Risk-aware proactive hardening of IoT and cyber-physical networks*

Tetiana Babenko[1,*,†], Kateryna Kolesnikova[1,†], Larysa Myrutenko[2,†], and Oleksandr Kruchinin[3,†]

[1] International Information Technology University, 34/1 Manas str., 050040 Almaty, Kazakhstan
[2] Taras Shevchenko National University of Kyiv, 64/13 Volodymyrska str., 01601 Kyiv, Ukraine
[3] National Technical University "Dnipro Polytechnic," 19 Dmytra Yavornytskoho ave., 49005 Dnipro, Ukraine

Abstract

Securing IoT deployments and cyber-physical systems grows more challenging as these infrastructures expand in scale and interconnection. This paper presents a framework for proactive network hardening that models the attack surface as a graph and uses this representation to guide defensive investments. The approach operates iteratively, cycling through network state assessment, attack technique prediction, risk calculation, and countermeasure selection. We integrate vulnerability data from the National Vulnerability Database with MITRE ATT&CK mappings, linking CVE identifiers through CWE weaknesses to adversary behaviors. Risk scores combine CVSS severity with contextual factors such as node criticality and zone placement. Experiments on simulated networks from 50 to 500 nodes demonstrate risk reductions between 15 and 31 percent, with computation times staying under 20 milliseconds for the largest topologies. The results suggest that combining patching with network segmentation outperforms single-strategy approaches, though optimal defense mixes vary with budget constraints.

Keywords

IoT security, cyber-physical systems, attack-surface graph, CVE, CVSS, cyber risk, proactive defense, network hardening, defense optimization

1. Introduction

The proliferation of IoT devices across critical infrastructure has fundamentally changed network security. What once meant protecting a handful of servers now involves thousands of sensors and embedded controllers scattered across environments never designed with cybersecurity in mind. These devices run outdated firmware, lack resources for robust security, and remain deployed for years without updates. The attack surface has exploded.

Cyber-physical systems add further complications. A compromised sensor in an industrial plant might seem insignificant, but it connects to controllers, supervisory systems, and eventually the corporate network. The Mirai botnet showed what happens when IoT vulnerabilities meet motivated adversaries [1], and campaigns against industrial control systems have demonstrated that attacker's chain minor footholds until they reach valuable targets [2]. Recent work on IoT botnet detection using hybrid neural architectures addresses the probabilistic nature of these threats [3], yet detection alone cannot solve expanding attack surfaces. Even agricultural systems now face cryptojacking risks [4].

Traditional security measures struggle here. Vulnerability scanners overwhelm teams with findings but offer little prioritization guidance. Automated asset discovery techniques have improved enumeration [5], and GNN-LSTM approaches show promise for attack vector reconstruction [6], but organizations still lack systematic ways to reason about where defensive investments have the greatest impact.

* CPITS 2026: Cybersecurity Providing in Information and Telecommunication Systems, February 28, 2026, Kyiv, Ukraine
* Corresponding author.
† These authors contributed equally.
babenko.tetiana.v@gmail.com (T. Babenko); kkolesnikova@iitu.edu.kz (K. Kolesnikova); myrutenko.lara@gmail.com (L. Myrutenko); kruchinin.o.v@nmu.ua (O. Kruchinin)
0000-0003-1184-9483 (T. Babenko); 0000-0002-9160-5982 (K. Kolesnikova); 0000-0003-1686-261X (L. Myrutenko); 0000-0001-5523-948X (O. Kruchinin)

This paper proposes a framework for proactive network hardening. Network topology is represented as a graph capturing assets and connectivity. Vulnerability data becomes actionable when linked to adversary behavior through CVE-CWE-ATT&CK mappings [7]. Defense selection is treated as an optimization problem where limited budgets must maximize risk reduction.

The framework operates iteratively: assessing the attack surface, predicting attack techniques, calculating risk, selecting countermeasures, and updating the model. Experiments on networks from 50 to 500 nodes show the framework scales well and outperforms single-category defense strategies.

Section 2 surveys related work. Section 3 presents the framework architecture. Section 4 reports experimental results and Section 5 concludes.

2. Related work

2.1. Vulnerability-based risk models

The Common Vulnerability Scoring System has become the standard for communicating vulnerability severity, providing scores between 0 and 10 based on attack characteristics and impact [8]. The National Vulnerability Database publishes CVSS scores alongside CVE identifiers, creating a repository that security tools routinely consume [9]. Despite widespread adoption, CVSS has recognized limitations for infrastructure-level decisions. The score reflects intrinsic vulnerability characteristics but ignores deployment context: a critical flaw in an internet-facing server poses far greater risk than the same flaw in an air gapped system, yet both receive identical ratings [10]. Spring and colleagues argued that CVSS conflates distinct concepts and proposed prioritization based on exploitation evidence [10]. The Exploit Prediction Scoring System addresses temporal factors by estimating exploitation probability [11], but it too operates at individual vulnerability level.

Another limitation concerns mapping vulnerabilities to adversary behavior. CVE identifiers indicate flaws exist but not how attackers use them. The Common Weakness Enumeration provides intermediate abstraction, categorizing vulnerabilities by underlying software flaw [12]. From CWE categories, one can link to MITRE ATT&CK techniques, bridging vulnerability data and threat intelligence [7].

2.2. Attack-surface and network-level security models

Recognizing that individual scores fail to capture systemic risk, researchers developed attack graphs to represent how adversaries chain vulnerabilities to reach targets [13, 14]. Sheyner and colleagues formalized attack graph generation using model checking [15], while Ou and colleagues introduced MulVAL, scaling to enterprise networks through logic-based representation [16]. The attack surface concept generalizes beyond specific chains to encompass all adversary interaction points. Manadhata and Wing proposed metrics for measuring attack surface size [17], though their framework focused on individual systems rather than networks.

Graph representations prove valuable for IoT environments because they capture heterogeneous connectivity patterns that perimeter-oriented models cannot accommodate. Figure 1 illustrates how an attack surface graph captures node types, zone boundaries, and compromise states within a unified model.

