Australian Information Security Management Conference

Conferences, Symposia and Campus Events

2015

Timing attack detection on BACnet via a machine learning approach

Michael N. Johnstone
Security Research Institute, Edith Cowan University

Matthew Peacock
Security Research Institute, Edith Cowan University

J I. den Hartog
Technische Universiteit Eindhoven

DOI: 10.4225/75/57b69b5bd938c
13th Australian Information Security Management Conference, held from the 30 November – 2 December, 2015 (pp. 57-64), Edith Cowan University Joondalup Campus, Perth, Western Australia.
This Conference Proceeding is posted at Research Online.
https://ro.ecu.edu.au/ism/181

TIMING ATTACK DETECTION ON BACNET VIA A MACHINE LEARNING APROACH

Michael N. Johnstone[1], Matthew Peacock[1], J.I. den Hartog[2]
[1]Security Research Institute, Edith Cowan University, Perth, Australia
[2]Technische Universiteit Eindhoven, Eindhoven, Netherlands
m.johnstone@ecu.edu.au, m.peacock@ecu.edu.au, j.d.hartog@tue.nl

Abstract

Building Automation Systems (BAS), alternatively known as Building Management Systems (BMS), which centralise the management of building services, are often connected to corporate networks and are routinely accessed remotely for operational management and emergency purposes. The protocols used in BAS, in particular BACnet, were not designed with security as a primary requirement, thus the majority of systems operate with sub-standard or non-existent security implementations. As intrusion is thus likely easy to achieve, intrusion detection systems should be put in place to ensure they can be detected and mitigated. Existing intrusion detection systems typically deal only with known threats (signature-based approaches) or suffer from a high false positive rate (anomaly-based approaches). In this paper we present an overview of the problem space with respect to BAS, and suggest that state aware machine learning techniques could be used to discover threats that comprise a collection of legitimate commands. We provide a first step showing that the concept can be used to detect an attack where legitimate write commands being sent in rapid succession may cause system failure. We capture the state as a 'time since last write' event and use a basic artificial neural network classifier to detect attacks.

Keywords
Building Automation, Intrusion Detection, Artificial Neural Networks, Security, Heating Ventilation and Air Conditioning

INTRODUCTION

The purpose of a building is to provide safety, security and comfort for workers, systems and goods which occupy the building. The use of building automation systems (BAS) allows the collective tuning of each building service, such as energy, Heating, Ventilation and Air Conditioning (HVAC), water and access control through one centralised system. With the potential to operate and manage multiple buildings from a central location (Heaton, 2015), BAS are increasingly being connected to corporate networks, and ultimately the Internet. Providing operability between historically serial based networking and existing communication cables, a number of open source BAS protocols are designed to operate on many physical media. One such protocol is BACnet, developed by the American Society of Heating, Refrigeration and Air-conditioning Engineers (ASHRAE), and released in 1995 as ASHRAE standard 135 (SSPC-135, 2015) and ISO standard 16 484-5 in 2003 (Hersent, Boswarthick & Elloumi, 2012). BACnet is widely used in government, industry and businesses around Australia for building management. This usage is expected to grow, especially with the development of the Internet of Things (IoT), making securing and protecting BACnet a critical problem

A difficulty faced in securing BAS, exemplified by BACnet, is the ability to detect legitimate commands which, when sent in a specific pattern, manifest as a cyber-physical attack. This paper introduces a proof of concept method of detecting a BACnet specific legitimate command attack chain using time based analysis of network traffic. The remainder of the paper discusses Building Automation Security, in particular the application and shortcomings of intrusion detection systems for BACnet. Network traffic classification schemes are presented, with artificial neural networks identified as a suitable method of detecting the given problem space. A software based simulation was deployed, with a number of experiments conducted for proof of concept testing against a generated dataset. An ANN was used as a classifier for this attack chain, with the results in regards to efficiency and effectiveness discussed. The paper concludes with what these results mean, and an overview of future work related to this research.

