MEGA-PT: A Meta-game Framework for Agile Penetration Testing

Yunfei Ge(✉) and Quanyan Zhu

New York University, New York, NY 11201, USA
{yg2047,qz494}@nyu.edu

Abstract. Penetration testing is an essential means of proactive defense in the face of escalating cybersecurity incidents. Traditional manual penetration testing methods are time-consuming, resource-intensive, and prone to human errors. Current trends in automated penetration testing are also impractical, facing significant challenges such as the curse of dimensionality, scalability issues, and lack of adaptability to network changes. To address these issues, we propose MEGA-PT, a meta-game penetration testing framework, featuring micro tactic games for node-level local interactions and a macro strategy process for network-wide attack chains. The micro- and macro-level modeling enables distributed, adaptive, collaborative, and fast penetration testing. MEGA-PT offers agile solutions for various security schemes, including optimal local penetration plans, purple teaming solutions, and risk assessment, providing fundamental principles to guide future automated penetration testing. Our experiments demonstrate the effectiveness and agility of our model by providing improved defense strategies and adaptability to changes at both local and network levels.

Keywords: Penetration Testing · Cyber Security · Meta-Game · Cyber Risk Assessment · Agile Defense

1 Introduction

With the exponential growth of network technologies and the escalating frequency of security incidents, cybersecurity has become a global concern [2,9]. In response to these challenges, penetration testing has emerged as a crucial solution for uncovering system vulnerabilities and assessing network security through authorized ethical attacks [4]. However, traditional manual penetration testing performed by skilled IT professionals has several limitations. It can be time-consuming, resource-intensive, and prone to human error. Relying solely on manual testing often falls short of identifying all vulnerabilities within the system. Thus, there is a need for automation and the integration of advanced threat intelligence into the penetration testing process, enabling a more efficient and scalable approach to enhancing cybersecurity.

Current proposed automated penetration testing methods are increasingly becoming non-standard, complex, and resource-consuming, despite tool advancements. Reinforcement learning (RL) or Markov Decision Process (MDP) based

A. Sinha et al. (Eds.): GameSec 2024, LNCS 14908, pp. 24–44, 2025.
https://doi.org/10.1007/978-3-031-74835-6_2

methods [3,4] suffer from the curse of dimensionality, as they define the state space as the collection of all known information for each machine on the network. Partially Observable Markov Decision Process (POMDP) methods [8] face scalability issues, making it unfeasible to model and solve for large networks. Additionally, these methods lack adaptability to changes, as they assume the network structure and software configuration remain unchanged to learn the optimal policy. Many proposed models do not follow the Tactics Techniques and Procedures (TTPs) in real cybersecurity practice, relying mainly on hypotheses and simulations, which undermines their transition to praxis. Furthermore, merely identifying vulnerabilities through penetration testing is insufficient; it is crucial to provide defense suggestions and risk analysis based on the testing to enhance overall security.

To address the limitations of current penetration testing methods, we propose a meta-game-based automated penetration testing framework (MEGA-PT). In this framework, the micro tactic game captures the interactions between the defender and attacker at each local node, while the macro strategy process models lateral movement and the attack chain across the entire network. This approach offers several key features: practical implications, as the sequential interactions in each micro tactic game follow the MITRE ATT&CK framework [7] and use extensive-form games to model attack/defense dynamics; distributed penetration testing, with modularized processes at each micro tactic game allowing for parallel computation; and adaptability to changes at both the local and network levels, ensuring efficient testing and scalability.

Our proposed model enables various security schemes, depending on the solution concept selected for the meta-game. This extension of penetration testing goes beyond vulnerability discovery to include defense strategy recommendations and risk analysis. Specifically, the model provides solutions for the following security schemes: optimal local penetration plans under certain defense strategies, purple teaming solutions for enhanced defense suggestions, and risk assessments at equilibrium. Our contributions can be summarized as follows:

1. We propose a meta-security game framework MEGA-PT for automated penetration testing, where micro tactic games at each local node are modeled as extensive-form games, and the macro strategy process is modeled as a Markov decision process.
2. We offer applicable solution concepts for security schemes aimed at vulnerability discovery, defense suggestion, and risk analysis.
3. Our experiments demonstrate the effectiveness of MEGA-PT by providing improved defense strategies and adaptability to changes at both local and network levels.
4. In essence, MEGA-PT establishes fundamental principles to drive the future of automated penetration testing and its practices.

2 Problem Formulation

Penetration testing is an ethical attack aimed at identifying system vulnerabilities, providing defense suggestions, and offering risk assessments. In this context,

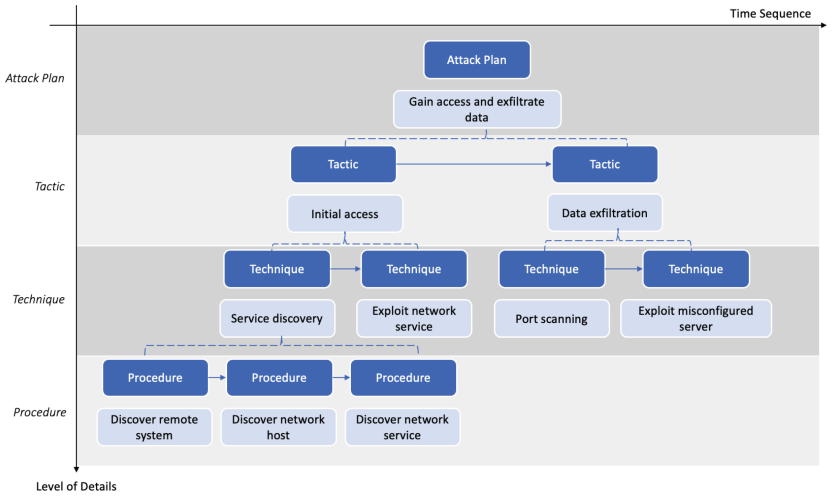


Fig. 1. Attack plan and tactics, techniques, and procedures (TTPs). Depending on the level of detail, each attack plan can be elaborated by a sequence of tactics (from left to right), where each tactic is composed of a sequence of techniques, and each technique can be described by a sequence of procedures.

