Stateful RORI-based countermeasure selection using hypergraphs

Gustavo Gonzalez-Granadillo[a,*], Elena Doynikova[c], Joaquin Garcia-Alfaro[b], Igor Kotenko[c], Andrey Fedorchenko[d]

[a] *Innovation, Cybersecurity Laboratory, Atos Research, Barcelona, Pere IV, 291-307, 08020, Spain*
[b] *Télécom SudParis, Paris-Saclay University, CNRS SAMOVAR UMR 5157, 9 rue Charles Fourier, 91011 EVRY, France*
[c] *Laboratory of Computer Security Problems, St. Petersburg Institute for Informatics and Automation (SPIIRAS), St. Petersburg 14, Liniya, 39, Russia*
[d] *ITMO University, 49 Kronverksky Pr., St. Petersburg 197101 Russia*

A R T I C L E   I N F O

Keywords:
Stateful RORI
Cost-sensitive metrics
Hypergraph models
Countermeasure selection
Network security

A B S T R A C T

Cost-sensitive metrics have been widely used during the past years as financial metrics that quantify the monetary costs and benefits of security investments, assess risks, and select countermeasures accordingly. However, due to the complexity of current attacks, and the level of dynamicity required in the estimation of the parameters composing the metrics, the use of a novel approach that considers restrictions, interdependency, as well as, the previous and current state at which the system is exposed to, is required. We propose in this article a Stateful Return on Response Investment (denoted by StRORI) that uses hypergraphs to model actions that have been previously deployed (e.g., at state ST_0) while the current state of the system (e.g., ST_1) is under analysis. As a result, StRORI is a dynamic tool that considers the changes of the system in terms of number of active devices, previously deployed countermeasures, the cost of adding a new countermeasure or suppressing a previously deployed one, and the effectiveness of a group of security measures due to the implementation of a given action. A case study is presented about the integration of the StRORI index with hypergraph models to assess countermeasures against cyber attacks.

1. Introduction

Information security models have been proposed as an approach to evaluate investments and returns (commonly referred to as the amount of losses that are avoided due to a security investment; losses that were expected to occur had these investments not been applied) [1]. Cost sensitive metrics i.e. Return On Investment (ROI) [2,3], Return On Security Investment (ROSI) [4–6], Return On Response Investment (RORI) [1,7] have been proposed for the quantitative evaluation of investments in the ICT domain. Most of the current cost-sensitive metrics use a great level of subjectivity or rely on expert knowledge while estimating each parameter composing the equation. Thus, results are generally approximations of a real environment that use best guesses rather than quantitative models in their estimations.

The RORI index had been previously proposed as a financial metric to evaluate multiple countermeasures and to select the best alternative against complex attack scenarios. Yet, this metric does not take into account the countermeasure history i.e., no information about the actions that have been previously deployed (state ST_0) while the current state of the system is under analysis (ST_1).As a result, it is not possible to compute the impact on the cost and effectiveness that the application or suppression of a countermeasure causes to the rest of actions already deployed.

In addition, attack scenarios are frequently represented with attack graphs. These latter have been widely proposed to analyze the path(s) taken by attackers during the exploitation of a vulnerability, and to evaluate all possible security measures to implement for stopping/mitigating the threat [8]. Attack and defense graphs have been utilized to model the behavior of attacks in a network infrastructure and the corresponding counter actions [9,10]. However, traditional attack graphs for the large information systems incorporate a lot of nodes and edges. Besides they should be constantly modified due to the changes in the information system that leads to the constant RORI recalculations and extension of historic RORI indexes list. Therefore they are complex for visual perception and require high time costs for processing [58,59]. Hypergraphs [11–13] are seen as a viable and prominent solution to compute exhaustive lists of attack scenarios, and to select the most effective countermeasures based on cost-sensitive metrics.

In this paper, we propose to integrate a Stateful Return On Response Investment (StRORI) index (as previously defined in [14]),

* Corresponding author.
E-mail addresses: gustavo.gonzalez@atos.net (G. Gonzalez-Granadillo), doynikova@comsec.spb.ru (E. Doynikova), joaquin.garcia_alfaro@telecom-sudparis.eu (J. Garcia-Alfaro), ivkote@comsec.spb.ru (I. Kotenko), fedorchenko@comsec.spb.ru (A. Fedorchenko).

with a Hypergraph model aiming at evaluating countermeasures based on financial and threat impact assessment functions. The ultimate goal of the StRORI metric is the ranking of countermeasures and the selection of the optimal candidate(s) to reduce (or eliminate) the impact of the detected attacks. StRORI considers all previously deployed security measures as well as the economical impact of implementing a new countermeasure or deactivating a deployed one.

The overall contributions on this paper are summarized as follows: (i) a quantitative model that evaluates security actions against complex attack scenarios at each state of the system; (ii) a process that dynamically evaluates the StRORI index on multiple candidates while considering restrictions, and inter-dependency among them; (iii) a methodology to compute the parameters composing the StRORI index; (iv) the integration of the StRORI index with a hypergraph model; and (v) the implementation of the model over a real attack scenario.

Paper Organization: Section 2 compares related work with our approach. Section 3 introduces and defines the stateful return on response investment metric. Section 4 introduces the hypergraph model. Section 5 presents the preventive approach used to select optimal countermeasures. Section 6 discusses about the reactive approach used for countermeasure selection. Section 7 details the integration between the StRORI Index and the Hypergraph approach. Section 8 describes the implementation of the model and provides a use case as an illustration of the hybrid approach. The paper concludes in Section 9.

2. Related work

The evaluation and selection of security measures as a way to react to cyber attacks is an open research area. Approaches like the one proposed by Kheir et al., [1] consider Service Dependency Frameworks (SDF) during the process of selecting the reaction mechanisms responsible of applying changes to the system's configuration. The main drawback of this approach is the inability to evaluate the monetary impact of selected security measures over its dependent services.

Samarji et al. [15], propose an approach that uses Situation Calculus to automatically generate attack graphs that leverage response systems means to estimate the global risk inferred by simultaneous ongoing attacks, and to reason about appropriate responses. Authors, however, do not estimate the risk of simultaneous attacks on the network service, which is crucial for response systems to react intelligently against the most dangerous and complex attacks.

Lippmann et al. [16] and Poolsappasit [17] use in their approaches attack graph models for the implementation of preventive and reactive security measures on the exploitation of the system's vulnerabilities. The preventive approach supports most attacks that can be discovered through network vulnerability scanners, but do not easily support attacks with multiple prerequisites. The reactive approach enables a system administrator to quantify the chances of network compromise at various levels, but it requires improvements in terms of scalability and efficiency.

Martinelli and Santini [18] suggest the use of Argumentation to provide automated support for security decisions. The manipulation of this reasoning process comes with a cost in terms of the chosen metrics. The main limitation of this type of approach is being able to determine the cost of manipulating a final decision by acting on the decision process itself.

Motzek et al. [19] have proposed an approach for selecting adequate response plans as a reaction to threats opposed on a company based on a multi-dimensional impact assessments. The approach considers a response financial and operational impact assessments. However, the proposed solution have the inherent limitations associated to the RORI index e.g., accuracy issues, inability to consider indirect increase of financial costs, inability to evaluate potential decrease of financial impact, no consideration of semantic implications of individual countermeasures.

More recently, Yaqoob et al. [20] propose a ROSI framework by calculating the impact of an attack on the whole business while considering its effect upon all critical assets. Authors use Bayesian theorems to determine likelihood of cyber-attacks and overcome uncertainty to a reasonable extent. The approach, however, has not yet been used in real time organizational environments, and the methodology has not yet been automated.

Furthermore, Soikkeli et al., [21] propose a framework for automated countermeasure selection based on cost impact analysis of the organisation's service loss and costs over a period of time. Authors consider attacker and defender actions, aiming for a cost-effective approach to maintaining service functionality. Although demonstrated and tested in simulated environments, the approach does not adopt a flexible dependency model and does not select a combination of countermeasures and recovery actions in the same decision.

With regard to the aforementioned limitations, the approach presented in this paper estimates the risk of simultaneous attacks against the system, and computes the cost of the final decisions by acting on the decision process itself, as well as, evaluates the impact of combined responses over dependent services. It builds over a hypergraph formalism complemented with a cost-sensitive metric used as an automated response selection mechanism that anticipates forecasted steps of an attacker aiming at disrupting the security of a given system.

3. Stateful return on investment metric

3.1. RORI Approach

The RORI model has been defined by Kheir et al. [1], as an index aiming at providing a common reference to compare different response candidates and to choose the optimal response.

More recently, Gonzalez-Granadillo et al. [7,22] propose an improvement of the RORI index that evaluates parameters such as the annual loss expectancy (ALE) that results from an intrusion or attack, the risk mitigation level (RM), the annual response cost (ARC), and the annual infrastructure value (AIV), as depicted in Eq. (1):

$$RORI = \frac{(ALE \times RM) - ARC}{ARC + AIV} \times 100. \tag{1}$$

The calculation of the parameters presented in Eq. (1) follows the approaches proposed by Kosutic, in [23], Locher, in [24] and Lockstep Consulting, in [25]. More information about the computation of each parameter composing the RORI index can be found in [7].

As a result, by using RORI, it is possible to select optimal countermeasure(s) against pre-defined attack scenarios. The metric provides a response relative to the size of the infrastructure by using the AIV parameter, this latter is correlated with the ALE of the system, which allows to compare the RORI result of different systems regardless of their size. The introduction of the AIV parameter handles the case of selecting no countermeasure, and allows to differentiate mutually exclusive countermeasures from partially and/or totally restrictive ones.

3.2. Stateful RORI (StRORI)

The Stateful Return On Response Investment Metric (StRORI) has been proposed as an extension of the RORI index [7,22]. The proposed StRORI considers dynamicity in the monitoring system (e.g., changes in the network due to the detection of new threats or

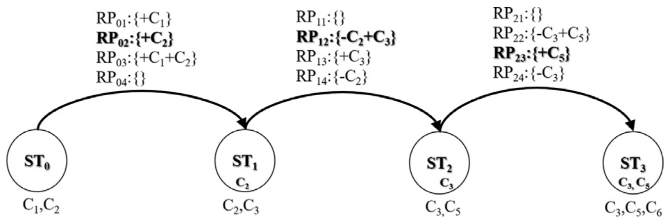


Fig. 1. Transition Process in the Dynamic RORI Evaluation.

the implementation of new security measures). The system is assumed to work under a given state (e.g., ST_0, ST_1, ..., ST_{n-1}, ST_n), for each of them the system captures the information related to each node composing the network as well as the pre-defined security configuration. Each evaluation run represents a unique snapshot of the system (state) thus, it is possible to compare the security parameters of the systems for multiple states.

Fig. 1 illustrates a case with three transitions (i.e., from ST_0 to ST_1, from ST_1 to ST_2, and from ST_2 to ST_3, and several countermeasures (i.e., C_1, C_2, C_3, C_5, C_6) to be evaluated during the different states of the system. Starting from ST_0, we assume that no security measure has been previously enforced in the system. At this state we perform the RORI evaluation (with C_1 and C_2), and we obtain four possible response plans (RP): (i) add C_1 (i.e., $RP_{01} = \{+C_1\}$); (ii) add C_2 (i.e., $RP_{02} = \{+C_2\}$; (iii) add C_1 and C_2 (i.e., $RP_{03} = \{+C_1, +C_2\}$); and (iv) No action, meaning that no mitigation action must be implemented (i.e., $RP_{04} = \{\}$). For this example, the RORI index indicates that the optimal countermeasure is RP_{02}, we therefore implement C_2 and the state changes to ST_1.

