This is a postprint of the paper presented at the Federated Conference on Computer Science and Information Systems 2016, September 04-07 2016 in Gdansk, Poland. DOI: 10.15439/2016F50

Situational Awareness Network for the Electric Power System: the Architecture and Testing Metrics

Damiano Bolzoni
SecurityMatters BV
Eindhoven, The Netherlands
Email: damiano.bolzoni@secmatters.com

Rafał Leszczyna
Gdańsk University of Technology
Faculty of Management and Economics
Narutowicza 11/12, Gdańsk, Poland
Email: rle@zie.pg.gda.pl

Michał R. Wróbel
Gdańsk University of Technology
Faculty of Electronics,
Telecommunications and Informatics
Narutowicza 11/12, Gdańsk, Poland
Email: wrobel@eti.pg.gda.pl

***Abstract*—The contemporary electric power system is highly dependent on Information and Communication Technologies which results in its exposure to new types of threats, such as Advanced Persistent Threats (APT) or Distributed-Denial-of-Service (DDoS) attacks. The most exposed components are Industrial Control Systems in substations and Distributed Control Systems in power plants. Therefore, it is necessary to ensure the cyber security of this critical infrastructure and develop new cyber security technologies able to protect from advanced cyber threats. In this paper a pioneering Situation Awareness Network for the electric power system is presented together with a set of metrics for its testing.**

I. Introduction

Modern energy infrastructures aim at reducing peak demand, shifting usage to off-peak hours, lowering total energy consumption and carbon dioxide footprint [1] or enabling consumers to control their power consumption based on local needs and real-time electricity price rates [2].

To meet these requirements it is necessary to ensure the continuous exchange of data between all points of the network. Although the communication infrastructure may partially exist, it is necessary to facilitate its vast expansion by increasing bandwidth (among the others due to the introduction of two-way communication as an inherent component of the new energy infrastructure and smart grid) and connecting consumers (residential, commercial, industrial, etc.). To reduce the costs which are incurred by this process, the Internet is often used as communication backbone for the energy management systems [1].

However, such an approach exposes the power system to a great security breach. Every network layer and technology used in the new energy infrastructure represents a potential target of a cyber-attack. This in particular refers to Industrial Control Systems (including SCADA) in substations and Distributed Control Systems (DCS) in power plants. Moreover in the recent years wireless networks have been widely employed as part of many industrial communication systems, which exposes the entire network to even greater risk [3].

Advanced Persistent Threats (APT) are dedicated attacks able to persistently target a specific entity and to cause an intended effect, such as an interruption to the power supply [4], [5]. DDoS attacks, on the other hand, attempt to delay, block or corrupt the communication in the grid [6].

Stuxnet [7] was the first wide manifestation of malware that was specifically designed to attack networked industrial control systems used in the power system. Detected for the first time in 2010, Stuxnet is a cyber worm able to infect process servers and Programmable Logic Controllers (PLCs) and alter physical processes. The ultimate goal of Stuxnet is to sabotage the attacked facility by reprogramming programmable logic controllers (PLCs) to render them operating out of their specified boundaries. Later studies revealed that Stuxnet was not the first threat of that type. In fact that it had its precursor called Flame that was undetected [8]. Flame is a large complex malware designed to aggressively gather information from its target systems. Apart of conventional information stealing methods it is able to capture Skype calls and record audio [9].

Since the manifestation of Stuxnet both information security experts and hackers have shown a much greater level of interest in this area. As a result, 64 ICS vulnerabilities were discovered in 2011 and 98 additional ones were announced in the first eight months of 2012 alone – more than the total number for the preceding seven years combined [10]. In parallel sophisticated attacks have been appearing – Duqu, Red October, Gauss and Black Energy – each of them more complex and advanced than its predecessor [9], [11]. Duqu was designed to steal information in preparation for a Stuxnet-like attack and it used new techniques never previously noted [9]. Red Dragon and Gauss utilise encryption in order to effectively penetrate the infiltrated information systems [11], [9]. Black Energy is the most recently discovered malware which aims at Industrial Control Systems used in critical infrastructures [12].

