Received December 2017; | Revised March 2018; | Accepted 2018

DOI: xxx/xxxx

LETTER

Hypergraph-driven Mitigation of Cyber-Attacks

G. Gonzalez-Granadillo[1] | E. Doynikova[2,3] | I. Kotenko[2,3] | J. Garcia-Alfaro[4]

[1]ATOS, Atos Research & Innovation, Cybersecurity Laboratory, Spain
[2]SPIIRAS, St. Petersburg Institute for Informatics and Automation, Russia
[3]ITMO University, St. Petersburg National Research University of Information Technologies, Mechanics and Optics, Russia
[4]IMT, Institut Mines-Telecom, Université Paris-Saclay, France

Correspondence
*Joaquin Garcia-Alfaro, Université Paris-Saclay, France. Email: jgalfaro@ieee.org

Summary

We extend a mitigation model that evaluates individual and combined countermeasures against multi-step cyber-attack scenarios. The goal is to anticipate the actions of an attacker that wants to disrupt a given system (e.g., an information system). The process is driven by a hypergraph formalism, enforced with a stateful return on response investment metric that optimally evaluates, ranks and selects appropriate countermeasures to handle ongoing and potential attacks.

KEYWORDS:
Mitigation, Return On Security Investment, Security Metrics, Attack Graphs, Network Security, Countermeasure Selection.

1 | INTRODUCTION

Quantitative financial metrics are useful in cyber security to evaluate mitigation plans and select the best countermeasures to handle attack scenarios[1]. The main drawback is the management of stateful meta-data, e.g., how countermeasures deployed at time $t-1$ shall be considered when evaluating a new group of countermeasures at time t. We argue that a plausible solution is the combination of quantitative financial metrics together with attack graph formalisms[2,3]. The challenge is data management, effective processing and visual perception when dealing with huge size attack graphs. An application of the hypergraphs allows us to reduce the size of processed attack graph by combining multiple links within one node[4].

In this letter, we propose to integrate a stateful quantitative financial metric, based on ROI-like (*Return on Investment*) concepts, together with a hypergraph model, in order to evaluate, rank and select optimal countermeasures based on financial and threat impact assessment functions. The resulting solution is evaluated at each state of the system while considering the already deployed countermeasures and the effects of adding or suppressing other security actions. Section 2 and 3 provide the background and review some related work. Section 4 proposes our solution. Section 5 describes the hypergraph-driven reduction of our attack graph model. Section 6 discusses limitations and advantages of our proposal. Section 7 concludes the paper.

2 | BACKGROUND

The Return On Response Investment (RORI) metric proposed in[1] is extended towards a new Stateful Return On Response Investment Metric (hereinafter denoted as StRORI). We assume a dynamic security monitoring process, where detection tools are permanently inspecting system and network events, in order to identify attack instances. To ease the presentation of the StRORI metric, we assume a discrete monitoring system based on temporal snapshots. Each snapshot provides a list with the different nodes affected in the attack scenario, as well as all the remainder security parameters. We propose to use attack graph specified in the next section for these snapshots. The evaluation process is assumed to be unique for each evaluation run and is performed using Equation 1.

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

We adapt the methodology proposed in[1] to compute the parameters composing the StRORI index, as presented in Table 1 .

TABLE 1 RORI Parameters

Parameter	Description	Computation	Improvements
ALE	Annual Loss Expectancy	$ALE = Cost \times ARO$, where $Cost$ refers to the monetary value associated to the severity of the attack, and ARO refers to the annual rate of occurrence	ALE depends directly on the severity and likelihood of the threat and it is independent on the countermeasure and the enforcement points.
AIV	Annual Infrastructure Value	$AIV = \sum_{i=0}^{n} AEC_i$, where AEC refers to the Annual Equipment Cost of all policy enforcement points (PEPs) that appears in the system's snapshot.	Unlike previous versions, AIV is presented as a dynamic parameter that changes its value at every snapshot of the system. It depends on the active PEPs of the system.
ARC	Annual Response Cost	$ARC = Ci + Cm + Cd + Odc + Ic$, where Ci= Cost of implementation, Cm = Cost of maintenance, Cd = Cost of deletion, Odc = Other direct costs; Ic = Indirect costs.	ARC considers the fact that a countermeasure (CM) is added, deleted or kept, as shown in Eq. 2.
RM	Risk Mitigation	$RM = EF \times COV$, where EF = countermeasure effectiveness, COV = countermeasure coverage.	RM considers the performed action (i.e., whether the countermeasure (CM) is added, deleted, or kept unchanged), as shown in Eq. 3.