RELATED WORK

Similar to the evolution of computing, security was not an original function of building automation systems. High levels of trust are placed in the users and the devices, due to the previously segregated and obscure nature of the systems (Novak, Treytl & Palensky, 2007; Celeda, Krejci & Krmicek, 2012). As noted by Johnstone (2009), systems are engineered for function, not security. This is a significant problem in an era where the Internet of Things is a reality. Heterogeneous connected devices need to be secure. Much like the advancement of other service-based systems, BAS were not designed with security as a paramount requirement. Early security of these systems revolved around isolationism through physical security and obscure proprietary protocols (Kastner, Neugschwandtner, Soucek & Newman, 2005). While these measures were acceptable when automation systems were not interconnected, this is not the case now and new approaches are needed. Peacock and Johnstone (2014) have highlighted the need for security in BMS, especially in BACnet systems. It is possible that an approach based on machine learning may be the way forward. Johnstone and Woodward (2012) had success with an ANN approach in a different security domain. Peacock and Johnstone (2014) concluded that modelling known attacks in a test bed BACnet controlled BAS network to generate malicious network traffic is a critical first step. In addition to this malicious traffic, normal operating traffic should also be recorded for baseline purposes to apply machine learning algorithms to traffic classification.

BACNET SECURITY

Of the three major open protocols (BACnet, KNX and LONworks), the BACnet standard has adapted the most to retrofit security techniques, defining the BACnet security service (BSS), a network layer security implementation. BSS applies encryption to each frame; implementations pre 2009 used DES-56, with an update to the standard in 2008 changing the default encryption algorithm to AES256. BACnet can provide integrity with Hashed Message Authentication Codes (HMAC), using SHA256 or MD5 for key. Encryption is not the only answer for BACnet security, with authentication of devices also of importance. Source authentication is implemented in BACnet as a device object identification number, however announcements of device numbers are peer based, meaning any device in the BACnet can request another devices ID, and any device can respond with an ID. As noted in (Holmberg, 2003; Celeda et al., 2012), this behaviour can be exploited, allowing any device in BACnet to be spoofed as another device can claim to have the requested device ID. While the BSS is defined in the standard as Addendum g, BSS is not widely implemented, with Newman (2013) stating “no company has yet implemented it in a commercially available product” Newman (2013, p. 44). Additionally, Newman (2013) suggests that BACnet/IP could be secured using IP security solutions, such as IPsec, TLS, or Kerberos. Given the vulnerabilities in these implementations, particularly TLS (Moller, Duong & Kotowicz, 2014); there remain significant security issues in BACnet.

INTRUSION DETECTION SYSTEM APPROACHES IN AUTOMATION AND CONTROL SYSTEMS

Intrusion Detection Systems (IDSs) in BAS is an immature topic area, with limited research and implementations (Kastner et al., 2005; Krejci, Celeda & Dobrovolny, 2012). Granzer and Kastner (2010) note that adaptation of IDS would be a useful complement with other security measures in BAS, particularly where authentication through encryption would have an impact on the availability of the BAS. In regards to the types of IDS implementations, Granzer and Kastner (2010) note that signature based IDSs would be inappropriate for BAS due to the increased overhead on the system for managing and storing signatures. Rather, Granzer and Kastner (2010) identify anomaly based IDS as a more appropriate solution, perhaps due to the changes in BAS network traffic over time of day, week and season Kaur, Tonejc, Wendzel and Meier (2015), Krejci et al. (2012), and the ability to tune anomaly based IDSs to these patterns. Application of anomaly detection in BACnet has been previously undertaken by Celeda et al. (2012), using a volume based entropy technique against BACnet flows developed in their previous work (Krejci et al., 2012). Analysis is conducted against network flows captured from a university network, with 3 classes of BACnet attack derived from Holmberg (2003) search for, namely, device spoofing, device discovery denial of service (DoS) and write-property attacks. No attacks were identified in the dataset, with write-properties not exhibiting a discernible pattern in the flow based approach.

In contrast, a range of ICS based IDS have been implemented (Cheung et al., 2007; Lin, Slagell, Di Martino, Kalbarczyk & Iyer, 2013; Dussel et al., 2010). Skopik, Friedberg and Fiedler (2014) states “All modern control systems are deploying IDSs as the threats against these systems increase”. A limitation of ICS based IDS, highlighted by Cheminod, Durante and Valenzano (2013) is evaluation of the performance impact of deployed IDS. Nader, Honeine and Beauseroy (2013) identify, the limited datasets of ICS systems in critical states hinders state based IDS research, as developing IDSs to determine when critical states occur would be more effective if critical state network captures can be identified. Previous work to overcome the lack of critical state datasets was conducted by Carcano et al. (2011), implementing an ICS IDS using state diagrams to model critical states. In addition to critical state analysis, Carcano, Fovino, Masera and Trombetta (2010) identified detecting chains of legitimate commands which manifest as malicious events is of interest; with an attempt conducted using network flow analysis against Modbus traffic sets. The difficulty faced with the proposed approach by Carcano et al. (2011) is the signature based method of the IDS, with previous identification of critical states, and mapping of known legitimate commands which can cause malicious actions to signatures.