Taking into account all these threats and the attacks already carried out, it is necessary to take countermeasures. Standard cyber security technologies and best practices – such as access control, anti-malware, firewalls, intrusion detection and

The study presented in this paper is based on work carried out in the DEnSeK (Distributed Energy Security Knowledge) project founded by the European Commission, Directorate-General for Home Affairs (Programme „Prevention, Preparedness and Consequence Management of Terrorism and other Security-related Risks" – CIPS, Project Reference: HOME/2012/CIPS/AG/ 4000003772) and partially supported from the project funds. It is also supported by the DS Programs of Faculty of Management and Economics and Faculty of Electronics, Telecommunications and Informatics of Gdańsk University of Technology.

prevention systems, defence in depth, and system hardening – are indispensable in protecting the power system. However, they are only a partial solution [4], [13], [14], [15].

To counter the evolved, highly sophisticated threats, advanced cyber security technologies are required, such as Security Information and Event Management (SIEM) systems, application whitelisting, and Trusted Platform Modules (TPM) [4], [13], [16] together with an efficient and effective risk assessment and management [17]. Developing and deploying Situation Awareness Networks (SANs) with SIEM software will improve situational awareness and will allow for better control and faster response to threats [18].

Such a Situation Awareness Network (SAN) has been developed in the project DEnSeK (Distributed Energy Security Knowledge) [19]. The project aimed at improving the security and resilience of the new energy infrastructure against cyber-threats by providing a platform for the security knowledge exchange between companies of the European energy sector and establishing a European Energy ISAC (Information Sharing and Analysis Centre) which enables interactive and real-time knowledge and information sharing between all involved parties [19].

In this paper the SAN architecture is presented along with the set of metrics to be used for its evaluation. To the best of authors' knowledge such a dedicated set of metrics for Situation Awareness Networks (SANs) has not been proposed so far, most probably because the concept of SAN is relatively new. It must be underlined that evaluation of Situation Awareness Networks is an area distinct from the quantitative assessment of the level of situational awareness. For the latter several approaches exist [20].

II. Situational Awareness Network Architecture

The Situational Awareness Network encompasses and combines a number of diverse network-based sensors, which facilitate network traffic and data monitoring and detection of various events. Collected and processed data sets are visualised to a SAN operator who responds to emerging threats.

The need for combining together multiple sensors stems from the observation that in the past half-decade monitoring tools have become more specialised and now they focus on specific threat vectors and/or analysis approaches. Hence, in order to offer a broad overview of network activities and potential issues, it is crucial to combine diverse monitoring engines.

The purpose of the visualization is two-fold. First, operators can spot anomalies that the automatic systems might not be able to detect or might not be configured to detect. In this case, a visualisation dashboard supports the analysis of a large amount of data as it reduces it significantly focusing on key parameters for detecting anomalies.

Secondly, once an event is reported by one of the automatic systems (for instance, a malware spread is detected), operators can leverage the visualisation dashboard to observe the way network traffic evolves and either confirm or reject the alert previously raised.

In the DEnSeK project a three-tier architecture of the Situational Awareness Network, presented in Fig. 1, was proposed. The lowest, data tier consists of sensors which collect network data. In the logic tier, Security Information and Event Management (SIEM) software processes data from sensors and transmits them to the top layer. Finally in the presentation tier, the dashboard visualises the data by a user-friendly operator interface.