The following state of the system (ST_1) considers the previously implemented countermeasures and updates the system's configuration accordingly. All authorized mitigation actions (even those already implemented in the system) are evaluated by the StRORI in order to find the optimal response plan. Assuming that we have two candidates: C_2 and C_3 (for this state C_1 is not considered as an appropriate candidate), we will have four possible response plans: (i) add C_2, in this case, no action is performed since C_2 has been implemented during the previous state of the system (i.e., $RP_{11} = \{\}$); (ii) add C_3, in this case C_2 must be uninstalled in order to install C_3 (i.e., $RP_{12} = \{-C_2 + C_3\}$); (iii) add both C_2 and C_3, in this case, only C_3 is added since C_2 is already enforced (i.e., $RP_{13} = \{+C_3\}$); and (iv) no countermeasure should be enforced, meaning that C_2 must be uninstalled (i.e., $RP_{14} = \{-C_2\}$). The resulting StRORI at ST_1 indicates the optimal action is to enforce RP_{12}, we must therefore uninstall C_2 and install C_3, the state changes to ST_2.

The process is repeated until reaching the final state (in this example ST_2) in which two countermeasures are evaluated (C_3, C_5, all other countermeasures are not considered in this state as candidates for evaluation). Four possible response plans are analyzed: (i) add C_3, in this case no action is performed, since C_3 is already installed (i.e., $RP_{21} = \{\}$); (ii) add C_5, in this case C_3 must be uninstalled in order to install C_5 (i.e., $RP_{22} = \{-C_3 + C_5\}$); (iii) add both C_3 and C_5, in this case, only C_5 is added since C_3 is already enforced (i.e., $RP_{23} = \{+C_5\}$); and (iv) no countermeasure should be enforced, meaning that C_3 must be uninstalled (i.e., $RP_{24} = \{-C_3\}$). The resulting StRORI at ST_2 indicates the optimal action is to enforce RP_{23}, we must therefore install C_5, the state changes to ST_3. Please note that for simplicity, we have used two countermeasures in each state, but the model allows n number of candidates to be evaluated in each state of the system.

The remainder of this section details the StRORI metric parameters and the methodology to compute them.

3.3. Computation of the StRORI parameters

This section details each of the parameters composing the StRORI index and provides a methodology to help in their computation.

3.3.1. Annual loss expectancy (ALE)

In the absence of security measures, *ALE* expresses the impact cost perceived by an organization as a consequence of an attack. *ALE* considers multiple losses: loss of assets (La), Loss of data (Ld), Loss of reputation (Lr), Legal procedures (Lp), Loss of revenues from existing clients or customers (Lrec), loss of revenue from potential clients (Lrpc), Other losses (Ol), as well as the Contracted insurance (Ci), and the annual rate of occurrence (ARO), as shown in Eq. (2).

$$ALE = (La + Ld + Lr + Lp + Lerc + Lrpc + Ol - Ci) \times ARO \tag{2}$$

ALE depends directly on the severity and likelihood of the threat and it is independent on the mitigation actions and the policy enforcement points.

3.3.2. Annual infrastructure value (AIV)

The *AIV* is computed as the sum of the Annual Equipment Cost (*AEC*) of all policy enforcement points (PEPs) that appears in the system's snapshot (Eq. (3)):

$$AIV = \sum_{i=0}^{n} AEC_i. \tag{3}$$

The *AEC* corresponds to all the costs associated to the use of the equipment regardless of the implemented countermeasure; e.g., Equipment usage (Ec), Personnel costs (Pc), Service costs (Sc), Other costs (Oc), and Resell Value (Rv), as shown in Eq. (4):

$$AIV = Ec + Pc + Sc + Oc - Rv. \tag{4}$$

Each PEP has an associated *AEC* that is estimated based on historical information and expert knowledge using Eq. (4). Contrary to the *ALE*, the value of the *AIV* changes at each snapshot of the system.

3.3.3. Risk mitigation (RM)

RM refers to the risk mitigation associated to a given countermeasure. The computation of this parameter depends on two factors (i) the value of the countermeasure effectiveness - EF (as presented in Section 3.3.3.1), and the value of the countermeasure coverage - COV (as presented in Section 3.3.3.2); and (ii) the performed action, i.e., considering if a new security action needs to be added (e.g., patch a vulnerability), or if the action must be uninstalled/deleted (e.g., unblock users, IPs, ports that were previously blocked), or if no action is required (keep unchanged).

The Risk Mitigation for individual countermeasures is calculated using Eq. (5):

$$if \begin{cases} CM(add) \rightarrow & RM = COV \cdot EF \\ CM(delete) \rightarrow & RM = 0 \\ CM(keep) \rightarrow & RM = Unchanged \end{cases} . \tag{5}$$

For a given state of the system, RM considers if the evaluated countermeasure is new (add a CM), for which the RM is computed as the product of the countermeasure coverage (COV) and its effectiveness (EF). However, if the response plan suggests the deletion of a previously implemented countermeasure (delete CM), then RM will be equal to zero, since the candidate will be disabled. Finally, if the response plan suggests the implementation of an already deployed countermeasure (keep CM), RM will remain the same as the one used in the previous state of the system for this countermeasure.

Table 1
Default effectiveness values associated to mitigation action types.

Mitigation Action Type	Protection	EF
Reboot	Very Low	1.00%
Shutdown	Low	10.00%
Backup	Medium	50.00%
Change Configuration	High	80.00%
Patching	Very High	100.00%

For multiple disjoint countermeasures, RM is computed as the sum of the effectiveness of each countermeasure times their corresponding coverage as depicted in Eq. (6):

$$RM(CM_1 \cup \ldots \cup CM_n) = \sum_{i=1}^{n} COV(CM_i) \cdot EF(CM_i). \tag{6}$$

However, for joint countermeasures, RM is computed as the sum of their individual coverage times their corresponding effectiveness, minus the intersection coverage times the minimum effectiveness value as shown in Eq. (7):

$$RM(CM_1 \cup \ldots \cup CM_n) = \sum_{i=1}^{n} COV(CM_i) \cdot EF(CM_i) \\ - [CM_i \cap \ldots \cap CM_n] \cdot EF_{min}(CM_i \ldots CM_n). \tag{7}$$

More details and examples on the computation of these Equations can be found in [7].

In addition, the RM computation must consider the previously deployed countermeasure(s) so that impact of adding, deleting or keeping a countermeasure is reflected in the effectiveness of the group of evaluated countermeasures. The following rules are considered while evaluating the risk mitigation of multiple countermeasures:

- If the proposed group of countermeasures are not deployed at state ST_0, the RM at state ST_1 is computed using Eq. (7). For instance, having two new and partially joint countermeasures to evaluate (i.e., CM_1, CM_2), the RM of their union is computed as $RM(CM_1 \cup CM_2)= CM_1.EF_1 + CM_2.EF_2 - [(CM_1 \cap CM_2) .EF_{\min}(CM_1, CM_2)]$;
- If one or more of the proposed countermeasures are already deployed at state ST_0, the RM evaluation at state ST_1 must subtract the RM value associated to each previously deployed countermeasure. For instance, having two partially joint countermeasures to evaluate at state ST_1 (i.e., CM_1, CM_2), and considering that CM_1 has been deployed at state ST_0, the RM of their union is computed as $RM(CM_1 \cup CM_2)= RM(CM_2)$;
- If all countermeasures to be evaluated at state ST_1 are already deployed in the system (at state ST_0), RM of the group of countermeasures remains the same as the one computed in the previous state;
- If one or more of the evaluated countermeasures are proposed to be deleted, RM for such countermeasure(s) is equal to zero, since this candidate is not considered in the RORI evaluation.

Effectiveness (EF) The effectiveness (EF) of a countermeasure represents the level at which a given action reduces the risk and/or consequences of an attack on the system. EF is intrinsic to the mitigation action type regardless of the threat it mitigates. For instance, a reboot action by itself provides a very low mitigation of a given threat, whereas a patching action provides a very high protection against it. Table 1 summarizes default values associated to mitigation action types. Each value has been assigned based on statistical data and expert knowledge [7].

It is important to note that (i) if the evaluated countermeasure was not previously deployed in the system, the effectiveness value shown in Table 1 is added in the RM computation; (ii) if the evaluated countermeasure is already deployed in the system, and the action to be taken proposes its deletion, the effectiveness value is subtracted in the RM computation; and (iii) if the proposed countermeasure is already deployed in the system, and the action to be taken proposes its implementation, we discard the candidate in the RM calculation.

Coverage (COV) The coverage (COV) of a given countermeasure represents the number of nodes to which a mitigation action is being executed over the total number of vulnerable nodes, as shown in Eq. (8):

$$COV = \frac{Q_i \cdot WF_i}{\sum_{j=0}^{n} QT_j \cdot WF_j}. \tag{8}$$

Where Q_i is the number of nodes from a PEP_type affected by a countermeasure; QT_j is the total number of active nodes in the system; WF_i is the weighting factor associated to the affected PEP_type; and WF_j is the weighting factor associated to each node type. This latter indicates the level of priority or criticality inherent to the type of PEP in the execution of a mission. For instance, personal computers (e.g., PC) are assigned a WF=1, Web servers (e.g., WEBSCADA) are assigned a WF=3, and Remote Terminal Units (i.e., RTU) are assigned a WF=5.

Note that if the evaluated countermeasure is not previously deployed in the system, the coverage value is computed as in Eq. (8). In addition, if the countermeasure is already deployed in the system, and the action to be taken proposes its deletion, the coverage value is subtracted in the RM computation. Further, if the proposed countermeasure is already deployed in the system, and the action to be taken proposes its implementation, we discard the candidate in the RM calculation.

3.3.4. Annual response cost (ARC)

ARC refers to the Annual Response Cost associated to a particular countermeasure. For individual evaluations, ARC includes Direct costs (e.g., Cost of implementation (Ci), Cost of maintenance (Cm), Cost of deletion (Cd)), Other direct costs (Odc); and Indirect costs (Ic), and the calculation considers the fact that a countermeasure is added, deleted or kept, as shown in Eq. (9):

$$if \begin{cases} CM(add) \rightarrow & ARC = Ci + Cm + Odc + Ic \\ CM(delete) \rightarrow & ARC = Cd \\ CM(keep) \rightarrow & ARC = Cm \end{cases}. \tag{9}$$

Similar to the computation of RM, the ARC considers if the evaluated countermeasure is new (add a CM), for which the ARC is computed as the sum of all direct and indirect costs associated to the countermeasure. If the countermeasure needs to be uninstalled (delete CM), then ARC will be equal to the cost of deletion (Cd). If the suggested countermeasure is already deployed in the system (keep CM), ARC will be equal to the cost of maintenance although no action is required.