There are limited examples of BACnet specific network attacks, with the majority driven by the BACnet wide area threat assessment (Holmberg, 2003). Given that other cyber-physical systems, such as Modbus have known attacks, it was theorised that applying the principles of these known attack types to BAS could reveal attack classes. The example selected was a water-hammer attack, a threat against pipelines, whereby a valve is closed suddenly causing a shockwave of force back through the pipe. In this context, closing the valve is a normal action of the system, which can have an unintended effect dependent on the time the valve is closed. Similarly in electrical motor fans, altering the fan speed is a normal process for temperature modulation in HVAC, however drastic or consistent fan speed changes over a short period of time can cause damage to the motor drive, or the fan shaft (Stanford III, 2011). Manipulation of the fan speed settings occurs in BACnet through writing assigned values to properties, from a BACnet controller. This is an identified attack class from (Holmberg, 2003; Celeda et al., 2012), namely write-property attack. Detecting these types of legitimate commands which have unintended consequences is a network traffic classification issue, which previously implemented BACnet IDS systems have not been tested for (Celeda et al., 2012).

NETWORK TRAFFIC CLASSIFICATION

Intrusion Detection Systems (IDSs) essentially solve a classification problem. A frame is presented and is classed as either safe or unsafe depending on some predefined rule(s) or application of artificial intelligence techniques. IDSs conventionally use some form of signature detection to match incoming frames with known problems (malformed packets, embedded trojans etc.). This approach works well when the attacks can be predetermined and profiled (thus providing signatures). When the problem space is less well-structured, such an approach lacks flexibility, which calls for an alternative that can detect “unknown unknowns”. Traditionally, computers attempting to solve such problems have not performed well. Computers calculate bounded problems swiftly, but do not do well at drawing inferences from incomplete data.

Rumelhart and McClelland (1986) contend that “In our view, people are smarter than today’s computers because the brain employs a basic computational architecture that is more suitable to deal with a central aspect of the natural information processing tasks that people are so good at” Therefore, an approach based on the structure of the human brain might be successful in solving this type of problem. A common approach is the Artificial Neural Network (ANN), which attempts to mimic the processing model of a brain, where neural synapses are summed, limited by an activation function to prevent overload. There are several other machine learning algorithms that solve classification problems, including Support Vector Machines and approaches based on Bayesian probability, however we chose ANNs as the most well-established approach. A neuron in an ANN consists of a node that has (potentially) multiple inputs and one output. The node activates the output if the sum of the inputs reaches a predefined value, this value being defined by an activation function, which can be binary, but need not be so. Each input can have individual weights assigned to it and it is not unusual to have a bias parameter added to the summation as shown in (1) and (2).

(1)

$$s_j = \sum_{i=1}^{n} w_i x_i$$

(2)

$$o_j = \phi(s_j + b_{ij})$$

$where$:
o_j = Output j
b_{ij} = Bias factor applied to node s_j of (1)
ϕ = The activation function

When a neuron is located in the output layer the local gradient is calculated by using equation (3). When a neuron is located in the hidden layer, we use equation (4) to calculate the gradient.

(3)

$$\delta_j = e_j \phi'_j(s_j)$$

(4)

$$\delta_j = -\frac{\partial \mathcal{E}}{\partial y_j} \phi'_j(s_j)$$

$where$:
Neuron $_j$ is hidden
$e =$ error term

We can map the logic of this single node to a network consisting of many nodes in three layers, an input layer, a middle (hidden) layer and an output layer, the latter of which defines the solution space. There are many ANN algorithms of varying efficacy, depending on the problem domain being addressed. One of the most popular is the Backpropagation algorithm, which works by providing feedback to prior nodes in the network and thus improving learning and subsequent classification. Backpropagation is computationally efficient and simple to compute as noted by Haykin (2009)

