Decision-Making for Intrusion Response: Which, Where, in What Order, and How Long?

Yunchuan Guo*, Han Zhang*†, Yongjun Li*†, Fenghua Li*†, Liang Fang*, Lingcui Zhang*†§,
Jin Cao ‡, and Hui Li‡
* Institute of Information Engineering, Chinese Academy of Sciences, China
† School of Cyber Security, University of Chinese Academy of Sciences, China
‡ State Key Laboratory of Integrated Services Networks, School of Cyber Engineering, Xidian University, China
§Corresponding author (zhanglingcui@iie.ac.cn)

***Abstract*—Generating fine-grained response policies is a fundamental problem for Intrusion Response Systems (IRSs). Although existing schemes determine countermeasures and defense points efficiently, they ignore the deployment orders and execution durations of the selected countermeasures, which may impact response performance. To address this problem, by considering four attributes (i.e., attack damage, deployment cost, negative impact on QoS, and security benefit), we propose a decision-making framework for IRSs to reach fine-grained decisions to balance attack damage and response cost. We formulate decision-making as a single-objective optimization problem. To efficiently solve this problem, a Genetic Algorithm with Three-dimensional Encoding (GATE) is proposed to not only select countermeasures and defense points, but also determine deployment orders and execution durations. Simulation results demonstrate the efficiency of our approach.**

***Index Terms*—Intrusion response systems, Decision-making, Genetic algorithm with three-dimensional encoding, Response policy**

I. Introduction

In recent years, with the drastic growth of network scales, intrusions have become more sophisticated and have led to serious losses [1]. For example, due to an intrusion, the power system of Venezuela suffered a five-day power failure and 22 of 23 states were influenced[1]. To defend against attacks, it is crucial to design intrusion response systems (IRSs) and make appropriate response decisions to dynamically eliminate potential consequences and reduce security risks [2].

Existing decision-making schemes for IRSs mainly focus on countermeasure selection and deployment. During countermeasure selection: a single or multiple countermeasures are selected from a set to handle malicious activities and provide the trade-off among different attributes (e.g., attack damage and countermeasure benefit). In countermeasure deployment, the defense points are determined to deploy the selected countermeasures to minimize attack losses and maximize response benefits [3]. From the above discussion, we can see that these schemes mainly answer two questions: (**Q1**) **which** countermeasures should be selected to mitigate the attack and (**Q2**) **where** should the selected countermeasures be deployed. In practice, in addition to Q1 and Q2, to accurately respond to an intrusion event as much as possible, an effective decision-making scheme should answer the following questions:

(**Q3**) **In What order** are the selected countermeasures deployed? To prevent an intrusion event, multiple countermeasures may be selected. Generally, the deployment order of the selected countermeasures influences the response effectiveness. For example, in practice, countermeasures "*service migration*" and "*block traffic*", which are often used to eliminate a Distributed Denial of Service (DDoS) attack, can be deployed in two sequences: "*block traffic*" $\mapsto$ "*service migration*" and "*service migration*" $\mapsto$ "*block traffic*", where "*block traffic*"$\mapsto$"*service migration*" means that "*block traffic*" and "*service migration*" are deployed first and second, respectively. In the first sequence, after "*block traffic*" is successfully deployed and carried out[2], the resource occupied by the DDoS attack will be released, and the released resource can be used to deploy countermeasure "*service migration*", thus the losses caused by the attack can be significantly decreased. On the contrary, if the two countermeasures are deployed in the reverse sequence, then in the affected system, the resource required to deploy "*service migration*" would be insufficient because the DDoS attack will occupy a large amount of the resources. As a result, "*service migration*" cannot be carried out successfully with a high probability. From this example, we can see that the deployment order dramatically influences the response effectiveness.

(**Q4**) **How long** do the selected countermeasures last? Certainly, the execution durations of the selected countermeasures severely impacts the response effectiveness. For example, in the example of (Q3), the stop time of "*block traffic*" should be the moment at which that the attacked service is successfully migrated. If "*service migration*" is completed, but "*block traffic*" continues to be executed, then the quality of services will be dramatically decreased. Thus, an appropriate duration is required for the selected countermeasure.

To answer the above questions, we propose a decision-making framework for IRSs to generate a fine-grained policy to respond to an intrusion event. The major contributions of this paper are as follows:

[1] https://www.nwaonline.com/news/2019/mar/09/power-failure-raises-tension-in-venezue-1/

[2] In this paper, we assume that a countermeasure is executed immediately after deployment. In practice, there could be an interval between the deployment and the execution, which we will consider in future work.

- By considering four attributes (i.e., attack damage, deployment cost, negative impact on QoS, and security benefit), we propose a decision-making framework for IRSs to generate a fine-grained response policy by formulating the decision-making problem (i.e., "which countermeasures are selected? In which defense points and in what order are they deployed? How long are they executed?") as a single-objective optimization problem.
- To solve the above optimization problem efficiently, we propose a Genetic Algorithm with Three-dimensional Encoding (GATE) to determine not only countermeasures and defense points, but also deployment orders and execution durations. In particular, to evaluate the response utilities, we quantify the above four attributes by combining the response start time with vulnerability surface coverage and service dependencies. Simulation results demonstrate the efficiency of our approach.

