Anomaly-Based Intrusion Detection System for Embedded Devices on Internet

Deepak Mehta, Alie El-Din Mady, and Menouer Boubekeur
United Technologies Research Center
Cork, Ireland
Email: {mehtad,madyaa,boubekm}@utrc.utc.com

Devu Manikantan Shila
United Technologies Research Center
East Hartford, Connecticut
Email: manikad@utrc.utc.com

***Abstract*—Embedded devices connected to the Internet ranging from garage door openers, home thermostats, home automation systems to automobiles, are continuously exploited by remote attack vectors. According to OWASP Internet of Things project, these vulnerabilities are due to insecure web interfaces, insufficient authentication and authorization, insufficient transport layer protection, broken cryptography, insecure software/firmware updates, or poor physical security. As opposed to PowerPC systems, embedded devices lack resources to run advanced attack detection or anti-virus softwares. Moreover, embedded devices are often mass produced (thousand to millions) and share a static security footprint. Hence, a successful attack on a single device can be replicated across other devices with minimal effort. There exists a significant need towards developing a resilient cyber security methodology that provides scalable and efficient *intrusion detection and resilient architecture*. In this paper, we present an efficient hierarchical anomaly-based intrusion detection method and resilient policy framework that enables the system to detect suspicious activity and continue the operation with minimum functionality.**

Keywords–Cyber Security; Embedded devices; Internet of Things; Intrusion Detection; Resilient policy;

I. INTRODUCTION

The continued rise of cyber attacks together with the evolving skills of the attackers, and inefficiency of the traditional security algorithms employed by the embedded devices to defend against advanced and sophisticated attacks, necessitate the development of novel defense and resilient techniques. Targeted aggressive attacks use well-researched and well-funded multi-vector tactics to introduce stealthy and persistent malware in connected embedded systems (i.e., Internet of Things) infrastructure systems. Examples include ThingBot, Ransomware [3]-[5], etc. The insecure composition of legacy devices with web interfaces (such as HTTP, PHP, etc.) [3] further enlarges the attack surfaces of these systems. Furthermore, vulnerabilities for embedded devices are discovered daily, which can be easily replicated to many other devices connected to the Internet. These factors highlight the importance of designing a scalable detection scheme that not only detect attacks but also minimize the attack impact and prevent spreading of attack over other similar embedded devices.

A handful of approaches exist along the tangents of attack detection for large scale systems and resilient algorithms for enabling minimal system services even under attack [6], [7], [8], [9], [10]. However, these approaches require high computational resources, which make it infeasible for embedded devices with limited resources. In addition, these approaches rely on operational data of the system, which in turn limits the ability to detect a wide variety of attacks.

This paper proposes an Intrusion Detection and Resilient Policy methodology for embedded devices connected to the Internet. In order to assist the development of the proposed approach, we have summarized various attack models for Internet connected embedded systems. This work aims to extend our preliminary proposal [1]. The paper is organized as follows: In Section II we first describe the overall methodology of hierarchical based intrusion detection and resilient policy for detection. In Section III we describe the details of our approach for anomaly-based intrusion detection system for embedded devices on internet. In Section IV we evaluate the developed methodology. Finally, in Section V we present conclusions and future work.

II. CYBER-PHYSICAL ATTACKS, DETECTION AND MITIGATION

A. Adversary models

We consider several different broad strategies an attacker may employ against Internet facing embedded devices [11]: a) *circumvention attack* finds exploits that does not depend on the security properties of the embedded device; b) *deputy attack* finds a way to exploit the vulnerabilities of a benign program in a malicious way; c) *brute force attack* attempts all possible cyber security keys until an exploit is found that succeeds; d) *dictionary attack* tries only some key space possibilities which are deemed most likely to succeed; e) *probing attack* uses probe packets to learn properties of the security method execution needed to construct an attack; f) *denial of service* attempts to make the IoT device unavailable; g) *backdoor attack* uses hardcoded credentials or passwords to gain access to the system; h) *code injection or reuse* attack uses vulnerable programs or coding errors to inject malicious code into the device.

