Quantifying the Impact of CVSS Score Ordering on Attack Paths

Baiardi Fabrizio, Sammartino Vincenzo, and Ruggieri Salvatore

Dipartimento di Informatica, Università di Pisa, Italia

Abstract. A critical open question in cyber risk management is whether the risk is defined by the mere presence of high-scoring vulnerabilities or by the specific topological arrangement of their severity. This paper discusses how the success of an actor depends heavily not only on the aggregate vulnerability scores but also on the distribution of these scores along attack paths. To investigate this problem, we apply adversary simulation via security twins and introduce a formal framework for *Permutation-Based Sensitivity Analysis* to quantify the impact of CVSS score positioning on adversary success by shuffling the CVSS scores of the steps of attack path. By applying this strategy in two distinct network topologies and considering different adversary strategies, we show that the order of CVSS scores fundamentally alters the risk profile. We also show that distributions where CVSS scores decrease towards the target strongly affect resource-constrained adversaries by trapping them in sunk-cost loops. Conversely, distributions where CVSS scores increase towards the target, are most effective at early deterrence. Furthermore, we provide empirical evidence that meshed topologies are significantly less sensitive to positional hardening in both directions than hierarchical ones, providing further support for the adoption of Zero Trust solutions.

Keywords: Security Twin, Sensitivity Analysis, CVSS, Attack Path Permutation, Adversary Simulation, Risk Assessment, Graph Theory.

1 Introduction

A fundamental challenge in risk management of cyber intrusions is the *sensitivity* to the system context of the outcomes of the attack paths enabling alternative intrusions, that is, the distribution of CVSS scores along a path. Standard risk metrics assign a static score (0.0–10.0) to a vulnerability regardless of its location on a path [1]. Consequently, a vulnerability with a CVSS score of 9.8 on a perimeter firewall is often managed as a vulnerability with the same score on an air-gapped internal server.

Although Attack Graph (AG) theory recognizes that reachability matters [2], [3], the predominant risk aggregation practice in both academia and industry reduces multi-step paths to scalar metricsmean CVSS, maximum CVSS, or product of probabilitiesthat are invariant to score ordering (formalized in Section 3.3, Proposition 1). This reductionism has persisted for two structural

reasons: first, early AG models were primarily designed for reachability analysis rather than adversary decision modeling; second, the sensitivity of adversary *dynamics* to score arrangement is not analytically tractable under standard probabilistic path models, necessitating simulation-based approaches that were computationally prohibitive until the advent of high-fidelity Security Twins [4]. Consider a valid attack path of three logical steps. Does assigning the CVSS sequence $\langle 9.0, 9.0, 4.0\rangle$ present the same risk as $\langle 4.0, 9.0, 9.0\rangle$? As Proposition 2 establishes, the memoryless path success probability is identical in both cases. Yet for a resource-constrained adversary with a time budget or a detection threshold, the arrangement is decisive.

This paper addresses the score distribution problem by adopting a security twin, ST, [4], [5], [6], [7]. By working on a live, digital model of an IT/OT infrastructure, defenders can simulate intrusions by threat actors to evaluate the resilience of their system. through a dynamic, behavioral analysis. In particular, we define a *Permutation-Based Sensitivity Analysis* (PBSA), a sensitivity analysis framework based on *controlled permutations of CVSS scores* that changes the success probability of an attack path and, hence, of the corresponding intrusion.

This paper is structured as follows. Sect. 2 discussed related work and Sect. 3 introduces the proposed framework. The following two sections describe, respectively, the experimental methodology we have developed and the target systems we target in our experiments. Experimental results are presented in Sect. 6 and discussed in the next section. The last section draws some preliminary conclusions and outlines future work.

Our contributions are:

1. **Formal Permutation Model:** We define a localized permutation operator Φ_k that swaps the CVSS scores of the last k steps of an attack path without altering the system architecture or the other attributes of a step.
2. **Positional Risk Quantification:** We analyze how the ordering of CVSS scores impacts three distinct adversary strategies: *Greedy Value*, *Max Probability*, and *Stealth*.
3. **Multi-Topology Validation:** We apply our methodology to two distinct target systems: a deep, hierarchical university network and a flat, meshed corporate network.

2 Related Work

This research intersects with Attack Graph analysis, Breach and Attack Simulation (BAS), and sensitivity analysis in probabilistic modeling. We define an *attack path* as a sequence of exploits that an adversary executes to reach a goal state. A path is *valid* if the post-conditions of exploit k satisfy the pre-conditions of exploit $k + 1$. The adversary selects exploits according to a strategy that encodes its preferences over risk, effort, and reward.

2.1 Attack Graphs and CVSS-Based Risk Metrics

Traditional Attack Graphs (AGs) describe potential sequences of exploits [8], and most models derive edge probabilities directly from CVSS Exploitability sub-scores [9]. A predominant body of work computes the *Mean Time to Compromise* (MTTC) or the probability of reaching a target node via methods such as model checking [2] or Bayesian network inference [3]. However, these models treat CVSS scores as *static node or edge attributes* that contribute additively or multiplicatively to a path-level aggregate, entirely disregarding the *sequential arrangement* of scores along the path.

The implicit assumption in prior work is that $\text{Risk}(\mathcal{P}) = f(\{\Sigma(\mathcal{P})\})$, i.e., risk is a function of the *set* of scores. Our framework challenges this assumption by demonstrating that $\text{Risk}(\mathcal{P}) = f(\langle\Sigma(\mathcal{P})\rangle)$, i.e., risk is a function of the *ordered sequence*. This distinction, while conceptually straightforward, has received no formal treatment in the attack graph literature. The gap is not merely academic: as we show in Section 6, two paths with identical aggregate CVSS profiles can exhibit success rate differences exceeding 60 percentage points for specific adversary strategies.

2.2 Sensitivity Analysis in Probabilistic Security Models

Sensitivity analysis is a well-established methodology in operations research and reliability engineering **saltelli2008global**, yet its application to cyber risk models remains limited. Existing CVSS-based sensitivity studies typically perform *score sensitivity*, varying the numerical value of a single vulnerability score and observing its effect on a global metric [10]. This paper introduces *positional sensitivity*, a structurally distinct form of analysis that holds the *values* of scores invariant while perturbing their *arrangement*. This is analogous to the distinction between marginal sensitivity and structural sensitivity in stochastic models, and constitutes a methodologically novel contribution with respect to prior CVSS-based analyses.

2.3 Adversary Modeling and BAS

Automated Red Teaming and BAS tools often rely on planning algorithms or Reinforcement Learning (RL) [11]. While these studies optimize the attacker's policy, they neglect the sensitivity of the policy to the underlying distribution of vulnerability scores. We build upon the actor definitions in [12] and test the robustness of these strategies against score ordering [10], [13], an evaluation dimension absent from existing BAS literature.

