# Attack Graph-based Countermeasure Selection using a Stateful Return on Investment Metric

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

[1] Atos Research & Innovation, Cybersecurity Laboratory, Spain
[2] St. Petersburg Institute for Informatics and Automation (SPIIRAS)
[3] Information Technologies, Mechanics and Optics (ITMO) University, Russia
[4] Télécom SudParis, Paris-Saclay University, CNRS SAMOVAR, France

**Abstract.** We propose 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 an attack graph formalism, enforced with a stateful return on response investment metric that optimally evaluates, ranks and selects appropriate countermeasures to handle ongoing and potential attacks.

## 1 Introduction

Network attacks are frequently represented as attack graphs, in order to identify the paths taken by an attacker in the exploitation of a given series of vulnerabilities, as well as to analyze all possible countermeasures that could be implemented to mitigate the attack [1, 12]. To compute exhaustive lists of possible attack scenarios, and to select the most effective countermeasures, attack graphs must rely on quantitative metrics that may base their analysis in cost-sensitive parameters.

With the above challenge in mind, we present the integration of a stateful return on response investment metric to the attack graph formalism presented in [2, 7]. The resulting combination allows to evaluate, rank and select optimal countermeasures based on complementary assessment functions (e.g., from both financial and security dimensions). The new metric is evaluated at each state of the system while considering the already deployed countermeasures and effects of adding or suppressing other security actions. Our contributions can be summarized as follows. We provide a network security model that evaluates individual and combined countermeasures against complex attack scenarios, in order to anticipate the actions of an attacker that wants to disrupt the security of a given system. The same process dynamically evaluates multiple countermeasure candidates while considering restrictions and inter-dependency among them. As a result, the optimal set of countermeasures is proposed and enforced over the system.

Paper Organization { Section 2 provides related work. Section 3 presents our construction. Section 4 concludes the paper.

## 2 Related Work

Kheir et al. [6] propose a process for the selection of security countermeasures by combining a service dependency framework and a cost-sensitive metric. The solution provides a systematic solution to applying policy rules while minimizing configuration changes and reducing resource consumption. Samarji et al. [11] combines a graph theoretic-solution and *situation calculus* to automatically generate mitigation graphs. Lippmann et al. [8] and Poolsappasit [10] use attack graph formalism to implement preventive and reactive countermeasures against vulnerability exploitation, accordingly. Martinelli and Santini [9] suggest the use *argumentation logic* to automate response reasoning under system attacks. The use of *argumentation logic* 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.

With regard to the aforementioned contributions, the approach presented in this paper may estimate the risk of simultaneous attacks against the system, and compute the cost of the final decisions by acting on the decision process itself, as well as, evaluate the impact of combined responses over dependent services. It builds over the attack graph formalism presented by Kotenko and Doynikova in [2, 7], complemented with a cost-sensitive metric that extends the work by Gonzalez et al. in [4,5]. The resulting formalism is used as an automated response selection mechanism, that anticipates forecasted steps of an attacker that aims at disrupting the security of a given system. The cost-sensitive metric builds upon the Return on Response Investment (RORI) index, initially proposed by Kheir et al. [6] as an extension of the Return On Security Investment (ROSI) index [13]. The metric provides a common reference to compare different countermeasures. Precise information about the computation of each specific parameter of the RORI index can be found in [4, 5].

## 3 Our Construction

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. It extends previous contributions presented in [2, 4, 7]. Its distinctive features are as follows: an opportunity of automated attack graph generation using network configuration and publicly available indexes for vulnerabilities; joint consideration of the attack probabilities and attack impact for the system assets; consideration of the attacker profile; connection with security events; preventive and reactive countermeasure selection. The goal is to represent, anticipate and handle attack actions performed by an attacker targeting a given system. We start with the core definitions. Then, we move to presenting the operation modes (e.g., preventive and reactive selection of countermeasures).

**De nition 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. $L \subseteq S \times S$), $\tau$ the relation between attack actions, and $P_c$ the discrete local conditional probability distributions.*

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

### 3.1 Preventive Mode, prior mapping of system attacks

By combining Definitions 1 and 2, we can now represent all the possible attack actions (e.g., vulnerability exploitations and information gathering) and transitions between the actions of a multi-step attack scenario [1, 12]. In addition, stateful information is represented under the action states in $S_t$. This enables the use of a preventive mode, prior detecting precise attack instances, to already evaluate both local and global levels of risk in the system. The goal is to apply an initial set of preventive countermeasures to reduce the global level of risk in the system. Further countermeasures, selected under a reactive mode, e.g., once precise attacks have been detected and mapped to the attack graph, are presented later in Section 3.2. Next, we provide definitions and processes used under the preventive mode.