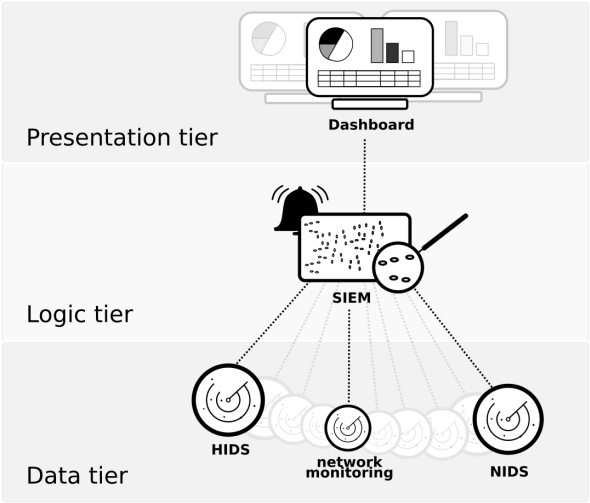


Fig. 1. SAN three-tier architecture

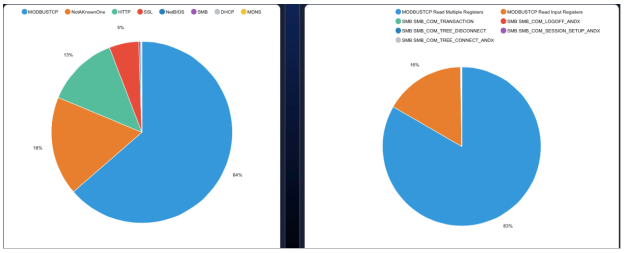
In the approach presented in this paper, SAN needs a signature-based NIDS (such as Snort [21], [22] and Suricata [23]), to detect well-known attack payloads, and several behavioural-based engines to analyse both payloads and flows for anomalies. As it relies on open-source/freely available tools, currently exist few alternatives for behavioural-based systems that can be used in production environments. One of them is Bro [24], [25], which can be used to code any type of algorithm on top of its protocol parsers.

On top of regular network monitoring tools and SIEMs that are available off-the-shelf, a visualisation dashboard is located. Its main role is to allow operators to observe the behaviour of the underlying industrial network. The dashboard provides several widgets, presented in Fig. 2 that can be instantiated to present various dimensions of network traffic (IP addresses, TCP ports, protocols, etc.) using different metrics (bytes, packets, protocol messages, etc.).

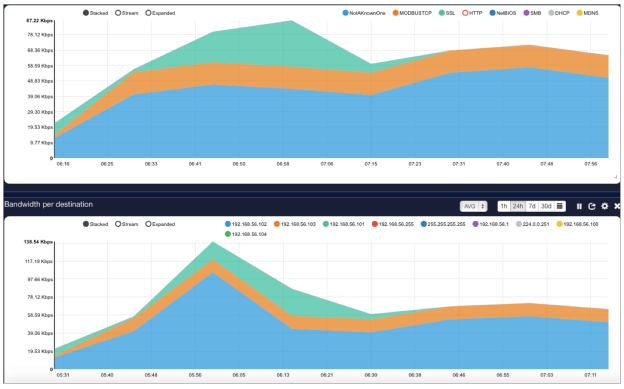
The central SIEM node is provided with Syslog (system logging) messages by various network-based sensors. This is a standard practice that enables required flexibility while providing all the necessary information. The visualisation dashboard leverages diverse software and components in order to deliver the extracted metadata to a central repository.

At the bottom of the architecture, a Linux-based computer equipped with Argos (a Linux-based SCADA alternative [26]) analyses network streams and extracts relevant metadata. Such data are sent via Apache Kafka [27] publish-subscribe messaging service to a central repository based on Druid [28]. This

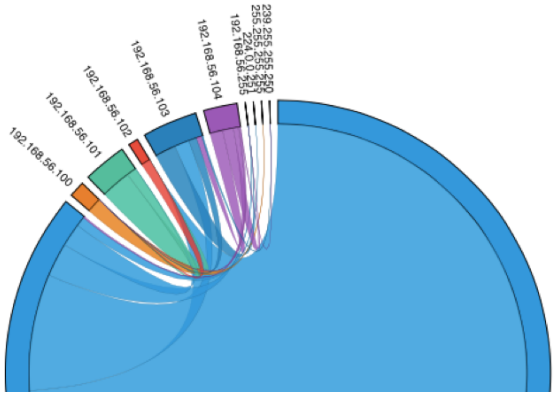
MOST WIEDZY Downloaded from mostwiedzy.pl



(a) Network protocols and commands used in the communication



(b) Bandwidth by protocol and destination



(c) Connections between network hosts

Fig. 2. SAN Dashboard widgets

is a real-time data store that takes advantage of an in-memory architecture to facilitate data aggregation and fast querying. Data processed by Druid are queried via graphical web widgets based on the D3JS framework.

The visual analytics component does not fulfil only the task of depicting network data flows and interactions in real-time, but can be also applied to guiding the development of specific controls and checks. Every organisation in fact will exhibit a slightly different network layout and configuration, even those within the same industrial sector and/or running the same software package. This diversity requires a certain degree of customisation of controls, to tackle the specificity of a certain environment.

