MT2M: Strategic Cost-based Optimization of Cyber Defense in Variable Constraints Systems

Jamil Ahmad Kassem[1,2*], Helena Rifà Pous[1,2] and Joaquin Garcia-Alfaro[3]

[1*]UOC-TECH Research Center, Computer Science, Multimedia and Telecommunication Studies, Universitat Oberta de Catalunya (UOC), Rambla del Poblenou, 154-156, Barcelona, 08018, Spain.

[2]CYBERCAT - Center for Cybersecurity Research of Catalonia, Spain.

[3]SAMOVAR, Télécom SudParis, Institut Polytechnique de Paris, 91120 Palaiseau, France.

Abstract

When confronted with advanced cyber threats, traditional cybersecurity methods often struggle with system performance and cost-effectiveness. Current approaches using game theory for passive applications, such as Moving Target Defense (MTD), lack flexibility in adapting to varying resource criticalities and rely on rigid cost assumptions that overlook the interdependencies among system components. Additionally, these approaches are complex and grow exponentially more complex with the network. This paper presents the *Multiple Target* Moving Target Defense *(MTD) Model* (MT2M), a strategic MTD framework based on Bayesian Stackelberg game theory to optimize cyber defense costs. Optimization is done while considering both resource criticality and node capacity. By transforming a complex, NP-hard cost problem into a linear one, MT2M enables scalable deployment. Numerical simulations demonstrate that the proposed framework achieves comparable security to traditional methods while reducing defense costs by up to **15%**. This research establishes a framework that can be later expanded to include a more variable and complex network configuration.

Keywords: Cybersecurity, Moving Target Defense, Cyber Defense, Game Theory, Logic Model

1 Introduction

When protecting large, interconnected networks, system managers face significant risks from advanced adversaries who can bypass defenses and exploit unknown attack strategies. To address this, research has shifted from focusing solely on expected attacks to managing systems in a more cost-effective way. Research moved beyond traditional detect–mitigate–recover cycles toward proactive security that minimizes costs [1].

A key proactive approach is MTD, an active defensive technique rooted in military practice and later adapted to secure digital communication and manage emerging security challenges [2–5]. As an active strategy, MTD reduces dependence on prior adversarial knowledge by focusing on system understanding and vulnerability management, for example by changing IP addresses to increase attacker uncertainty [6, 7]. However, these strategies can be costly, particularly against rational adversaries targeting critical infrastructures with persistent *Denial of Service* (DoS) attacks and information disclosure threats [8–10].

To support cost-aware decision-making in such settings, researchers increasingly employ game theory, which studies strategic interactions among rational agents [11, 12]. In cybersecurity, game-theoretic models typically represent defenders and adversaries as players, enabling the analysis of optimal defense strategies and the prediction of attacker behavior [13, 14]. Depending on the scenario, different frameworks such as zero-sum, Stackelberg, and Bayesian games are used. These frameworks allocate resources, place sensors, distribute security budgets, and study investments in security technologies [15–19].

Within MTD, administrators must decide how to implement defenses ("HOW"), which assets to protect ("WHAT"), and when to apply changes ("WHEN") [20]. This work focuses on optimizing costs associated with the "WHEN" dimension. The main contribution is a game-theoretic optimization framework for dynamically managing multiple interdependent resources, improving cost efficiency while maintaining strong defensive capabilities.

Existing game-theoretic MTD models often assume independent resources and fixed cost. This assumption limits their applicability to real-world systems where resources are interconnected and their criticality varies. These approaches also struggle to scale, as they frequently depend on exhaustive search or oversimplified formulations that sacrifice optimality. This paper introduces the MT2M, which addresses these limitations through the following key contributions:

Highlights

- Transformation of an NP-hard cost optimization problem into a linear formulation;
- Account for resource criticality and node capacity;
- Scenarios that cover all relevant system configurations;
- Identification of key system variables that govern cost-effective defense strategies;
- Validation and comparative analysis via numerical simulations, showing significant cost reductions and security improvements over traditional methods.

The remainder of this paper is organized as follows. Section 2 surveys related work. Section 3 introduces the model along with practical scenarios. Section 4 analyzes optimal strategies for both players, while Section 5 reports numerical results. Finally, Section 6 concludes the paper.

2 Related Work

While MTD provides an alternate methodology in the realm of cybersecurity, it is encumbered by substantial cost issues [21]. To decrease these financial constraints, researchers have explored three distinct strategies. The first involves alterations in the technology employed for MTD and modifications in network configuration. Another approach consists of applying MTD to specific components of the system. Ultimately, and with the easiest applicability, scholars have refined the rate at which MTD is implemented [22].

A noteworthy contribution to the academic literature combining the simplicity and innovation of employing deception is presented in the research conducted by Feng *et al.* [10]. The model synthesizes deception and false messaging with MTD to enhance overall effectiveness. The work features a defender who strategically relocates critical resources and employs deceptive communication to mislead adversaries. Although this study illustrates cost reductions through the use of deception, it is not without limitations. Specifically, it does not accommodate scenarios involving multiple resources, and certain assumptions restrict its applicability in real-world contexts (*e.g.*, the defender's knowledge of attack timing and cost). Finally, the use of deception is subject to strict conditions and primarily offers no significant cost reduction.

Kassem *et al.* [23] take a step beyond Feng *et al.* [10] and consider multiple resources within the network. The authors propose an optimization for implementing MTD, achieving equivalent security results while reducing defense costs. The authors construct this model utilizing Stackelberg's game theory, alongside utility functions that account for the intercorrelation of the costs among the resources. The findings demonstrate improved performance relative to previous studies; however, they do not incorporate the perspective of variable criticalities. The model addresses the problem in a straightforward approach that can serve as a starting point for more complex and advanced work.

An alternative representation portrays the system as a series of states, each characterized by distinct gains and losses. An illustration of this perspective is presented in Shukla *et al.* [24], who propose a model for resource management in networked control systems that utilizes game theory. This model integrates a closed-loop approach with a linear state feedback controller. The model optimizes the defender's cost efficiency while considering the significance of individual nodes within the entire framework. Nonetheless, the model finds nodes as separate entities and fails to address resource allocation within each node.

Artificial intelligence has also garnered broad interest from researchers for optimizing costs in cybersecurity. Charpentier *et al.* [25] apply deep reinforcement learning to design an adaptive MTD system. While their work represents significant progress in the implementation of MTD and other emerging defense technologies, it exhibits notable

limitations at the microlevel of network analysis. Specifically, critical resources within the network are treated as a single unified entity, rather than as distinct, manageable units. At the macro level *Machine Learning* (ML), there are inherent limitations when addressing optimization and network modeling challenges. In particular, ML requires retraining when substantial modifications occur within the system network, rendering prior training obsolete. Moreover, the solution obtained from ML is not guaranteed to be a good solution and requires traditional mathematical modeling [26, 27]. Finally, ML needs a large dataset for training, which is not always available [28].

Optimization of cybersecurity from a broader perspective is studied as a process for selecting defense mechanisms. One of the significant contributions in this area is Temghart *et al.* [29], who propose a modified quantal response model. The model integrates bounded rationality and preferences into the defender's decision-making process. The primary objective is the cost-effectiveness of the defense mechanisms. The model provides a framework for the defender to select from a set of countermeasures, which are then deployed to mitigate potential losses. Although this approach lays a solid foundation for automating cyber defense, it overlooks the strategic dimensions of cybersecurity. Specifically, the model addresses the issue in a unidimensional manner, failing to consider the diversity of resources. We summarize the main findings from the related work in Table 1.

Table 1: Comparison of Optimization Approaches

Approach	**Resource Correlation**	**Variable Constraints**	**Advantages**	**Disadvantages**
Feng *et al.* [10]	No	No	Simplicity and confusion	Restricted Application
Shukla *et al.* [24]	No	Yes	long Term and Customization	Complexity and scalability
Kassem *et al.* [23]	Yes	No	Resource relation	Simplistic
Charpentier *et al.* [25]	No	Yes	High Adaptability	Needs training
Temghart *et al.* [29]	Partial	Yes	Automation of security	Traditional defense
Our approach, MT2M	Yes	Yes	Scalable, Cost-efficient	Static, single adversary

Previous research has demonstrated substantial progress in utilizing MTD and optimizing defense performance. We have found that these contributions can be further enhanced by considering the system resources and the correlations they share. Furthermore, where these contributions introduce complexity into the mathematical model, we aim for a simple solution while maintaining a high level of security. This paper proposes MT2M, which enables the management of multiple critical resources within a network. MT2M is designed to be particularly effective against lateral movement attacks and DoS attacks.

3 Game Model

This section introduces the two-player game model and its main components. Section 3.1 discusses a practical framework and application scenarios for MT2M. Section 3.2 introduces the mathematical framework. Section 3.3 shows how the knowledge of the two agents. Finally, Section 3.4 describes the strategy of the defender. A list of key variables used throughout this paper is presented in Table 2.

Table 2: Table of symbols and meaning as used in the paper.

Symbol	Meaning
π	Adversary strategy
c_m	Defender resource migration cost
n	Number of nodes
m	Number of resources
i	Index for resources
k	Index for nodes
$N_m(i)$	Number of moves or resource i
R	Set of resources
r_i	Critical resource i
$T_c(k)$	Criticality of node k
$E_i(k)$	Expected impact
T_R	Total criticality of set R
$\alpha(i,k)$	Probability of moving resource i to node k
$\mathbb{C}(k)$	Defense cost

3.1 Practical Framework

The practical framework of the system comprises multiple servers that host critical services, including web applications, databases, and API gateways. The network operates on a path-switching framework. This network is inspired by the work introduced by Abdelkhalek *et al.* [30]. In the Abdelkhalek *et al.* [30] network, critical resources are protected by using multiple paths. The system uses path switching to mitigate targeted DoS attacks. We present an example of this network in Figure 1. The setup consists of three servers connected to the network through four paths. The network is connected to a client who is using the network services. The setup also includes an adversary who is targeting one of the four paths.