For multiple joint and disjoint countermeasures, the annual response cost is equal to the sum of all individual countermeasure's cost as shown in Eq. (10):

$$ARC(CM_1 \cup \ldots \cup CM_n) = \sum_{i=1}^{n} ARC(CM_i). \tag{10}$$

In addition, the calculation of the ARC parameter must consider the following conditions:

- If the proposed group of countermeasures are not deployed at state ST_0, the ARC at state ST_1 is computed using Eq. (10). For instance, having two new and partially joint countermeasures to evaluate (i.e., CM_1, CM_2), the ARC of their union is computed as $ARC(CM_1 \cup CM_2)= ARC(CM_1) + ARC(CM_2)$;

- If one or more of the proposed countermeasures are already deployed at state ST_0, the ARC computation must subtract the ARC value associated to each previously deployed countermeasure. For instance, having two partially joint countermeasures to evaluate at state ST_1 (i.e., CM_1, CM_2), and considering that CM_1 has been deployed at state ST_0, the ARC of their union is computed as $ARC(CM_1 \cup CM_2)= ARC(CM_2)$;
- If all countermeasures at state ST_1 are already deployed in the system (at state ST_0), the ARC of the group of countermeasures only includes the cost of maintenance;
- If one or more countermeasures is proposed to be deleted from the system, the ARC computation only includes the cost of the deployed action.

4. Hypergraph model

We introduced our approach briefly in [26]. In this paper we describe it in details, we provide description of integration of the hypergraph approach and StRORI metric and demonstrate joint application of the hypergraph model and StRORI metric on the case study.

Attack graphs are used to represent all possible steps performed by the attacker in the system. This section proposes an attack graph approach used for the countermeasure selection in the static and dynamic system operation modes, as suggested by us in [9,28,29] and by Nespoli et al. in [27]. The approach evolves the attack model by using a Bayesian approach [17] specified in Definitions 1 and 2. The goal is to represent, anticipate and handle attack actions performed by an attacker targeting a given system.

Definition 1 (Attack Graph)**.** A graph $G = (S, L, \tau, P_c)$ where S contains the nodes of the graph (i.e., the set of attack actions), L represents the set of links between actions (s.t. $L \subseteq S \times S$), τ the logical relation between attack actions (in our implementation relation AND is specified using sequential attack actions, while OR relation is specified using actions with the same parent), and P_c the discrete local conditional probability distributions (in our implementation it is the conditional probability matrix for the whole graph).

Definition 2 (Attack Action)**.** A tuple $S = (H, V, Sc, St, Pr)$, where H identifies the host to which the attack action is applicable, V represents the vulnerability used in the attack action, Sc the process implemented by the attacker that do not use vulnerabilities, e.g. to get information about the host (in our implementation we use CAPEC attack patterns for this goal [30]), St the state of attack action (successfully implemented or not), and Pr the probability that the attack action is in St ($Pr \in [0, 1]$).

Thus, the attack graph incorporates all known attack sequences in the computer networks, where each sequence consists of attack actions (it can be vulnerability exploitation or attack step that does not use vulnerabilities, for example, footprinting) and links between them. Link specifies transition from one attack action to another depending on the post-conditions of the parent action (obtained privileges) and preconditions of the child action (required privileges to implement attack action). Simplified example is given in Fig. 2: the left part – fragment of the computer network, the right part – fragment of attack graph for it. In Fig. 2 each graph node is vulnerability exploitation attack action (vulnerabilities are represented with their CVE ids), that incorporates vulnerabilities with the same pre and post conditions. Each graph node is specified as follows: *H_NAME* : *AccessVector_Auth_GainedPrivileges_AccessComplex* (*H_NAME* – host name; *AccessVector*, *Auth*, and *AccessComplex* – CVSS access vector, CVSS authentication requirements, and CVSS access complexity for the vulnerability, accordingly; and *GainedPrivileges* – the privileges on a host after vulnerability exploitation).

Attack graph extension by assigning Bayes probabilities Pr to the graph nodes considering available subjective data named Bayesian attack graphs are usable formalism to forecast attack development and to trace attack sources. We calculate prior unconditional probabilities (Pr_* in Fig. 2) for the graph nodes using CVSS values [13]. St allows considering security incidents in the reactive mode of attacks prevention (by changing the state of attack action represented with attack graph node to the ‘successfully implemented’). For example, a security incident occurred in node “DS1” its state is changed to “successfully implemented” (St=1). This results in changing of attack probability for this node (posterior probability $Pr_{p(DS1)}$) and, consequently, for other connected nodes (ancestors and descendants). Changing of probabilities for the ancestor nodes in its turn results in changing of St for some of these nodes to the successfully implemented (if there are several paths to the node with incident then for the path with maximum probability is St changes to 1). Thus the dynamics is introduced into the proposed model.

Application of Bayesian attack graphs is limited by the possibility of the attack sequences with cycles in the computer network [31,32]. The second challenge of the modern computer networks is their huge size that hampers their effective automatic and manual processing and visual perception.

We suggest using hypergraphs to overcome aforementioned challenges [26]. Hypergraphs are widely used in research on the information security for the networks modeling [33–35], for anonymous communications [36,37], for the alert correlation [38], for modeling of the security dependencies [39], for describing security properties [40]. In [41] an algorithm for the attack graph generation based on the hypergraph partitioning is provided. In [42] and [43] hypergraphs are used with the similar to our goal for the generalization of the graph model. But in [42] authors use it to specify logical statement for the security violation on the basis of logs. In [43] authors abstract the attack scenarios using a hierarchy of activity types for the intrusion goals. While our proposal is based on the detailed attack scenarios that are automatically constructed considering known vulnerabilities of the analyzed system and interconnections between them (see Definitions 1 and 2) and it allows quantification of the attack actions for the revelation of the most dangerous scenarios and further selection of countermeasures. We use hypergraph definition provided in [44].

Definition 3 (Hypergraph)**.** The hypergraph H is a graph $H = (X, U; R)$, where $x \in X = x_i / i \in I$- vertexes, $u \in U = u_j / j \in J$- edges, R- predicate that specifies if x and u are incident in H. In the hypergraph an edge can incorporate arbitrary number of vertexes [44]. Hypergraph edges are pre-processed to get acyclic graph (to use Bayesian approach). In Fig. 3(b) three subgraphs are consequentially linked in one hypergraph for the demonstration purposes, whereas in Fig.3(c) the final acyclic graph is provided.

Besides, the size of the graph in Fig.3(c) is reduced compared to the initial attack graph in Fig. 3(b). It allows us to overcome the second aforementioned challenge combining multiple links within one node. More demonstrative example of an attack graph that has few subgraphs that are connected via single node is represented in Fig. 3(a). In this case we represent these subgraphs as nodes of hypergraph (Fig. 3(c)). It allows us to preprocess these subgraphs (nodes) separately and then process final hypergraph in case of preventive processing. From the another hand, in case of reactive processing, hypergraph is used to localize compromised subgraph and then only this subgraph is used for further calculations to reduce processing time. Thus hypergraphs simplifies dynamical analysis of the large-scale complex attack graphs.

While transition from an attack graph to hypergraph seems to be a complication of the model, we consider it reasonable for a number of advantages. First of all, unlike a graph, a hypergraph

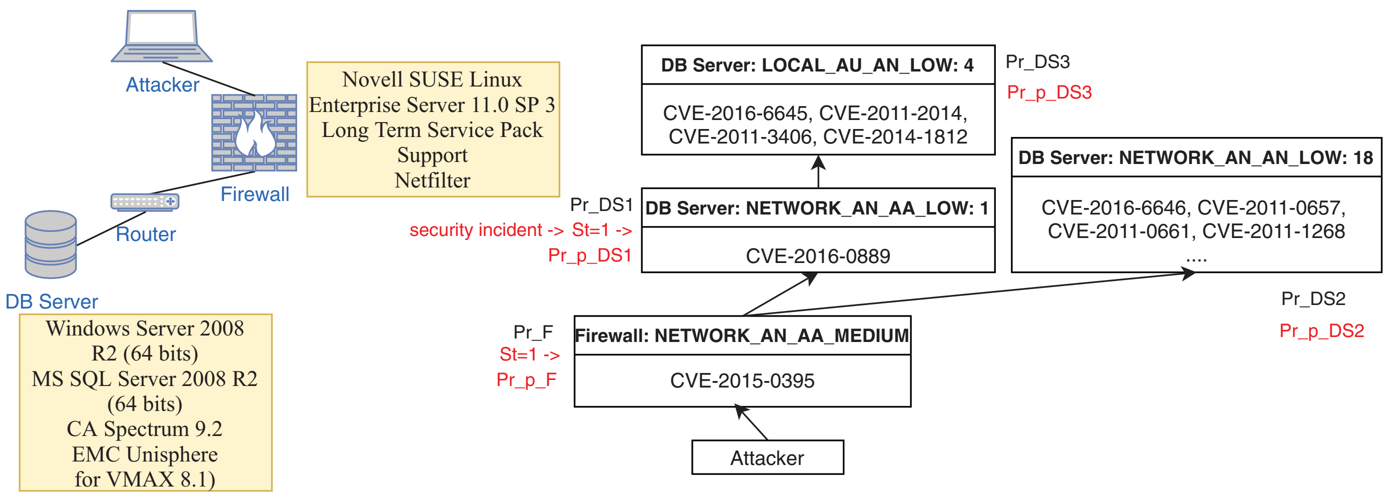

Attacker
Firewall
Router
DB Server
Novell SUSE Linux Enterprise Server 11.0 SP 3 Long Term Service Pack Support Netfilter
Windows Server 2008 R2 (64 bits) MS SQL Server 2008 R2 (64 bits) CA Spectrum 9.2 EMC Unisphere for VMAX 8.1)
DB Server: LOCAL_AU_AN_LOW: 4
CVE-2016-6645, CVE-2011-2014, CVE-2011-3406, CVE-2014-1812
Pr_DS3
Pr_p_DS3
DB Server: NETWORK_AN_AN_LOW: 18
CVE-2016-6646, CVE-2011-0657, CVE-2011-0661, CVE-2011-1268 ....
Pr_DS1
security incident -> St=1 -> Pr_p_DS1
DB Server: NETWORK_AN_AA_LOW: 1
CVE-2016-0889
Pr_DS2
Pr_p_DS2
Pr_F
St=1 -> Pr_p_F
Firewall: NETWORK_AN_AA_MEDIUM
CVE-2015-0395
Attacker

Fig. 2. Fragment of the computer network (left) and appropriate attack graph (right).