End users can perform an assessment of their network through the visual analytics component to baseline network behaviour, discover misconfigurations and assess issues. After this initial assessment, end users can select key indicators that can point at operational issues or cybersecurity breaches. To provide some examples, RTUs used in the field typically exchange data with the SCADA master via long-lasting connections that could be running for weeks or months. In case an RTU loses too often connectivity and re-establishes connectivity frequently, this could indicate an issue with the device itself (for instance, end of life) or with the network infrastructure (because of a wireless link). Key indicators can be enforced by writing a specific script in Bro, or a signature in Snort and/or Suricata.

III. Metrics in the Testing and Evaluation Process

Testing is an integral part of software development process. In the document „Standard Glossary of Software Engineering Terminology" IEEE defines *testing* as the „process of operating a system or a component under specified conditions, observing or recording the result and making an evaluation of some aspect of the system or component" [29].

The goal of testing is to detect the difference between existing and required conditions and to evaluate the features of the software items [30]. Currently, testing is a mature and well-defined area of software engineering. Good testing process design should ensure the repeatability, manageability and measurability [31].

As a part of the development of the Situational Awareness Network for the DEnSeK project, integration tests have been carried out. Their aim was to validate a selection of SAN components and check their operational capability in a complex test environment. During the tests appropriate interaction between the components was verified.

The tests were performed in the cyber security laboratory of one of the largest European electricity companies. They proved that the architecture and system components were properly selected and the system operates as intended [32].

Despite the positive results of the tests, the lack of quantitative indicators made it difficult to objectively assess the results. It was only possible to grade binary – it works or does not work. The extent to which the requirements are met, however, could not be determined in a measurable way. As far as only integration tests are concerned, this binary evaluation is sufficient. Nevertheless the majority of evaluations require higher precision.

Thus in order to enable objective evaluation of a software product and its development process *software metrics* were introduced. A *software metric* is a „quantitative measure of the degree to which a system, component or process has given attribute" [29]. The knowledge gathered on the basis of metrics should lead to an improved process and products [33]. The metrics can be divided into two groups: product metrics and metrics for testing process [34].

Metrics from the first group are used to provide information about the quality and maturity of the tested product. They facilitate early detection of product flaws and related problems and enable their more accurate correction or elimination. In

MOST WIEDZY Downloaded from mostwiedzy.pl

addition, metrics provide quantitative criteria which may be used in the process of acceptance of the final product.

The latter group contains metrics that allow for monitoring of the progress of the testing process and its results after the execution. They are used on one hand, to evaluate the effectiveness of the testing process. On the other hand, they provide test termination criteria.

The use of software metrics as objective evaluation criteria is extremely important in the management of the software development process [35].

Software metrics have been developed practically for all application domains. However, to the best of authors' knowledge the metrics for Situational Awareness Platforms have not been proposed so far. This is most probably due to the fact that the concept of Situation Awareness Network and the implementing it platforms are relatively recent.

In order to fill this gap the relevant research studies have been investigated to provide a comprehensive set of metrics for testing SANs. The metrics' proposals in several fields have been identified and analysed, including Intrusion Detection and Prevention Systems, Security Information and Event Management systems as well as general domains such as software engineering, testing or cyber security. The data collected allowed for selecting the relevant metrics.

When choosing the metrics, the Jaquith's [36] recommendations were taken into account. According to him, good metrics should be [36]:

- consistently measured,
- expressed as cardinal number or percentage,
- expressed using unit of measure,
- contextually specific,
- possible to obtain at reasonable cost.

In order to design the test procedure for the SAN system, a set of metrics was selected. Metrics described in Section IV regard the testing process. They facilitate the control and management of the testing process, as well as deciding when to end it.

Other metrics are related to the product. In the evaluation, the product is understood as a complete SAN system. Given the characteristics of the system the metrics are also divided into two groups. In Section V cyber security metrics are presented. They allow for evaluating the core SAN functionality answering the question of how the system copes with the detection of security threats. The last group of metrics, presented in Section VI, refer to system usability. As one of the functions of the SAN, provided by the Dashboard, is the visualisation of security threats to an operator, the quality of user interface is very important.

