Decision Support for Mission-Centric Network Security Management

Michal Javorník, Jana Komárková, Lukáš Sadlek, Martin Husák
Institute of Computer Science, Masaryk University, Brno, Czech Republic
javornik@ics.muni.cz, komarkova@ics.muni.cz, sadlek@mail.muni.cz, husakm@ics.muni.cz

***Abstract*—In this paper, we propose a decision support process that is designed to help network and security operators in understanding the complexity of a current security situation and decision making concerning ongoing cyber-attacks and threats. The process focuses on enterprise missions and uses a graph-based mission decomposition model that captures the missions, underlying hosts and services in the network, and functional and security requirements between them. Knowing the vulnerabilities and attacker's position in the network, the process employs logical attack graphs and Bayesian network to infer the probability of the disruption of the confidentiality, integrity, and availability of the missions. Based on the probabilities of disruptions, the process suggests the most resilient mission configuration that would withstand the current security situation.**

***Keywords*—Cyber situational awareness, Decision support, Attack graph, Bayesian network, Mission resilience**

I. Introduction

The rising complexity of today's communications networks and information systems makes it more difficult to protect critical information infrastructures (CII). In particular, it is difficult to eliminate all the vulnerabilities and to protect all the components of the CII. When breached (and the breach will happen), each reconfiguration needs to be carefully considered as it can impact the mission as well, and the mission operation interruption consequences can be catastrophic.

Nowadays, we stand in a situation, when the environment is complex, and fully automated incident response is risky. Thus, there is a need to keep a human operator in the loop [1]. The operators need to be supported by an analytical system that would increase their Cyber Situation Awareness (CSA), i.e., perception of the elements in the environment within a volume of time and space, the comprehension of their meaning and the projection of their status in the near future [1]. The CSA needs to be supported by analytical systems capable of handling inputs, estimating the state of the network and attack, and assessing the impacts of attack and defensive actions in the context of the critical missions.

A promising approach to prioritize cyber threats and find resilient mission configurations is to focus on mission or business objectives that must be achieved despite the presence of threats. In such a case, we refer to mission-centric cyber situational awareness [2]. The operator is given a task to find a most resilient mission configuration, which requires evaluating the effects of the threats on the mission operation. Such evaluation is difficult because, first, the attack takes effect on the infrastructure levels, affecting hosts and dependent on the current network configuration, and there is a need to estimate how the effects propagate further up to the mission level, which is more abstract. Second, the projection of the situation in the future is important. The complexity of the task is well beyond the ability of a human operator to comprehend. Therefore, we propose decision support for mission-centric network security management. To answer the challenge, we propose a mission-centric approach to decision support for mission operation management that processes the machine-readable information on the protected network, vulnerabilities detected in the network, and description of enterprise missions of the organization that operates the network. By combining the three sources of information, we can calculate a potential impact of vulnerabilities to enterprise mission, and assess the most resilient mission configurations.

This paper is divided into six sections. In Section II, we briefly describe the related work on mission-centric decision support. In Section III, we describe the decision support process. Section IV provides a more in-depth insight into the crucial parts of the process, including the equations and algorithms. Section V concludes the paper and outlines further research directions.

II. Related Work

The key aspect of the mission-centric approach is the ability to map the assets to missions. For example, Sun et al. [3] proposed a mission-task-asset map to associate mission with its tasks with corresponding assets. This map is built on top of a system object dependency graph that captures the intrusion propagation at a very low level. The map is discovered by pattern mining in the interactions of the tasks with the objects in the dependency graph. The authors used a Bayesian network based on the mission-task-asset map and the system object dependency graph to infer the probability of a mission task being tainted using intrusion evidence. Guion and Reith [4] mapped the mission to a cyber terrain that consists of systems, devices, data, processes, cyber personas, and other network entities, the control of which offers a marked advantage to an attacker or defender and as a concept is supportive for mission-based impact assessment. The author surveys current methodologies for mission mapping and the existing tools that seek to provide CSA through mission mapping. Recently, Silva and Jacob [2] focused on a mission-centric risk assessment methodology. They pointed out the need for switching from threat-centric and vulnerability-centric to mission-

centric approaches as the focus on a mission gives a more comprehensive picture from the human operator perspective. The authors also summarized the proposed risk assessment methodology required for mission-centric assessment. The risk assessment preparation requires modeling the enterprise missions and assets and relations between them, specifying the risk measurement criteria, and so forth. Subsequently, as part of conducting the risk assessment, all threats and their impact on the assets need to be identified. Lastly, linking the enterprise model to the identified threats is required to estimate the disruption of the mission.

