The Illusion of Optimal Defense: Static Interdiction Under Adaptive and Persistent Attackers

Jeongkeun Shin[1], Siyuan Zhai[1], L. Richard Carley[1], Kathleen M. Carley[1]

[1] CASOS Center, Carnegie Mellon University, Pittsburgh, United States
jeongkes@cs.cmu.edu, siyuanzh@andrew.cmu.edu, lrc@andrew.cmu.edu, kathleen.carley@cs.cmu.edu

Abstract

As cyber threats evolve into sophisticated multi-stage campaigns, organizations increasingly rely on optimized yet static defense strategies to protect their networks. However, these static interdiction models often underestimate the capabilities of intelligent adversaries who learn and adapt to defensive barriers. This paper proposes an adaptive attacker model based on reinforcement learning to evaluate the robustness of static defense strategies derived from the Critical Node Problem (CNP). Unlike conventional simulations that assume predictable attack patterns, our agent leverages Q-learning to dynamically discover bypass trajectories while adhering to realistic operational constraints such as local persistence and entry commitment. We evaluate this adaptive agent against defense sets optimized via integer linear programming across various budget constraints. Experimental results demonstrate that the adaptive attacker agent consistently outperforms stochastic baselines by identifying efficient detour paths, revealing a significant security gap in static defense evaluations. Our findings underscore the need for dynamic defense mechanisms that account for the evolving intelligence of modern cyber threats.

Introduction

As the cybersecurity landscape rapidly evolves, cyberattacks have progressed from isolated incidents into sophisticated multi stage attack campaigns. Adversaries leverage complex chains of attack techniques to move laterally within corporate networks, ultimately aiming to achieve high impact objectives such as ransomware deployment. However, most organizations lack sufficient cybersecurity budgets to effectively counter these threats, leaving defenders with the critical challenge of deciding where and how to allocate limited security resources. In particular, designing defense strategies that can maximally disrupt attack pathways under constrained budgets has become a central problem in modern cybersecurity research.

In practice, many organizations either rely on security products provided by well-known vendors to address cyber threats or establish optimized defense strategies derived from one-time security consulting engagements. However, once deployed, such defense strategies are typically operated in a static manner without significant modification, which limits their effectiveness over time. This limitation is also observed in network interdiction studies that model cyber attack flows as networks and aim to identify theoretically optimal blocking points under constrained budgets.

Although existing approaches provide a strong baseline for defensive planning, the evaluation of defense strategies commonly relies on repeated simulations in which the attacker agent is assumed to start from scratch in every run without retaining any knowledge from past attack attempts. This implicitly assumes that attackers repeatedly follow static or predictable attack paths that have been frequently observed in the past. In contrast, real world attackers are learning and adaptive entities. When attacks fail due to a specific defense strategy, attackers adjust their behavior dynamically and explore alternative attack paths. Even in the absence of direct failure experiences, attackers actively seek alternative trajectories by leveraging shared knowledge within hacker groups or information obtained from online hacking communities such as dark web forums. This includes intelligence about zero-day vulnerabilities or weakly defended components witchin a target organization.

As a result, static defense strategies inherently have a limited period of effectiveness against dynamically adapting adversaries. Evaluating the effectiveness of defense strategies without accounting for such adaptive attack behavior may therefore lead to an overestimation of their true defensive impact.

To address this gap, this paper proposes an adaptive attacker model based on reinforcement learning (Sutton, Barto et al. 1998) to evaluate the robustness of static defense strategies. We specifically focus on how an adversary can learn to bypass theoretically optimal interdiction points by accumulating experience through repeated interactions with a defended network. By modeling the attack process as a Markov Decision Process (MDP) (Bellman 1957), we develop an agent that not only identifies successful trajectories but also optimizes its attack path under various budget constraints.

A key feature of our model is the implementation of realistic operational constraints such as local persistence and entry commitment. Unlike conventional agents that terminate upon detection, our agent persists through failed attempts and explores alternative detours. This behavior reflects the strategic persistence observed in Advanced Persistent Threat

(APT) groups. Furthermore, we evaluate this adaptive agent against a baseline of defense node sets derived using optimization algorithms. This allows us to measure the performance degradation of static defenses when they encounter an intelligent opponent.

The contributions of this study are as follows. First, we provide a framework for simulating adaptive attack behaviors that can bypass static interdiction strategies. Second, we demonstrate that traditional defense evaluations often overestimate security by ignoring the learning capabilities of adversaries. Finally, we analyze the trade-off between defense budgets and attack efficiency to suggest directions for more resilient and dynamic defensive planning. Through this work, we aim to bridge the gap between theoretical network interdiction and the practical necessity of adaptive cybersecurity.

Related Works

This section reviews prior studies that model cyberattack scenarios as networks and identify critical nodes within them, as well as research that employs reinforcement learning (Sutton, Barto et al. 1998) to model adaptive attacker agents in cybersecurity contexts.

Critical Node Problem in Cybersecurity

The Critical Node Problem (CNP) centers on identifying a subset of nodes whose removal or interdiction results in the maximum disruption of network connectivity (Lalou, Tahraoui, and Kheddouci 2018; Arulselvan et al. 2009). In the cybersecurity domain, this framework has been instrumental in modeling cyberattacks as graph-based structures, allowing network defense to be formulated as a formal minimization or optimization problem.