3 Formal Framework

This section describes the ST as an input of an emulation engine, formalizes the relationship between CVSS scores and exploit success probability, and introduces the permutation operator Φ_k as a tool for topological sensitivity analysis.

To rigorously quantify the impact of vulnerability positioning, our framework decouples the *existence* of a vulnerability and its score from its *location* [14].

3.1 The Security Twin as a Probabilistic Graph

We model a ST not merely as a static map of assets, but as an input of an emulation engine that merges the information in the twin with the one about an attacker strategy to emulate step-by step how the strategy selects the next attack in its intrusions and builds an attack path. In this way, the twin becomes a computational substrate for a discrete-time stochastic process [15].

Vertex and Edge Definitions A ST represents an ICT/OT infrastructure as a directed property graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where vertices $\mathcal{V}$ represents the atomic entities of the infrastructure, including workstations, servers, and network devices. The edges $\mathcal{E}$ represent possible lateral movements or privilege escalations. An edge $e_{u,v}$ exists if and only if there is a logical or physical path from asset u to asset v (reachability) and asset v possesses a known vulnerability exploitable from u.

The Semantics of the Edge Tuple Each edge is paired with the attributes $\omega(e) = \langle s_{u,v}, c_{u,v}, \eta_{u,v} \rangle$ that govern the adversary's interaction with the system:

- $s_{u,v} \in [0, 10]$ is the **CVSS Base Score** of the vulnerability. It represents the intrinsic severity of the flaw.
- $c_{u,v} \in \mathbb{R}^+$ is the **Time Cost**, an abstract measure of the effort to execute the exploit. It ranges from seconds for a script to hours for a brute-force.
- $\eta_{u,v} \in [0, 1]$ is the **Noise Level** and it models the interaction with an **Intrusion Detection System (IDS)**. It represents the probability that the adversary action generates a signature or anomaly that triggers an IDS alert.

Mapping Scores to Probability Our framework assumes that the *Probability of Success* $P_{\text{succ}}(e)$ of an adversary attempting to cross edge e is a monotonically non-decreasing function of the CVSS Base Score $s_{u,v}$:

$$P_{\text{succ}}(e_{u,v}) = f(s_{u,v}) \approx \frac{s_{u,v}}{10} \tag{1}$$

This linear approximation is grounded in the semantics of the CVSS Base Score specification **cvssv3spec**. Specifically, the three sub-metrics that most directly govern exploitability—*Attack Complexity* (AC), *Privileges Required* (PR), and *User Interaction* (UI)—are each inversely correlated with score magnitude: a CVSS 9.8 vulnerability by construction implies AC:Low, PR:None, UI:None, which corresponds to a remotely and trivially executable exploit. Conversely, a CVSS 2.0 score implies AC:High, PR:High, UI:Required, corresponding to an exploit with severe preconditions.

Formal Justification. Let $\phi : [0, 10] \to [0, 1]$ be the inverse of the normalized CVSS Attack Complexity contribution. CVSS v3.1 assigns ϕ(AC:Low) = 0.77 and ϕ(AC:High) = 0.44. The linear mapping $P_{\text{succ}} = s/10$ constitutes a monotone approximation of the underlying piece-wise function defined by the CVSS specification, consistent with the monotonicity property empirically validated in [12].

Acknowledged Limitations. We acknowledge that this mapping constitutes a simplification. In practice, a high-scoring vulnerability may be temporarily unexploitable due to missing exploit code, environmental factors (e.g., CVSS Environmental Metrics), or active mitigations not captured by the Base Score. The CVSS Temporal Metrics (Exploit Code Maturity, Remediation Level) could refine P_{succ}, and their integration constitutes a direction for future work (Section 8). Nonetheless, for the purposes of this study, which seeks to isolate the effect of *positional arrangement* rather than score magnitude, the monotone assumption is both analytically sufficient and empirically grounded in prior literature.

3.2 The Logic of Score Permutation

Risk assessment strategies that aggregate scores (e.g., "Average CVSS of 7.5") neglect topological information. To measure risk sensitivity to the score *distribution*, we rearrange the scores associated with an attack path without altering the path's logical validity, i.e., without changing the pre- and post-conditions of the exploits [16] paired with the steps of the path.

Let $\mathcal{P} = \langle e_1, e_2, \ldots, e_L \rangle$ be a valid attack path discovered by an actor from an entry point to a critical asset. Let $\Sigma(\mathcal{P}) = \langle s_1, s_2, \ldots, s_L \rangle$ be the sequence of CVSS scores associated with the edges of this path, where s_i is the score of e_i.

We define the *Last-k Score Permutation Operator*, denoted as Φ_k, as a transformation that acts upon the sequence $\Sigma(\mathcal{P})$. The operator preserves the values of the first $L - k$ elements and permutes the final k elements.

$$\Sigma' = \Phi_k(\Sigma(\mathcal{P})) = \langle s_1, \ldots, s_{L-k}, \pi(s_{L-k+1}, \ldots, s_L) \rangle \tag{2}$$

where π is a permutation of k elements. Crucially, the resulting sequence Σ' is then re-mapped to the original path $\mathcal{P}$. The edge e_i takes on the new score s'_i. This generates a new path with the same logical steps, but it alters the path score distribution. In this way, it can describe scenarios where defenders apply distinct patching strategies (e.g., hardening the perimeter vs. hardening the core) while the underlying vulnerabilities remain constant.

3.3 Formal Properties of the Permutation Operator

We now establish the formal properties of Φ_k to ground the subsequent empirical analysis.

Definition 1 (Score Sequence Space). Let $\mathcal{S}_L = [0, 10]^L$ denote the space of all possible score sequences of length L. A valid attack path $\mathcal{P}$ of length L induces a point $\Sigma(\mathcal{P}) \in \mathcal{S}_L$.

Definition 2 (Permutation Orbit). The *Last-k permutation orbit* of $\Sigma(\mathcal{P})$ is the set:

$$\mathcal{O}_k(\Sigma) = \{\Phi_k(\Sigma) : \pi \in \mathrm{Sym}(k)\} \tag{3}$$

where $\mathrm{Sym}(k)$ is the symmetric group of degree k. The orbit has cardinality $|\mathcal{O}_k(\Sigma)| \leq k!$, with equality when all k suffix scores are distinct.