The rest of this paper is organized as follows. Section II reviews the current related works. The decision making framework and the problem are presented in Section III. Section IV describes the details of GATE. Section V presents our experimental results. Finally, we conclude this paper in Section VI.

II. Related Work

As mentioned before, most existing efforts for decision-making in IRSs mainly focus on countermeasure selection and deployment.

Countermeasure selection: The existing schemes for selecting the countermeasure to balance the attack damage and response cost can be roughly divided into two categories: single countermeasure selection and multi-countermeasure selection. In the first category, by evaluating intrusion cost, collateral damage, and positive effects, Chung et al. [4] and Kheir et al. [5] adopted a return-on-response-investment (RORI) index to rank and select countermeasures. Considering the severity of attacks and using a knowledge-based decision table, Nadeem et al. [6] proposed an adaptive and flexible IRS to select an optimal countermeasure dynamically. Although a single countermeasure is efficient in single-path intrusion or in multi-path intrusion with cardinality 1, it is not valid in multi-path intrusions with other cardinality values, because in this case a single countermeasure cannot cut off attacks on multiple paths. One efficient scheme for the problem is to simultaneously select multiple countermeasures to reduce the potential risks and maximize the overall response utility [7], [8]. For example, considering the overlap of attack surface coverage of countermeasures, Granadillo et al. [9], [10] evaluated the benefits of all possible countermeasures combinations and selected an optimal set of countermeasures to maximize the RORI index. To reduce the solution space, Li et al. [11] adopted the Non-dominated Sorting Genetic Algorithm-II to optimize and determine multiple countermeasures dynamically. However, in these schemes, the influence of countermeasure deployment on the security benefit is ignored.

Countermeasure deployment: Much effort has been devoted on determining a defense point for the selected countermeasure. For example, considering the vulnerabilities' surface coverage, Shameli-Sendi et al. [12] formulated countermeasure deployment (including countermeasure selection) as a multi-objective optimization problem, and identified the optimal defense point from the Pareto optimal set that consists of the best possible solutions to balance security performance and cost. To characterize the relationships between countermeasures and defense points, Fessi et al. [13] proposed a multi-attribute genetic algorithm to select countermeasures and defense points that have the least negative effects on the whole system by encoding an individual as a binary matrix of "countermeasure-defense point". From the perspective of overall optimization, Li et al. [14] adopted a greedy algorithm to identify multiple defense points and defend multi-path attacks, by considering the value of defense points. However, in these schemes, the influence of deployment orders and execution durations on the security benefit is ignored.

III. Decision-Making Framework and Problem Statement

As shown in Fig. 1, the decision-making framework for IRSs, receiving alerts from Intrusion Detection Systems (IDSs), generates the response policy by evaluating attack damage, deployment cost, negative impact on QoS, and security benefit. In our framework, a response policy includes four basic metrics: the selected countermeasures, the defense points, the deployment orders, and execution durations. To accurately state our problem, we first define the notations used, as follows.

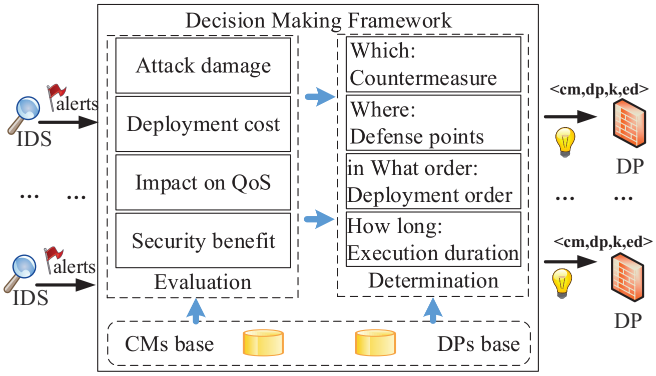
$CM = \{cm_0, cm_1, \cdots, cm_{n-1}\}$ is a set of predefined countermeasures stored in the database, where n is the number of the candidate countermeasures. For example, a countermeasure may be "patch", "reboot", or "block traffic".

$DP = \{dp_0, dp_1, \cdots, dp_{m-1}\}$ is a set of the defense points where the countermeasures can be deployed, where m is the number of candidate defense points. For example, a defense point may be a firewall, a router, or a web server.

$k \in K$ $(K \subseteq \{0\} \cup N^+)$ denotes the deployment order of the selected countermeasure deployed on a defense point, where N^+ is the set of positive integers. In practice, we often set an upper bound q $(q < mn)$ of the countermeasure's deployment order.

$ed \in \{0\} \cup N^+$ denotes the execution duration unit of the selected countermeasure. Note: in our work, we assume that the execution duration of the selected countermeasure is discrete. In practice, we often set an upper bound ed_{max} and a lower bound ed_{min} of the countermeasure's duration.