$$if \begin{cases} CM(add) \rightarrow & ARC = Ci + Cm + Odc + Ic \\ CM(delete) \rightarrow & ARC = Cd \\ CM(keep) \rightarrow & ARC = 0 \end{cases} \tag{2}$$

$$if \begin{cases} CM(add) \rightarrow & RM = COV \cdot EF \\ CM(delete) \rightarrow & RM = 0 \\ CM(keep) \rightarrow & RM = Unchanged \end{cases} \tag{3}$$

As a result, the higher the $StRORI$ value, the better the countermeasure or set of countermeasures. More information about the computation of the $StRORI$ parameters can be found in[5]

3 | RELATED WORK

Several approaches have been proposed in the evaluation and selection of security countermeasures. Kheir et al.[6] for instance, propose a Service Dependency Framework (SDF) to assist the response process in selecting the enforcement points capable of applying a dynamic response rule. SDF introduces a requires/provides model of service dependencies and provides a systematic treatment of the dependency model which aims at applying policy rules while minimizing configuration changes and reducing resource consumption. The main drawback of this approach is the inability to evaluate the impact of selected responses over its dependent services.

Samarji et al.[7] propose to combine Graph Theory and a new formal description of individual, coordinated, and concurrent attacks in order to model attack graphs that cover the three attacks types. The approach 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.

Martinelli and Santini[8] suggest the use of Argumentation to provide automated support for security decisions. Argumentation supports reasoning when direct resolution is not possible due to inherent, unresolved logical conflicts. This reasoning adapts well to problems where multiple causes for a specific anomalous behavior are possible, and multiple countermeasures can be taken to mitigate the problem. 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.

More recently, Motzek et al.[9] 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 impact assessment (RFIA) based on a cost-sensitive metric (i.e., return on response investment) and a geometrical tool (i.e., attack volume model), as well as, a response operational impact assessment (ROIA) based on mission and resource dependency models. 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.

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.

4 | OUR PROPOSAL

This section specifies the foundations of our approach used for the countermeasure selection. It summarizes previous contributions presented in[3] that evolves the attack model by using a Bayesian approach[10] 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)**.** An attack graph is a graph $G = (S, L, \tau, Pc)$ 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 relation between attack actions, and Pc the discrete local conditional probability distributions.

Definition 2 (Attack Action)**.** An attack action is a 5-tuple $S = (H, V, Sc, St, Pr)$, where H identifies the attacked host, V the exploited vulnerability, Sc the process used by the attacker to get information about the host, and Pr the probability that the attack action is in state St ($Pr \in [0, 1]$).

By combining Definitions 1 and 2, we are able to represent all the possible attack actions (e.g., vulnerability exploitations and information gathering) and transitions between the actions of a multi-step attack scenario. In addition, stateful information is represented under the action states in St.

Definition 3 (Preventive risk calculation)**.** A precise level of risk is associated to each node of the attack graph. Risk levels are calculated on the scale from 0 (minimum) to 100 (maximum) for the attack graph nodes to outline threats (attack sequences) that should be prevented using the classic risk Eq. 5.

Definition 4 (Event mapping)**.** It follows an event model E_i to process security incidents and responses under the reactive mode, such that E_i is a 3-tuple (T_i, H_i, Te_i), where T_i is the event fixing time; H_i is the event fixing host; and Te_i is the event type. Events are mapped on the attack graph considering the event fixing host H_i. Graph nodes that correspond to the compromised host H_i are outlined. Then, considering event type Te_i (e.g., security properties violation or illegitimate access) attack graph nodes that have appropriate post-conditions are selected.

