Securing Manufacturing Using Blockchain

Z. Jadidi, A. Dorri, R. Jurdak, and C. Fidge
Queensland University of Technology (QUT), Cyber Security Cooperative Research Centre

***Abstract*—Due to the rise of Industrial Control Systems (ICSs) cyber-attacks in the recent decade, various security frameworks have been designed for anomaly detection. While advanced ICS attacks use sequential phases to launch their final attacks, existing anomaly detection methods can only monitor a single source of data. However, analysis of multiple security data could provide more comprehensive and system-wide anomaly detection in industrial networks. In this paper, we present an anomaly detection framework for ICSs that consists of two stages: i) blockchain-based log management where the logs of ICS devices are collected in a secure and distributed manner, and ii) multi-source anomaly detection where the blockchain logs are analysed using multi-source deep learning which in turn provides a system wide anomaly detection method. We validated our framework using two ICS datasets: a factory automation dataset and a Secure Water Treatment (SWaT) dataset. These datasets contain physical and network level normal and abnormal traffic. The performance of our new framework is compared with single-source machine learning methods. The precision of our framework is 95% which is comparable with single-source anomaly detectors. However, multi-source analysis is more robust because it can detect anomalies from multiple sources simultaneously, while achieving comparable precision for each of the sources.**

***Index Terms*—Industrial control systems, anomaly detection, log management, blockchain, deep learning, sequence classification**

I. Introduction

With the advances in the Internet and emergence of new technologies such as Internet of Things (IoT), traditional industrial control systems (ICSs) are shifting from isolated sites to interconnected networks. The high degree of connectivity increases the security risks in ICSs as malicious nodes may access the devices and cause malfunctions. The existing ICSs are a collection of interconnected industrial legacy systems (operational technology (OT) networks) integrated with information technology (IT) networks [1]. ICS networks contain heterogeneous interconnected components such as: remote terminal units (RTU), programmable logic controllers (PLC), historian servers, and human-machine interfaces (HMI) [2]. Studies show that the number of documented attacks targeting these ICS infrastructures by bypassing IT security has increased dramatically in recent years [3]. This has raised concerns for anomaly detection in ICS networks.

One of the fundamental methods proposed in the literature to detect anomalies is to analyse the logs generated by ICS devices, that potentially reflect their behaviour and thus the actions that happened in the network, using machine learning algorithms. Different supervised and unsupervised machine learning (ML) methods have been employed to detect anomalous behaviour in logs generated by ICS networks, hosts or sensors [4]. These methods have improved the accuracy of anomaly detection in a single type of ICS logs. However, they only monitor a specific component of a network and they cannot scale to large networks with multiple sources of log generation.

A ML algorithm uses logs as input to decide on anomalies, thus it is highly critical to ensure the security and integrity of the log files and to prevent logs from being deleted accidentally or deliberately. The integrity of the logs prevents malicious nodes from changing the content of the log once it is generated by the devices. Malicious nodes may attempt to remove their trace after conducting an attack by removing the logs produced by the device. The logs are normally stored in the device itself or may be sent to a centralised cloud server. Even if the logs are sent to the cloud, the attacker can remove the logs by compromising the cloud server. In some cases, the logs might be removed accidentally by the employees of a company which in turn removes the historical behaviour of the devices that complicates the attack detection process.

Conventional logging systems rely on the logs produced by either a single device or devices in the same site. However, a manufacturer might have multiple sites in geographically distributed locations which all might be the target of the same malicious behaviour. Sharing log information of all the devices in distributed locations facilitates detection of anomalous behaviours. Thus, logging in manufacturing demands a distributed, secure, auditable and immutable solution.

Blockchain can potentially address the outlined challenges due to its salient features which includes decentralisation, security, auditability and immutability [5]. In blockchain all transactions, i.e., communications between participants, are broadcast and verified by all participants. Particular nodes, known as validators, collect transactions and store them in a blockchain in the form of blocks by following a consensus algorithm. The latter ensures blockchain security and establishes trust in the state of the ledger among untrusted participants. The transactions and blocks are publicly available to the participants which introduces high auditability. Each block is chained to the previous block in the blockchain by storing the hash of the previous block which in turn makes it impossible to modify the content of the previously stored information in the blockchain and thus introduces immutability.

