Industrial Control System Defence: Debugging ICS Maintenance Network Traffic

Angela Smith, Adam Wedgbury, Philippe Biondi, Hugh Soulsby, Kevin Jones
Cyber Security Architecture, Innovation and Scouting
Airbus, Quadrant House, Celtic Springs Business Park, Duffryn, Newport NP10 8FZ
www.airbus.com/cyber-innovation
angela.smith@airbus.com

Industrial Control System (ICS) third-party maintenance introduces security risk into an organisation, as access is granted for performance of named maintenance tasks on industrial equipment, but there is currently no fine-grained way to monitor the activity. This paper applies Machine Learning to ICS network traffic, in order to alert operational staff to unauthorised activity.

The work describes a method for identifying deviations, by characterising network traffic purpose, and applying software to dissect, learn and monitor maintenance traffic, then presenting results in a chart.

Network Security, Industrial Control System, Machine Learning, Maintenance, Cyber Defence

1. INTRODUCTION

Detection of unwanted behaviour in information technology is a required activity for legal and financial reasons, and this applies to industrial environments as much as in traditional information technology environments. An additional common risk in industrial networks is the need to permit remote third-party maintainers access to perform periodic maintenance. This raises the possibility that the maintainer might trigger activity which has adverse effects; it is desirable to mitigate this risk through monitoring and alerts.

There have been a number of efforts to identify anomalous activity using machine learning of network traffic, for example Dada et al. (2017), Feng et al. (2017) and Wang et al. (2017). This approach has been applied primarily to common and open protocols, often for intrusion detection, but the authors believe it is equally applicable to proprietary and niche industrial protocols such as used in ICS, and for another purpose: maintenance supervision.

This paper describes a series of development and experimentation activities which were conducted to determine whether this technique could help protect industrial control systems during remote maintenance by third-party organisations. It is organised as follows: brief Background and Methodology sections, a discussion of relevant Prior Work, details of the experiment and findings, a discussion of future work, and conclusions drawn.

2. BACKGROUND

Detection of unwanted behaviour in IT networks is provided by Intrusion Detection Systems (IDS), which can be host-based (HIDS) or network-based (NIDS); they analyse logs, data and processes for anomalous behaviour, and raise alerts.

Industrial Control System (ICS) networks have additional requirements: the Human-Machine Interface (HMI) controlling the ICS may have trouble running HIDS because of the hardware's age which is often old, since replacement would mean downtime and costs.

There are also protocol-related problems, in that ICS network protocols are often proprietary, and commonly have bit-oriented elements which are not necessarily obvious. Additionally, ICS are very sensitive to network latency, so monitoring must minimise delays.

Where the "intruder" is authenticated into the network and has authorisation to perform maintenance upon the ICS, and where most activity on the ICS is acceptable, identifying anomalous activity has to depend on knowing what is normal in the specific context. Machine Learning (ML) is a potentially useful approach to this. ML has been used with some success in characterising activity and identifying anomalies in traditional IT networks, as discussed by Dada et al. (2017).

3. METHODOLOGY

The design science research paradigm has been adopted in this research, as the design of the prototype is intended to result in new knowledge which can be used in the development of future industrial intrusion detection systems.

The approach is positivist with quantitative methods, in that the ability to apply machine learning to protocols used in ICS maintenance, to detect anomalous behaviour will be assessed by numerical results arising from experiments.

4. PRIOR WORK

This section will discuss relevant existing knowledge including a brief discussion of ICS protocols; the application of machine learning in the networking context; organisational risk management and third-party maintenance risk.

4.1. ICS Protocols

There are many ICS network protocols used by maintainers to communicate with controllers during maintenance sessions, and these are often associated with particular vendors. For instance, Siemens has the S7 Communication proprietary protocol for Programmable Logic Controllers (PLC), OMRON has FINS for its PLCs, Tridium has FOX for building automation devices, Mitsubishi Electric has MELSEC-Q proprietary protocol for its devices, then there are the IEC 60870-5 suite for power and water installations, DNP3 for power installations, and the open and popular Modbus protocol.

ICS network protocols are often proprietary, with full documentation not in the public domain; the protocols are often binary, so text-based approaches will not work, and adherence to the protocol sometimes varies between device models, which may matter when trying to apply a standardised monitoring process across a diverse population which has evolved over many years.