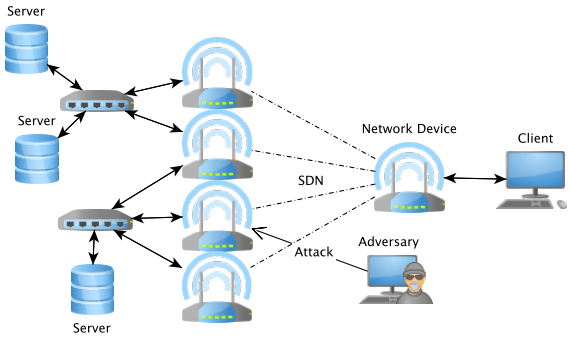


Fig. 1: An example of network setups that use path switching and OpenFlow protocols.

We move to the presentation of attacks and calculating related metrics. Attacks can vary from DoS attacks, which aim to exhaust system resources, to targeted breaches, where attackers seek to exfiltrate sensitive data. Real-world examples of the impact of such attacks include a 12-hour interruption at the Apple Store, resulting in a financial impact of $25 million [31]. Another example is found in the 5-hour outage at Delta Airlines, which led to 2,000 canceled flights and an economic loss of $150 million. While these incidents are extreme examples involving large corporations, small businesses are also at significant risk, with downtime costing an average of $9,000 per minute [32].

In MT2M the impact of an attack is determined by the criticality of the affected resources, which is influenced by both the number of users served and the nature of the data managed. Data collection for estimating resource criticality and interdependencies can be achieved through return-on-investment (ROI) analysis [33]. The ROI metric provides an accurate and cost-oriented representation of the importance of network entities. The calculation of resource criticality is outside the scope of this paper. It is assumed to be predetermined (this value could be cost-oriented, using established metrics such as those presented by Mootzek *et al.* [34] and Gonzalez *et al.* [35]).

Though MT2M is inspired by the work of Abdelkhalek *et al.* [30], it can also be applied to other setups. MTD can be applied using Container Management Systems, such as Kubernetes[1]. The following examples offer alternative implementation scenarios for MT2M.

- Dynamic Service Relocation: Critical services, such as authentication, payment, or database services, are relocated between containers on different nodes. Routing is managed through policies using load balancers (*e.g.*, Kubernetes Ingress controllers[2]) and service meshes (*e.g.*, Istio[3]), which control traffic flow and dynamically adjust routing rules [37].
- Dynamic load redistribution: Critical system components are periodically redistributed between nodes. Multizone clusters with autoscaling enabled can leverage dynamic policies to move workloads based on resource costs (*e.g.*, using Kubernetes Node Affinity) [38].
- Ephemeral Services with Autogenerated Secrets: Critical services are periodically terminated and recreated with new instances to enhance security. For example, token generation services can be regenerated at regular intervals, making it harder for attackers to predict or reuse service endpoints. This functionality can be implemented using Kubernetes Jobs or CronJobs.
- Network Path Randomization: Communication paths between critical components (*e.g.*, front-end, API gateway, back-end services) are randomized. Service meshes (*e.g.*, Istio) control traffic flow, using dynamic path rewrites or traffic shifting [39].
- Adaptive Scaling with Moving Target Logic: Dynamically adjust resources, with services moving between nodes. Resources are periodically shuffled across nodes(*e.g.*, using the Kubernetes Horizontal Pod Autoscaler) [40].

[1] An open-source platform designed to automate the deployment, scaling, and management of containerized applications [36]
[2] A Kubernetes component that manages external access to cluster services.
[3] An infrastructure layer that manages communication between microservices.

- Serverless-Based Critical Function Movement: Critical functions are moved to serverless environments (*e.g.*, AWS Lambda, Google Cloud Functions), where their execution context is ephemeral and distributed. Non-persistent critical functions (*e.g.*, data processing or user verification) are implemented as serverless services [41].

3.2 Defining the Model

To streamline the optimization problem, we represent the practical network discussed in Section 3.1 as a mathematical model. MT2M consists of n nodes and m resources, where $n > 1$ and $m > 1$. In this context, resources represent servers, while nodes represent paths within the network.

The actions of the defender and adversary are based on utility and costs. The costs incurred by the defender agent include two components: the defense cost and the impact of the attack. The defense cost refers to the expenses associated with implementing the defense strategy, as detailed in Section 3.4. The attack impact reflects the expected damage that the adversary is likely to cause during an attack.

Three variations of MT2M are introduced to examine the constraints and flexibility of the defender's strategy:

1. No Criticality: All resources have equal criticality, serving as the base for the model.
2. Multiple Targets: Resources vary in criticality within the network.
3. Single Target: Nodes are limited to hosting only one resource at a time.

Adversarial Attack Path The adversarial attack plan assumes that the adversary is persistent, adaptive, and resourceful, operating under conditions of uncertainty and ambiguity. The attack process typically follows a structured approach:

1. Reconnaissance: Gather intelligence about system architecture, network topology, and potential weaknesses. Reconnaissance is done through passive monitoring, active scanning, or social engineering. This process includes identifying exposed services, outdated software, and user behaviors that can be exploited.
2. Vulnerability Identification: Analyze system weaknesses such as unpatched software, misconfigurations, or zero-day vulnerabilities.
3. Exploitation: Execute attacks through various methods:
   - DoS and *Distributed Denial of Service* (DDoS) Attacks: Overwhelm critical system components, causing resource exhaustion and service unavailability.
   - Zero-Day Exploits: Target unknown or unpatched vulnerabilities, allowing adversaries to bypass security mechanisms undetected.
   - Insider Threats: Involve trusted users (intentionally or unintentionally) weakening system security, leading to privilege escalation or unauthorized data access.
   - Ransomware Attacks: Encrypt critical data and demand payment in exchange for decryption keys.
   - Misconfiguration Exploits: Take advantage of weak security settings to gain unauthorized access.

3.3 Assumptions

As described in the previous section, both adversaries and defenders have a set of prior knowledge on which to base their strategy.

Adversary's Knowledge and Limitations

Known

- The total number of resources and nodes.
- The overarching strategy utilized by the defender.
- The probabilistic distribution of resource positions.
- The overall criticality of the system.

Unknown

- The precise locations of the resources.
- The criticality and value of each resource.

Since adversaries lack precise information about the location and criticality of resources, they rely on probabilistic estimations based on past observations. By scanning the system, they can predict resource allocation and movement. When assessing criticality, adversaries typically adopt a conservative approach, treating all resources as equal and assigning each one a criticality value that reflects the system-wide average.

Adversaries often rely on publicly available external observations and information [42]. They can analyze the services a system offers and estimate user usage, but do not have access to internal configurations [43], resource interdependencies, or specific operational details. This uncertainty is heightened in environments such as *Software Defined Networking* (SDN) [44]. Although adversaries can discreetly gather configuration information [45], it usually pertains to network setups rather than the criticality of specific resources. This limits their ability to accurately assess the importance of individual resources.

Research has shown that adversaries use attack graphs to visualize potential attack paths [46]. However, without in-depth knowledge of the system's internal structure, these graphs only provide a high-level view. While they can help estimate the overall criticality of the system, they don't offer insights into the criticality of individual components. Additionally, studies suggest that attackers may display cognitive biases, often concentrating on prominent or well-known services [47]. This focus leads them to make assumptions about system criticality based on observable factors, rather than the actual internal importance of resources.

Defender's Knowledge and Limitations

Known

- The strategy used by the adversary to select a target.
- Complete knowledge of the current system state, including resource positions and criticalities.

Unknown

- The exact timing of the attack.

3.4 Logic and Methodology

The defender's strategy of resource movement is built upon the knowledge of the adversary. This knowledge allows the adversary to anticipate the defender's actions and devise a corresponding attack plan. This interaction is modeled as a two-player game, specifically a Bayesian Stackelberg game [48]. This classical approach to game theory seeks to establish realistic strategies for both players. In the game, the leader (defender) makes decisions while anticipating the follower's (adversary) response. The leader takes into account the uncertainty in the follower's preferences and adjusts the strategy accordingly.

The resources, denoted as r_i, belong to the set $R = \{r_1, r_2, \ldots, r_m\}$. Each resource has an associated criticality, denoted by $R_c = \{r_c(1), r_c(2), \ldots, r_c(m)\}$ where R_c is in descending order, and total criticality $T_R = \sum_{i=1}^{m} r_c(i)$. The resources are distributed across a total of n nodes. In the single-target scenario, each node is limited to containing only one resource. In contrast, other scenarios allow nodes to hold multiple resources without any limitations.

The resource locational probabilities are represented with a matrix A, where $\alpha(i,k)$ represents the probability that resource r_i is located in node k. The expected impact of the attack is the dot product of the column k of the attacked node and the resource criticality vector, *i.e.*, $E_i(k) = \sum_{i=1}^{m} \alpha(i,k) \cdot r_c(i)$. An example A is shown below:

$$A = \begin{pmatrix} \alpha(1,1) & \alpha(2,1) & \ldots & \alpha(m,1) \\ \alpha(1,2) & \alpha(2,2) & \ldots & \alpha(m,2) \\ \vdots & \vdots & \ddots & \vdots \\ \alpha(1,n) & \alpha(2,n) & \ldots & \alpha(m,n) \end{pmatrix}$$

Let $\mathbb{C}(k)$ denote the expected cost incurred by the defender when executing this defense strategy. This cost comprises two primary components:

- Expected Impact of the Attack ($E_i(k)$).
- Total Cost of Defense: This includes the expenses associated with implementing the defense strategy. Several factors influence the cost of defense:
  - Movement Cost (c_m): Determined by the network configuration and r_c.
  - Frequency and Method of Movement.