**Definition 3 (Preventive Risk Calculation.)** *Under the preventive mode, a precise level of risk is associated to each node of the attack graph. It relies on a product combination of two main parameters:* AttackImpact×AttackPotentiality.

The value of the AttackImpact parameter (cf. Equation 1) is a linear combination of potential damages in terms of confidentiality, integrity and availability (denoted in Equation 1 as cImpact, iImpact, aImpact) of the asset in case of exploitation of vulnerabilities considering CVSS indexes [3]; as well as the criticality of such assets in terms of confidentiality, integrity and availability (denoted as cCrit, iCrit, aCrit in Equation 1).

$$\text{AttackImpact} = (\text{cCrit} \times \text{cImpact}) + (\text{iCrit} \times \text{iImpact}) + (\text{aCrit} \times \text{aImpact}) \quad (1)$$

The AttackPotentiality parameter refers to the vulnerability probability associated to each node of the graph. It is computed by using a total probability formula, considering both a local vulnerability probability $p$, and a conditional probability $P_c$ that considers all the possible states of its ancestors $P_a$. If compromising a node requires to compromise all the parent nodes, then $P_c$ is set to zero when it exists an $S_i$ in $P_a$ whose exploitation state is marked as $False$; otherwise, $P_c$ equals $p$. If compromising a node requires to compromise at least one parent node, then $P_c$ is set to zero when $\forall S_i \in P_a$ the exploitation state is marked as $False$; otherwise, $P_c$ equals $p$. The value of $p$ is computed as follows:

$$p = \begin{cases} 2 & \text{AccessVector AccessComplexity Authentication} \quad \text{(root nodes)} \\ 2 & \text{AccessComplexity Authentication} \quad \text{(other nodes)} \end{cases} \quad (2)$$

where AccessVector, AccessComplexity, and Authentication are extracted from the CVSS indexes [3] associated to the list of vulnerabilities defined for each node, and normalized between 0 and 1, using the 2 factor in Equation 2. The global estimation of the risk level of an attack sequence is defined as the combination of the minimum probability of the attack nodes and the maximum impact.

Based on the combination of AttackImpact and AttackPotentiality, we can now conduct a selection of countermeasures for those nodes of the graph with a risk level that exceeds a predefined threshold. The process is conducted by using a countermeasure selection index in terms of Efficiency, Cost and Collateral Damages associated with each countermeasure (or combination of countermeasures). The value of such an index can directly be obtained by using the RORI metric (cf. references [4, 5]).

The process (summarized in Figure 1) aims at maximizing the countermeasure selection index for each node of the graph. In turn, this leads to maximizing the reduction of risk as a whole. First, the countermeasures with zero-cost expenses are implemented (Step 1). A determination is made on whether or not there are still uncovered nodes (Step 2), so that countermeasures that impact over all the security properties are sorted according to their impact area (Step 2a) and a countermeasure selection index is calculated accordingly (Step 3). The measure that affects the largest number of the graph nodes and properties is selected, the next countermeasures are selected according to the largest mismatch of the covered nodes. If there are countermeasures that affect the same number of nodes, then multiple lists are generated (Step 2a) and the following steps are performed for all lists (the list with maximum countermeasure selection index is selected). If there are countermeasures that affect the same nodes, then multiple countermeasures are added on the same level of the list. Countermeasures that maximize the selection index are selected from the list of countermeasures that impact all the security properties on each level (i.e., confidentiality, integrity and availability).

If there are still uncovered nodes (Step 4), then countermeasures that impact two or less security properties are sorted similarly to the first list, starting from the measure that impacts the largest number of the not covered nodes (Step 4a). Similar rationale as in the first case is considered to compute the countermeasure selection index (Step 3). That is, if there are still nodes under risk (Step 4a), then countermeasures that impact separate vulnerabilities are selected (Step 5). In the end, the list with the maximum countermeasure selection index is selected and enforced, to conclude the process (Step 6).