Proposition 1 (Metric Preservation). The permutation operator Φ_k is metric-preserving with respect to any score-aggregate metric $\bar{s}(\Sigma) = g(\{s_1, \ldots, s_L\})$ that depends only on the *multiset* of scores:

$$\forall \pi \in \mathrm{Sym}(k) : \bar{s}(\Phi_k(\Sigma)) = \bar{s}(\Sigma) \tag{4}$$

Proof. By construction, Φ_k permutes a subset of scores without altering their values; hence the multiset $\{s_1, \ldots, s_L\}$ is invariant. Any function g defined on multisets (e.g., sum, mean, max) is therefore constant over $\mathcal{O}_k(\Sigma)$. □

Proposition 1 constitutes the formal justification for why aggregate-based metrics are *insufficient* for risk differentiation: they are constant over the entire orbit $\mathcal{O}_k$, rendering them insensitive to the structural properties that, as we show empirically, are decisive for adversary success.

Proposition 2 (Monotone Sensitivity Bounds). For an adversary that evaluates each step independently (i.e., π^* is a greedy policy), the path success probability $P_{\mathrm{path}}(\Sigma) = \prod_{i=1}^{L} f(s_i)$ is permutation-invariant:

$$P_{\mathrm{path}}(\Phi_k(\Sigma)) = P_{\mathrm{path}}(\Sigma) \quad \forall \pi \in \mathrm{Sym}(k) \tag{5}$$

Proof. The product is commutative over real numbers. □

This result establishes a critical baseline: *the overall path success probability is identical across all permutations for a memoryless adversary.* Therefore, any empirically observed difference in success rate must be attributed to *dynamic adversary behavior*—specifically, early path abandonment, resource depletion, or IDS-triggered termination—rather than to changes in the raw product of probabilities. This formally motivates the need for a simulation-based approach (Section 4) over analytical path probability computation.

Definition 3 (Topological Sensitivity, refined). Let $M(S, T)$ be a scalar performance metric for strategy S on Security Twin T. The *Last-k Topological Sensitivity* under permutation operator Φ_k^π is:

$$\Delta_k^\pi(S) = \frac{|M(S,T) - M(S, \Phi_k^\pi(T))|}{M(S,T)} \tag{6}$$

The *worst-case sensitivity* and *best-case sensitivity* over the orbit are:

$$\Delta_k^{\max}(S) = \max_{\pi \in \mathrm{Sym}(k)} \Delta_k^\pi(S) \tag{7}$$

$$\Delta_k^{\min}(S) = \min_{\pi \in \mathrm{Sym}(k)} \Delta_k^\pi(S) \tag{8}$$

In this work, we focus on the two extreme sorted permutations $\Phi_k^\uparrow$ and $\Phi_k^\downarrow$, which we conjecture (and empirically verify) to approximate $\Delta_k^{\max}$ for monotone adversary utility functions.

3.4 Defensive Archetypes: Hard Shell vs. Hard Core

The operator Φ_k can model two philosophies of network defense that correspond to sorted permutations and represent extreme structural configurations.

Ascending Sort ($\Phi_k^{\uparrow}$): The "Hard Shell" Model This operator reorders the last k CVSS scores in a path **ascending order** ($s'_i \leq s'_{i+1}$) and assigns them to the edges $e_{L-k+1} \dots e_L$.

- **Structure:** The scores of the last k steps are assigned from the lowest (Hardest) to the highest (Easiest).
- **Defence Strategy:** "Perimeter-Focused" where the adversary encounters the toughest resistance immediately at the entry point (e.g., a fully patched firewall or strict VPN). However, if the perimeter is breached, the internal network is soft due to, among others, unpatched legacy systems with High CVSS scores.
- **Hypothesis:** We expect this topology will deter risk-averse actors early but fail catastrophically if breached.

Descending Sort ($\Phi_k^{\downarrow}$): The "Hard Core" Model This operator reorders the last k CVSS scores in **descending order** ($s'_i \geq s'_{i+1}$) and assigns them to the edges $e_{L-k+1} \dots e_L$.

- **Structure:** The scores of the final k steps are assigned from the highest (Easiest) to the lowest (Hardest).
- **Defence Strategy:** "Defense-in-Depth" where the critical assets are the most hardened. The outer layers may be vulnerable (e.g., a phishing-prone workstation with a Critical browser bug), acting as a lure. But as the adversary moves closer to the Data Center, the vulnerabilities become scarce and difficult to exploit.
- **Hypothesis:** We expect this topology will trap actors in a "sunk cost" loop, increasing dwell time.

3.5 Sensitivity Metrics

Finally, to quantify the impact of these permutations, we define the *Topological Sensitivity* Δ_k. Let $M(S, \mathcal{G})$ be a performance metric (such as Success Rate or Average Time) for an adversary strategy S on a ST T. The sensitivity is the normalized deviation of the metric under permutation:

$$\Delta_k(S) = \frac{|M(S,T) - M(S, \Phi_k(T))|}{M(S,T)} \tag{9}$$

A high value of Δ_k denotes that the adversary's performance is heavily dependent on the specific order of scores, implying that "where" a patch is applied is as critical as "which" patch is applied.

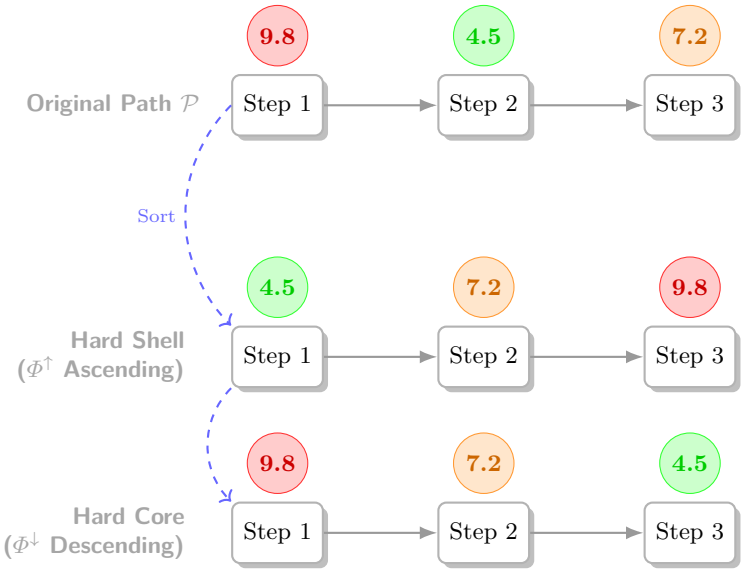

Fig. 1: Visual representation of the Permutation Operator Φ_k. The logical steps remain fixed, but the CVSS scores (difficulty) are redistributed to create Hard Shell or Hard Core profiles.

4 Methodology

The experimental methodology follows a three-step pipeline executed through the NotLine platform [17] that includes a simulation engine to emulate the behavior of a threat actor [18]. The platform pairs each actor with both a goal, a set of access rights the actor aims to acquire and a strategy that defines how the actor chooses an attack given its current access rights and its goal.