Given the above notations, we define a response policy $\mathcal{I}$ for an intrusion event as a set of meta-policies (i.e., $\mathcal{I} = \{\mathbb{I}_1, \mathbb{I}_2, \cdots, \mathbb{I}_l\}$), where the meta-policy $\mathbb{I} = \langle cm, dp, k, ed \rangle$ denotes that the selected countermeasure cm deployed on the defense point dp in the k-th order is executed with a duration of ed.



Note: “IDS” represents a intrusion detection system, “CMs” represents countermeasures, and “DPs” represents defense points.

Fig. 1. Our Decision Making Framework

The goal is to generate an appropriate response policy for an intrusion event to maximize the response utility. In our work, the response utility relies on four attributes: attack damage, deployment cost, negative impact on QoS, and security benefit, explained as follows. (1) *Attack damage* (denoted by AD) is the amount of system damage caused by an attack. (2) *Deployment cost* (denoted by $DC(\mathcal{I})$), refers to the resource cost to deploy policy $\mathcal{I}$. (3) *Negative Impact on QoS* (denoted by $IQ(\mathcal{I})$) is the negative impact of the activated policy $\mathcal{I}$ on QoS. (4) *Security benefit* (denoted by $SB(\mathcal{I})$) is used to denote the improved security performance when policy $\mathcal{I}$ is activated. Section IV gives the schemes for evaluating the four attributes in detail.

Given AD, $DC(\mathcal{I})$, $IQ(\mathcal{I})$, and $SB(\mathcal{I})$, we use the return-on-response-investment (RORI) index [9] to define response utility $\phi(\mathcal{I}) = SB(\mathcal{I})/(AD + DC(\mathcal{I}) + IQ(\mathcal{I}))$. Given the above notations, we formulate the decision making for IRSs as a single-objective optimization problem, as follows.

$$\begin{aligned} &\mathcal{I}^* = \arg max\{\phi(\mathcal{I})\} \\ s.t.\ \ &SB(\mathcal{I}) > 0,\ \ DC(\mathcal{I}) < \omega_{dc},\ \ IQ(\mathcal{I}) < \omega_{iq} \end{aligned} \tag{1}$$

where $\mathcal{I}^*$ is the optimal policy, ω_{dc} and ω_{iq} are the thresholds of acceptable deployment cost and negative impact on QoS, respectively.

According to the above discussion, we see that the response policy for an intrusion event with n countermeasures, m defense points, q deployment orders and $(ed_{max} - ed_{min} + 1)$ duration time units has $((ed_{max} - ed_{min} + 1)^{n \cdot m \cdot q})$ candidate meta-policies. In other words, the number of the candidate meta-policies increases exponentially with the number of countermeasures, defense points, deployment orders, and durations. Thus, it is difficult to obtain exact solutions to this problem.

IV. Genetic Algorithm with Three-Dimensional Encoding

In this paper, we propose GATE to quickly obtain an appropriate response policy from the candidate meta-policies.

A. 3D Encoding

In GATE, each individual is encoded as a solution of the decision-making problem, where each solution is a policy consisting of a set of meta-policies. According to the meta-policy definition described in Section III, we take the four metrics (i.e., countermeasures, defense points, deployment orders, and execution durations) into consideration to design the encoding scheme. In practice, for given cm_i, dp_j, and k, the corresponding execution duration is a unique value (denoted as $ed_{i,j,k}$). Thus, a three-dimensional array can be used to encode an individual (denoted as $\mathcal{I}$) in GATE, where the three dimensions are candidate countermeasures, defense points, and deployment orders, respectively. Each element (i, j, k) in the array and its value $ed_{i,j,k}$ forms a meta-policy $\mathbb{I}_{i,j,k} = \langle cm_i, dp_j, k, ed_{i,j,k}\rangle$, which means that the selected countermeasure cm_i is deployed on the defense point dp_j in the k-th order and executed for $ed_{i,j,k}$ time units. If the value of an element (i, j, k) is larger than zero (i.e., $ed_{i,j,k} > 0$), then $\mathbb{I}_{i,j,k}$ is a valid meta-policy. All valid meta-policies for an attack form its final response policy $\mathcal{I}$.

Fig. 2 gives an example of individual encoding, where there are two valid meta-policies ($\mathbb{I}_{4,3,2} = \langle cm_4, dp_3, 2, 5\rangle$ and $\mathbb{I}_{6,5,1} = \langle cm_6, dp_5, 1, 1\rangle$, respectively). This policy means that countermeasure cm_6 is first deployed on defense point dp_5 with a duration of 10 time units and countermeasure cm_4 is second deployed on defense point dp_3 with a duration of 5 time units.