Additionally, ICS are very sensitive to network latency because of real-time communication needs in their networks, so maintenance session monitoring must not introduce potentially damaging delays which interfere with normal operations.

4.2. Third-Party Maintenance

Third-party maintenance is intended to be a help to an organisation, but it could equally be a threat, although a vague one. It could be compared with submitting to surgery, in that it requires trust, creates vulnerability, and reduces control. As described

by Stewart et al. (2008) in a comprehensive practitioner-oriented security study guide created for the Certified Information Systems Security Professional (CISSP):

Not all problems that an IT infrastructure will face have definitive countermeasures or are even a recognizable threat. There are numerous vulnerabilities against which there are no immediate or distinct threats and against such threats there are few countermeasures. Many of these vulnerabilities lack direct-effect countermeasures, or the deployment of available countermeasures offers little in risk reduction. (Stewart et al., 2008, p545)

Such is the situation for remote third-party maintenance: it appears to be in the organisation's interest, but adverse effects could arise, and there is no granular monitoring available to alert when unexpected and unauthorised activities take place. Stewart goes on to list and describe threat categories, and it is apparent that several of those threats could come about via third-party maintenance, including mistakes, malicious destructive acts, and accidents. Countermeasures for such threats is central to this work.

In considering the nature of third-party maintenance, it is useful to consider temporal and strategic/tactical aspects, as the environment and staff attitudes and expectations might be affected by these.

Third-party maintenance can be strategic; that is, routine and pro-active, such as periodic processes needed to ensure industrial installations continue to function efficiently. For example, Steinke and Rickel (2011) describes considerations when choosing industrial maintainers for the automotive field, listing likely functions as comprehensively testing the industrial equipment, or help organisations exploit an opportunity, such as upgrading software to the latest build (potentially enabling new features), patch newly discovered bugs, improve communications protocol throughput, or provide better control functionality.

Strategic third-party maintenance could also include improving operations by replacing low capacity production equipment in response to rising demand. In these cases, there will be adequate time to plan the production outage to enable the work, and management pressure will be minimised, so staff oversight of the maintenance activity will be optimal. That said, there will still be pressure to complete within the agreed maintenance window.

Third-party maintenance can also be tactical; that is, non-routine and reactive, such as dealing with an adverse incident like equipment failure or degradation. In these cases, there will be less time to plan a production outage, and proper oversight risks being jeopardised by management pressure

and conflicting priorities to identify the immediate cause, preserve evidence of wrongdoing, and restore operations.

In either case, granular monitoring and automated alerting to deviations from plan are extremely useful. In fact, monitoring, logging and automated alerting are vital elements to ensure proper security management of any information technology or industrial control system. Monitoring permits the capture of state and actions, logging preserves the information as evidence in case the need for investigation and/or legal action arises, and automated alerting reduces the delay before intervention can be applied, and therefore maximises the chance of damage limitation.

In addition, automated alerting may in some cases be linked to automatic remedial action, which will vary depending upon the context, such as ending a session and revoking access authorisation; triggering an audio-visual alert; instigating emergency stop, or triggering lock-down.

4.3. Machine Learning

There is considerable prior work applying machine learning to network protocols in general, to provide intrusion detection functionality. This approach has been used by Dada et al. (2017), who performed experiments to compare six machine learning algorithms for effectiveness in the network intrusion detection role, with the outcome that Logistic Model Tree Induction performed best out of the tested set, which also included Support Vector Machine (SVM). Wang et al. (2017) focused on network protocol identification, so that known (labelled) and unknown protocols could be identified. A range of machine learning algorithms were tested and compared for effectiveness, and determined that Lunex performed best in that particular scenario. None of these have an industrial protocol element, nor a third-party maintenance aspect.

There is evidence of some application of ML to ICS network traffic; for instance, Alves et al. (2018) discusses identifying the normal network pattern (anomaly-based detection) and attack signatures (signature-based detection), and detecting man-in-the-middle attacks. Alves et al. goes on to propose and develop an architecture with ML embedded as an Intrusion Prevention System (IPS) on a PLC.