The main criterion taken into account when selecting the metrics was the possibility of their straightforward implementation to assess SAN platform at every stage of development. In addition, a set of metrics was chosen to cover as widely as possible all aspects of the testing process.

IV. Metrics for testing process

Testing metrics are widely used in the field of software testing. Their aim is to „provide information about the testing status of a software product" [34].

Quadri and Farooq [34] divided testing metrics into several groups. First of all they highlighted the metrics related to measuring time, such as time required to run a test, time interval between failures or number of failures in specific time interval. After that the metrics for evaluating test efficiency, source code coverage and quality were described. Finally metrics related to defect identification and fixing were presented.

Chen et al. [37] conducted an in-depth analysis of software metrics, examining the effectiveness of a set of complementary metrics for cost, time, and quality to measure the quality of test process. Based on the result they proposed four new testing metrics: two related to product improvements and two related to costs.

Kaur et al. [35] surveyed, classified and systematically analysed the metrics proposed in the previous decades. They discussed advantages or disadvantages for each product metric along with its need and purpose. The suitability, effect, data calibration and interpretation of metrics was also evaluated.

Based on the studies as well as the specificity of the DEnSeK project four testing metrics were selected.

A. Source code coverage

The source code coverage metric enables evaluating the confidence in the effectiveness of a test suite. The metric is defined as follows:

$$SC = \frac{St_t}{St} \tag{1}$$

where:

- SC – source code coverage,
- St_t – number of statements of a source code covered by test suite
- St – number of statements of a source code,

The metric shows what part of the source code has been covered with tests. If the value is too low, there should be written additional test cases for uncovered source code.

B. Test case defect density

Test case defect density metric indicates whether the test cases are effective and efficient in their ability to detect a larger number of defects. It is defined as:

$$DD = \frac{F}{TE} \times 100\% \tag{2}$$

where:

- DD – test case defect density,
- F – failures detected,
- TE – number of executed test cases.

MOST WIEDZY Downloaded from mostwiedzy.pl

C. Failures detection rate

Failures detection rate metric test indicates whether the prepared tests are time effective in terms of the number of detected defects per unit time. The metric is defined by the following formula:

$$FD = \frac{F_T}{T} \tag{3}$$

where:

- FD – failures detection rate
- F_T – failures detected in T time
- T – number of business days used for testing

D. Test improvement in product quality

Test improvement in product quality metric shows the relation between the number of weighted defects detected and the size of the product release. It is defined as:

$$TI = \frac{W_p}{KCSI} \tag{4}$$

where:

- TI – test improvement in product quality,
- W_p – number of weighted defects found in one specific test phase,
- $KCSI$ – number of new or changed source lines of code in thousands.

The higher this number, the higher is the improvement of the quality of the product contributed during this test phase.

V. Cyber security metrics

Cyber security metrics are strictly related to the functional operation of the Situational Awareness Network. The selection was made among the metrics defined for security systems such as Intrusion Detection Systems. One of the main problems that SAN operators would face is the reliability of the threats detection. There are two main aspects to be taken into consideration: false positives and true negatives. [38]

A *true positive* is when SAN informs about threat that really exists. This is the desired situation. A *false positive* takes place when SAN informs about threat that does not occur. A *true negative* refers to the situation when SAN does not inform about threat that really occurs.

Using these terms three metrics have been defined. Additionally two metrics based on the research conducted by Bayuk and Mostashari [39] were proposed.

A. Accuracy

Accuracy metric describes the proportion of true results (both true positives and true negatives) in the population of all network events. It is defined as:

$$A = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

where:

- A – detection accuracy
- TP – number of true positives
- TN – number of true negatives
- FP – number of false positives
- FN – number of false negatives

A higher value indicates a more reliable system operation.