To differentiate between resources with identical movement frequencies, we consider the waiting time before each resource moves again. This takes into account the increased predictability and potential vulnerability associated with consistent movement patterns. Accordingly, the defender's cost function is defined in Equation 1.

$$\mathbb{C}(k) = E_i(k) + c_m \sum_{i=1}^{m} \left(r_c(i) \cdot N_m(i) \cdot (1 - \max \alpha(i))^2 \right) \tag{1}$$

The term $\max \alpha(i)$ represents the maximum probability for resource i, *i.e.*, the highest value in column i of A. The use of $1 - \max \alpha(i)$ to highlight the correlation between resources.

4 Defense Strategy

This section outlines the preferred strategy of the defender. First, in Section 4.1, two examples show the primary methodologies to develop the final strategy. Section 4.2 identifies potential scenarios in which to apply the preferred strategy. Following this, Section 4.3, these methodologies are used to determine the general solution.

4.1 Strategy's Insights and Proof of Algorithms

The defender minimizes the total cost of defense through two primary optimization strategies. The first strategy focuses on reducing the probability of moving nodes by lowering α. The second strategy aims to minimize criticality across all nodes. This optimization challenge can be viewed as a variation of the subset sum problem [49], where a set of numbers is divided into subsets that add up to a specific target value.

The defender can allocate resources across multiple nodes, aiming for an even distribution of resources among them. However, transferring resources between a greater number of nodes incurs additional costs. Moreover, due to correlations among resources, it's preferable to distribute movements across multiple resources rather than concentrating changes on a single one. This scenario is similar to the classic NO-hard Bin Packing Problem [50]. The Bin Packing Problem contains items of varying sizes that must be packed into a finite number of bins without exceeding their capacities. The added constraints accounting for their correlations introduce additional complexity. These factors align the problem more closely with the Multi-Container Packing Problem [51], which is also known to be strongly NP-hard.

To solve the problem and optimize costs, we form a solution tailored to the model at hand. Initially, the defender assumes a setup where resource criticalities are uniformly distributed across all nodes. The NP-hard problem of optimizing costs is simplified to a linear problem that can be easily solved. The solution is obtained through the optimization steps.

Minimizing Movement Cost The defender focuses on minimizing the cost of defense while maintaining the same $T_c(k)$. In the context of the bin packing problem, minimizing movement cost can be seen as placing items in as few bins as possible. To achieve this, resources are redistributed among nodes, following an approach analogous to the resource allocation problem proposed in [52]. The primary objective is to increase the maximum value α within each column of the resource matrix. For this, instead of dividing a resource across multiple nodes, the defender consolidates it within a single node. By iteratively applying this process to different pairs of resource nodes (i, k), the defender reduces the frequency of resource movement while maintaining the same maximum $T_c(k)$.

Minimizing Movement Cost with Resource Correlation After minimizing the movement cost, the defender focuses on reducing the cost associated with the correlation of resources $((1 - \alpha)^2)$. In this phase, the movement cost is distributed more evenly across multiple resources, rather than being concentrated on a single resource. The defender averages the total criticality T_R for each node on all non-zero αs. Although this approach may lead to a marginal increase in movement cost, it ultimately results in a lower overall cost.

4.2 Application Scenarios

The defender builds on the insights from Section 4.1 and applies them in different scenarios. These scenarios are used to derive the mathematical model for the defender algorithm in Section 4.3.

Constant Scenario: Constant Criticality Multiple Target Attack In this case, the defender faces minimal constraints, thereby possessing a wide range of maneuverability. The resources are uniformly valued in terms of criticality, normalized to a value of 1. It is assumed to have fewer nodes than resources to represent a realistic situation better. Although this configuration is not realistic, it serves as a foundational starting point for developing the primary methodology.

Variable Scenario: Variable Criticality Multiple Target Attack This scenario builds on the previous one, with the additional complexity of varying criticality for resources. One of the most likely cybersecurity threats in this context is DoS attack, which can target multiple resources simultaneously. A notable example is the 2015 BlackEnergy attack in Ukraine, which resulted in a system-wide power grid outage [53].

Single Scenario: Variable Criticality Single Target Attack In this scenario, there is a capacity limitation for each node. Specifically, the sum of probabilities for each node cannot exceed one ($\sum_{i=1}^{m} \alpha(i,k) \leq 1$). As a result, the number of resources cannot exceed the number of nodes. In a single target scenario, the adversary targets a specific system resource, such as through phishing or man-in-the-middle attacks [54].

4.3 Algorithm of the Defender

The defender uses the steps introduced in Section 4.1 to build a general strategy tailored to the specific scenario in Section 4.2 while adhering to the constraints defined in Section 3.2.

The defense strategy of the defender is presented by defining the movement of resources between nodes. To streamline this strategy, the following steps are taken to form the final approach.

- Minimize $E_i(k)$ by minimizing criticality across all nodes ($T_c(k)$). Since the maximum cannot be lower than the average, it follows that the minimum is the average *i.e.*, $\frac{T_R}{n}$. The defender begins by balancing the criticality distribution of resources across nodes to retain a minimum $E_i(k)$.
- Minimize the cost of movement, while maintaining $T_c(k)$.
- Minimize the correlation cost of movement $(1 - \max \alpha(i))^2$ shown in Equation 1.

4.3.1 Constant Scenario

To find the optimum strategy, the defender repeats the optimization steps to form the solution presented in Algorithm 1. The solution is divided into two parts that serve two main functionalities. The first is presented in Algorithm 1, where the allocation of resources is defined. This part functions in the following steps.

1. Initialize the variables and initialize A.
2. Set $\alpha = \frac{m}{n\lceil \frac{m}{n} \rceil}$ for resources in nodes that do not have stationary resources.
3. Use a loop that sets α for the stationary resources.

Algorithm 1 Strategy of the defender in the no criticality scenario

$L \leftarrow \lceil \frac{m}{n} \rceil \mathbb{M}_n^m$ ▷ Number of moving resources
$S \leftarrow \frac{m-L}{n-\mathbb{M}_n^m}$ ▷ Number of stationary resources per node
for $k = 1 \rightarrow \mathbb{M}_n^m$ **do** ▷ nodes without stationary resources
  for $i = 1 \rightarrow L$ **do** ▷ define the non-zero values
    if $i > (k-1)\lceil \frac{m}{n} \rceil$ & $i \leq k\lceil \frac{m}{n} \rceil$ **then**
      $\alpha(i,k) \leftarrow m \cdot \frac{1}{n\lceil \frac{m}{n} \rceil}$
for $k = 1 \rightarrow n - \mathbb{M}_n^m$ **do** ▷ nodes with stationary resources
  for $i = 1 \rightarrow m - L$ **do** ▷ define the non-zero values
    if $i > (k-1) \cdot S$ & $i \leq k \cdot S$ **then**
      $\alpha(i + L, k + \mathbb{M}_n^m) \leftarrow 1$
$matrix(\mathbb{M}_n^m + 1 : n, 1 : L) = \frac{1}{n\lceil \frac{m}{n} \rceil} \cdot formM(n - \mathbb{M}_n^m, L)$

Due to the methodology presented in Section 4.1, the obtained solution produces an α matrix with only diagonal values. To address this characteristic of the solution, we present Algorithm 2, which iteratively divides a given matrix into smaller square matrices that are set diagonally. The diagonal value is equal to the size of the smaller matrix.

Algorithm 2 Recursive α function

procedure $formM(m, n)$
  $M \leftarrow zeros(m, n)$ ▷ start with zero m, n matrix
  if $m = 1$ *or* $n = 1$ **then** ▷ stop if n or m is 1
    return ones(m,n) ▷ return an mxn matrix of ones
  else
    $Minimum \leftarrow m$ ▷ start with mxm matrix
    if $m > n$ **then** ▷ test if $m > n$
      $Minimum \leftarrow n$ ▷ change to nxn matrix
      $M(n + 1 : m, 1 : n) = formM(m - n, n)$
    else
      $M(1 : m, n + 1 : n) = formM(m, n - m)$
    for $i = 1$ to $Minimum$ **do**
      $M(i, i) \leftarrow Minimum$ ▷ Fill diagonally

4.3.2 Variable Scenario

The variable builds on the Constant scenario and establishes variable criticality. Although higher criticality may be seen as equivalent to an increased number of resources m, the costs associated with reallocating resources can be further optimized.

Our proposed solution is introduced in Algorithm 3. Starting from the constant scenario solution, the defender cost is further optimized through resource substitution between nodes. The steps of the algorithm are as follows:

1. Calculate the number of nodes to allocate the resource, ending at node k_l.
2. Iteratively set the α value for r_i is set until k_l.
3. Each node is filled before proceeding to the next, where a total of α_i.
4. The algorithm does not move to the next node until α_i is assigned.