Early research leveraged the CNP to identify minimum blocking sets, defined as the smallest collection of nodes or edges whose removal eliminates all possible attack paths from an entry point to a target. For instance, Jha et al. used model checking to generate comprehensive attack paths and formulated the defense task as a minimum hitting set problem (Jha, Sheyner, and Wing 2002). While they established that this problem is NP-complete, they also introduced a greedy approximation algorithm to achieve polynomial-time efficiency. Building on this, Wang et al. explored cost-aware defense by enumerating hardening options in disjunctive normal form in order to identify the most economical configuration that secures all paths (Wang, Noel, and Jajodia 2006). To address the inherent exponential complexity of such enumerations, Islam and Wang proposed a heuristic approach that incorporates exploit dependencies to approximate near-optimal solutions more efficiently (Islam and Wang 2008).

Recent studies have shifted toward large-scale, empirical modeling of attack behaviors. Shin et al. constructed an attack flow network by consolidating the technical maneuvers of 19 ransomware groups and applied network centrality measures to pinpoint critical techniques within ransomware campaigns (Shin et al. 2025b). This work was subsequently expanded to include 220 attack groups and campaigns, utilizing optimization algorithms to determine the minimum set of technique nodes required to fully disrupt ransomware, data destruction, and financial theft operations (Shin, Carley, and Carley 2026). Furthermore, by framing defense planning as a budget-constrained interdiction problem, they identified scenario-specific technique sets that maximize defensive utility under restricted resource allocations.

Reinforcement Learning-Based Adversarial Attack Modeling

Recent advancements in cybersecurity have shifted from static rule-based models toward autonomous agents capable of dynamic decision-making. Reinforcement Learning (RL) (Sutton, Barto et al. 1998) has emerged as a predominant framework for this purpose, as it allows an agent to learn optimal attack trajectories through sequential interactions with a network environment, effectively capturing the adaptive nature of modern adversaries.

In the context of graph-structured data, Dai et al. formulated adversarial attacks as an RL problem, modeling the attacker as an adaptive agent that sequentially modifies the graph topology through edge perturbations (Dai et al. 2018). This approach demonstrated that agents can perform effective black-box attacks without prior access to gradient information. Extending this to a competitive setting, Wang et al. modeled the interaction between an attacker and a defender as a multi-agent DRL game, where the attacker learns jamming policies while the defender simultaneously retrains its policy to disrupt the adversary's learning process (Wang et al. 2020).

Beyond theoretical graph perturbations, RL has been extensively applied to simulate realistic cyberattack procedures. Oh et al. developed an adaptive deep RL agent that performs a variety of sequential actions, such as credential theft and spoofing, and adjusts its strategy based on environmental rewards rather than fixed scripts (Oh et al. 2024). Similarly, López-Montero et al. applied RL to autonomous penetration testing, where the agent balances the cost of actions with the discovery of vulnerabilities to learn an environment-aware policy (López-Montero et al. 2025). These studies collectively highlight that learning-based attackers can overcome the limitations of traditional, rule-based simulation methods by adapting to specific system configurations.

Despite these advancements, existing reinforcement learning based attack models primarily focus on abstract graph modifications or penetration testing simulations in small scale environments. In particular, there is a lack of studies that evaluate how such learning agents respond to defense strategies optimally derived from optimization theory on large scale empirical datasets. Our work addresses this gap by evaluating budget constrained optimal defense strategies designed for static flow interdiction against an adaptive reinforcement learning attacker trained on an empirically grounded attack flow network dataset.

Dataset

In this study, we use the Cyber Attack Flow Intelligence Network (CAFIN) dataset constructed and validated by Shin et

