ad-Hoc Networks

Gabriela F. Cretu, Janak J. Parekh, Ke Wang, Salvatore J. Stolfo
Department of Computer Science
Columbia University
New York, US
{gcretu, janak, kewang, sal}@cs.columbia.edu

Abstract—**Mobile Ad-hoc NETworks (MANETs) pose unique security requirements and challenges due to their reliance on open, peer-to-peer models that often don't require authentication between nodes. Additionally, the limited processing power and battery life of the devices used in a MANET also prevent the adoption of heavy-duty cryptographic techniques. While traditional misuse-based Intrusion Detection Systems (IDSes) may work in a MANET, watching for packet dropouts or unknown outsiders is difficult as both occur frequently in both malicious and non-malicious traffic. Anomaly detection approaches hold out more promise, as they utilize learning techniques to adapt to the wireless environment and flag malicious data. The anomaly detection model can also create device behavior profiles, which peers can utilize to help determine its trustworthiness. However, computing the anomaly model itself is a time-consuming and processor-heavy task. To avoid this, we propose the use of model exchange as a device moves between different networks as a means to minimize computation and traffic utilization. Any node should be able to obtain peers' model(s) and evaluate it against its own model of "normal" behavior. We present this model, discuss scenarios in which it may be used, and provide preliminary results and a framework for future implementation.**

Keywords-mobile ad-hoc networks; intrusion detection; anomaly detection; model exchange; profiling; model aggregation

I. INTRODUCTION

MANETs, or Mobile Ad-hoc NETworks, have recently gained adoption in a broad variety of environments thanks to improvements in wireless networking technology and the need for rapid mobile deployment. However, MANETs pose unique security requirements and challenges. Since they enable devices to enter and leave a network without previous authentication or certification, a MANET node cannot be assumed to be trusted. Traditional security approaches, like firewalls, do not extend well to MANETs, where both benign and malicious parties have full access to communicate with peers. Additionally, the limited processing power and battery life of these devices also prevent the adoption of heavy-duty cryptographic techniques. While traditional misuse-based IDSes may work in a MANET, the efficacy of these techniques is in question. A traditional IDS might watch for packet dropouts or unknown outsiders as a sign that an inbound communication may be malicious. In a MANET however, both of these occurrences are commonplace amongst benign nodes as well. Additionally, most MANET-based misuse detectors have focused on routing-specific attacks, e.g. [5], at the cost of ignoring actual application vulnerabilities.

Anomaly detection approaches hold out more promise, as they utilize learning techniques to enable adaptation to the wireless environment and to the tasks and communications being utilized in that environment. Anomaly detectors generate a *model* of the observed data (traffic, behavior, etc), and compare new data against this model to check for anomalies. It is relatively simple to determine if peer communications fit that model, and to establish policies ignoring data that is flagged as malicious (e.g., [9, 10]). The model also acts as a *profile of device behavior*, which can be utilized by peers to help determine its trustworthiness by comparing their mutual models exchanged between the devices.

This concept extends the notion of mutual authentication; rather than proving one's trustworthiness via a certificate or a credential, here MANET nodes are authenticated by their behavior -- a profile of how they typically interact. Other nodes may validate the node by conformance to their own profiles, and to ensure the new node subsequently behaves in conformance with their announced profile.

Early work in building anomaly detectors for MANETs was primarily focused on header-level and routing-level features [2, 3, 4, 7]. Computing the anomaly model for traffic payload or other rich feature sets is a time-consuming and processor-heavy task, one that needs to be avoided in a battery-conscious, reduced-communication environment. To solve this, we propose the use of model exchange in a MANET to provide a balance between the need for adaptation as a device moves between different networks and the need to minimize computation and traffic utilization. Any node should be able to obtain peers' model(s) and evaluate it against its own model of "normal" behavior. The node should be able to either integrate the peer's model with its own to get a better idea of legitimate traffic being conducted on the network, or to flag the peer as suspicious if the profile is significantly different than its own.

