Designing a Methodological Framework for the Empirical Evaluation of Self-Protecting Systems

Andrea Montemaggio*, Stefano Iannucci†, Tanmay Bhowmik†, John Hamilton*
*Center for Cyber Innovation, †Computer Science and Engineering
Mississippi State University
Mississippi State, MS

***Abstract*—Increasingly, cyber attacks against enterprises and governments make use of automated tools. For this reason, and given the importance of a timely protection, in the last decade there has been a push in researching methodologies to automate the full defense life-cycle of computer systems. The two core phases of this life-cycle are Intrusion Detection and Intrusion Response. However, while some progress has been done on the former, the latter is still at an early stage. This is due to several factors, among which the lack of a standardized methodology for the validation and comparison of Intrusion Response methodologies. In this paper, we attempt to fill this gap by introducing a methodological framework for the quantitative empirical evaluation of self-protecting systems, based on the metrics of response time and cost. An experimental design is also provided and its applicability is illustrated by the means of a template experiment.**

I. Introduction

Increasingly, cyber attacks against enterprises and governments make use of sophisticated automated tools [1]–[3]. This poses new challenges to computer security, system administrators and Security Operation Center (SOC) operators who must plan and deploy effective countermeasures in a timely manner. Security operators must protect systems with little to no automation and decision-making support [4].

Despite efforts to define a formal Incident Response (IR) life-cycle management process, such as the *Computer Security Incident Handling Guide* [5] developed by the National Institute of Standards and Technology (NIST), very little has been done regarding the automation of the entire IR life-cycle. The two core phases of the framework realized by NIST are: *Detection and Analysis* and *Containment Eradication and Recovery* which, from the automation stand-point, can be mapped to two broad areas of research, namely, *Intrusion Detection* and *Intrusion Response*. However, while a plethora of works are focused on the former area, only a few address the problem of automating the response to an ongoing attack, and the automation of both phases is crucial for the realization of a self-protecting system. A comprehensive survey of frameworks that are able to automatically devise and apply countermeasures to cyber attacks in a semi-automated or fully-automated fashion can be found in [4].

According to [4], one of the characterizing aspects of these frameworks is the type of reaction they are capable of providing. A framework providing a *static* reaction is essentially a risk assessment tool that can be useful to find vulnerabilities of a system at design time, or even at run time if it includes penetration test capabilities, but is of little to no use to counter an ongoing cyber attack. On the other hand, a framework which is capable of *dynamic* reactions is able to actively help system administrators to defend a running system.

In any case, before being ready to operate in a given environment, self-protecting systems need to be fed with some sort of configuration, knowledge-base, model, or training data that are dependent on the target environment. The nature of these configurations vary and their complexity ranges from a simple static table mapping attack signatures to defense actions, to trained Machine Learning models. However, this configuration must be provided, ultimately, by humans and its quality inescapably affects the run-time behavior of the system. For instance, a wrong or incomplete configuration of the static mapping table, the omission of actions in the system or attacker model, as well as inadequate training data would have an equally detrimental effect on the self-protection capabilities. Furthermore, every model is different and captures different aspects of the environment, such as the topology of the system to protect and its behavior, the behavior of the attacker, or even the activity of legitimate users. As a result, it is unlikely that different self-protecting systems can be directly comparable.

In other words, we argue that the representational power of these environment models, in conjunction with the human factors involved in crafting them for a given operational setting or even influencing the run-time as in the *human-in-the-loop* paradigm for critical systems [6], have a non-negligible impact on the performance of self-protecting systems. In this perspective, we propose an empirical methodology for the effectiveness evaluation of a self-protecting system that takes into account both run-time and configuration-time aspects, together with their associated human factors.

However, the quantitative evaluation of a self-protecting system requires the following elements that still need to be defined and standardized: (i) a shared set of measures

978-1-7281-8414-2/20/$31.00 ©2020 IEEE
DOI 10.1109/ACSOS-C51401.2020.00059

and their operationalization, (ii) a methodology to collect, analyze and evaluate data, and (iii) a reference system running in (iv) a standardized cyber-range.