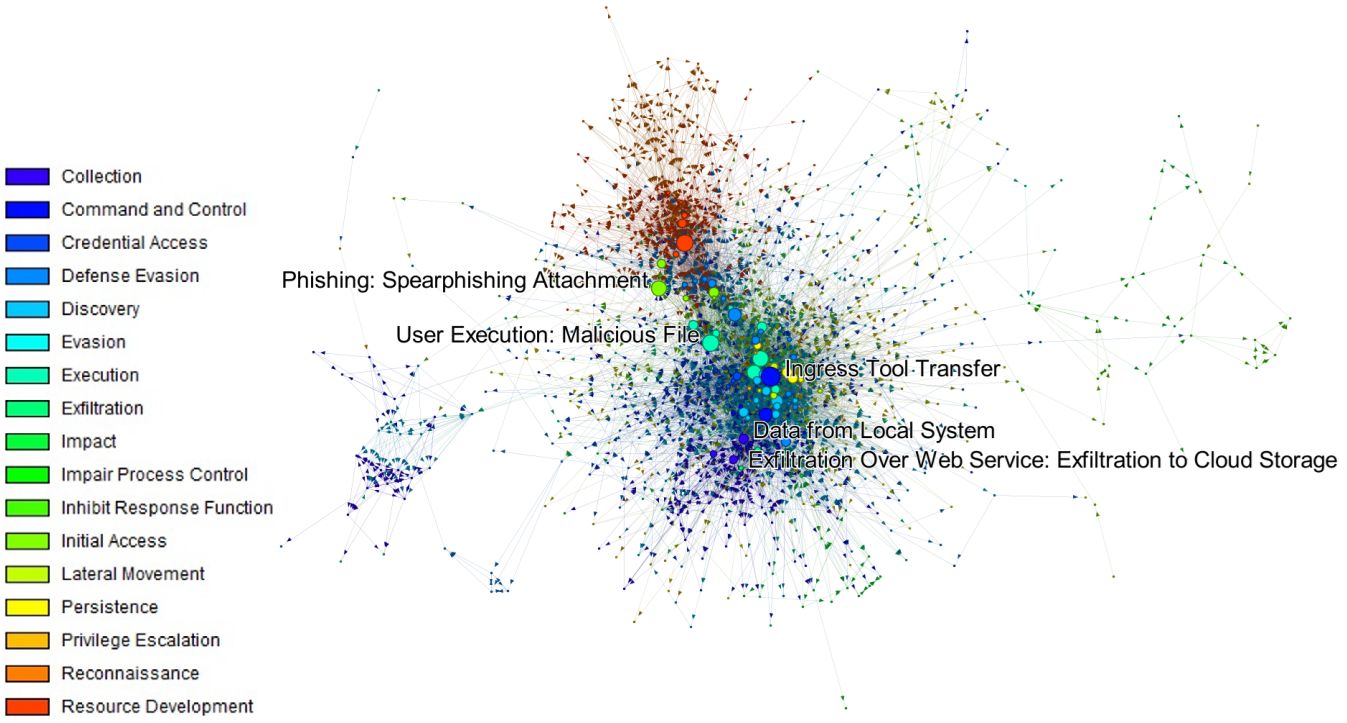


Figure 1: The visualization of Cyber Attack Flow Intelligence Network (CAFIN) (Shin, Carley, and Carley 2026).

al. (Shin, Carley, and Carley 2026). In CAFIN, nodes represent MITRE ATT&CK (Strom et al. 2018) techniques, while edges denote empirically observed transitions between techniques. The dataset integrates a total of 1,031 technical and analytical reports to provide a high-fidelity representation of adversarial attack behaviors observed in real-world environments. Based on these reports, the authors precisely modeled attack flows for 170 MITRE ATT&CK groups and 50 distinct cyberattack campaigns (Shin et al. 2025a) using the MITRE ATT&CK framework (Strom et al. 2018). By integrating these 220 cases, CAFIN effectively captures the complex transition relationships among MITRE ATT&CK techniques that occur in real attack scenarios.

The CAFIN network consists of 1,110 MITRE ATT&CK technique nodes and 6,653 links. In this study, the network is visualized using ORA (Carley 2018; Altman, Carley, and Reminga 2020), as shown in Figure 1.

The CAFIN dataset is used as the underlying graph for the attack simulation conducted in this work. By adopting this dataset, the adaptive attacker agent is trained and evaluated on a network that reflects historical attack patterns rather than theoretical or synthetic models. This empirical foundation enables a more rigorous evaluation of how defense strategies derived from the same framework perform against evolving attack trajectories.

Cyber Attack & Defense Scenario

This section delineates the ransomware attack campaigns conducted over the Cyber Attack Flow Intelligence Network (CAFIN) and the corresponding defensive strategies

Source Technique Node	
Category	**Technique Code**
Phishing	T1566 · T1566.001 · T1566.002 · T1566.003 · T1566.004
Valid Accounts	T1078 · T1078.001 · T1078.002 · T1078.003 · T1078.004
External Remote Services	T1133
Drive-by-Compromise	T1189
Target Technique Node	
Scenario	**Technique Code**
Ransomware	T1486

Table 1: Source-Target Technique Mapping for Ransomware Attack Scenario

categorized by budgetary constraints. Following the experimental setup established by Shin et al. in their study on CAFIN (Shin, Carley, and Carley 2026), the adversary initiates the intrusion through four primary vectors: **Phishing**, **Valid Accounts**, **External Remote Services**, and **Drive-by-Compromise**, as illustrated in Table 1. These vectors encompass 12 distinct MITRE ATT&CK techniques (Strom et al. 2018) that serve as entry points into the targeted organization. Following the initial breach, the attacker performs a series of continuous technical transitions guided by CAFIN. The ultimate objective of the campaign is to reach T1486 (Data Encrypted for Impact), a representative MITRE ATT&CK technique that signifies a successful ransomware attack.

We leverage the analysis results provided by Shin et

al. (Shin, Carley, and Carley 2026), that evaluated defense requirements for ransomware attack scenarios using the CAFIN dataset. To prevent attack paths originating from the 12 distinct initial access techniques described in Table 1 from reaching T1486, Shin et al. formulated the problem as a minimum multi-vertex cut problem (Dahlhaus et al. 1994). By solving this formulation using Integer Linear Programming (ILP) (Garg, Vazirani, and Yannakakis 1996), they determined that defending a minimum of 25 specific MITRE ATT&CK technique nodes is required to completely isolate T1486 from all entry vectors.