Jakobson [5] discussed the need to achieve mission resiliency and proposed mission resiliency based on two interacting dynamic processes: the process of mission operation situation management and the process of cyber situation management. The author discussed architecture and enabling technologies of such mutually adaptive processes that keep the mission persisting even if the network that supports the mission may be compromised by a cyber attack.

There is still a need for an implementation of mission-centric decision support and impact assessment. The existing works on impact assessment do not consider the mission perspective and focus on IT components. The response selection model (REASSESS) [6] allows mitigating network-based attacks by incorporating a response selection process that evaluates negative and positive impacts associated with each countermeasure. The considered negative effects are the disturbance of the service caused by the action, which takes into account the importance of the service and the level of disturbance. The considered positive effects are the response success rate for a given response and alert. The concept of response success rate does not require any deep theoretical analysis of dependencies and is based purely on the experience, however very imprecise in practice since networks change too often to derive any meaningful historical data. Shin et al. [7] developed a cybersecurity risk model based on Bayesian networks that represents the probability of cyber-attacks on IT systems. Recently, Huang et al. [8] proposed a framework for assessing the impacts of cyber attacks in cyber-physical systems. The risk assessment is based on predicting the probability of sensors and actuators being compromised from conditional probabilities captured in a Bayesian network. The probability distribution is then used by a stochastic hybrid system model to predict the possible evolution of the physical system.

III. Decision Support Process

The goal of the decision support process is to suggest the most resilient mission configuration for an ongoing attack endangering established mission security requirements. In the context of this article, we consider the *mission* as a collection of arrangements of mission-supportive processes enabling to deliver the resulting mission's functionality. A *mission-supportive process* is an essential activity, delivered by people through cyber components, supplying particular desired functionalities. Our approach stands on the following four assumptions. First, the mission usually allows more configuration alternatives that meet all functional requirements. The mission can be supported by various combinations of supportive processes, IT services, and cyber components to avoid a single point of failure in critical systems. Each combination that enables the mission operation is called *mission configuration*. The different mission configurations can be derived from the AND/OR relations between the mission and its dependencies [5], [9]. Second, the mission supportive processes are regarded as critical assets to be protected. Primarily, we focus on processes instead of supportive cyber components (network hosts and services). The critical supportive cyber components, vulnerabilities discovered on the components, and relevant interactions are derived consequently in the context of the specific mission configuration. Third, the mission's requirements on confidentiality, integrity, and availability (CIA), also referred to as *mission security requirements*, are maximized while satisfying the required mission's functionality. Keeping the mission operational means that all the functional requirements are fully satisfied. Finally, the decision making support is built on the mission's ability to adaptation and reconfiguration in the face of adversary activities.

Our approach captures both interrelated aspects of the analyzed mission: its functional requirements, and its security requirements. All the functional requirements of the mission must be satisfied all the time in order to keep the mission operational. First, we need to identify and decompose the missions and define their functional and security requirements. This is done only once unless the missions change. Subsequently, we continuously react to changes in the current security state and recalculate mission resilience. This consists of updating the attacker's position and discovering vulnerabilities, attack vectors in the network, and creating Bayesian attack graphs (BAG) to infer the most probable threats to missions. Finally, decision making takes place. The steps of the process are discussed in detail in the following subsections.

A. Mission Decomposition and Definition of Requirements

The first phase of the proposed process is mission decomposition. We covered the first phase in detail in our previous work [9]. Briefly, the mission decomposition model is a graph where nodes are mission supportive processes, IT services, and cyber components. Mission supportive processes represent the missions and their requirements on CIA attached to them as a numerical value. IT services are abstract representations of particular components of the mission supportive processes. Supportive cyber components are mapped to particular hosts and services in the network. The directed edges in the graph are dependencies (functional requirements) between the mission supportive processes and IT services, and IT services and cyber components. Further, there might be AND/OR nodes and edges between them to model more complex dependencies. For example, if a mission supportive process depends on two IT services, there is an extra AND node, and the edges go from the mission supportive process to the AND node, and from the AND node to the two IT services. The same

principle applies to OR nodes and combinations of nodes. The mission decomposition model is built after a consultation with a domain expert or administrator of the mission or the critical systems. The requirements on CIA must be established at the level of individual processes. Their settings are the responsibility of the domain experts.