### 3.2 Reactive Mode, posteriori to the mapping of system attacks

Under the reactive mode, 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 the phases defined below.

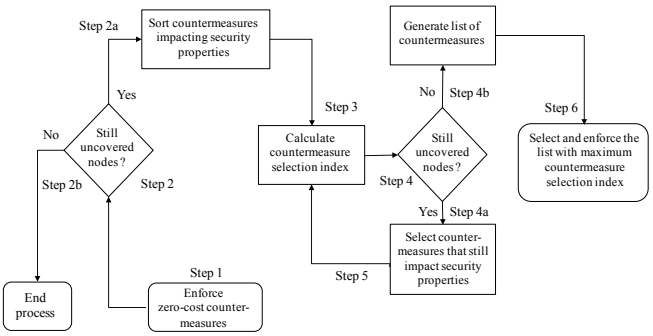



**Fig. 1.** Workflow of the preventing countermeasure selection process

**Definition 4 (Attack 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 ($T_i$, $H_i$, $T_{ei}$), where $T_i$ is the event fixing time; $H_i$ is the event fixing host; and $T_{ei}$ 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 $T_{ei}$ (e.g., security properties violation or illegitimate access) attack graph nodes that have appropriate post-conditions are selected.*

**Definition 5 (Risk Update.)** *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 is defined according to the maximum CVSS access complexity of these steps. The attacker skill level* asl *is used for the recalculation of the local probability for the next graph nodes as depicted in the following equation*

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

*where the 2 and $\frac{1}{2}$ factors are used in the above equations in order to get medium values from access complexity and attacker skills, which results into a probability value from 0 to 1.*

Based on the the aforementioned mapping and risk update processes, a reactive selection of countermeasures can now be conducted, whenever an attack reported by the system increases the accepted level of risk for some nodes. The main difference between the preventive and reactive mode relies on the mapping of real instances of attacks identified in the system. Some countermeasures may be selected during the preventive phase, but only enforced during the reactive

phase (e.g., software tokens that can be used to enable multi-factor authentication). This parameter and some others (i.e., affected vulnerability, impact area, impact type, affected security properties) are specified in the countermeasure model. The set of the available countermeasures is added to the database before the countermeasure selection process. The set of the available countermeasures in the reactive mode depends on the countermeasures set selected during the preventive mode. To conduct the reactive countermeasure selection process, the RORI metric proposed in [4, 5] is extended towards a new Stateful Return On Response Investment Metric (hereinafter denoted as StRORI), presented in the sequel.

### 3.3 Stateful Return On Response Investment (StRORI)

We propose an improvement in the computation of the parameters composing the formula in [4, 5], so that the new metric considers the state at which the RORI evaluation is performed. 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 that 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. The evaluation process is assumed to be unique for each evaluation run.

Figure 2 depicts a simple case with two transitions (i.e., from $T_0$ to $T_1$, and from $T_1$ to $T_2$). In the initial state of the system ($T_0$) we assume that no countermeasure from the authorized mitigation action list has been deployed. At $T_0$ we perform the RORI evaluation with two candidates (e.g., $C_1$, $C_2$) and we have three possible lists of countermeasures: (i) add $C_1$ (i.e., $L_{01} = \{+C_1\}$); (ii) add $C_1$ and $C_2$ (i.e., $L_{02} = \{+C_1 + C_2\}$); (iii) No operation, meaning that no mitigation action must be implemented (i.e., $L_{03} = \{\}$). In case the RORI index indicates the best action is to implement $L_{01}$, we implement $C_1$ and the state changes to $T_1$.

At $T_1$, we perform a new snapshot of the system that considers the number of active nodes and updates the system's configuration (e.g., consider previously implemented countermeasures). The RORI index is performed at this state by evaluating all authorized mitigation actions (even those already implemented in the system) to find the best list of countermeasures. Assuming that we evaluate $C_1$ and $C_2$, we will have four possible lists: (i) add $C_1$, meaning that no action

