Enhancing Response Selection in Impact Estimation Approaches

Gabriel Klein[1], Andres Ojamaa[2], Pavel Grigorenko[2],
Marko Jahnke[1], Enn Tyugu[3]

[1] Fraunhofer Institute for Communication, Information
Processing and Ergonomics FKIE
Neuenahrer Str. 20, 53343 Wachtberg, Germany

[2] Institute of Cybernetics at Tallinn Technical University
Akadeemia tee 21, 12618 Tallinn, Estonia

[3] Cooperative Cyber Defence Centre of Excellence
Filtri tee 12, 10132 Tallinn, Estonia

Abstract: The number of attacks against computer systems is steadily increasing. Network administration personnel often have a wide variety of response measures against these attacks at their disposal. In previous work, a methodology was introduced for efficiently assessing the effects of countermeasures on network resources before their actual application and thus determining the most appropriate response. Building on this, we now propose a method of dynamically weighting the metrics used to evaluate the different responses. Instead of a fixed linear combination of metrics we introduce Pareto optimal combinations of the individual metrics and the combined cost measure. This allows a more flexible way of emphasizing the importance of individual metrics in different situations. The methodology was prototypically implemented in CoCoViLa, a powerful simulation engine for visually specified optimization problems.
Keywords: Denial-of-service attacks, automated response, response evaluation, response metrics, Pareto optimality

1. Introduction

Along with the rising number of computer systems connected to the Internet which are infected with malware, the danger of large-scale denial-of-service attacks occurring also increases. To maximize the speed and reliability of response measures against such attacks, it is desirable to select and apply response measures automatically. In GrADAR (Graph-based Automated Denial-of-Service Attack Response), the selection of responses is made according to an estimation of the measures' impact on the protected system. Here, the impact is estimated according to different criteria, so-called metrics. Currently, the overall cost of a response

measure is defined as a linear combination of the different metric values in which each metric has a different weighting reflecting its relative importance. A higher flexibility can be attained by performing a Pareto optimization of the individual metric values as well as their linear combination. With this, multiple objectives regarding the metrics can be achieved; for example, a response measure with a maximal value for a certain metric can be chosen which also has a high overall rating.

The rest of this paper is structured as follows: Section 2 introduces related work in which Pareto optimality is used for multi-objective optimization. Thereafter, Section 3 gives a brief introduction to GrADAR. This is followed by a description of how Pareto optimization can be used to select response measures more flexibly and with regard to multiple objectives (Section 4). Section 5 gives details on our implementation in CoCoViLa, a visual simulation system. This is followed by Section 6 which presents first results of our work. Section 7 then summarizes our work and provides an outlook on further activities.

2. Related Work

In [1], Horn et al. introduce a Pareto criterion into the selection operator of a genetic algorithm to enable multi-objective optimization. As opposed to a scalar fitness function where the solution can be very sensitive to parameter changes, this allows a more robust selection of non-dominated solutions.

Douligeris [2] studies Pareto optimality in a telecommunications context. Here, flow control is managed by solving a problem with the two objectives maximal throughput and minimal delay. Pareto optimal solutions are then compared according to fairness.

From a network security point of view, the following contributions are interesting. Gu et al. [3] propose an intrusion detection system in which two different feature extraction approaches are used to construct event classifiers. The combination of the advantages of both systems into a single objective would require advance knowledge. Therefore, a multi-objective optimization is performed which yields Pareto optimal solutions.

In [4] and [5], Ojamaa et al. describe a graded security expert system which enables choosing security measures for information assurance. In the security model, the combination of the two objectives low cost and high confidence is achieved by an optimization technique based on dynamic programming. The user is presented with a Pareto optimality trade-off curve permitting the choice of the most appropriate security measures.

3. Graph-based Automated Denial-of-Service Attack Response

Graph-based Automated Denial-of-Service Attack Response (GrADAR, [6], [7]) is a framework for assessing the effect of response measures against denial-of-service

attacks on the availability of network services. This section describes the GrADAR model and introduces the required terminology.