2.3. Defense selection and security control optimization

Given limited budgets, organizations must choose investments for maximum effect. Dewri and colleagues formulated network hardening as combinatorial optimization, seeking minimum-cost countermeasure sets [18]. Game-theoretic models consider attacker responses explicitly [19], though they require assumptions about attacker rationality that may not hold. Greedy algorithms trading optimality for tractability perform surprisingly well in practice, as Wang and colleagues demonstrated [20]. The intuition is that security investments exhibit diminishing returns: early countermeasures address severe risks while subsequent investments yield smaller improvements.

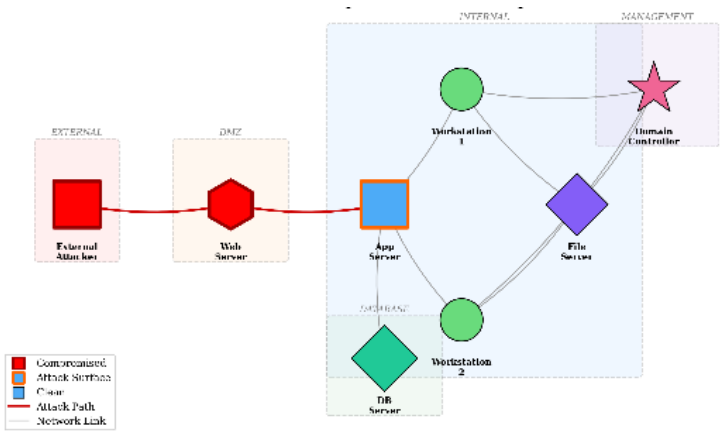

Figure 1. Attack surface graph illustrating node types, zone boundaries, and compromise states

2.4. Research gap

Despite progress in each area, significant gaps remain in their integration. Vulnerability models provide severity scores but ignore network context. Attack graphs capture topology but treat vulnerabilities as binary conditions without severity weighting. Defense optimization assumes quantifiable risk but relies on simplified models. Figure 2 illustrates the mapping chain from CVE through CWE to ATT&CK that enables integration, using Log4Shell as an example: CVE-2021-44228 maps to CWE-917, enabling techniques T1190 and T1059.

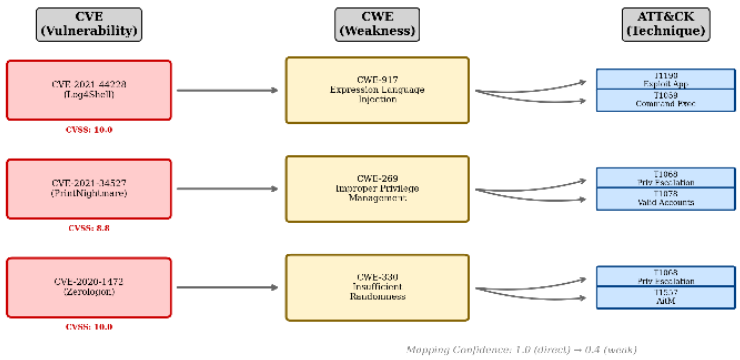

Figure 2: CVE ⟶ CWE ⟶ ATT&CK mapping chain linking vulnerabilities to adversary techniques.

This paper addresses the gap by combining calibrated risk scores, graph-based topology, and cost-aware defense selection into an integrated optimization framework.

2.5. Framework architecture

This section describes the engineering framework used to represent IoT and cyber-physical infrastructures, assess cyber risk, and select defensive actions. The approach integrates vulnerability data with network topology to produce actionable risk estimates that guide countermeasure deployment. Our design draws on established principles from attack graph analysis [13, 14] while extending them to address the specific characteristics of IoT environments.

2.6. System representation

The foundation of our approach is a graph-based model capturing the essential structure of networked infrastructure. Following the formalism established in network security analysis [19], we represent the system as a directed graph:

$$G=(V,E) \tag{1}$$

where V is the set of vertices corresponding to network assets and $E$is the set of edges representing communication links that could enable attack propagation.

Each vertex $v \in V$ represents an asset in the network and carries a small set of attributes that describe how it behaves and why it matters. The function $\tau(v)$ indicates the role of the asset, whether it acts as a sensor, a gateway, or a server of some kind, which reflects common IoT architectural patterns [20]. At the same time, each node is placed within a security zone $z(v)$, such as a field or control zone, since experience shows that trust boundaries tend to shape how attacks actually spread [21]. The attribute $a(v)$ captures the relative importance of the asset on a normalized scale and is later used when estimating the impact of compromise.

Edges represent connectivity through which attacks propagate. An edge $(u,v) \in E$ indicates an attacker controlling u could reach v through existing paths. Externally reachable nodes form the set $Vext \subseteq V$of potential entry points. This representation accommodates heterogeneous, multi-zone topologies characteristic of cyber-physical systems [22]. As illustrated in Figure 1, this captures node types, zone boundaries, and compromise states within a unified model.

2.7. Vulnerability and impact modeling

Each node v is annotated with a set of known vulnerabilities drawn from the National Vulnerability Database [9]:

$$V(v)=(CVE_i, CVSS_i) \vee i=1, ldots, n_v \tag{2}$$

where each entry pairs a CVE identifier with its associated$CVSS$ severity score [8]. Rather than treating vulnerabilities as binary conditions as in classical attack graph approaches [14], we retain severity information that influences downstream risk calculations. The aggregate vulnerability score for a node is computed as:

$$S_{vuln}(v)=max_i(\frac{CVSS_i}{10}) \tag{3}$$

This formulation follows recommendations for vulnerability prioritization that emphasize severity-based ranking [10]. Alternative aggregation functions such as weighted sums or probabilistic combinations are possible, though empirical studies suggest maximum-based approaches perform well for prioritization tasks [23].

Mapping vulnerabilities to adversary behavior requires connecting CVE identifiers to attack techniques through the Common Weakness Enumeration [12]. Each CVE maps to CWE categories describing the underlying software flaw, and these categories link to MITRE ATT&CK techniques [7]. This chain transforms static vulnerability lists into predictions about which attack techniques adversaries might employ.

Impact modeling assigns each node a value reflecting its operational Importance:

$$I(v)=w_{tau} \cdot tau(v)+w_a \cdot a(v) \tag{4}$$

where w_τ and w_a are weighting coefficients, $\tau(v)$ captures inherent criticality based on node function, and $a(v)$ represents organization-specific asset valuation. High-value components such as domain controllers and database servers contribute more to total risk than field sensors. This approach aligns with asset-centric security modeling that prioritizes protection based on business value [24].