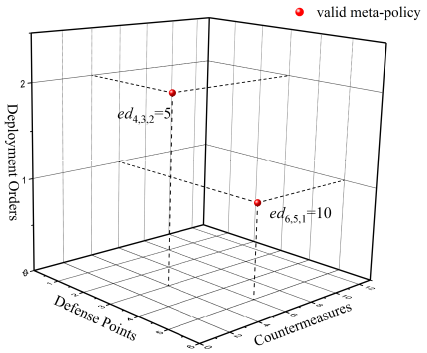


Fig. 2. Individual Encoding Example

In practice, some countermeasures cannot be deployed on specific defense points. Besides, we assume that: (1) Multiple countermeasures cannot be deployed on the same defense point in the same deployment order. (2) A countermeasure cannot be deployed in multiple deployment orders on the same defense point. We formally formulate the characteristics and assumptions as three constraints, as follows.

Constraint 1. Let $\Psi(cm) = \{dp_1, dp_2, \ldots, dp_{m'}\}$ denote the set of defense points where cm can be deployed. For $cm_i \in CM$, there exists no $k \in K$ such that $dp_j \in DP \backslash \{\Psi(cm_i)\}$ and $ed_{i,j,k} > 0$.

Constraint 2. Given $cm_a \in CM$, $dp_j \in DP$, and $k \in K$, if $ed_{a,j,k} > 0$, then there exists no $cm_i \in CM \backslash \{cm_a\}$ such that $ed_{i,j,k} > 0$.

Constraint 3. Given $cm_i \in CM$, $dp_j \in DP$, and $c \in K$, if $ed_{i,j,c} > 0$, then there exists no $k \in K\backslash\{c\}$ such that $ed_{i,j,k} > 0$.

B. Genetic Algorithm Procedure

GATE includes five operations: population initialization, crossover, mutation, fitness calculation, and natural selection, discussed as follows.

1) Population Initialization: In GATE, the initial population is randomly generated by creating N three-dimensional arrays. Note: during the population initialization, constraints 1, 2 and 3 should be satisfied to avoid the problem of premature convergence.

2) Crossover: The crossover of a population is defined as follows.

(1) Three individuals are selected (without replacement) randomly from the population to form three pairs.

(2) For each pair of individuals ($\mathcal{I}_A$ and $\mathcal{I}_B$), we first create a cube block with random size and location for one individual, then the cube block with the same size and location for the other individual. Two descendant individuals ($\mathcal{I}'_A$ and $\mathcal{I}'_B$) are generated using the following operator.

$$\begin{cases} ed(\mathcal{I}'_A)_{i,j,k} = \lceil \mu \cdot ed(\mathcal{I}_A)_{i,j,k} + (1-\mu) \cdot ed(\mathcal{I}_B)_{i,j,k} \rceil \\ ed(\mathcal{I}'_B)_{i,j,k} = \lceil \mu \cdot ed(\mathcal{I}_B)_{i,j,k} + (1-\mu) \cdot ed(\mathcal{I}_A)_{i,j,k} \rceil \end{cases} \tag{2}$$

where $ed(\mathcal{I}_A)_{i,j,k}$ and $ed(I_B)_{i,j,k}$ are the values of element (i,j,k) in the block of $\mathcal{I}_A$ and $\mathcal{I}_B$ respectively, $ed(\mathcal{I}'_A)_{i,j,k}$ and $ed(\mathcal{I}'_B)_{i,j,k}$ are the values of the element (i,j,k) in the descendants $\mathcal{I}'_A$ and $\mathcal{I}'_B$ respectively, $\mu = \phi(\mathcal{I}_A)/(\phi(\mathcal{I}_A) + \phi(\mathcal{I}_B))$ is a weight factor, and $\phi(\mathcal{I})$ is the response utility function defined in Section III.

(3) The six descendant individuals generated by the three pairs of individuals are added to the offspring population.

(4) Repeat (1)-(3) until the size of the offspring population reaches $2N$.

3) Mutation: To improve the diversity of the population, we apply a non-uniform mutation operator to each individual in the population with a probability pm. For each mutated individual, we create a cube block with random size and location.

4) Fitness calculation: To simplify the fitness calculation, we first check whether an individual satisfies constraints 1, 2 and 3. If the individual violates these constraints, then its fitness value is set to zero. Otherwise, we employ the utility function defined in Section III to evaluate its fitness value.

Similarly to [15], we adopt three impact functions describing the impact of attack damage, security benefit and negative impact on QoS over time, as follows.

(a) Constant impact function (CIF):

$$\varphi = w_1 (w_1 \geq 0) \tag{3}$$

(b) Linear impact function (LIF):

$$\varphi = \begin{cases} w_1 * t + w_2 & \text{if } w_1 * x \geq -w_2, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

(c) Exponential impact function (EIF):

$$\varphi = e^{w_1 * t + w_2} \tag{5}$$

where weight factors w_1 and w_2 can be set independently.

As shown in Section III, the utility function includes four parts: *attack damage*, *deployment cost*, *negative impact on QoS*, and *security benefit*. The respective evaluation approaches are discussed as follows.

(1) *Attack Damage* changes over time and can be evaluated by using the cumulative sum of the elementary impacts of the attack in the interval $[0, \lambda]$, where λ is the response start time, formally defined as:

$$AD = \int_0^{\lambda} \varphi_{AD}(t)\, dt \tag{6}$$

where $\varphi_{AD}(t)$ is an instance of the predefined impact functions (i.e., CIF, LIF and EIF).