In this respect, we attempt to address (i) and (ii) by designing a methodological framework for the empirical and quantitative evaluation of self-protecting systems based on the Monitor, Analyze, Plan, Execute (MAPE) reference framework for autonomic computing [7]. Specifically, the present work aims to answer the following research question with an empirical study.

> *How to quantitatively evaluate the effectiveness of a self-protecting system in responding to a security incident?*

This paper is organized as follows. Firstly, Section II addresses (i) by introducing a set of measures and their operationalization. Secondly, our evaluation methodology covering (ii) is presented in Section III and an experimental design is proposed in Section IV. Then, Section V illustrates the application of the methodology through a hypothetical experiment and addresses data collection and analysis issues. Finally, an elucidation of the threats to validity of the proposed methodology follows in Section VI, while Section VII discusses related works and Section VIII concludes the paper.

II. Measures for Incident Response Effectiveness

While the importance of temporal measures for the quantitative characterization of the security properties of information systems and cyber operations is widely recognized [8]–[10], alternative models, such as cost-sensitive models [11] and game-theoretic models [12], have also been proposed. In this section, we give the operational definition of three temporal measures, depicted in Figure 1, and one cost measure characterizing incident response (IR) cyber operations.

However, before delving into these definitions, we need to introduce the notion of *security policy*. According to NIST [13], «security policies define the objectives and constraints for the security program. Policies are created at several levels, ranging from organization or corporate policy to specific operational constraints (e.g., remote access). In general, policies provide answers to the questions "what" and "why" without dealing with "how". Policies are normally stated in terms that are technology-independent». We restrict this general definition to the scope of self-protecting systems and regard a security policy as an arbitrarily complex Boolean expression defined over the observable state variables of a system, which is satisfied iff the system is secure.

Attack Duration (AD): We define this measure as the duration of a malicious activity, regardless of the reason for its termination (e.g., the attack was successful, the attack was stopped by a defense action, etc.). When the malicious activity leads to a successful attack, this metric is commonly referred to as Time To Compromise (TTC), which is agreed to be a valid measure of the security of a system [9], [10].

Time To Detect (TTD): The amount of time needed to detect that a malicious activity is taking place on a system, after said activity has started. In the context of a self-protecting system, we define this measure as the amount of time that it takes to detect a violation of the security policy, after the malicious activity has started. In general, the TTD metric corresponds to the time needed to complete the *Detection and Analysis* phase defined by the NIST IR framework. In particular, for self-protecting systems based on MAPE, this metric corresponds to the time needed to complete the *Monitor* and *Analyze* phases.

Time To Respond (TTR): The amount of time needed to respond to a security incident until full remediation, after the detection of the malicious activity causing it has occurred. For a self-protecting system, we define this measure as the amount of time needed to bring the system into a state where the security policy is satisfied, after a violation has been detected. In general, the TTR metric corresponds to the time needed to complete the *Containment, Eradication, and Recovery* phase described in the NIST IR framework. In particular, for self-protecting systems based on the MAPE loop, this metric corresponds to the time needed to complete the *Plan* and *Execute* phases. However, for critical self-protecting systems, sometimes the concept of *human-in-the-loop* [6] is introduced, with the aim of having a human expert verify the automatically computed defense strategy before its execution. For this reason, we break down the TTR as follows: Planning Time (PT), which measures the amount of time a self-protecting system spends planning a defense policy (MAPE *Plan* phase), after a security policy violation has been detected; Thinking Time (TT), which measures the amount of time a human operator needs to evaluate and eventually acknowledge a defense plan; Execution Time (ET), which measures the amount of time spent in the execution of the defense policy on the actual system (MAPE *Execution* phase).