EXPERIMENTAL ENVIRONMENT

Simulation was chosen as the method of data collection, due to the limitation of available BACnet data sources, and the proof of concept nature of the research. Additionally, due to attacks being executed for network capture, a live system was not appropriate. The software used for simulation included the SCADA Engine BACnet device simulator 2.0 running on a Windows XP service pack 2 virtual machine('BACnet Device Simulator 2.0', 2009), and a Ubuntu 13.04 virtual machine, running the Open source BACnet Stack ver 0.8.2 (Karg, 2015). The virtual machines were hosted on a Macbook Pro 2015, with OS X 10.10.5, 16Gb of memory and 2.9GHz i5 processor. The topology of the simulation environment is shown as Figure 1, Scadaengine was used for the simulated temperature sensor and fan, while the BACnet stack acted as the controller, receiving temperature readings from the simulated temperature sensor, and sending fan speed commands to the simulated fan.

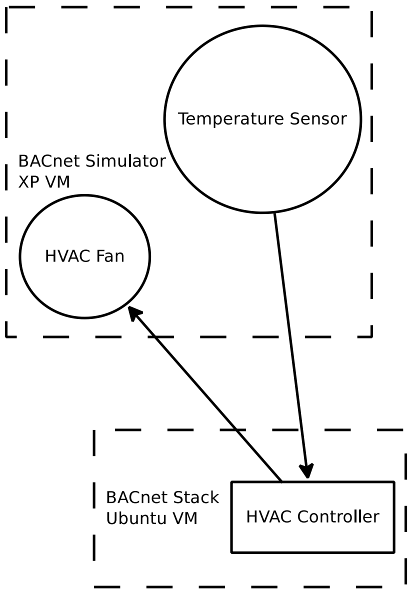

Figure 1: Network Topology of simulated BACnet HVAC environment

The simulated scenario was that of a simplistic partial HVAC system, using 1 temperature sensor, 1 fan and 1 HVAC controller communicating between the two devices. In accordance with HVAC design practices, fan speed changes occurred at a minimum 15 second interval, to replicate time delays present to prevent fan drive damage Stanford III (2011) A dormant threat in the controller pseudo-randomly interacted with the fan, sending legitimate fan speed change commands in quick succession of each other, with the intent to cause physical damage to the fan.

The data set generated contained 3 hours of BACnet simulated data, equating to 25000 frames. Pre-processing of the data involved removing all but the write frames, which were paired into events, resulting in 1000 write events. This pre-processing of the network traffic was implemented to replicate a real-time IDS, whereby non-matching frames would be discarded. The time difference (Δt) between same frames (writes) are the events of interest as we are attempting to detect a subtle attack whereby all frames are legitimate commands. The first 500 events were used to train an ANN (a backpropagation algorithm defined by 1-4) and the second 500 to test effectiveness of the trained ANN in classifying previously unseen events. The hardware used for ANN analysis was a Macbook Pro, with OS X (10.9.2) configured with 16Gb of memory and a 2.2GHz i7 processor.

RESULTS

The ANN was optimised through testing the number of iterations, the number of hidden layers and the learning rate. Testing began with a simple three-layer network (i.e., a single hidden layer). The results were encouraging, with a training time of 6.349 seconds and a classification accuracy of 90.4% against the test data set, with a classification time of 0.005 seconds. The ANN was then extended to four layers (i.e., an input layer, two hidden layers and an output layer). In this ANN, the training time was 5.718 seconds with a classification accuracy of 100.0% (classification time 0.006 sec.). With three hidden layers, the training time was 2.059 seconds and had a classification accuracy of 100.0% (classification time 0.007 sec.). Figure 2 shows that both three and five hidden layers appear to be optimal, the classification time is longer for five hidden layers. (0.008 sec.).