2.8. Exposure and reachability

Vulnerability severity alone does not determine risk. A critical flaw in an unreachable system poses less danger than a moderate flaw in an exposed one. Exposure modeling captures this distinction by

considering network reachability from compromised nodes, building on reachability analysis techniques developed for attack graph construction [15].

Let $V_c \subseteq V$ denote the set of currently compromised nodes. We define reachability as a binary function indicating whether attack paths exist:

$$Reach(v) = \begin{cases} 1 & \exists u \in V_c : u \rightsquigarrow v \\ 0 & otherwise \end{cases} \tag{5}$$

Computing reachability efficiently requires careful algorithm design, particularly for large networks. We employ breadth-first traversal from compromised nodes, similar to approaches used in scalable attack graph tools [25]. The complexity remains linear in the number of edges, making the approach tractable for networks with hundreds of nodes.

Zone-based weighting reflects that nodes in less protected zones face greater exposure:

$$Z(v) = \frac{1}{1 + d(v)} \tag{6}$$

where $d(v)$represents the minimum number of zone boundaries between external entry points and node v. This formulation draws on defense-in-depth principles where multiple security layers provide cumulative protection [26]. The combined exposure score is:

$$E(v) = Z(v) \times Reach(v) \tag{7}$$

This models the attack surface dynamically rather than relying on static vulnerability counts. The attack surface metric proposed by Manadhata and Wing [16] similarly emphasizes that exposure depends on system configuration and connectivity rather than vulnerability presence alone.

2.9. Risk formulation

The risk associated with each node combines exposure, vulnerability severity, and potential impact:

$$R(v) = E(v) \times S_{vuln}(v) \times I(v) \tag{8}$$

This multiplicative formulation follows the standard risk equation where risk equals probability times impact [27]. The structure ensures that all three factors must be present for significant risk to emerge. A highly vulnerable node that is unreachable poses minimal risk, as does a reachable node with no known vulnerabilities

Total network risk aggregates across all nodes:

$$R(G) = \sum_{v \in V} R(v) \tag{9}$$

To prevent unrealistic zero-risk states, we incorporate a residual risk floor εrepresenting irreducible uncertainty:

$$R_{total}(G) = max(R(G), \varepsilon) \tag{10}$$

Figure 3 presents the risk matrix plotting attack probability against potential impact for techniques identified across a sample network. The quadrant structure follows standard risk management practice [28, 29], with techniques in the upper-right requiring immediate attention while those in the lower-left warrant monitoring rather than urgent action. Table 1 shows how different node types map to risk categories based on their typical exposure and asset value characteristics

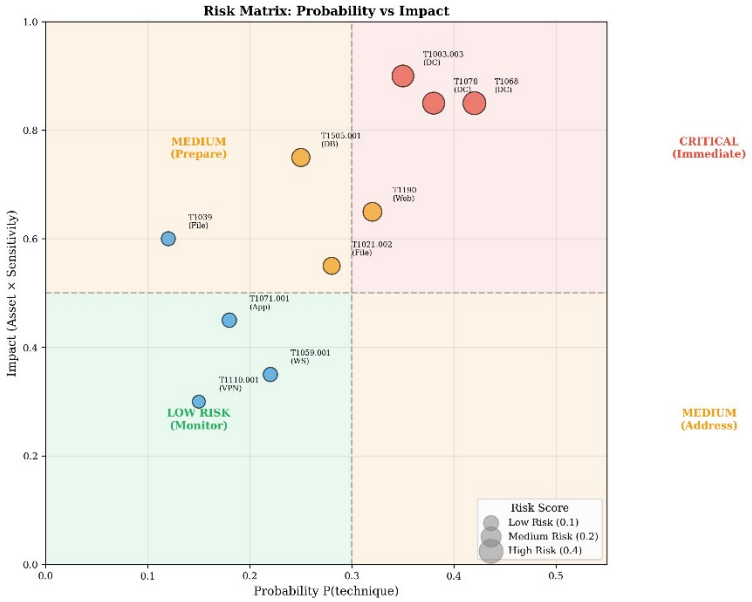

Figure 3: Risk matrix: probability vs impact

Table 1. Risk categorization by node type and zone placement.

Node Type	Typical Zone	Base Impact	Exposure Factor
External Gateway	DMZ	Medium	High
Web Server	DMZ	Medium	High
Application Server	Internal	High	Medium
Database Server	Data	Critical	Low
Domain Controller	Management	Critical	Low
Workstation	Internal	Low	Medium
Field Sensor	Field	Low	Variable

2.10. Defense optimization

Having established how risk propagates through the network model, we now turn to the question of what to do about it. The goal here is conceptually simple: given a fixed budget, select the combination of security controls that reduces aggregate risk as much as possible. In practice, this turns out to be surprisingly difficult because defenses interact in complex ways and the search space grows combinatorially with the size of the control catalog.

2.11. Defense catalog

We organize available security controls into a structured catalog where each defense d carries three pieces of information: its category, its deployment cost C_d, and its primary effect on the risk model. The cost values are normalized against a baseline security investment unit, which allows organizations to map these abstract numbers to their actual procurement and operational expenses. A cost of 0.15 might correspond to a few thousand dollars for a small organization or tens of thousands for a large enterprise, but the relative ordering stays consistent.

Table 2 shows a representative subset of the catalog we used in experimental evaluation. The full catalog contains additional controls, but these capture the main categories that matter for IoT and cyber physical environments. Some readers might notice that we have not included certain controls like security awareness training or physical access restrictions. This omission is deliberate. We focus on technical controls whose effects can be modeled quantitatively through the attack surface graph.

Human factors while undeniably important, require different modeling approaches that fall outside our current scope.