Cumulative Response Cost (CRC): We define this measure as the total cost of executing a sequence of response actions to resolve the incident, according to a given cost function. Let $L(a) : A \mapsto \mathbb{R}$ be an arbitrary cost function that associates a cost to the execution of a response action, where A is the set of all the possible response actions that can be executed on a given system. Being $P : [1, N] \mapsto A$ a finite sequence of actions $(a_i)_{i=1}^{N}$ representing a response plan of length N, we can formally define the CRC for the plan P as:

$$CRC(P) = \sum_{i=1}^{N} L\big(P(i)\big) \tag{1}$$

As common in the domain of automated intrusion response and with the objective of simplifying the formulation of the problem, we assume that the cost of executing

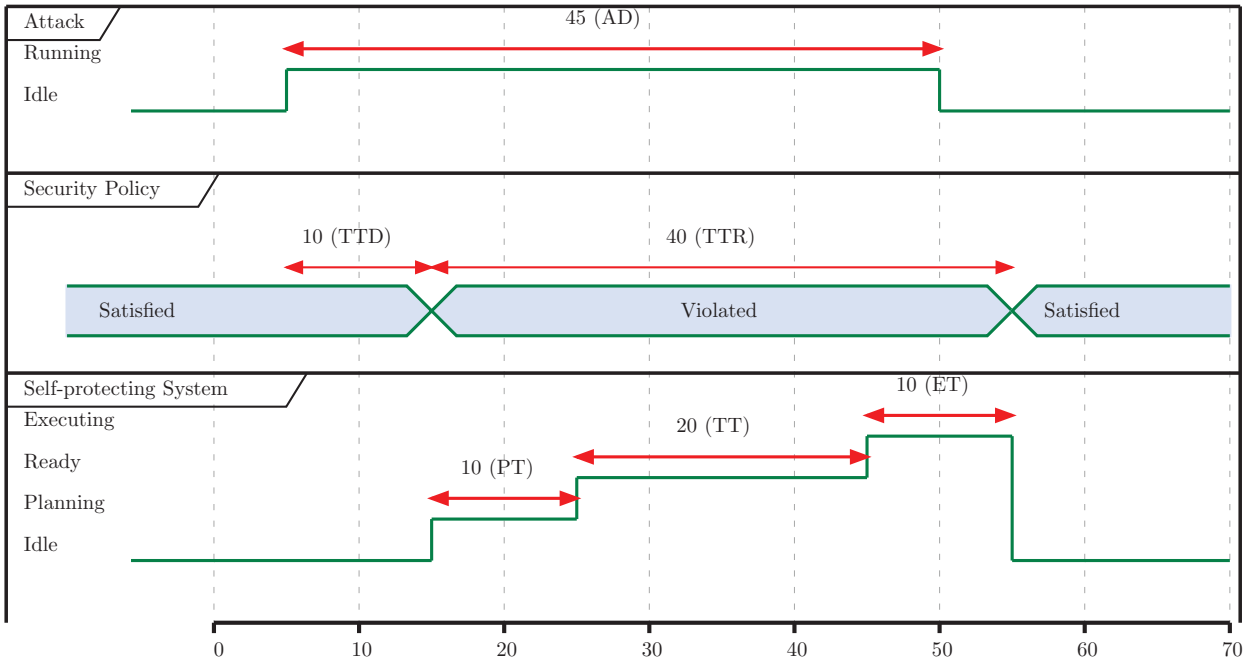

Fig. 1: A timing diagram showing the significant state changes happening throughout a generic security incident occurring to a self-protecting system (time unit is not relevant). All the temporal measures defined in Section II are represented here.

an action does not depend on the state in which the action is executed [14]–[17].

III. Methodology

According to the autonomic computing reference architecture specified in [18], an autonomic system comprises an *autonomic manager*, which acts as the controller for a *managed artifact*. In the context of this study we refer to the self-protecting system we want to evaluate as the composition of these two units. Furthermore, we assume the managed artifact to be an arbitrary system and we will refer to it as the *managed system*.

As stated in Section II, two prominent models to quantify the security of a system are based on time and cost, both from an offensive and defensive perspective. As a consequence, we choose these two dimensions to characterize the effectiveness of a self-protecting system.

In addition to defining effectiveness measures and their operationalization in the context of self-protecting systems, we design an experiment based on a Controlled Cyber Defense eXercise (CDX) [19], [20] involving participants sampled from a population of subjects having previous experience in participating in a CDX.