To further reflect the practical reality that organizations cannot defend all technique nodes simultaneously due to budget constraints, Shin et al. additionally formulated the defense planning problem as a budget-constrained interdiction problem. This optimization model aims to maximize the disconnection of (s, t) pairs while simultaneously increasing the shortest path length of any remaining attack trajectories. Given that fully preventing ransomware attacks requires defending 25 technique nodes, they considered defensive technique sets derived under restricted budget levels, increasing in increments of five from 5 to 20.

Ransomware Campaign Defense Strategy	
Budget	**Technique Code**
5	T1204 · T1204.002 · T1059.003 · T1656 · T1016
10	T1204 · T1204.001 · T1204.002 · T1059 · T1059.003 · T1059.007 · T1218.001 · T1656 · T1016 · T1102
15	T1190 · T1204 · T1204.001 · T1204.002 · T1059 · T1059.003 · T1059.007 · T1218.001 · T1027 · T1406 · T1656 · T1036 · T1036.001 · T1069.002 · T1102
20	T1190 · T1203 · T1204 · T1204.001 · T1204.002 · T1059 · T1059.001 · T1059.003 · T1059.007 · T1218.001 · T1027 · T1406 · T1656 · T1036 · T1036.001 · T1003 · T1069.002 · T1016 · T1570 · T1102
25	T1047 · T1059.003 · T1106 · T1484.001 · T1070.001 · T1070.003 · T1070.004 · T1112 · T1562.001 · T1016 · T1033 · T1082 · T1087.002 · T1217 · T1482 · T1021.001 · T1072 · T1570 · T1560.001 · T1071.001 · T1219 · T1567 · T1567.002 · T1485 · T1496.001

Table 2: Budget-dependent Defense Strategies Against Ransomware Attack Campaigns (Shin, Carley, and Carley 2026)

These budget-dependent defensive strategies form the basis for configuring which attack techniques are defended at each budget level, and are summarized in Table 2. In the subsequent simulations, we evaluate the performance of the adaptive attacker agent against these predefined interdiction sets to analyze the effectiveness of defense strategies under varying levels of security investment.

Methodology

Baseline Attack Simulation

Shin et al. designed a rule-based attack simulation grounded in CAFIN to evaluate the performance of optimal defense strategies derived under different budget levels (Shin, Carley, and Carley 2026). This simulation models a non-learning stochastic attacker that does not perform strategic optimization or learning, but instead navigates the network infrastructure according to empirically defined attack flow weights and probabilistic transition selection encoded in CAFIN. In this study, we adopt Shin et al.'s simulation procedure (Shin, Carley, and Carley 2026) presented in Algorithm 1 as the baseline to compare and evaluate the performance of our proposed adaptive attacker agent.

Algorithm 1: Baseline Attack Simulation Procedure

Require: Initial technique set S, target technique set T, defense-prepared set D, total attempt budget L_{total}, per-transition retry limit L_{each}, attack flow graph G

1: **Initialization:** Select a starting technique at random from S, and aim to reach one of the target techniques in T.If no directed path exists between the selected pair, terminate the simulation.
2: **Candidate Identification:** At the current technique, identify all outgoing transitions that can still reach the target and have not been excluded due to repeated failures.
3: **Probabilistic Transition Selection:** Select the next technique by sampling from the candidate transitions according to empirical attack flow weights.
4: **Defense Interaction:** If the selected transition leads to a defense-prepared technique, record the failure, remain at the current technique, and increase the retry counter for that transition. Transitions exceeding the retry limit L_{each} are excluded from further consideration.
5: **Backtracking:** If no feasible transitions remain at the current technique, backtrack to the previous technique and continue exploration.
6: **Termination:** The simulation terminates successfully upon reaching the target technique, or unsuccessfully when the total attempt budget is exhausted or all feasible paths are eliminated.

The simulation initiates by randomly selecting an entry point s from the initial technique set S. At each discrete step, the agent identifies a set of candidate transitions that maintain a valid directed path to the target set T, thereby ensuring basic reachability. The selection of the subsequent technique is governed by a probability distribution derived from the empirical attack flow weights in G.

To reflect the persistence of advanced cyber threats, the baseline simulation incorporates local retry and backtracking mechanisms. When a transition leads to a defended node $d \in D$, the agent remains at the current node and attempts to explore alternative outgoing paths. Repeated attempts toward the same defended node are allowed up to L_{each} times, after which the corresponding transition is permanently excluded from the action space of the current episode.

If all available outgoing transitions are eliminated after multiple failed attempts, the agent backtracks to the previous node to explore alternative attack paths. The simulation terminates successfully when the attacker reaches any node

in the target set T, and fails when the total attempt budget L_{total} is exhausted or no feasible attack paths remain.

Advanced Adaptive Attack Simulation

In this study, we model the cyber attack pathfinding problem as a Markov Decision Process (Bellman 1957) on a directed graph. Building upon the rule-based baseline simulation used by Shin et al. (Shin, Carley, and Carley 2026), we integrate Q-learning (Watkins and Dayan 1992) to model an adaptive attacker agent that explores an attack network in which defense mechanisms are applied to specific technique nodes (Algorithm 2).