Table 2. Defense catalog with categories, costs, and primary effects

Defense ID	Category	Cost	Primary Effect
PATCH-CRIT	Patching	0.15	Vulnerability (Critical CVEs)
PATCH-HIGH	Patching	0.25	Vulnerability (High CVEs)
SEG-DMZ	Segmentation	0.25	Exposure (DMZ isolation)
SEG-INT	Segmentation	0.40	Exposure (Internal zones)
SEG-MICRO	Segmentation	0.80	Exposure (Microsegmentation)
MFA-ADMIN	Authentication	0.15	Attack success (Admin)
MFA-ALL	Authentication	0.40	Attack success (All users)
PAM	Access Control	0.60	Privilege escalation
CRED-GUARD	Credential Protection	0.25	Credential theft
EDR-ADV	Monitoring	0.45	Detection (Advanced)
SIEM	Monitoring	0.50	Detection (Network-wide)
IDS	Detection	0.35	Network intrusion

The way different defenses affect the risk model depends on their category. Patching directly reduces the vulnerability score $S_{vuln}(v)$by removing specific CVEs from the affected nodes. If a node has three critical vulnerabilities and we patch two of them, the vulnerability score drops accordingly. Segmentation works differently. It reduces exposure $E(v)$by either increasing the zone distance or outright removing reachability paths between network segments. Microsegmentation is expensive but powerful because it can isolate individual workloads rather than just network zones. Authentication controls like MFA reduce the probability that attackers successfully execute techniques requiring valid credentials. We model this as a multiplicative reduction in attack success probability. Monitoring defenses are trickier to model because they do not prevent attacks directly. Instead, they increase detection probability, which we represent as a reduction in effective risk under the assumption that detected attacks can be mitigated before causing full impact

2.12. Defense impact evaluation

Evaluating a candidate defense requires simulating its effect on the attack surface graph and then recomputing network risk using the formulation from Section 3. This sounds straightforward enough, but there is a subtlety that deserves attention. When we deploy a defense set $D \subseteq D_{\text{all}}$, the resulting risk at each node becomes:

$$R_D(v) = E_D(v) \times S_{\text{vuln,D}}(v) \times I(v) \tag{11}$$

The subscript D indicates that both exposure and vulnerability scores have been adjusted to reflect deployed defenses. *Impact* $I(v)$ remains unchanged because it represents inherent asset value, which defenses do not modify.

Now here is where things get interesting. Defense effects do not simply add up. If one control reduces risk by 30% and another reduces it by 40%, deploying both does not give you 70% reduction. The actual reduction is less because of overlap and redundancy. Two firewalls blocking the same traffic provide little additional benefit over one. We capture this using a multiplicative combination model:

$$\Delta R_{\text{combined}} = 1 - \prod_{d \in D} (1 - \Delta R_d) \tag{12}$$

This formulation comes from reliability engineering where it models redundant system components [30]. The intuition is that each defense independently has some probability of failing to prevent an attack. Only when all defenses fail does the attack succeed. Multiplying these failure probabilities gives the combined failure rate, and subtracting from one yields the combined success rate. This naturally produces diminishing returns, which matches what security practitioners observe in the field. The first few controls make a big difference, but piling on more eventually yields minimal additional benefit [19].

Figure 4 visualizes how different defenses impact specific ATT&CK techniques. The matrix format makes it easy to see coverage gaps. Patching handles exploitation techniques like T1190 quite well, often achieving 85% to 90% effectiveness. Segmentation addresses lateral movement but has less effect on initial access. Credential protection controls target a different cluster of techniques entirely. Looking at the matrix, one quickly realizes that no single defense category covers everything. Effective protection requires combining controls from multiple categories, which is exactly what the optimization in the next subsection attempts to achieve.

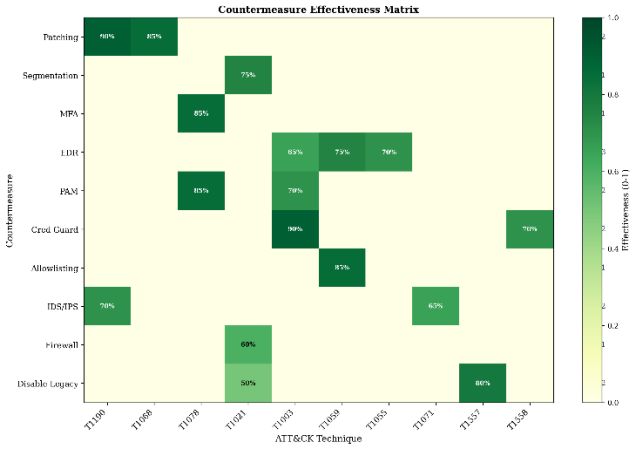

Figure 4: Countermeasure effectiveness matrix

2.13. Budget-constrained selection

With the impact model in hand, we can formulate defense selection as an optimization problem. The objective is to maximize total risk reduction subject to a budget constraint:

$$\max_{D' \supseteq D} \left(R(G) - R_{D'}(G) \right) \text{subject to} \sum_{d \in D'} C_d \leq B \tag{13}$$

where B represents the available security budget. This formulation should look familiar. It closely resembles the classic knapsack problem, which is known to be NP-hard [17]. For small catalogs and budgets, exact solutions are feasible through dynamic programming or integer linear programming. But as the catalog grows, exact methods become impractical.

We adopt a greedy approach that has proven effective in similar security resource allocation problems [31]. At each iteration, the algorithm evaluates all remaining candidate defenses and selects the one providing maximum marginal risk reduction per unit cost:

$$d^* = \arg\max_{d \in D \setminus D} \frac{R_D(G) - R_{D \cup d}(G)}{C_d} \tag{14}$$

The selected defense gets added to the deployed set, the budget decreases by its cost, and the process repeats. Termination occurs when any of three conditions holds: aggregate risk drops below some acceptable threshold R_{target}that the organization has defined, the budget B runs out, or the marginal improvement falls below a convergence threshold δ indicating that further investment yields negligible benefit.

Why does greedy work reasonably well here? The key insight is that risk reduction exhibits sub modularity, meaning the marginal benefit of adding a defense decreases as more defenses are already deployed. This property, which follows from the diminishing returns model in equation (12), guarantees that greedy achieves at least 63% of the optimal solution in the worst case [32]. In practice, we often observe much better performance because security investments tend to target distinct risk factors rather than overlapping heavily.