B. Hierarchical based Intrusion Detection

This approach considers two level of Intrusion Detection System (IDS), as shown in Figure 1: local IDS and supervisory IDS. The local IDS is deployed on every embedded device, which uses various information, such as power consumption, memory usage and environmental data to learn and build a time series based statistical model. The resulting statistical model is used to detect any anomalous behaviors at the device layer and the anomalous findings are further reported to the

 ISBN: 978-1-61208-585-2

supervisory IDS for decision making. The supervisory IDS, deployed at the gateway, learns and builds a data correlation model that captures the dependencies between all connected devices during the deployment phase (we assume normal operational behavior during the period of installation). When an anomaly is reported from the local IDS, the supervisory IDS uses the data correlation model to confirm the intrusion based on other devices behavior (e.g., the behavior of other correlated devices will not change when the device is attacked, which is used as an evidence). In order to prevent supervisory IDS from detecting attacks, an attacker has to learn the group of correlated devices and tamper them accordingly, which is a complex task. In the event of an attack, supervisory IDS will apply a resilient policy to: a) thwart attacks on other similar devices by trigerring a change in the configuration of the devices; b) isolates the attacked devices and continues to provide the same services via use of virtual sensors. In this paper we will focus on supervisory intrusion detection.

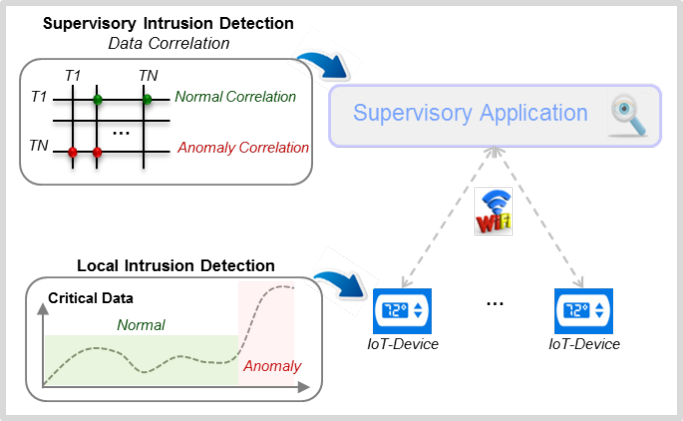

Figure 1. Hierarchical based Intrusion Detection

C. Resilient Policy

The supervisory IDS applies a resilient policy to initially isolate the attacked device from other devices. The supervisory IDS uses a combination of the data correlation model and the local statistical model to build a *virtual sensor* [10]. This virtual sensor uses prediction algorithms, such as Kalman Filtering to predict the actuation values supplied by the attacked device, and deliver the same services (e.g., actuation values) without the help of the attacked device. Moreover, the supervisory IDS also triggers a change in system configuration (e.g., a defense depending on the attack detected) to the correlated devices to prevent spreading of attack to other devices.

III. Supervisory Intrusion Detection

The core function of any IDS is to gather and analyse information in order to identify any intrusion. When the context is cyber-physical system or Internet of Things, IDS should not only monitor cyber-related metrics (e.g., network activity, CPU speed, log files) but also physical processes/measurements that govern behaviour of physical devices. IoT or sensor data consists of a continuous stream of data (aka time-series) where the time interval between successive updates could vary from milliseconds to minutes. The data produced, usually pertains to the information about the physical state of a system, i.e., temperature, pressure, voltage, power consumption, flow rate, speed, acceleration, etc. The goal is to detect intrusion not only in cyber space but also in physical space. For example, the data reported by an IoT sensor could be far from its normal behaviour or an actuator could behaves in a highly erratic manner.

The existing intrusion detection techniques can be broadly classified into two categories: *knowledge-based* and *anomaly-based* [12].

- A knowledge-based IDS uses a database of patterns/signatures (a footprint specified in terms of data packets, number of failed attempts, upper and lower bounds physical measurements etc.) of previously known attacks and system vulnerabilities. Periodically, the current signature is checked with the database to identify and prevent the same attacks in the future. The advantages of knowledge-based intrusion detection system is that it is highly affective towards well known attacks and has low false positive rate. The disadvantages are that it cannot identify new attacks and the database would need frequent updates.
- The anomaly-based [13] intrusion detection system builds a profile (or a data-model) of the *normal* behaviour using either statistical or unsupervised machine learning methods. It then uses the normal profile to flag any deviations from that profile as alerts. The advantages of anomaly-based IDS is that it can identify new attacks, but the disadvantage is that it is prone to high false positive rate.

Both approaches have been extensively studied. A reader is referred to [12] for more details.

In the following sections, we shall describe a novel approach for supervisory intrusion detection. More precisely, we will exploit the relationship between a set of given time-series for detecting anomalies. This could either be used on its own or and it could be used as an additional feature of another algorithm to improve its efficiency.