the term *attacker* refers to the penetration testing agent, while the *defender* represents the system security management engine. To describe attacker behaviors within a security program, Tactics, Techniques, and Procedures (TTPs) are commonly used. Figure 1 illustrates the hierarchy between these terms. For security strategy analysis, we focus on Tactics and Techniques in penetration testing, omitting the detailed Procedures.

To describe interactions in penetration testing using TTPs, we propose a meta-security game over a network graph. The macro strategic game represents strategic attack activities between nodes, while the micro tactic game details tactic-level attack procedures on each local node. Let the directed graph $G = \langle \mathcal{V}, \mathcal{E} \rangle$ represent the target network topology, where $\mathcal{V}$ is a set of nodes (e.g., server, database, device), and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of directed edges representing connections (e.g., SSH, RDP, cloud services) from node u to node v. Self-loops are allowed as they indicate continued exploration of the same node. Let $v^0 \in \mathcal{V}$ be the initial foothold in the system. Figure 2 shows an example of the networked system topology. The penetration tester, as an ethical attacker, aims to explore available information, exploit discovered vulnerabilities, and influence critical assets inside the network.

2.1 Micro Tactic Games

Game-Theoretic Modeling

When an attacker gains access to one node, there are multiple steps involved before he completes the exploration and exploitation process on the node.

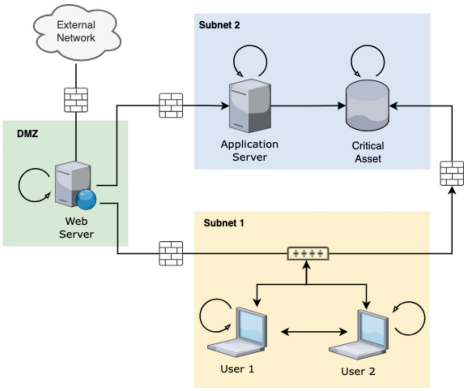


Fig. 2. Illustration of the networked system topology. The system contains 5 nodes (web server, application server, 2 user devices, and critical asset). The penetration testing starts from the web server, which is open to the external network.

To model the sequential moves at each local node, we use the concept of an extensive-form game tree to explicitly and visually represent the sequential moves, possible outcomes, and information available at each decision point in a strategic interaction. Figure 3 illustrates an example of a game tree at the web server. The attacker can choose to perform reconnaissance on hosts and services on the web server and then exploit the host to perform privilege escalation. The defender can choose to accept or deny the access request based on their defense policy. Depending on the privilege levels, the attacker could collect different credentials in the game, leading to various expected tactic outcomes and connecting to different nodes in the network.

Definition 1 (Micro Tactic Game (MTG)). *The Micro Game of the MEGA-PT is defined by a set of Micro Tactic Games (MTG)* $\{\Gamma^v\}_{v\in\mathcal{V}}$ *where* $\mathcal{V}$ *is the set of nodes in the system. Given a node* $v \in \mathcal{V}$ *in the network, the MTG on node* v *can be represented by an extensive-form game tuple* $\Gamma^v = \langle \mathcal{N}\cup\{c\}, \mathcal{H}^v, P, \{\mathcal{A}_i^v\}_{i\in\mathcal{N}\cup c}, \sigma_c^v, \{u_i^v\}_{i\in\mathcal{N}}, \mathcal{Z}^v\rangle$*, where each components represents:*

- ***Players*** $\mathcal{N} = \{a, d\}$ *There are two main players in the game: the attacker (a) and the defender (d). Additionally, c is the nature that represents the system randomness.*
- ***Histories*** $\mathcal{H}^v$ *Each vertex in the game tree* $h \in \mathcal{H}^v$ *corresponds to a unique sequence of actions taken from the beginning of the game.*
- ***Turn Function*** $P : \mathcal{H}^v \mapsto \mathcal{N} \cup \{c\}$ *The function* $P(h)$ *determines whose turn it is to make a move at each decision point for a given history vertex* h*.*
- ***Techniques*** $\mathcal{A}_i^v$ $\mathcal{A}_i^v$ *is a set of techniques that player* i *can take.* $A(h)$ *denote the feasible techniques for player* $i = P(h)$ *at vertex* $h \in \mathcal{H}^v$*.*
- ***System Randomness*** $\sigma_c^v \in \Sigma_c^v$ *Nature's fixed policy* σ_c^v *specifies the system randomness, which could be related to network traffic load, randomized system configuration, hardware failures, etc.*

- ***Tactic Expected Outcomes*** $\mathcal{Z}^v$ *$\mathcal{Z}^v$ represents the finite set of possible outcomes for each attack sequence in the MTG. These outcomes correspond to the results observed at the leaf vertices of the game tree, which could be the credentials to user devices, authorized connection to the server, no vulnerability found, etc.*
- ***Utilities*** $u_i^v : \mathcal{Z}^v \mapsto \mathbb{R}$ *The utility function u_i^v determines the payoff or cost player i receives when reaching a certain outcome.*