(2) *Deployment Cost.* The deployment cost $DC(\mathcal{I})$ of policy $\mathcal{I}$ (or individual $\mathcal{I}$, since individual $\mathcal{I}$ corresponds to a policy, as mentioned above) is the sum of the deployment costs of its meta-policies, defined as:

$$DC(\mathcal{I}) = \sum_{i<n} \sum_{j<m} \sum_{k<q} DC_{\mathbb{I}_{i,j,k}} \tag{7}$$

where $DC_{\mathbb{I}_{i,j,k}}$ denotes the cost of deploying meta-policy $\mathbb{I}_{i,j,k}$. Similarly to [14], it is evaluated using the resource consumption of the selected countermeasure as well as the importance of the corresponding defense point, defined as:

$$DC_{\mathbb{I}_{i,j,k}} = \begin{cases} DT(cm_i)^{\alpha} \cdot RS(cm_i)^{\beta} \cdot Im(dp_j)^{\gamma} & ed_{i,j,k} > 0, \\ 0 & ed_{i,j,k} = 0. \end{cases} \tag{8}$$

where $DT(cm_i)$ is the time consumed to deploy countermeasure cm_i, $RS(cm_i) \in N^+$ is the resource consumption level of countermeasure cm_i, and $Im(dp_j) \in N^+$ is the importance level of defense point dp_j. α, β and γ are the weight factors.

(3) *Negative Impact on QoS.* Although we obtain positive security benefits through the deployment of countermeasures, the latter also lead to some negative impacts on QoS. For example, a web service may be interrupted during the execution of "*block all traffic*". Due to the service dependencies, the negative impact on QoS can be divided into two categories: the direct impact and indirect impact, defined as:

$$IQ(\mathcal{I}) = \sum_{i<n} \sum_{j<m} \sum_{k<q} (DIQ_{\mathbb{I}_{i,j,k}} + IIQ_{\mathbb{I}_{i,j,k}}) \tag{9}$$

where $DIQ_{\mathbb{I}_{i,j,k}}$ and $IIQ_{\mathbb{I}_{i,j,k}}$ denote the direct and indirect impact of deploying meta-policy $\mathbb{I}_{i,j,k}$, respectively. The direct impact is defined as:

$$DIQ_{\mathbb{I}_{i,j,k}} = \sum_{s_p \in S_{\mathbb{I}_{i,j,k}}} Im(s_p) \times Imq_{s_p, \mathbb{I}_{i,j,k}} \tag{10}$$

where $S_{\mathbb{I}_{i,j,k}}$ is the set of services directly influenced by meta-policy $\mathbb{I}_{i,j,k}$, $Im(s_p) \in [0,1]$ denotes the importance of service s_p, and $Imq_{s_p,\mathbb{I}_{i,j,k}}$ is the impact of meta-policy $\mathbb{I}_{i,j,k}$ on the quality of service s_p, denoted as:

$$Imq_{s_p,\mathbb{I}_{i,j,k}} = \int_0^{ed_{i,j,k}} \varphi_{IQ_{s_p,\mathbb{I}_{i,j,k}}}(t)\, dt \tag{11}$$

where function $\varphi_{IQ_{s_p,\mathbb{I}_{i,j,k}}}(t)$ is one of the predefined impact functions (i.e., CIF, LIF and EIF).

The indirect impact $IIQ_{\mathbb{I}_{i,j,k}}$ can be evaluated as:

$$IIQ_{\mathbb{I}_{i,j,k}} = \sum_{s_p \in S_{cm_i}} \sum_{s_l \in D(s_p)} d_{l,p} \times Ims_l \times Imq_{s_l,\mathbb{I}_{i,j,k}} \tag{12}$$

where $D(s_p)$ denotes the set of services depending on service s_p and $d_{l,p} \in [0,1]$ represents the dependency degree of service s_l on service s_p.

(4) *Security Benefit.* For individual $\mathcal{I}$, we use vulnerabilities surface coverage [12] to define security benefit $SB(\mathcal{I})$, as follows.

$$SB(\mathcal{I}) = \sum_{i<n} \sum_{j<m} \sum_{k<q} \sum_{v \in V(cm_i, dp_j)} \Delta SB_{v,\mathbb{I}_{i,j,k}} \tag{13}$$

where $V(cm_i, dp_j)$ is the set of vulnerabilities covered by countermeasure cm_i deployed on the defense point dp_j, and $\Delta SB_{v,\mathbb{I}_{i,j,k}}$ is the increased benefit brought by the meta-policy $\mathbb{I}_{i,j,k}$, defined as follows.