Algorithm 3 strategy of the defender in the multiple target scenario

$k \leftarrow 1$ ▷ index for nodes
$\alpha_k \leftarrow 1$ ▷ remaining α to set
for $i = 1 \rightarrow m$ **do**
    $\alpha_i \leftarrow n \times \frac{r_c(i)}{T_R}$
    while $\alpha_i \geq \alpha_k$ & $k < n$ **do** ▷ exhaust α_i
        $\alpha(i,k) \leftarrow \alpha_k \times \frac{T_R}{n \times r_c(i)}$
        $\alpha_i \leftarrow \alpha_i - \alpha_k$ ▷ update remaining α
        $k \leftarrow k + 1$ ▷ move to the next node
        $\alpha_k \leftarrow 1$ ▷ reset remaining α for node k
    $\alpha(i,k) = \alpha_i \times \frac{T_R}{n * r_c(i)}$ ▷ set last α for node i
    $\alpha_k = \alpha_k - \alpha_i$ ▷ subtract alpha from current node

4.3.3 Single Scenario

Due to the constraint of a single-target attack, the defender is primarily restricted by the maximum capacity of each node. The multiple target strategy previously described in 3 can be repeated only if $n \geq \lceil \frac{T_R}{r_c(m)} \rceil$, where $r_c(m)$ is the smallest criticality in the system. Otherwise, the strategy would not maintain the condition $\sum_{i=1}^{m} \alpha(i,k) \leq 1$; an alternative strategy is presented in the following section.

To facilitate resource management, resources are categorized into two groups according to their criticality, using the condition $r_c(i) < \frac{T_R}{n}$. This categorization results in two groups: R_H and R_l, where the condition is fulfilled for R_l. The sizes of the sets R_H and R_l are indicated by $|R_H|$ and $|R_l|$, respectively. The proposed strategy is starts with the resources that are more difficult to place, namely R_l. To address the hard limit of $\sum_{i=1}^{m} \alpha(i,k) \leq 1$, the defender's strategy is presented sequentially on the nodes by iterating through resources within each node. The strategy is presented in two iterations. In Algorithm 4, all resources in R_l are treated as a single resource with a criticality equal to the average of all criticalities in R_l, and similarly for R_H.

1. Place the maximum amount of resources R_l. The node capacity limits are respected, and the total criticality per node is minimized.
2. Iterates through the resources in R_l.
3. Fill the remaining space in nodes using resources fromR_H.

Algorithm 4 First step of the single target defender strategy

1: $Average_H \leftarrow \frac{\sum_{r=1}^{|R_H|} R_H(i)}{|R_H|}$ ▷ average criticality of R_H
2: $Average_l \leftarrow \frac{\sum_{r=1}^{|R_l|} R_l(i)}{|R_l|}$ ▷ average criticality of R_l
3: **if** $Average_l = 0$ **then** ▷ test if low set is empty
4: $k \leftarrow 0$ ▷ index for nodes
5: **else**
6: $\alpha_H \leftarrow \frac{T_R - n \cdot Average_l}{n(Average_H - Average_l)}$ ▷ α for R_H
7: **for** $k = 1 \rightarrow \frac{|Rl|}{1-\alpha_H}$ **do**
8: **for** $i = 1 \rightarrow |R_H|$ **do** ▷ loop through R_H
9: $\alpha(i,k) = \frac{\alpha_H}{|R_H|}$
10: **for** $i = |R_H| + 1 \rightarrow m$ **do** ▷ loop through R_l
11: $\alpha(i,k) = \frac{1-\alpha_H}{|R_l|}$
12: **if** $k > n$ **then return** ▷ return if k exceeds n
13: **for** $i = 1 \rightarrow |R_H|$ **do** ▷ loop through R_H
14: $\alpha(i,k) = \frac{\frac{T_R}{n} - Average_l \cdot |R_l|(1 \mod \frac{1-\alpha_H}{|R_l|})}{|R_H| \cdot Average_H}$
15: **for** $i = |R_H| + 1 \rightarrow m$ **do** ▷ loop through R_l
16: $\alpha(i,k) = 1 \mod \frac{1-\alpha_H}{|R_l|}$ ▷ remaining α

Algorithm 5 serves as a continuation of Algorithm 4. Algorithm 4 handles the resources characterized by low criticality R_l. After managing most of the low criticality set R_l, the algorithm 5 manages any residual resources as indicated in R_l, along with those of higher criticality as defined in R_H, operating similarly to Algorithm 3. The algorithm starts by iterating through nodes. We define the remaining α for the current resource as α_R. A similar remaining criticality per node, C_k, is used for all nodes. When α_R and C_k reach zero, the algorithm moves to the following resource and node, respectively. In each step, the algorithm tries to exhaust either α_R or C_k, depending on which is higher.

5 Performance Evaluation

To evaluate the performance of MT2M, we start by presenting a use case example in Section 5.1. Section 5.2 describes the experiment setup. We study the performance of the variable scenario where nodes can host multiple resources in Section 5.3. Finally, we explore the single scenario where nodes can host a maximum of one resource in Section 5.4.

5.1 Use Case Example

We start by giving a representation of the network setup. We consider a network topology consisting of 20 hosts, which represent resources, and 6 distinct communication paths, representing nodes. The system is created in the Mininet simulator. This setup

Algorithm 5 Second step of the single target defender strategy

$i \leftarrow 1$ ▷ start from first node in R_H
$\alpha_R \leftarrow 1 - \sum_{j=1}^{k} \alpha(i,j)$ ▷ remaining α for resource i
$C_k \leftarrow \frac{T_R}{n}$ ▷ remaining criticality in k
while $k \leq n \ \ \& \ \ i \leq |R_H|$ **do**
    while $\alpha_R \leq 0 \ \ \& \ \ i < |R_H|$ **do**
        $i \leftarrow i + 1$
        $\alpha_R \leftarrow 1 - \sum_{j=1}^{k} \alpha(i,j)$
    if $\alpha_R \geq \frac{C_k}{r_c(i)}$ **then**
        $\alpha(i,k) = \frac{C_k}{r_c(i)}$
        $\alpha_R \leftarrow \alpha_R - \alpha(i,k)$ ▷ remaining α for i
        $k \leftarrow k + 1$ ▷ move to the next node
        $C_k \leftarrow \frac{T_R}{n}$ ▷ remaining criticality for k
    else
        $\alpha(i,k) = \alpha_R$
        $C_k \leftarrow C_k - \alpha_R \cdot r_c(i)$ ▷ remaining criticality in k
        $i \leftarrow i + 1$ ▷ move to the next resource
        if $i \leq |R_H|$ **then**
            $\alpha_R \leftarrow 1 - \sum_{j=1}^{k} \alpha(i,j)$ ▷ remaining α for r_i

allows us to simulate and analyze the effectiveness of path diversification and switching policies in mitigating potential reconnaissance and attack vectors. Figure 2 shows the example network.

5.2 Experiment Setup

Evaluation is done through multiple scenarios where we vary system variables such as m, n and standard deviation of r_c. The standard deviation is calculated as $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(r_i - \frac{T_R}{n})^2}$. The experiment is run in Matlab where it is repeated 100 times and the cost values are averaged over the 100 iterations.

In addition to MT2M, a baseline is established that maximizes defense, where the defender continuously moves resources according to the work of Feng *et al.* [10]. The baseline achieves the maximum possible security from MTD, but at the expense of maximizing defense cost. A third model is based on the work of Kassem *et al.* [23], which does not consider variable criticality. The following analyses detail how MT2M performs in comparison to these established benchmarks, highlighting strengths and weaknesses in various aspects. The test experiments are based on the scenarios described in Section 4.2. The constant scenario, which serves as a prerequisite for the variable scenarios, is not included in the experiments.

MT2M introduces the concept of criticality, which is not present in the models by Feng and Kassem. Consequently, for comparison and alignment purposes with the preceding research frameworks, the criticality of a resource, when not defined, is characterized as the average of the overall criticality of the system, *i.e.*, $\frac{T_R}{m}$. The

implementation of our proposed algorithm can be found in a companion GitHub repository [55].

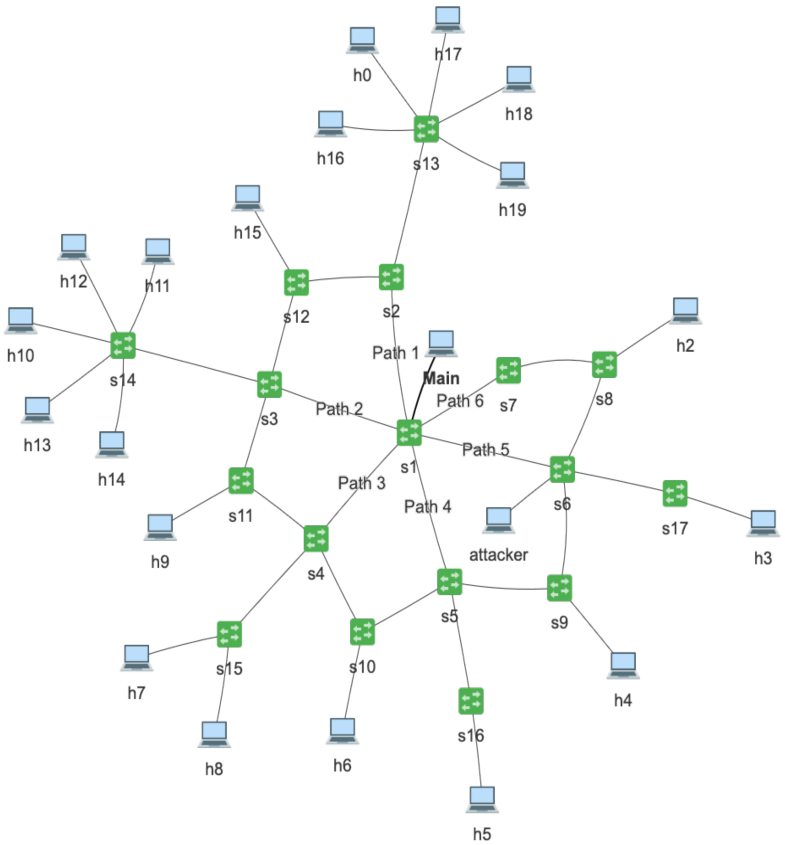


Fig. 2: Network topology used for numerical evaluation. The system consists of 20 resources (hosts) connected via six nodes (paths). An external host attempts to access the system while an adversary observes and interacts with the network. The configuration supports moving the target across different paths leading to the host.