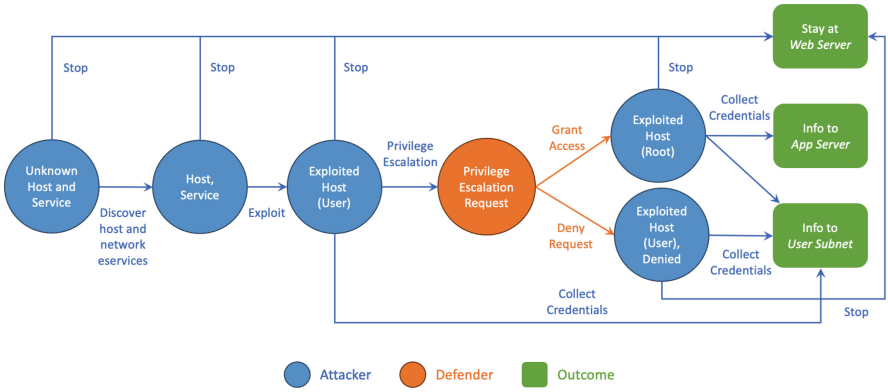


Fig. 3. Micro Tactic Game at the web server. The attacker needs to discover the host and services on the node, requesting privilege escalation to collect the credentials leading to other nodes. The defender could grant or deny the attacker's request depending on the defense strategy. The players' sequence of actions would lead to different expected tactic outcomes.

In the context of TTPs, the attack tactic at the current node corresponds to a sequence of techniques, while the outcomes represent the high-level tactical goals. In this work, we assume that the tactical outcomes are either staying in the current node or leading to another node that can be connected from the current node. Thus, with a slight abuse of notation, we denote $\mathcal{Z}^v = \{u \mid u \in \mathcal{V}, (v, u) \in \mathcal{E}\}$.

Penetration Plans

Before we define tactics or strategies, we need to understand the basis on which players make their decisions. For any strategic player, decisions are made based on the current knowledge of the situation. However, it is sometimes challenging for the player to obtain the complete interaction history due to partial observations. Consequently, there are decision vertices in the game tree that the player cannot distinguish between. In an extensive-form game, this is called an *information set*. In this work, we refer to this information set as a *knowledge set*.

Definition 2 (Knowledge Set). *Given the MTG at node $v \in \mathcal{V}$, a knowledge set $I_i \subseteq \mathcal{H}^v$ of a player i represents a set of decision vertices where the player i has the same available techniques and cannot distinguish between the vertices.*

The concept of knowledge set helps us better describe the decision-making process for both players. Given the MTG at the current node, attackers can construct their tactics in different ways. One approach is to have a sequential plan of techniques from the beginning to the end of the game. This step-by-step pure penetration or defense plan assigns a single technique to each possible knowledge set. Players can also randomize over single-technique plans at each knowledge set, known as a mixed penetration or defense plan.

Definition 3 (Pure Penetration (Defense) Plan). *Consider the MTG defined in Definition 1. Given the MTG Γ^v, a pure penetration or defense plan at node $v \in \mathcal{V}$ for player $i \in \mathcal{N}$ is a mapping $q_i^v : \mathcal{I}_i \mapsto \mathcal{A}_i^v$ that assigns a technique $q_i^v(I_i) \in A(I_i)$ for every knowledge set $I_i \in \mathcal{I}_i$. Denote Q_i^v as the set of all possible pure penetration or defense plans for player $i \in \mathcal{N}$ at this micro game. The pure penetration or defense plan for the entire system is defined as the set $\{q_i^v\}_{v \in \mathcal{V}}$, with $i = a$ means the attacker and $i = d$ means the defender.*

Definition 4 (Mixed Penetration (Defense) Plan). *Consider the MTG defined in Definition 1. Given the MTG Γ^v, a mixed penetration or defense plan at node $v \in \mathcal{V}$ for player $i \in \mathcal{N}$ is a probability distribution over all of player i's pure penetration plans, i.e., $\sigma_i^v \in \Delta(Q_i^v)$. Denote Σ_i^v as the set of all possible mixed penetration or defense plans for player $i \in \mathcal{N}$ at this micro game. The mixed penetration or defense plan for the entire system is defined as the set $\{\sigma_i^v\}_{v \in \mathcal{V}}$, with $i = a$ means the attacker and $i = d$ means the defender.*

The other approach is to focus on each knowledge set instead of defining the step-by-step actions of the entire game. At each knowledge set, a probabilistic distribution is assigned over the feasible techniques. This corresponds to the behavioral strategy in extensive-form games. Instead of planning everything ahead, this policy focuses on the decisions in each knowledge set. In this work, we call it an operational search plan. Denote by $\mathcal{I}_i$ the collection of knowledge sets of player $i \in \mathcal{N}$. By definition, for every knowledge set $I_i \in \mathcal{I}_i$, let $A(I_i)$ be the set of possible actions at I_i. Formally, the definition is given as follows.

Definition 5 (Operational Search Plan). *Given the MTG at node $v \in \mathcal{V}$, an operational search plan for player $i \in \mathcal{N}$ is a function mapping each of his knowledge set to a probability distribution over the set of possible techniques at that knowledge set, given by:*

$$b_i^v : \mathcal{I}_i \mapsto \bigcup_{I_i \in \mathcal{I}_i} \Delta(A(I_i)), \tag{1}$$

such that $b_i^v(I_i) \in \Delta(A(I_i))$ for all $I_i \in \mathcal{I}_i$. Denote B_i^v as the all admissible set of operational search policies of player $i \in \mathcal{N}$ at this MTG.