B. Change of Security State and Resilience Calculation

Resilience calculation reacts to a change in the security situation, e.g., a discovery of a new vulnerability in the network and intrusion detection that discloses an attacker's position or updates an attacker's privileges. Discovery of a vulnerability on a supportive cyber component poses a new threat to the component and the IT services and mission supportive processes that depend on it. Attacker's position and privileges give us information about the possible action the attacker may take. The position implies the attacker's reach, i.e., the supportive cyber components that are reachable by the attacker. The attacker's privileges are the privileges the attacker has on a controlled system. Such information is mostly provided by third-party tools, such as vulnerability scanners and intrusion detection systems.

We employ two principles of recalculation of the mission's resilience. First, we calculate the resilience for every individual mission configuration. We calculate how likely can a particular mission's configuration be affected, i.e., the probability of its successful disruption in terms of endangering established security requirements. Second, we use the probability as a measure. The probability function assigns a non-negative real number to every mission configuration that satisfies established functional requirements. The number represents the probability of successful disruption of established security requirements. The most resilient configuration is the one with the lowest probability of disruption.

Based on parameters derived from the mission decomposition model, we employ a logical attack graph to represent the current security situation, i.e., the position and privileges of the attacker and enumeration of supportive cyber components and their vulnerabilities. Subsequently, we employ the statistical inference mechanism of a graphical probabilistic model (Bayesian network) that reflects the situation. The resulting Bayesian network provides the capabilities of the inference mechanisms that allows for calculating the probability of an attacker reaching the target privilege, i.e., the probability of disruption of a particular security requirement. The mathematical background of both the above-mentioned tools will be described in the following sections.

In our work, the selection of possible mission configurations and the most resilient mission configuration is considered as a constraint satisfaction and optimization problem [9]. Constraints satisfaction represents the satisfaction of functional requirements. The formal expression of mission decomposition model (Boolean formula or Constraint AND/OR tree) enables an automatic generation and derivation of all mission configurations satisfying established functional requirements, i.e., constellations of mission supportive processes, IT services, and cyber components and their interactions. Constraints optimization represents the optimization of CIA requirements. We compare the probabilities of disruption of each satisfying mission configuration. The calculation of the probability of disruption of a supportive cyber component is using an exploitability score of a vulnerability. Such scores are provided, for example, in CVSS[1] scores and metrics. In order to keep the mission operational, we maximize the mission's security requirements while fully satisfying its functionality requirements.

C. Decision Making

When the mission resilience is recalculated, the process comes to its final stage, the decision making. The output of the previous stage of the process is the calculation of risks for all the configurations of the mission's supportive cyber components. In this stage, there is a need to select the most resilient mission configuration. For each mission configuration, we consider the worst attack scenario, i.e., the scenario with the highest probability of a successful exploit. The configuration with the lowest computed probability is selected as the most resilient. However, there are three security requirements (CIA) that may be affected by a vulnerability. If the exploitation affects only one or two of the requirements, we have to employ a utility function that would produce a single value out of three probabilities. There are numerous alternatives to choose a utility function. Simple, yet reasonable, options would be to use the sum or maximum of the three values calculated for CIA. In addition, the utility function may be redefined based on past experience or priorities of the stakeholders, which could regard some factors in CIA more important than others.

With a defined utility function, we get a single comparable value for every mission configuration's worst-case scenario. Thus, we can select the most resilient configuration and put it on the output of the decision support process. Nevertheless, it is up to administrators and other stakeholders to take the final decision and reconfigure the network. Automation of network reconfiguration and incident response is out of the scope of this work.

IV. Implementation of the process

In this section, we bring insights into the implementation of the decision support process. First, we take a look at how to create an attack graph [10] and how to extend it using the Bayesian networks [11]. Subsequently, we provide algorithms that constitute the process.