Feng et al. (2017) applies a type of Recurrent Neural Network (RNN) called Long Short-Term Memory (LSTM) to ICS traffic using Modbus, to learn packet sequences of ICS traffic and provide an intrusion detection system (IDS). LSTM incorporates inter-packet dependency, therefore provides a temporal context, and the network traffic is between controllers and devices (e.g., actuators and sensors), rather than solely PLC programming

request messages.

Hadžiosmanović et al. (2012) considers the application of n-gram based algorithms to network traffic anomaly detection. The ICS protocol Modbus was tested using the Anagram algorithm, with excellent profiles for true and false positives. Overall, n-gram approaches find performance penalties when these algorithms are used for protocols with highly variable data payloads, and for networks with many different protocols or network nodes.

Hasan et al. (2017) takes a different approach from Feng, using a constraint-oriented protocol definition to define normal and identify anomalies. The test-bed environment is air traffic control, which has a parallel with ICS, in that the number of protocols present is limited, which helps when learning all traffic present. Hasan also incorporates a temporal aspect with inter-packet dependency.

Zhanwei and Zenghui (2019) considers the problem of ICS protocol data field tampering, as seen in man-in-the-middle attacks. Their work also creates a behaviour model of normal activity, to enable the detection of abnormal activity, thereby creating an anomaly-oriented IDS. The experiments use Modbus/TCP, an open protocol, in a part-simulated environment. The area of the packets learnt and monitored are data fields, and an aim was to learn allowed value ranges for particular functions. Good detection results were achieved at identifying when data field tampering occurred, attempting to set a value outside the 'normal' range, with false positive rates in single figures, and false negatives slightly higher.

Zhanwei's work does not, however, address the issue of authorised third-party maintenance, the authorised outsider, and activity which is authorised only occasionally. There is little existing research known in that area, and similarly, there is little existing work on network intrusion detection of proprietary ICS protocols with machine learning. This work aims to address this gap, specifically by investigating whether ML can be applied to ICS network traffic to identify anomalous activity during third-party maintenance.

4.4. Risk Management

Most organisations have duties to stakeholders, who may be shareholders, customers, suppliers, etc., to ensure appropriate risk management is conducted on their systems. Risk management is an ongoing iterative process which involves periodic reviews considering existing risks, their impact and their likelihood. A process of prioritisation enables the decision-making process about where funds must be allocated, whether it is a mitigation which reduces or eliminates the probability the risk will be realised, or alternatively reduces the potential

impact. Another option which may be applied is the transference of risk, for instance through insurance or contractual agreement.

This work sits in the category of reducing or eliminating the probability of realising the risk that third-party maintenance could lead to unwanted activity in an ICS, causing the owning organisation an adverse impact, be that financial, environmental, reputational, or other.

4.5. Third-Party Maintenance Risk

Maintenance includes work done to ensure the continued operation of the industrial hardware which is controlled by the ICS, such as periodic flushing, synchronising, and calibration routines. It also includes updates to the software loaded on the PLC. In either case, maintenance network traffic will pass through the industrial network to the PLC, and as such can be monitored.

Industrial equipment maintenance is a specialised task, which third-party remote organisations are often contracted to perform, because of the specialised tools, skills and knowledge required. Regular maintenance is usually conducted during a so-called maintenance window, and this activity includes manipulating, or programming, the controlling PLC.

Remote maintenance represents a security risk to all these organisations, as third-party organisations could accidentally cause a security incident by triggering undesirable and unapproved maintenance activities, or the maintainer organisation could be used by a hostile actor to attack the industrial organisation's systems.

Such a risk could cause material, financial or reputational damage to an organisation, and compromising an ICS could have significant consequences.

For instance, if just 0.1% of the UK production and manufacturing industry were incapacitated for one week, due to unwanted third-party maintenance activity, it could cause lost production to the value of over GBP 7.3 million (Office for National Statistics, 2015). In addition, there could be lost or damaged materials and industrial equipment, plus human injury or death, which can trigger legal and compensation costs, plus investigative costs.

However, the potential loss is even greater, as ICS are not only used in production and manufacturing. A specialised type of ICS is also used in Building Management Systems (BMS) to control and monitor many essential services including lighting, air conditioning, power, security and fire (Centre for the Protection of National Infrastructure, no date). This means that ICS are providing essential services to many buildings containing service companies, and the potential losses are more than five times those

in production.

A one-week loss of output affecting just 0.1% of the services sector could cost as much as GBP 38.5 million.