In this work, we assume that all the players have **perfect recall**; i.e., the player remembers every piece of information that he knows from the past, including his moves, the other player's moves, or chance moves. Under this assumption, we can always find the equivalence between the operational search plan is equivalent to the mixed penetration plans at the MTG of each node.

Theorem 1 (Planning Equivalence). *In every MTG in extensive form, if player $i \in \mathcal{N}$ has perfect recall, then for every mixed penetration plan there exists an equivalent operational search plan, and vice versa.*

Proof. For interested readers, the proof of the theorem follows Kuhn's theorem [1,5] in extensive-form games.

Theorem 1 indicates the equivalence between the mixed penetration plan and the operational search policy. The mixed penetration plan can be reduced to an operational search policy, and conversely, the operational search policy can generate a mixed penetration plan. The mixed penetration plan provides a holistic offline view, assigning a probability to each possible sequence of interactions. In contrast, the operational search policy describes the online decision-making process of the players. The equivalence allows us to choose the appropriate plan for the corresponding security purpose. The theorem, on the one hand, indicates that we can synthesize an operational strategy once we compute or are given a penetration plan. On the other hand, it suggests that a penetration plan or the course of actions of an attacker can be constructed after the penetration testing by obtaining the attacker's strategy at each decision point.

2.2 Security Schemes and Solution Concepts

Depending on the security goals, our framework is able to describe different security schemes and provide corresponding solution concepts.

Optimal Local Penetration Plan

The primary goal of penetration testing is to identify vulnerabilities in the target system. This includes not only surface-level vulnerabilities that can be detected through vulnerability scanning but also deeper vulnerabilities that can only be discovered through a sequence of attack actions. Penetration testing provides a thorough examination of the system, and this type of security scheme is commonly known as red teaming. Red teaming involves simulating a malicious attacker to assess the effectiveness of the current defense policy. In this context, the defender's strategy remains fixed, while the attacker responds optimally to the defense policy. Red teaming aims to determine the optimal local penetration plan $\sigma_a^{v,red}$ that maximizes the attacker's utility, given the defender's strategy and the inherent system randomness in the system. The solution concept for the optimal local penetration plan is defined as follows:

Definition 6 (Optimal Local Penetration Plan). *For the MTG at node $v \in \mathcal{V}$, given the defense strategy σ_d^v and system randomness σ_c^v, the optimal local*

penetration plan is a probability distribution over all attacker's pure penetration plans, i.e., $\sigma_a^v \in \Delta(Q_i^v)$ *given by*

$$\sigma_a^{v,red}(\sigma_d^v, \sigma_c^v) \in \arg\max_{\sigma_a^v \in \Sigma_a^v} u_a^v(\sigma_a^v, \sigma_d^v, \sigma_c^v), \tag{2}$$

where $u_a^v(\sigma_a^v, \sigma_d^v, \sigma_c^v)$ *is the expected utility of outcome generated following the plan profile* $\Phi^v = (\sigma_a^v, \sigma_d^v, \sigma_c^v)$.

The optimal local penetration plan is an ex-ante strategy where the attacker or penetration tester has complete information about the local node, including defense strategy, possible vulnerabilities, and system randomness. With this information, an optimal pure or mixed penetration plan can be obtained to visualize different attack chains along with their outcomes and probabilities. However, in practice, the attacker or penetration tester may not have complete information. Their penetration plan is developed through learning (e.g., via machine learning or reinforcement learning) without complete prior knowledge about the local node. The practically used penetration plan belongs to the family of operational search plans, which is a mapping from the knowledge set to the probability distribution over the set of possible techniques.

Remark 1 (Optimal v.s. Practical). The practically used penetration plan is equivalent to the optimal local penetration plan when the penetration tester's learning results are perfect. This is possible when the penetration testing agent or attacker, through its learning process, has identified the exact set of actions to take in each state to maximize the expected reward, as if it had known the full model from the beginning. Under these conditions, the practically used penetration plan is identical to the optimal operational search plan. According to Theorem 1, this is thus equivalent to the optimal local attack policy in Definition 6.

The optimal local penetration plan generates useful byproducts that help describe the penetration plan. One such byproduct is the ***course of action***, which describes the realized sequence of attack techniques derived from the penetration plan. Another important concept is the ***tactic outcome probability***, which represents the total probability of reaching any outcome of the game $z \in \mathcal{Z}^v$. Let $H^z \subset \mathcal{H}$ be the set of leaf vertices with outcome $z \in \mathcal{Z}^v$. Define $L(h^z) = \{(h_1, a_1), (h_2, a_2), \dots\}$ as the sequences of vertices and actions leading to the leaf vertex $h^z \in H^z$. The tactic outcome probability is defined as follows.

Definition 7 (Tactic Outcome Probability). *For the MTG at node* $v \in \mathcal{V}$*, given the nature's fixed policy (if any) and the plan profile of the attacker and the defender, i.e.,* $\Phi^v = (\sigma_a^v, \sigma_d^v, \sigma_c^v)$*, we define* $\tau^v : \mathcal{Z}^v \mapsto [0, 1]$ *as the tactic outcome probability. We use* $\tau^v(z)$ *to denote the probability of reaching outcome* $z \in \mathcal{Z}^v$ *as*

$$\tau^v(z \mid \Phi^v) == \sum_{h^z \in \mathcal{H}^z} \left[\prod_{(h_j, a_j) \in L(h^z)} \sigma_i^v(a_j) \mathbb{1}_{\{P(h_j)=i\}} \right], \tag{3}$$

where $P(h_j)$ is the turn function and $\sigma_i^v(a_j)$ is the probability that action a_j is chosen by player $i = P(h_j)$.