4.1 Step 1: Path Discovery via Actor Simulation

First, we run a set of experiments using the simulation engine in "Discovery Mode". Each experiment simulates a threat actor that applies one of three strategies: Greedy, Max Probability and Stealth [19]. The actors have the same goal: the control of an Active Directive node and the exfiltration of a file.

Greedy Value Strategy (High Risk, High Reward) This is an aggressive strategy motivated by immediate gain. Each node n in the target system is assigned a business value $V(n)$. Under a worst-case assumption, the actor knows this value and it calculates a Return on Investment (ROI) metric for each action it can execute. Then, it selects the action that maximises the ratio of potential value to reach the goal with respect to estimated time cost, disregarding detection risk.

Max Probability Strategy (Technological Determinism) This strategy represents a risk-averse but technically proficient attacker. Instead of business value, this actor prioritizes the reliability of exploits and actions.

Stealth Strategy (The APT Approach) This is the strategy of a sophisticated human adversary operating under a "detection budget." The actor maintains a cumulative noise counter η. For every potential action, it evaluates the associated detection probability $P_{det}(a)$. In practice, this actor will bypass a fast, noisy exploit (e.g., a brute-force attack) in favour of a slow, silent technique (e.g., pass-the-hash), effectively trading time for survival.

The goal of Step 1 is not to discover the best strategy, but to identify the set of all viable attack paths $\mathbb{P}$ that each strategy can actually discover and traverse in the baseline Target System. For each discovered path $\mathcal{P} \in \mathbb{P}$, we extract the sequence of CVSS scores $\Sigma(\mathcal{P})$.

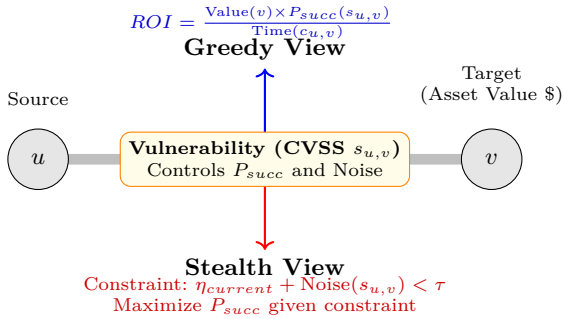

Fig. 2: Divergent Agent Heuristics. The same edge with a specific CVSS score is evaluated differently: the Greedy agent (top) sees it as an ROI opportunity, while the Stealth agent (bottom) sees it primarily as a noise budget cost.

4.2 Step 2: Modification of CVSS Attributes

We generate derived instances of the Target System by modifying the attributes of the edges in the ST. For a fixed k (varying from 2 to the maximum path length L_{max}), we apply the permutation operators $\Phi_k^{\uparrow}$ (Hard Shell) and $\Phi_k^{\downarrow}$ (Hard Core) to the sequence of scores identified in Step 1. We then replace the property $s_{u,v}$ of each edge e in the ST with the new permuted value. If an edge e belongs to multiple paths, its modified CVSS score is determined by the path with the highest traversal frequency in Step 1 to avoid conflicts. This step models a reconfiguration of the success probabilities of attacks and hence of risk distribution as when perimeter nodes are patched while internal nodes are left with high-severity flaws, or vice versa. The reconfiguration does not affect the logical network topology or the edges.

4.3 Step 3: Validation and Statistical Convergence

We employ a Monte Carlo method and a statistical convergence protocol to ensure the robustness of our results rather than running a fixed number of simulations. We execute adversary simulations until the quantitative metrics, mainly the number of new intrusion paths, stabilize. [20] To this purpose, the engine computes the running mean μ and standard deviation σ of the number of intrusion paths. The engine runs further simulations until the width of the 95% confidence interval for the mean is within a 5% relative error margin:

$$\frac{1.96 \cdot \sigma}{\sqrt{N}} \leq 0.05 \cdot \mu \tag{10}$$

This stopping criterion ensures that the differences between strategies are statistically significant and not artefacts of random variance. Both the confidence level and the relative error can be tuned according to the target system and the required accuracy.

Dynamic Path Re-routing: It is important to note that while the permutation is applied to the edges of the paths identified in Step 1, the agents in Step 3 are not forced to strictly replay these paths. Since each strategy makes dynamic decisions based on the current state of the graph, the modification of CVSS scores alters the cost-benefit landscape. Consequently, an agent may deviate from the original path $\mathcal{P}$ and select an alternative route $\mathcal{P}'$ that the strategy previously neglected but has now become the best one due to the score permutation. This "path switching" behavior is a critical component of the analysis, as it measures not just the resilience of a specific path, but the overall resilience of the target system against adaptive adversaries.

Adaptive Adversary Assumption: Let $\pi^*(G)$ be the optimal sequence of actions generated by an agent on a ST S that the permutation Φ_k transforms into $S' = \Phi_k(S)$. Our simulation explicitly accounts for the fact that:

$$\pi^*(S') \neq \pi^*(S) \tag{11}$$

This inequality holds strictly for topologies with cycles or redundancy, where the agent can re-optimize its route. In strictly hierarchical topologies, the set of feasible paths is often a singleton, forcing $\pi^*(S') \approx \pi^*(S)$, which results in an higher sensitivity to permutations.

5 Experimental Setup

To ensure that our findings are related to topological properties, we designed our experiments around two real, rather different target environments. We have built the STs of these environments using the NotLine platform. The environments represent opposing archetypes of network architecture: the "Deep Hierarchy" and the "Flat Mesh" which are characterized, respectively by segmentation and chokepoints the former and by interconnectivity and redundancy the latter. These differences support the sensitivity of adversary strategies in different structural constraints.

5.1 System A: A Deep Hierarchical Network

The first system, *System A*, is a traditional university departmental network designed according to the "Defense-in-Depth" principle.

Topological Structure Structurally, the network topology of the system is a directed tree with a high depth-to-width ratio. It is strictly segmented into four concentric security zones, separated by stateful firewalls that enforce rigorous Access Control Lists (ACLs).

- **Zone 1 (Perimeter):** The Demilitarized Zone (DMZ), hosting public-facing services such as Web Servers and Mail Relays. This is the only entry point for an external attacker.
- **Zone 2 (Userland):** A Student and Staff Wi-Fi segment. This zone is characterized by high entropy, transient devices, and a larger attack surface, but it has limited routing privileges to the core.
- **Zone 3 (Management):** The Admin LAN, containing workstations for system administrators. This is a pivotal zone, as it stores the credentials to access the core.
- **Zone 4 (Core):** The Secure Data Center, hosting the Domain Controller and the critical database (the "Flag").