A. Attack Graph

Attack graphs are popular tools for representing cyber attacks and related phenomena [10]. For example, they may depict all the possible paths from the attacker's initial position to the desired target using the exploitation of vulnerabilities present on the hosts in the network and constituting a mission. In our concept, we employ a so-called logical attack graph (LAG) [12]. LAG is defined as a directed bipartite graph representing dependencies between exploits

[1]Common Vulnerability Scoring System, https://www.first.org/cvss/

and privileges (security conditions), formally: $(Exploits \cup Privileges, Prerequisities \cup Postrequisities)$.
The exploits and privileges constitute the set of vertices, and prerequisites and postrequisites constitute the set of directed edges of the graph. Prerequisites represent privileges allowing exploitation of the relevant vulnerability. Postrequisites represent privileges resulting from a successful exploit of the relevant vulnerability. Formally:

$Prerequisities \subseteq Privileges \times Exploits$

$Postrequisities \subseteq Exploits \times Privileges$

The above representation is based on an assumption of the so-called monotonicity principle [12], i.e., the attacker does not relinquish privilege once obtained. The importance of this principle is that the attack graph results in a directed acyclic graph (DAG). To build an attack graph for specific mission configuration and specific attacker's position, we need *i)* mission decomposition model with the constellation of supportive cyber components and their interactions, *ii)* knowledge about the vulnerabilities in the network, namely on supportive cyber components, and *iii)* attacker's position, privileges, and goals. We chose MulVAL (Multihost, multistage, Vulnerability Analysis) [13] as the most convenient tool to generate an attack graph. MulVAL can automatically process vulnerability specification from publicly available sources. The complexity of the algorithm for building an attack graph is polynomial. The resulting attack graph provides information about causality relationships among involved security entities. The causality relationship identification provides qualitative (unmeasurable) information that is necessary for subsequent quantification of the security state.

B. Bayesian Network

We have employed a Bayesian network (BN) abstraction as a tool for modeling uncertainties of the cybersecurity situation and performing the decision support analysis. BN can be formally defined as $BN = (DAG, Q)$ where DAG is directed acyclic graph, nodes represent random variables; arcs represent conditional (in)dependencies among these random variables, and Q represents the Conditional Probability Distribution (CPD) for each random variable.

In general, due to its intrinsic complexity, it is not straightforward, nor trivial, in any case, to capture all the relevant uncertainty aspects. The goal of the analytical process is to describe the probability distribution of random variables representing the attacker's target privileges, i.e., the privileges attacker pursues. The Joint Probability Distribution (JPD) gives us the desired quantitative picture of the whole situation and covers all aspects of related random variables necessary to answer any question. Certain assumptions about conditional independence among the random variables must be met for a BN to represent this distribution properly. Each random variable must be conditionally independent of all its non-descendants in the graph given its parents. In other words, knowing the security state of the parents, any additional information about the other variables gives us no information about the mentioned variable. These assumptions are naturally met as a result of the character of the attack graph.

The attack graph gives us prior knowledge about relevant causal relationships to build a DAG that forms the BN. We need to calculate the conditional probability distributions (conditional probability tables) for individual nodes (discrete random variables) of the BN. The appropriate metric must be interpretable as a conditional probability. Parameters for the calculation are derived automatically from trustworthy sources, such as NVD. We use the exploitability metrics of CVSSv3 to estimate the probabilities of successful exploits.

Two basic situations result from the attack graph. The corresponding parents' nodes are in the relation of logical AND, which means that all the prerequisites for successful exploit must be met, or the corresponding parents' nodes are in the relation of logical OR, which means that at least one of the prerequisites for successful exploit must be met. Calculation of probability corresponding to the logical AND can be formally expressed as:

$$p(X_i|parents(X_i)) = \prod_E p(e_i) \tag{1}$$

in the case when all the prerequisites $X_i = True$, otherwise the probability is equal to 0. Calculation of probability corresponding to the logical OR can be formally expressed as:

$$p(X_i|parents(X_i)) = 1 - \prod_E (1 - p(e_i)) \tag{2}$$

in the case when at least one of the prerequisites $X_i = True$, otherwise the probability is equal to 0. The $p(e_i)$ represents the probability of successful exploit e_i connecting X_i with its parents in the corresponding attack graph, and E represents the set of these nodes.