In GrADAR, the so-called dependency graph $\hat{G} = (\hat{V}, \hat{E})$ models the ideal state of a network. Its vertices $\hat{V}$ correspond to the network resources and the edges $\hat{E}$ signify availability dependency relationships between the resources. Vertices $r_i \in \hat{V}$ are labeled with a dependency function D_{r_i} according to which, a resource's availability can be estimated based on the availability of antecedent resources. Additionally, the edges $e_{i,j} \in \hat{E}$ are labeled with a dependency weighting function $w_{i,j} : [0,1] \to [0,1]$ which signifies the degree to which resource r_i is dependent on resource r_j $(r_i \triangleright r_j)$.

A second graph $G = (V, E)$, the so-called accessibility graph, reflects the actual current state of the network. Its vertices $r_i \in V$ correspond to those in the dependency graph but are labeled with an availability value $A(r_i) \in [0,1]$. The set of edges E is a subset of $\hat{E}$ $(E \subseteq \hat{E})$ and an edge $e_{i,j} \in E$ reflects the ability of resource r_i to access another resource r_j.

Using information in both these graph structures now allows the estimation of availability values of resources for which availability is not directly observable. For a resource r_i, $r_i \triangleright r_j$ and $r_i \triangleright r_k$, the availability of r_i can be predicted using the following formula:

$$A(r_i) = D_{r_i}(w_{i,j}((A(r_j), w_{i,k}A(r_k)))$$

Figure 1 shows an example of both a dependency and an accessibility graph. Here, a user D (also modeled as a network resource) us dependent on the availability of a local operating system and a running HTTP service to perform some task, e. g. browsing a Web shop. The HTTP service itself is again dependent on the availability of the IP stack, in turn dependent on the operating system. The accessibility graph on the right shows a reduced availability of the IP resource, possibly due to an overloaded link to the nearest router (not displayed in the graph). Because of the availability dependency relationships between the resources, this results in a reduced availability of the user node, manifested, for example, by a reduction in speed of the user's browsing experience.

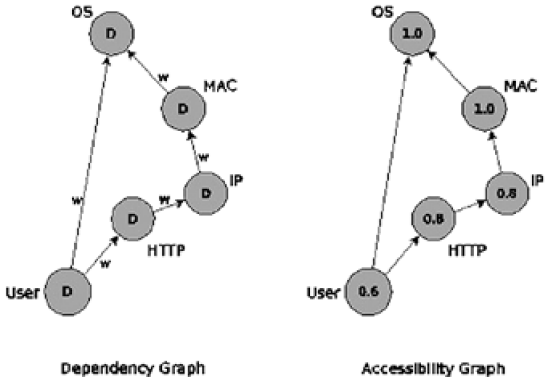

Figure 1. Example of a dependency graph and a corresponding accessibility graph

To estimate the effects of response measures on the availability of network services, each response is virtually applied to the model. The effects are then quantified and the most appropriate response is applied to the real-world network.

Virtual application of response measures is performed by modifying the dependency and accessibility graph in one or more of the following ways:

1. change availability values of nodes (G only),
2. adding/removing vertices ($\hat{G}$ and G), or
3. adding/removing edges ($\hat{G}$ and G).

After these changes have been made, the availability dependency relationships in the dependency graph need to be used by an update algorithm (e. g. a recursive depth-first search) to estimate the effect these changes have on the availability of other resources. For a more detailed description of availability propagation, please refer to [8].

Two possible response measures for the scenario depicted in Figure 1 could be the following:

1. Dynamic reallocation of the available bandwidth on the IP link. This could result in an increased availability of the local IP stack (corresponding to item 1 above).
2. Utilization of a second IP link to perform a form of load balancing. This would introduce a second IP node and a second MAC node into the graph, along with the corresponding availability dependency relationships (corresponding to items 2 and 3 above).