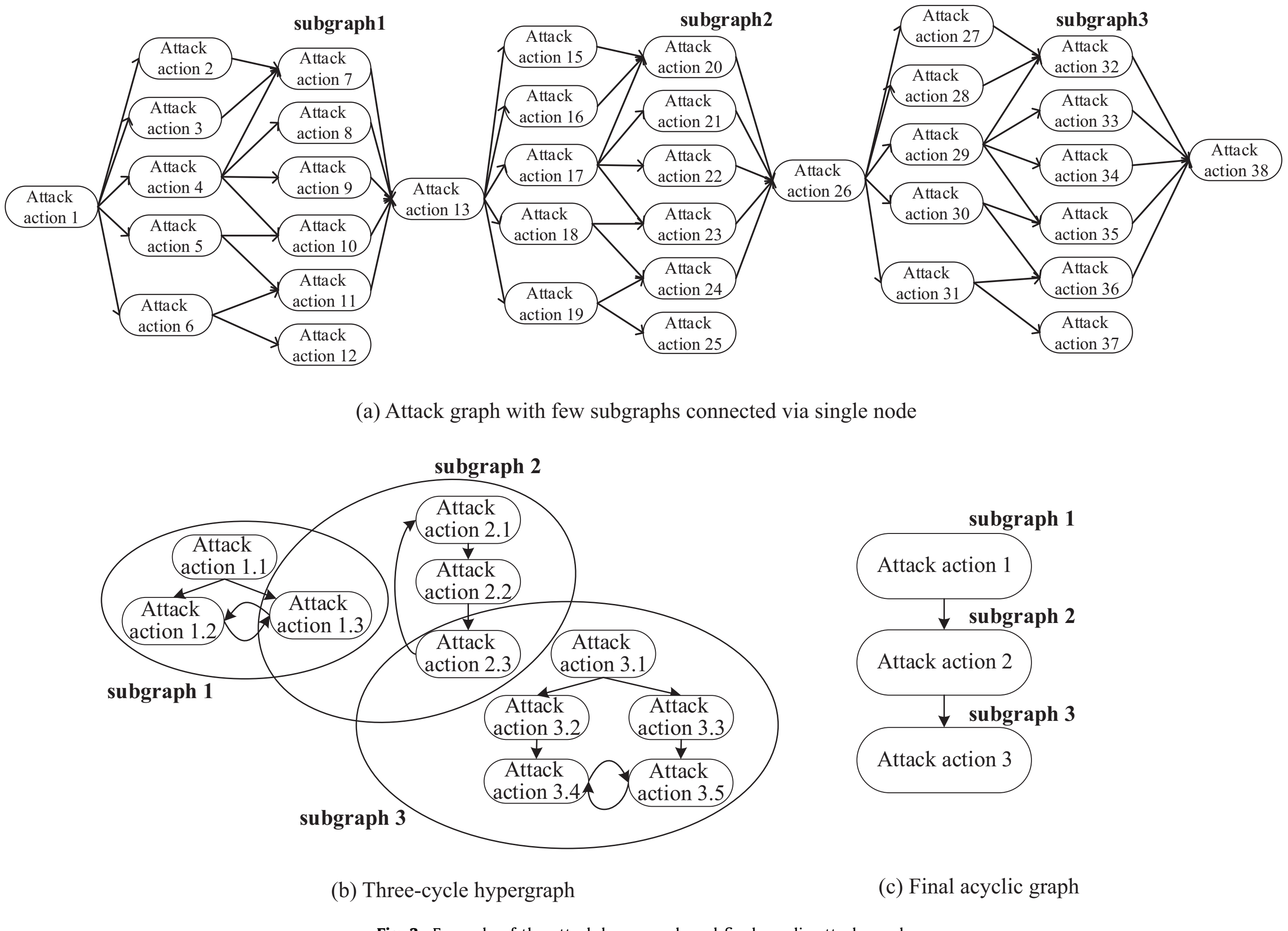


Fig. 3. Example of the attack hypergraph and final acyclic attack graph.

allows grouping related objects (vertexes) and displaying *n*-ary relations by its nature, besides, it allows their nesting [45,46].

The limitations that made us searching for new methods of attack modeling despite the fact that attack graphs shown them well, are described in the beginning of this section. In particular, this is a problem with cycles and the challenge of processing and visual representation of huge graphs. The advantage of using hypergraphs considering the first challenge (cycles processing) consists in the fact that each hypergraph's edge can be processed as separate graph. It allows processing of the separate subgraphs, in particular, handling loops.

The advantage of using hypergraphs considering the second challenge (analysis of huge graphs) consists in the fact that each hypergraph's edge can incorporate a number of vertexes and edges, that allows decreasing graph size on the upper level (combining multiple vertexes within a single edge) and processing this decreased model first. Then we can focus on the detected problem (e.g. compromised subgraph). Besides, such a hypergraph will be easier to read when visualizing.

Another significant advantage of hypergraphs is the ability to include additional semantic information inside the hyperedge, i.e. to group related objects of different types within one edge. In particular, within a single edge, it is possible to combine both vulnerabilities, weaknesses, exploits, attacks, events, countermeasures and the relationships between them, without changing the general idea of the hypergraph to display the attack path, but providing it with additional information. Now we use it to consider attack action and countermeasure within single hyperedge, it allows calculating *StRORI* index and implementing countermeasures for different network segments (combiming segment attack subgraph within one hyperedge) instead of implementing them for specific vulnerabilities.

Thus, we outline three types of hyperedges: (i) Hyperedges that incorporate vertexes and edges forming cycles, (ii) Hyperedges that incorporate vertexes and edges of single network segment, and (iii) Hyperedges that incorporate objects of different types, namely, attack actions and countermeasures.

Technically, hypergraph models are probed to be more efficient than classical graph models for modeling and computing relational data[47]. In this sense, hypergraphs provide significant improvements in runtime (up to 30 times faster than graphs models). In addition, hypergrahs require fewer Locally Optimal Block Preconditioned Conjugate Gradient (LOBPCG) iterations than graph models, and thus converge up to 6 times faster than graphs. Furthermore, operators (e.g., Laplacian) apply up to 17 times faster for hypergraphs models than graph models[47]. The main limitation of hypergraphs is that they are more difficult to visualize and draw on papers than its alternative graphical models[48]. To overcome this problem, several research studies have introduced a variety of methods for their visualization[49–52].

Finally, from our point of view, the hypergraph will allow us to join within a single model all models of interaction in scope of security analysis. It simplifies common model (that incorporates all objects, namely, vulnerabilities, weaknesses, exploits, attacks etc.) instead of its complication.

5. Static mode of calculations

This section provides specification of parameters used in the risk calculation and countermeasure selection for the static mode of calculations. In the static mode of calculations we do not consider dynamic information, i.e. information on the detected attack instances.

5.1. Prior risk calculation

We introduce static mode of calculations to determine the common level of risk in the analyzed system and to select the preventive countermeasures that will allow reducing the risk if necessary. We calculate the common level of risk on the basis of local risk levels that, in their turn, we calculate using attack graph. The attack graph represents all possible multi-step attack scenarios in the system (Definition 1). Each attack scenario is a chain of sequential actions connected on the basis of pre and post conditions w.r.t. vulnerability exploitations and Bayesian probabilities (Definitions 1 and 2) [53,54].

Definition 4 (Local risk level)**.** The local risk levels indicate whether countermeasures should be implemented in the analysed system to prevent attack scenarios. The local risk levels are calculated for each node of the attack graph using the standard risk Eq. (11):

$$\text{Risk} = \text{AI} \times \text{Pr}, \tag{11}$$

where AI is represented in the form of a linear combination of damages for the asset confidentiality I_c, integrity I_i and availability I_a in case of successful implementation of attack action (i.e. vulnerability exploitation) corresponding to the graph node; and the criticality of confidentiality C_c, integrity C_i and availability C_a of these assets (Eq. (12)):

$$\text{AI} = (\text{C\_c} \times \text{I\_c}) + (\text{C\_i} \times \text{I\_i}) + (\text{C\_a} \times \text{I\_a}). \tag{12}$$

I_c, I_i and I_a are calculated considering CVSS impact indexes for the appropriate vulnerability [55]. The Pr parameter represents probability of successful implementation of attack action (i.e. vulnerability exploitation) corresponding to the graph node (corresponds to the Pr parameter from Definition 2). We use a total probability formula to calculate the Pr on the basis of a local vulnerability probability p and a conditional probability Pc. The Pc parameter allows considering all the possible states St of the graph node ancestors Pa while calculating Pr (corresponds to the Pc parameter from Definition 1). Pc value depends on the logical relation between attack action parents Pa. If parent nodes are connected with AND relation, Pc is set to zero if there is at least one action in Pa whose exploitation state St_i is *False*; otherwise, Pc equals p. If parent nodes are connected with OR relation, Pc is set to zero if for all actions in Pa the exploitation state St_i is *False*; otherwise, Pc equals p. The p parameter is calculated using Eq. (13) for the root nodes of the attack graph, and it is calculated using Eq. (14) for other nodes:

$$p = 2 \times \text{AV} \times \text{AC} \times \text{Au}, \tag{13}$$

$$p = 2 \times \text{AC} \times \text{Au}. \tag{14}$$

Eqs. (13) and 14 are adopted from the CVSS equation for the *Exploitability* index [55] normalized between 0 and 1 using the 2 factor. AV represents the access required to exploit a vulnerability, AC represents the complexity of exploitation of the appropriate vulnerability, and Au represents if the additional authentication methods required to exploit a vulnerability [55].

The risk level for attack sequences is calculated as the combination of the minimum probability of the attack nodes and the maximum impact. The common risk level for the analyzed system is calculated as the maximum risk level of the attack sequences.

The described countermeasure selection formalism allows selecting preventive countermeasures considering the most vulnerable nodes of graph (i.e. nodes with a risk level that exceeds a predefined threshold) and introduced countermeasure selection index.

5.2. Optimal countermeasure selection in the preventive mode

The countermeasure selection is implemented for the graph nodes with unacceptable risk level (risk level that exceeds the predefined threshold). To select the optimal countermeasures set we bypass attack graph considering countermeasure impact area (subgraph, graph node, vulnerability) and countermeasure impacted properties (confidentiality, integrity, availability, or their combinations).

Definition 5 (Preventive Countermeasure selection)**.** The countermeasure selection for each object is performed using a countermeasure selection index *csi*, considering countermeasure efficiency for the risk level mitigation (i.e., *Efficiency*), countermeasure cost (i.e., *Cost*) and countermeasure collateral damage (i.e., *CD*), as depicted in Eq. (15):

$$csi = Efficiency - Cost - CD. \tag{15}$$

We assume that maximization of the *csi* on the impact subarea leads to its maximization on the impact area as a whole.

The algorithm of the countermeasure selection is recursive in terms of the impact area: the same actions are repeated for the countermeasures that impact subgraph (several graph nodes), graph nodes and separate vulnerabilities. The algorithm pseudo code is provided in Algorithm 1.

Algorithm 1 Pseudo code of the countermeasure selection algorithm.

```
1: For all nodes risk_nodes with risk > threshold
2:     implement zero-cost patches
3:   redefine risk_nodes
4: Sort countermeasures and generate list cms1
5: While risk_nodes is not empty and end of cms1 is not reached
6:     Get cm1 from cms1
7:     For all cm from cm1
8:         calculate csi_sg1
9:     Select cm with max csi_sg1, sum csi_sg and csi_sg1
10:    Add cm to cm_sg
11:    remove covered nodes from risk_nodes
12: Sort countermeasures and generate list cms2
13: While risk_nodes is not empty and end of cms2 is not reached
14:     Get cm2 from cms2
15:     For all cm from cm2
16:         calculate csi_sg2
17:     Select cm with max csi_sg2, sum csi_sg and csi_sg2
18:    Add cm to cm_sg
19:    remove covered nodes from risk_nodes
20: Select countermeasures that impact separate vulnerabilities
21: Select the list cm_sg with max csi_sg
```

The goal of the countermeasure selection process is to reduce the common risk level for the analyzed system with minimum costs. We implement Algorithm 1 that maximizes the countermeasure selection index for each attack graph node to reach this goal:

(row 2): we start with the countermeasures with zero-cost expenses;

(row 4): if there are still attack graph nodes with risk level that exceeds predefined threshold (*risk_nodes*), we sort countermeasures *cms*1 that cover subgraph from the one that affects the largest number of the graph nodes and security properties (i.e. confidentiality, integrity and availability), the next countermeasures are selected according to the largest mismatch of the covered nodes and properties; *cms*1 is a matrix where on the same row there are countermeasures that cover the same number of graph nodes;

(row 6–11): we calculate *csi* for each countermeasure in the processed row of the *cms*1 and select the countermeasure with maximum *csi* (max *csi_sg*1); we add this max *csi_sg*1 to the common *csi* for the subgraph countermeasures (*csi_sg*) and we add an appropriate countermeasure to the list of the selected subgraph countermeasures (*cm_sg*); we repeat these steps while there are nodes in the *risk_nodes* or not processed rows in the *cms*1;

(row 12–19): we repeat steps from rows 4–11 for the countermeasures that cover separate attack graph nodes *cms*2; *csi* for these countermeasures are added to the common *csi csi_sg* and the selected countermeasures are added to the list *cm_sg*;

(row 20): we repeat steps from rows 4–11 for the countermeasures that cover separate vulnerabilities;

(row 21): we select generated countermeasures list *cm_sg* with maximum *csi_sg* for the implementation.