5.3 Variable Scenario

To evaluate MT2M, an experiment is designed that operates analogously to that in Section 3.2. Different experiments consider the variation of a single variable while keeping the others constant. The variables are system-defining and include the number of nodes n, the number of resources m, the total criticality T_R, and the standard deviation of the criticality. To calculate standard deviation we use the following expression $\sqrt{\frac{\sum_{i=1}^{m}(r_c(i)-\frac{T_R}{m})^2}{m}}$.

5.3.1 Variable Scenario Simulation Experiments

We start with the variable criticality scenario from Section 4.2. The simulation is run over four comprehensive experiments by varying the following.

1. Node count n.
2. Resource count m and total criticality T_R.
3. Resource count m, while the total criticality T_R remains constant.
4. Standard deviation of criticality.

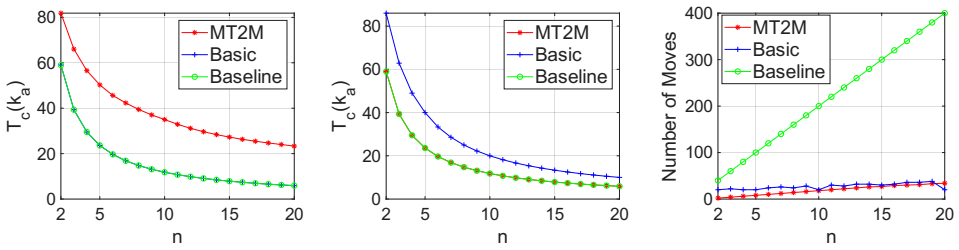
The simulations are performed in MATLAB where the simulations are repeated 100 times. In each of the four experiments, the system's performance is evaluated by calculating and presenting three key metrics.

1. $E_i(k)$as seen by the adversary.
2. $E_i(k)$ as seen by the defender.
3. Number of moves of all the resources.

5.3.2 Variable Scenario Simulation Results

Each experiment includes three figures showing the three key metrics. From the defender's perspective, MT2M and the baseline perform identically in all experiments, as shown in Figures 6(b), 5(b), 3(b), and 4(b). From the adversary's perspective, previous work and the baseline again align, while MT2M lags as shown in Figures 6(a), 5(a), 3(a), and 4(a). This effect arises because, as discussed in Section 3.3, the adversary does not know resource-specific criticalities and instead uses $\frac{T_R}{m}$, which leads to overestimating $E_i(k)$.

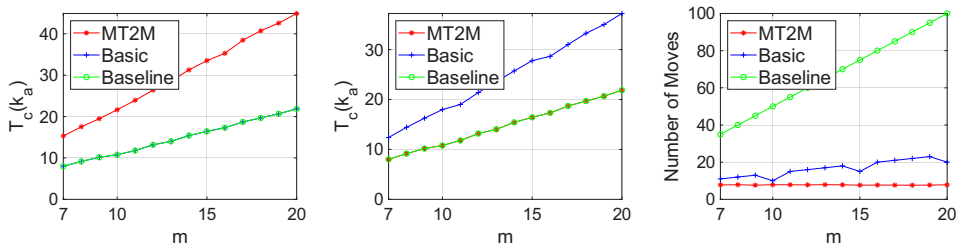
Varying Node Count n: The findings indicate a reduction in overall costs as n increases, as shown in Figure 3. This outcome is predictable due to having more flexibility in moving resources.



(a) $E_i(k)$ of adversary (b) $E_i(k)$ of defender (c) Number of moves of R

Fig. 3: System performance when varying the number of nodes n. The figures include Feng *et al.* [10] work (baseline) and a model where resource criticality is not considered (Previous).

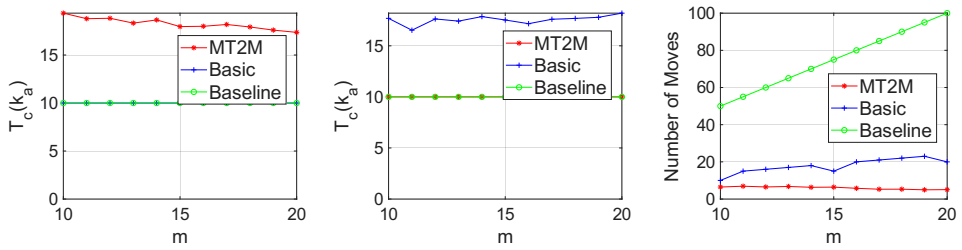
Varying m and T_R: The relationship between m and the cost of MT2M is proportional, as demonstrated in Figures 5 and 4. Although higher overall system criticality may present greater potential benefits for the adversary, it simultaneously leads to a reduction in the average cost per resource. This phenomenon is attributed to the increased maneuverability margin provided to the defender as m increases.



(a) $E_i(k)$ of adversary (b) $E_i(k)$ of defender (c) Number of moves of R

Fig. 4: System performance when varying the number of resources m and total criticality T_R. The figures include Feng *et al.* [10] work (baseline) and a model where resource criticality is not considered (Previous).

Varying m: Having explored the impact of varying R, the third experiment explores that of m. Figures 5(a) and 5(b) illustrate minimal influence on $E_i(k)$. The principal finding is derived from the examination of the number of moves, as shown in Figure 5(c), where MT2M demonstrates an apparent reduction. In contrast, the other two models exhibit an increase in moves.

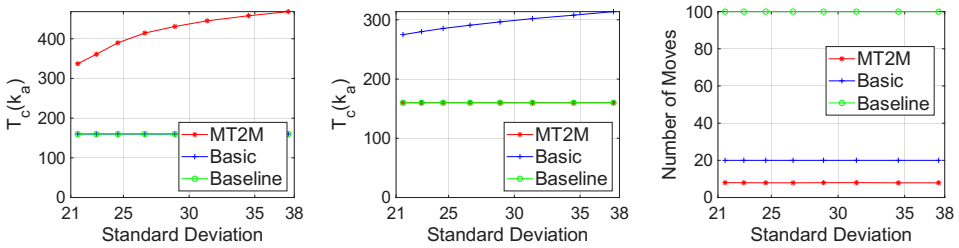


(a) $E_i(k)$ of adversary (b) $E_i(k)$ of defender (c) Number moves of R

Fig. 5: System performance when varying the number of resources m and total criticality T_R remains the same. The figures include Feng *et al.* [10] work (baseline) and a model where resource criticality is not considered (Previous).

Variation Standard Deviation The final experiment examines the impact of altering the standard deviation of r_i. Although the defender cannot directly control the standard deviation or the overall set of resources, this value can be modified by changing system clustering and configuration.

Figure 6 shows that the crucial variable defining MT2M is the standard deviation. Kassem's work [23], which employs the average criticality rather than the actual measure, exhibits progressively deteriorating performance as the standard deviation increases. This decrease is due to the increasing deviation between the actual and average criticality values. Similarly, adversaries experience a similar increase in expected impact within MT2M. The number of moves does not show variation since T_R remains constant.



(a) $E_i(k)$ of adversary (b) $E_i(k)$ of defender (c) Number of moves of R

Fig. 6: System performance when varying criticality standard deviation. The figures include Feng *et al.* [10] work (baseline) and a model where resource criticality is not considered (Previous).

5.3.3 Variable Scenario Discussion

This section analyzes the performance of MT2M in the four key experiments of Section 5.3.2. The goals of this analysis are as follows.

- Key insights: The most significant findings of the simulations are discussed, highlighting how the defender could move forward when setting up a preferred defense strategy.
- The significance of each variable: Each variable is studied separately, and a relationship is established between the variable and the performance of MT2M.

Comparison with Previous Work

The following conclusions can be drawn based on the analysis of the four experiments and their correlation with previous work.

1. $E_i(k)$ from the adversary side: Although MT2M exhibits the highest expected impact for the adversary compared to previous work, the real impact remains minimum (same as baseline). The adversary perceives a high impact, even when the actual impact is lower.
2. Number of Moves: MT2M demonstrates the least number of moves between the three models. MT2M takes advantage of having multiple resources and criticalities to replace defense cost with a higher risk for some resources while proportionally decreasing it for others. Specifically, where the model is studied from a cost perspective, the overall costs remain the same.
3. The parameters m and n have a direct impact on the total expected cost of the defender, and increasing their values proves beneficial to the defender. By modifying the size of each cluster of nodes and resources, the defender can enhance the system to minimize the expected cost per resource.
4. An increase in the standard deviation increases the uncertainty for the adversary, prompting them to anticipate a greater potential $E_i(k)$. Therefore, it is advantageous for the defender to organize resources with high and low criticalities in combination while clustering resources of medium criticalities independently.