B. Detection rate

Detection rate determines the effectiveness of threats' detection. When the metric value is closer to 1, the system is more effective. A value of 1 means that each threat has been detected.

$$DR = \frac{TP}{TP + FN} \tag{6}$$

where:

- DR – detection rate
- TP – number of true positives
- FN – number of false negatives

C. False positive rate

False positives are one of SAN biggest issues. Their frequent occurrence significantly undermines the effectiveness of the SAN. Efforts should be made to the lowest value of this indicator.

$$FPR = \frac{FP}{FP + TP} \tag{7}$$

where:

- FPR – detection accuracy
- FP – number of false positives
- TP – number of true positives

D. Mean Time Between Failures

Mean time between failures (MTBF) is a standard metric that describes reliability of the system. In the case of SAN failure is defined by the occurrence of either false positive or true negative. The metric is defined as:

$$MTFB = \frac{\sum_2^{NF}(B_n - E_{n-1})}{NF - 1} \tag{8}$$

where:

- $MTFB$ – Mean Time Between Failures
- B_n – beginning of *n*-th failure
- E_n – end of *n*-th failure
- NF – number of failures

E. Time To Protect

The metric is defined as the mean time between the detection of the threat and noticing it by the operator. In this way, both the effectiveness and efficiency of the system, as well as the legibility of the information about the threat on the dashboard, are evaluated.

$$TTP = \frac{\sum_1^{NT}(A_n - D_n)}{NT} \tag{9}$$

where:

- TTP – Time To Protect

MOST WIEDZY Downloaded from mostwiedzy.pl

- A_n – time of n-th threat notice
- D_n – time of n-th threat detection
- NT – number of threat detections

VI. User experience metrics

In addition to testing against the above criteria, software should be evaluated in terms of usability. This is particularly relevant to software interface, but not exclusively. As far as the DEnSeK SAN is concerned, the Dashboard requires special attention in regard to usability. Based on the metrics proposed by Tullis and Albert [40], the following metrics are proposed for evaluation of the SAN usability.

A. Task success

This metric enables measuring the extent to which a user is able to perform a given task. Task success can be measured binary (succeed/failed), or as a level of success. The tasks with a lower coefficient of success must be analysed to detect the elements of the user interface which cause problems.

B. Time-on-task

Time-on-task allows for measuring the time required to complete a specific task. The faster a user can complete a task, the experience is better. In the DEnSeK project the metric serves for evaluating the efficiency of the Dashboard.

C. Efficiency

In contrast to the previous metric, which concerned time, the efficiency metric enables measuring the amount of work required to complete a task. For instance such an effort can be expressed by means of the number of mouse clicks or keystrokes.

D. Errors

This metric allows for detecting improperly designed user interface elements that cause users' confusion. It is measured as the number of user errors when performing a task. Errors may be related to spelling, pressing a wrong key, etc.

E. Learnability

The learnability metric supports examining whether and how user productivity increases with the better knowledge of the system. Measuring learnability requires intense studies spanning a long period of time. For this reason it is often left out.

VII. Conclusion

The metrics described in the paper are used to evaluate Situational Awareness Network (SAN) system developed in the DEnSeK project. The SAN was designed as a three-tier architecture. The lowest tier encompasses a number of sensors for network monitoring. In the middle tier, the SIEM software collects and processes the data from the sensors. Finally, the dashboard on the top tier visualizes information about the threats.

In order to select the appropriate set of metrics a thorough literature analysis was conducted. To the best of authors' knowledge the metrics for SAN have not been proposed so far. Therefore software metrics developed for several related fields, including cyber security, Intrusion Detection and Prevention Systems, SIEM systems, software engineering and testing have been analysed. The study made it possible to derive a set of metrics for testing Situational Awareness Networks.

The selected metrics were divided into three groups. The first group contains metrics related to the evaluation of the testing process, the second – to the effectiveness of threat detection, and the last – to the usability of the dashboard. The metrics are used at each stage of the SAN development. In addition they will be applied during final product evaluation and acceptance process.