The risk to BMS is not theoretical. For example, a BMS hack of a Google building was reported in 2013. The culprits (cybersecurity experts) stated 'If Google can fall victim to an ICS attack, anyone can' (Lacey, 2013).

The area under risk can be widened further to include Critical National Infrastructure (CNI), as ICS are found in many systems essential to daily life, such as water treatment, power generation, and telecommunications systems (Shodan, no date).

Indeed, although not a remote maintenance attack, the principle of CNI vulnerability is demonstrated by the Stuxnet attack which targeted ICS in nuclear facilities, causing the destruction of key components. While it is not known whether intended by the deployers of Stuxnet, an additional consequence was the cost to Iran of a substantial amount of money (Moos, 2015). These consequences could be experienced by any other CNI providers around the world.

5. EXPERIMENT

This section describes the threat model considered, test-bed used, the method of operation of the prototype, the development undertaken, the live experiment details, findings, and application of the work.

5.1. Threat Model

Threats in scope for this experiment are unauthorised activities arising during third-party or remote maintenance sessions, whether as a result of mistake, miscommunication, misunderstanding, or even the maintainers themselves being compromised. In the latter case, malicious actors or malware could use the maintainer's access rights during a particular maintenance window to perform unauthorised activity. In any of these scenarios, the requests sent to a PLC would not match the requests associated with the approved activities. This would be detectable in network traffic traces, and should be apparent in the results of this behavioural monitoring experiment.

5.2. Test-Bed

Activities in this work made use of a test-bed consisting of a small scale water treatment ICS installation (see Figure 1). A Human Computer Interface controls the installation by sending messages to

Figure 1: Water Treatment Test Bed

a Programmable Logic Controller (PLC). The PLC opens valves for a variable period of time, causing each tank to drain into the next, before water is pumped back to the top tank.

The PLC equipment are OMRON modules, including an Ethernet Communications Unit which enables the HMI and PLC to communicate via Ethernet; the test-bed uses OMRON's proprietary automation protocol Factory Interface Network Service (FINS) over User Datagram Protocol (UDP). The control activity is cyclic, so the same sequence of FINS traffic is repeated periodically.

All traffic on the PLC/HMI switch ports is mirrored to the laptop's port, so it can monitor activity without affecting performance.

5.3. Operation

The prototype is intended to be used in three modes, which support the three phases needed for the ML, as follows:

1. Capture: for capturing 'good' traffic samples, and using them to create 'training data', which will be used to train a machine learning algorithm.
2. Training: for instantiating and training a machine learning algorithm to recognise 'good' data.
3. Work: for monitoring live traffic data, and using the trained machine learning algorithm to compare it with the learnt 'good' data, and plotting this in a graph, which also shows deviation alerts.

5.4. Initial Development

An initial prototype application was developed to capture FINS network traffic and train a classifier

upon it. The prototype was built using C++ and the dlib machine learning library (King, 2009) to provide a classifier, and the operating phases (capture, training, and monitoring).

In choosing a classifier, it was noted that Dada, which compared the effectiveness of machine learning techniques for intrusion detection, revealed that good performance was achieved by Logistic Model Tree Induction, Multi-Layer Perceptron Neural Networks, and Support Vector Machine (SVM) classification. In further efforts to select an optimal technique, Weka 3.8 (Frank et al., 2016) was used to assess a sample of the training data and obtain a recommendation; the recommended classifier was multi-class Support Vector Machine (SVM), so this was selected.

A review of OMRON documentation (OMRON, 2001) and (OMRON, 2009) shows there is some variation in the FINS Command Frame Format dependent upon the hardware and configuration in use. The FINS protocol fields are illustrated in Figure 2 and Figure 3. In the test-bed request messages are as follows (fields 1 byte, unless stated):

1. Header
 - (a) ICF (Information Control Field)
 - (b) RSV (Reserved)
 - (c) GCT (Gateways)
 - (d) DNA, NDA1, DNA2 (Destination Network, Node and Unit Addresses)
- 3 bytes total
 - (e) SNA, SDA1, SNA2 (Source Network, Node and Unit Addresses)
- 3 bytes total
 - (f) SID (Service Identifier)
2. Command
 - (a) MRC, SRC (Main and Sub Request Command Codes)
- 2 bytes total
3. Parameter/Data (2000 bytes max). Where the command is a read /write command, the PARAMETER/DATA begins:
 - (a) MEM (Memory Area Code affected)
 - (b) ADDR (Beginning Address in memory area)
- 3 bytes
 - (c) NUM (Number of items)
- 2 bytes

The initial prototype used the following fields: ICF, SID, and CMD (that is, both MRC and SRC).

