An Intrusion Detection System for Constrained WSN and IoT Nodes Based on Binary Logistic Regression

Christiana Ioannou and Vasos Vassiliou
Department of Computer Science, University of Cyprus
RISE - Research Centre on Interactive Media, Smart Systems, and Emerging Technologies
{cioannou, vasosv}@cs.ucy.ac.cy

***Abstract*—In this paper we evaluate the feasibility of running a lightweight Intrusion Detection System within a constrained sensor or IoT node. We propose mIDS, which monitors and detects attacks using a statistical analysis tool based on Binary Logistic Regression (BLR). mIDS takes as input only local node parameters for both benign and malicious behavior and derives a normal behavior model that detects abnormalities within the constrained node. We offer a proof of correct operation by testing mIDS in a setting where network-layer attacks are present. In such a system, critical data from the routing layer is obtained and used as a basis for profiling sensor behavior. Our results show that, despite the lightweight implementation, the proposed solution achieves attack detection accuracy levels within the range of 96% - 100%.**

I. Introduction

Security is a critical subject for WSNs and the IoT, as it is mandatory and trivial at the same time. Security breaches increase with IoT networks as they inherit the security vulnerabilities of WSNs and they are directly addressable and routable over the Inernet [1]. Direct access to the Internet exposes the network to more attacks and offers the chance to easily and directly intervene with the resource constrained network [2].

Compromising sensor and IoT networks may result in inaccurate and/or misleading information thus failing in achieving the network's goal. Security techniques for wired networks have long been proposed but require memory and CPU power and cannot be used in low-power low-scale devices, such as those found at the edge of IoT networks. Potentially new forms of attacks and data compromises have to be identified and tackled with, as early as possible, to avoid the wide spread of falsified data or the denial of them. This is the general motivation for the use of Intrusion Detection Systems (IDS).

Intrusion detection systems (IDSs) are widely used for detecting unauthorized or malicious behavior within a network. Malicious behavior is defined as the network behavior created by compromised nodes with the intend to disrupt and/or compromise a network's mission [3]. The attacks' behavior depends on the network layer they are targeting and their goal. There are two main detection techniques; namely: pattern detection which identifies known attacks and anomaly detection, which identifies known, but most importantly novel attacks, within the network.

We propose mIDS, an IDS that uses anomaly detection, based on Binary Logistic Regression (BLR), to detect the presence of an attack locally in each constrained node. In this paper we focus on the detection of two representative routing attacks, namely Selective Forward and Blackhole. mIDS uses benign and malicious data from the routing network layer of each constrained node to derive detection modules. At the evaluation stage, the detection modules are implemented and installed at each constrained node. mIDS is evaluated at real-time in various network topologies achieving accuracy levels within the range 96% - 100%.

The rest of the paper is structured as follows: Section 2 describes the mIDS methodology and Section 3 presents our results. Section 4 concludes our work.

II. Methodology

To construct mIDS we took the following steps: (1) implement routing layer attacks (2) monitor and gather local sensor activity in benign and malicious scenarios at different network topologies, and (3) define local sensor activity using BLR. The rest of the section presents the steps taken to derive the BLR detection modules.

A. Network Routing Layer Attacks

For the current work we aim to evaluate BLR as a detection technique for Selective Forward and Blackhole routing layer attacks as these are described in previous works [3]–[5]. Selective Forward and Blackhole attacks aim in exploiting routing vulnerabilities and reroute traffic to the compromised node.

Selective Forward was implemented with two versions, the Forwarding Ratio (FR) and the Block Node (BN). The Forwarding Ratio version drops packets based on a predefined ratio r. For the current research r was set to 50%. When the compromised node has to forward a neighbor's packet it does so with a probability equal to (1-r) or drops it with probability r. The Block Node attack targets specific neighbor node's packets to block. The selection of the node to be blocked is

This research is part of a project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N739578 and the government of the Republic of Cyprus through the Directorate General for European Programmes, Coordination and Development.

TABLE I
PARAMETERS MONITORED

Network Layer	
Announcements Received	**Packets Sent**
Packets Forwarded	**Packets Dropped**
Packets Received	

done randomly and changes during run time. The intruder can define the period of time before switching to the next victim node. For the current work, the blocking period was set as the transmission of 10 packets.