To select the most appropriate of the available response measures, they are evaluated with respect to multiple criteria, or metrics. There are currently four different metrics: success (δ_S), durability (δ_D), application costs (δ_C) and error-proneness (δ_E). These can be given individual weights by factors $w_S, w_D, w_C, w_E \in \mathbb{R}^+$.

Let $\Theta = \{\theta_1, \ldots, \theta_n\}$ be the set of all possible response measures. The best response measure θ_{best} is then determined by a suitability function which minimizes the costs and the error-proneness and maximizes the success and durability, i. e.

$$\theta_{best} = \arg\min_{\theta \in \Theta} \left(S(\theta) \right),$$

where $S(\theta) = w_C \cdot \delta_C(\theta) + w_E \cdot \delta_E(\theta) - w_S \cdot \delta_S(\theta) - w_D \cdot \delta_D(\theta)$ is the linear combination of metric values and $\delta_C, \delta_E, \delta_S, \delta_D$ are functions $\delta_i : \Theta \to \mathbb{R}$ which represent the metrics.

Figure 2 shows an overview of the GrADAR approach. The ideal state of the network captured in the dependency graph is augmented with availability information for some resources provided by an implemented intrusion detection or network management system. An update algorithm utilizing the availability dependency relationships between resources is used to estimate availability values for resources for which no values were provided by the IDS/NMS.

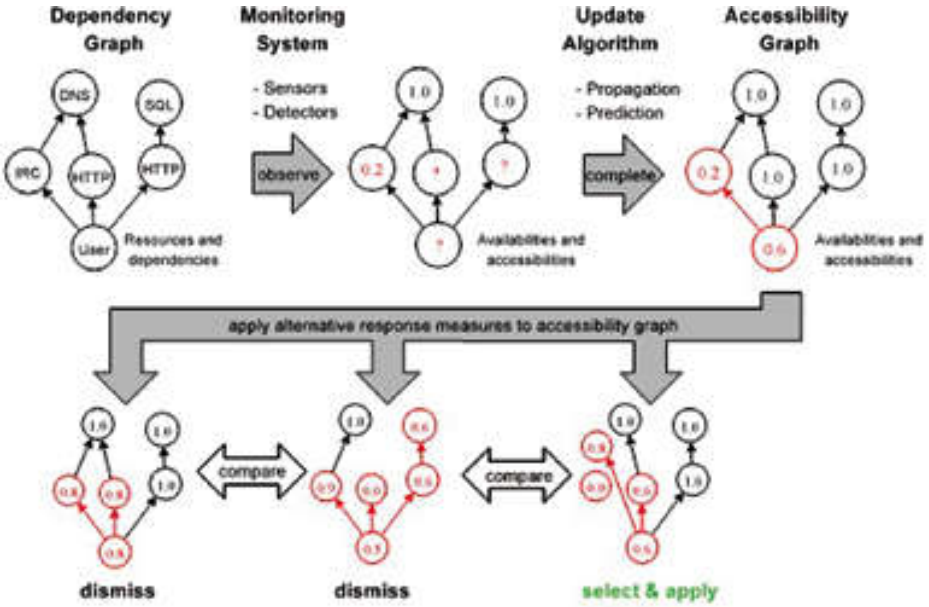

Figure 2. Overview of the GrADAR approach

Real-world response measures correspond to transformations of the two graphs. For each possible response measure, the graphs are individually modified. The resulting response graphs provide a measure for the resulting availability after the application of the corresponding countermeasure. However, this is only one of the possible metrics according to which countermeasures can be evaluated (see above). After an appropriate response measure has been chosen and applied, the corresponding response graph serves as dependency graph for the next iteration of GrADAR.