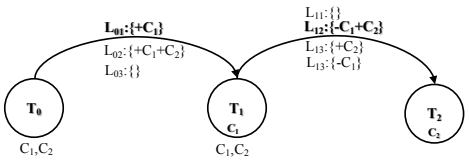


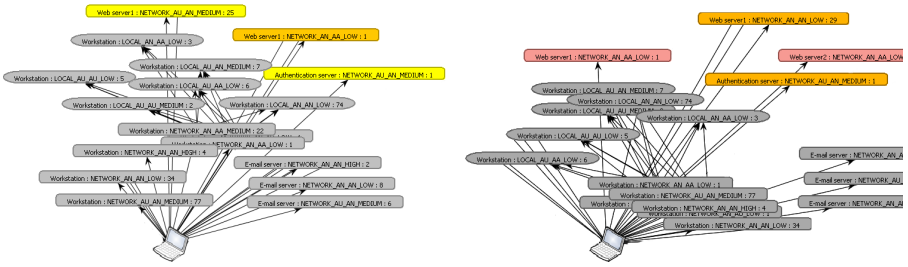

**Fig. 2.** Transition Process in the Stateful RORI evaluation

must be performed since $C_1$ is already implemented (i.e., $L_{11} = \{\}$); (ii) add $C_2$, meaning that $C_1$ must be uninstalled in order to install $C_2$ (i.e., $L_{12} = \{-C_1 + C_2\}$); (iii) add $C_1$ and $C_2$, meaning that only $C_2$ will be added since $C_1$ is already implemented (i.e., $L_{13} = \{+C_2\}$); and (iv) no operation, meaning that $C_1$ must be uninstalled since no mitigation action must be implemented (i.e., $L_{13} = \{-C_1\}$). 5 In case the RORI index at $T_1$ indicates the best action is to implement $L_{12}$, we must uninstall $C_1$ and install $C_2$ and the state changes to $T_2$. The process is repeated for a new snapshot of the system. A complete methodology for computing each parameter of the RORI metric, and related processes, is available in [4].

### 3.4 Validation of the Approach

The countermeasure selection process discussed in Section 3.3 allows extending the graph-driven selection process previously presented in [2, 7] by using the new countermeasure coverage areas provided by the StRORI metric. Such areas shall be computed for all the available countermeasures as soon as new attack instances are identified. Each state of the attack graph after a new attack event is processed leads to the transition state depicted in Figure 2. Countermeasure coverage is used to update those attack graph nodes whose risk level exceeds a predefined threshold.

Figure 3 shows a sample attack graph generated by our proposal. Sample attack graph representation generated by a proof-of-concept prototype. Low risk nodes are depicted in gray, medium risk nodes are depicted in yellow. High and critical risk level nodes that require preventive countermeasures are represented with orange and red colors, accordingly. The first security incident is generated as a result of the detection of a web-server vulnerability exploitation. After the processing of the security incident the next nodes are included to the list for the countermeasure selection as soon as risk levels for these nodes exceed the



(a) Preventive Mode (b) Reactive Mode

**Fig. 3.** Sample attack graph representation generated by our proof-of-concept prototype. Low risk nodes are depicted in gray, medium risk nodes are depicted in yellow. High and critical risk level nodes that require preventive countermeasures are represented with orange and red colors, accordingly. Further details and views of the attack graphs are available on-line at `http://j.mp/stRORI`

threshold: nodes that correspond to the Web server 1; nodes that correspond to the Web server 2; and nodes that correspond to the DB Server (Figure 3). For example we review the next countermeasures: shutdown service/host (EF=10%, COV=1, ALE=3000, ARC=80, AIV=30000); enable/disable additional firewall rules (EF=80%, COV=0,7, ALE=3000, ARC=200, AIV= 30000); block suspicious connection (EF=80%, COV=1, ALE=3000, ARC=0, AIV=30000); block ports/IP addresses (EF=80%, COV=1, ALE=3000, ARC=80, AIV=30000). Resulted StRORI index for the countermeasures: StRORI(shutdown service/host) = 0,7; StRORI (enable/disable additional firewall rules)=4,9; StRORI(block suspicious connection)=8; StRORI (block ports/IP addresses)=7,7. The selected countermeasures considering the maximum StRORI index: block suspicious connection. Further details and views of the attack graphs are available on-line at `http://j.mp/stRORI`.