Comparison with Existing Stackelberg-Based MTD Models

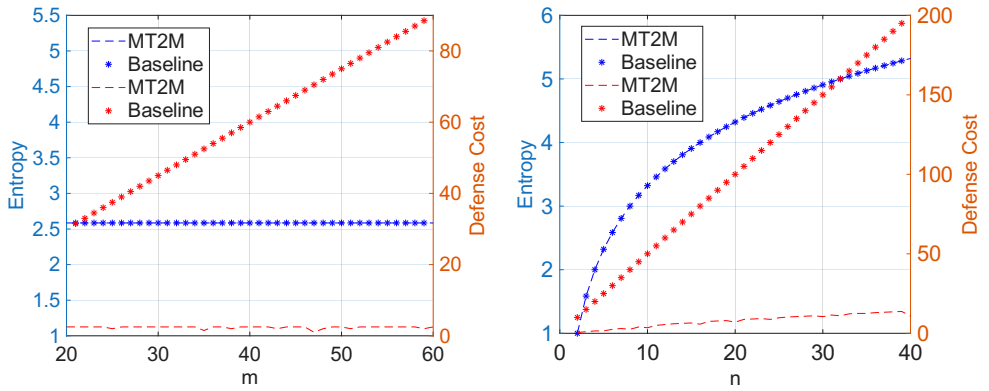
To contextualize the effectiveness of MT2M, we compare it with two established works: Feng *et al.* [10] and Kassem *et al.* [23], which utilize Stackelberg game theory for MTD strategies. Comparison is done by using entropy since it offers a solid evaluation of MTD. Entropy measures the unpredictability of system states. Higher entropy indicates greater variability in resource placement, making the system harder for adversaries to predict and target. The defenders' entropy is defined in Equation 2.

$$H_{\text{def}} = -\sum_{k=1}^{n} \left(\frac{1}{T_R} \sum_{i=1}^{m} r_i \cdot \alpha(i,k) \right) \log \left(\frac{1}{T_R} \sum_{i=1}^{m} r_i \cdot \alpha(i,k) \right) \tag{2}$$

To study entropy, we introduce two scenarios with $c_m = \frac{1}{4}$. In the first setup, we vary $n = 2 \rightarrow 40$ with $m = 20$. In the second setup, we vary $m = 20 \rightarrow 60$ with $n = 6$. In both setups, we study the total cost of defense, which is the number of moves multiplied by c_m, and the entropy. The results are shown in Figure 7. In the figure, blue plots are associated with entropy, and red plots are associated with costs. We use a baseline similar to the one in 5.3.2, where resources are equally divided among all nodes. The baseline offers the highest possible entropy, offering a good comparison point for MT2M.

The two figures show that MT2M matches the baseline's performance but at a lower cost. In Figure 7(a), increasing m does not change entropy, since entropy depends on n, but it does increase defense cost without any benefit. In contrast, Figure 7(b) shows that increasing n raises both entropy and defense cost, although entropy grows quickly at first and then more slowly once n exceeds 20. The defense cost increases approximately linearly, with MT2M exhibiting a much smaller slope than the baseline.

This reduction arises from the design of MT2M, which distributes MTD actions across resources to minimize total cost.



(a) Entropy varying m and $n = 20$.

(b) Entropy varying n and $m = 20$.

Fig. 7: A graph comparing entropy with cost, showing how increasing entropy also increases the defense cost.

Table 3 presents a theoretical and partial empirical comparison of key outcomes. These values are obtained using analogous system setups and translated payoff matrices when possible. Due to differences in utility formulation, comparisons are approximate but provide insights into the comparative benefits of MT2M. Due to the complexity of finding a numerical expression for the number of moves in the model presented in [23], we represent it using the following expression.

$$N_m = \begin{cases} m \times n & \text{if } m < 2 \text{ or } n < 2 \\ n + f(m-n, n) & \text{if } m \geq 2 \text{ and } n \geq 2 \text{ and } m > n \\ m + f(m, n-m) & \text{otherwise} \end{cases} \tag{3}$$

Table 3: Comparison with Existing Stackelberg-Based MTD Models

Model	**Strategy Type**	**MTD Cost (Defender)**
Feng *et al.* [10]	Static probability	$c_m \times m \times n$
Kassem *et al.* [23]	Dynamic Intervals	N_m
Our approach, MT2M	Dynamic intervals	$c_m(5.6 \times log(2n) + n - 7.8)$

While the work in [10] demonstrates a slightly higher entropy and uncertainty, higher defense costs accompany this increase. In MT2M, the primary focus is on

reducing costs while maintaining the expected impact of attacks. To assess the holistic performance of MT2M, we evaluate it along three critical dimensions: computational efficiency, cost optimization, and security enhancement.

- Computational Efficiency: MT2M calculation time complexity is equal to $O(m*n)$, showing minor overhead due to the increased size of A matrix. In comparison, static strategies complete faster but lack adaptability.
- Cost Optimization: Using our utility function, the defender's cumulative cost was reduced by 15% compared to static Stackelberg implementations. The matrix format enables the selection of optimal MTD intervals based on expected attacker response, resulting in more cost-effective defense cycles.
- Security Enhancement: MT2M retains the same expected cost as the always-on time-based MTD.

5.4 Single Scenario

In the case of a single scenario, MT2M's performance is illustrated through a numerical example. Consider a configuration with 6 nodes and 4 resources with criticalities $R = 6, 5, 4, 2$. This example highlights how the adversary perceives the system before launching an attack. After applying the strategy presented in this paper, the defender arrives at the following position matrix:

$$R = \begin{pmatrix} 6 \\ 5 \\ 4 \\ 2 \end{pmatrix} A = \begin{pmatrix} \frac{5}{54} & 0 & 0 & 0 & \frac{47}{108} & \frac{17}{36} \\ \frac{5}{54} & 0 & \frac{8}{27} & \frac{17}{30} & \frac{2}{45} & 0 \\ \frac{5}{54} & \frac{41}{72} & \frac{73}{216} & 0 & 0 & 0 \\ \frac{13}{18} & \frac{5}{18} & 0 & 0 & 0 & 0 \end{pmatrix} \tag{4}$$

5.4.1 Defender's Expected Impact

The defender has a clear understanding of the system. Based on this understanding, the defender can calculate $T_c(k)$ by multiplying A with the individual criticality values R. This provides a more accurate picture of potential losses. Taking the first node as an example, the expected loss would be $6 \times \frac{5}{54} + 5 \times \frac{5}{54} + 4 \times \frac{5}{54} + 2 \times \frac{13}{18} = \frac{17}{6}$. Where $\frac{17}{6}$ is the average T_R over all nodes. By applying this method to all nodes, the defender obtains the average expected attack impacts for all nodes.

5.4.2 Adversary's Expected Impact

The adversary, on the other hand, lacks detailed knowledge covering the actual impact of the attack. They replace the individual criticality of each resource with the average criticality of all resources. This leads to a set of resources $R = \frac{17}{4}, \frac{17}{4}, \frac{17}{4}, \frac{17}{4}$. For instance, the adversary calculates the expected impact for the first node as $\frac{17}{4} \times \frac{5}{54} + \frac{17}{4} \times \frac{5}{54} + \frac{17}{4} \times \frac{5}{54} + \frac{17}{4} \times \frac{13}{18} = \frac{17}{4}$. Repeating the same approach on all nodes produces the expected attack impact as $\{\frac{17}{4}, \frac{1037}{288}, \frac{2329}{864}, \frac{289}{120}, \frac{2175}{1067}, \frac{289}{144}\}$.

5.4.3 Exploiting the Misinformation

By understanding the limitations of the adversary, the defender can use this to their advantage. The defender knows that specific nodes will appear more attractive to the adversary because the adversary relies on the average criticality of the nodes. However, the defender also knows that these nodes have low criticalities. This creates a dilemma for the adversary.

- Attack the node with the highest $T_c(k)$ but with lower criticality resources.
- Attack the node with the highest r_i, but lower $T_c(k)$ and attack success rate.

5.5 Model complexity

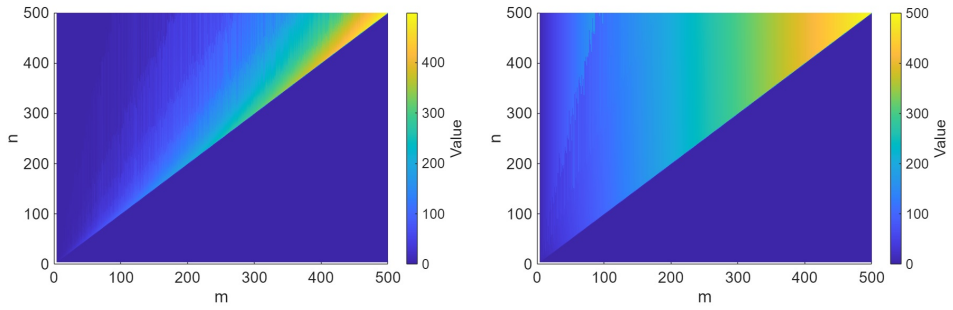
In this section, we study the complexity of MT2M, starting with the multiple target scenario, followed by the single one.

5.5.1 Variable Scenario

The complexity of this scenario is shown in Algorithm 3. The solution is composed of a nested loop. The parent loop runs m times while the child runs n times. The two loop counts sum to $T(m, n) = m \times n$. The overall complexity is $\Theta(mn)$.

5.5.2 Single Scenario

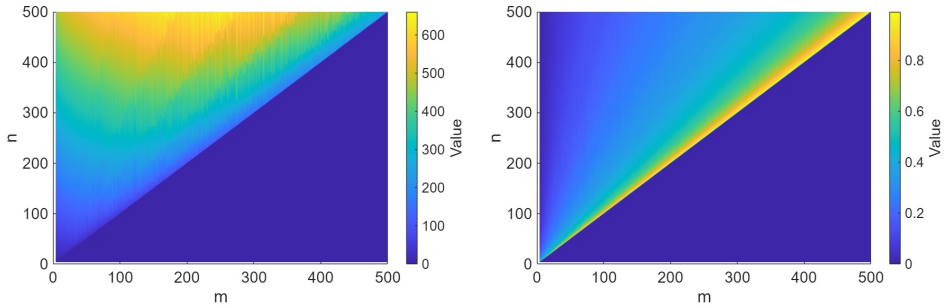
The nested loop structure of the optimization algorithm primarily drives the complexity of the single scenario model. The nested loops are shown in Algorithms 4 and 5. Algorithms 4 shows loops one and two, where loop two is nested in loop one. Algorithms 4 shows loops three and four, where loop four is nested in loop three. The number of iterations for each nested set is visualized in figures 8 and 9. Note that in the figure, the single target is not defined when $m > n$. Due to the single target limitation, all those dark blue zones where n is below m are set to zero.