More formally, according to our methodology, the effectiveness evaluation of a given self-protecting system is conducted through an empirical study aimed to test the following research hypotheses:

H1: The Mean Time To Respond (MTTR) measured for the experimental group using the given autonomic manager to respond to a cyber attack carried against the managed system is lower than the MTTR measured for the control group.

H2: The Mean Cumulative Response Cost (MCRC) measured for the experimental group using the given autonomic manager to respond to a cyber attack carried against the managed system is lower than the MCRC measured for the control group.

Finally, we assume that the experimental and control groups both defend an instance of the same managed system. However, the control group is not equipped with an autonomic manager taking care of responding to the cyber attack, thus the response for this group is carried out manually, namely, with a null autonomic manager. While it is possible to use a different autonomic manager for the control group than the null manager, this would unnecessarily complicate the experiment. However, this assumption does not prevent individual studies from comparing different autonomic managers with each other. In this regard, for instance, statistical techniques such as Adjusted Indirect Comparison or Mixed Treatment Comparison [21] can be used in a meta-analysis of multiple empirical evaluation studies conducted with the same methodology.

IV. Experimental Design

A managed system containing purposely vulnerable components is prepared, documented and multiple instances of it are deployed. The experiment participants

play as the blue team in the CDX with the objective to defend this system from a set of 10 automated attacks.

The experiment we design follows an approach based on repeated measures with matched pairs [22]. Participants are randomly assigned to two groups: an experimental group EG having access to the autonomic manager we want to evaluate, and a control group CG. In order to mitigate the potential confounding effect of different levels of cyber security expertise among the subjects, which has been observed in other studies such as [23], a proficiency test is administered to the participants and matched pairs are formed according to the test scores. Then, each individual in a pair of participants with a comparable score is randomly assigned to EG or CG.

Specialized training could be required to configure and use the autonomic manager and the security tools deployed along with the managed system such as monitoring systems, IDSs, and IPSs. Therefore, all the participants are given the documentation of the managed system and trained on both the autonomic manager and the security tools at the very beginning of the study, regardless of their subsequent assignment to EG or CG. Should the autonomic manager require its definition, the training also includes the development of a model of the managed system, given its documentation and without any knowledge of its vulnerabilities. In particular, the participants are asked to anticipate the set A of the possible response actions for the given managed system and provide, for each $a \in A$, an estimation of the cost function $L(a)$ and a high-level functional description of the action.

Furthermore, a shared security policy derived from the business requirements of the managed system is provided to the participants of the experiment, in order to establish an unambiguous definition of security for the given managed system.

After the groups are formed, parallel instances of the CDX are run, one for each participant. Given an automated exploit script for each vulnerable component in the system, a random sequence of exploits is run on each CDX instance and repeated measures of TTR and CRC are taken. However, during the CDX a participant may need to manually perform a custom action that was not anticipated in the model of the managed system. In this case, the participant is required to provide a high-level description of the action by filling a form at the end of the CDX session. Afterward, all the collected extra actions are presented to all the participants, anonymously, to have an estimate of the value of the cost function assigned. Finally, in order to mitigate the potential effect of the different experience level among the participants, only the estimate given by the other participant in the same matching pair is considered.

V. Experiment Template

In this section, we illustrate the application of our methodology by the means of a template experiment following the design described in Section IV. Such experiment aims to evaluate the effectiveness of an autonomic manager (AM) with an empirical study involving 20 Computer Science and Engineering graduate students with previous experience in participating to a CDX. The AM is compared to a semi-manual protection of the managed system: participants in the EG are equipped with the AM, while participants in the CG are provided with the managed system instrumented with a Security Information Event Management (SIEM). Similarly to most commercial security solutions, the SIEM provides intrusion detection and event correlation functionalities supporting the *Detection and Analysis* phase of the NIST framework.

In the following, we define a plan to collect data in such setting and describe the suggested analysis techniques to apply.

A. Data Collection Plan

The experiment requires data to be collected from a variety of sources: (i) the cyber security proficiency test, (ii) the set of the anticipated response actions and the corresponding values of the cost function, (iii) the TTR for each attack, (iv) the defense plan and its CRC for each attack, (v) any unanticipated response action run to defend the system, and (vi) an estimate of the value of the cost function for these additional actions.