### 3.5 Discussion

The main advantages of our ongoing construction are the following. We use a cost-sensitive metric to evaluate response goodness of single and combined actions against individual and multiple attack scenarios. The approach allows to rank and select the most suitable countermeasure or group of them against a given attack in a particular state of the system. The approach provides a response relative to the size of the infrastructure, which allows to compare the evaluation results of different systems regardless of their size. The model allows to handle the case of selecting no countermeasure, which results into a value of zero, meaning that no gain is expected if no solution is implemented. It also considers restrictions and conflicts among countermeasures (e.g., mutually exclusive, partially or totally restrictive countermeasures).

In addition, the proposed aproach considers interdependence among countermeasures (i.e., how the application of a countermeasure affects the effectiveness of others). We, therefore, consider the impact of adding, modifying and/or suppressing a series of countermeasures previously deployed or enabled in different parts of the system.

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.

## 4 Conclusion

We have proposed a mitigation security model that evaluates individual and combined countermeasures against multi-step attack scenarios. The process is driven by an attack graph formalism, enforced with a stateful return on response investment metric. The resulting construction optimally evaluates, ranks and selects appropriate countermeasures to handle the evolution of system risks. The approach provides preventive mitigation, prior identification of system attacks;

and reactive mitigation, once attacks instances have been mapped to the attack graph. Future work will concentrate on a more thorough analysis of the approach presented in this paper towards near-continuous time dimensional domains.

Acknowledgments { E. Doynikova and I. Kotenko acknowledge support from the Russian Science Foundation under grant number 15-11-30029. G. Gonzalez-Granadillo and J. Garcia-Alfaro acknowledge support from the European Commission under grant number 610416 (PANOPTESEC project).

## References

1. F. Cuppens, F. Autrel, Y. Bouzida, J. Garcia, S. Gombault, and T. Sans. Anti-correlation as a criterion to select appropriate counter-measures in an intrusion detection framework. *Annals of Telecommunications*, 61(1):197–217, 2006.
2. E. Doynikova and I. Kotenko. Countermeasure selection based on the attack and service dependency graphs for security incident management. In *Conference on Risks and Security of Internet and Systems*, pages 107–124. Springer, 2015.
3. Forum of Incident Response and Security Teams. Common vulnerability scoring system v3.0 specification document. Technical Paper, Last Accessed July 2017. version: release20170402.
4. G. Gonzalez-Granadillo, M. Belhaouane, H. Debar, and G. Jacob. RORI-based countermeasure selection using the OrBAC formalism. *International journal of information security*, 13(1):63–79, 2014.
5. G. Gonzalez-Granadillo, J. Garcia-Alfaro, E. Alvarez, M. El-Barbori, and H. Debar. Selecting optimal countermeasures for attacks against critical systems using the attack volume model and the RORI index. *Computers & Electrical Engineering*, 47:13–34, 2015.
6. N. Kheir, N. Cuppens-Boulahia, F. Cuppens, and H. Debar. A service dependency model for cost-sensitive intrusion response. In *15th European Symposium on Research in Computer Security (ESORICS 2010)*, pages 626–642. Springer, 2010.
7. I. Kotenko and A. Chechulin. Computer attack modeling and security evaluation based on attack graphs. In *Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), 2013 IEEE 7th International Conference on*, volume 2, pages 614–619. IEEE, 2013.
8. R. P. Lippmann, K. Ingols, K. P. C. Scott, K. Kratkiewicz, M. Artz, and R. Cunningham. Validating and restoring defense in depth using attack graphs. In *Military Communications Conference (MILCOM 2006)*, pages 1–10. IEEE, 2006.
9. F. Martinelli and F. Santini. Debating cybersecurity or securing a debate? In *Symposium on Foundations and Practice of Security*, pages 239–246. Springer, 2014.
10. N. Poolsappasit, R. Dewri, and I. Ray. Dynamic security risk management using bayesian attack graphs. *IEEE Transactions on Dependable and Secure Computing*, 9(1):61–74, 2012.
11. L. Samarji, F. Cuppens, N. Cuppens-Boulahia, W. Kanoun, and S. Dubus. Situation Calculus and Graph Based Defensive Modeling of Simultaneous Attacks. *CSS*, 8300:132–150, 2013.
12. B. Schneier. Modelling security threats. *Dr. Dobbs Journal*, 1999.
13. W. Sonnenreich, J. Albanese, and B. Stout. Return on security investment (ROSI)-a practical quantitative model. *Journal of Research and practice in Information Technology*, 38(1):45–56, 2006.