As regards prototype phases, Capture performed traffic capture from the network, with FINS/UDP message identification being based upon the

Figure 2: Header, Command Fields in OMRON FINS protocol - adapted from (OMRON, 2009, Section 1-3)

header's ICF field. The packet was captured when the ICF bitwise field value was 0x80, 0x81, 0xC0 or 0xC1, indicating that the message was a request or response message (each with or without confirmatory reply).

Selected fields from the message (ICF, SID, CMD) were saved to file in the form of lines of comma separated variables, to act as training data.

The Training phase instantiated a classifier, fitted it to the training data in the Capture file, and persisted the classifier algorithm to file.

The Monitoring phase performed the following activities: captured packets; identified FINS messages; extracted the selected fields from each; asked the classifier to classify the extract, and then sent results to the console.

Testing of this initial prototype were conducted off-line, using a replayed PCAP file containing FINS traffic from the test-bed. During this testing, several weaknesses were observed:

- Creating sufficiently varying bad labelled data for training was difficult, as bad behaviour is anything not pre-labelled as good; this is a very large set in a repetitive simple ICS environment, and it is onerous to assess and label such large quantities of low-level data.
- It was complex to try alternative classifiers on the captured data, as this required considerable reprogramming of the C++ prototype to use alternative classes from the dlib machine learning library (King, 2009).
- A good packet might be considered bad in some contexts, so the prototype must learn traffic sequences, rather than individual packets.
- The FINS messages being assessed included both request and response messages, but response messages are superfluous to monitoring maintainer actions, and add bulk to the data which must be processed.

5.5. Re-Development

The weaknesses mentioned above were addressed by reprogramming the prototype, as follows: the

Python language and the Scikit-learn module (Pedregosa et al., 2011) were used in place of C++ and dlib-ml (King, 2009); from this module, the one-class SVM classifier were chosen for use, as this required only good samples to train the classifier; a capability was also added to support learning and monitoring of configurable-length packet sequences (described in more detail below), so each new packet triggered creation of a new sequence in both capture and monitoring phases, and finally, the command/request messages (ICF field value 0x80 or 0x81) were retained but the responses (0xC0 or 0xC1) were discarded in the capture and monitoring phases, so that all training and classification could be conducted on request messages only.

Development was conducted using the same pcap file as in earlier work, but once the new prototype was mature, testing was shifted to run on live traffic, and this revealed that the classification results were not consistently identifying good command sequences.

Further examination of the traffic revealed the SID field was varying more widely than anticipated, so the prototype could not consistently identify the control actions being triggered; further reading revealed the SID represents the process originating the transmission, whereas a subsequent field, the MEMORY AREA represents the memory area being read/written. The latter was suspected to be a more suitable feature for identifying undesirable command sequences.

This was confirmed with examination of live traffic, which revealed the memory area field was varying more in line with control actions, so the decision was taken to replace the SID with the MEMORY AREA in the capture and monitoring phases. Since the MEMORY AREA is outside the FINS header (see Figure 3), the minimum packet size used by the prototype needed to be extended.

In addition, a chart was added to visualise the traffic classification progress, so that the classification trend is apparent in context of the whole monitoring session. This chart presents a 100-term moving average of the classifications, in which the raw classifications are +1 for good/expected (also called 'in-lier'), and -1 for bad/anomalous (also called 'out-lier').

The reason for using a moving average is to reduce the 'data deluge' effect of too much data being available (a categorisation data point for every one of thousands of packets would not be humanly manageable), and consequently useful information being hard to pick out. It reduces the impact of occasional out-lier categorisations, and allows the recent trend to take precedence.

Figure 3: Parameters and Data Fields in OMRON FINS protocol - adapted from (OMRON, 2009, Section 1-3)

5.6. Sequences

The prototype learns and monitors packet sequences of length 2 (2-grams), rather than individual packets, each packet being paired with the subsequent package. This is illustrated in Table 1, which shows a small sample of training data. Data from one of the packets emphasised first in cyan; a sample sequence is emphasised next in green, and a complete record, including machine learning label, is emphasised last in red. As the prototype uses one-class SVM, only one label is used (1) to indicate inliers (permitted sequences).