The BN enables the compact representation of JPD using the following equation:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i|parents(X_i)) \tag{3}$$

For the derivation of resilience metric, we need to calculate the probability of the successful exploit causing the attacker will reach mission-critical privileges. Random variables representing the critical privileges will be marginalized. As the JPD gives us a complete picture of the situation, the marginalization enables us to quantify the desired aspect, i.e., the probability expressing that an attacker reaches desired privilege. Calculation of the desired unconditional probability, i.e., the quantity from the distribution we are interested in, can be formally expressed as:

$$p(X_a) = \sum_{(X_1,\ldots,X_{a-1},X_{a+1},\ldots,X_n)} \prod_{i=1}^{n} p(X_i|parents(X_i)) \tag{4}$$

The BAGs and equations listed here are used in the implementation of the analytical process as described in the following subsection.

C. The Algorithm of the Process

Herein, we provide the algorithms behind the decision support process discussed in the previous section. The main function of the process is depicted in Algorithm 1 as the $analytical\_process$ procedure. First, we will get the possible configurations for each mission in the $mission\_file$. For each configuration, we get its cost. The cost of configuration is the maximal cost among raw_costs on which the $utility\_function$ was applied. Raw cost is a triple of probabilities for CIA. The best configuration for the mission is the one with the minimal cost.

The $mission\_file$ is the main input for the process. This JSON file represents constrained AND/OR tree decomposing mission into supportive processes, IT services, and cyber components [9]. Thus, we serialized the trees to JSON files for the purposes of experimental implementation. The $get\_possible\_configurations()$ procedure reads the mission description from the input file in search for possible configurations. The configuration consists of cyber components that together must be running in order to support the requirements of a mission and, thus, are in the AND logical relationship.

The way how we get all possible configurations is as follows. For each vertex from the root of the tree (root is mission node – there are multiple roots) traverse the subtree. If we approach the AND node, we add all of its successors to the list of IDs that will be further processed because all of them must be in the configuration. On the other hand, if we approach the OR node, we will recursively process all of its successors separately because one of them must be in the configuration (and we do not want to have redundant configuration, so we use only one subtree of such OR node in one configuration).

The $generate\_mulval\_input()$ function and $generate\_attack\_graph()$ procedure prepare and run the processing of the inputs by the MulVAL tool. The analytical process also requires the data from the environment, such as a list of annotated vulnerabilities and enumeration of hosts and vulnerabilities in the network. We used a database based on a data model for cyber situational awareness proposed in our previous work [14]. The database structures the data from various tools to a graph structure. The vulnerabilities in CVE format[2] are taken from NVD[3], including the CVSS scores that are used in the calculations. The presence of vulnerabilities in the network is checked using common vulnerability scanners. Attacker's position is inferred from intrusion detection alerts processed by a CSIRT team in a request tracking software that exports the alerts into the database. The experimental implementation of the process is, thus, independent of primary data sources, and only requires the inferred data to be stored in the aforementioned database. The attack goals (required by MulVAL) are potential losses of CIA in the cyber components due to the exploitation of a vulnerability.

[2]Common Vulnerabilities and Exposures, https://cve.mitre.org/

[3]National Vulnerability Database, https://nvd.nist.gov/

Fig. 1. Algorithm of the analytical process.

```
1  procedure analytical_process(mission_file)
2      configurations ← get_possible_configurations(mission_file)
3      for mission in configurations do
4          for configuration in configurations[mission] do
5              configuration_cost ← 0
6              for goal_component in configuration do
7                  generate_mulval_input(goal_component, configuration)
8                  generate_attack_graph()
9                  raw_cost ← create_BAG()
10                 actual_cost ← utility_function(raw_cost)
11                 if actual_cost > configuration_cost then
12                     configuration_cost ← actual_cost
13         best_configuration ← take_configuration_with_minimal_cost()
14 procedure create_BAG()
15     if no attack paths then return (0.0, 0.0, 0.0)
16     incidence_list ← ARCS.CSV
17     vertex_set ← VERTICES.CSV
18     model ← BayesianModel()
19     for edge in incidence_list do model.add_edge(edge)
20     for node in vertex_set do
21         switch node do
22             case LEAF do
23                 if "vulExists" in node_description then
24                     cpd ← TabularCPD(variable=node_id,
25                           variable_cardinality=2,
26                           values=[[1 - exploitability], [exploitability]] )
27                     model.add_cpds(cpd)
28                 else
29                     cpd ← TabularCPD(variable=node_id,
30                           variable_cardinality=2,
31                           values=[[0], [1]] )
32                     model.add_cpds(cpd)
33             case AND do
34                 cpd ← TabularCPD(variable = node_id,
35                       variable_cardinality = 2,
36                       values=[[1, ..., 1, 0.2], [0, ..., 0, 0.8]],
37                       evidence = predecessors,
38                       evidence_cardinality=[2, ..., 2] )
39                 model.add_cpds(cpd)
40             case OR do
41                 cpd ← TabularCPD(variable=node_id,
42                       variable_cardinality=2,
43                       values = [[1, ..., 1, 0], [0, ..., 0, 1]],
44                       evidence=predecessors,
45                       evidence_cardinality=[2, ..., 2] )
46                 model.add_cpds(cpd)
47     return infer_probabilities(model)
```