Algorithm 2: Adaptive Attacker Simulation Procedure

Require: Directed graph G, start node set S, target node set T, blocked node set B, per-transition retry limit L_{each}, total step budget per episode L_{total}, number of episodes M, exploration rate ϵ

Initialize Q-table $Q(s, a)$ and super source node v_{super}
Add edges (v_{super}, s) for all $s \in S$
for episode $= 1$ to M **do**
  Initialize $path \leftarrow [v_{\text{super}}]$ and $E_{\text{excl}} \leftarrow \emptyset$
  Initialize failure counter $C(u, v) \leftarrow 0$ for all edges
  Set $v_{\text{curr}} \leftarrow v_{\text{super}}$ and $steps \leftarrow 0$
  while $v_{\text{curr}} \notin T$ **and** $steps < L_{\text{total}}$ **do**
    $steps \leftarrow steps + 1$
    Identify valid neighbors $\mathcal{N}$ of v_{curr} that are not excluded and can reach at least one node in T
    if $\mathcal{N}$ is empty **then**
      if $|path| > 2$ **then**
        Set $v_{\text{curr}} \leftarrow$ top of $path$
        Pop the top element from $path$
      else
        Terminate episode (failure)
      end if
    else
      Select $v_{\text{next}} \in \mathcal{N}$ using ϵ-greedy policy
      if $v_{\text{next}} \in B$ **then**
        $C(v_{\text{curr}}, v_{\text{next}}) \leftarrow C(v_{\text{curr}}, v_{\text{next}}) + 1$
        Update Q-table with reward R_{block}
        if $C(v_{\text{curr}}, v_{\text{next}}) \geq L_{\text{each}}$ **then**
          $E_{\text{excl}} \leftarrow E_{\text{excl}} \cup \{(v_{\text{curr}}, v_{\text{next}})\}$
        end if
      else
        Set $v_{\text{curr}} \leftarrow v_{\text{next}}$
        Push v_{curr} onto $path$
        Update Q-table with reward R_{success} if $v_{\text{curr}} \in T$, otherwise R_{step}
      end if
    end if
  end while
end for

Environment Modeling In this study, CAFIN (Shin, Carley, and Carley 2026) is used as a directed graph $G = (V, E)$. The set V represents MITRE ATT&CK (Strom et al. 2018) technique nodes, and E denotes the empirically observed transition frequencies between techniques. To represent the attacker's initial intrusion process, we introduce a super source node v_{super}. The twelve initial access technique nodes listed in Table 1 are defined as the set of entry points S. The graph is augmented with edges $\{(v_{super}, v) \mid v \in S\}$, and each added edge is assigned a weight of one.

The defender employs a static interdiction strategy determined by the available defense budget, which is represented as a set of blocked technique nodes $B \subset V$. The attacker has no prior knowledge of B and gradually discovers the blocked nodes through interactions with the network during the attack simulation.

Reinforcement Learning-based Attacker Model The agent operates within this environment defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$:

- **State Space ($\mathcal{S}$):** A state s_t is defined by the tuple (v_t, v_{target}), where v_t is the current node and v_{target} is the ultimate objective.
- **Action Space ($\mathcal{A}$):** The action a_t is the selection of a neighboring node $v_{next} \in Neighbors(v_t)$. To ensure reachability, the action space is pruned by excluding transitions that cannot reach the target node, such that only actions with at least one feasible path from v_{next} to v_{target} are considered.
- **Transition Function ($\mathcal{P}$):** Given the selected neighbor v_{next}, the transition is deterministic. If $v_{next} \in B$, the agent remains at v_t and the next state is (v_t, v_{target}). Otherwise, the agent moves to v_{next} and the next state becomes (v_{next}, v_{target}).
- **Reward Function ($\mathcal{R}$):** The agent receives feedback based on the outcome of the transition:

$$R(s_t, a_t) = \begin{cases} R_{success} & \text{if } v_{next} = v_{target} \\ R_{block} & \text{if } v_{next} \in B \\ R_{step} & \text{otherwise} \end{cases} \quad (1)$$

where $R_{success} \gg 0$ incentivizes goal achievement, $R_{block} < 0$ penalizes detection, and $R_{step} < 0$ encourages shorter paths.

Operational Constraints for Realistic Attack Modeling To ensure a fair comparison with the baseline simulation and to reflect realistic adversarial behaviors, the adaptive attacker agent operates under specific operational constraints regarding persistence and movement. These constraints govern how the agent handles defensive barriers and explores alternative paths.

1. **Collision:** If the agent selects a node $v \in B$, it receives R_{block}, stays at u, and updates its Q-value. The counter $C(u, v)$ is incremented.
2. **Exclusion:** If $C(u, v) \geq L_{each}$, the edge (u, v) is added to E_{excl}. Future action selection at node u filters out any v where $(u, v) \in E_{excl}$.
3. **Backtracking with Commitment:** If the feasible action space becomes empty due to exclusions, the agent backtracks to v_{t-1}. However, backtracking to v_{super} is prohibited to enforce **Entry Commitment**. This ensures that

once an entry node is chosen, the attacker must succeed or fail via that entry point.