The proficiency test (i) is administered at the beginning of the study and comprises 20 multiple choice questions about cyber security principles and practices. The relevant data collected is the test score for each participant.

The set of actions and their cost values (ii) are collected for each participant through a form to be filled as a result of an initial assessment of the managed system. In the case the AM under evaluation requires a formal model of the response actions that includes a cost model, this data can be extracted automatically from the model built by the participants in the training phase.

While for the EG the TTR measure for each attack (iii) can be obtained from the AM that continuously monitors whether the security policy is satisfied or not, for the CG the response-start time is retrieved from the alert produced by the SIEM and the response-end time from the logs of an external monitoring system that continuously check the business requirements.

A research assistant taking notes while the CDX is running helps with the collection of the response plan (iv) both for the EG and the CG. For the EG, the assistant keeps track of any extra response action (v) the participant needs to run, whilst we extract from the AM the actions that are planned and run automatically and merge these two data sets to obtain the actual response plan. Instead, to collect the plan for the CG, we rely only on note-taking. Additionally, although the utility of note-taking for the EG is marginal, having the assistant present in both groups avoids that any potential influence on the participant could affect the results.

As outlined in Section IV, the collection of the cost value for the extra actions (vi) requires two steps. First, at the end of the CDX the participant is required to fill a form where any extra response action run is reported and associated with a functional description. Second, all the participants are asked to provide an estimate of the cost value, for all the collected extra actions, according to their descriptions.

Finally, after collecting all the response plans, the actual CRC is calculated as described in Equation (1). Since providing an estimate of the cost value of an action requires significant modeling experience, we use normalized cost values to mitigate the effects of the participant's experience on the CRC measure. All the anticipated and executed actions are classified according to the provided functional description and, for each class, the average value of the cost function is taken and used for all the actions in that class.

B. Data Analysis Plan

With 10 participants per group and 10 observations (attacks) our sample size is 100, which is sufficient to assume a normal sampling distribution. Under this assumption, we test the two directional research hypotheses stated in Section III with a one-tailed Student's t-test. As customary in empirical research [22], a significance level $\alpha = 0.05$ can be assumed.

As described in Section IV, for each group we take repeated measures of TTR and CRC under 10 different attack situations. The sequence of attacks each participant faces while running the CDX is randomized to distribute over the entire CDX session any potential emotionally discouraging (encouraging) effect the participant may experience countering an attack perceived as particularly tough (mild).

Nonetheless, the specific attack under which we measure TTR and CRC must be considered as another independent variable in our experiment, which may have a non-negligible effect on TTR, CRC or both. Thus, we conduct a two-way ANOVA analysis to study how the *Attack* and *Group* factors, as well as their interaction $Group \times Attack$, affects TTR and CRC.

VI. Threats to Validity

Although the experimental design described in Section IV allows for the collection of samples large enough to obtain significant results from statistical analysis, a major threat to the external validity of the proposed methodology is that it considers a single exemplar of a managed system. While we expect the effectiveness of a self-protecting system to be positively correlated to the complexity of the managed artifact, we cannot generalize the validity of the methodology until it is applied to systems of varying complexity. Moreover, a modern information system changes over time, thus it rarely satisfies the hypothesis of stationarity this study implies. In this respect, the experiment design could be revised to take non-stationarity into account, for instance, by introducing the alteration of the managed system as a new independent variable.

Another potential threat to validity stems from the CRC measure we adopted to characterize the effectiveness of a self-protecting system. First, the CRC is dependent on the cost function and, ultimately, on the impact each response action has on the business of the organization running the managed system; as a result, its significance cannot be readily generalized across different organizations. However, it is still relevant when used to compare different security solutions within a specific organization. Second, CRC is an indirect measure that estimates the response cost for a given plan with a linear model, therefore, it might not be representative of the real cost sustained to carry out a real incident response. In this regard, risk analysis techniques and business data from previous security incidents could be used to adjust the cost model to better fit a particular organization or scenario.