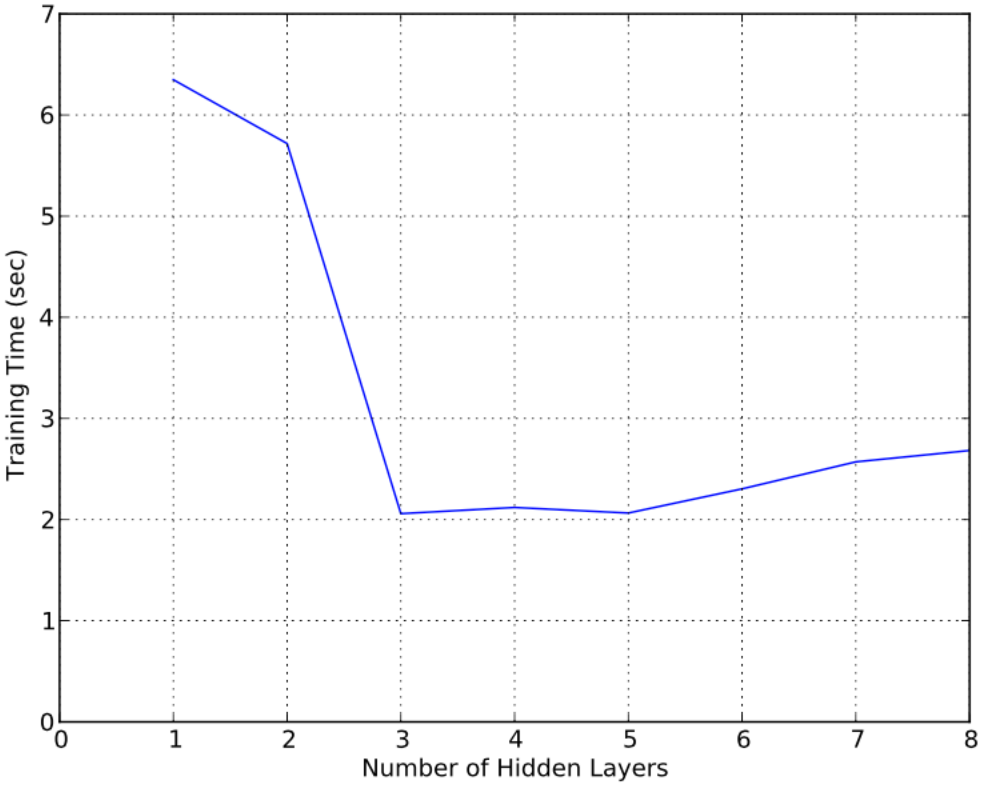

Figure 2: Number of hidden layers compared to the training time

Given that Backpropagation is a gradient descent method, it is possible for the ANN to not perform well if it does not converge on the global error maximum. We examined the error rate as illustrated in Figure 3 to ensure that the ANN was not being trapped in a local (non-optimal) minimum. Table 1 shows that the training time for the ANN increases linearly with the number of iterations, this is not an issue provided training time is not the rate determining step. There was a 10% decrease in classification accuracy when the learning rate was varied from 1.0 to 0.1. The learning rate is a measure of the size of the step taken down the gradient, so large values of learning rate correspond to smaller steps (which would be more accurate, but take longer to converge).

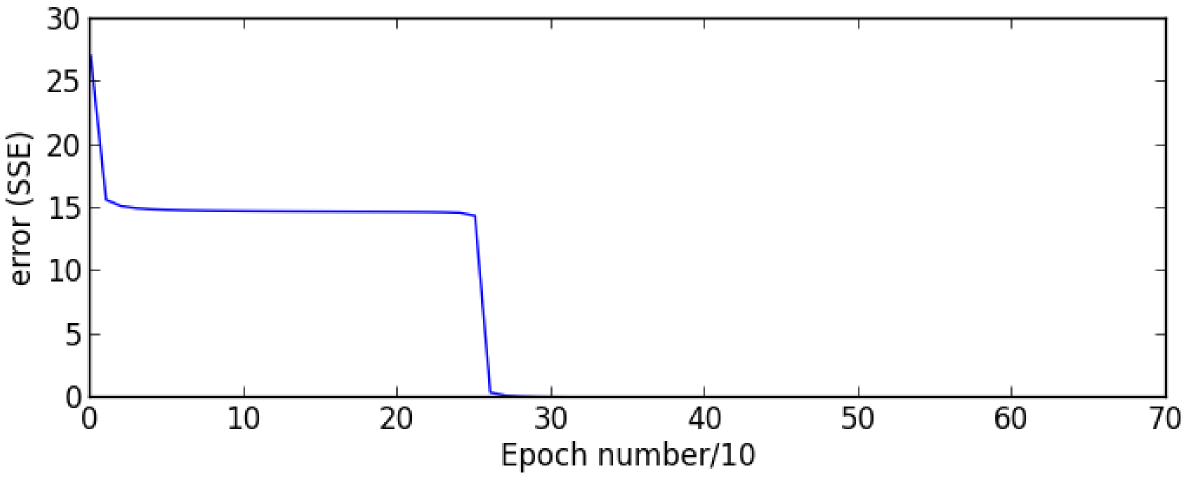

Figure 3: Error convergence in the ANN.