References

[1] R. Kyusakov, J. Eliasson, J. Van Deventer, J. Delsing, and R. Cragie, "Emerging energy management standards and technologies - Challenges and application prospects," in *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, 2012. doi: 10.1109/ETFA.2012.6489674. ISBN 9781467347372

[2] F. Maturana, R. Staron, K. Loparo, R. Ambre, and D. Carnahan, "Simulation-based environment for modeling distributed agents for smart grid energy management," in *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2011*, 2011. doi: 10.1109/ETFA.2011.6059124. ISBN 9781457700187. ISSN 1946-0740

[3] G. Dini and M. Tiloca, "On simulative analysis of attack impact in Wireless Sensor Networks," in *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, 2013. doi: 10.1109/ETFA.2013.6648059. ISBN 9781479908622. ISSN 19460740

[4] Y. Aillerie, S. Kayal, J.-p. Mennella, R. Samani, S. Sauty, and L. Schmitt, "Smart Grid Cyber Security," 2013.

[5] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, "A Survey on Cyber Security for Smart Grid Communications," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 4, pp. 998–1010, 2012. doi: 10.1109/SURV.2012.010912.00035. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6141833

[6] W. Wang and Z. Lu, "Cyber security in the Smart Grid: Survey and challenges," *Computer Networks*, vol. 57, no. 5, pp. 1344–1371, apr 2013. doi: 10.1016/j.comnet.2012.12.017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128613000042

[7] N. Falliere, L. O. Murchu, and E. Chien, "W32.Stuxnet Dossier," Symantec Security Response, Tech. Rep., 2011.

[8] D. Kushner, "The real story of stuxnet," *IEEE Spectrum*, vol. 50, pp. 48–53, 2013. doi: 10.1109/MSPEC.2013.6471059

[9] P. Shakarian, J. Shakarian, and A. Ruef, *Introduction to Cyber-warfare*. Elsevier, 2013. ISBN 9780124078147. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780124078147000087

[10] P. Technologies, "SCADA Safety in Numbers," Tech. Rep., 2012.

[11] N. Virvilis and D. Gritzalis, "The Big Four - What We Did Wrong in Advanced Persistent Threat Detection?" in *2013 International Conference on Availability, Reliability and Security*. IEEE, sep 2013. doi: 10.1109/ARES.2013.32. ISBN 978-0-7695-5008-4 pp. 248–254. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6657248

[12] ICS-CERT, "Alert (ICS-ALERT-14-281-01B) Ongoing Sophisticated Malware Campaign Compromising ICS (Update B)," 2014.

[13] A. Carcano, A. Coletta, M. Guglielmi, M. Masera, I. N. Fovino, and A. Trombetta, "A Multidimensional Critical State Analysis for Detecting Intrusions in SCADA Systems," *Industrial Informatics, IEEE Transactions on*, vol. 7, no. 2, pp. 179–186, 2011. doi: 10.1109/TII.2010.2099234

[14] A. Felkner and A. Kozakiewicz, "More Practical Application of Trust Management Credentials," in *Proceedings of the 2015 Federated Conference on Computer Science and Information Systems*, ser. Annals of Computer Science and Information Systems, M. Ganzha, L. Maciaszek, and M. Paprzycki, Eds., vol. 5. IEEE, 2015. doi: 10.15439/2015F95 pp. 1125–1134. [Online]. Available: http://dx.doi.org/10.15439/2015F95

MOST WIEDZY Downloaded from mostwiedzy.pl

[15] O. Rysavy, J. Rab, and M. Sveda, "Improving security in SCADA systems through firewall policy analysis," in *Proceedings of the 2013 Federated Conference on Computer Science and Information Systems*, M. P. M. Ganzha L. Maciaszek, Ed. IEEE, 2013, pp. pages 1423—-1428.

[16] M. Chakraborty, N. Chaki, and A. Cortesi, "A New Intrusion Prevention System for Protecting Smart Grids from ICMPv6 Vulnerabilities," in *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems*, ser. Annals of Computer Science and Information Systems, M. P. M. Ganzha L. Maciaszek, Ed., vol. 2. IEEE, 2014. doi: 10.15439/2014F287 pp. pages 1539—-1547. [Online]. Available: http://dx.doi.org/10.15439/2014F287

[17] A. Bialas, "Experimentation tool for critical infrastructures risk management," in *Proceedings of the 2015 Federated Conference on Computer Science and Information Systems*, ser. Annals of Computer Science and Information Systems, M. Ganzha, L. Maciaszek, and M. Paprzycki, Eds., vol. 5. IEEE, 2015. doi: 10.15439/2015F77 pp. 1099–1106. [Online]. Available: http://dx.doi.org/10.15439/2015F77

[18] H. Khurana, M. Hadley, and D. Frincke, "Smart-grid security issues," *IEEE Security & Privacy Magazine*, vol. 8, no. 1, pp. 81–85, jan 2010. doi: 10.1109/MSP.2010.49. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5403159

[19] "DEnSeK (Distributed Energy Security Knowledge) - project website." [Online]. Available: http://www.densek.eu/

[20] M. R. Endsley and D. J. Garland, *Situation Awareness Analysis and Measurement*. CRC Press, Inc., 2000.