A. Correlation-based Anomaly Detection

Problem Setting. We are given a database of unlabelled n time series $T = \{t^1, t^2, \ldots, t^n\}$ containing both normal and anomalous sub-sequences. The assumption is that the majority of them are normal. The problem is to detect anomalous subsequence within a given time-series by exploiting a set of corresponding sub-sequences of other time-series when possible.

The overall methodology of the proposed anomaly-based intrusion detection is shown in Figure 2. It first transforms the input data by aggregation and discretization prior to learning the model that represents the normal behaviour of the signals. The parameters of the model are then tuned to improve the overall performance of the method. We shall now describe each step in detail.

Transformation: Aggregation and Discretization

The aggregation step transforms a sequence of k consecutive values of one (or more) time-series by a representative value using a chosen aggregation function.

 ISBN: 978-1-61208-585-2

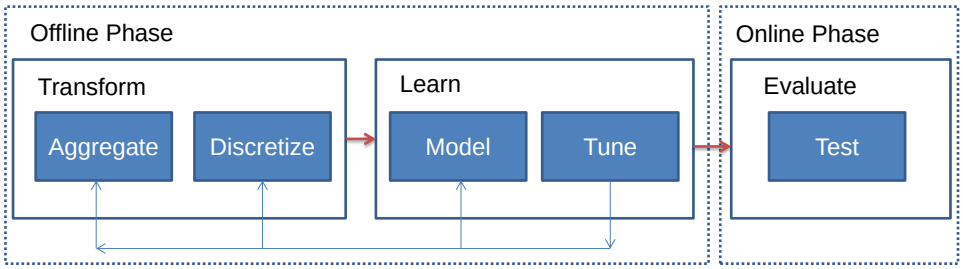

Figure 2. Overall Methodology for Anomaly-based Intrusion Detection

Given a training database of n series, $T_{train} = \{t^1, \ldots, t^n\}$, we transform it into $n \times (n-1)$ time-series denoted by $S = \{s^{pq} | p \leq n \wedge q \leq n\}$. Each time-series s^{pq} is a sequence of values, i.e., $\langle s_1^{pq}, \ldots, s_{m-k+1}^{pq} \rangle$, where each s_i^{pq} is an aggregation of sub-sequences $\langle t_i^p, \ldots, t_{i+k}^p \rangle$ and $\langle t_i^q, \ldots, t_{i+k}^q \rangle$ with a representative value when sliding a window of size k by one step at a time. The two sub-sequences are aggregated using normalised cross-correlation function (NCC). The cross-correlation function (aka. cross-covariance function) provides a measure of similarity of two sub-sequences, which is computed as follows:

$$s_i^{pq} = NCC(\langle t_i^p, \ldots, t_{i+k}^p \rangle, \langle t_i^q, \ldots, t_{i+k}^q \rangle$$

$$= \frac{\sum_i^{i+k} t_i^p \times t_i^q}{\sqrt{\sum_i^{i+k} (t_i^p)^2 \times \sum_i^{i+k} (t_i^q)^2}} \quad (1)$$

The normalized cross-correlation scoring is straightforward to interpret. It returns a value between +1 and -1 inclusive, where 1 means the two sub-sequences are exactly the same, 0 means they are very different from each other, and -1 they are exactly opposite. An example of positive correlation and no correlation between different pairs of sensors reporting temperatures is shown in Figure 3. The aggregation function is not restricted

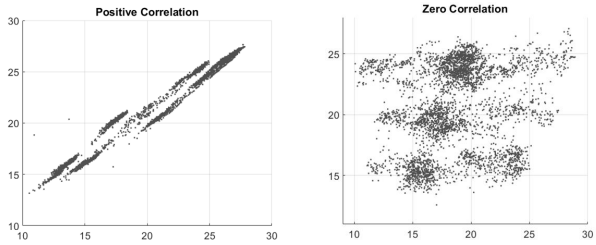

Figure 3. Example: Positive correlation (left) and No correlation (right) between temperatures readings of sensors

to NCC. Any reasonable function can be used instead.

The next step is the discretization of a given time-series, where the goal is to transform the time-series from a sequence of continuous values to a sequence of discrete intervals by dividing the amplitude range of the input time-series. There are several ways in which the intervals can be chosen. The simplest way is to create equal bin size and a more sophisticated approach is to use clustering. In this paper we have implemented the former approach. Each interval could be represented by a unique number or by an alphabet. We introduce a parameter d to denote the number of discrete intervals. Furthermore, let α_i^{pq} denotes the i^{th} discrete-interval associated with the transformed time-series s^{pq}. Discretization can decrease the dimensionality of the data and it can increase the efficiency of the algorithm for anomaly detection. A good overview on discretization is provided in [14].