Figure 5 shows the complete iterative workflow. The cycle begins with POSITION, which constructs the attack surface graph from current network state. PREDICT identifies feasible attack techniques based on exposed vulnerabilities. RISK applies the calibrated model to produce prioritized scores. DEFEND runs the greedy selection procedure just described. Finally, UPDATE propagates the effects of deployed defenses back into the model, and the cycle repeats. This iterative structure allows the framework to adapt as the threat landscape evolves or as new vulnerabilities emerge.

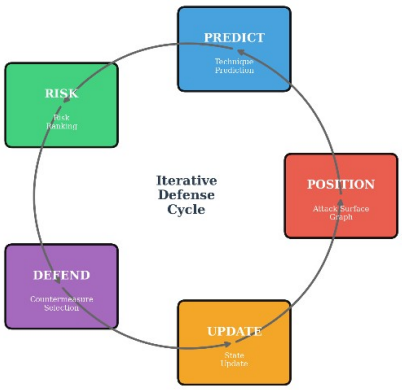

Figure 5: Iterative defense cycle

Figure 6 illustrates the Pareto frontier that emerges from this optimization. Each point on the frontier represents an efficient defense combination where no alternative configuration achieves better risk reduction at lower cost. Points below the frontier are dominated and should never be chosen. The greedy algorithm traces a path along this frontier, allowing organizations to see exactly what they get for each additional unit of investment. This visualization proves valuable in budget discussions because it quantifies the security implications of funding decisions

2.14. Efficiency considerations

Recomputing risk for a single defense candidate requires two operations: updating reachability in the attack surface graph and recalculating node-level risk scores. Graph traversal uses breadth-first search from compromised nodes, running in time $O(|V|+|E|)$for a graph with $|V|$ nodes and $|E|$ edges. Risk aggregation iterates over all nodes once, adding another $O(|V|)$ term. Evaluating the entire defense catalog costs $O(|D|\times(|V|+|E|))$ per greedy iteration.

The number of iterations is bounded by the smaller of $min(|D|)\frac{B}{min(C_d)}$ since each iteration either depletes the catalog or exhausts the budget. Overall complexity remains polynomial and scales linearly with network size when the defense catalog is fixed. For networks ranging from 50 to 500 nodes, optimization completes in tens of milliseconds on commodity hardware. This makes the framework suitable for interactive use.

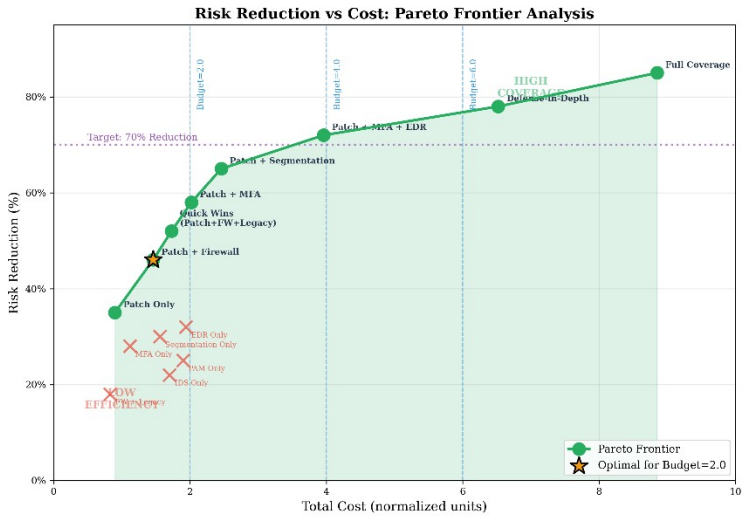

Figure 6: Pareto frontier. Risk reduction vs cost

2.15. Experimental setup

The synthetic networks range from 50 to 500 nodes, containing field sensors, gateways, application servers, database servers, and management nodes arranged across security zones mirroring typical industrial architectures [33—35]. Field devices sit at the edge, gateways connect operational technology to enterprise infrastructure, and critical servers occupy restricted internal zones. Vulnerability profiles use real CVE data from the National Vulnerability Database [9], spanning the full CVSS severity range.

Network generation followed Hahn and colleagues' approach for realistic industrial topologies [36]. Edge density varies by zone, with sparse field networks and higher enterprise connectivity. Attack entry points were placed at boundaries consistent with IoT threat models [37].

The defense catalog includes 25 countermeasures covering patching, segmentation, authentication hardening, and monitoring. The greedy optimizer used a 0.01 convergence threshold and maximum 10 iterations, though most runs terminated earlier. This follows optimization literature on balancing quality against computational cost [38].

Experiments ran on a laptop with Intel i7 and 16 GB memory. We avoided high-performance resources deliberately. Security analysts work on standard hardware, and a framework requiring cluster access would see limited adoption.

2.16. Scalability results

Table 3 summarizes how the framework scales with network size. The 50-node network completed optimization in under 4 milliseconds, achieving 31.5% risk reduction. The 100-node case took roughly the same time but produced a smaller reduction of 16%, which initially puzzled us until we examined the topology more closely. It turns out the larger network had vulnerabilities distributed more evenly across zones, leaving fewer high-impact targets for the optimizer to address. The 200-node and 500-node networks required 8 and 15 milliseconds respectively, both completing in three iterations.

Table 3. Scalability results across network sizes

Nodes	Runtime (ms)	Iterations	Initial Risk	Final Risk	Reduction (%)
50	3.4	2	2.07	1.42	31.5
100	4.0	2	2.92	2.46	16.0
200	8.3	3	3.48	2.82	19.0
500	15.4	3	4.26	3.35	21.3

What stands out here is that runtime grows sublinearly with network size. Doubling from 100 to 200 nodes roughly doubles the computation time, and going from 200 to 500 nodes less than doubles it again. This matches the complexity analysis from Section 4.4 and aligns with performance characteristics reported for similar graph-based security tools [39]. The risk reductions vary between 16% and 31%, which reflects differences in network topology and vulnerability distribution rather than any limitation of the optimization itself.

Figure 7 shows how aggregate risk decreases over optimization iterations for the 200-node network. The steepest drop occurs in the first iteration when high-impact defenses get selected. Subsequent iterations add refinements but with diminishing effect, exactly as the submodularity argument predicted. This convergence pattern resembles results from other greedy optimization approaches in network security [19].