We present this model, discuss scenarios in which it may be used, provide early results about model integration and comparison, and provide a framework for future implementation. While worms and similar malicious payloads have not yet become prevalent on MANETs, it's only a matter of time before such intrusion detection techniques are necessary [1].

II. Model distribution

In a MANET, we make the fundamental assumption that *most* nodes cannot (or prefer not to) compute an anomaly model for payloads, due to the lack of traffic, battery power, or computation ability. This requires the existence of a node that is sufficiently powerful to perform anomaly model learning and can bootstrap the MANET's model set. Depending on the location of this node, several different distribution models can be adopted:

- Use a server/desktop entity to generate the anomaly model. This is ideal for situations where the MANET is running a replica or a lightweight version of the desktop application (e.g., SMTP messaging or HTTP data transfer). In these cases, training can be done on the desktop and the model distributed to the MANET nodes when possible: (a) If the MANET nodes have WAN connectivity, they can initiate download requests to obtain the latest model from the server. (Some WAN topologies now allow for "push" models, which could be leveraged to let the desktop administer the download interval.) A hierarchical distribution can also be accomplished, whereby a single MANET node downloads the data over a potentially expensive WAN link and then utilizes the WLAN links to distribute the updated model to neighboring nodes. (b) Without WAN connectivity, MANET nodes can be initialized before deployment. This is a natural arrangement for "syncable" handheld devices (e.g., Palm/WinCE PDAs), which often have a cradle at the office/base and allow one-touch synchronization. We call this mechanism *pre-charging*. Ideally, the handheld device would contact the desktop at a regular basis, but high-quality models can reduce this dependency. Synchronization can also be accomplished with intermittent network links.

- If a desktop cannot be deployed, a more powerful MANET node can be deployed, with sufficient processing and/or battery power to perform anomaly training. This "supernode" would listen promiscuously to all visible traffic on the MANET, generate models, and distribute them to the (potentially weaker) peers. This model is decentralized and does not require WAN connectivity. However, the supernode does not see traffic that is not routed within its vicinity. A workaround to enable broader model coverage would entail periodic traffic reports from all nodes; these traffic samples should be sufficient to construct a representative model.

- Use a precomputed anomaly model. This scenario is worst-case, but can be practical in situations where the MANET's behavior is well-defined and follows a standard protocol definition. This is a variation on the first scenario, but one where the regular synchronization requirement is dropped.

- Introduce node(s) from a different MANET who has been able to compute an anomaly model. Much like the previous scenario, this works best when MANET functionality is well-defined and compatible with the other MANET.

Degraded modes can also be adopted, i.e., in scenarios where anomaly models are unavailable, mobile nodes can adopt a "defensive" posture and reject otherwise accepted traffic. While model exchange imposes an additional restriction as opposed to standalone misuse or anomaly detectors, we believe that the savings in computation time and the benefits justify these requirements.

III. Model aggregation/profiling

Once models are exchanged, they must be processed with relation to the node's own model. We propose two different mechanisms for doing this, depending on the *similarity* of the models exchanged. (A more precise definition of model similarity is implementation-specific; one is discussed in section IV of this paper.)

For models that are *similar*, the likelihood is that there are multiple MANET nodes accomplishing similar tasks and whose behavior closely matches the first node. In these cases, we are interested in *aggregating* the models to produce one unified view of the current MANET and to reduce false suspicions of anomalous behavior. This model of aggregation also enables the incremental, decentralized evolution of the MANET as the nature of the tasks and the distribution of nodes changes – in essence, it is a low-cost metalearning algorithm. The idea of aggregation was previously used in MANETs for alerts; [6] demonstrated that, by integrating security-related information at the protocol level from a wider area, false positive rate and detection rate can be improved. We believe that model aggregation will have a similar effect on the false positive rate and will give a better characterization of the environment. Appropriate aggregation algorithms must be careful to avoid the equivalent of a *learning attack*, e.g., an attacker that gradually poisons models until the anomaly classifier accepts malicious traffic as legitimate. We believe the short nature of MANET communications naturally avoids this; future work will evaluate our architecture's vulnerability to this attack technique.