Q-Learning Update Rule The agent updates its Q-table using the standard Q-learning (Watkins and Dayan 1992) update rule. Even upon encountering a blocked node, where the physical state does not change, the agent updates the Q-value for the attempted action to reflect the penalty and learn the optimal policy π^*:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\Big[R_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)\Big] \quad (2)$$

For blocked transitions, the "next state" for the Q-update is considered the current state v_t but with the reduced available actions, specifically excluding the blocked path.

Experimental Setup

Before presenting the experimental results, we describe the reward structure and training hyperparameters used for the adaptive attacker.

To guide the agent toward successful and efficient attack paths, we define a multi-objective reward function. The agent receives a substantial positive reward of $+1000$ upon reaching the target node ($T1486$). To discourage detection and encourage efficiency, a block penalty of $R_{block} = -50$ is applied when the agent attempts an interdicted transition, and a minor step cost of $R_{step} = -0.1$ is assigned for each movement. If the agent fails to reach the target within the step limit of 10,000 or exhausts all feasible paths, the episode terminates unsuccessfully without receiving the success reward.

The adaptive attacker agent is trained using a Q-learning algorithm (Watkins and Dayan 1992) over 1,000 episodes. An ϵ-greedy strategy (Sutton, Barto et al. 1998) is employed with an initial exploration rate of 1.0, decaying at a rate of 0.999 per episode to a minimum value of 0.05. The persistence limit L_{each} is set to 10, matching the baseline simulation to ensure a controlled and fair comparison.

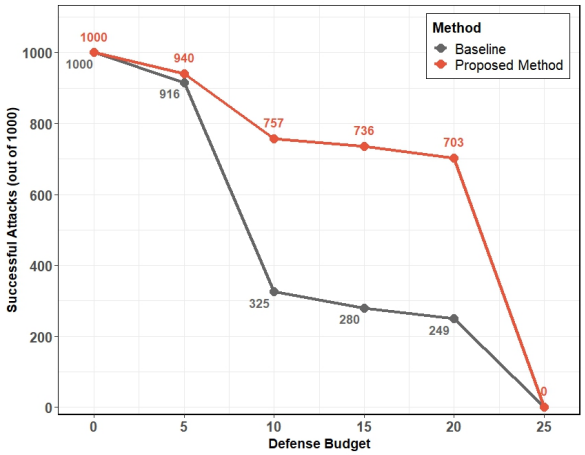
To comprehensively evaluate the resilience of the defense strategies and the adaptability of the attacker, we use the following four metrics:

- **Total Success Count:** We measure the total number of successful attacks across 1,000 simulations for each defense budget. This metric provides a macroscopic view of how effectively optimal static interdiction reduces the overall success rate of adaptive versus non-adaptive adversaries.
- **Training Success Dynamics:** To observe the learning efficiency, we track the success rate in 100-episode intervals. This allows us to analyze how quickly the attacker agent adapts to specific defensive barriers and converges toward a stable attack policy.
- **Average Transition Attempts (Success Only):** We record the average number of transition attempts for successful episodes. By comparing this between the baseline and the adaptive attacker agent, we evaluate the cost-effectiveness of the attack and the impact of the persistence mechanism on bypassing defenses.
- **Average Path Length Dynamics (Success Only):** We monitor the evolution of the average path length (number of hops) in successful cases throughout the training process. This metric indicates whether the agent is not only learning to succeed but also optimizing its trajectory to find the most efficient route to the target.

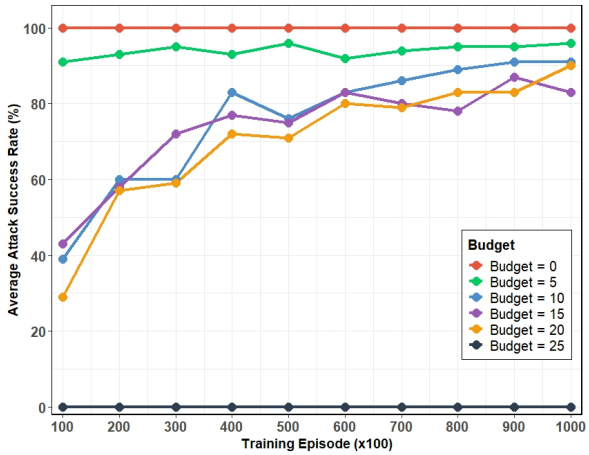
Results

This section describes the simulation results evaluating the adaptive attacker agent against static defense strategies. The results are summarized in Figure 4. Specifically, Figure 4a shows the Total Success Count, and Figure 4b presents the Training Success Dynamics. Figure 4c illustrates the Average Transition Attempts for successful cases, while Figure 4d displays the Average Path Length Dynamics for successful cases. For comparative analysis in Figure 4a and 4c, we utilize the baseline simulation results from Shin et al. which conducted 1,000 attack simulations under various defense budgets.