[21] "Snort Home Page." [Online]. Available: http://www.snort.org/

[22] Z. Zhou, "The study on network intrusion detection system of Snort," in *2010 International Conference on Networking and Digital Society*, vol. 2. IEEE, may 2010. doi: 10.1109/ICNDS.2010.5479341. ISBN 978-1-4244-5162-3 pp. 194–196.

[23] OISF, "Suricata - Open Source IDS / IPS / NSM engine." [Online]. Available: http://suricata-ids.org/

[24] "The Bro Network Security Monitor," 2016. [Online]. Available: https://www.bro.org/

[25] G. K. Varadarajan, "Web Application Attack Analysis Using Bro IDS," 2012. [Online]. Available: http://www.sans.org/reading-room/whitepapers/detection/web-application-attack-analysis-bro-ids-34042

[26] "Argos," 2016. [Online]. Available: https://sourceforge.net/projects/argos-scada-en/

[27] "Apache Kafka: a high-throughput distributed messaging system," 2016. [Online]. Available: http://kafka.apache.org/

[28] "Druid." [Online]. Available: http://druid.io/

[29] A. September, "IEEE Standard Glossary of Software Engineering Terminology/IEEE Std 610.12-1990," p. 96, 1990. [Online]. Available: http://www.amazon.com/Standard-Glossary-Engineering-Terminology-610-12-1990/dp/155937067X

[30] J. Radatz, A. Geraci, and F. Katki, "IEEE standard glossary of software engineering terminology," *IEEE Std*, vol. 610121990, p. 121990, 1990.

[31] I. Burnstein, T. Suwanassart, and R. Carlson, "Developing a testing maturity model for sofiware test process evaluation," in *Test Conference, 1996*, 1996. ISBN 0780335406 pp. 581–589.

[32] R. Leszczyna, R. Małkowski, and M. R. Wróbel, "Testing Situation Awareness Network for the Electrical Power Infrastructure," *Acta Energetica*, vol. 1, pp. 270–276, 2015.

[33] P. Goodman, *The Practical Implementation of Software Metrics*. McGraw-Hill, Inc., 1993.

[34] S. Quadri and S. Farooq, "Notable Metrics in Software Testing," *5th National Conference on Computing For Nation Development - INDIACom-2011*, pp. 273–276, 2011.

[35] A. Kaur, B. Suri, and A. Sharma, "Software testing product metrics-A Survey," in *National Conference on Challenges & Opportunities in Information Technology*, 2007, pp. 1–6.

[36] A. Jaquith, *Security Metrics, Replacing Fear, Uncertainty, and Doubt*. Addison-Wesley Professional, 2007.

[37] Y. Chen, R. L. Probert, and K. Robeson, "Effective test metrics for test strategy evolution," pp. 111–123, 2004.

[38] D. Kang, D. Fuller, and V. Honavar, "Learning classifiers for misuse and anomaly detection using a bag of system calls representation," *Sixth Annual IEEE SMC Information Assurance Workshop*, 2005. doi: 10.1109/IAW.2005.1495942

[39] J. L. Bayuk and A. Mostashari, "Measuring cyber security in intelligent urban infrastructure systems," in *2011 8th International Conference & Expo on Emerging Technologies for a Smarter World*. Ieee, nov 2011. doi: 10.1109/CEWIT.2011.6135873. ISBN 978-1-4577-1591-4 pp. 1–6.

[40] W. Albert and T. Tullis, *Measuring the user experience: collecting, analyzing, and presenting usability metrics*. Newnes, 2013. ISBN 9780124157811

MOST WIEDZY Downloaded from mostwiedzy.pl