4. Pareto Optimal Response Selection

Pareto optimality or efficiency is a concept originating in economics. Broadly speaking, a Pareto optimum of a group of individuals is a state in which any change to the benefit of an individual would at the same time be to the detriment of another individual. More formally, an n-dimensional tuple $x_1,\ldots,x_n \in A$ is a Pareto optimum of the set A if there is no tuple $y_1,\ldots,y_n \in A$ with $y_i \geq x_i$ for $i = 1,\ldots,n$ where "≥" is a superiority relation. The set of all Pareto optimal outcomes is called a Pareto set.

To extend the rather static assessment of response measures through the weighted linear combination of individual metrics, we propose to also analyse responses by presenting the results in the form of Pareto sets. This is theoretically possible for all available metrics, i. e. for all response measures θ_i, the tuple $\left(\delta_C(\theta_i),\delta_E(\theta_i),\delta_S(\theta_i),\delta_D(\theta_i),S(\theta_i)\right)$ can be represented. Note that subsequently, the weighted combination of metrics $S(\theta)$ will be treated as a metric as well, signifying the overall "cost" of the response measure. However, to retain overall manageability, analysis should be restricted to two or three metrics. We can, for example, plot the best possible values of $S(\theta)$ for certain response vectors against the values

of selected metrics, e. g. application costs, for these responses. A reasonable choice would be to compose the metrics reflecting gains, $S^+(\theta)=\delta_S(\theta)+\delta_D(\theta)$, and those reflecting the losses, $S^-(\theta)=\delta_C(\theta)+\delta_E(\theta)$, and to plot the curve relating $S^+(\theta)$ and $S^-(\theta)$. The final response choice will then explicitly take into account both the overall quality and the costs.

It is important to have a convenient way of selecting different Pareto variables and plotting different Pareto curves. The next section gives a brief overview of a tool developed for this purpose.

5. Model-based Implementation

The aim of the present approach is to develop an automatic response selection method by experimenting with different ways of the response selection. To facilitate the experiments, we have developed a visual model-based software tool for representing accessibility graphs and problems on these graphs. This is a GrADAR software package developed for the CoCoViLa platform [9]. The package provides assets for specifying response selection problems and for high-level control of computations on the graph. It contains components for resources, optimization methods and for visualization of results. CoCoViLa supports problem solving on higher-order constraint networks [10] that can be easily used for propagating availability values on the accessibility graphs.

The first application of the software was to analyze the effect of response measures by automatically propagating workload values (red arrows) and availability values (green arrows) of resources. This is shown in Figure 3. The problem was visually specified as a scheme that was a union of dependency and accessibility graphs extended with an analysis component (Propagator). Nodes representing resources had a number of parameters that were observed and adjusted in a property window of a node.