Operational Characteristics The system comprises 60 active hosts running a mix of Linux (Ubuntu/CentOS) and Windows Server, exposing over 250 distinct services. The defining characteristic of System A is the length and linearity of its attack paths. To reach its target from the attack surface, a threat actor has to successfully implement an attack path with from 6 to 8 steps. The system topology imposes a structural constraint on a path with very few alternative routes. If a node in the chain is hardened (low CVSS), it acts as a chokepoint. This makes System A the ideal testbed for analyzing the impact of linear permutations (Φ_k) on adversary persistence till reaching the goal of the intrusion.

5.2 System B: The Flat Meshed Network

The second target system, *System B*, corresponds to a modern, open-plan corporate office or a cloud-native environment. The system comprises 85 active hosts with a heterogeneous mix of consumer-grade OSs (Windows 10/11) and embedded firmware, exposing over 400 services.

Topological Structure Structurally, the topology of this system approximates a highly connected graph with a low diameter and a high clustering coefficient. Unlike System A, there is minimal vertical segmentation.

- **The Flat Plane:** Workstations, IoT devices (smart printers, cameras), and local file servers coexist on the same logical VLAN or are routed with permissive "Any-Any" rules.

– **Lateral Movement:** The graph exhibits a high branching factor. Once an adversary compromises a single node (e.g., a receptionist's PC), they have immediate line-of-sight to a multitude of other nodes, including lateral peers and servers.

Operational Characteristics The defining characteristic of System B is redundancy. The average attack path is short (from 3 to 4 steps) and the number of alternative paths is much larger than in System A. If a specific node is hardened, the adversary can easily pivot to a neighbor to bypass the obstacle. This environment serves as a stress test for the "Hard Shell" hypothesis: namely to determine the effectiveness of hardening the perimeter when the internal network enables rapid, chaotic lateral movement.

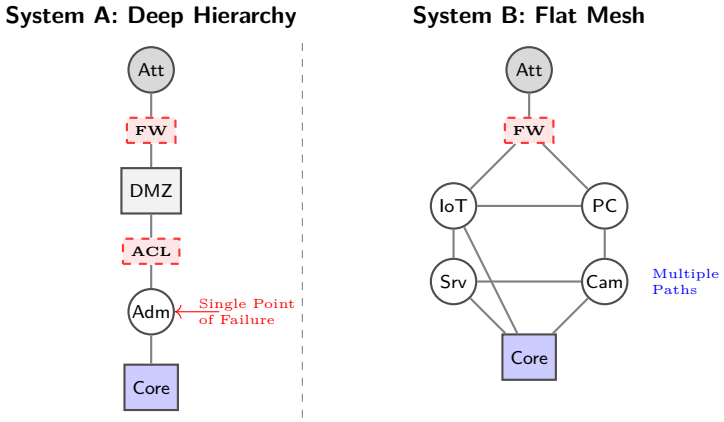

Fig. 3: Structural comparison of Target Systems. System A (left) forces linear traversal through chokepoints, making it sensitive to permutation. System B (right) offers redundant paths, diluting the impact of localized hardening.

5.3 Vulnerability Distribution and Simulation Parameters

For both systems, the NotLine platform builds the baseline ST and populates it with real-world vulnerability data. It also maps Common Platform Enumerations (CPEs) of the simulated assets to the National Vulnerability Database (NVD) to assign the proper CVSS scores. In the baseline configuration, the distribution of CVSS scores may be assumed as random, reflecting a typical unmanaged network where critical vulnerabilities (CVSS 9.0+) appear randomly on both perimeter and internal assets.

Defensive Posture (IDS): To simulate a realistic network environment, both systems are equipped with a simulated Intrusion Detection System (NIDS).

The simulations assume the NIDS monitors all edges and triggers an alert if the cumulative noise generated by an agent within a time window exceeds a specific threshold τ. The IDS modelling is critical for evaluating the Stealth strategy, as noisy actions are often associated with low-CVSS, brute-force exploits and they increase the probability of detection and subsequent blocking.

6 Experimental Results

As a robust basis for discussion, we present a set of experimental results covering different permutation depths (k) and comparing the distinct topologies of, respectively, System A and System B. All reported values satisfy the 95% confidence interval criterion defined in Section 4.

6.1 Impact of Permutation Depth (k)

As a first example of how the depth of the permutation k affects intrusions, we consider System A. Figure 4 shows how permutations affect the success rate of the *Max Probability* actor.

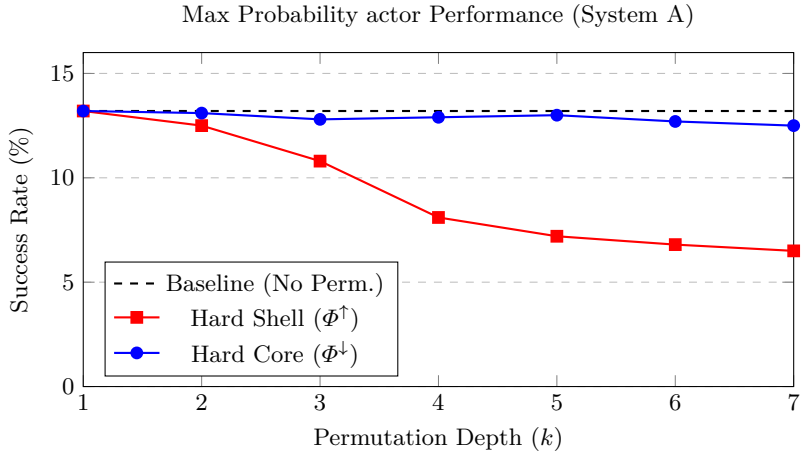

Fig. 4: Divergence of Success Rate as a function of permutation depth k for System A.

These results show that the **Hard Shell** configuration with Ascending CVSS causes a sharp decline in success rates as the actor struggles to breach the entry nodes. Conversely, the **Hard Core** configuration with Descending CVSS results in a stable success rate.

Table 1 details the metrics for all strategies across $k = 2, 4, 6$. We report data for both the **Hard Shell** ($\Phi^{\uparrow}$) and the **Hard Core** ($\Phi^{\downarrow}$) permutations to show the bidirectional sensitivity.

Table 1: Impact of Permutation Depth (k)). Hard Shell vs Hard Core.