$$\Delta SB_{v,\mathbb{I}_{i,j,k}} = \begin{cases} 0 & SB_{v,\mathbb{I}_{i,j,k}} - SB_{v,\mathbb{I}_{i',j',k'}} \leq 0, \\ SB_{v,\mathbb{I}_{i,j,k}} - SB_{v,\mathbb{I}_{i',j',k'}} & otherwise. \end{cases} \tag{14}$$

where $SB_{v,\mathbb{I}_{i,j,k}}$ is the security benefit of meta-policy $\mathbb{I}_{i,j,k}$ on the coverage of vulnerability v and $\mathbb{I}_{i',j',k'}$ is the meta-policy with the largest security benefit on the coverage of vulnerability v before the meta-policy $\mathbb{I}_{i,j,k}$ is deployed. $SB_{v,\mathbb{I}_{i,j,k}}$ is defined as:

$$SB_{v,\mathbb{I}_{i,j,k}} = E_{cm_i,dp_j} \cdot TSB_{\mathbb{I}_{i,j,k}} \tag{15}$$

where E_{cm_i,dp_j} is the effectiveness of countermeasure cm_i deployed on defense point dp_j. $TSB_{\mathbb{I}_{i,j,k}}$ is the cumulative sum of the elementary impacts of the benefit of the meta-policy $\mathbb{I}_{i,j,k}$ in the interval [ts, te] and is evaluated as:

$$TSB_{\mathbb{I}_{i,j,k}} = \begin{cases} \int_{ts}^{te} \varphi_{SB_{\mathbb{I}_{i,j,k}}}(t)\, dt & ed_{i,j,k} > 0, \\ 0 & ed_{i,j,k} = 0. \end{cases} \tag{16}$$

where $\varphi_{SB_{\mathbb{I}_{i,j,k}}}(t)$ can be one of the predefined impact functions. ts and $te = ts + ed_{i,j,k}$ are the start and end time units of executing countermeasure cm_i.

5) Natural Selection: During natural selection, we keep the N best individuals for the next generation and discard other individuals from the population. Our algorithm terminates if one of the following conditions is satisfied: (C1) the number of iterations is greater than the predefined threshold σ_1, or (C2) the improvement degree of the best fitness value is less than the predefined threshold σ_2 for a predefined number sn of generations.

Given the above operations, we design the genetic algorithm with three-dimensional encoding as shown in Algorithm 1.

V. Evaluation

A. Experimental Setup

To validate the effectiveness of our framework, we establish an experimental network consisting of a trust zone and a demilitarized zone (DMZ).

Algorithm 1: GATE

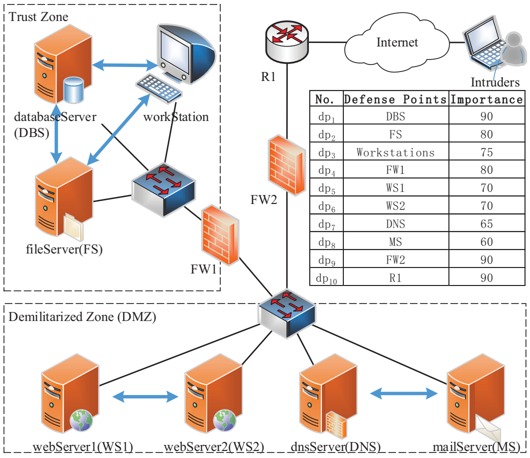
```
Input: $CM$, $DP$, $ed_{max}$, $ed_{min}$, $q$, and the attack alert;
Output: response policy $\mathcal{I}$;
Algorithm phase:
Randomly generate initial population $P$ of size $N$
($N \leq 3$) while satisfying the predefined constraints;
Evaluate the attack damage $AD$ using the input alert;
while none of the predefined termination conditions (i.e.,
(C1) and (C2)) is satisfied do
    Generate a new population $P_{new} \leftarrow NULL$;
    while $|P_{new}| < 2N$ do
        Randomly select three individuals $\mathcal{I}_A$, $\mathcal{I}_B$ and
        $\mathcal{I}_C$ from $P$ without replacement;
        ($\mathcal{I}_{AB}$, $\mathcal{I}_{BA}$) $\leftarrow$ Crossover($\mathcal{I}_A$, $\mathcal{I}_B$);
        ($\mathcal{I}_{BC}$, $\mathcal{I}_{CB}$) $\leftarrow$ Crossover($\mathcal{I}_B$, $\mathcal{I}_C$);
        ($\mathcal{I}_{AC}$, $\mathcal{I}_{CA}$) $\leftarrow$ Crossover($\mathcal{I}_A$, $\mathcal{I}_C$);
        $P_{new} \leftarrow P_{new} \cup \{\mathcal{I}_{AB}, \mathcal{I}_{BA}, \mathcal{I}_{BC}, \mathcal{I}_{CB}, \mathcal{I}_{AC}, \mathcal{I}_{CA}\}$;
    end
    Apply a non-uniform mutation operator to each
    individual in $P_{new}$ with a probability $pm$;
    Calculate the fitness value of each individual in
    $P_{new}$ by evaluating attack damage $AD$, deployment
    cost $DC$, negative impact on QoS $IQ$ and security
    benefit $SB$;
    Select $N$ best individuals in $P_{new}$ to generate the
    next generation $P'$;
    $P \leftarrow P'$;
end
```

As shown in Fig. 3, the trust zone comprises an administrator's workstation, a database server (DBS) and a file server (FS). The DMZ comprises four servers (i.e., web servers 1 and 2 (WS1 and WS2), a DNS server (DNS), and a mail server (MS), respectively). The full topology is available in our GitHub repository. Assuming that the FS is the target of attackers and 34 candidate countermeasures are available to respond to the attack (i.e., $n = 34$). Because the FS is in the trust zone, it cannot be directly accessed by an attacker. If attackers want to invade the FS, they have to achieve their goal indirectly by first invading WS1 or WS2. The table in Fig. 3 presents the importance scores of ten defense points (i.e., $m = 10$).