Table 1: Iterations of the ANN vs. Training Time of the Network.

Iterations	Time (sec.)
100	0.875
200	1.710
300	2.593
400	3.522
500	4.302

CONCLUSION AND FUTURE WORK

We provide a novel solution to one proof of concept attack against BACnet devices, through the use of ANN classification of time differences between frames of the same type. Though drawing general conclusions based on this preliminary work would be premature, it does indicate that taking into account specific aspects of the system state, in this case the time since the last write event, can help in detection of specific types of threats that BACnet devices and in general BAS face. By considering the context and threat model we have concluded that the time since the last write event is the relevant state information. Without considering this, the machine learning failed to find the attack; the critical information was hidden amongst the noise. By making the state explicit the machine learning perfectly captures the attack. The critical state information that we consider here is relatively simple requiring only the time since the last write event. With more complex BAS more complex classes of attack and thus more complex critical state models could be conducted. For example for the action 'open window' relevant context may include 'windy outside' and 'door open'. Correct critical state modelling will be key for creating a successful detection engine. While considering a general attack model and corresponding potentially critical states of the system is more specific than trying to learn without any attacker model (which, as our results support, will simply not work) yet still a lot more general than a signature (or other blacklisting heuristic) for detecting attacks. We conclude that, while still being limited, our results support that state-based intrusion detection is a promising line to pursue.

Future work will thus try to take our promising preliminary results and extend them to fully validate state-based intrusion detection as a suitable approach for the BAS domain. The ANN approach used in this paper only aimed at showing the possibility to distinguish the attacks from the normal traffic. It does not necessarily reflect the best approach for an intrusion detection system. In future work we will look at different anomaly detection mechanisms considering key intrusion detection requirements, for example, the ability to understand alerts, so called white-box approaches as promoted by Costante, den Hartog, Petkovic, Etalle and Pechenizkiy (2014) and the potential for live system data analysis. We will also increase the relevance of the experiments by expanding to a larger data set, with a larger simulated BACnet. Further classes of attacks which could be detected using this approach will be investigated, along with corresponding critical state information needed. Also legitimate command attacks applied against other similar protocols, such as Modbus, Siemens S7 and Zigbee will be considered.

ACKNOWLEDGMENTS

This work has been partially funded by the European Commission via grant agreement no. 611659 for the AU2EU FP7 project.

REFERENCES

Carcano, A., Coletta, A., Guglielmi, M., Masera, M., Fovino, I. & Trombetta, A. (2011, May). A multidimensional critical state analysis for detecting intrusions in scada systems. Industrial Informatics, IEEE Transactions on, 7(2), 179–186. doi:10.1109/TII.2010.2099234

Carcano, A., Fovino, I. N., Masera, M. & Trombetta, A. (2010). State-based network intrusion detection systems for scada protocols: a proof of concept. In Critical information infrastructures security (pp. 138–150). Springer.

Celeda, P., Krejci, R. & Krmicek, V. (2012). Flow-based security issue detection in building automation and control networks. In Information and communication technologies (Chap. 7, Vol. 7479, pp. 64–75). Lecture Notes in Computer Science. Springer Berlin Heidelberg. doi:10.1007/978-3-642-32808-4 7

Cheminod, M., Durante, L. & Valenzano, A. (2013). Review of security issues in industrial networks. Industrial Informatics, IEEE Transactions on, 9(1), 277–293. doi:10.1109/tii.2012.2198666

Cheung, S., Dutertre, B., Fong, M., Lindqvist, U., Skinner, K. & Valdes, A. (2007). Using model-based intrusion detection for scada networks. In Proceedings of the scada security scientific symposium (Vol. 46, pp. 1–12).

Costante, E., den Hartog, J., Petkovic, M., Etalle, S. & Pechenizkiy, M. (2014). Hunting the unknown - whitebox database leakage detection. In Proceedings of 28th ifip wg 11.3 conference on data and applications security and privacy (dbsec2014) (Vol. 8566, pp. 243–259). LNCS.