In addition to model aggregation, models have a second use: they act as a behavioral *profile* of nodes. This enables peers to determine whether or not to communicate with a particular node. If the peers' models are very similar to the node in question, it suggests that the node is performing similar tasks, and is benign. A node with a *dissimilar model* is likely sending out substantially different and potentially malicious content. For example, a node sending out worm packets will generate a substantially different content distribution than a benign node; see section IV for further discussion. A determination can be made via a simple comparison, which yields a *similarity metric*. (In fact, similar profiles may then be aggregated as discussed.) Nodes can then use a predefined policy to threshold the metric and decide whether or not to cooperate with the node in question.

IV. Case study: PAYL models

To verify our hypothesis, we examine the use of model exchange with the Anomalous Payload-based Network Intrusion Detector (PAYL), developed in the Intrusion Detection Systems Lab at Columbia University. PAYL has favorable characteristics for MANET model exchange; in particular, it uses small-size models that can be easily exchanged, profiled and aggregated between nodes. In this section, we will introduce the PAYL sensor and show the methods that can be used to accomplish the above mentioned tasks. We will also show early experimental results that validate these methods.

A. *PAYL: content based anomaly detector*

We provide here a brief introduction to the technology behind the PAYL sensor, while in-depth studies can be found in [9, 10]. This anomaly detector relies on the fact that network traffic differs significantly depending on the target port and length of the payload observed. During the *training phase*, incoming packets on a given port are frequency analyzed, and the distributions are clustered together based on the payloads. This clustering process results in a number of centroids that characterize the traffic for the chosen port and packet length. By considering all the centroids obtained for different port and length, we obtain a PAYL model. Incoming packets are compared against this model in the *detection phase* to check for anomalies. It is also possible to check bidirectional traffic in the same manner and to detect worms by performing ingress/egress correlation.

Of particular interest in the MANET case is that PAYL models are small in size (~50K after compression) and can therefore be exchanged between the nodes of a low-bandwidth network.

B. *PAYL model aggregation*

As mentioned before, PAYL models are composed of centroids that capture payload byte distribution. With PAYL's incremental learning technique [9], merging models is as simple as averaging one model onto the other. If, for a specific payload length, only one of the models contains one or more computed centroids, the aggregate one will simply inherit these centroids.

This aggregation algorithm requires linear execution time (relative to the size of the model), thus satisfying the computational limitations typical to MANETs. Also, since PAYL models only contain statistical distributions, they can be distributed without encryption, as sensitive content will not be revealed.

C. *PAYL model profiling*

The profiling technique for PAYL models provides multiple levels of detail for performing analysis. We explore only some of the methods here, leaving room for more investigation in the future. At the first level, we extract the payload length distribution for the two models that we compare, and compute the Manhattan distance [9] between them. If this distance is greater than a significant threshold, we might conclude that the models display a significant difference. If not, we perform the analysis at a higher level of detail and compare based on the distance between each model's centroids for any port and packet length. This can be done either by considering only the predominant centroid for each packet length, or all centroids contained in the model.

The advantage of performing a multi-level analysis is that negative answers for divergent models can be returned very quickly, while in-depth comparisons will be performed only for very similar models. As a result, this method can satisfy low computational constraints.

D. *Experimental results*

In order to confirm the characteristics of the PAYL model profiling and aggregation, we have conducted a series of experiments on a set of four models, which we name *model1* through *model4*.

- *model1* and *model2* were generated on machines accepting similar traffic (as both machines service the same population).

- *model3* was generated on a machine that sees a more complex traffic, including more media data.

- *model4* was built out of mostly abnormal traffic populated with Code Red II, an IIS WebDAV exploit, etc.