In our experiments, we set the default value of the parameters to $pm = 0.1$, $N = 210$, $\sigma_1 = 500$, $\lambda = 0.1$, $q = 3$, $\sigma_2 = 10^{-8}$, $sn = 20$, $ed_{min} = 0$ and $ed_{max} = 100$. To evaluate the influence of the impact function on fitness, three impact functions (i.e., CIF, LIF, and EIF, as shown in Section IV) were considered. For simplicity, an attack is associated with one of the three impacts, i.e. the CIF, the LIF, or the EIF scenario, respectively. We set the parameters of the three functions as follows: $w_1 = 40$ for CIF, $w_1 = 35$ and $w_2 = 35$ for LIF, $w_1 = 1$ and $w_2 = 10$ for EIF, respectively.

B. Decision-Making Process

1) Decision-making results: Experimental results show that our approach can generate an appropriate response policy ef-



No.	Defense Points	Importance
dp_1	DBS	90
dp_2	FS	80
dp_3	Workstations	75
dp_4	FW1	80
dp_5	WS1	70
dp_6	WS2	70
dp_7	DNS	65
dp_8	MS	60
dp_9	FW2	90
dp_{10}	R1	90

Fig. 3. Experimental Network Topology

fectively. Taking the LIF scenario as an example, the generated policy contains 3 meta-policies (i.e., ⟨*BlockIPPort(WS1,http)*, *WS1*, 0, 11⟩, ⟨*BlockIPPort(WS2,http)*, *WS2*, 0, 26⟩, and ⟨*CloseConnection(ftp)*, *FS*, 1, 1⟩), which means that countermeasures "*BlockIPPort(WS1,http)*" and "*BlockIPPort(WS2,http)*" are first deployed on WS1 and WS2 respectively, and their executions last 11 and 26 time units. Then, the countermeasure "*CloseConnection(ftp)*" is deployed on the FS with an execution duration of 1 time unit. The generated policy is reasonable: when an attack is detected, the first thing that should be done is to forbid access of external users to WS1 and WS2. Then, the next thing is to avoid the negative impact on the FS by closing any connections to it.

2) Response start time v.s. fitness: Fig. 4 presents the change of the fitness value of the fittest individual over response start time λ. From this figure, we see that the fitness value decreases as λ increases in the three scenarios. This means that we should respond to intrusion events as early as possible. Obviously, the fitness value in the EIF scenario is much smaller than that in the other two scenarios due to the accumulation effect of the attack damage. Besides, the fitness value in the LIF scenario is often greater than that in the CIF scenario in the case of $\lambda < 0.5$. The result is opposite in the case of $\lambda > 0.5$. This phenomenon is to be expected: higher attack damages mean lower fitness.

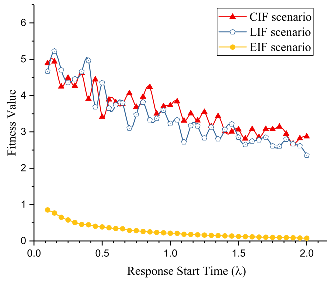


Fig. 4. Response start time v.s. fitness

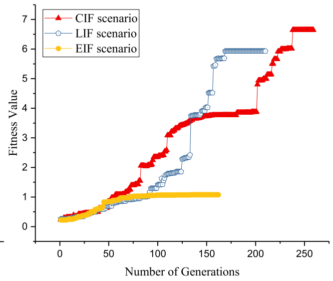


Fig. 5. Generations v.s. fitness

3) Generations v.s. fitness: Fig. 5 shows the trend of fitness values over the number of generations, where we see that the fitness value increases monotonically with the number of generations. When GATE terminates, the best fitness value obtained in the CIF scenario is greater than that of the other scenarios.

C. Computational Efficiency

As described in section III, the solution space of the decision-making problem is $(ed_{max} - ed_{min} + 1)^{nmq}$. Obviously, the time complexity of a traverse algorithm and GATE are $O((ed_{max} - ed_{min} + 1)^{nmq})$ and $O((TNnmq(ed_{max} - ed_{min} + 1))^2)$ respectively, where T is the number of generations when the terminal conditions are met and N is the population size. From the above results, we see that if $(ed_{max} - ed_{min} + 1) \leq (TNmnq)^{(2/(nmq-2))}$, then the traverse algorithm is more efficient, otherwise GATE is more appropriate. Fig. 6 and 7 compare the traverse algorithm with GATE from the aspect of computing time. Note: according to the above results, n, m and q influence the time complexity of the two algorithms equally, thus we randomly select one parameter (i.e., n) to conduct our experiments. Each approach was implemented for 100 runs. Fig. 6 and 7 show that when $n > 3$ or $ed_{max} > 4$, the consumed time of the traverse algorithm increases exponentially with the increasing of n and ed_{max}, while the consumed time of GATE almost remains constant. This means that GATE is preferable to the traverse algorithm.