Several countermeasures for the separate graph nodes can have higher *csi* than one countermeasure for the subgraph, and several countermeasures for the separate vulnerabilities can have higher *csi* than one countermeasure for the graph node. That's why we obtain several alternative countermeasure lists for selection in the row 21.

6. Dynamic mode of calculations

We introduce dynamic mode of calculations for the reactive response to the detected cyber attacks. The selected countermeasures should prevent propagation of the detected attacks timely, i.e. before the severe damage. We outline three main phases of countermeasure selection. First, we map detected security incidents on the attack graph nodes (Incident mapping phase). This changes the state *St* and attack probability *Pr* (Definition 2) for the mapped nodes and launches the second phase (Risk recalculation). On the second phase the risks for the ancestors and descendants of the mapped graph nodes are recalculated. It allows determining the a priori and a posteriori steps of an attacker, i.e. to track the attack. If (when) the recalculated risk levels exceed the predefined threshold the third phase starts (Countermeasure selection).

The remaining of this section defines all aforementioned phases.

Definition 6 (Incident Mapping)**.** It follows an incident model E_i to process security incidents and responses under the reactive mode. The incidents are correlated from the security events by the correlation tools. E_i is a 3-tuple (T_i, H_i, Tp_i), where T_i is time of the incident; H_i is an object (e.g. host) affected by the incident; and Tp_i is the incident type. We map detected security incidents on the attack graph nodes to track ongoing attacks and update security assessments. We consider an object (e.g. host) H_i affected by the incident to decrease the number of attack graph nodes for mapping – only the nodes that correspond to the object H_i are selected (we use parameter H from the Definition 2). Then we use the incident type Tp_i to map an incident on the attack graph nodes that have appropriate post-conditions (we use parameter V from the Definition 2, where V post-conditions are specified via CVSS impact indexes). Possible incident types are CIA violation or illegitimate access. The descendants of the mapped graph nodes show possible future attack steps, while the ancestors show previous attack steps. We use time of the incident T_i to relate several incidents to the same attack sequence and to track direction of the attack propagation.

Definition 7 (Risk Recalculation)**.** Incident mapping changes the states *St* (to True), attack probabilities *Pr* and risk levels for the mapped nodes. In its turn it changes the risk levels for their ancestors and descendants (i.e. attack sequences that go through the compromised node). We recalculate probability values for the ancestors using Bayes theorem. The algorithm for determination of the previous attacker steps is based on the maximum probability change for the graph nodes. Determination of the previous attack steps, in its turn, allows determining the attacker skill level *asl*. We propose to calculate it as the maximum CVSS access complexity *AC* of these steps and use it to recalculate the local probability *p* for the compromised node descendants as $p = 2 \times \frac{\mathrm{AC+asl}}{2} \times \mathrm{Au}$, where the 2 and $\frac{1}{2}$ factors are used in order to get medium values from access complexity and *asl*, which results into a probability value from 0 to 1. Then we recalculate probability values for the descendants using the formula of total probability and new *p* values. The algorithm for determination of the following attacker steps is similar to the one for the previous attack step and it is based on the maximum probability change for the graph nodes.

Definition 8 (Reactive Countermeasure Selection)**.** A reactive countermeasure selection starts when risk for any attack graph nodes exceeds predefined threshold. It is based on the aforementioned processes of incident mapping and risk recalculation. Unlike preventive mode the goal of the countermeasure selection in the reactive is both to decrease risk to the acceptable level and to stop real instances of attacks identified in the system. The set of the available countermeasures is stored in the database. The countermeasures are specified using the set of parameters, including an affected vulnerability, impact area, impact type, affected security properties and implementation mode, i.e. some countermeasures selected in the preventive mode, maybe used in the reactive mode (e.g., firewalls can be used to enable/disable additional firewall rules). Thus the set of the available countermeasures in the reactive mode depends on the countermeasures set selected during the preventive mode. The attack action model presented in Section 5 is extended with available countermeasures for the countermeasure selection goals. The algorithm of the countermeasures selection is similar to the algorithm in the static mode except the set of the countermeasures for implementation and considered attack graph nodes.

The algorithm pseudo code is provided in Algorithm 2.

Algorithm 2 Pseudo code of the countermeasure selection algorithm in dynamic mode.

```
1:  Input data: security incident $E_i$
2:  Map $E_i$ on the attack graph
3:  Get graph_nodes corresponding to the $E_i$
4:  For all graph_node from graph_nodes
5:     $St = True$
6:     Recalculate $Pr$
7:     Recalculate risk
8:     For all ancestors of graph_nodes
9:       Recalculate $Pr$
10:      Determine previous attack steps attack_steps_pr
11:      Calculate attacker skill level $asl$
12:    For all descendants of graph_nodes
13:      Recalculate $Pr$ considering $asl$
14:      Recalculate risk
15:      if risk $>$ threshold add graph_node to risk_nodes
16: For all nodes from risk_nodes
17:    implement zero-cost patches
18:    redefine risk_nodes
19: Go to row 4 of Algorithm~~1
```

The *csi* index used for the countermeasure selection does not consider some parameters. Besides, the proposed countermeasure selection algorithm has high complexity as soon as it requires risk recalculation for each available countermeasure that limits its application in the near real time. We implement the StRORI index introduced in Section 3.2 to overcome this limitation.

7. Integration of the hypergraph-based approach and StRORI index

StRORI index and hypergraph-based approach are easily compatible as soon as attack graph allows one to outline attacks that should be prevented while StRORI allows one to select countermeasures for their mitigation. This section describes integration of the StRORI index to the countermeasure selection technique based on the attack graphs and provides brief description of the countermeasure selection prototype.

The proposed countermeasure selection technique integrated with StRORI metric extends the previous research presented in [9,29]. StRORI metric replaces *csi* metric specified in Section 5 and introduce all advantages of the StRORI metric to the graph-driven countermeasure selection process.

The connection between the attack graph and countermeasure selection technique (Section 3.2) is as follows: processing of each new incident and following implementation of countermeasures leads to the transition process between system states as represented in Fig. 1.

Table 2
StRORI and attack graph maping.

StRORI Model	Attack Graph Model
ALE	$AttackImpact \times AttackPotentiality$
AIV	$\Sigma Cost(i)$
RM	$Cov \times CE$
ARC	$CC + CD$

The mapping between all metrics composing the StRORI index and the metrics calculated based on the attack graph is depicted in Table 2.

The attack action model is extended to consider implemented countermeasures as described in Section 3.2 *Cdi*: $S = (H, V, Sc, St, Cdi, Pr)$.

In the current implementation the *annual loss expectancy* parameter of the StRORI metric can be instantiated by the attack impact calculated with the attack graph techniques provided in [9,29].

AIV is computed as the sum of costs of the policy enforcement points in the system and it was partially considered in Cost metric from Section 5.2 as the cost of tools required to implement countermeasures. But *AIV* is more complex index and its application should lead to more adequate countermeasure selection.

Countermeasures are described using the following set of parameters $CM = (V, P, M, Cov, AI, SI, CC, CE, CD)$, where V – vulnerability that is affected by the countermeasure, P – platform or configuration where the countermeasure can be implemented, M – system operation mode (static or dynamic), Cov – countermeasure coverage considering attack graph (subnet / subgraph / graph node / hosts / software and hardware / vulnerability), AI – impact on the network configuration or the attack graph (remove/add/modify node/link), SI – impact on the service dependency graph (remove/add/modify node/link). CC, CE, CD represent three indexes, countermeasure cost, countermeasure effectiveness and collateral damage from the countermeasure implementation, accordingly.

Countermeasure coverage areas shall be determined for all available countermeasures considering attack graph and corrected when new incidents are fixed. Countermeasure coverage is specified considering attack graph nodes with high risk level.

8. Implementation, use case and experiments

Let us provide a brief description of the developed security assessment and countermeasure selection prototype and application of the StRORI index in its scope. The prototype incorporates the following main architectural components: the component of input data gathering; the component of data processing; the security assessment component; the component of countermeasure selection; database; and visualization system. The components of input data gathering and processing gather and normalize security data from different sources, including open security databases, network scanners and experts. Gathered data are stored in the database. The security assessment component generates attack graph and implements risk calculation techniques. Calculation of the StRORI index is implemented for the experiments as the part of the countermeasure selection component. Visualization system represents obtained results. The prototype is implemented in Java version 1.8.0_45.

Further in this section we describe the results of joint application of attack graphs and StRORI for the countermeasure selection in preventive and reactive modes. For this goal we briefly describe attack scenario composed of multiple stages, assess the severity of

Table 3
Hardware and software of the test network [56].

Host	Software
Web server *Accreditation* (Massif-2)	Windows Server 2008 R2 (64 bits) JBoss AS 5.0.1 Snare agent ApacheStruts2 framework (cpe:2.3:a:apache:struts:2.0.11.2:*:*:*:*:*:*:*) (cpe:/a:apache:struts:2.0.11.2)
Web server *Sport Entries* (Massif-1)	Windows Server 2008 R2 (64 bits) JBoss AS 5.0.1 Snare agent ApacheStruts2 framework
Authentication server (Massif-3)	SUSE Enterprise Linux 11 SP1 (32 bits) NetIQ eDirectory server 8.8.7.1 (eDirectory 8.8 SP7 vanilla)
Internal firewall	SUSE Enterprise Linux 11 SP1 (32 bits) Netfilter

the attack in each stage and evaluate the different countermeasures using the StRORI.

8.1. Use case

We use the test case "Olympic Games" that was developed in scope of the MASSIF FP7 Project by AtoS company [56,57] and scenario of the "low and slow" attack to demonstrate our approach.

Network software and hardware are listed in Table 3. Web server Accreditation, web server Sport Entries and Authentication server are critical for this network. Accreditation and Sport Entries applications are accessible over the Internet. NetIQ eDirect is used for the authentication, eDirect data access is implemented using LDAP (Lightweight Directory Access Protocol) encapsulated in SSL (port 636); web applications Accreditation and Sport Entries use ApacheStruts2 framework (port 8080 is used for the web pages access) supported by JBoss AS (port 443).

The "low and slow" attack is implemented to get access to confidential data. In case of success, this attack can result in reputation loss and requires serious restoration costs. The attack steps are listed below.