Learning Model and Tuning. The goal of this step is to learn a data-model that captures the normal behaviour and tune the parameters of the model in order to maximise the detection of the anomalies while minimising the false positive rate. In the following, we describe how we generate a model for a given signal.

Let f_i^{pq} denotes the frequency of the discrete-interval α_i^{pq} observed in the time-series s^{pq}. We also introduce a parameter λ to denote the minimum percentage of non-anomalous sub-sequences within any time-series. Recall that the initial assumption was that most of the sub-sequences are normal but very few might be anomalous. The general idea is to select a set of discrete intervals that combined together represent normal portion of the time-series, which should be at least λ percentage of the m subs-sequences of window-size k within a time-series.

Let N^{pq} be the set of discrete intervals that are normally observed within time series s^{pq} with respect to the parameter λ. The subset of the discrete intervals classified as normal, i.e., $N^{pq} \subseteq \{\alpha_1^{pq}, \ldots, \alpha_d^{pq}$, is computed as follows:

- The sum of the frequencies of discrete intervals covered by N^{pq} must be greater than a given threshold, i.e., $\sum_{\alpha_j^{pq} \in N^{pq}} (f_j^{pq}/m) \geq \lambda$.
- If the i^{th} discrete interval is classified as normal ($\alpha_i^{pq} \subseteq N^{pq}$) then any j^{th} interval occurring more than the i^{th} interval ($f_j^{pq} \geq f_i^{pq}$) must also be classified as normal ($\alpha_j^{pq} \subseteq N^{pq}$).
- Minimise the number of discrete intervals classified as normal subject to the above two constraints.

For a given time-series (signal or sensor), the above step is repeated with respect to each other signal. The data-model that captures the normal behaviour of a time-series is encoded as a Boolean matrix where each row correspond to another time-series and each column corresponds to a discrete interval. An example of a Boolean matrix model for a time-series t^5 is shown in Table I. The last 4 columns denote the number of discrete intervals, i.e., $d = 4$. Each row corresponds to the set of discrete intervals that are classified as normal (when the value is 1) with respect to the q^{th} time-series which belongs to the set $\{t^1, t^2, t^3, t^4\}$.

The anomaly score of a given window of a time-series is computed by first checking whether the correlation associated

TABLE I. AN EXAMPLE OF A BOOLEAN MATRIX MODEL.

$(p=5)$	q	1	2	3	4
N^{51}	1	1	0	0	1
N^{52}	2	1	0	0	1
N^{53}	3	0	0	1	1
N^{54}	4	0	1	0	0

with the sub-sequence of another time-series falls in a discrete interval classified as normal. This is done with respect to each corresponding sub-sequence of other time-series. The anomaly score of the window is the number of discrete-intervals associated with other time-series falling in the normal category. We also introduce anomaly threshold, denoted as ϕ, as another parameter. The anomaly score is compared with the threshold, and if it is greater than the threshold than the window is classified as anomalous. In the final step, the following parameters are tuned:

1) The aggregation step introduced the parameter k to denote the length of the window.
2) The discretization step introduced the parameter d to denote the number of discrete intervals.
3) The modelling step introduced the parameter λ to denote the percentage of the number of sub-sequences assumed to be normal within a time-series.
4) The final parameter is the attack-threshold denoted as ϕ.

IV. EVALUATION.

In this section we present preliminary results. For the evaluation purpose we experimented with two sets of data:

1) The historical data for thermostat temperatures, where 12 sensor data have been used. This data is collected at the demo-site at Cork Institute of Technology (CIT), where the demo-site has an energy management system controls a small size smart-grid covering several buildings [2].
2) Real-CPU, memory, and temperature data obtained from three TI CC3200-LAUNCHXL IoT devices, considering CPU usage, memory stack size and temperature value. This data was collected from a simple demo for internet connected thermostat demonstration. the devices was using WiFi to communicate with a centralized server to send the temperature values and receive any actuation instructions.

We divided the data into training data and test data.

Training Parameters. For training purpose we restricted the values of the parameters as defined before. The size of the window was restricted to the set $\{10, 20, 40\}$. The number of discrete intervals was chosen from the set $\{10, 20\}$. The maximum percentage of the sequences assumed to be normal was chosen from the set $80\%, 85\%, 90\%, 95\%, 100\%$. The attack score threshold was chosen from the set $0.01, 0.02, 0.05, 0.1, 0.15, 0.2, 0.5$. During the training phase, the parameters of the model for representing the normal behaviour was tuned from the above combination of parameter space.