As an example, figure 1 displays two centroids built for the same payload length and the same port in *model1* and in *model4*.

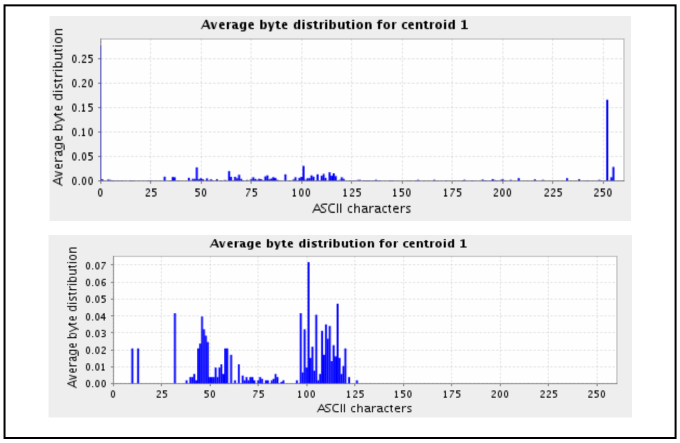

Figure 1. First centroid for port 80, length 1058 for model4 (top) and model1 (bottom)

The profiling method discussed in IV.C was used to perform comparisons between these models. It correctly decided that *model1* and *model2* are similar, while *model3* and *model4* present significant differences relative to other models. Numerical results reported by this method at various level of detail can be found in Table I, Table II and Table III.

TABLE I. MANHATTAN DISTANCES BETWEEN VARIOUS PAYLOAD LENGTH DISTRIBUTIONS

Manhattan distance between length distributions			
model1 ***model2***	***model1*** ***model3***	***model1*** ***model4***	***model3*** ***model4***
0.4210	1.5201	1.8981	0.7898

TABLE II. AVERAGE OF MANHATTAN DISTANCES COMPUTED BETWEEN THE FIRST CENTROIDS OF EACH POSSIBLE LENGTH IN EACH MODEL

Average Manhattan distances between first centroids			
model1 model2	***model1 model3***	***model1 model4***	***model3 model4***
0.5946	0.7400	1.6368	1.6330

TABLE III. AVERAGE OF MANHATTAN DISTANCES COMPUTED BETWEEN ALL CENTROIDS CORRESPONDENT TO EACH POSSIBLE LENGTH OF EACH MODEL

Average Manhattan distances between all centroids			
model1 model2	***model1 model3***	***model1 model4***	***model3 model4***
0.4276	0.6112	1.5220	1.5096

These tables show two different metrics for comparison (payload length distributions and payload content distributions). We observe that *model1* is quite similar to *model2* with respect to their average payload length distributions and content distributions. *model4* clearly appears different than the other models for both length and content distributions. However, while *model1* and *model3* are similar with respect to their content distributions, there is a significant discrepancy between their length distributions. This fact leads us to the conclusion that we need to explore more ways of calculating similarity between the models. Our ongoing research is focused on correlating these and other metrics for better performance.

After profiling the neighbor models, a MANET node could be ready to aggregate its own model with ones that are similar to it.

We tested the aggregation method to prove that we do not lose important information from the individual models. We tested both the simple (unaggregated) models and the aggregated models against the same data. Based on Table 4, we can observe that even if the number of alerts is different for each simple model, the aggregated models do not significantly shorten the spectrum of alerts that each simple model can generate. The aggregated model shows similar behavior to the ones used to create it, which implies that the aggregation method is reliable.

TABLE IV. TESTING PAYL USING MODEL1 AND MODEL2, AND THEIR AGGREGATE

No. of packets used for each test	**No. of alerts generated using different models**		
total # packets / # content packets	***model1***	***model2***	***model1+2***
127023 / 10414	149	184	149
304182 / 21812	2705	2829	2672
276332 / 26294	9684	11128	9669
353897 36780	11201	3394	2187

In Table V, we observe that there is a significant difference between the number of alerts generated by *model1* and *model3*, leading us to conclude that the two are not similar models. We plan to capture more data for similarity measurements in future experiments.