Step 1 – scanning of the web servers for SQL injections. Result: no vulnerabilities. **Event 1** – unsuccessful attempt of SQL injection on the web servers.

Step 2 – behavior and code analysis. Result: SportEntries application uses the ApacheStruts2 framework.

Step 3 – searching for the ApacheStruts2 framework vulnerabilities. Result: vulnerabilities are detected (Struts version is vulnerable to OGNL injection).

Step 4 – exploitation of the vulnerability. Result: opportunity to execute processes on the web server (integrity violation).

Step 5 – searching for the JBoss Application Platform vulnerabilities (most likely application server for Struts2) to deploy remote shell. Result: remote execution vulnerability is detected.

Step 6 – deployment of the remote shell on the web server using JBoss Application Platform vulnerability to leverage the remote execution of processes. Result: remote shell is deployed.

Step 7 – attempt of the local administrative account brute force using remote shell and "slow and low" approach. This step took two weeks. Result: unsuccessful. **Event 2** – unsuccessful attempts of the local administrative account brute force on the web server.

Step 8 – attempt of the internal network recognition. This step took 1 month. Result: port tcp/ldaps (636) is opened (where NetIQ eDirect authentication server is running). **Event 3** – network scanning.

Step 9 – searching for the NetIQ eDirect server vulnerabilities. Result: vulnerability which allows remote privilege escalation is uncovered.

Step 10 – exploitation of the NetIQ eDirectory server vulnerability. This step took two weeks. Result: root on the Authentication server. **Event 4** – eDirectory process crashes multiple times.

Step 11 – get list of the valid user credentials from the Authentication server using network eavesdropping or brute force and apply consistently to the web server. This step took two weeks. Result: success – access to the web server. **Event 5** – attempt to login using various user credentials from the one host.

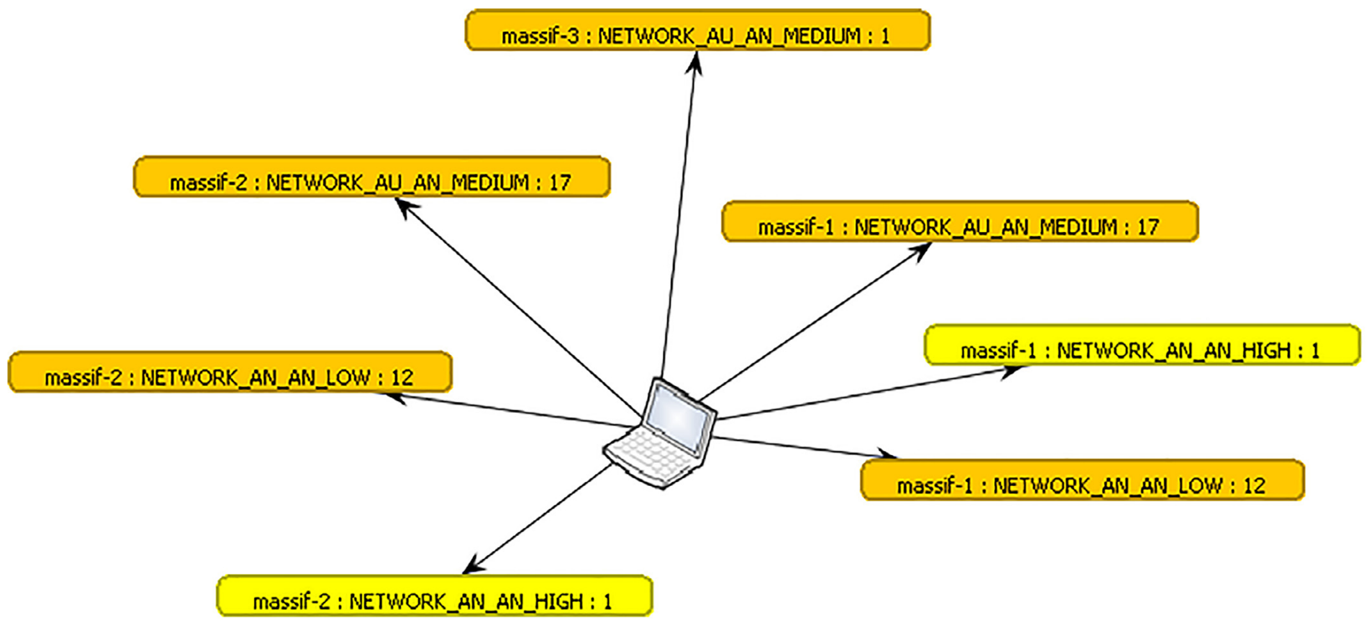
8.2. Preventive mode

At first, we calculated risks for the described system in preventive mode using the attack graph formalism described in Section 4 and the technique described in Section 5.1. Fig. 4,a represents an attack graph generated using our prototype for the described test network in the preventive mode. The nodes of the graph are colored depending on the calculated risks: green color for the low risk of compromise, yellow color for the medium risk level, orange color for the high risk level, and finally, red color – for the critical risk level. It should be noticed that there are only medium and high risk level nodes on the graph. There is firewall in the test network which coverage area covers nodes under the risk. As soon as there are no nodes with a critical risk level we do not select additional countermeasures on this step. More detailed views of the attack graph and demo of our prototype are available on-line at http://j.mp/stRORI.

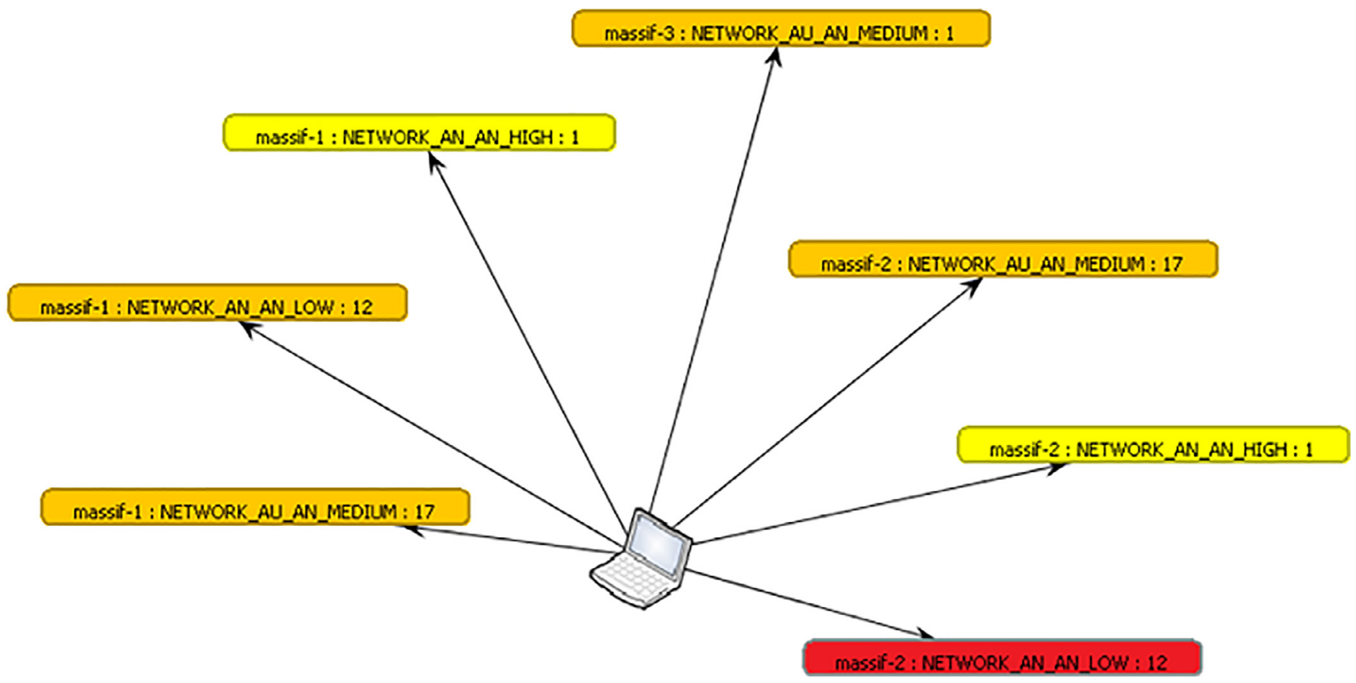
8.3. Reactive mode

In the reactive mode we process detected incidents to prevent serious impact. Let us demonstrate it for the attack sequence provided in Section 8.1. In terms of attack graph path, the sequence incorporates nodes that correspond to the CAPEC attack patterns and CVE exploitation. It can be represented as follows:

1. CAPEC-66: SQL Injection (scanning of the web servers for SQL injections); Indicator: Too many false or invalid queries to the database, especially those caused by malformed input (unsuccessful attempt of SQL injection on the web servers). Attackers' position – "Attacker" node in Fig. 4 as soon as our graph does not include CAPEC attack pattern nodes in current implementation.
2. CAPEC-118: Collect and Analyze Information (behavior and code analysis). Attackers' position – "Attacker" node in Fig. 4 as soon as our graph does not include CAPEC attack pattern nodes in current implementation.
3. No name action (searching for the ApacheStruts2 framework vulnerabilities). Result: Struts version is vulnerable to OGNL injection (CVE-2008-6504). Attackers' position – "Attacker" node in Fig. 4.
4. CVE-2008-6504 exploitation. Integrity Impact: Partial (opportunity to execute processes on the web server). "Massif-2" node in Fig. 4.
5. No name action (searching for the JBoss Application Platform vulnerabilities – most likely application server for Struts2 – to deploy remote shell). Result: remote execution vulnerability is detected (CVE-2017-12149). "Massif-2" node in Fig. 4.
6. CVE-2017-12149 exploitation (deployment of the remote shell on the web server to leverage the remote execution of processes). Confidentiality, Integrity and Availability Impact: Partial (remote shell is deployed). "Massif-2" node in Fig. 4.
7. CAPEC-70: Try Common or Default Usernames and Passwords (attempt of the local administrative account brute force using remote shell and "slow and low" approach). Attackers' position – "Massif-2" node in Fig. 4 as soon as our graph does not include CAPEC attack pattern nodes in current implementation.


massif-3 : NETWORK_AU_AN_MEDIUM : 1
massif-2 : NETWORK_AU_AN_MEDIUM : 17
massif-1 : NETWORK_AU_AN_MEDIUM : 17
massif-1 : NETWORK_AN_AN_HIGH : 1
massif-2 : NETWORK_AN_AN_LOW : 12
massif-1 : NETWORK_AN_AN_LOW : 12
massif-2 : NETWORK_AN_AN_HIGH : 1

(a) Static Mode of Calculations


massif-3 : NETWORK_AU_AN_MEDIUM : 1
massif-1 : NETWORK_AN_AN_HIGH : 1
massif-2 : NETWORK_AU_AN_MEDIUM : 17
massif-1 : NETWORK_AN_AN_LOW : 12
massif-2 : NETWORK_AN_AN_HIGH : 1
massif-1 : NETWORK_AU_AN_MEDIUM : 17
massif-2 : NETWORK_AN_AN_LOW : 12

(b) Dynamic Mode of Calculations

Fig. 4. Representation of the test attack graph generated using our proof-of-concept prototype. More details are available on-line at http://j.mp/stRORI.

Result: unsuccessful. Time: two weeks. Indicator: Many incorrect login attempts are detected by the system (unsuccessful attempts of the local administrative account brute force on the web server).