A compromised node under the influence of a Blackhole attack, lures traffic towards it by advertising that it is one hop away from the Sink node. The Blackhole attack is used to complement the Selective Forward attack to attract more traffic towards the compromise node and selectively drops packets; thus affecting a wider range of the network [6].

B. Monitoring Local Sensor Activity

To collect the necessary network parameters needed to evaluate the detection capability of mIDS we made use of RMT. The Run-Time Monitoring Tool (RMT) is used to monitor local sensor activity at run-time and at predefined time intervals. RMT is implemented in Contiki and collects local sensor activity from the MAC and network layer [7]. RMT is used to collect local sensor activity from benign and malicious scenarios for the mIDS training stage. At the evaluation stage, RMT supplies the detection modules with real time local sensor layer activity to evaluate. Table I lists all the parameters monitored by RMT at the network layer. The parameters in bold are the ones that the BLR identifies as the most important ones.

C. Benign and Malicious Scenarios

The methodology we adopted for the training and evaluation of mIDS is based on the creation of two different types of scenarios over three different topologies. Figure 1a shows the network topology we used for the training phase and Figure 1b and 1c show the network topolies we used for the evaluation phase. Each network topology has 25 constrained nodes, one of which is the Sink node, shown in dark color. The network topology in Figure 1a places the Sink node in the middle of the grid, in the network topology in Figure 1b the Sink node is at the top of the grid, and in the network topology in Figure 1c the Sink node is placed at a random position and so are the other sensor nodes.

In a benign scenario all nodes within the network are executing a benign application. For each topology we run 10 benign simulations each with a different random seed. In a malicious scenario one node within the network is executing a malicious application. For each network topology we executed 24 malicious scenarios for the Selective Forward attack, and 24 malicious scenarios for the Selective Forward and Blackhole attack. The RMT was installed in every node to monitor local sensor activity.

The scenarios were evaluated in a 15-minute experimental time. The nodes started transmitting data after the first 2 minutes of experimental time. The 2 minute window was set to allow the sensor nodes to be connected and reach a steady state. The RMT monitoring time interval was set to 30 seconds, thus the data retrieved in the 15 minute experimental time was for 26 monitoring periods. Only the first 25 monitoring time intervals were used, as experimental time expired before certain nodes had the chance to compute the last monitoring time interval.

In both benign and malicious applications, the data rate was set to 1 packet of 48 bytes per second (for an effective data rate of 384 bps). Every node was generating 30 packets per monitoring time interval and if it was a relay node, it was also forwarding the packets received by its neighboring nodes. At the end of the experimental time, the data gathered for each malicious scenario at the network layer was for 25 monitoring periods from 24 constrained nodes. At each monitoring period data was gathered from the Network Layer.

The location of the sensor node can define the impact of the malicious attack within the network [6]. To derive a more general detection model and eliminate the location factor, we averaged the data of each malicious node in every malicious scenario based on the distance from the Sink node and the monitoring time interval. The average data per monitoring time was used as an input for training mIDS.

D. mIDS Detection Modules

The BLR statistical tool is used to derive mIDS detection modules. BLR is used to predict a binary dependent variable based on a set of independent variables. In this research study the dependent variable to be predicted is whether the node activity is caused by a compromised sensor node or not. The independent variables are the set of network layer activities shown in Table I.