Ahmed et al. [6] designed a blockchain-based solution for log management which in turn demonstrated the feasibility of storing log information in a blockchain. Similarly, Schorradt et al. [7] studied the feasibility of using blockchain to store logging information for ICSs. The existing solutions for log management using a blockchain only consider sharing data

978-1-6654-0392-4/20/$31.00 ©2020 IEEE
DOI 10.1109/TrustCom50675.2020.00262

neglecting how to detect anomalous behaviour based on the collected data. Motivated by the above discussions, in this paper we develop an anomaly detection framework for ICSs. Our new framework comprises two main phases which are:

- *Logging* that stores the log information of the geographically dispersed devices in a blockchain.
- *Anomaly detection* where a particular node in the network runs a multi-source deep learning algorithm to detect anomalies in logs received from multiple sources.

The multi-source anomaly detector transforms the heterogeneous inputs into the same format. The output of the deep learning identifies the input source generating the anomalous behaviour. The multi-source deep learning uses Long Short Term Memory (LSTM) networks to analyse input sequences from ICS sources [8]–[10]. To reduce the delay in detecting anomalies, anomaly detection is run at two levels. The first level is in each site where the site manager runs the deep learning algorithm on the logs collected from local devices to find anomalies. Site managers in the same organisation may share their knowledge or information about any malicious activity which in turn facilitates anomaly detection in all sites. In the second level, called an organisation level, a computationally capable device runs deep learning on all logs in the blockchain to detect anomalies which in turn facilitates detecting more complicated malicious behaviours.

The rest of this paper is organised as follows. Section II explains background and related works. Section III discusses the threat model used in our study. Section IV describes the anomaly detection framework proposed in this paper. Evaluation datasets are explained in Section V. Section VI discusses the experimental results of the presented framework. Section VII concludes the paper.

II. Background and Related Works

This study presents an anomaly detection framework for the ICS architecture provided in the Purdue model [11]–[13]. In this section, we provide a background discussion on existing anomaly detection methods in ICSs in Section II-A and using blockchain in log management in Section II-B.

A. Anomaly Detection

Analysis of ICS logs has been investigated in the literature as one of the fundamental methods to detect anomalies. Feng and Chana [15] combined the content of the network packets transmitted between ICS components and their time-series data to detect anomalies. This combinational method can predict the future behaviour of the data created from a gas pipeline SCADA system and detect anomalies with high performance.

Machine learning methods have been applied to anomaly detection in ICS networks [16]. The detection rates of Deep Neural Networks (DNN) and one-class Support Vector Machines (SVM) were compared using a Secure Water Treatment (SWaT) dataset captured from a simulated fully operational raw water purification plant. DNN generates fewer false positives and better F1 measure than SVM. Kravchik and Shabtai [4] applied Convolutional Neural Networks (CNN) to detect anomalies which is then evaluated using SWaT dataset. The statistical deviation of the predicted value from the observed value were measured to detect anomalies. The results show the efficiency of 1D convolutional networks in predicting time-series prediction tasks and in detecting anomalies in ICS networks.

Stateful anomaly detection based on the cumulative sum (CUSUM) of residuals was proposed by Ghaeini et al. [17] to analyse physical process logs. Their proposed method uses state dependent detection thresholds to control an attacker trying to manipulate ICS data processes. It has three key features: i) it can predict system state from historical data of the network, ii) it provides cumulative sum of residuals for monitoring ICSs, and iii) it provides state-aware anomaly detection. The evaluation results prove that the proposed method achieves less time-to-detect of attacks and fewer false alarms compared to other methods.