Purple Teaming Defense Plan

Purple teaming is a collaborative cybersecurity assessment that combines attack and defense strategies to enhance the overall security posture of a system. While red teaming penetration testing predicts the attacker's behavior, purple teaming focuses on improving the defense policy to mitigate potential attacks. This approach corresponds to a Stackelberg game or leader-follower model, where the defender, as the leader, enforces their strategy on the attacker, as the follower. The defender must anticipate the attacker's responses to the defense strategy and optimize the defense policy accordingly, resulting in a bi-level optimization problem. Penetration testing provides credible predictions of the attacker's penetration plan, enabling proactive defense with purple teaming. The solution concept for the purple teaming defense plan is defined as follows:

Definition 8 (Optimal Purple Teaming Defense Plan). *For the MTG at node $v \in \mathcal{V}$, given the system randomness $\sigma_c^v \in$, the optimal purple teaming defense plan includes two parts: $\sigma_d^{v,pur} \in \Sigma_d^v$ is the optimal purple teaming defense plan, which is a probability distribution over all defender's pure defense plans, i.e.,$\sigma_d^{v,pur} \in \Delta(Q_d^v)$; $\sigma_a^{v,*} \in \Sigma_a^v$ is the anticipated optimal local penetration plan for the attacker given the defense plan.*

$$\sigma_d^{v,pur}(\sigma_c^v) \in \max_{\sigma_d^v \in \Sigma_d^v} \quad u_d^v(\sigma_a^{v,*}, \sigma_d^v, \sigma_c^v) \tag{4}$$

$$s.t. \quad \sigma_a^{v,*} \in \arg\max_{\sigma_a^v \in \Sigma_a^v} u_a^v(\sigma_a^v, \sigma_d^v, \sigma_c^v). \tag{5}$$

The inner optimization problem aligns with the optimal local penetration plan as defined in Definition 6, aiming to predict the worst-case attacker behavior under the current defense strategy. Penetration testing, utilizing learning techniques, determines the attacker's anticipated response to a given defense strategy. To implement purple teaming defense in practical settings, organizations undergo an iterative process where the defender tests a defense strategy, observes the worst-case attack, and then adjusts the defense to achieve better utility.

Risk Assessment at Equilibrium

Another important venue penetration testing contributes to is the risk assessment of the system. Instead of focusing on individual attack events, a risk assessment would take into account the average or the steady state of the long-term behaviors of the attacker and defender in the long run. The concept of equilibrium in game theory offers a natural way to analyze these steady-state strategic interactions within the system. A Nash Equilibrium (NE) in the MTG provides a solution where no player has an incentive to deviate from their strategy. Formally, the solution concept for risk assessment is defined as follows:

Definition 9 (Nash Equilibrium-Informed Risk Assessment). *For the micro tactic game at node $v \in \mathcal{V}$, given the system randomness $\sigma_c^v \in \Sigma_c^v$, the Nash equilibrium-informed risk assessment is a plan profile $(\sigma_a^{v,*}, \sigma_d^{v,*})$, where $\sigma_a^{v,*}$ is the equilibrium penetration plan for the attacker and $\sigma_d^{v,*}$ is the equilibrium defense plan for the defender. The Nash equilibrium plans satisfy*

$$u_i^v(\sigma_i^{v,*}, \sigma_{-i}^{v,*}) \geq u_i^v(\sigma_i, \sigma_{-i}^{v,*}), \tag{6}$$

for all admissible $\sigma_i^v \in \Sigma_i^v$ and for all $i \in \mathcal{N}$.

To practically solve the game, we consider a refinement of Nash equilibrium in sequential games: Subgame Perfect Nash Equilibrium (SPNE). In addition to satisfying the conditions of Nash equilibrium, SPNE requires that strategies remain in equilibrium at every possible subgame of the overall game. It can be solved using backward induction as the game-theory version of the dynamic programming principles.

Theorem 2. *For every finite micro tactic game at node $v \in \mathcal{V}$ with fixed system randomness $\sigma_v^c \in \Sigma_c^v$, the game with perfect recall has a subgame perfect Nash equilibrium in mixed or operational search penetration/defense plans. The game with perfect information has a subgame perfect Nash equilibrium in pure penetration and defense plans [6].*

Theorem 2 states that we can always find the risk assessment equilibrium in mixed plans or operational search policies, even with imperfect information. Mixed penetration plans provide the probability of the entire attack/defense action sequence occurring, offering a holistic view for analysis purposes. On the other hand, operational search policies focus on what happens in each knowledge set, providing a fine-grained strategy. The equivalence between them allows us to zoom in or out as needed, facilitating flexible and comprehensive analysis.

2.3 Macro Strategic Process

One key component in the MTG is the utility function for each outcome, $u_i^v(z)$, for all $z \in \mathcal{Z}^v$ and $i \in \mathcal{N}$. Utilities represent the payoff or cost of staying or moving to the next node and must be evaluated globally, considering neighboring nodes and their connections. After local exploration and exploitation, the attacker can use obtained credentials or discovered vulnerabilities to move to different nodes, a process known as *lateral movement*. The attacker's movement and the creation of the attack kill chain depend on the network topology and the expected utilities of each node. We model this decision-making process across the network using an MDP, referred to as a Macro Strategic Process (MSP).