8. CAPEC-300: Port Scanning (attempt of the internal network recognition). This step took 1 month). Attackers' position – "Massif-2" node in Fig. 4 as soon as our graph does not include CAPEC attack pattern nodes in current implementation. Result: port tcp/ldaps (636) is opened (where NetIQ eDirect authentication server is running). Time: 1 month.
9. No name action (searching for the NetIQ eDirect server vulnerabilities). Result: vulnerability which allows remote privilege escalation is uncovered (zero-day). Attackers' position "Massif-2" node in Fig. 4.
10. Exploitation of the NetIQ eDirectory server vulnerability. Result: root on the Authentication server. "Massif-3" node in Fig. 4, no CVE node as soon as vulnerability was unknown. Time: two weeks. Indicator: eDirectory process crashes multiple times.
11. CAPEC-70: Try Common or Default Usernames and Passwords (get list of the valid user credentials from the Authentication

server using brute force and apply consistently to the web server). This step took two weeks. "Massif-2" node in Fig. 4 as soon as our graph does not include CAPEC attack pattern nodes in current implementation. Result: access to the web server. Time: two weeks. Indicator: Many incorrect login attempts are detected by the system.

Fig. 4 represents an attack graph generated using our prototype after incidents processing for the "low and slow" attack described in Section 8.1. An unsuccessful attempt of SQL injection on the web servers (step 1 of the attack) results in the first security incident. After processing of the first security incident the distribution of risk levels is not changed (as soon as attackers' position is outside the test network). After the second security incident (step 7 of the attack) the attackers' position is "Massif-2" node. The risk level of the attack graph nodes related to the "Massif-2" exceeds the threshold (Fig. 4,b). The following countermeasures were considered in the countermeasure selection process:

- 'enable/disable additional firewall rules with EF=80%, COV=0,7, ALE=€ 3000, ARC=€ 200, AIV= € 30000 and resulting StRORI=4,9;
- block suspicious connection with EF=80%, COV=1, ALE=€ 3000, ARC=0, AIV=€ 30000 and resulting StRORI=8;
- "block ports/IP addresses" with EF=80%, COV=1, ALE=€ 3000, ARC=€ 80, AIV=€ 30000 and resulting StRORI=7,7;
- "shutdown service/host" with EF=10%, COV=1, ALE=€ 3000, ARC=€ 80, AIV=€ 30000 and resulting StRORI=0,7.

As it was mentioned above the list of countermeasures for the implementation is constructed considering the StRORI index calculation the countermeasures with maximum StRORI index are selected. In our test case it is countermeasure "block suspicious con-

Table 4
Comparison among different countermeasure selection approaches.

Models	Advantages	Disadvantages
Service Dependency Framework [1]	It considers service dependencies in the process of selecting reaction strategies. It evaluates intrusion and response impact. It considers response collateral damages and positive response effects as they reduce intrusion costs.	Inability to evaluate monetary impact of selected security measures over its dependent services. Time is not considered in the computation of the risk impact. It does not consider the impact of new attack steps that are made possible by the current intrusion if no response is enacted.
Situation Calculus [15]	Integration of the graph theory with attack graph models. Automatic generation of attack graphs and appropriate responses. Evaluation of the global risk associated to simultaneous ongoing attacks	It does not estimate the risk of simultaneous attacks on the network service. It does not consider financial benefits of implementing countermeasures in the system.
Attack Graph Models [16,17]	It considers the implementation of preventive and reactive security measures on the exploitation of the system's vulnerabilities. The approach supports most attacks that could be discovered through network vulnerability scanners. It enables the quantification of the network to be compromised at different levels.	The model does not easily support attacks with multiple prerequisites. It requires improvements in terms of scalability and efficiency. Financial benefits of the implementation of security measures is not taken into account
Argumentation [18]	Suitable where multiple causes for a specific anomalous behavior are possible, and multiple countermeasures can be taken to mitigate the problem. It supports reasoning when direct resolution is not possible due to inherent, unresolved logical conflicts. It considers the cost of manipulating a final decision by acting on the decision process itself.	No real effort has been spent to use the approach in avoiding the manipulation of decision-making processes in Cybersecurity scenarios. No scoring system has been defined to be associated with Abstract Argumentation Frameworks (AAFs) in order to enable solution optimization.
RORI[7,22]	It allows the selection of multiple countermeasure. Response relative to the size of the infrastructure. It considers the case of selecting no countermeasure. It differentiates mutually exclusive measures from partially and/or totally restrictive ones.	High level of estimation in the computation of the model. Interdependency among countermeasures is not considered. No history of the actions is considered. No impact of suppressing a countermeasure.
Response Financial Impact Assessment (RFIA) [19]	It considers the financial benefits of restoring and protecting potentially threatened operational capabilities. It assesses potential impacts that efficient mitigation actions may inadvertently cause on the organization in an operational perspective. It uses a multi-dimensional optimization procedure to select response plans.	Inherent limitations associated to the RORI index e.g., accuracy issues, inability to consider indirect increase of financial costs, inability to evaluate potential decrease of financial impact, no consideration of semantic implications of individual countermeasures.
ROSI Framework [20]	The impact of an attack is computed on the whole business by considering its effect upon all critical assets. Uncertainty issues are partially overcome by using Bayesian theorems. It uses well-known standards (e.g., ISO-27001) for asset inventory development. It uses a mathematical formula to quantify and prioritize assets.	The framework needs to be tested in real time organizational environments. The methodology is not yet automated. It only analyses the impact of single security investment upon whole infrastructure, no methodology is provided for multiple and simultaneous investments.
Automated Countermeasure Selection Framework [21]	The framework provides an automated selection of countermeasures based on cost impact analysis. It considers the organisation's service loss and costs over a period of time. It uses both attacker and defensive actions in their analysis. Synthetic graphs are used to represent network dependencies and vulnerabilities.	Lack of a flexible dependency model. It does not select a combination of countermeasures and recovery actions in the same decision. Although the framework estimates the impact of disabling schemes based on network connectivity, it does not include this information in the attack or dependency graphs.
StRORI and Hypergraphs	It considers the state at which a countermeasure is implemented in the impact analysis. It allows the computation of exhaustive lists of attack scenarios by using hypergraphs. It estimates the risk of simultaneous attacks against the system and proposes optimal actions to act upon.	It requires a great level of accuracy in the estimation of parameters. The application of the risk assessment methodology is limited to the countermeasure selection algorithm complexity. Visualization of hypergraphs are limited to the methods and algorithms used

nection”. More detailed views of the attack graph and demo of our prototype are available on-line at http://j.mp/stRORI.

8.4. Experiments and discussion

We conducted experiments for the different generated attack sequences and network configurations. For the experiments we used PC with Intel Core i7 CPU and 8 GB RAM. We measured the next parameters: operating time of the countermeasure selection technique; risk values for the attack graph nodes before and after the implementation of the selected countermeasures; losses in case of attack implementation with and without the selected countermeasures.

In the reactive mode countermeasure selection process from the risk recalculation stage to the StRORI calculation and countermeasure selection stage takes no more than 2 s for the 500 hosts architectures. Comparison of the risk levels calculated using attack graphs showed risk mitigation to the predefined threshold in case of the countermeasures implementation. Comparison of losses showed significant benefit in the interval from 20 to 80 percent for the different attack sequences in case of the countermeasures implementation.

An application of the StRORI index gives a gain in time in the reactive mode of the countermeasure selection, and allows considering of the already implemented. While application of the graph-based approach increases the benefit from the countermeasures implementation due to the forecasting of the attack steps. Thus, we can conclude on the advantage of the joint application of the graph-based approach and StRORI index. Application of the suggested joint metric and graph-based countermeasure selection is justified for the large distributed networks to counteract ongoing multi-step cyber-attacks.

The developed tool and underlying approach use a cost-sensitive metric to evaluate efficiency of single and combined countermeasures against individual and multiple attack scenarios and to select the most suitable countermeasure or group of them for the implementation in a particular state of the system. The proposed metric also allows taking into account the case of selecting no countermeasure. It considers the size of the infrastructure which allows using it for the countermeasure evaluation in different systems regardless of their size. Compared to [1] the advantage of the proposed approach consists in the consideration of probability of attacks. Compared to the [15] the approach considers simultaneous attack scenarios. Compared to the [18] our approach considers the cost of the final decision in the decision process.

Compared to the previously proposed by the authors countermeasure selection model [7,22] it considers restrictions and conflicts among countermeasures as well as interdependence among countermeasures. Besides, the approach takes into account countermeasures history. The main limitation of the proposed model is requirements to the estimation accuracy for the parameters that compose the StRORI index. Our risk assessment technique allows overcoming this limitation by considering relative values on these parameters.

The basis of the risk assessment approach is Bayesian attack graph. Compared to [16] our graph allows one to support attacks with multiple prerequisites. The main limitation of Bayesian attack graphs application to forecast attack development and sources consists in the possibility of the attack sequences with cycles in the analyzed system. In this study we propose hypergraph approach to overcome this limitation. It distinguishes this study from [17] and our previous research on attack graphs [29,32]. Besides, integration of the hypergraph together with the StRORI index allows overcoming another limitation of our countermeasure selection technique on the basis of attack graphs in the near real time [9] related to the high complexity of the developed algorithm.

We introduce dynamics to the model using states *St* specified for each attack graph node. The set of states *St* defines system security state at each point of time. System can change its state from secure to insecure with some probability. Countermeasures should decrease this probability in the preventive mode and prevent state changing in the reactive mode, i.e. prevent propagation of the ongoing attack.

Table 4 compares the main advantage and disadvantage aspects associated to the different approaches for the countermeasure evaluation and selection.

9. Conclusion

We proposed in this paper a hybrid approach that combines attack graphs and a cost sensitive metric to analyze the impact of security countermeasures and select the optimal set of actions based on financial and threat impact assessment functions.

The main advantages of the approach described in this study are the following: (i) We use a cost-sensitive metric to evaluate efficiency of single and combined countermeasures against individual and multiple attack scenarios; (ii) The approach allows selecting the most suitable countermeasure or group of them for the implementation in a particular state of the system, (iii) The approach considers the size of the infrastructure, which allows using it for the countermeasure evaluation in different systems regardless of their size, (iv) The developed countermeasure selection index allows considering the case of selecting no countermeasure; (v) The approach considers restrictions and conflicts among countermeasures as well as interdependence among countermeasures; (vi) The approach takes into account countermeasures history.

In terms of limitations, we can observe that a great level of accuracy is required in the estimation of the different parameters of our construction. This is overcome by the use of a risk assessment methodology that considers relative values on all the elements composing the StRORI index. The proposed risk assessment methodology allows decreasing the risk of successful attacks in the preventive mode and preventing ongoing attacks in the reactive mode by forecasting attack development and sources and implementing countermeasures timely. In its turn, application of our risk assessment methodology was limited in real time with the countermeasure selection algorithm complexity. This is overcome by the use of hypergraph formalism and StRORI index.

Future work will concentrate in evaluating the approach in multi-step threat scenarios with multiple countermeasures to be analyzed simultaneously. In addition, we plan to extend the third type of hyperedges, and introduce the hyperedge that incorporates objects related to one incident on the different levels of abstraction, for example, events related to the incident, attack related to the incident, exploits used in this attack, vulnerabilities used in the attack, weaknesses used in the attack and countermeasures against the attack.