Strategy	**Metric**	**Baseline**	$k=2$		$k=4$		$k=6$	
			Shell	**Core**	**Shell**	**Core**	**Shell**	**Core**
Greedy	Success (%)	11.4	10.9	12.1	6.2	18.5	4.1	19.5
	Time (h)	28.6	29.1	28.2	14.2	41.8	11.5	45.3
Max Prob	Success (%)	13.2	12.5	13.1	8.1	12.9	6.8	12.7
	Time (h)	45.2	46.0	45.5	22.5	68.4	18.2	75.1
Stealth	Success (%)	13.2	13.0	13.1	14.5	10.1	15.1	8.5
	Time (h)	65.4	68.2	66.1	72.1	88.3	81.4	95.2

The data in Table 1 show a clear gradient. For the *Greedy* and *Max Probability* actors, the **Hard Shell** effect intensifies as k increases. At $k = 6$ that covers almost the entire path length, the Greedy actor's success rate collapses to 4.1%, confirming that front-loading resistance is highly effective in hierarchical networks against impatient adversaries. Conversely, the **Hard Core** columns show the "Greedy Paradox": as k increases, the Greedy actor's success rate *improves* and it reaches 19.5% at $k = 6$. This confirms that softening the perimeter enables aggressive actors to penetrate deeper and pivot. Instead, the *Stealth* actor suffers in the Hard Core scenario by dropping to 8.5% success. This is due to the noise accumulation at the end of the path that becomes prohibitive.

The Stealth Anomaly: A counter-intuitive result in Table 1 is the performance of the *Stealth* agent under the Hard Shell configuration. Unlike other strategies, its success rate increases (from 13.2% to 15.1% at $k = 6$). This occurs because the Hard Shell topology concentrates the low-CVSS (high-risk/high-noise) vulnerabilities at the perimeter. If the Stealth agent successfully bypasses this initial barrier, it enters an internal network populated by High-CVSS vulnerabilities. High scores often correlate with reliable, "clean" exploits (e.g., direct RCE vs. noisy brute-forcing). Consequently, the agent accumulates minimal additional noise during deep traversal, staying below the IDS threshold τ more effectively than in the Baseline, where noisy vulnerabilities are scattered throughout the path.

6.2 Topological Comparison: System A vs. System B

We compare the robustness of a Hierarchical Network (System A) against a Meshed Network (System B) for a fixed permutation depth of $k = 4$. We apply both the **Hard Shell** ($\Phi^{\uparrow}$) and **Hard Core** ($\Phi^{\downarrow}$) permutations to verify if the topological resilience of System B holds in both directions.

Table 2 shows a fundamental difference in resilience. In System A, the Hard Shell permutation reduces the Greedy actor's success rate by 45.6%. In System B, the same permutation only reduces the success rate by 12.9%. Furthermore,

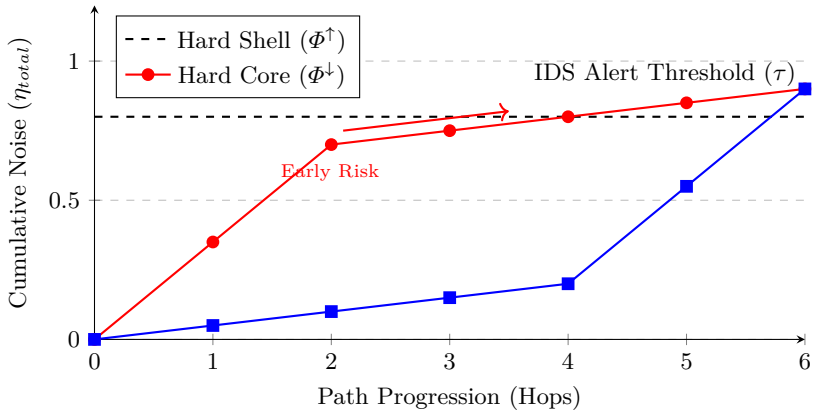

Fig. 5: Noise Accumulation Profiles for the Stealth Actor. Under Hard Shell (Red), high noise is generated early at the perimeter. Under Hard Core (Blue), the actor remains stealthy until deep in the network, delaying detection until containment is critical.

Table 2: Topological Sensitivity Comparison ($k = 4$). System A vs System B.

Permutation	Metric (Greedy)	System A	System B	Gap
Hard Shell	Baseline Succ.	11.4%	18.5%	-
	Permuted Succ.	6.2%	16.1%	-
	Change (%)	**-45.6%**	**-12.9%**	**32.7 pp**
Hard Core	Baseline Succ.	11.4%	18.5%	-
	Permuted Succ.	18.5%	20.2%	-
	Change (%)	**+62.2%**	**+9.1%**	**53.1 pp**

the **Hard Core** test reveals an even larger disparity. In System A, the Hard Core permutation causes a massive spike in Greedy success (+62.2%) due to the linear "sunk cost" trap. In System B, the increase is marginal (+9.1%). This confirms that meshed topologies are inherently resistant to positional hardening due to permutations in *either* direction. In System B, the high connectivity enables the actor to simply route around hardened nodes, whether at the perimeter or core, effectively dampening the impact of the permutation.

6.3 Path Stability Analysis

To verify whether the agents adapted their strategy by selecting alternative routes under permutation, we measured the *Path Retention Rate* (PRR). The

PRR quantifies the percentage of simulations where the agent implements the same sequence of actions in the Permuted environment as in the Baseline one.

Table 3: Path Retention Rate (PRR) under Hard Shell Permutation

Topology	**Path Retention Rate**
System A (Hierarchical)	99.2%
System B (Meshed)	97.4%

Table 3 shows that in System A, the path is almost topologically determined and the agent has virtually no alternatives. The retention is still high (97.4%) in System B but with a measurable increase in deviation. This indicates that in a small but significant number of cases, the agent can find an alternative route in the mesh topology when the primary path becomes too "expensive" due to the Hard Shell permutation.

7 Score Ordering and Success of Intrusions

Our results show the strong impact of score ordering along an attack path on intrusion outcomes. In particular, the order of the vulnerability scores in an attack path is as critical as their scores. This section explores some other implications of these results.

7.1 The Hierarchy of Risk Factors: Topology vs. Strategy

The juxtaposition of the Path Retention Rate (PRR) data (Table 3) and the Permutation Impact data (Table 1) reveals a hierarchical structure of risk factors.

1. **Topology Dictates the Path:** The extremely high PRR ($> 97\%$) shows that the network physical, logical connections and segmentation are the dominant constraints. Regardless of its strategy or of vulnerability scores, they force the agent into specific "chokepoints" with little liberty degrees.
2. **Complexity Distribution Dictates the Outcome:** While the path remains static, the *success* varies drastically under score permutation (e.g., Greedy success drops from 11.4% to 4.1%). This proves that the *order* of complexity (represented by CVSS scores) is more critical than the strategy itself in determining success. A "Hard Shell" profile stops an attack path regardless of whether the strategy is Greedy or Max Probability.
3. **Strategy Dictates the Cost:** The strategy mainly influence the *Time to Compromise* and the interaction with the IDS. For instance, while Greedy and Max Probability share similar success trends, their time profiles diverge. The Stealth strategy is the outlier, as it is the only one capable of exploiting the "Soft Core" of a Hard Shell topology to improve its success rate, effectively inverting the risk profile if compared to the other strategies.