(a) Loop 1 (b) Loop 2

Fig. 8: Number of iterations as m and n are varied between 4 and 500 of Algorithm 4.

Algorithm 4 complexity (Figure 8): The first two plots of Figure 8 illustrate how the number of iterations of loops one and two scales with m and n. The figure shows a higher dependence on m, especially in loop 2. In the second loop, n shows a negligible impact on the number of iterations. The total complexity for the parent and child loops is roughly equal to m. This dependence on m is because Algorithm 4 is mainly concerned with allocating resources with low criticality. The total complexity of the two loops is m^2.



(a) Loop 1 (b) Loop 2

Fig. 9: Number of iterations as m and n are varied between 4 and 500 of Algorithm 5.

Algorithm 5 complexity (Figure 9): Figure 9(b) shows that the child loop has a higher number of iterations when m and n values are closer. The number of iterations for the child loop we found to be $\frac{m}{n}$. Since m cannot exceed n, the total number of iterations for the child loop does not exceed one. The higher number of iterations and the main impact on complexity are seen in the parent loop. Figure 9(a) shows a parabolic variation of iterations as m and n vary. Based on the experiment, we found the number of iterations to be $\frac{5m(n-m)}{n}$. Algorithm 5 handles the resources over the remaining nodes, making it dependent on both m and n. The total complexity of the two loops is approximately $\frac{5m^2(n-m)}{n^2}$.

The single scenario complexity is the sum of the two algorithms $T(m,n) = \frac{5m^2(n-m)}{n^2} + m^2$. Since loops are independent, calculations can be parallelized and rapidly calculated through modern processing units. In large networks where n and m grow proportionally, the dominating behavior is $\Theta(mn)$.

5.6 Limitations and Future Work

While this study provides valuable insights into cost optimization within MTD frameworks using game-theoretic approaches, several limitations are still present:

- *Single Adversary Assumption.* In practice, systems often face multiple adversaries with varied skill levels and objectives. In a multiple-adversary scenario, the landscape is more complex and requires more nuanced defense mechanisms.

- *-p-][m mStatic System and Attacker Behavior.* We assume relatively static adversary behavior and a fixed system state. However, real-world adversaries can adapt in response to defenses, and cyber-physical systems themselves often undergo dynamic changes. MT2M is primarily formulated as a pre-attack planning tool in which the defender computes an optimal MTD configuration before an attack instance. Within a single attack, the adversary is modeled with static behavior; however, the framework can be re-optimized between attacks.
- *Lack of Real-Time Adaptivity.* MT2M does not utilize mechanisms for real-time adaptation based on evolving threat intelligence. This decreases performance in rapidly changing networks, where threats must be countered in real-time.

6 Conclusion

Traditional cyber defense strategies are ineffective in addressing advanced, emerging, and previously unrecognized threats, necessitating the development of more dynamic and adaptive solutions. This research investigates the cost optimization of MTD through Bayesian Stackelberg game theory, crafting optimal strategies for the defender and system manager. This approach allows defenders to reduce costs by strategically shifting resources, effectively diverting attacks. Optimal defensive scenarios were established, demonstrating how resource redistribution across nodes can reduce costs, taking into account the criticality of resources and node capacity constraints.

Our numerical simulations confirm that increasing the number of nodes and resources strengthens network security and cost efficiency, as the defender gains more flexibility in resource allocation. The findings further suggest that a system composed of a diverse range of resources increases the uncertainty faced by adversaries, thus strengthening the effectiveness of MTD strategies. The practical implementation of MTD schemes presents challenges, mainly due to the operational costs associated with continuous resource reallocation. To address this, MT2M introduces a cost-efficient network management framework. The framework reduces unnecessary movements while maintaining robust security and minimizing overall defender costs.

On a broader overview of MT2M, we discuss some candidate research. We include this research as more than just future work, as it enhances MT2M by expanding into wider disciplines of study. MT2M serves as a foundation for future research to investigate the application of machine learning techniques. These techniques, particularly reinforcement learning, facilitate the development of defense policies that evolve based on observed attacker behaviors. This could maintain optimal security postures with minimized costs. A natural extension is to relax the static adversary assumption by incorporating learning-based attacker models and repeated interactions. This would enable analyzing whether the cost benefits of MT2M persist when adversaries adapt their strategies over multiple attacks. Further applications can be found in developing methodologies to assess and prioritize different segments of the attack surface. By identifying areas with higher risk or criticality, defenders can allocate resources more effectively. As such, defenders ensure robust security while optimizing costs.

Acknowledgments — This work was supported by the Spanish Ministry of Science and Innovation through the projects PID2021-125962OB-C31 "SECURING/CYBER" and PID2024-156914OB-C41 "SAFE/CYBER". Additional funding was provided by the ARTEMISA International Chair in Cybersecurity (C057/23) and the DANGER Strategic Project of Cybersecurity (C062/23), both funded by the Spanish National Institute of Cybersecurity through the European Union NextGenerationEU and the Recovery, Transformation, and Resilience Plan.

References

[1] Lei, C., Zhang, H.-Q., Tan, J.-L., Zhang, Y.-C., Liu, X.-H.: Moving target defense techniques: A survey. Security and Communication Networks **2018**(1), 3759626 (2018) https://doi.org/10.1155/2018/3759626

[2] Sengupta, S., Chowdhary, A., Sabur, A., Alshamrani, A., Huang, D., Kambhampati, S.: A survey of moving target defenses for network security. IEEE Communications Surveys & Tutorials **22**(3), 1909–1941 (2020) https://doi.org/10.1109/COMST.2020.2982955

[3] Ghosh, A., Pendarakis, D., Sanders, W.: Moving target defense co-chair's report-national cyber leap year summit 2009. Technical report, Federal NITRD Program, Washington, DC, USA (2009). https://www.nitrd.gov/nitrdgroups/images/b/bd/National_Cyber_Leap_Year_Summit_2009_CoChairs_Report.pdf

[4] Douligeris, C., Mitrokotsa, A.: Ddos attacks and defense mechanisms: a classification. In: Proceedings of the 3rd IEEE International Symposium on Signal Processing and Information Technology (IEEE Cat. No. 03EX795), pp. 190–193. IEEE, Darmstadt, Germany (2003). https://doi.org/10.1109/ISSPIT.2003.1341092 . IEEE

[5] Segovia-Ferreira, M., Rubio-Hernan, J., Cavalli, A.R., Garcia-Alfaro, J.: A Survey on Cyber-Resilience Approaches for Cyber-Physical Systems. ACM Computing Surveys (2025) https://doi.org/10.1145/3652953

[6] Zscaler: What is Deception Technology? Importance & Benefits| Zscaler (2023). https://www.zscaler.com/resources/security-terms-glossary/what-is-deception-technology Accessed 2023-08-02

[7] Gonzalez-Granadillo, G., Dubus, S., Motzek, A., Garcia-Alfaro, J., Alvarez, E., Merialdo, M., Papillon, S., Debar, H.: Dynamic risk management response system to handle cyber threats. Future Generation Computer Systems **83**, 535–552 (2018) https://doi.org/10.1016/j.future.2017.05.043

[8] Jia, Q., Sun, K., Stavrou, A.: MOTAG: Moving Target Defense against Internet Denial of Service Attacks. In: 2013 22nd International Conference on Computer Communication and Networks (ICCCN), pp. 1–9. IEEE, Nassau, Bahamas (2013). https://doi.org/10.1109/ICCCN.2013.6614155

[9] Wright, M., Venkatesan, S., Albanese, M., Wellman, M.P.: Moving Target Defense against DDoS Attacks: An Empirical Game-Theoretic Analysis. In: 3rd ACM Workshop on Moving Target Defense, pp. 93–104. ResearchGate, Vienna, Austria (2016). https://doi.org/10.1145/2995272.2995279

[10] Feng, X., Zheng, Z., Cansever, D., Swami, A., Mohapatra, P.: A signaling game model for moving target defense. In: IEEE INFOCOM 2017 - IEEE Conference on Computer Communications, pp. 1–9. IEEE, Atlanta, GA, USA (2017). https://doi.org/10.1109/INFOCOM.2017.8057200

[11] A Petrosyan, L.a.: Recent Advances in Game Theory and Applications. Springer, Cham, Switzerland (2016)

[12] John, v.N.H., Oskar, M.: The Theory of Games and Economic Behaviour. Princeton University Press, Princeton, NJ, USA (1944)

[13] Kiennert, C., Ismail, Z., Debar, H., Leneutre, J.: A survey on game-theoretic approaches for intrusion detection and response optimization. ACM Computing Surveys (CSUR) **51**(5), 1–31 (2018)

[14] Yang, S., Zhang, Y., Wu, C.: Attack-Defense Quantification Based On Game-Theory. arXiv. https://doi.org/10.48550/arXiv.1902.10439

[15] Li, X., Meng, M., Hong, Y., Chen, J.: A Survey of Decision Making in Adversarial Games. arXiv. https://doi.org/10.48550/arXiv.2207.07971

[16] Milosevic, J., Dahan, M., Amin, S., Sandberg, H.: Strategic Monitoring of Networked Systems with Heterogeneous Security Levels. arXiv. https://doi.org/10.48550/arXiv.2304.04131

[17] Khouzani, M.H.R., Sen, S., Shroff, N.B.: An Analytical Approach to the Adoption of Asymmetric Bidirectional Firewalls: Need for Regulation? arXiv. https://doi.org/10.48550/arXiv.1203.1687