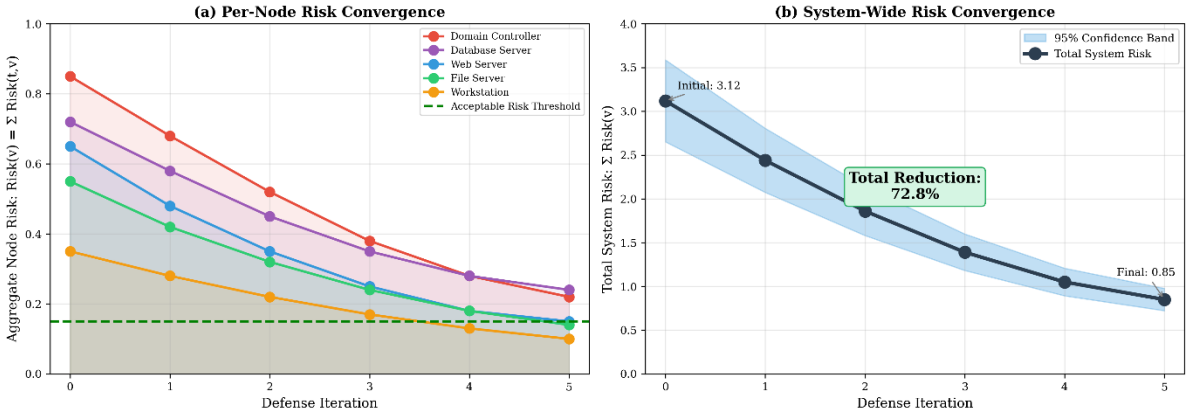

Figure 7. Aggregate risk convergence over optimization iterations for the 200-node network.

2.17. Budget sensitivity

A natural question is how much security improvement you can buy with different investment levels. We fixed the network at 100 nodes and varied the budget from 2.0 to 20.0 normalized cost units. Table 4 presents the results.

Table 4. Risk reduction under varying budget constraints.

Budget	Final Risk	Reduction	Defenses Deployed	Effectiveness
2.0	2.51	14.3%	2	0.285
5.0	2.46	16.0%	4	0.109
10.0	2.46	16.0%	7	0.051
20.0	2.46	16.0%	7	0.051

The pattern here is interesting. With only 2.0 budget units, the optimizer selects two defenses and achieves 14.3% reduction. Increasing the budget to 5.0 allows four defenses and pushes reduction to 16%. But here is the thing: doubling the budget again to 10.0 or even 20.0 produces no additional improvement. The optimizer deploys seven defenses in both cases and stops because marginal returns have dropped below the convergence threshold.

This plateau effect illustrates diminishing returns in concrete terms. After the most impactful countermeasures are deployed, additional investment yields progressively less benefit. The effectiveness metric in the rightmost column captures this directly: at budget 2.0, each unit of investment buys 0.285 units of risk reduction, but at budget 10.0 these drops to 0.051. Similar diminishing returns have been observed in empirical studies of security investment [40]. Security managers facing budget discussions might find this data useful. It suggests that for this particular network, investments beyond roughly 5.0 cost units provide minimal additional protection. Figure 8 visualizes the cost-effectiveness relationship, showing how marginal benefit decreases as cumulative investment increases.

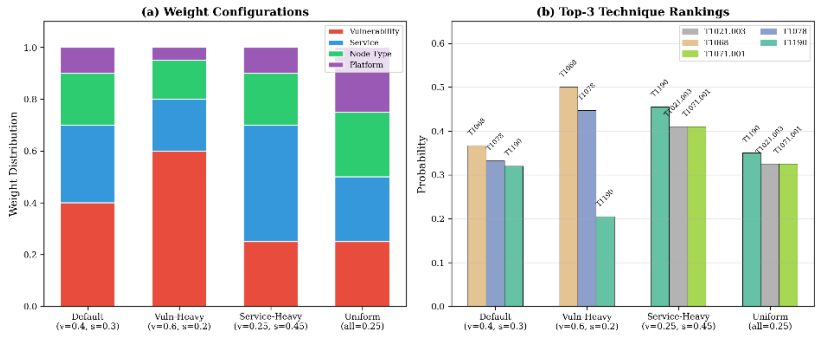

Figure 8. Cost-effectiveness relationship showing diminishing marginal returns with increasing investment.

The knee of the curve sits around budget 4 to 5, after which additional spending yields minimal improvement.

2.18. Sensitivity analysis

Risk models require parameter choices, and those choices carry some subjectivity. We wanted to know whether reasonable variations would produce wildly different recommendations. If they did, that would be a problem. Sensitivity analysis is standard in risk assessment [28, 29], though many security frameworks skip it entirely.

We tested six weight configurations. The baseline uses values from Section 3. Alternatives emphasized different factors: CVE severity, service exposure, prerequisite conditions. For each, we recorded top priority techniques and computed Kendall tau correlation against baseline rankings.

The results were reassuring. T1114.002 and T1190 consistently appeared at the top regardless of configuration. Kendall tau values showed strong correlation across rankings, meaning minor parameter adjustments did not scramble the prioritization.

This stability probably comes from the multiplicative structure. When exposure, vulnerability, and impact multiply together, extreme values in any factor tend to dominate. A node with critical vulnerabilities in an exposed zone scores highly whether vulnerability weight is 0.3 or 0.5. The model seems robust to reasonable parameter uncertainty, which matches findings from other multiplicative frameworks [27]. We would not claim it handles arbitrary parameterizations gracefully. But for values a practitioner might reasonably choose, outputs stay stable.

2.19. Discussion and limitations

The framework handled networks of several hundred nodes without major slowdowns, which matters for practical use. Risk reductions ranged from 15% to 31%. That might not sound dramatic, but in our conversations with practitioners, even the lower end of that range corresponds to fewer incidents per year when you apply it across real infrastructure. It adds up.

One thing the budget experiments made clear is how quickly returns diminish. Past a certain point, spending more does not help much. This is not a new observation. Security economics research has pointed out that optimal spending tends to be lower than people expect [41]. But seeing it quantified for a specific network makes the argument easier to have with leadership.

Now, the limitations. The model trusts its inputs, and inputs are never perfect. Scanners miss vulnerabilities. Topology diagrams go stale. Asset owners, well, they tend to believe their systems matter more than they do. We cannot fix that.