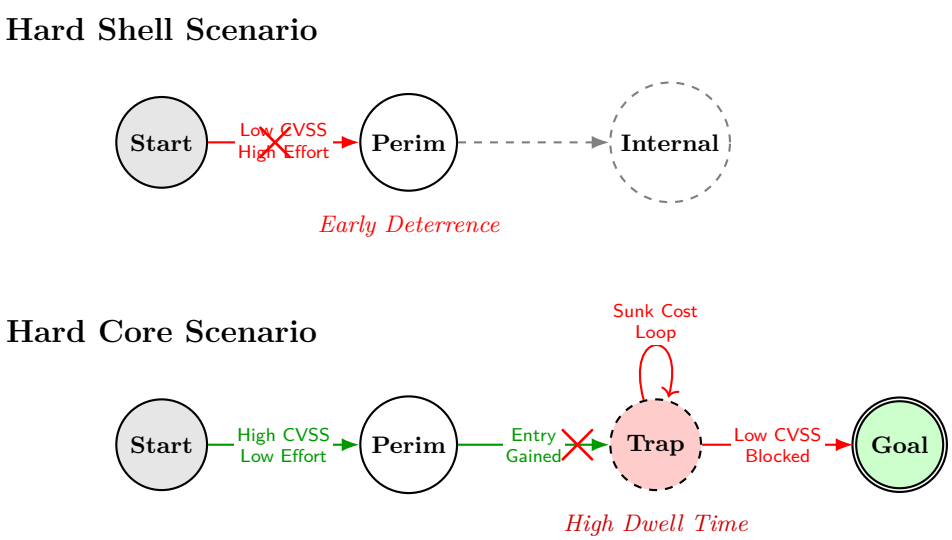

Fig. 6: Conceptual model of Adversary Dynamics. In the **Hard Shell** (top), the attacker is repelled at the perimeter. In the **Hard Core** (bottom), the attacker easily breaches the perimeter but gets trapped in a "Sunk Cost" loop, increasing time-on-target without success.

This hierarchy suggests that while "Zero Trust" (Physical Topology) is the ultimate goal, the immediate tactical risk is controlled by the *Vulnerability Topology* (Patch Management).

7.2 Adversarial Dynamics and the Sunk Cost Trap

The behavior of the *Greedy* actor under the *Hard Core* topology offers an insight into the impact of the strategy. One might expect that hardening core assets (lowering their CVSS) would universally improve security. Yet, according to our results, a soft perimeter (High CVSS) can be counter-productive for an aggressive, value-seeking attacker because by allowing the actor an easy entry, this topology encourages the accumulation of a "sunk cost." The actor establishes a foothold, bypasses the initial detection filters, and is then free to probe the internal network repeatedly. This results in a paradox where a system designed to protect the "crown jewels" but that neglects the perimeter, may actually simplify lateral movement. The actor, having easily breached the outer wall via a critical vulnerability, does not simply give up when it hits the hard core. Instead, it persists in the intermediate layers, potentially pivoting to secondary targets that were not the primary goal but represent real operational losses. This highlights that a defense strategy that relies too heavily on depth can sometimes fail to deter the initial breach, transforming a prevention problem into a much harder containment problem.

7.3 Early Detection vs. Deep Containment

The performance of the *Stealth* actor introduces a complex trade-off for defenders, which we term the "Stealth Tax." In a *Hard Shell* environment (Low CVSS at perimeter), the adversary is forced to invest its "noise budget" early to breach the perimeter, often resorting to noisier techniques to overcome the lack of easy exploits. This maximizes the probability of detection at the edge, where response actions (such as IP blocking) are simple and low-risk.

In contrast, a *Hard Core* environment enables the stealthy actor to traverse the network quietly via High CVSS (often clean RCE) exploits and it accumulates noise only when it is already deep inside the Data Center. While our results show that this topology eventually stops the actor and decreases the success rate, the operational implications of a failure are severe because if the adversary is detected only at the final steps, it is already inside the trusted zone. The "Tax" paid for this deep defense is the loss of early warning. Defenders should therefore choose either a *Hard Shell* strategy that faces constant perimeter noise but keeps the core clean, or a *Hard Core* strategy with a low number of alarms but with the risk of a catastrophic, late-stage detection.

7.4 Architectural Resilience: The Need for Zero Trust

The stark difference in sensitivity between the hierarchical System A and the meshed System B (Table 2) offers a simulation-based validation of a Zero Trust solution. In the hierarchical network, the permutation of CVSS scores had a decisive impact because the topology forced the adversary through specific chokepoints. The defender could effectively control the flow of the attack by manipulating specific nodes. Instead, in the meshed network, the impact of these permutations was significantly dampened (a drop of only 12.9% vs 45.6% for Hard Shell, and +9.1% vs +62.2% for Hard Core). The high connectivity allowed the actors to simply route around hardened (Low CVSS) nodes, finding "soft" alternative paths that bypassed the intended defenses. This confirms the dynamic path re-routing capability described in Section 4 and implies that in modern, flat network architecturessuch as cloud VPCs or Kubernetes clustersreliance on topological hardening is increasingly futile. Since the adversary can pivot dynamically, the concept of a "Hard Shell" evaporates. In such environments, security cannot be positional; it must be ubiquitous. This supports the shift towards Zero Trust solutions, where every interaction is treated as a perimeter crossing, effectively forcing a "Hard Shell" condition at every single hop, regardless of its location.

7.5 Path Hardening vs. Cut-Set Hardening

The Path Retention Rate (PRR) data (Table 3) informs the optimal remediation strategy.

- **System A:** With a PRR of 99.2%, the adversary is effectively locked into a single attack path. Here, defenders can confidently apply **Path Hardening**.

Remediation should be focused on the vulnerabilities along this static path. Breaking any link in this chain stops the attack with near certainty.
- **System B:** Although the PRR is high (97.4%), alternative paths (the remaining 2.6%) pose a strategic risk. If defenders simply patch the vulnerabilities on the primary path, the adversary will likely shift to these alternative routes. Therefore, in meshed environments, defenders should prefer **Cut-Set Hardening** not only to break the *current* path, but also to to prevent the adversary from pivoting to backup routes.

8 Conclusion, Limitations and Future Work

This paper has presented a formal and empirical analysis of positional sensitivity in CVSS-scored attack paths via Permutation-Based Sensitivity Analysis (PBSA) within a Security Twin. Proposition 1 and Proposition 2 (Section 3.3) establish the formal basis for why aggregate-based metrics are structurally blind to the effects we measure. Experimentally, we verified that the topological ordering of CVSS scores is as critical as the scores themselves in determining intrusion outcomes. We identified a "Greedy Paradox" where softening the perimeter to harden the core benefits aggressive adversaries, and we quantified the noise accumulation penalty that Hard Core topologies impose on Stealth actors.