Table 1: Sample of data extracted from FINS packets and labelled in the capture phase

Class	ICF	SID	MEM	ICF	SID	MEM
1	128	137	257	128	137	257
1	128	137	257	128	177	257
1	128	177	257	128	177	257
1	128	177	257	128	137	257
1	128	137	257	128	137	257
1	128	137	257	128	177	257
1	128	177	257	128	177	257
1	128	177	257	128	137	257
1	128	137	257	128	137	257
1	128	137	257	128	177	257
1	128	177	257	128	177	257
1	128	177	257	128	137	257

5.7. Live Experiments and Findings

In subsequent informal live experiments, the initial capture of training data encompassed the test-bed's cycle time of about a minute, which produced over 3000 FINS requests. Some initial

trials were conducted to determine the optimal command sequence length, first trying sequences of 10 commands, then sequences of 2 commands, in both the training and monitoring phases. The shorter sequence setting was found to provide more convincing detection of 'bad' (anomalous) activity, so this was retained.

The formal live tests then commenced. This involved training the classifier on the captured data, and then running a number of monitoring sessions, all including collection and classification of 500 FINS messages. Each session involved varying amounts of interference with the PLC programming, by using the HMI to override timer and valve settings in the test-bed, which triggers FINS write commands to the PLC. These interference activities are analogous to undesirable maintenance activity, or 'bad' commands.

The first session was conducted without any user

Figure 4: Monitoring without HMI interference

interference, and the resultant chart (see Figure 4) showed a fairly regular pattern with a moving average classification value around 0.9. This can be treated as a baseline for normal activity.

The second session included HMI interference late in the session, and the chart (see Figure 5) showed the same regularity as the baseline, until after 350 sequences, when the pattern is interrupted. The moving average shows a significant decline in the good/bad classification, approaching -0.4.

The third session included HMI interference early in the session, and the chart (see Figure 6) showed disruption to the regular pattern after only 150 sequences, with a decline in the good/bad moving average from 1 towards 0. After HMI interference

ceased, this recovered gradually towards 0.9.

5.7.1. Findings

These results indicate that the technique of applying SVM ML to FINS network traffic sequences can detect FINS commands which are unexpected for the learnt scenario. This indicates the potential feasibility of out-of-band network monitoring and alerting to anomalous FINS activity, during regular maintenance sessions. This work could be applied to other industrial protocols, provided they are sufficiently well understood, either through published specification or detailed protocol analysis and testing.

5.8. Application

It is envisaged that, in order to apply this to a range of expected maintenance activities, training would be required on each activity, building up a library of possible permitted maintenance activities. When entering a training or monitoring session, it would be necessary to know which maintenance activity would be used. If training, a new algorithm will be created and added to the library. If monitoring, the corresponding algorithm will be pre-loaded before the session begins, so that any anomalous can be activity detected.

In order to enable this, an offline procedure will be required to ensure prior warning and authorisation is in place. The operational staff will need to be informed by the maintainers when they initially request to perform remote maintenance, stating the activities required, so that it can be cleared to go ahead. The operators will also need a user interface to select from the library of allowed activities at the pre-arranged time, and choose the one requested, so effective monitoring can take place. In addition, integration with the usual alerting system will be needed, so that for instance, a composite status dashboard can be updated, and normal alert procedures followed.

6. FUTURE WORK

Beyond the intended usage as security monitoring for maintenance sessions, this approach could also be applied to support quality management for industrial activities, during both maintenance sessions and normal operations. This is proposed as another topic for future research.