CRediT authorship contribution statement

Gustavo Gonzalez-Granadillo: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing - original draft, Visualization. **Elena Doynikova:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing - original draft, Visualization. **Joaquin Garcia-Alfaro:** Conceptualization, Methodology, Formal analysis, Supervision, Project administration, Funding acquisition. **Igor Kotenko:** Conceptualization, Methodology, Formal analysis, Supervision, Project administration, Funding acquisition. **Andrey Fedorchenko:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources.

Acknowledgment

The research was partially supported by the grants of the RFBR (18-07-01488, 18-29-22034, 19-07-01246), budgetary subject 0073-2019-0002, and by Government of the Russian Federation (Grant 08-08).

References

[1] Kheir N, Cuppens-Boulahia N, Cuppens F, Debar H. A service dependency model for cost-sensitive intrusion response. In: Proceedings of the 15th European symposium on research in computer security; 2010. p. 626–42.
[2] Jeffrey M. Return on investment analysis for e-business projects. In: Internet Encyclopedia, Hossein Bidgoli Ed., vol. 3; 2004. p. 211–36.
[3] Schmidt M. Return on investment (ROI): meaning and use. Encyclopedia of business terms and methods; 2011. Available at: http://www.solutionmatrix.com/return-on-investment.html
[4] Sonnenreich W, Albanese J, Stout B. Return on security investment (ROSI) - a practical quantitative model. J Res Pract Inf Technol 2006;38(1).
[5] Brocke J, Strauch G, Buddendick C. Return on security investment - design principles of measurement system based on capital budgeting. In: Proceedings of the conference of information systems technology and its applications; 2007. p. 21–32.
[6] Kim D, Lee T, In H. Effective security safeguard selection process for return on security investment. In: Proceedings of the Asia-Pacific services computing conference; 2008.
[7] Gonzalez-Granadillo G, Belhaouane M, Debar H, Jacob G. RORI-Based countermeasure selection using the orBAC formalism. Int J Inf Secur 2014;13(1):63–79.
[8] Kolomeec MV, Chechulin AA, Kotenko IV. Methodological primitives for phased construction of data visualization models. J Internet Serv Inf Secur (JISIS) 2015;5(4):60–84.
[9] Doynikova E, Kotenko I. Countermeasure selection based on the attack and service dependency graphs for security incident management. In: Proceedings of the international conference on risks and security of internet and systems; 2015.
[10] Ingols K, Chu M, Lippmann R, Webster S, Boyer S. Modeling modern network attacks and countermeasures using attack graphs. In: Proceedings of the computer security applications conference; 2009.
[11] Barany I. Applications of graph and hypergraph theory in geometry. In: Proceedings of the combinatorial and computational geometry, 52. MSRI Publications; 2005.
[12] Brandes U, Cornelsen S, Pampel B, Sallaberry A. Blocks of hypergraphs applied to hypergraphs and outerplanarity. In: Proceedings of the 21st international workshop on combinatorial algorithms (IWOCA 2010); 2010. p. 201–11.
[13] Pu L, Faltings B. Hypergraph learning with hyperedge expansion. In: Proceedings of the Joint European conference on machine learning and knowledge discovery in databases, vol. 7523; 2012. p. 410–25.
[14] Gonzalez-Granadillo G, Doynikova E, Kotenko I, Garcia-Alfaro J. Attack graph-based countermeasure selection using a stateful return on investment metric. Foundations and Practice of Security (FPS), Lecture Notes in Computer Science, vol. 10723. Springer; 2018.
[15] Samarji L, Cuppens F, Cuppens-Boulahia N, Kanoun W, Dubus S. Situation calculus and graph based defensive modeling of simultaneous attacks. In: Proceedings of the 5th international symposium on cyberspace safety and security - CSS; 2013.
[16] Lippmann RP, Ingols K, Scott C, Piwowarski K, Kratkiewicz K, vArtz M, Cunningham R. Validating and restoring defense in depth using attack graphs. In: Proceedings of the IEEE military communications conference (MILCOM 2006); 2006. p. 1–10.
[17] Poolsappasit N, Dewri R, Ray I. Dynamic security risk management using bayesian attack graphs. IEEE Trans Depend Secur Comput 2012;9(1):61–74.
[18] Martinelli F, Santini F. Debating cybersecurity or securing a debate? Found Pract Secur 2014:239–46.
[19] Motzek A, Gonzalez-Granadillo G, Debar H, Garcia-Alfaro J, Moller R. Selection of pareto-efficient response plans based on financial and operational assessments. J Inf Secur 2017.
[20] Yaqoob T, Arshad A, Abbas H, Amjad MF, Shafqat N. Framework for calculating return on security investment (ROSI) for security-oriented organizations. Future Gen Comput Syst 2019. doi:10.1016/j.future.2018.12.033.
[21] Soikkeli J, Munoz-Gonzalez L, Lupu EC. Effcient attack countermeasure selection accounting for recovery and action costs. In: Proceedings of the 14th international conference on availability, reliability and security; 2019.
[22] Gonzalez-Granadillo G, Garcia-Alfaro J, Alvarez E, El-Barbori M, Debar H. Selecting optimal countermeasures for attacks against critical systems using the attack volume model and the RORI index. Computers and Electrical Engineering Journal 2015;47:13–34.
[23] Kosutic D. Is it possible to calculate the return on security investment (ROSI)?. The ISO 27001 and ISO 22301 Blog, 27001 Academiy; 2011.
[24] Locher C. Methodologies for evaluating information security investments - what basel II can change in the financial ind. ECIS 2005.
[25] Consulting L. A guide for government agencies calculating return on security investment. Sydney, N.S.W.: Government Chief Information Office; 2004.
[26] Gonzalez-Granadillo G, Garcia-Alfaro J, Kotenko I, Doynikova E. Hypergraph–driven mitigation of cyber-attacks. Internet Technol Lett (ITL) 2018;1(3):6.
[27] Nespoli P, Papamartzivanos D, Mrmol FG, Kambourakis G. Optimal countermeasures selection against cyber attacks: a comprehensive survey on reaction frameworks. IEEE Commun Surv Tutor 2018;20(3):1361–96.
[28] Kotenko I, Stepashkin M. Attack graph based evaluation of network security. Commun Multimed Secur 2006:216–27.
[29] Kotenko I, Chechulin A. Computer attack modeling and security evaluation based on attack graphs. In: Intelligent Data Acquisition and Advanced Computing Systems; 2013. p. 614–19.
[30] MITRE. Common attack pattern enumeration and classification. 2019. Last accessed April.
[31] Wang L, Islam T, Long T, Singhal A, Jajodia S. An attack graph-based probabilistic security metric. In: Proceedings of the data and applications security conference; 2008. p. 283–96.
[32] Doynikova E, Kotenko I. Enhancement of probabilistic attack graphs for accurate cyber security monitoring. In: Proceedings of the 14th international conference on advanced and trusted computing. IEEE; 2017. p. 1492–7.
[33] Berge C. Hypergraphs: combinatorics of finite sets. North-Holland Mathematical Library, North-Holland, vol. 45; 1989.
[34] Vigna G. A topological characterization of TCP/IP security. In: FME 2003, 2805. Heidelberg: Springer; 2003. p. 914–39.
[35] Vigna G, Kemmerer RA. Netstat: a network-based intrusion detection approach. In: Proceedings of the ACSAC; 1998. p. 25–34.
[36] Hibaoui AE, Vallet L. Hypergraph model for anonymous communications. In: Proceedings of the ICMCS; 2012.
[37] Li Y, Shen H. Towards identity disclosure control in private hypergraph publishing. In: Proceedings of the 16th Pacific-Asia conference (PAKDD): advances in knowledge discovery and data mining, Part II; 2012.
[38] Morin B, Me L, Debar H, Ducasse M. M2d2: A formal data model for IDS alert correlation. In: Proceedings of the RAID, 2561; 2002. p. 115–27.
[39] Baiardi F, Suin S, Telmon C, Pioli M. Assessing the risk of an information infrastructure through security dependencies. In: Proceedings of the CRITIS, 4347. Heidelberg: Springer; 2006. p. 42–54.
[40] Pieters W. Ankh: Information threat analysis with actor-network hypergraphs. Enschede, Centre for Telematics and Information technology, University of Twente; 2010. Ctit technical report series.
[41] Li H, Wang Y, Cao Y. Searching forward complete attack graph generation algorithm based on hypergraph partitioning. Procedia Comput Sci 2017;107:27–38.
[42] Johnson CR, Montanari M, Campbell RH. Automatic management of logging infrastructure. In: Proceedings of the national centers of academic excellence - workshop on Insider Threat, St. Louis, MO, USA; 2010.
[43] Guzzo A, Pugliese A, Rullo A, Sacca D. Intrusion detection with hypergraph-based attack models. In: Proceedings of the workshop on graph structures for knowledge representation and reasoning (GKR); 2013.
[44] Zykov AA. Hypergraphs. Russ Math Surv 1974;29(6):89–154.
[45] Klarman S., Available at: u.-d.-w.-h.-e.. Modelling data with hypergraphs. 2017.
[46] Bergami G.. Hypergraph mining for social networks. In: University of Bologna, (2013/2014).
[47] Wolf M, Klinvex A, Dunlavy D. Advantages to modeling relational data using hypergraphs versus graphs. In: Proceedings of the IEEE high performance extreme computing conference (HPEC); 2016.
[48] Mow WH. Hypertrellis: a generalization of trellis and factor graph. In: Proceedings of the codes, systems, and graphical models. The IMA Volumes in Mathematics and its Applications, vol. 123; 2001. p. 167–94.
[49] Eschbach T, Gunther W, Becker B. Orthogonal hypergraph drawing for improved visibility. J Graph Algorithms Appl 2006;10(2):141–57.
[50] Kaufmann M, Van Kreveld M, Speckmann B. Subdivision drawings of hypergraphs. In: Proceedings of the 16th international symposium on graph drawing; 2009. p. 396–407.
[51] Anjum AN, Bressan S. Hypergraph drawing by force-directed placement. In: Proceedings of the 28th international conference on database and expert systems applications; 2017. p. 387–94.
[52] Valdivia P, Buono P, Plaisant C, Dufournaud N, Fekete JD. Analyzing dynamic hypergraphs with parallel aggregated ordered hypergraph visualization. IEEE Trans Vis Comput Graph 2019.
[53] Schneier B. Modelling security threats. Dr Dobb's J 1999.
[54] Cuppens F, Autrel F, Bouzida Y, Garcia-Alfaro J, Gombault S, Sans T. Anti-correlation as a criterion to select appropriate counter-measures in an intrusion detection framework. Ann Telecommun 2006;61(1):197–217. Springer
[55] Forum of incident response and security teams. 2019. Common Vulnerability Scoring System v3.0 Specification Document, last accessed April.
[56] MAnagement of security information and events in service infrastructures (MASSIF). FP7 Project (2010–2013).
[57] Atos company. Official website (https://atos.net/en/).
[58] Ingols K, Lippmann R, Piwowarski K. Practical attack graph generation for network defense. In: Proceedings of the 22nd annual computer security applications conference (ACSAC 2006); 2006.
[59] Barik MS, Sengupta A, Mazumdar C. Attack graph generation and analysis techniques. Def Sci J 2016;66(6):559–67.