TABLE V. TESTING PAYL USING MODEL1 AND MODEL2, AND THEIR AGGREGATE

No. of packets used for each test	**No. of alerts generated using different models**		
total # packets / # content packets	***model1***	***model2***	***model1+2***
127023 / 10414	149	81	148
304182 / 21812	2705	1789	2613
276332 / 26294	9684	1138	9530
353897 / 36780	11201	2919	11040

V. CONCLUSIONS

We proposed the use of anomaly detection model exchange in a MANET environment, and its limitations and dynamics. We also introduced an initial feasibility study of model exchange by using the PAYL anomaly detector.

As discussed previously, there is further research to be conducted, including further development of algorithms for aggregation and comparison (including metalearning), design and implementation of an automated model exchange infrastructure, and possibly the use of other anomaly detection models, such as those computed by the PAD algorithm [11]. We also did not completely address the possibility of *mimicry attacks* against the aggregation and profiling techniques discussed in this paper. There are several strategies to avoiding mimicry attacks against anomaly detectors [8]; we will evaluate their applicability in later work. Finally, we also intend to explore the effect of Byzantine behavior on model aggregation.

REFERENCES

[1] R. G. Cole, N. Phamdo, M. A. Rajab, A. Terzis, “Requirements on Worm Mitigation Technologies in MANETS”, Proceedings of the 19th Workshop on Principles of Advanced and Distributed Simulation, Monterey, CA, June 2005.

[2] Y. Huang and W. Lee, “A Cooperative Intrusion Detection System for Ad Hoc Networks”, Proceedings of the ACM Workshop on Security in Ad Hoc and Sensor Networks, Fairfax, VA, October 2003.

[3] A. Patwardhan, J. Parker, A. Joshi, M. Iorga and T. Karygiannis, "Secure Routing and Intrusion Detection in Ad Hoc Networks", Proceedings of the 3rd International Conference on Pervasive Computing and Communications, Kauai Island, Hawaii, March 2005.

[4] D. Sterne, P. Balasubramanyam, et. al., "A General Cooperative Intrusion Detection Architecture for MANETs", Proceedings of the 3rd IEEE International Workshop on Information Assurance, University of Maryland, March 2005.

[5] D. Subhadrabandhu, S. Sarkar, and F. Anjum, "Efficacy of Misuse Detection in Adhoc Networks", Proceedings of the 2004 First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, Santa Clara, CA, October 2004.

[6] B. Sun, K. Wu and U. W. Pooch, "Alert aggregation in mobile ad hoc networks", Proceedings of the 2003 ACM workshop on Wireless security, San Diego, CA, September 2003.

[7] G. Vigna, S. Gwalani, K. Srinivasan, E. M. Belding-Royer and R. A. Kemmerer, "An Intrusion Detection Tool for AODV-Based Ad hoc Wireless Networks", Proceedings of 20th Annual Computer Security Applications Conference, Tucson, AZ, December 2004.

[8] D. Wagner and P. Soto, “Mimicry Attacks on Host-based Intrusion Detection Systems", Proceedings of the 9th ACM Conference on Computer and Communications Security, Washington, DC, 2002.

[9] K. Wang and S. J. Stolfo, "Anomalous Payload-based Network Intrusion Detection", Proceedings of Recent Advance in Intrusion Detection, France, September 2004.

[10] K. Wang, G. Cretu and S. J. Stolfo, " Anomalous Payload-based Worm Detection and Signature Generation", Proceedings of Recent Advance in Intrusion Detection, Seattle, September 2005.

[11] S. J. Stolfo, F. Apap, E. Eskin, K. Heller, S. Hershkop, A Honig and K. Svore, “A Comparative Evaluation of Two Algorithms For Windows Registry Anomaly Detection”, Journal of Computer Security, 2005. (unpublished).