When the MulVAL tool processes the inputs and generates the attack graphs, we proceed with the construction of a BAG for the final decision, as denoted in the $create\_BAG()$ procedure in Algorithm 1. The algorithm first reads the files $ARCS.CSV$ and $VERTICES.CSV$ that were created by previous procedures as outputs of MulVAL and contain the vertices and edges of the generated attack graph. The attack graph generated by MulVAL contains nodes with three types of labels: LEAF, AND and OR. The exploits discussed in Section IV are the nodes with the AND label. Prerequisite nodes are either leaves (label LEAF) or not (label OR). When all the nodes and edges are loaded into a model, the algorithm continues with the CPD calculation as follows. The nodes expressing the existence of a CVE are given the probability of success equal to the exploitability score from CVSS. These nodes always have the LEAF label, and their description

starts with *vulExists*. Other nodes with label LEAF have the probability of being TRUE equal to 1.0 (the same as in MulVAL). The nodes with AND label have CPD table with $2^{number\_of\_predecessors}$ rows and all of them have a conditional probability for TRUE equal to 0.0, but the last one where all of the parents are TRUE has conditional probability of being TRUE equal to 0.8 (this probability of successful exploit is taken from MulVAL [13]). If a node has the OR label, then it joins two paths in a graph. The number of rows in the CPD table is $2^{number\_of\_predecessors}$. All of the rows have the probability of success equal to 1, but the row where all of the parents are FALSE has 0 probability of success. The $Tabular\_CPD()$ function constructs the CPD tables for the nodes in the BAG for later computations.

The $create\_BAG()$ procedure in Algorithm 1 implements the equation introduced in Section IV-B. Namely, equation (1) is used in AND switch case on line 35 and equation (2) in OR switch case on line 42. Further, it uses the JPD explained in equation (3) and calculation of unconditional probability explained in equation (4) in $infer\_probabilities()$ function once the BAG is built. In our experimental implementation, the $infer\_final\_probability()$ function calls pgmpy [15], which implements inference over Bayesian model. The inference of final unconditional probability is computed for each type of goal (CIA) separately using equation (4).

When we go back to the Algorithm 1, we need to calculate the actual cost for the combination of probabilities related to the CIA using the utility function. In the experimental implementation, the $utility\_function$ is a sum. However, the utility function may be suited to fit the needs and priorities of the organization. To select the most resilient network configuration, i.e., the configuration with the minimal cost, we use the $take\_configuration\_with\_minimal\_cost()$ procedure. The selection of the most resilient configuration terminates the process, and it is up to the operators to reconfigure the network, which is out of the scope of this work.

V. Conclusion

We described a mission-centric approach to decision support for network security management that allows for calculating and selecting the most resilient mission configuration concerning mission requirements and current threats. The selection of the most resilient network configuration may be achieved by selecting the configuration with the lowest probability of mission disruption through the exploitation of its supportive cyber components in terms of confidentiality, integrity, and availability. While mission modeling and process overview were outlined in our previous work [9], here we described the remaining phases of the decision support process, i.e., calculating mission impacts and selecting the resilient configurations using attack graphs and Bayesian networks. Within the description, we pinpointed the most intriguing parts of the process, such as fundamental equations for the computation and potentially computationally intensive tasks.