Several other areas of further work are recommended on this prototype to more fully develop and explore its utility to operational staff. Firstly, it would be beneficial to refine and improve the presentation

Figure 5: Monitoring with HMI interference late in the session

Figure 6: Monitoring with HMI interference early in the session

of classification information, including incorporating a threshold-breach notification. This would provide an alert to operational staff and is potentially usable for terminating connectivity of the remote maintainer. Secondly, it is noted that training the classifier has a random seed element, so results can vary; improvements should be explored to attempt to ensure that the classifier configuration is optimised each time training occurs. One way to do this is by removing the randomness, but this can lead to classifier over-fitting to the environment it is monitoring. This may, however, be acceptable in a very constrained ICS environment with few protocols, all with consistent data payload formats, and small numbers of nodes. Additional research is suggested to examine the implications of such deliberate over-fitting, considering the effect on results consistency, and also the potential costs and benefits in terms of training time. Thirdly, the prototype does not check PLC addressing, but if deployed in a network position that sees traffic for multiple PLCs, this feature is necessary to distinguish traffic for a specific monitored PLC, and should be added.

Fourthly, it would be potentially useful to explore live modelling of altered memory areas, as an additional use of the captured training data; by visually presenting the training data for each maintenance process, and overlaying monitored traffic, operational staff could be provided with another insight into the maintenance process.

Fifthly, the current prototype has been developed to work with a small sample set of PLCs, and it should be extended; by accommodating other automation protocols, such as PROFIBUS (Process Field Bus) (IEC 61158), which is used by Siemens' Simatic controller range, many more device types could be monitored. A useful precursor to this work would be a survey of PLC protocols to determine their suitability for this approach.

Finally, some smaller modifications are recommended to the existing prototype:

- Since the only ICF field values which are captured / monitored are 0x80 and 0x81, and both mean the same for the purposes of this work, the ICF field is not believed to be useful to anomaly detection and should be removed from anomaly detection. The prototype should, however, continue to use ICF field to filter out 0xC0 and 0xC1 (ICF response codes).
- The CMD field for both read and write messages are captured / monitored; since read messages are superfluous to monitoring PLC programming change, they should be removed.
- It may be useful to consider the field containing "Number of Elements" to the machine learning configuration. The rationale for this is while it

may be normal to write to a particular memory area as part of a maintenance session, it may be anomalous if the number of bytes written is larger than normal. Including this feature may enhance the dimensions of protection to the PLC.

7. CONCLUSION

This work has provided evidence supporting the concept that ICS maintenance sessions can be debugged using machine learning on network traffic. The experiments clearly show that one-class SVM classification can be used to detect anomalous / undesirable activity during ICS maintenance by monitoring the network traffic for FINS devices, and has the potential to enhance operational security in industrial environments. Additionally, this work demonstrates context (the use of sequences) is vital in the network traffic learning process.

This work is applicable to industrial network protocols whose structure is sufficiently understood to enable selection of fields which identify the intended purpose of each message.

More work is required to refine and improve the approach, including improvements to the accuracy of classifications, presentation of information to the end user, and supporting integration into a working environment. Also, significantly, the work has been developed using a small sample set of PLCs, and needs to be substantially extended to support more PLC types and their network protocols.

REFERENCES

- Alves, T., Das, R. and Morris, T., 2018 *Embedding Encryption and Machine Learning Intrusion Prevention Systems on Programmable Logic Controllers* IEEE Embedded Systems Letters, vol. 10, no. 3, pp. 99–102 [Online]. DOI: 10/gdrbn6.
- Dada, E.G., Bassi, J.S., Adekunle, O.O., 2017. *An Investigation into the Effectiveness of Machine Learning Techniques for Intrusion Detection*. Arid Zone Journal of Engineering, Technology and Environment 13, 764-778.
- Feng, W., Zhang, Q., Hu, G. and Huang, J. X., 2014 *Mining network data for intrusion detection through combining SVMs with ant colony networks* Future Generation Computer Systems, vol. 37, pp. 127–140 [Online]. DOI: 10.1016/j.future.2013.06.027.
- Feng, C., Li, T. and Chana, D., 2017 *Multi-level Anomaly Detection in Industrial Control Systems via Package Signatures and LSTM Networks* 2017