While these ML-based anomaly detection methods focus on a single type of input data, our multi-source deep learning can detect anomalies in data received from multiple heterogeneous sources. The existing works in anomaly detection assume that the log data is transmitted securely and reliably without any modification of the log content, while in real-world scenarios, the attacker may attempt to remove its trace or alter the logs to cover its track. Most of the existing works assume the devices that generate logs are in the same site while in most manufacturers the devices might be in geographically dispersed locations [34]. Third parties may also be involved in manufacturing, e.g., a service center that services devices. It is critical to maintain a comprehensive log. The log information contains privacy-sensitive information about the manufacturer and thus should be kept private from third-parties. We employ blockchain technology to address the outlined challenges. The immutability of the blockchain makes it impossible for malicious nodes to alter the logs. The anonymity offered by the blockchain enhances the privacy of the companies. The transactions are sealed using asymmetric encryption which protects data from tampering during transmission. We discuss the existing blockchain-based log management systems next.

B. Blockchain in Log Management

In this section, we study log management systems that are based on blockchain. Ahmad at el. [6] proposed a blockchain-based logging framework that employs a private blockchain where only authorised nodes are allowed to participate in storing logs. Byzantine Fault Tolerance (BFT) is employed as the underlying consensus algorithm to reduce the overheads associated with storing new blocks in the blockchain.

Pourmajidi and Miranskyy [30] proposed Logchain, a blockchain-assisted log storage. Logchain consists of two tiers which are a: i) Circulated chain that comprises of a circulated blockchain that has a genesis block, i.e., the first block in the ledger, and termination block that is the last block in the ledger and is linked back to the genesis block forming a circulated chain, and ii) Super chain that is the main blockchain that contains the hash of the circulated blockchain which increases the throughput of the blockchain. In a similar attempt, Tomescu and Devadas [31] proposed Catena, where a log of a number of events is stored in a blockchain to improve throughput. A logging server receives the events and generates a log once the number of events reaches a pre-defined value.

Castaldo and Cinque [32] proposed a blockchain-based logging system to store the logs related to health data exchange between countries. The information from the health system is sent to a server that exchanges the data with the requestee while storing the corresponding log in the blockchain. Rane and Dixit [33] proposed a framework to store log data of cloud sources in the blockchain. A node controller collects log information from different sources and sends the encrypted version to the blockchain. The intended receiver can read data after decrypting with the associated private key.

While the existing works in blockchain-based logging systems focus only on storing information in the blockchain, our solution provides a comprehensive framework to store and process log information to detect anomalous behaviour in ICSs. Most of the existing solutions rely on conventional blockchains, while such blockchain management involves packet and computational overheads which are far beyond the capabilities of the devices in manufacturing industry.

III. Threat Model

In this section, we discuss our assumed threat model. ICS devices accept communications from the Internet as the administrators may remotely control, monitor, and update the devices. Only devices authorised by ICS admins can participate in the blockchain. We assume that an attacker cannot access the devices physically and thus the data generated by the device is accurate. Logging information stored in the blockchain is not visible to all participants and just authorised nodes have read/write permissions. Secure asymmetric encryption algorithms are employed which cannot be compromised by malicious nodes. It is assumed that a device generates a log corresponding to an event immediately after conducting the event. The log is broadcast to the neighboring devices in the same site. The device logs in each site are forwarded to a multi-source deep learning detector to monitor all devices and detect anomalies. The output of the anomaly detector will be securely shared with other sites using the blockchain.

IV. Anomaly detection in ICSs

In this section, we outline the details of our new anomaly detection framework. The manufacturer establishes a private blockchain and authorises the devices to join the blockchain by generating a genesis transaction, i.e., the first transaction in a ledger, for each device that can be used by the device to chain its following transactions. Figure 1 represents a high level view of our framework. The participants in the private blockchain have different read/write permissions in a way that only validators, i.e., the group of nodes that store the blocks in the blockchain, have write permission. Byzantine Fault Tolerance (BFT) is employed as the underlying consensus algorithm. To reduce the storage overhead of the blockchain,