Definition 5 (Risk recalculation)**.** Mapping the security event on the attack graph results in recalculation of the risk levels for the attack sequences that go through the compromised node, considering new attack probability values. The probability for the previous nodes is recalculated using Bayes theorem, whereas for the next nodes we use the formula of total probability considering that the state of the compromised node is changed to True. The previous attacker steps are defined on the basis of the maximum probability change for the previous graph nodes. The attacker skill level *asl* is defined according to the maximum value of the Common Vulnerability Scoring System (CVSS) $AccessComplexity$ in these steps on the scale {0.35;0.61;0.71} (low, medium and high skills, accordingly)[11]. The attacker skill level *asl* is used for the recalculation of the local probability for the next graph nodes as depicted in Equation 4

$$p = \begin{cases} 2 \times \text{AccessVector} \times \frac{\text{AccessComplexity+asl}}{2} \times \text{Authentication} & \text{(root nodes)} \\ 2 \times \frac{\text{AccessComplexity+asl}}{2} \times \text{Authentication} & \text{(other nodes)} \end{cases} \tag{4}$$

In Equation 4 the 2 and $\frac{1}{2}$ factors are used in order to get medium values from $AccessComplexity$ and the attacker skill level (*asl*), which results into a probability value from 0 to 1.

Definition 6 (Countermeasure selection)**.** Based on the the aforementioned mapping and risk recalculation processes, a selection of countermeasures can now be conducted. This process relies on the mapping of real instances of attacks identified in the system. The countermeasures are enforced by the PEPs of the system during this phase whenever an attack reported by the system increases the accepted level of risk for some nodes (e.g., software tokens that can be used to enable multi-factor authentication). To select countermeasures we propose to use $StRORI$ index.

$$\text{Risk} = \text{AttackImpact} \times \text{AttackPotentiality} \tag{5}$$

The parameters composing Eq. 5 are presented in Table 2 . We use vulnerability scores from the open sources (namely, from vulnerability databases) to compute the risks automatically.

At this stage, new countermeasures are selected and activated to stop the propagation of ongoing attacks. On the basis of real instances of detected security violations, a priori and a posteriori steps of an attacker are mapped, and the level of risks of the attack-graph nodes is updated. The process undertakes three main phases: Event mapping, Risk recalculation, and Countermeasure selection.