VII. Related Works

In their survey on reaction frameworks [4], Nespoli et al. consider the response time as an important factor for determining the effectiveness of a response system. Another criterion the authors followed to evaluate the surveyed frameworks concerned the quality of the produced response. However, in this regard, the authors also note that while established metrics and scoring systems exist for security vulnerabilities, there is a lack of commonly used measurement systems for quantitative assessment of countermeasures. Our work tries to bridge this gap choosing well-known security metrics, providing their operationalization in the context of self-protecting systems, and designing a methodology for performance evaluation.

Lee et al. in [11] introduce a cost-sensitive model for IDSs and IRSs based on risk-assessment concepts. The authors develop a cost model to drive intrusion and response decision-making activities and identify the cost factors to be considered for the development and evaluation of these systems, i.e., damage cost, response cost, and operational cost. The same topic has also been addressed by Shameli-Sendi et al. in [24], where the authors classify three types of cost model: the *static model* associates a constant cost with a response action; the *statically evaluated model* defines a cost function to evaluate the effects of the execution of the response action; the *dynamically evaluated model* also takes into account the system state. The framework introduced in this paper is agnostic of how the cost is attributed to a specific response action, as long as it has no dependency on the system state.

In their empirical study [3], Harrison and White utilize data gathered from the 2009 National Collegiate Cyber Defense Competition to validate the effectiveness of common security measures found in literature, by analyzing the correlation of the team score to the security measures

applied and the order in which they have been applied. Besides providing a validation model for many security practices whose effectiveness was mainly based on common sense, the authors remark the importance of a prompt application of security measures, as well as the need for tools aiding system administrators to achieve this goal. Similarly, we propose an empirical study as an evaluation methodology. However, in our work this methodology is applied to evaluate how a self-protecting system performs when compared to an incident response carried out manually.

Finally, a radically different approach to evaluate security automation tools is taken by Sommestad et al. in [25] for their Cyber Security Modeling Language (CySeMoL) vulnerability assessment expert system. Cyber security experts have been engaged in a Turing test aimed to verify the hypothesis the proposed tool could perform as an expert in devising an attack tree for a given target system. Similarly to our approach, the authors apply an empirical method to evaluate the performance of a tool for security automation. Their experiment was designed to evaluate a static reaction framework and the collected data consisted in the answers to a questionnaire administered to two experts who were in charge of evaluating the solutions produced by both other experts and the tool. In contrast, our experimental design suits the evaluation of a dynamic reaction framework and empirical data is collected during the experiment run-time via objective measurements.

VIII. Conclusions and Future Work

In the quest for tools promoting effective automation of security incident response workflows, this study provides an empirical methodology to evaluate the effectiveness of self-protecting systems, a set of measures and their operationalization for this context, and an experimental design. Nonetheless, further research is needed for defining a standardized managed system and cyber-range architecture in order to run actual experiments and collect data to validate our approach.

Moreover, as elicited in Section VI, additional efforts are also needed to generalize the methodology to information systems of arbitrary complexity.

References

[1] M. Papadaki, S. Furnell, L. BL, and R. PL, "Enhancing response in intrusion detection systems," *Journal of Information Warfare*, vol. 2, pp. 90–102, 01 2002.

[2] A. Guarino, "Autonomous intelligent agents in cyber offence," in *2013 5th International Conference on Cyber Conflict (CYCON 2013)*, June 2013, pp. 1–12.

[3] K. Harrison and G. White, "An empirical study on the effectiveness of common security measures," in *2010 43rd Hawaii International Conference on System Sciences*, Jan 2010, pp. 1–7.

[4] P. Nespoli, D. Papamartzivanos, F. G. Mármol, and G. Kambourakis, "Optimal countermeasures selection against cyber attacks: A comprehensive survey on reaction frameworks," *IEEE Communications Surveys Tutorials*, vol. 20, no. 2, pp. 1361–1396, Secondquarter 2018.

[5] P. Cichonski, T. Millar, T. Grance, and K. Scarfone, "Computer security incident handling guide," *NIST Special Publication*, vol. 800, no. 61, pp. 1–147, 2012.