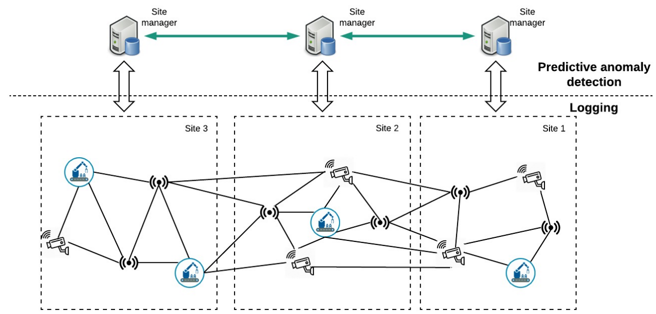

Fig. 1. An overview of our framework.

it is assumed that only nodes with enough storage capacity will store the blockchain. The validators are authorised by the manufacturer and can be the site managers. It is possible that multiple third parties are involved in the private chain of a manufacturer to store their logging information. As the log information is privacy-sensitive to the manufacturer, it is critical to ensure that only authorised nodes can read and access such data. The manufacturer announces a public key (PK), known as PK_{data} that shall be used by the devices to encrypt log data. PK_{data} and the associated private key are known to the entities that need to read log data as further discussed in Section IV-B.

Blockchain is only employed as a shared trusted database to store the log of the devices while the rest of the communications of the devices are happening in an off-chain channel, e.g., through the Internet. Our framework thus consists of two phases which are logging, where devices store logs in the blockchain, and predictive anomaly detection, where logs are analysed to detect malicious activities which are discussed in greater detail below.

A. Logging

In this phase, the devices store their logs in the private chain. Only authorised devices that are installed by the manufacturer can participate and share information with other parties. Thus, during the bootstrapping the manufacturer must generate a genesis transaction for each device that is structured as follows:

$T\_ID \,||\, PK \,||\, PK_{data}$

where *T_ID* represents the identity of the transaction which is essentially the hash of the transaction content, *PK* is the public key employed by the device to generate the next transaction, and PK_{data} is the public key that must be used by the device to encrypt data. This ensures data confidentiallity as only the intended receiver can read the data.

When an event occurs, the device generates a *Log Store (LS)* transaction that is structured as follows:

$T\_ID \,||\, P\_T\_ID \,||\, Log \,||\, PK \,||\, Signature$

where *P_T_ID* is the hash of the previous transaction generated by the device that ensures only the authorised nodes can participate in the blockchain. *Log* is the log of the device encrypted with PK_{data}. The last two fields are the *PK* of the device and the corresponding signature. The device broadcasts the *LS* transaction to its neighboring nodes to be stored in the blockchain.

The immutability offered by the blockchain makes it impossible for the malicious entities to modify previously stored logs which in turn increases the security of the framework. Having discussed the logging process, we next discuss predictive anomaly detection.

B. Predictive anomaly detection

The main objective of this phase is for the site manager to detect anomalies. Our framework introduces two levels of anomaly detection which are the site and organisation level. In each site a device is dedicated to run the deep learning algorithm, referred to as the site manager, to detect anomalies. Details of the deep learning algorithm are introduced later in this section. If the site manager detects any anomaly or suspicious behaviour from a particular user, it informs other site managers in the private chain by generating an Anomaly Alert (t^{aa}) that is structured as follows:

T_ID || *P_T_ID* || *M_PK* || *Log* || *PK* || *Signature*

where *M_PK* is the *PK* of the suspicious node. The malicious node may change their *PK* in different sites which makes it challenging to detect anomalies based only on the *PK* of the malicious node. Thus, the site manager stores the pattern of an anomaly in the *Log* field which assists the site managers to detect the same pattern of actions.

An anomaly might involve devices in different sites that makes it challenging for the site managers to detect the attack as they analyse the log data of their corresponding devices. As outlined earlier, the logs are stored in the blockchain, thus any node that participates in the blockchain and knows the decryption key of the data can access the logs. At the organisation level, the manufacturer dedicates a device to run a deep learning algorithm on top of the blockchain data which increases the chance of detecting attacks that involve multiple devices in different sites.