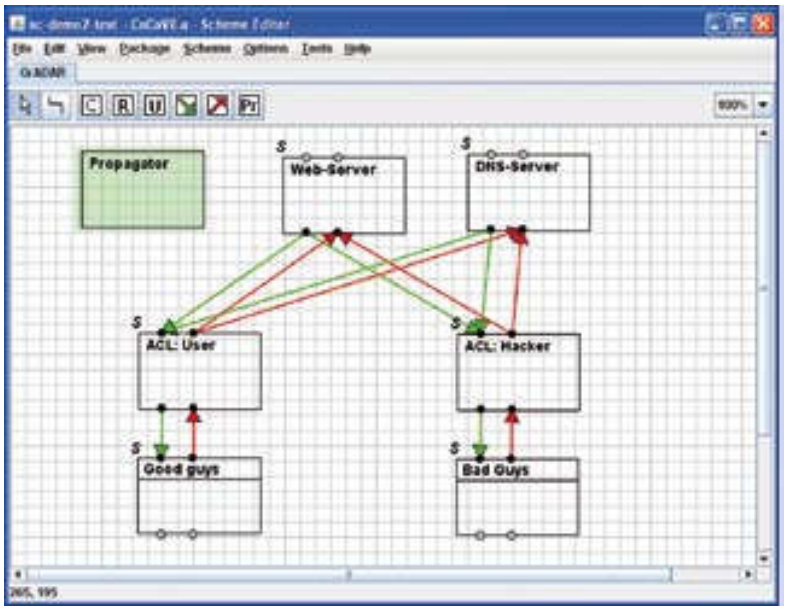

Figure 3. Visual specification of a response analysis problem

There are two possibilities for specifying resource availability values. On the one hand, values can be manually entered in the resource properties window. This supports offline simulations of the effects of countermeasures on static resource scenarios and can be useful for trial-and-error determination of novel countermeasures. On the other hand, there are also interfaces to arbitrary back-end management systems from which live values can be obtained. Thus, the simulation engine has a dual use as a monitoring system operating on real-life values. This can aid in real-time countermeasure evaluation for network administration personnel.

Figure 3 shows a visual specification of the response analysis problem. Its menu bar contains buttons for all types of components and connectors. A specification was built by using buttons of the menu bar, and by introducing parameter values of components through their pop-up property windows. A properties window for a resource node (Web-Server) is shown on the right side in Figure 4.

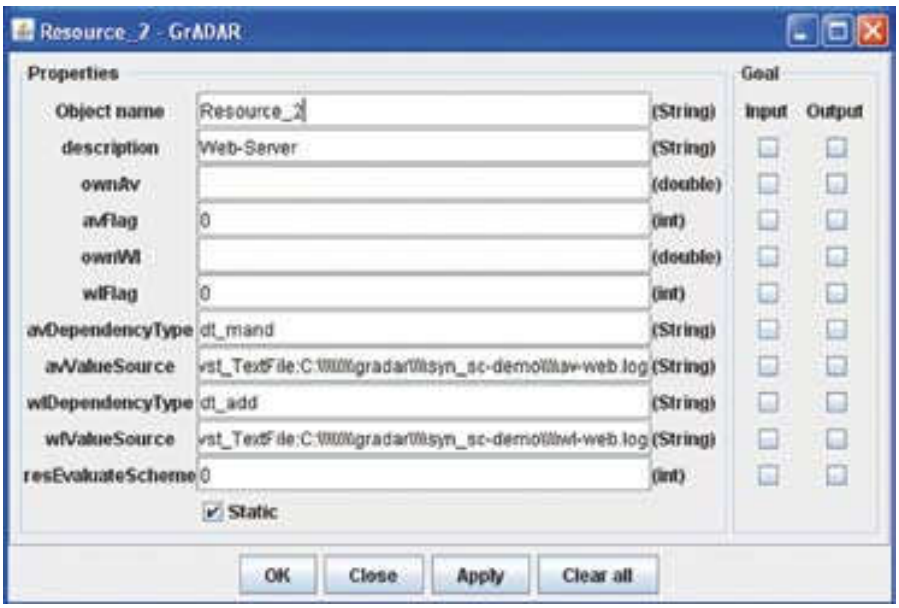

Figure 4. Properties window of the Web server resource

The problem can be specified in a textual language as well. In fact, a textual specification is always automatically generated from a visual specification, if the latter has been given. The ability to generate the scheme in textual format is especially useful because scheme generation can be automated and processed offline.

6. First Results

The main application of the developed software is calculation of various Pareto sets, using an optimization component for selecting the best solution from a given set of possible response measures. This problem is described by accessibility graph, extended with visualization, optimization and response measures nodes; see Figure 5 for a simplified example.

The developed GrADAR package enables decision-making (choosing a response) by first determining a set of admissible responses and thereafter either finding the best response or plotting a Pareto curve to help with the choice. The computations are performed as follows. The set of possible responses Θ is a set of tuples constructed from possible response measures for the resource nodes of the accessibility graph. A tuple of response measures constitute the description of a response θ, we call it a response vector. The optimizing component is able to produce all required values of the response vector and distribute its components to the resource nodes for calculations. The results of calculations are collected from the resource nodes and passed back to the optimizing component. This collection and distribution of responses is described in the optimizing component simply by the following CoCoViLa statement:

```
alias responseVector = (*.response);
```

where `response` must be the name of a response action in each resource node.