MSP Modeling

Definition 10 (Macro Strategic Process (MSP)). *The Macro Strategic Process of the MEGA-PT is defined by an MDP Λ^g. Given the target networked system $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$, the MSP for the attacker can be represented by a tuple $\Lambda^g = \langle \mathcal{S}, \mathcal{A}^g, T, R, \gamma \rangle$, where each component represents:*

- ***Network Nodes*** $\mathcal{S} = \mathcal{V}$**:** *The nodes in the network form the state space. Each node or state* $v \in \mathcal{V}$ *processes an MTG* Γ^v *as defined in Definition 1.*
- ***Connections*** $\mathcal{A}^g = \mathcal{E}$**:** *The connections between the nodes are the attacker's action space, which is equivalent to all the directed edges in the network.*
- ***Transition Success Probability*** $T : \mathcal{S} \times \mathcal{A}^g \mapsto \Delta(\mathcal{S})$**:** *This function describes the success rate of the lateral movement attempt between the nodes.*
- ***Movement Rewards*** $R : \mathcal{S} \times \mathcal{A}^g \times \mathcal{S} \mapsto \mathbb{R}$**:** *The immediate reward or penalty for the attacker when trying to laterally move along the edge to the other node.*
- ***Discounting Factor*** $\gamma \in (0, 1]$**.**

From a global perspective, each node in the network $v \in \mathcal{V}$ can be viewed as a state in the Markov Decision Process. The edges in the network indicate the lateral movement of the attacker within the network. Whether the attack attempt is successful depends on the capability of the attacker. For simplicity, in this work, we assume the transition success probability is defined as follows. For every $s \in \mathcal{V}$ and $a^g \in \mathcal{A}^g$,

$$T(s' \mid s = v, a^g = (v, u)) = \begin{cases} 1 & \text{if } u = v \text{ and } s' = v, \\ c_a & \text{if } u \neq v \text{ and } s' = u, \\ 1 - c_a & \text{if } u \neq v, \text{ and } s' = v, \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

If the attacker chooses to stay at the same node, the self-loop edge will lead to the same state with probability one. If the attacker chooses to use any outgoing edge and move to another node, the attempt will succeed with probability $c_a \in [0, 1]$, which represents the attacker's capability. If the attempt fails, the attacker will stay at the same node.

The goal of penetration testing is to estimate the potential damage an attacker can inflict by compromising the network and affecting system production. A positive reward is given when the attacker enters a node, with the reward value depending on the node's importance to the system. Conversely, staying at the same node indicates that the attacker either failed to move to another node or that the information obtained from the MTG was insufficient for progression. Therefore, staying at the same node results in a negative penalty for the attacker. The movement reward function is given by the following equation:

$$R(s = v, a^g = (v, u), s') = \begin{cases} M_a & \text{when } s' = v, \\ \bar{V}(v) & \text{when } s' = u\,, \forall u \in \mathcal{V} \setminus \{v\}. \end{cases} \tag{8}$$

where $M_a \in \mathbb{R}^-$ is a penalty for the attacker staying at the same node without progressing towards the target. $\bar{V} : \mathcal{V} \mapsto \mathbb{R}^+$ is the reward for entering the state. This value depends on the production importance of the node $v \in \mathcal{V}$ to the target system.

Global Attack Strategy
Unlike traditional MDPs, where the attacker can freely choose actions to optimize expected utility, in the realistic penetration testing settings, the attack strategy at the network level depends on explorations at the local nodes. If the attacker does not find any vulnerabilities leading to the next node, they cannot move forward. Therefore, the global attack strategy in the MSP relies on the outcomes of the MTG.

For each MTG Γ^v at node $v \in \mathcal{V}$, the optimal local penetration plans generate the tactic outcome probability as defined in Definition 7. Since the outcome space of the MTG is equivalent to the set of the outgoing edges at node v, i.e., $\mathcal{Z}^v = \{u \mid u \in \mathcal{V}, (v,u) \in \mathcal{E}\}$, we can view the tactic outcome probability $\tau^v(z)$ as the probability that the attacker will choose action $a^g = (v,z)$ for the MSP. Formally, it leads to the following definition.

Definition 11 (Global Attack Strategy). *Consider the MTG defined in Definition 1 and the MSP defined in Definition 10. The global attack strategy in MSP is a mapping from the state space to the global action space, i.e.,* $\pi^g : \mathcal{S} \mapsto \Delta(\mathcal{A}^g)$. *For node* $v \in \mathcal{V}$, *given the MTG* Γ^v *and the local plan profile* $\Phi^v = (\sigma_c^v, \sigma_a^v, \sigma_d^v)$, *the global attack strategy is given by*

$$\pi^g(a^g \mid s) = \pi^g(a^g = (v,z) \mid s = v) = \tau^v(z \mid \Phi^v), \qquad \forall z \in \mathcal{Z}_v, \tag{9}$$

where $\tau^v(z \mid \Phi^v)$ *is the tactic outcome probability as defined in Definition 7.*

The global attack strategy in the MTG outlines the cyber kill chain and the sequence of tactics across the entire system. Rather than focusing on the details at each local node, the MSP connects all the nodes in the network, offering a comprehensive risk assessment for the entire system. This holistic approach allows organizations to better understand the interconnections within their network and the cascading effects of vulnerabilities throughout the system.