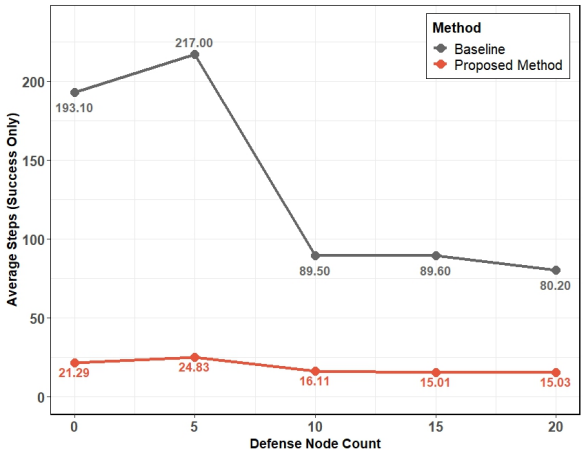
- **Effectiveness of Static Defense and Attacker Adaptability (Figure 2a)**: The results indicate that the attack success rate decreases for both models as the static defense budget increases. Both approaches reached a 0% success rate when 25 technical nodes were interdicted, which validates the fundamental effectiveness of the optimization-based defense. However, the proposed adaptive attacker agent recorded a significantly higher number of successful attacks across all other budget levels compared to the baseline. This suggests that the adaptive attacker can dynamically identify vulnerable detour paths within a static defense system. It also implies that existing static evaluation methods may underestimate the capabilities of intelligent adversaries.
- **Learning Convergence of Attack Success (Figure 2b)**: We measured the attack success rate every 100 episodes to observe the progress of the adaptive attacker. Excluding the case where the budget is zero, the success rate generally started at a low level and increased progressively as training continued. This outcome demonstrates that the agent learns from previous failures to find efficient routes for achieving ransomware objectives. It effectively reflects the behavior of Advanced Persistent Threat (APT) actors who refine their strategies based on historical interactions with a target environment.
- **Analysis of Attack Efficiency and Transition Attempts (Figure 2c)**: We analyzed the average number of transitions required to reach the destination in successful episodes. In baseline, the average transition attempts for the baseline decreased sharply near a budget of 10. According to Shin et al., this occurs because complex attack paths involving human vulnerabilities are completely blocked by the defense (Shin, Carley, and Carley 2026). As a result, only cases that exploit relatively simple system vulnerabilities are recorded as successful,



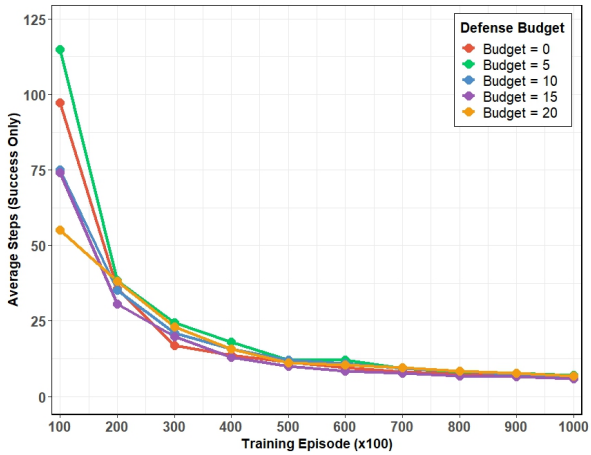
(a) Count of Successful Attacks Across 1000 Simulations



(b) Training Dynamics of Attack Success Rate



(c) Average Transition Attempts (Success Cases)



(d) Training Dynamics of Attack Path Length (Success Cases)

Figure 2: Simulation Results.

which leads to a lower average step count for the baseline. In contrast, our adaptive agent reached the target within 15 to 25 steps on average even in high-budget scenarios. This indicates that the adaptive attacker agent identifies more efficient routes compared to the baseline by establishing goal-oriented detour strategies instead of wandering randomly when encountering barriers.

- **Optimization of Attack Trajectories (Figure 2d)**: Finally, we measured the total attack path length in successful cases during the training process. Across all budget levels, the agent identified shorter and more efficient routes over time by learning from past mistakes. This trend proves that the attacker does not simply learn to succeed but also optimizes its trajectory to reach the objective quickly while minimizing exposure to detection.

In summary, the experimental results demonstrate that while optimal static interdiction significantly hinders stochastic attackers, it remains vulnerable to learning-based adversaries. The adaptive attacker agent does not simply increase its success rate through repetitive attempts. Instead, it systematically optimizes its attack trajectory by learning to bypass defensive barriers and identifying the most efficient pathways to the target. This adaptation process highlights a critical gap in current cybersecurity evaluations. Specifically, static metrics may provide a false sense of security by failing to account for the dynamic persistence and intelligence of modern cyber threats. Therefore, these findings underscore the necessity of developing more robust and adaptive defense mechanisms that can respond to evolving adversarial behaviors.

Discussion and Conclusion

This study analyzed the limitations of static network defense mechanisms through a reinforcement learning-based adaptive attacker agent. The experimental results demonstrate that while static defense strategies derived from optimization techniques are effective against non-adaptive attackers, they remain vulnerable to learning-based adversaries who

can identify detour paths. The adaptive agent not only identified defensive barriers through repeated attempts, but also optimized its own trajectory to maximize attack efficiency. These findings suggest that fixed defense configurations will eventually be exposed to intelligent attackers. Therefore, our results underscore the urgent need for a more dynamic and flexible defense paradigm.