[6] M. Albanese, H. Cam, and S. Jajodia, "Automated cyber situation awareness tools and models for improving analyst performance," in *Cybersecurity systems for human cognition augmentation*. Springer, 2014, pp. 47–60.

[7] "An architectural blueprint for autonomic computing," IBM, Tech. Rep., Jun. 2005.

[8] F. Cohen, "Simulating cyber attacks, defences, and consequences," *Computers & Security*, vol. 18, no. 6, pp. 479 – 518, 1999.

[9] R. Ortalo , Y. Deswarte , and M. Kaaniche , "Experimenting with quantitative evaluation tools for monitoring operational security," *IEEE Transactions on Software Engineering*, vol. 25, no. 5, pp. 633–650, Sep. 1999.

[10] H. Holm , M. Ekstedt , and D. Andersson , "Empirical analysis of system-level vulnerability metrics through actual attacks," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 6, pp. 825–837, Nov 2012.

[11] W. Lee, W. Fan, M. Miller, S. J. Stolfo, and E. Zadok, "Toward cost-sensitive modeling for intrusion detection and response," *J. Comput. Secur.*, vol. 10, no. 1-2, pp. 5–22, Jul. 2002.

[12] A. Fielder, E. Panaousis, P. Malacaria, C. Hankin, and F. Smeraldi, "Decision support approaches for cyber security investment," *Decis. Support Syst.*, vol. 86, no. C, pp. 13–23, Jun. 2016.

[13] K. A. Stouffer, V. Y. Pillitteri, S. Lightman, M. Abrams, and A. Hahn, "Guide to industrial control systems (ics) security," *NIST Special Publication*, vol. 800, no. 82 Rev. 2, 2015.

[14] S. Iannucci and S. Abdelwahed, "Model-based response planning strategies for autonomic intrusion protection," *ACM Trans. Auton. Adapt. Syst.*, vol. 13, no. 1, pp. 4:1–4:23, Apr. 2018.

[15] Q. Chen, J. Lambright, and S. Abdelwahed, "Towards autonomic security management of healthcare information systems," in *2016 IEEE First International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*, 2016, pp. 113–118.

[16] C. Mu and Y. Li, "An intrusion response decision-making model based on hierarchical task network planning," *Expert Systems with Applications*, vol. 37, no. 3, pp. 2465 – 2472, 2010.

[17] N. Stakhanova, S. Basu, and J. Wong, "A cost-sensitive model for preemptive intrusion response systems," in *21st International Conference on Advanced Information Networking and Applications (AINA '07)*, 2007, pp. 428–435.

[18] P. Lalanda, J. A. McCann, and A. Diaconescu, *Autonomic computing: principles, design and implementation*. Springer Science & Business Media, 2013.

[19] K. Geers, "Live fire exercise: Preparing for cyber war," *Journal of Homeland Security and Emergency Management*, vol. 7, 01 2010.

[20] "Cyber defence exercise locked shields 2013. after action report," The NATO Cooperative Cyber Defence Centre of Excellence, Tech. Rep., 2013.

[21] H. Kim, L. Gurrin, Z. Ademi, and D. Liew, "Overview of methods for comparing the efficacies of drugs in the absence of head-to-head clinical trial data," *British journal of clinical pharmacology*, vol. 77, no. 1, pp. 116–121, jan 2014.

[22] P. C. Cozby and S. C. Bates, *Methods in Behavioral Research*. New York: McGraw-Hill, 2017.

[23] J. Mcclain, A. Silva, G. E. Aviña, and C. Forsythe, "SANDIA REPORT Measuring Human Performance within Computer Security Incident Response Teams," Tech. Rep., 2015.

[24] A. Shameli-Sendi, N. Ezzati-Jivan, M. Jabbarifar, and M. Dagenais, "Intrusion response systems: survey and taxonomy," *Int. J. Comput. Sci. Netw. Secur*, vol. 12, no. 1, pp. 1–14, 2012.

[25] T. Sommestad, M. Ekstedt, and H. Holm, "The cyber security modeling language: A tool for assessing the vulnerability of enterprise system architectures," *IEEE Systems Journal*, vol. 7, no. 3, pp. 363–373, Sep. 2013.