- 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Denver, CO, USA, IEEE, pp. 261–272 [Online]. DOI: 10/gf39jn (Accessed 25 June 2019).
- Frank, E., Hall, M. A. and Witten, I. H., 2016 *The WEKA Workbench. Data Mining: Practical Machine Learning Tools and Techniques, Fourth Edition* Morgan Kaufman [Online]. Available at https://www.cs.waikato.ac.nz/ml/weka/Witten_et_al_2016_appendix.pdf (Accessed 6 May 2019).
- Hadžiosmanović, D., Simionato, L., Bolzoni, D., Zambon, E. and Etalle, S., 2012 *N-Gram against the Machine: On the Feasibility of the N-Gram Network Analysis for Binary Protocols* Balzarotti, D., Stolfo, S. J., and Cova, M. (eds), Research in Attacks, Intrusions, and Defenses, Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp. 354–373.
- Hasan, M. S., Dean, T., Imam, F. T., Garcia, F., Leblanc, S. P. and Zulkernine, M., 2017 *A Constraint-based intrusion detection system* Proceedings of the Fifth European Conference on the Engineering of Computer-Based Systems - ECBS '17, Larnaca, Cyprus, ACM Press, pp. 1–10 [Online]. DOI: 10/gf39nn (Accessed 25 June 2019).
- King, D. E., 2009 *Dlib-ml: A Machine Learning Toolkit* The Journal of Machine Learning Research, vol. 10, no. 1, pp. 1755–1758.
- Lacey, S., 2013 *Hackers Penetrate Google's Building Management System* [Online]. Available at <https://www.greentechmedia.com/articles/read/hackers-penetrate-googles-building-management-system> (Accessed 9 February 2019).
- Moos, J., 2015 *Cyber Forensics in a Post Stuxnet World* ITNOW, vol. 57, no. 4, pp. 32–33 [Online]. DOI: 10/gfxj8f.
- Office for National Statistics, 2015 *A Short Guide to the National Accounts* Office for National Statistics [Online]. Available at <http://www.ons.gov.uk/ons/guide-method/method-quality/specific/economy/national-accounts/articles/2011-present/uk-national-accounts—a-short-guide-2014.pdf> (Accessed 9 February 2019).
- OMRON, 2001 *FINS Commands REFERENCE MANUAL* OMRON [Online]. Available at https://www.myomron.com/downloads/1.Manuals/Networks/W227E12_FINS_Commands_Reference_Manual.pdf (Accessed 27 April 2019).
- OMRON, 2009 *SYSMAC CS and CJ Series CS1W-ETN21 (100Base-TX) CJ1W-ETN21 (100Base-TX) Ethernet Units Construction of Applications OPERATION MANUAL* OMRON [Online]. Available at https://assets.omron.eu/downloads/manual/en/v2/w421_cj1w-etn21_cs1w-etn21_ethernet_units_-_construction_of_applications_operation_manual_en.pdf.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A. and Cournapeau, D., 2011 *Scikit-learn: Machine Learning in Python* MACHINE LEARNING IN PYTHON, p. 6.
- Shodan (n.d.) *Industrial Control Systems* [Online]. Available at <https://www.shodan.io/explore/category/industrial-control-systems> (Accessed 2 February 2019).
- Steinke, B., Rickel, B., 2011 *Make sense of third-party maintenance for your automatic welding system: working with third-party PM providers allows more focus on core competencies*. Plant Engineering 65, 49-.
- Stewart, James Michael, Tittel, E., Chapple, M. and Stewart, James M. (2008) *Cissp: Certified Information Systems Security Professional Study Guide* Fifth Edition (includes CD-ROM), Indianapolis, UNITED STATES, John Wiley & Sons, Incorporated [Online]. Available at <http://ebookcentral.proquest.com/lib/open/detail.action?docID=353313> (Accessed 8 May 2019).
- Stirland, J., Jones, K., Janicke, H. and Wu, T., 2014 *DEVELOPING CYBER FORENSICS FOR SCADA INDUSTRIAL CONTROL SYSTEMS* The Society of Digital Information and Wireless Communication, pp. 98–111 [Online]. Available at https://www.researchgate.net/publication/266477470_Developing_Cyber_Forensics_for_SCADA_Industrial_Control_Systems (Accessed 5 June 2017).
- Wang, J., Zhou, L., Lu, X., Ying, H. and Wang, H., 2017 *Towards a Novel Protocol Analysis Framework for Industrial Control Systems* Smart Computing and Communication, Springer, Cham, pp. 449–456 [Online]. DOI: 10.1007/978-3-319-73830-7_44 (Accessed 6 May 2019).
- Zhanwei, S. and Zenghui, L. (2019) *Abnormal detection method of industrial control system based on behavior model* Computers & Security, vol. 84, pp. 166–178 [Online]. DOI: 10.1016/j.cose.2019.03.009.