In future work, we plan to extend this research in several directions to address the identified limitations. First, we will analyze optimal defense cycles and replacement strategies to disrupt the learning process of adaptive attackers within limited budget constraints. Second, we plan to develop an environment where adaptive attackers and defenders interact in real time to find optimal solutions using game-theoretic approaches such as Stackelberg games (Brückner and Scheffer 2011). Finally, we intend to move beyond binary defense models by integrating organizational simulations that probabilistically model both system and human vulnerabilities (Shin et al. 2025c). This will allow for the implementation of more realistic cybersecurity simulation environments. In conclusion, the adaptive attack modeling presented in this study provides a crucial foundation for designing and validating next-generation dynamic defense systems.

Acknowledgments

This research was supported in part by the Minerva Research Initiative under Grant #N00014-21-1-4012 and by the Center for Computational Analysis of Social and Organizational Systems (CASOS) at Carnegie Mellon University. The views and conclusions are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Office of Naval Research or the US Government.

References

Altman, N.; Carley, K. M.; and Reminga, J. 2020. ORA user's guide 2020. *Carnegie-Mellon Univ. Pittsburgh PA Inst of Software Research International, Tech. Rep*, 2: 2.

Arulselvan, A.; Commander, C. W.; Elefteriadou, L.; and Pardalos, P. M. 2009. Detecting critical nodes in sparse graphs. *Computers & Operations Research*, 36(7): 2193–2200.

Bellman, R. 1957. A Markovian decision process. *Journal of mathematics and mechanics*, 679–684.

Brückner, M.; and Scheffer, T. 2011. Stackelberg games for adversarial prediction problems. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 547–555.

Carley, K. M. 2018. ORA: a toolkit for dynamic network analysis and visualization. In *Encyclopedia of social network analysis and mining*, 1693–1702. Springer.

Dahlhaus, E.; Johnson, D. S.; Papadimitriou, C. H.; Seymour, P. D.; and Yannakakis, M. 1994. The complexity of multiterminal cuts. *SIAM Journal on Computing*, 23(4): 864–894.

Dai, H.; Li, H.; Tian, T.; Huang, X.; Wang, L.; Zhu, J.; and Song, L. 2018. Adversarial attack on graph structured data. In *International conference on machine learning*, 1115–1124. PMLR.

Garg, N.; Vazirani, V. V.; and Yannakakis, M. 1996. Approximate max-flow min-(multi) cut theorems and their applications. *SIAM Journal on Computing*, 25(2): 235–251.

Islam, T.; and Wang, L. 2008. A heuristic approach to minimum-cost network hardening using attack graph. In *2008 New Technologies, Mobility and Security*, 1–5. IEEE.

Jha, S.; Sheyner, O.; and Wing, J. 2002. Two formal analyses of attack graphs. In *Proceedings 15th IEEE Computer Security Foundations Workshop. CSFW-15*, 49–63. IEEE.

Lalou, M.; Tahraoui, M. A.; and Kheddouci, H. 2018. The critical node detection problem in networks: A survey. *Computer Science Review*, 28: 92–117.

López-Montero, D.; Álvarez-Aldana, J. L.; Morales-Martínez, A.; Gil-López, M.; and García, J. M. A. 2025. Reinforcement learning for automated cybersecurity penetration testing. *arXiv preprint arXiv:2507.02969*.

Oh, S. H.; Kim, J.; Nah, J. H.; and Park, J. 2024. Employing deep reinforcement learning to cyber-attack simulation for enhancing cybersecurity. *Electronics*, 13(3): 555.

Shin, J.; Carley, L. R.; and Carley, K. M. 2026. Cyber Attack Flow Intelligence Network: Backbone Analysis and Defense Strategy Optimization. Manuscript under revision at IEEE Transactions on Network Science and Engineering.

Shin, J.; Zhai, S.; Carley, L. R.; and Carley, K. M. 2025a. Attack Flow Network Models of MITRE ATT&CK Groups and Campaigns. Technical report, CASOS. Technical Report, forthcoming.

Shin, J.; Zhai, S.; Carley, L. R.; and Carley, K. M. 2025b. Network Analysis of Attack Flows in Ransomware Groups and Campaigns. In *International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation*, 66–75. Springer.

Shin, J.; Zhai, S.; Carley, L. R.; and Carley, K. M. 2025c. Simulating cyber defense: the impact of phishing training and system updates on mitigating damage from hybrid phishing and watering hole attacks. *The Journal of Defense Modeling and Simulation*, 15485129251365259.

Strom, B. E.; Applebaum, A.; Miller, D. P.; Nickels, K. C.; Pennington, A. G.; and Thomas, C. B. 2018. MITRE ATT&CK: Design and philosophy. In *Technical report*. The MITRE Corporation.

Sutton, R. S.; Barto, A. G.; et al. 1998. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge.

Wang, F.; Zhong, C.; Gursoy, M. C.; and Velipasalar, S. 2020. Adversarial jamming attacks and defense strategies via adaptive deep reinforcement learning. *arXiv preprint arXiv:2007.06055*.

Wang, L.; Noel, S.; and Jajodia, S. 2006. Minimum-cost network hardening using attack graphs. *Computer Communications*, 29(18): 3812–3824.

Watkins, C. J.; and Dayan, P. 1992. Q-learning. *Machine learning*, 8(3): 279–292.