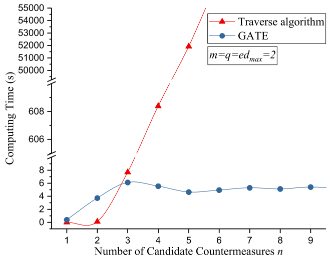


Fig. 6. Number of candidate countermeasures v.s. computing time

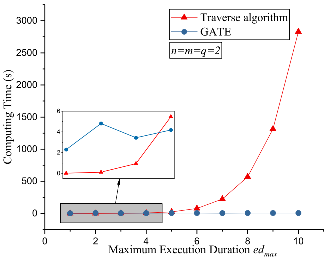


Fig. 7. Maximum execution duration v.s. computing time

VI. Conclusions

In this paper, by considering four attributes (i.e., attack damage, deployment cost, negative impact on QoS, and security benefit), a decision-making framework for IRSs is proposed to generate a fine-grained policy. We formulate decision-making as a single-objective optimization problem. To solve the problem efficiently, we use three-dimensional encoding for individuals and propose an extended genetic algorithm to not only select countermeasures and defense points, but also to determine deployment orders and execution durations. Experimental results show the efficiency of our approach.

References

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

[2] Z. Inayat, A. Gani, N. B. Anuar, M. K. Khan, and S. Anwar, "Intrusion response systems: Foundations, design, and challenges," *Journal of Network and Computer Applications*, vol. 62, pp. 53–74, 2016.

[3] A. S. Sendi, M. Cheriet, and A. Hamou-Lhadj, “Taxonomy of intrusion risk assessment and response system,” *Computers & Security*, vol. 45, pp. 1–16, 2014.

[4] C. Chung, P. Khatkar, T. Xing, J. Lee, and D. Huang, “NICE: network intrusion detection and countermeasure selection in virtual network systems,” *IEEE Transactions on Dependable and Secure Computing*, vol. 10, no. 4, pp. 198–211, 2013.

[5] N. Kheir, N. Cuppens-Boulahia, F. Cuppens, and H. Debar, “A service dependency model for cost-sensitive intrusion response,” in *In Proc. of ESORICS 2010*.

[6] A. Nadeem and M. P. Howarth, “An intrusion detection & adaptive response mechanism for manets,” *Ad Hoc Networks*, vol. 13, pp. 368–380, 2014.

[7] M. Gunasekharan, S. Basu, and G. R. Santhanam, “Selecting the minimal set of preferred responses to counter detected intrusions,” in *In Proc. of CISRC 2017*.

[8] G. Gonzalez Granadillo, S. Dubus, A. Motzek, J. García, E. Alvarez, M. Merialdo, S. Papillon, and H. Debar, “Dynamic risk management response system to handle cyber threats,” *Future Generation Comp. Syst.*, vol. 83, pp. 535–552, 2018.

[9] G. Gonzalez Granadillo, M. Belhaouane, H. Debar, and G. Jacob, “Rori-based countermeasure selection using the orbac formalism,” *International Journal of Information Security*, vol. 13, no. 1, pp. 63–79, 2014.

[10] G. Gonzalez Granadillo, G. Jacob, H. Debar, and L. Coppolino, “Combination approach to select optimal countermeasures based on the rori index,” in *In Proc. of INTECH 2012*.

[11] X. Li, C. Zhou, Y.-C. Tian, and Y. Qin, “A dynamic decision-making approach for intrusion response in industrial control systems,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 5, pp. 2544–2554, 2019.

[12] A. S. Sendi, H. Louafi, W. He, and M. Cheriet, “Dynamic optimal countermeasure selection for intrusion response system,” *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 5, pp. 755–770, 2018.

[13] B. A. Fessi, S. Benabdallah, N. Boudriga, and M. Hamdi, “A multi-attribute decision model for intrusion response system,” *Information Sciences*, vol. 270, pp. 237–254, 2014.

[14] F. Li, Y. Li, Z. Yang, Y. Guo, L. Yin, and Z. Wang, “Selecting combined countermeasures for multi-attack paths in intrusion response system,” in *In Proc. of IEEE ICCCN 2018*.

[15] B. A. Fessi, M. Hamdi, S. Benabdallah, and N. Boudriga, “A decisional framework system for computer network intrusion detection,” *European Journal of Operational Research*, vol. 177, no. 3, pp. 1824–1838, 2007.

[16] T. N. Dinh and M. T. Thai, “Assessing attack vulnerability in networks with uncertainty,” in *In Proc. of IEEE INFOCOM 2015*.