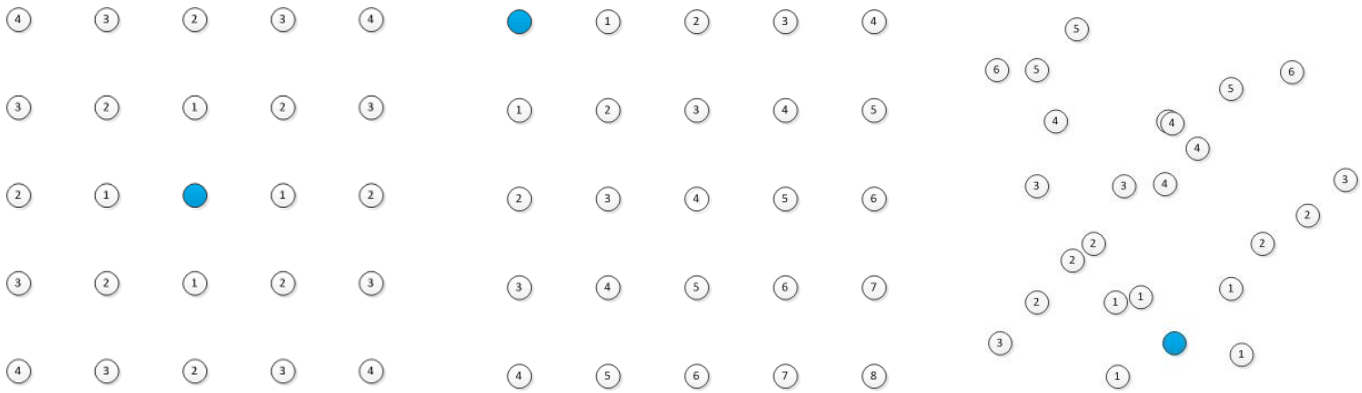
There are two phases in this approach, the training and the evaluation phase. The training phase determines whether the method can be used to identify malicious activity and which independent variables are the most significant. The end result of the training phase is a regression model to be used in the evaluation phase [5]. The evaluation phase tests the model at real-time.

The dependent variable has two values; 0, which means ”benign”, and 1 which means ”malicious”. Equation (1) shows probability P, the probability that the network activity is ”malicious”.

$$P = \frac{e^{\alpha+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_n x_n}}{1+e^{\alpha+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_n x_n}} \tag{1}$$

where n is the number of independent variables. Each x_i represents the value of the ith independent variable, α is a constant, β is the regression coefficient. For more information about BLR refer to [5] and references therein.

For the training phase, training sets were created using the independent variables gathered by RMT in the network topology were the Sink node is placed in the middle. To



(a) Sink node in the middle of the grid (b) Sink node on top left corner of the grid (c) Random Placement of the Sink node

Fig. 1. Network Topologies Used for Training and Evaluation

TABLE II
BLR NETWORK LAYER MODEL -SELECTIVE FORWARD

	Estimated Values	
Variables	**Forwarding Ratio**	**Block Node**
α	-0.91	0.98
$\beta_{PacketsForwarded}$	-1.37	-1.26
$\beta_{PacketsSent}$	0.04	-0.04
$\beta_{AnnouncementsRecvd}$	0.22	0.25
$\beta_{PacketsDropped}$	8.84	12.38

TABLE III
BLR NETWORK LAYER MODEL -SELECTIVE FORWARD AND BLACKHOLE

	Estimated Values	
Variables	**Forwarding Ratio**	**Block Node**
α	-25.03	43.15
$\beta_{PacketsForwarded}$	0.076	-0.056
$\beta_{PacketsSent}$	-0.156	-2.98
$\beta_{AnnouncementsRecvd}$	-0.49	-34.88
$\beta_{PacketsDropped}$	2.56	8.9

classify the vectors, whether they are taken from benign or malicious nodes, we included the dependent variable. The classification process allows the logistic regression to evaluate the significance of each independent variable in identifying the nature of the activity.

The final training set is shown in equation (2).

$$\underline{X}_{training} = \{t_{i,1}, t_{i,2}, \ldots, t_{i,j}, \ldots, u_{k,1}, u_{k,i}, \ldots, u_{k,j}\} \quad (2)$$

where $\mathbf{t}_{i,j}$ represents the metric j made by a benign sensor node at the time interval i and $\mathbf{u}_{k,j}$ represents the metric j made by viral sensor node at the time interval k.

The vectors in the set $X_{training}$ are used as the input of independent variables to BLR to derive the model. The coefficients for each attack as well as the significant independent variables are shown in Tables II, and III. Both types of attacks have the same significant independent variables.