At the present stage we use a brute-force search for determining the best response for given arguments, generating all possible values of response θ. A Pareto set of pairs (x, y) is constructed as follows. Values x and y of the Pareto coordinates are calculated for all possible values of θ. The response θ' with the best value of y is selected for each given value of x. A Pareto set (the set of selected points (x, y)) is plotted. Also, a table can be constructed with rows representing a response θ' for each point (x, y). As we have noted in Section 3, different metrics can be used as the variables x and y. Figure 5 shows a visual specification of a problem and a Pareto set for x representing normalized gains S^+ and $y = 1 - S^-$, where S^- represents normalized costs. Another potentially interesting Pareto set is for x representing δ_S (success of the countermeasure) and y representing S, the overall cost measure for the countermeasure. This would enable an administrator to choose a response measure which maximizes the resulting network availability while minimizing the overall cost.

We would like to emphasize that the user can quickly analyze multiple trade-off situations by connecting various ports of the optimizer component outputting different metric values to the ports of the graph component. New, arbitrarily complex metrics can be defined using equations and existing Java methods in the specification window.

In order to be able to analyze responses that introduce new elements into the graph, we use a supergraph of the accessibility graph that includes all possible extensions, e. g. the servers that can be added as responses. When the availability of all these resources is zero, we get the initial accessibility graph that can be extended. Actually, the node Backup Webserver in Figure 5 is just a node with zero initial availability, i. e. it can be added to the network as a response.

The graph window in Figure 5 shows the points of the calculated Pareto set as red rectangles (only the points with the highest y value for each x value are visible). Each point represents the estimated outcome of applying a response. The graph

displays a tool tip on each rectangle which contains the index and the exact x and y values of the corresponding response (the tool tip also shows the values for points with the same x value not in the Pareto set). The index can be used for looking up the response steps leading to this outcome.