In our framework, a multi-source deep neural network (MS-DNN) is deployed to solve the problem of anomaly detection in data received from multiple ICS devices over different Purdue layers [24]. Our MS-DNN solution uses sequence classification to detect anomalies. Figure 2 shows the architecture of ICS anomaly detection using a MS-DNN. MS-DNN is a supervised classifier which needs a labeled dataset. These labels are shown as Target in Figure 2. The presented MS-DNN solution has multiple inputs which are transformed into the same format. These pre-processed inputs will be combined to generate an understandable input for MS-DNN. The combined inputs will be a cell array maintaining the historical data of all inputs. The output of MS-DNN is the number of device logs being monitored by the anomaly detector. MS-DNN is a classifier that detect anomalies in a site level and identifies the ICS device generating the anomalies [21], [22].

Deep learning neural networks are an end-to-end solution for network traffic analysis. They can extract features from raw data and use the extracted features for classification tasks. In our MS-DNN, a bidirectional LSTM (BLSTM) network is used to train a deep neural network to make predictions based on sequence data received from ICS devices, classify the sequences into devices, and identify whether the device behaves normally. The output of the deep learning corresponds to the number of end devices. The BLSTM has inputs from different sources and it has 100 hidden units. A BLSTM uses two different LSTM networks trained for sequential inputs, and hence, it could improve the performance of LSTM in classification processes. In this paper, the number of hidden units was based on the number of input features received from multiple sources, and it was obtained by trial and error. In our MS-DNN, the LSTM layer is connected to a fully connected layer followed by a softmax layer and a classification layer [26], [27].

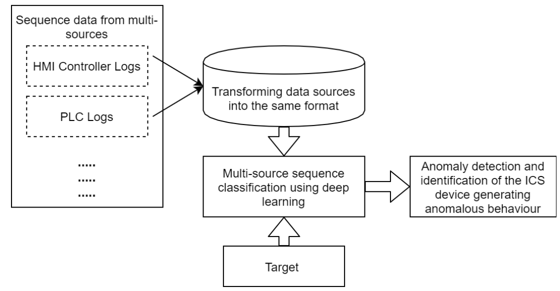

Fig. 2. The architecture of anomaly detection in an ICS network using multi-source deep learning

The performance of the MS-DNN was evaluated based on two publicly available ICS datatsets, and its accuracy and precision were compared with other classifiers [25]. Two MS-DNNs were separately used for anomaly detection in site manager levels. However, anomaly detection at an organisation level is not evaluated in this paper.

Once the manufacturer or the site manager detects an anomaly, they shall inform the "incident response team" (IRT) to take further actions and secure the network. To this aim, they generate a t^{aa} as discussed earlier in this section and include the relevant information about the attack. On receipt of a t^{aa} the IRT first verifies the signature of the transaction generator to ensure an authorised device has generated the same. Then, the data about the anomaly is analysed to decide on proper actions. The decision processing in this step is beyond the scope of this paper. The IRT uses the blockchain information to verify the log files and find the liable party. Benefiting from the blockchain immutability, the IRT and other authorities can trust the log information stored in the blockchain.

V. Datasets

Two datasets were used to evaluate our MS-DNN solution. These datasets provide multiple ICS data sources required for training of the MS-DNN.

- The first dataset (SWaT dataset) is of a small scale industrial water treatment process, which is a six stage system. The dataset was generated by the iTrust cyber security research center [28]. The communication protocol used for the automation was Modbus. The attacks on the systems were spoofing and man-in-the-middle. The SWaT dataset includes physical data collected from sensors and actuators and network data.
- The second dataset (factory automation dataset) was generated from a three-stage, laboratory-based factory automation process consisting of a conveyor belt sorting system, a water tank system, and a pressure vessel system. The process is automated using SIEMENS PLCs. Thus, the communication protocol is S7comm [29]. All the attacks on the system were flooding attacks from an attacker machine on the same network.