2.4 Meta Penetration Game and Playbook

Policy evaluation offers a way to estimate the effectiveness of the global attack strategy π^g in terms of expected cumulative utilities. Similar to traditional MDPs, policy evaluation of the global attack strategy computes the value functions using the Bellman equations. For all states $s \in \mathcal{S}$, the value function under π^g is given by

$$V^{\pi^g}(s) = \sum_{a^g \in \mathcal{A}^g} \pi^g(a^g \mid s) \sum_{s' \in \mathcal{V}} T(s' \mid s, a^g) \left[R(s, a^g, s') + \gamma V^{\pi^g}(s') \right]. \tag{10}$$

The value at each node in the system describes the expected return starting from that node and then acting according to the global attack strategy π^g. For the players at the MTG, the utility of each outcome describes the expected reward of taking that action and moving to the next node in the macro strategy process. Thus, we define the utility functions in the MTG as follows.

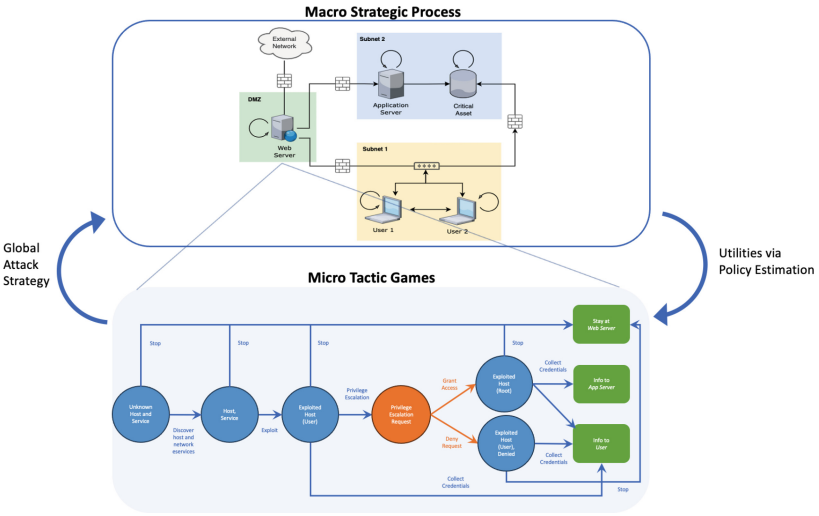


Fig. 4. Relationship between Macro Strategic Process and Micro Tactic Games. The local penetration plans in the micro games affect the global attack strategy, while the policy evaluation at the macro process helps provide the utilities in the micro games.

Definition 12 (MTG Utilities). *Given the global attack strategy* $\pi^g \in \Pi^g$*, the attacker's utility functions of reaching outcome* $z \in \mathcal{Z}^v$ *in the MTG at node* $v \in \mathcal{V}$ *are defined as the*

$$u_a^v(z = u) = \sum_{s' \in \mathcal{V}} T(s' \mid s = v, a^g = (v, u)) \left[R(s, a^g, s') + \gamma V^{\pi^g}(s') \right], \tag{11}$$

where V^{π^g} *is the policy evaluation value function in* (10)*. The defender's utility is the opposite of the attacker, i.e.,* $u_d^v(z) = -u_a^v(z)$ *for all* $z \in \mathcal{Z}^v$*.*

Figure 4 illustrates the relationship between the MSP and the MTGs. The MSP defines the global attack strategy, forming an attack kill chain and providing estimated values for each node through policy evaluation under the current strategy. These estimated values represent the expected outcome utilities at each MTG, guiding the formulation of detailed penetration plans at each node. The sequence of attack and defense techniques at the local node influences the global attack strategy in the macro view, emphasizing how lateral movement is determined by exploration and exploitation. This iterative process continues until a meta-solution is reached. Together, the MSP and the MTGs constitute a meta-security game that captures decision-making in penetration testing at both network and node levels.

Definition 13 (Meta-Security Game). *Given the network system graph* $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$*, the meta-security game is composed of two parts:* $\Xi = \langle \{\Gamma^v\}_{v \in \mathcal{V}}, \Lambda^g \rangle$*, where* $\{\Gamma^v\}_{v \in \mathcal{V}}$ *is the set of MTGs as defined in Definition 1 and* Λ^g *is the macro strategy process as defined in Definition 10.*

The MSP and the MTGs are inherently coupled, as the local penetration plans in the MTGS naturally affect the global attack strategy, while the policy evaluation at the MSP helps provide the utilities in the MTGs. Hence, a holistic solution concept is necessary for the proposed meta-security game.

Definition 14 (Meta Penetration Playbook). *Consider the meta-security game* $\Xi = \langle \{\Gamma^v\}_{v\in\mathcal{V}}, \Lambda^g \rangle$ *defined in Definition 13, the meta penetration playbook* $\xi = \langle \{\Phi^v\}_{v\in\mathcal{V}}, \pi^g \rangle$ *is composed of two elements:*

- ***Local Penetration profile:*** $\Phi^v = (\sigma_a^{v,*}, \sigma_d^{v,*}, \sigma_c^v)$ *constitutes the local penetration plans of all players for the MTG at node* Γ^v *for each* $v \in \mathcal{V}$,
- ***Global Attack Strategy:*** π^g *is the global attack strategy in the macro strategy process,*

which satisfy two conditions:

- ***Policy Dependency:*** *The global attack strategy* π^a *at the macro strategy process depends on the local penetration plans* $\{\Phi^v\}_{v\in\mathcal{V}}$ *as defined in Definition 11,*
- ***Value Dependency:*** *For each MTG at node* $v \in \mathcal{V}$*, the utility of each tactic's expected outcome depends on the policy evaluation results of global attack strategy* π^g *according to Definition 12.*

In a global view, a complete cyber attack kill chain comprises a sequence of tactics. The global attack strategy guides how to compose this attack kill chain within the target system. Within each tactic, there is a sequence of techniques. The local penetration profile at each node describes the decision-making process of the players to complete these technique sequences. The policy and value dependencies connect the macro and micro solutions, helping us to form an efficient and consistent meta-penetration playbook for the meta-security game.