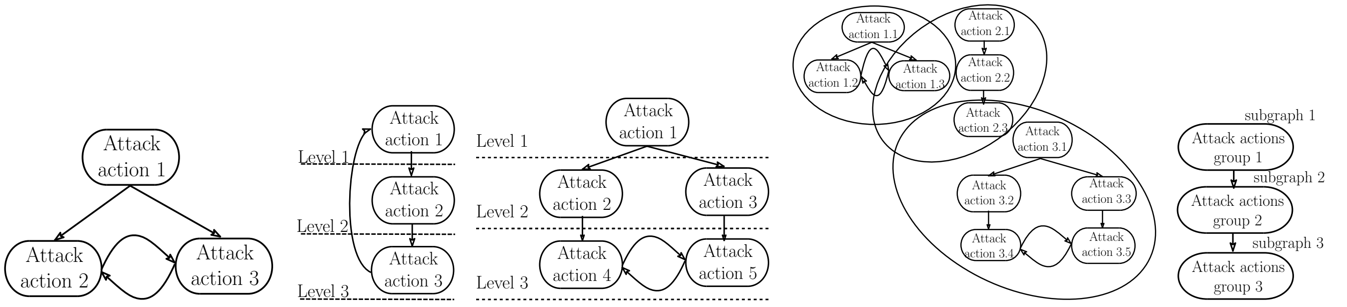
TABLE 2 Risk Parameters

Parameter	Description	Computation	Values
$AttackImpact$	Attack impact	$AttackImpact = cCrit \times cImpact + iCrit \times iImpact + aCrit \times aImpact$ (if $AttackImpact > 100$ then $AttackImpact = 100$)	[0;100]
$cImpact$, $iImpact$, $aImpact$	Potential damages of the asset confidentiality, integrity and availability, accordingly, in case of successful attack step (namely, successful exploitation of vulnerabilities)	Calculated considering Common Vulnerability Scoring System (CVSS) indexes $ConfidentialityImpact$, $IntegrityImpact$ and $AvailabilityImpact$ and their values[11], accordingly.	{0.0;0.275;0.660} (Low, Medium, and High impact, accordingly)
$cCrit$, $iCrit$, $aCrit$	Criticality of assets in terms of confidentiality, integrity and availability, accordingly	Calculated on the scale of ranks from 0 to 100 considering cost of the assets for the organization[2].	[0;100]
$AttackPotentiality$ (Pr)	Probability of attack success	$Pr = \prod_{n=1}^{i} Pc(S_i \mid Pa[S_i])$, where S – the attack graph node, $Pa[S]$ – the attack graph node descendants, and Pc – the attack graph node conditional probabilities (calculated on the basis of the local probabilities)[2].	[0;1]
$Localprobability$ (p)	Local probability of compromise for the graph nodes	$p = 2 \times AccessVector \times AccessComplexity \times Authentication$ for the attack graph root nodes, $p = 2 \times AccessComplexity \times Authentication$ for other nodes[2].	[0;1]
$AccessVector$	CVSS index that specifies access to the vulnerability[11]	Set using CVSS specification[11].	{0.395;0.646;1.0}
$AccessComplexity$	CVSS index that specifies complexity of the vulnerability exploitation[11]	Set using CVSS specification[11].	{0.35;0.61;0.71}
$Authentication$	CVSS index that specifies if additional authentication is needed to exploit vulnerability[11]	Set using CVSS specification[11].	{0.45;0.56;0.704}

5 | HYPERGRAPH-DRIVEN REDUCTION

Considering the work of Wang et al.[12], we outline three types of cycles and appropriate ways to process them[13] (see Figures 1 (a),(b) and (c)). Cycle of the type 1 (the graph nodes are located on the same level of the graph structure) can be removed as soon as the probability to reach "Attack action 2" and "Attack action 3" nodes directly from the "Attack action 1" node is higher than through the additional node. Cycle of the type 2 (the targeted node is located on the higher level of the graph structure than the source node) can be removed too as soon as for an attacker it has no sense to return back. Cycles of the type 3 should be processed as soon as we can not neglect them. We mark the links of the cycle as nonexistent and use them in the process of the attack graph analysis separately (input link is used for the calculations for the node considering that the output link does not exist). Notice that we use hypergraphs to implement aforementioned modifications.

In the hypergraph H an edge can incorporate arbitrary number of vertexes[14] $H = (X, U; R)$, 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. We specify each cycle as hypergraph edge and pre-process them to get acyclic graph (to use Bayesian approach). In Figure 1 (d) three subgraphs are consequentially linked in one hypergraph for the demonstration purposes, whereas in Figure 1 (e) the final acyclic graph is provided. This final acyclic graph can be used to calculate attack probabilities on the basis of Bayes theorem.



(a) Cycle of the type 1 (b) Cycle of the type 2 (c) Cycle of the type 3 (d) Initial hypergraph (e) Final hypergraph

FIGURE 1 (a, b, c) Types of cycles for the attack graph. (d) Sample hypergraph. (e) Final acyclic hypergraph.

6 | DISCUSSION

While Bayesian attack graphs are usable formalisms to forecast attack development and sources considering available subjective data, their application is limited by the possibility of the attack sequences with cycles in the computer network. The hypergraph approach is proposed to overcome this limitation by representing possible attack actions (e.g., vulnerability exploitations, information gathering) and transitions between them. This parameter and some others (i.e., affected vulnerability, impact area, impact type, affected security properties) are specified in the countermeasure model.

The proposed approach is evaluated at each state of the system (e.g., each node of the attack graph) by considering restrictions and inter-dependency among countermeasures (i.e., how the application of a countermeasure affects the effectiveness of others), as well as the previously implemented countermeasures in the system. As a result, we are able to evaluate not only the impact of adding a new security measure, but also the fact that some of the previously deployed actions could be modified and/or deactivated in a different state of the system. Adding a new countermeasure implies that the action was not previously executed (e.g., patching a given vulnerability). Deleting a countermeasure implies that we must suppress a previously executed action (e.g., unlock a given user or port that was previously blocked). Keeping a countermeasure unchanged implies that no operation is taken.

Dynamics is introduced to the model using states St as each time point system security state depends on the state of the graph nodes, i.e. set of states St. In the dynamic mode, countermeasures are selected to prevent propagation of the detected attack. On the basis of the detected security events, previous and next steps of an attacker are identified, and risks of the corresponding attack graph nodes are recalculated. The set of the available countermeasures is added to the database before the countermeasure selection process. The set of the available countermeasures for the reactive countermeasure selection depends on the equipment of the policy enforcement points mentioned in Sec. 2.