These datasets provide network traffic captured in two locations: 1) Packet capture (Pcap) logs captured from ICS devices like PLCs, and 2) Pcap files captured from a switch connected to the attacker's network. The two datasets in this paper demonstrate two ICS sites, A and B, and they are used to evaluate the performance of our MS-DNN based anomaly detection at the site management levels.

VI. Evaluations and Results

The MS-DNN uses BLSTM to classify sequence data received from ICS devices. The BLSTM algorithm uses sequence data as input and makes predictions based on the time steps of the sequence data. The SWaT dataset contains time series for two data sources from physical devices and network traffic. Each sequence has a number of features, separately extracted from each source, and it varies in length. The maximum number of features is defined as the required number of features in all input sequences in deep learning. Twelve TCP/IP packet header features were manually extracted from network Pcap files. Therefore, input sequences were defined with twelve features which show the maximum number of features in our datasets.

The training dataset contains 2000 samples and the testing dataset has 1000 samples. The output detects anomalies in input sequences and identifies the device generating the anomalous behaviour [23]. Figure 3 shows the accuracy of our MS-DNN with different iterations in the training phase in the SWaT dataset. As this figure illustrates, the accuracy is 100% after 1000 iterations.

The F1 score is the performance metric used in this paper, as shown in Eq. 1 [4]. The MS-DNN was separately evaluated using the SWaT dataset and the Factory automation dataset, and its results are shown in Table I.

$$F_1 = 2.\frac{precision \cdot recall}{precision + recall} \quad (1)$$

Table I shows the experimental results of the implemented MS-DNN in comparison with other studies [4]. As shown, while the MS-DNN analyses multiple inputs, the precision and F1 metrics are still similar to other machine learning based

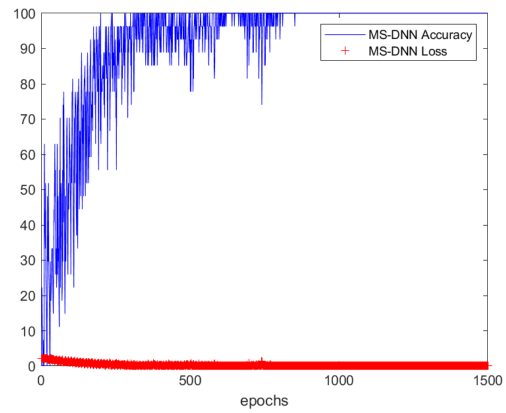

Fig. 3. MS-DNN Training accuracy and errors with the SWaT dataset

TABLE I
VALIDATION RESULTS FOR MS-DNN

Dataset	Method	Precision	Recall	F1	Accuracy
Factory automation dataset	MS-DNN	0.96	0.80	0.87	0.97
SWaT	MS-DNN	0.95	0.82	0.88	0.95
SWaT	SVM [4]	0.93	0.70	0.80	...
SWaT	1D CNN combined records [4]	0.97	0.79	0.87	...
SWaT	1D CNN ensembled records [4]	0.87	0.85	0.86	...

anomaly detectors with a single-source input. According to the Purdue model, ICS networks have a five-layer structure with different types of devices. For anomaly detection using single source analysis, a machine learning method is required for each type of data. However, the MS-DNN is trained by the historical behaviour of multiple ICS devices. Sequential dependencies between input data points are used for sequential classification tasks. The LSTM in the MS-DNN can learn long term dependencies in sequences and this helps to detect the anomalous device in input sources. Therefore, the MS-DNN receives sequence data from input sources and classifies the data into normal / abnormal classes. It also identifies the source generating the anomalies.

VII. Conclusion and Future works

In this paper, we provided an anomaly detection framework for ICS networks. This framework uses blockchain to securely store device logs. In addition, multi-source deep learning was used to detect anomalies at two levels, the site and organisation levels. Site-level anomaly detection was implemented to analyse device logs in each site. Then, the organisation level shared the knowledge of malicious activity in each site with other sites. The performance of our new anomaly detection framework was evaluated using two ICS datasets and the results showed high precision which was comparable with single-source anomaly detection methods.