III. RESULTS

We implemented the BLR training models as part of our lightweight IDS called mIDS within the Contiki OS. Our experiments were conducted using the Tmote Sky platform

TABLE IV
BLR DETECTION MODELS

Attack		Network Layer
Selective Forward	Forwarding Ratio	✓
	Block Node	✓
Selective Forward & Blackhole	Forwarding Ratio	✓
	Block Node	✓

(also known as TelosB). At predefined time intervals RMT provides local sensor activity, from the Network Layer, to mIDS, which in turn evaluates the sensor activity and classifies it as benign or malicious in real time. The current section presents the results of our evaluation of mIDS. We show that mIDS detection models can detect attacks with high accuracy at the layer the attack exploits. We also show that mIDS can detect malicious behavior in unknown network topologies. We trained our BLR model with only one network topology and evaluated our BLR model with two other network topologies, one of which is a network topology with random node placement.

The evaluation of our BLR models has been performed using the COOJA simulator that runs real sensor code. The COOJA simulator allows the formation of large-scale WSNs and has proven to produce realistic results [8]. We developed BLR detection models for each type of attack as shown in Table IV.

For each network topology and for each attack, we executed 24 malicious scenarios. In each malicious scenario one node is under the influence of an attack. At predefined intervals of one minute, called epochs, mIDS starts evaluating the data retrieved by RMT since the last epoch, using the BLR models. At the training stage, the value '0' was assigned for the benign activity and '1' for the malicious activity. If the BRL model's detection probability P exceeds a predefined threshold such as 0.5, the sensor activity is classified as malicious since it gets closer to the value of 1. The execution time of each run was set to 15 minutes. During the first two minutes any alarms that may be raised are ignored, since the network is not considered to be at a steady state. We average our results per attack and

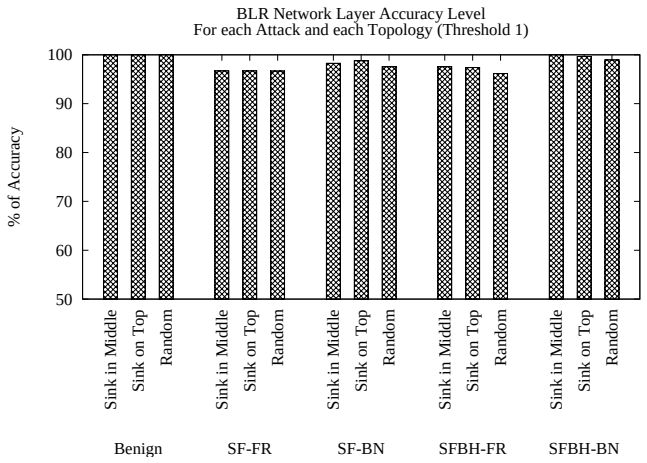


Fig. 2. Average accuracy level of BLR Network Layer model (Threshold 1)

present them using the Accuracy level (see equation (3)). The True Positive/Negative values are the correct predictions of the model. True Positive means that the model was given an activity that was benign and was correctly classified. The same applies to True Negative values when given a viral activity and the model correctly identified it as viral. The Total Population value is the total number of network activity evaluated, benign and viral.

$$ACC = \frac{\sum TruePositives + \sum TrueNegatives}{\sum TotalPopulation} \quad (3)$$

A. BLR Network Layer Model

We evaluated the BLR Network Layer models using the highest BLR threshold of 1, meaning that when BLR detection model had the result of 1 then the input network activity was generated by a malicious node. Setting up a high threshold has the disadvantage of possible high false negative alarms but also minimizes the false positive alarms. Our results were encouraging as the BLR Network Layer model was able to reach 100% accuracy level. Figure 2 presents the average accuracy level results of the BLR Network Layer model for each attack type and for each topology. In the network topology in which the Sink was in the middle, as shown in Figure 2, we achieved 100% accuracy level with the Benign scenario and the malicious scenario SFBH-BN. An accuracy level of 100% states that the Network Layer BLR model was able to detect the malicious nodes in the malicious scenario without raising false alarms. In the Benign scenario the BLR Network Layer model did not raise any alarm indicating that no attack was present in the network. The lowest accuracy level was 96.70% by Selective Forward - Forwarding Ratio attack.