In our future work, we are going to evaluate the analytical process in operational settings to infer the computation times and reliability of the calculations with respect to real-world missions and network configurations. We expect the evaluation in a live environment to be a challenging problem due to the need for modeling a sufficient number of illustrative missions. The results will also depend on the current security situation and the capabilities of the network operators to discover vulnerabilities.

Acknowledgement

This research was supported by the Security Research Programme of the Czech Republic 2015 - 2020 (BV III / 1 VS) granted by the Ministry of the Interior of the Czech Republic under No. VI20172020070 Research of Tools for Cyber Situational Awareness and Decision Support of CSIRT Teams in Protection of Critical Infrastructure.

References

[1] A. Kott, C. Wang, and R. F. Erbacher, *Cyber defense and situational awareness*. Springer, 2015, vol. 62.

[2] F. Silva and P. Jacob, "Mission-centric risk assessment to improve cyber situational awareness," in *Proceedings of the 13th International Conference on Availability, Reliability and Security*. ACM, 2018, p. 56.

[3] X. Sun, A. Singhal, and P. Liu, "Who touched my mission: Towards probabilistic mission impact assessment," in *Proceedings of the 2015 Workshop on Automated Decision Making for Active Cyber Defense*. ACM, 2015, pp. 21–26.

[4] J. Guion and M. Reith, "Cyber terrain mission mapping: Tools and methodologies," in *2017 International Conference on Cyber Conflict (CyCon US)*. IEEE, 2017, pp. 105–111.

[5] G. Jakobson, *Mission Resilience*. Cham: Springer International Publishing, 2014, pp. 297–322.

[6] S. Ossenbühl, J. Steinberger, and H. Baier, "Towards automated incident handling: How to select an appropriate response against a network-based attack?" in *IT Security Incident Management & IT Forensics (IMF), 2015 Ninth International Conference on*. IEEE, 2015, pp. 51–67.

[7] J. Shin, H. Son, R. K. ur, and G. Heo, "Development of a cyber security risk model using bayesian networks," *Reliability Engineering & System Safety*, vol. 134, pp. 208 – 217, 2015.

[8] K. Huang, C. Zhou, Y. C. Tian, S. Yang, and Y. Qin, "Assessing the physical impact of cyberattacks on industrial cyber-physical systems," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 10, pp. 8153–8162, Oct 2018.

[9] M. Javorník, J. Komárková, and M. Husák, "Decision support for mission-centric cyber defence," in *Proceedings of the 14th International Conference on Availability, Reliability and Security*, ser. ARES '19. New York, NY, USA: ACM, 2019, pp. 34:1–34:8.

[10] C. Phillips and L. P. Swiler, "A graph-based system for network-vulnerability analysis," in *Proceedings of the 1998 Workshop on New Security Paradigms*, ser. NSPW '98. New York, NY, USA: ACM, 1998, pp. 71–79.

[11] A. A. Ramaki, M. Khosravi-Farmad, and A. G. Bafghi, "Real time alert correlation and prediction using Bayesian networks," in *Information Security and Cryptology (ISCISC), 2015 12th International Iranian Society of Cryptology Conference on*. IEEE, 2015, pp. 98–103.

[12] X. Ou, W. F. Boyer, and M. A. McQueen, "A scalable approach to attack graph generation," in *Proceedings of the 13th ACM Conference on Computer and Communications Security*, ser. CCS '06. New York, NY, USA: ACM, 2006, pp. 336–345.

[13] X. Ou, S. Govindavajhala, and A. W. Appel, "MulVAL: A Logic-based Network Security Analyzer," in *Proceedings of the 14th Conference on USENIX Security Symposium - Volume 14*, ser. SSYM'05. Berkeley, CA, USA: USENIX Association, 2005.

[14] J. Komárková, M. Husák, M. Laštovička, and D. Tovarňák, "CRUSOE: Data Model for Cyber Situational Awareness," in *Proceedings of the 13th International Conference on Availability, Reliability and Security*, ser. ARES 2018. New York, NY, USA: ACM, 2018, pp. 36:1–36:10.

[15] A. Ankan and A. Panda, "pgmpy: Probabilistic graphical models using python," in *Proceedings of the 14th Python in Science Conference (SCIPY 2015)*, 2015.