Once a manufacturer detects anomalous behaviour, it can inform other manufacturers of the detected attack through a threat intelligence layer. In this layer, all manufacturers and other involved parties communicate to share anomaly signs. A public blockchain can be employed in this layer to increase trust among the participants. This layer also enables participants to generate attack signatures and share which in turn enhances the security of the framework and reduces the delay involved in generating signatures. This method can also be used to verify and authenticate communications as only registered devices here can participate in the main chain [18]–[20]. These are the challenges which will be investigated in our future work.

ACKNOWLEDGMENT

The authors acknowledge the support of the Commonwealth of Australia and Cybersecurity Research Centre Limited.

REFERENCES

[1] Hunting and responding to industrial intrusions (2017). [online] Available at: https://www.dragos.com/wp-content/uploads/2017-Review-Hunting-and-Responding-to-Industrial-Intrusions.pdf.
[2] Tuptuk, N., & Hailes, S. (2018). Security of smart manufacturing systems. Journal of manufacturing systems, 47, 93-106.
[3] Hu, Y., Yang, A., Li, H., Sun, Y., & Sun, L., A survey of intrusion detection on industrial control systems. International Journal of Distributed Sensor Networks. 1-14 (2018).
[4] Kravchik, M., & Shabtai, A. (2018, January). Detecting cyber attacks in industrial control systems using convolutional neural networks. In Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy (pp. 72-83).
[5] Dorri, A., Kanhere, S. S., Jurdak, R., & Gauravaram, P. (2019). LSB: A Lightweight Scalable Blockchain for IoT security and anonymity. Journal of Parallel and Distributed Computing, 134, 180-197.
[6] Ahmad, A., Saad, M., Bassiouni, M., & Mohaisen, A. (2018, November). Towards blockchain-driven, secure and transparent audit logs. In Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (pp. 443-448).
[7] Schorradt, S., Bajramovic, E., & Freiling, F. (2019, November). On the feasibility of secure logging for industrial control systems using blockchain. In Proceedings of the Third Central European Cybersecurity Conference (pp. 1-6).
[8] Ouyang, W., Chu, X., & Wang, X. (2014). Multi-source deep learning for human pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2329-2336).
[9] Jurgovsky, J., Granitzer, M., Ziegler, K., Calabretto, S., Portier, P. E., He-Guelton, L., & Caelen, O. (2018). Sequence classification for credit-card fraud detection. Expert Systems with Applications, 100, 234-245.
[10] Kolosnjaji, B., Zarras, A., Webster, G., & Eckert, C. (2016, December). Deep learning for classification of malware system call sequences. In Australasian Joint Conference on Artificial Intelligence (pp. 137-149). Springer, Cham.
[11] SANS Institute: Reading Room - Industrial Control Systems / SCADA, https://www.sans.org/reading-room/whitepapers/ICS/secure-architecture-industrial-control-systems-36327
[12] Automation, R. (2011). Converged Plantwide Ethernet (CPwE) Design and Implementation Guide.".
[13] NCCIC (2016). Recommended practice: Improving industrial control systems cybersecurity with defence-in-depth strategies. US-CERT Defence In Depth, https://www.us-cert.gov/sites/default/files/recommended_practices/NCCIC_ICS-CERT_Defense_in_Depth_2016_S508C.pdf
[14] https://collaborate.mitre.org/attackics/index.php/Main_Page
[15] Feng, C., Li, T., & Chana, D. (2017, June). Multi-level anomaly detection in industrial control systems via package signatures and LSTM networks. In 2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN) (pp. 261-272). IEEE.