Dussel, P., Gehl, C., Laskov, P., Buber, J.-U., Stormann, C. & Kastner, J. (2010). Cyber-critical infrastructure protection using real-time payload-based anomaly detection. In E. Rome & R. Bloomfield (Eds.), Critical information infrastructures security (Vol. 6027, pp. 85–97). Lecture Notes in Computer Science. Springer Berlin Heidelberg. doi:10.1007/978-3-642-14379-3 8

Granzer, W. & Kastner, W. (2010). Communication services for secure building automation networks. In Industrial electronics (isie), 2010 ieee international symposium on (pp. 3380–3385). doi:10.1109/isie. 2010.5637999

Haykin, S. (2009). Neural networks and learning machines (3rd). Upper Saddle River, NJ.: Pearson. Heaton, A. (2015). The changing face of building automation. Retrieved from https://sourceable.net/changing-face-building-automation/#

Hersent, O., Boswarthick, D. & Elloumi, O. (2012). The internet of things key applications and protocols. John Wiley & Sons, Ltd.

Holmberg, D. G. (2003). Bacnet wide area network security threat assessment. NIST.

Johnstone, M. N. (2009). Security requirements engineering- the reluctant oxymoron. In Proceedings of the 7th Australian information security management conference (pp. 31–38). Perth, Western Australia.

Johnstone, M. N. & Woodward, A. (2012). Towards effective algorithms for intelligent defense systems. In CSS 2012 (pp. 498–508). LNCS. Heidelberg: Springer.

Karg, S. (2015). Bacnet protocol stack ver 0.8.2. http://sourceforge.net/projects/bacnet/.

Kastner, W., Neugschwandtner, G., Soucek, S. & Newman, H. (2005, June). Communication systems for building automation and control. Proceedings of the IEEE, 93(6), 1178–1203. doi:10.1109/JPROC.2005.849726

Kaur, J., Tonejc, J., Wendzel, S. & Meier, M. (2015). Securing bacnets pitfalls. In H. Federrath & D.

Gollmann (Eds.), Ict systems security and privacy protection (Vol. 455, pp. 616–629). IFIP Advances in Information and Communication Technology. Springer International Publishing. doi:10.1007/978- 3-319-18467-8 41

Krejcı, R., Celeda, P. & Dobrovolny, J. (2012). Traffic measurement and analysis of building automation and control networks. In Dependable networks and services (pp. 62–73). Springer.

Lin, H., Slagell, A., Di Martino, C., Kalbarczyk, Z. & Iyer, R. K. (2013). Adapting bro into scada: building a specification-based intrusion detection system for the dnp3 protocol. In Proceedings of the eighth annual cyber security and information intelligence research workshop (5:1–5:4). CSIIRW '13. Oak Ridge,

Tennessee: ACM. doi:10.1145/2459976.2459982

Moller, B., Duong, T. & Kotowicz, K. (2014). This poodle bites: exploting the ssl 3.0 fallback. Google. Nader, P., Honeine, P. & Beauseroy, P. (2013, September). Intrusion detection in scada systems using oneclass classification. In Signal processing conference (eusipco), 2013 proceedings of the 21st European (pp. 1–5).

Newman, H. M. (2013). Bacnet: the global standard for building automation and control networks.

Novak, T., Treytl, A. & Palensky, P. (2007). Common approach to functional safety and system security in building automation and control systems. In Emerging technologies and factory automation, 2007. etfa. ieee conference on (pp. 1141–1148). Retrieved from http://dx.doi.org/10.1109/efta.2007.4416910 BACnet Device Simulator 2.0. (2009). http://www.scadaengine.com/software6.html.

Peacock, M. & Johnstone, M. N. (2014). An analysis of security issues in building automation systems. In Proceedings of the 12th Australian information security management conference (pp. 100–104). Perth, Western Australia. Retrieved from http://ro.ecu.edu.au/ism/170

Rumelhart, D. & McClelland, J. (1986). Parallel distributed processing: exploration in the microstructure of

cognition. Cambridge, MA: MIT Press.

Skopik, F., Friedberg, I. & Fiedler, R. (2014, February). Dealing with advanced persistent threats in smart grid ict networks. In Innovative smart grid technologies conference (isgt), 2014 ieee pes (pp. 1–5). doi:10.1109/ISGT.2014.6816388

SSPC-135. (2015, August). Bacnet addenda and companion standards. Retrieved from http://www.bacnet.org/Addenda/

Stanford III, H. W. (2011). Hvac water chillers and cooling towers: fundamentals, application, and operation (2nd). CRC Press.