The defense catalog is another weak spot. We treat countermeasures as if they have fixed costs and predictable effects. Reality is messier. Secure development practices, for instance, clearly help prevent vulnerabilities [42], but quantifying that benefit within our model is hard. We have not solved it yet.

There is also the temporal problem. Networks change. Attackers adapt. Research has shown that attack patterns evolve [43], and someone watching your defenses might simply find a different way in. Game theory could help here, but it requires assumptions we are not comfortable making.

Finally, we tested on synthetic networks, not production systems. And we ignored the human side entirely. This framework is a tool, not an oracle.

3. Conclusion

This paper presented a framework for proactive network hardening that combines graph-based attack surface modeling with cost-aware defense optimization. The approach integrates vulnerability data from CVE through CWE to ATT&CK mappings, network topology for exposure calculations, and greedy optimization for budget-constrained defense selection.

Experiments on synthetic networks from 50 to 500 nodes showed the framework scales well, completing optimization in under 20 milliseconds. Risk reductions between 15% and 31% were achieved, with budget sensitivity analysis revealing strong diminishing returns that may help security managers justify investment decisions. The framework has limitations. It assumes accurate input data, operates on static snapshots, and ignores human factors. Future work should address uncertainty through Bayesian extensions, capture attacker adaptation via game-theoretic models, and integrate with continuous monitoring for near real-time operation.

The framework is not a replacement for security expertise but a tool that makes expert reasoning more systematic. When analysts understand why the model recommends specific defenses, they can apply judgment to refine those recommendations. This combination of algorithmic optimization and human oversight seems more promising than either approach alone.

Declaration on Generative AI

While preparing this work, the authors used the AI programs Grammarly Pro to correct text grammar and Strike Plagiarism to search for possible plagiarism. After using this tool, the authors reviewed and edited the content as needed and took full responsibility for the publication's content.

References

[1] M. Antonakakis et al., Understanding the Mirai Botnet, in: 26th USENIX Secur. Symposium, 2017, pp. 1093–1110, https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/antonakakis

[2] R. Lee, M. Assante, T. Conway, Analysis of the Cyber Attack on the Ukrainian Power Grid, E-ISAC and SANS ICS, Defense Use Case, March 18, 2016, https://nsarchive.gwu.edu/sites/default/files/documents/3891751/SANS-and-Electricity-Information-Sharing-and.pdf

[3] T. Babenko et al., Hybrid GNN-LSTM Architecture for Probabilistic IoT Botnet Detect. with Calibrated Risk Assess., Computers, vol. 15, no. 1, article 26, 2026, doi:10.3390/computers15010026

[4] T. Babenko, et al., Risk Assess. of Cryptojacking Attacks on Endpoint Systems: Threats to Sustainable Digital Agriculture, Sustainability, vol. 17, no. 12, article 5426, 2025

[5] T. Babenko, K. Kolesnikova, O. Abramkina, Y. Vitulyova, Automated OSINT Techniques for Digital Asset Discovery and Cyber Risk Assess., Computers, vol. 14, no. 10, article 430, 2025

[6] Y. Vitulyova, et al., A Hybrid Approach Using Graph Neural Networks and LSTM for Attack Vector Reconstruction, Computers, vol. 14, no. 8, article 301, 2025.

[7] B. Strom, A. Applebaum, D. Miller, K. Nickels, A. Pennington, C. Thomas, MITRE ATT&CK: Design and Philosophy, MITRE Technical Report, July 2018, https://apps.dtic.mil/sti/trecms/pdf/AD1108016.pdf

[8] FIRST, Common Vulnerability Scoring System v3.1: Specif. Document, Forum of Incident Response and Secur. Teams, June 2019, https://www.first.org/cvss/v3-1/specification-document