[16] Inoue, J., Yamagata, Y., Chen, Y., Poskitt, C. M., & Sun, J. (2017, November). Anomaly detection for a water treatment system using unsupervised machine learning. In 2017 IEEE International Conference on Data Mining Workshops (ICDMW) (pp. 1058-1065). IEEE.
[17] Ghaeini, H. R., Antonioli, D., Brasser, F., Sadeghi, A. R., & Tippenhauer, N. O. (2018, April). State-aware anomaly detection for industrial control systems. In Proceedings of the 33rd Annual ACM Symposium on Applied Computing (pp. 1620-1628).
[18] Tounsi, W., & Rais, H. (2018). A survey on technical threat intelligence in the age of sophisticated cyber attacks. Computers & security, 72, 212-233.
[19] Wagner, C., Dulaunoy, A., Wagener, G., & Iklody, A. (2016, October). MISP: The design and implementation of a collaborative threat intelligence sharing platform. In Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security (pp. 49-56).
[20] Riesco, R., Larriva-Novo, X., & Villagra, V. A. (2019). Cybersecurity threat intelligence knowledge exchange based on blockchain. Telecommunication Systems, 1-30.
[21] Liu, L., Chen, C., Zhang, J., De Vel, O., & Xiang, Y. (2019). Insider threat identification using the simultaneous neural learning of multi-source logs. IEEE Access, 7, 183162-183176.
[22] Crammer, K., Kearns, M., & Wortman, J. (2008). Learning from multiple sources. Journal of Machine Learning Research, 9(Aug), 1757-1774.
[23] Jurgovsky, J., Granitzer, M., Ziegler, K., Calabretto, S., Portier, P. E., He-Guelton, L., & Caelen, O. (2018). Sequence classification for credit-card fraud detection. Expert Systems with Applications, 100, 234-245.
[24] Li, J., Wu, W., Xue, D., & Gao, P. (2019). Multi-Source Deep Transfer Neural Network Algorithm. Sensors, 19(18), p. 3992.
[25] Rizzo, R., Fiannaca, A., La Rosa, M., & Urso, A. (2015, September). A deep learning approach to DNA sequence classification. In International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics (pp. 129-140). Springer, Cham.
[26] Yildirim, Ö. (2018). A novel wavelet sequence based on deep bidirectional LSTM network model for ECG signal classification. Computers in biology and medicine, 96, 189-202.
[27] au.mathworks.com. (n.d.). Sequence Classification Using Deep Learning - MATLAB & Simulink - MathWorks Australia. [online] Available at: https://au.mathworks.com/help/deeplearning/ug/classify-sequence-data-using-lstm-networks.html [Accessed 26 Jul. 2020].
[28] Goh, J., Adepu, S., Junejo, K. N., & Mathur, A., A Dataset to Support Research in the Design of Secure Water Treatment Systems. In International Conference on Critical Information Infrastructures Security. 88-99 (2017)
[29] Myers, D., Suriadi, S., Radke, K., & Foo, E. (2018). Anomaly detection for industrial control systems using process mining. Computers & Security, 78, 103-125.
[30] Pourmajidi, W., & Miranskyy, A. (2018, July). Logchain: blockchain-assisted log storage. In 2018 IEEE 11th International Conference on Cloud Computing (CLOUD) (pp. 978-982). IEEE.
[31] Tomescu, A., & Devadas, S. (2017, May). Catena: Efficient non-equivocation via bitcoin. In 2017 IEEE Symposium on Security and Privacy (SP) (pp. 393-409). IEEE.
[32] Castaldo, L., & Cinque, V. (2018, February). Blockchain-based logging for the cross-border exchange of eHealth data in Europe. In International ISCIS Security Workshop (pp. 46-56). Springer, Cham.
[33] Rane, S., & Dixit, A. (2019, January). BlockSLaaS: Blockchain assisted secure logging-as-a-service for cloud forensics. In International Conference on Security & Privacy (pp. 77-88). Springer, Singapore.
[34] Drias, Z., Serhrouchni, A., & Vogel, O. (2015, August). Analysis of cyber security for industrial control systems. In 2015 International Conference on Cyber Security of Smart Cities, Industrial Control System and Communications (SSIC) (pp. 1-8). IEEE.