In the malicious scenarios in which the accuracy level was less than 100%, the BLR Network Layer model had False Positives. The BLR Network Layer model was not able to detect all malicious nodes within the network. However, when an alarm was raised by a malicious node, the detection and the identification of the malicious node was correct.

Figure 2 shows our results when the Sink is on top of the network. The lowest accuracy level was again 96.70% and this was exhibited by the Selective Forward - Forwarding Ratio scenarios. The Benign scenario derived 100% accuracy level in all three topologies including our random topology shown in Figure 2.

The attacks that had the least accuracy level are the Selective Forward attacks. The reason being is that they select which packet to forward or which node to block randomly. The Forwarding ratio attack has a 50% probability to forward the packet and the decision is made at run time. The Block Node attack, does not forward the packet of a specific node at a predefined number of packets received interval. The block node may not forward any packets at the specific interval; therefore, the malicious node may not affect the network. When Blackhole attack is present and lures traffic toward the malicious node, the effects are more evident and the malicious nodes are detected more easily.

TABLE V
MIDS: SENSOR NODES - MEMORY

	RAM	Flash
TelosB [10]	10KB	48KB
Zolertia Z1	8KB	92KB
Inga-2 [11]	16KB	128KB
MAXFOR: [12]		
MTM-CM5000-MSP	10KB	48KB
MTM-CM3300-MSP	10KB	48KB
MTM-CM3000-MSP	10KB	48KB
Wismote	16KB	256KB

TABLE VI
MEMORY OVERHEAD OF MIDS

	RMT	mIDS
ROM	25.95 KB	34.69 KB
RAM	6.36 KB	6.35 KB

B. mIDS Overhead

The current section presents the memory and energy overhead that mIDS imposes on the sensor nodes. To retrieve the energery overhead we enable Contiki O/S' powertrace tool, a run-time power profiling tool that uses power state tracking to estimate the power consumtption of each node [9]. Enabling the powertrace tool imposes additional memory. The overheads presented here are the maximum possible overheads that mIDS will impose on the sensor node. Each sensor node had the powertrace, RMT and mIDS tools enabled. mIDS can be customized to fit the needs of the network and at the same time to decrease computational and energy overheads

We trained and evaluated our IDS taking into consideration the memory overheads imposed on sensor nodes. Our experiments were conducted using the Tmote Sky platform (also known as TelosB) that has 10KB of RAM memory and 48 KB of Flash memory. Table V shows the memory availability for various wireless sensor nodes. Table VI shows in KB

TABLE VII
OPERATING CONDITIONS IN TMOTE SKY

Typical Conditions	Min	NOM	MAX	Unit
Voltage	2.1		3.6	V
Free air temperature	-40		85	C
MCU on, Radio RX		21.8	23	mA
MCU on, Radio TX		19.5	21	mA
MCU on, Radio off		1800	2400	μA
MCU idle, Radio off		54.5	1200	μA
MCU standby		5.1	21.0	μA

TABLE VIII
MIDS ENERGY AND POWER OVERHEAD

	Energy (mJ)	**Power (mW)**
mIDS	951155	106
Powertrace	94559.27947	105

the memory required by mIDS. The first column of the table shows the memory overhead of the RMT tool without enabling mIDS. The next column shows the memory required in ROM and RAM, when we enable mIDS. When mIDS is enabled, it imposes extra 9.58 KB of ROM and 0.15KB of RAM memory. The mIDS memory requirements are less than what sensors have available. Furthermore, the size of ROM can be decreased by customizing mIDS to the needs of the application and detection. RMT currently monitors network layers that are not used in the detection phase. Compared to the work of [3], their IDS' memory overhead starts with 46KB of ROM and increases as the number of the nodes within the network increase.

$$\begin{aligned} Energy\,(mJ) = (&transmit * 19.5mA + listen * 21.8mA \\ &+ CPU * 1.8mA + LPM * 0.0545mA) \\ &* 3V/4096 * 8 \end{aligned} \tag{4}$$

$$Power(mW) = \frac{Energy(mJ)}{Time(s)} \tag{5}$$

1) Power and Energy Overhead: To determine the energy and power overheads of mIDS we simulated a two sensor node network, which included the Sink node and a sensor node, for 15 minutes. We evaluated two scenarios; in the first scenario the sensor node only uses the powertrace tool whereas in the second scenario both powertrace and mIDS are enabled. We used the energy equation shown in (4) which is also used in [3] and applied the Tmote Sky operating conditions shown in Table VII. To derive the power consumption we used equation (5) taken from [3]. Table VIII shows the energy consumption