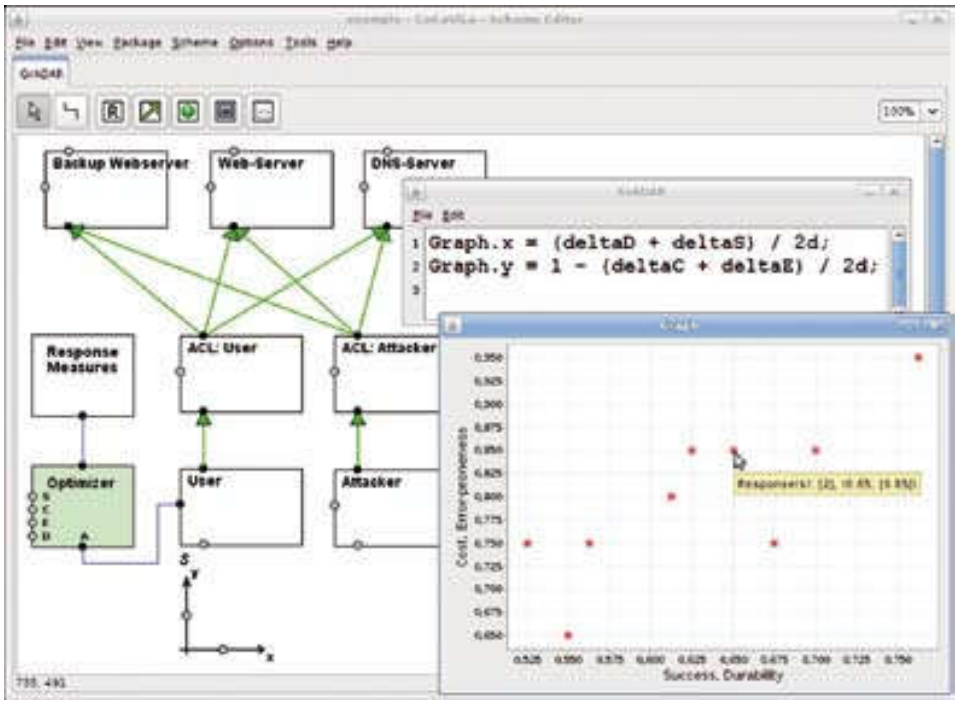

Figure 5. Plot of the Pareto set (gains, costs)

7. Conclusion and Further Work

We have presented a methodology and prototypical implementation for dynamically modifying the weighting of metrics for evaluating the effects of pre-defined countermeasures against computer network attacks. Instead of a fixed combination of metric values, Pareto sets now allow more flexible and more differentiated determination of the most appropriate reaction to a detected attack.

Currently, only the evaluation of pre-defined countermeasures is supported. However, it is possible that through recombination of elementary response steps, new, more sophisticated responses can be generated. This includes the definition of a permissible order in which response steps can be concatenated and the elimination of infeasible or erroneous results. This requires further research into the necessary changes to the model.

Acknowledgements

The authors would like to thank the Federal Office for Information Management and Information Technology of the German Armed Forces, the Cooperative Cyber Defence Centre of Excellence, the Estonian Defence Forces Training and

Development Centre of Communication and Information Systems, and the Estonian Ministry of Defence (grant No. 372/0807) for the support of this work. The second author would like to thank the Estonian Information Technology Foundation and the Tiger University programme for partial support of this work.

REFERENCES

[1] Horn J., Nafpliotis N., and Goldberg D., A niched pareto genetic algorithm for multiobjective optimization. In: Proceedings of the 1st IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence, pages 82-87, 1994.

[2] Douligeris C., Multiobjective flow control in telecommunication networks. In: INFOCOM '92. Eleventh Annual Joint Conference of the IEEE Computer and Communications Societies, IEEE, volume 1, pages 303-312, May 1992.

[3] Gu Y., Zhou B., and Zhao J., PCA-ICA ensembled intrusion detection system by pareto-optimal optimization. Inform. Technol. J., 7:510-515, 2008.

[4] Ojamaa A., Tyugu E., and Kivimaa J., Pareto-optimal situation analysis for selection of security measures. In: Proceedings of the 4th IEEE Workshop on Situation Management SIMA 2008, San Diego, CA, USA, November 2008.

[5] Kivimaa J., Ojamaa A., and Tyugu E., Graded security expert system. In: R. Setola and S. Geretshuber, editors, Critical Information Infrastructures Security, volume 5508 of LNCS, pages 279-286. Springer-Verlag, 2009.

[6] Jahnke M., Tölle J., Thul C., and Martini P., Validating GrADAR – An Approach for Graph-based Automated DoS Attack Response. In: Proceedings of the 34th IEEE Conference on Local Computer Networks (LCN2009), Zurich, Switzerland, 2009.

[7] Jahnke M., Klein G., Tölle J., and Martini P., Protecting Military Networks with GrADAR – Graph-based Automated DoS Attack Response. In: Proceedings of the Military Communication Conference 2009, Prague, Czech Republic, 2009.

[8] Jahnke M., Graph-based Automated Denial-of-Service Attack Response. PhD thesis, University of Bonn, 2009.

[9] Grigorenko P., Saabas A., and Tyugu E., Visual tool for generative programming. In: Proceedings of the ACM SIGSOFT Symposium on the Foundations of Software Engineering. ACM Press, 2005.

[10] Tyugu E. and Uustalu T., Higher-order functional constraint networks. In: Constraint Programming, volume 131 of NATO ASI Series F: Computer and System Sciences, pages 116-139. Springer-Verlag, 1994.