8.1 Limitations and Threats to Validity

We explicitly acknowledge the following limitations, which delineate the scope of the present contribution.

L1 — Monotone CVSS-to-Probability Mapping. The linear approximation $P_{\text{succ}} = s/10$ (Section 3.1) abstracts away CVSS Temporal and Environmental Metrics, exploit availability, and runtime mitigations. While formally justified for the purposes of positional analysis (Proposition 2), its precision in absolute probability estimation is limited. Calibration against empirical exploit data (e.g., ExploitDB, EPSS scores) constitutes a priority for future work.

L2 — Perfect Attacker Knowledge. The current formulation assumes that the adversary has full observability of the CVSS scores and, by extension, of the exploitability of each edge. This is an idealization. In realistic threat scenarios, defenders may employ deception techniques or vulnerability obfuscation to present adversaries with a misleading attack surface. Future work will investigate PBSA under *partial observability*, where the perceived score sequence $\hat{\Sigma}$ diverges from the true sequence Σ.

L3 — Synthetic Topological Archetypes. The validation is conducted on two environments representing structural extremes (deep hierarchy vs. flat mesh). While this design choice enables controlled isolation of topological variables, it may not capture the sensitivity profile of hybrid architectures (e.g., partially segmented clouds, enterprise networks with legacy subnets). Generalization of the PBSA framework to empirically sampled enterprise topologies is a planned extension.

L4 — Static CVSS Semantics. CVSS Base Scores are static and do not evolve with the threat landscape. Dynamic scoring systems (e.g., EPSS, SSVC) that incorporate exploit activity and operational context may alter the sensitivity profiles observed here. The integration of dynamic scoring within the permutation operator Φ_k is a non-trivial extension that we defer to future work.

8.2 Future Work

Building on the formal framework established in Section 3.3, future work will expand the permutation operator Φ along three directions:

1. **Coupled Permutations:** Permuting both CVSS Score s and Detection Risk P_{det} simultaneously to model the trade-off between exploit reliability and operational stealth under IDS constraints.
2. **Dynamic Reconfiguration (Moving Target Defense):** Using the ST to apply "Virtual Permutations" in real-time by modifying VLAN assignments or firewall ACLs, implementing a Moving Target Defense (MTD) strategy grounded in the PBSA framework.
3. **Partial Observability and Deception:** Extending the adversary model to operate under misinformation, where the perceived score sequence diverges from the true one, to evaluate the effectiveness of honeypots and vulnerability masking as countermeasures to positional hardening strategies.

References

[1] The MITRE Corporation, *Cve*, 2024.
[2] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J. M. Wing, "Automated generation and analysis of attack graphs," in *Proc. 2002 IEEE Symp. on Security and Privacy*, IEEE, 2002, pp. 273–284.
[3] J. Carleton and S. Krishnamoorthi, *Automated breach and attack simulation: The cost and risk reduction revolution is here*, 2020.
[4] F. Baiardi and V. Sammartino, "From digital twins to ai agents: A synthetic data paradigm for next-generation cybersecurity," in *Artificial Intelligence in Cybersecurity: Unlocking the Power of Large Language Models*, CRC Press, 2026.
[5] C. Zhou, H. Yang, X. Duan, D. Lopez, A. Pastor, Q. Wu, M. Boucadair, and C. Jacquenet, "Network digital twin: Concepts and reference architecture," Internet Engineering Task Force, Internet-Draft draft-irtf-nmrg-network-digital-twin-arch-05, Mar. 2024, 25 pp.
[6] V. Sammartino, F. Baiardi, and S. Ruggieri, "A Security Twin to Defeat Intrusions in Cyber Physical Systems," in *ESREL SRA-E 2025*, 2025.
[7] V. Sammartino, "A framework for proactive cyber-resilience: Non-intrusive modeling for autonomous defense," in *DS-RT 2025*, 2025.
[8] L. Wang, S. Jajodia, A. Singhal, P. Cheng, and S. Noel, "Security metrics: A attack graph based approach," *Handbook of Information and Communication Security*, pp. 1–26, 2014.

[9] K. Kaynar, "A taxonomy and systematic review of data-driven attack graph generation and analysis," *Journal of Information Security and Applications*, vol. 27, pp. 1–21, 2016.

[10] F. Baiardi, S. Ruggieri, and V. Sammartino, "Anticipating Disasters through a Security Twin," in *SPRINGER OPTIMIZATION AND ITS APPLICATIONS - ARES 2024*, 2024.

[11] A. Applebaum, J. Baker, D. Beck, and M. Haase, "Attack flows beyond atomic behaviors," *MITRE Engenuity*, 2022.

[12] F. Baiardi, V. Sammartino, and S. Ruggieri, "Graph-based cyber defence using a security twin," in *29th International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*, 2025.

[13] F. Baiardi, S. Ruggieri, and V. Sammartino, "Security twins e il futuro della previsione di intrusioni cyber," *ICT Security*, 2025.

[14] F. Baiardi, S. Ruggieri, and V. Sammartino, " AI-enabled Cybersecurity using Synthetic Data," in *2025 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, Los Alamitos, CA, USA: IEEE Computer Society, Mar. 2025, pp. 140–145.

[15] F. Baiardi and V. Sammartino, "Quantifying resilience of cyber-physical systems to zero-day threats: A security twin-based what-if analysis framework," in *Proceedings of the 36th European Safety and Reliability Conference (ESREL 2026)*, European Safety and Reliability Association (ESRA), Braga, Portugal, Jun. 2026.

[16] F. Baiardi and F. Tonelli, "Twin based continuous patching to minimize cyber risk," *European Journal for Security Research*, vol. 6, pp. 211–227, Dec. 2021.

[17] F. Baiardi, V. Sammartino, and S. Ruggieri, "Notline: A non-intrusive automated platform to build a digital twin," in *2025 29th International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*, 2025, pp. 1–8.

[18] F. Baiardi and V. Sammartino, "Simulation-powered cybersecurity: Real-time risk assessment via non-intrusive security twin," *The Journal of Supercomputing*, 2026, Special Issue: Simulation-Powered Innovation: Driving the Future of Digital Ecosystems.

[19] F. Baiardi, V. Sammartino, and S. Ruggieri, "Evaluating adversary strategies through a security twin," in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2026.

[20] F. Baiardi and V. Sammartino, "A quantitative framework for the validation of twin-based cyber defense," in *37th European Modeling & Simulation Symposium (EMSS 2025), held within the 22nd International Multidisciplinary Modeling & Simulation Multiconference (I3M 2025)*, 2025.