TABLE IX
MIDS: BATTERY CONSUMPTION

	% Consumption
mIDS	1.88
Powertrace	1.86

in mJ and the power in mW of our two scenarios. When enabling mIDS (which also enables powertrace) the power and energy increase by 0.58%. Table IX shows the percentage of the battery consumption for the two scenarios.

IV. CONCLUSIONS

In this paper we presented and evaluated mIDS, an Intrusion Detection System, for constrained WSN and IoT nodes. Detection modules were created in an offline training stage using Binary Logistic Regression using benign local node activity and malicious activity from two representative routing attacks. In the real-time evaluation stage, mIDS achieved accuracy levels within the range of 96% up to 100%. Four local sensor parameters from the network layer were enough to detect the presence of the attacks with accuracy more than 96%. Using training data from one network topology was also sufficient to detect attacks in similar network topologies, in regards to the size and network density. mIDS was evaluated using multiple network topologies, two of which were not used in the training phase. We found that the energy and memory overheads imposed by mIDS are not deterrent factors to use mIDS. Furthermore, the overheads can be further reduced, as mIDS can be customised based on the needs of the applications.

REFERENCES

[1] J. Zhou, Z. Cao, X. Dong, and A. V. Vasilakos, "Security and Privacy for Cloud-Based IoT: Challenges," *IEEE Communications Magazine*, vol. 55, 2017.

[2] US-CERT. (2016, Nov) Alert (TA16-288A) Heightened DDoS Threat Posed by Mirai and Other Botnets. [Online]. Available: https://www.us-cert.gov/ncas/alerts/TA16-288A

[3] S. Raza, L. Wallgren, and T. Voigt, "SVELTE: Real-time Intrusion Detection in the Internet of Things," *Ad hoc networks*, vol. 11, no. 8, pp. 2661–2674, 2013.

[4] A. P. R. da Silva, M. H. T. Martins, B. P. S. Rocha, A. A. F. Loureiro, L. B. Ruiz, and H. C. Wong, "Decentralized Intrusion Detection in Wireless Sensor Networks," in *Proceedings of the 1st ACM International Workshop on Quality of Service &Amp; Security in Wireless and Mobile Networks*, ser. Q2SWinet '05. New York, NY, USA: ACM, 2005, pp. 16–23.

[5] C. Ioannou, V. Vassiliou, and C. Sergiou, "An Intrusion Detection System for Wireless Sensor Networks," in *2017 24rd International Conference on Telecommunications (ICT)*, May 2017.

[6] C. Ioannou and V. Vassiliou, "The Impact of Network Layer Attacks in Wireless Sensor Networks," in *International Workshop on Secure Internet of Things (SIoT 2016)*, Crete, Greece, Sep. 2016.

[7] C. Ioannou, V. Vassiliou, and C. Sergiou, "RMT: A Wireless Sensor Network Monitoring Tool," in *Proceedings of the 13th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks*, ser. PE-WASUN '16. New York, NY, USA: ACM, 2016.

[8] F. Österlind, "Improving Low-Power Wireless Protocols With Timing-Accurate Simulation," 2011.

[9] A. Dunkels, J. Eriksson, N. Finne, and N. Tsiftes, "Powertrace: Network-Level Power Profiling for Low-power Wireless Networks," Swedish Institute of Computer Science, Tech. Rep., 2011.

[10] *Tmote Sky Ultra Low Power IEEE 802.15.4 compliant wireless sensor module*, Moteiv Corporation, 6 2006.

[11] F. Büsching, U. Kulau, and L. Wolf, "Demo: INGA - An Inexpensive Node for General Applications," in *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '11. Seattle, WA, USA: ACM, 2011.

[12] *Sensor Network Makes Sensational World*, MAXFOR Technology Inc, 8 2011.