The main limitation of the countermeasure selection technique in the near real time consists in the high algorithm complexity as it requires risk recalculation for each selected countermeasure. The StRORI index described in Sec. 2 has been integrated to overcome this limitation.

7 | CONCLUSION

We present a countermeasure selection formalism that connects attack actions on the basis of pre and post conditions w.r.t. vulnerability exploitations and Bayesian probabilities. The approach enables us to represent all the possible attack actions (e.g., vulnerability exploitations and information gathering) and transitions between them on a multi-step attack scenario. As a result, it is possible to use a prior detecting of precise attack instances to evaluate local and global levels of risk in the system. The goal is to apply an initial set of policy enforcement points to reduce the global level of risk in the system. Further countermeasures are selected and enforced in a dynamic mode of the system operation once precise attacks have been detected and mapped to the attack graph. For the timely and rational countermeasure selection the StRORI index is used. Perspectives of future work include the evaluation of multi-step threat scenarios, featuring simultaneous analysis of multiple countermeasures at the same time.

AUTHOR CONTRIBUTIONS

All authors contributed equally to the manuscript.

CONFLICT OF INTEREST

The authors declare no potential conflict of interests.

ACKNOWLEDGEMENT

E. Doynikova and I. Kotenko acknowledge support from the grants of the RFBR (16-29-09482, 18-07-01488), partial support of budgetary subjects 0073-2015-0004, 0073-2015-0007, and Grant 074-U01.

References

1. Gonzalez-Granadillo G., Belhaouane M., Debar H., Jacob G. RORI-based countermeasure selection using the OrBAC formalism. *International journal of information security.* 2014;13(1):63–79.
2. Doynikova E., Kotenko I. Countermeasure Selection Based on the Attack and Service Dependency Graphs for Security Incident Management. *Conference on Risks and Security of Internet and Systems. Springer.* 2015;:107–124.
3. Gonzalez-Granadillo G., Doynikova E., Kotenko I., Garcia-Alfaro J. Attack Graph-based Countermeasure Selection using a Stateful Return on Investment Metric. *10th International Symposium on Foundations and Practice of Security.* 2017;.
4. Pu L., Faltings B. Hypergraph Learning with Hyperedge Expansion. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases.* 2012;7523:410–425.
5. Gonzalez-Granadillo G., Alvarez E., Motzek A., Merialdo M., Garcia-Alfaro J., Debar H. Towards an Automated and Dynamic Risk Management Response System. *Nordic Conference on Security IT Systems.* 2016;:37–53.
6. Kheir N., Cuppens-Boulahia N., Cuppens F., Debar H. A service dependency model for cost-sensitive intrusion response. *15th European Symposium on Research in Computer Security (ESORICS 2010).* 2010;:626–642.
7. Samarji L., Cuppens F., Cuppens-Boulahia N., Kanoun W., Dubus S. Situation Calculus and Graph Based Defensive Modeling of Simultaneous Attacks. *CSS.* 2013;8300:132–150.
8. Martinelli Fabio, Santini Francesco. Debating Cybersecurity or Securing a Debate?. *Symposium on Foundations and Practice of Security.* 2014;:239–246.
9. Motzek A., Gonzalez-Granadillo G., Debar H., Garcia-Alfaro J., Moller R. Selection of Pareto-efficient Response Plans based on Financial and Operational Assessments. *Journal on Information Security.* 2017;:60–84.
10. Poolsappasit N., Dewri R., Ray I. Dynamic security risk management using bayesian attack graphs. *IEEE Transactions on Dependable and Secure Computing.* 2012;9(1):61–74.
11. Forum of Incident Response and Security Teams . Common Vulnerability Scoring System v3.0 Specification Document Technical Paper, Last Accessed July 2017.
12. Wang L., Islam T., Long T., Singhal A., S. Jajodia. An Attack Graph-Based Probabilistic Security Metric. *22nd annual IFIP WG 11.3 working conference on Data and Applications Security.* 2008;:283–296.
13. Doynikova E., Kotenko I. Enhancement of probabilistic attack graphs for accurate cyber security monitoring. *14th International Conference on Advanced and Trusted Computing.* 2017;:1492–1497.
14. Zykov A. A.. Hypergraphs. *Russian Mathematical Surveys.* 1974;29(6):89–154.

□