[9] Natl. Inst. of Standards and Technol., Natl. Vulnerability Database, 2024, Available: https://nvd.nist.gov/
[10] J. Spring, et al., Prioritizing Vulnerability Response: A Stakeholder-Specific Vulnerability Categorization, White Paper, Softw. Eng. Inst., Carnegie Mellon University, 2019, https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=636379
[11] J. Jacobs, et al., Exploit Prediction Scoring Syst. (EPSS), Digital Threats: Res. and Pract., vol. 2, no. 3, article 20, June 2021, doi:10.1145/3436242
[12] MITRE Corporation, Common Weakness Enumer.: A Community-Developed List of Softw. and Hardw. Weakness Types, https://cwe.mitre.org/
[13] C. Phillips, L. Swiler, A Graph-Based System for Network-Vulnerability Analysis, in Proc. New Secur. Paradigms Workshop (NSPW'98), Charlottesville, VA, Sept. 1998, pp. 71–79, doi:10.1145/310889.310919
[14] O. Sheyner, J. Haines, S. Jha, R. Lippmann, J. M. Wing, Automated Generation and Analysis of Attack Graphs, in Proc. IEEE Symposium on Secur. and Priv. (S&P'02), Oakland, CA, May 2002, pp. 273–284, https://www.cs.cmu.edu/~scenariograph/sheyner-wing02.pdf
[15] X. Ou, S. Govindavajhala, A. W. Appel, MulVAL: A Logic-Based Network Secur. Analyzer, in Proc. 14th USENIX Secur. Symposium, Baltimore, MD, July–Aug. 2005, pp. 113–128, https://www.usenix.org/legacy/event/sec05/tech/full_papers/ou/ou.pdf
[16] P. Manadhata, J. Wing, An Attack Surface Metric, IEEE Transactions on Softw. Eng., vol. 37, no. 3, pp. 371–386, May–June 2011. doi:10.1109/TSE.2010.60
[17] R. Dewri, N. Poolsappasit, I. Ray, D. Whitley, Optim. Secur. Hardening Using Multi-Objective Optimization on Attack Tree Models of Networks, in Proc. 14th ACM Conference on Computer and Communications Secur. (CCS'07), Alexandria, VA, Oct. 2007, pp. 204–213. doi:10.1145/1315245.1315272
[18] T. Alpcan, T. Başar, Netw. Secur.: A Decis. and Game-Theoretic Approach, Camb. University Press, 2010, ISBN: 978-0521119320, doi:10.1017/CBO9780511760778
[19] L. Wang, T. Islam, T. Long, A. Singhal, S. Jajodia, An Attack Graph-Based Probabilistic Secur. Metric, in Proc. 22nd Annual IFIP WG 11.3 Working Conference on Data and Applications Secur. (DBSec 2008), London, UK, July 2008, LNCS vol. 5094, pp. 283–296, doi:10.1007/978-3-540-70567-3_22
[20] P. Sethi, S. Sarangi, Internet of Things: Architectures, Protocols, and Applications, Journal of Electrical and Computer Eng., vol. 2017, Article ID 9324035, p. 25, 2017, doi:10.1155/2017/9324035
[21] A. Wool, A Quantitative Study of Firewall Configuration Errors, IEEE Computer, vol. 37, no. 6, pp. 62–67, June 2004, doi:10.1109/MC.2004.2
[22] E. Knapp, J. Langill, Industrial Network Security. Securing Critical Infrastructure Networks for Smart Grid, SCADA, and Other Industrial Control Systems, 2nd ed. Waltham, MA, USA: Syngress/Elsevier, 2015, doi:10.1016/C2013-0-06836-3
[23] L. Allodi, F. Massacci, Comparing Vulnerability Severity and Exploits Using Case-Control Studies, ACM Trans. Inf. Syst. Secur. (TISSEC), vol. 17, no. 1, Article 1, pp. 1–20, August 2014, doi:10.1145/2630069
[24] S. Bistarelli, F. Fioravanti, P. Peretti, Defense Trees for Econ. Evaluation of Secur. Investments, in Proc. ARES, 2006, pp. 416–423. doi:10.1109/ARES.2006.45
[25] K. Ingols, R. Lippmann, K. Piwowarski, Practical Attack Graph Generation for Network Defense, in Proc. ACSAC, 2006, pp. 121–130, doi:10.1109/ACSAC.2006.39
[26] NSA, Defense in Depth: A Practical Strategy for Achiev. Inf. Assurance in Today's Highly Networked Environments, Natl. Secur. Agency, March 2010. Available: https://csis.pace.edu/~lombardi/sciences/computer/systems/windows/docs/defenseindepth.pdf
[27] D. Hubbard, R. Seiersen, How to Measure Anything in Cybersecur. Risk, 2nd ed. Hoboken, NJ, USA: John Wiley & Sons, 2023
[28] ISO/IEC 27005:2018, Inf. Technol. — Secur. Techniques — Inf. Secur. Risk Manag., 3rd ed., International Organization for Standardization, Geneva, Switzerland, July 2018
[29] H. Hnatiienko, V. Hnatiienko, R. Zulunov, T. Babenko, L. Myrutenko, Method for Determining the Level of Criticality Elements when Ensuring the Functional Stability of the System

based on Role Analysis of Elements, in Proc. Workshop Cybersecur. Providing in Inf. and Telecommun. Systems (CPITS 2024), Kyiv, Ukraine, Feb. 2024, vol. 3654, pp. 301–311

[30] R. Billinton, R. Allan, Reliability Evaluation of Eng. Systems: Concepts and Techniqes, 2nd ed. New York, NY, USA: Plenum Press/Springer, 1992, doi:10.1007/978-1-4899-0685-4

[31] S. Noel, L. Wang, A. Singhal, and S. Jajodia, Measuring Secur. Risk of Networks Using Attack Graphs, International Journal of Next-Generation Computing (IJNGC), vol. 1, no. 1, pp. 135–147, July 2010, doi:10.47164/ijngc.v1i1.8

[32] G. Nemhauser, L. Wolsey, M. Fisher, An Analysis of Approximations for Maximizing Submodular Set Functions — I, Mathematical Programming, vol. 14, no. 1, pp. 265–294, December 1978, doi:10.1007/BF01588971

[33] IEC 62443, Secur. for Industrial Automation and Control Systems, International Electrotechnical Comm., Geneva, Switzerland, https://www.iec.ch/taxonomy/term/778

[34] E. Byres, J. Lowe, The Myths and Facts behind Cyber Secur. Risks for Industrial Control Systems, in Proc. VDE Kongress, vol. 116, Berlin, Germany, October 2004, pp. 213–218.

[35] S. McLaughlin et al., The Cybersecur. Landscape in Industrial Control Systems, Proc. of the IEEE, vol. 104, no. 5, pp. 1039–1057, May 2016. doi:10.1109/JPROC.2015.2512235

[36] A. Hahn, A. Ashok, S. Sridhar, M. Govindarasu, Cyber-Physical Secur. Testbeds: Architecture, Application, and Evaluation for Smart Grid, IEEE Trans. Smart Grid, vol. 4, no. 2, pp. 847–855, June 2013. doi:10.1109/TSG.2012.2226919

[37] S. Boyd, L. Vandenberghe, Convex Optimization. Cambridge, UK: Cambridge University Press, 2004, https://stanford.edu/~boyd/cvxbook/

[38] N. Idika, B. Bhargava, Extending Attack Graph-Based Secur. Metrics and Aggregating Their Application, IEEE Trans. Dependable Secure Comput., vol. 9, no. 1, pp. 75–85, January–February 2012. doi:0.1109/TDSC.2010.61

[39] L. Gordon, M. Loeb, The Econ. of Inf. Secur. Invest., *A*CM Trans. Inf. Syst. Secur. (TISSEC), vol. 5, no. 4, pp. 438–457, Novemb. 2002. doi:10.1145/581271.581274

[40] R. Anderson, T. Moore, The Econ. of Inf. Secur., Sci., vol. 314, no. 5799, pp. 610–613, Oct. 27, 2006. doi:10.1126/science.1130992

[41] V. Grechko, T. Babenko, L. Myrutenko, Secure Softw. Developing Recommendations, in Proc. IEEE International Scientific-Practical Conference: Problems of Infocommunications Sci. and Technol. (PIC S&T), 2019, pp. 1–5. doi:10.1109/PICST47496.2019.9061529

[42] T. Babenko, S. Toliupa, Y. Kovolova, LVQ Models of DDoS Attacks Identif., in Proc. 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Eng. (TCSET), 2018, pp. 1–4. doi:10.1109/TCSET.2018.8336253