[18] Wadhawan, Y., Neuman, C.: Defending cyber-physical attacks on oil pipeline systems: A game-theoretic approach. In: Proceedings of the 1st International Workshop on AI for Privacy and Security. PrAISe '16. Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2970030.2970032

[19] Aljaradat, A., Sarkar, G., Shukla, S.K.: Modelling cybersecurity impacts on digital payment adoption: A game theoretic approach. Journal of Economic Criminology **5**, 100089 (2024) https://doi.org/10.1016/j.jeconc.2024.100089

[20] Cai, G.-l., Wang, B.-s., Hu, W., Wang, T.-z.: Moving target defense: state of the art and characteristics. Frontiers of Information Technology & Electronic Engineering **17**(11), 1122–1153 (2016)

[21] Jalowski, L., Zmuda, M., Rawski, M.: A survey on moving target defense for networks: A practical view. Electronics **11**(18), 2886 (2022) https://doi.org/10.3390/electronics11182886

[22] Li, H., Shen, W., Zheng, Z.: Spatial-Temporal Moving Target Defense: A Markov Stackelberg Game Model. arXiv. https://doi.org/10.48550/arXiv.2002.10390

[23] Ahmad Kassem, J., Rifà-Pous, H., Garcia-Alfaro, J.: Revisiting a probabilistic moving target defense strategy to handle attacks against network nodes with multiple resources. In: Lecture Notes in Networks and Systems, Vol. 1284, Advances in Information and Communication, vol. 1284 LNNS, pp. 536–554 (2025). https://doi.org/10.1007/978-3-031-85363-0_34

[24] Shukla, P., An, L., Chakrabortty, A., Duel-Hallen, A.: A robust stackelberg game for cyber-security investment in networked control systems. IEEE Transactions on Control Systems Technology **31**(2), 856–871 (2023) https://doi.org/10.1109/TCST.2022.3207671

[25] Charpentier, A., Neal, C., Boulahia-Cuppens, N., Cuppens, F., Yaich, R.: Real-time defensive strategy selection via deep reinforcement learning. In: Proceedings of the 18th International Conference on Availability, Reliability and Security. ARES '23, pp. 1–11. Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3600160.3600176

[26] Xu, L., Skoularidou, M., Cuesta-Infante, A., Veeramachaneni, K.: Modeling Tabular data using Conditional GAN. arXiv. https://doi.org/10.48550/arXiv.1907.00503

[27] Bello, I., Pham, H., Le, Q.V., Norouzi, M., Bengio, S.: Neural Combinatorial Optimization with Reinforcement Learning. arXiv. https://doi.org/10.48550/arXiv.1611.09940

[28] Bengio, Y., Lodi, A., Prouvost, A.: Machine Learning for Combinatorial Optimization: a Methodological Tour d'Horizon. arXiv. https://doi.org/10.48550/arXiv.1811.06128

[29] Abdelkarim, A.T., Marwan, M., Baslam, M.: Stackelberg security game for optimizing cybersecurity decisions in cloud computing. Security and Communication Networks **2023**(1), 2811038 (2023) https://doi.org/10.1155/2023/2811038 https://onlinelibrary.wiley.com/doi/pdf/10.1155/2023/2811038

[30] Abdelkhalek, M., Hyder, B., Govindarasu, M., Rieger, C.G.: Moving target defense routing for sdn-enabled smart grid. In: 2022 IEEE International Conference on Cyber Security and Resilience (CSR), pp. 215–220 (2022). https://doi.org/10.1109/CSR54599.2022.9850341

[31] Atlassian: Calculating the cost of downtime (2024). https://www.atlassian.com/

incident-management/kpis/cost-of-downtime Accessed 2025-01-23

[32] Shepherd, D.: Why DNS exploits continue to be a top attack vector in 2024 (2024). https://www.tahawultech.com/home-slide/why-dns-exploits-continue-to-be-a-top-attack-vector-in-2024/ Accessed 2025-01-23

[33] Jeffrey, M.: Return on investment analysis for e-business projects. Internet Encyclopedia **3**, 211–236 (2004)

[34] Motzek, A., Granadillo, G., Debar, H., Garcia-Alfaro, J., Möller, R.: Selection of pareto-efficient response plans based on financial and operational assessments. EURASIP Journal on Information Security **2017** (2017) https://doi.org/10.1186/s13635-017-0063-6

[35] Gonzalez-Granadillo, G., Doynikova, E., Garcia-Alfaro, J., Kotenko, I., Fedorchenko, A.: Stateful rori-based countermeasure selection using hypergraphs. Journal of Information Security and Applications **54**, 102562 (2020) https://doi.org/10.1016/j.jisa.2020.102562

[36] Palavesam, K.V., Krishnamoorthy, M.V., S M, A.: A comparative study of service mesh implementations in kubernetes for multi-cluster management. Journal of Advances in Mathematics and Computer Science **40**(1), 1–16 (2025) https://doi.org/10.9734/jamcs/2025/v40i11958

[37] Chandramouli, R., Butcher, Z., *et al.*: Building secure microservices-based applications using service-mesh architecture. NIST Special Publication **800**, 204 (2020)

[38] Tamiru, M.A., Tordsson, J., Elmroth, E., Pierre, G.: An experimental evaluation of the kubernetes cluster autoscaler in the cloud. In: 2020 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), pp. 17–24 (2020). https://doi.org/10.1109/CloudCom49646.2020.00002

[39] Chavez, A.R., Stout, W.M.S., Peisert, S.: Techniques for the dynamic randomization of network attributes. In: 2015 International Carnahan Conference on Security Technology (ICCST), pp. 1–6 (2015). https://doi.org/10.1109/CCST.2015.7389661

[40] Xu, H., Cheng, G., Yang, X., Liu, W., Zhou, D., Guo, W.: Multi-dimensional moving target defense method based on adaptive simulated annealing genetic algorithm. Electronics **13**(3) (2024) https://doi.org/10.3390/electronics13030487

[41] Marin, E., Perino, D., Di Pietro, R.: Serverless computing: a security perspective. Journal of Cloud Computing **11**(1), 69 (2022)

[42] Faircloth, J.: Chapter 2 - Reconnaissance. In: Faircloth, J. (ed.) Penetration

Tester's Open Source Toolkit (Third Edition), pp. 29–93. Syngress, ??? (2011). https://doi.org/10.1016/B978-1-59749-627-8.10002-9

[43] Cheimonidis, P., Rantos, K.: Dynamic Risk Assessment in Cybersecurity: A Systematic Literature Review. Future Internet **15**(10), 324 (2023) https://doi.org/10.3390/fi15100324

[44] Conti, M., Gaspari, F.D., Mancini, L.V.: Know Your Enemy: Stealth Configuration-Information Gathering in SDN. arXiv (2016). https://doi.org/10.48550/arXiv.1608.04766

[45] Corporation, T.M.: Techniques - ICS | MITRE ATT&CK® (2020). https://attack.mitre.org/techniques/ics/ Accessed 2025-05-09

[46] Li, E., Kang, C., Huang, D., Hu, M., Chang, F., He, L., Li, X.: Quantitative Model of Attacks on Distribution Automation Systems Based on CVSS and Attack Trees. Information **10**(8), 251 (2019) https://doi.org/10.3390/info10080251

[47] Huang, S., Zhu, Q.: PsybORG+: Cognitive Modeling for Triggering and Detection of Cognitive Biases of Advanced Persistent Threats (2024). https://doi.org/10.48550/arXiv.2408.01310

[48] Paruchuri, P., Pearce, J.P., Marecki, J., Tambe, M., Ordonez, F., Kraus, S.: Efficient algorithms to solve bayesian stackelberg games for security applications. In: AAAI, pp. 1559–1562. AAAI, Palo Alto, CA, USA (2008)

[49] Kleinberg, J., Tardos, E.: Algorithm Design, p. 491. Pearson, Upper Saddle River, N.J. (2003)

[50] Coffman, E.G., Courcoubetis, C., Garey, M.R., Johnson, D.S., Shor, P.W., Weber, R.R., Yannakakis, M.: Bin packing with discrete item sizes, part i: Perfect packing theorems and the average case behavior of optimal packings. SIAM Journal on Discrete Mathematics **13**(3), 384–402 (2000) https://doi.org/10.1137/S0895480197325936

[51] Fukunaga, A.S., Korf, R.E.: Bin completion algorithms for multicontainer packing, knapsack, and covering problems. Journal of Artificial Intelligence Research **28**, 393–429 (2007) https://doi.org/10.1613/jair.2106 1110.2209 [cs]

[52] Gil Herrera, J., Botero, J.F.: Resource Allocation in NFV: A Comprehensive Survey. IEEE Transactions on Network and Service Management **13**(3), 518–532 (2016) https://doi.org/10.1109/TNSM.2016.2598420

[53] Duo, W., Zhou, M., Abusorrah, A.: A survey of cyber attacks on cyber physical systems: Recent advances and challenges. IEEE/CAA Journal of Automatica Sinica **9**(5), 784–800 (2022) https://doi.org/10.1109/JAS.2022.105548

[54] Sasi, T., Lashkari, A.H., Lu, R., Xiong, P., Iqbal, S.: A comprehensive survey on IoT attacks: Taxonomy, detection mechanisms and challenges. Journal of Information and Intelligence **2**(6), 455–513 (2024) https://doi.org/10.1016/j.jiixd.2023.12.001

[55] Kassem, J.A.: Strategic Cost-based Optimization of Cyber Defense in Variable Constraints Systems. GitHub. Accessed: February 11, 2025 (2025). https://github.com/JamilahKassem/Variable-Criticality-Network-Defense