Attack Model. To test the performance of our approach we injected the attack by perturbing the test data, which relied on three parameters: (1) *disturbance magnitude* reflects the percentage of the amplitude changed in the original value. The set of percentage values that were used are $\{25\%, 20\%, 15\%, 10\%, 5\%\}$. Both increase and decrease was allowed. (2) *attack window size* denotes the size of the window chosen for injecting perturbation. (3) *disturbance behaviour* defines whether the changed introduced over a window was fixed or variable.

The results for the two sets are summarised in the following confusion matrices:

TABLE II. CONFUSION MATRIX FOR THERMOSTAT SENSORS

	detected	not detected
intrusion	94.5%	6.5%
no intrusion	7.6%	92.4%

TABLE III. CONFUSION MATRIX FOR TI IOT DEVICES

	detected	not detected
intrusion	99.8%	0.2%
no intrusion	4.6%	95.4%

The results clearly show that the good performance of the proposed approach. Most of the attacks that were not detected were those where the amplitude changes was very close to the original values. The data for the TI IoT devices had hardly any noise so any deviation from the normal behaviour was detected as intrusion, which explains the good performance of the approach.

V. CONCLUSION

In this paper, we have proposed an Intrusion detection methodology for IoT embedded devices. The methodology is based on a hierarchical design in order to distribute the computational resources over the IoT devices and increase the methodology scalability. Our approach is based on observing the devices performance and its correlation to similar devices. Experimental results shows that the efficiency of the proposed approach for detecting suspicious activities.

In future we plan to investigate the application of this technique with more rich dataset in particular related to the manufacturing domain. Currently we have assumed that the data is consisting of continuous domains. Therefore, in future we would like to extend this technique to consider events and categorical data. Also, There are many variants of our approach that also deserve future investigation.

ACKNOWLEDGEMENT

The research leading to these results in part has received funding from the European Unions Horizon 2020 research and innovation programme under Grant Agreement No. 731558.

REFERENCES

[1] A. Mady, D. Mehta, D. M. Shila, and M. Boubekeur, “Towards resilient cyber security for embedded devices on internet,” 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT) (2016), pp. 12, Dec. 2016.

[2] V. Valdivia et al., “Sustainable building integrated energy test-bed,” Power Electronics for Distributed Generation Systems (PEDG), 2014 IEEE 5th International Symposium on, pp. 16, 2010.

 ISBN: 978-1-61208-585-2

[3] S. Haider, D. Manikantan Shila, and M. van Dijk, "Security agents for embedded intrusion detection," Circuit Cellar Magazine, Mar. 2015.

[4] B. Donohue, "Beware the thingbot," www.blog.kaspersky.com, Jan. 2014.

[5] "OWASP Internet of Things project," www.owasp.org/index.php, 2014.

[6] A. A. Cardenas, P. K. Manadhata, and S. P. Rajan, *Big data analytics for security*, IEEE Security & Privacy, pp. 74-76, Dec. 2013.

[7] A. Valdes and K. Skinner, *Adaptive, Model-Based Monitoring for Cyber Attack Detection*, Recent Advances in Intrusion Detection Volume 1907 of the series Lecture Notes in Computer Science pp 80-93, 2000.

[8] F. Pasqualetti, F. Dörfler, and F. Bullo, *Attack Detection and Identification in Cyber-Physical Systems–Part I: Models and Fundamental Limitations*, arXiv preprint arXiv:1202.6144, 2012.

[9] H. Fawzi, P. Tabuada, and S. Diggavi, *Secure Estimation and Control for Cyber-Physical Systems Under Adversarial Attacks*, Automatic Control, IEEE Transactions on, pp. 1454-1467, Jan. 2014.

[10] K. Paridari et al., *Cyber-Physical-Security Framework for Building Energy Management System*, In 2016 ACM/IEEE 7th International Conference on Cyber-Physical Systems (ICCPS), 2016.

[11] D. Evans, A. Nguyen-Tuong, J. Knight, *Effectiveness of moving target defenses*, Moving Target Defense, 2011.

[12] R. Mitchell and I.-R. Chen, "A survey of intrusion detection techniques for cyber-physical systems," ACM Comput. Surv., vol. 46, no. 4, pp. 55:1–55:29, Mar. 2014.

[13] C. Varun, A. Banerjee, and V. Kumar, *Anomaly detection: A survey*, ACM computing surveys (CSUR), 2009.

[14] D. Cheboli, *Anomaly Detection of Time-Series*, Phd Thesis, 2010.

 ISBN: 978-1-61208-585-2