3 Computation

To determine the optimal meta-penetration playbook, we propose the following algorithm to find the exact solution. In this section, we use the purple teaming defense as the penetration scheme and solution concept to describe the computational process. For other security schemes, the general structure of the algorithm remains the same, but the method for obtaining the local penetration profile in each MTG differs (line 7 in Algorithm 1).

To analyze the risks of each node and evaluate the effectiveness of the system defense, we define the network risk score as a measurement metric. For each node $v \in \mathcal{V}$ in the system, we are interested in whether the attacker has access to this node, and what is the expected damage he can create. Let $V_{max} \in \mathbb{R}^+$ be the maximum damage that the attacker could cause. Given the meta-security game Ξ and the corresponding meta penetration playbook ξ, the network risk score of node $v \in \mathcal{V}$ is a normalized risk value $Risk(v \mid \xi) \in [0, 1]$ given by $V^{\pi^g}(v)/V_{max}$ if $V^{\pi^g}(v)$ is non-negative; otherwise the score is set to 0.

Algorithm 1: Purple Teaming Meta Penetration Playbook Algorithm

Input: Meta-security game $\Xi = \langle \{\Gamma^v\}_{v\in\mathcal{V}}, \Lambda^g \rangle$

Set the utilities u_i^v in each MTG to arbitrary value;
repeat
  Micro Penetration Profile Computation:
    For every MTG at $v \in \mathcal{V}$, compute the purple teaming penetration plan profile $\Phi^v = (\sigma_a^{v,*}, \sigma_d^{v,pur}, \sigma_c^v)$;
    Compute the attack strategy π^a under $\{\Phi^v\}_{v\in\mathcal{V}}$ using (9) ;
  Macro Attack Strategy Evaluation:
    Compute the value function V^{π^g} of Λ^g using (10);
    Update the utilities u_i^v in each MTG Γ^v;
until *Meta penetration playbook converges*;

Result: Meta penetration playbook $\xi = \langle \{\Phi^v\}_{v\in\mathcal{V}}, \pi^g \rangle$.

Table 1. Movement rewards for the attacker in the network.

Web Server	User Devices	App Server	Critical Asset	Operation Down	Penalty
0	5	20	30	100	−15

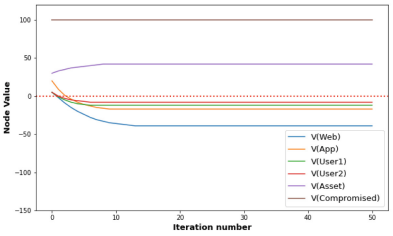
4 Case Study

We use the network topology depicted in Fig. 2 as a case study to demonstrate the effectiveness of MEGA-PT. The system consists of 5 nodes, including the web server, two user devices, the application server, and the critical asset. The MTG trees for each node are illustrated in Appendix A. These game trees align with attack scenarios from the MITRE ATT&CK model and can be adjusted to fit specific system structures. We evaluate the performance of our model through numerical experiments conducted in a self-built Python simulator. While the model's applicability extends to practical systems given the network topology and vulnerability trees, the details on how to gather this information are beyond the scope of this paper.

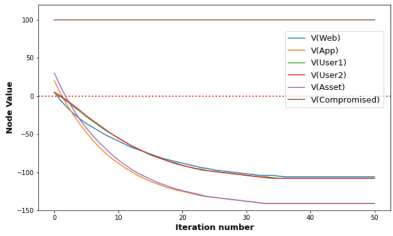
The penetration testing agent acts as an attacker entering the system from the external network, starting at the web server. The goal is to penetrate the system and potentially affect operations at the critical asset. We assume that there is an artificial node in the network representing a successful compromise of operations. Once the attacker reaches this node, the penetration process is considered terminated. In our experiments, we set the parameters as follows: the immediate rewards for entering each node and the penalty are specified in Table 1. The attacker's capability is denoted as $c_a = 0.8$ by default, and we use $\gamma = 0.9$ for the policy evaluation process.

4.1 Optimal Penetration Plan and Purple Teaming

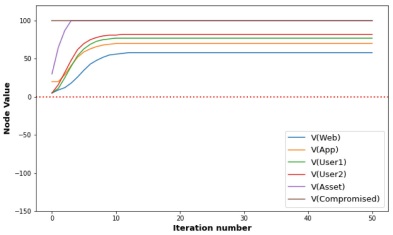
Figure 5 illustrates the value of each node in the network during the meta penetration playbook computation. We consider three types of attackers with differ-



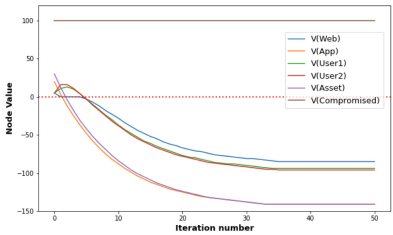
(a) Fixed defense, Weak Attacker $c_a = 0.2$.



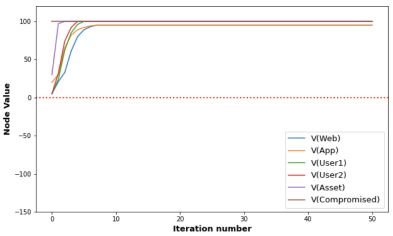
(b) Purple Teaming. Weak Attacker $c_a = 0.2$.



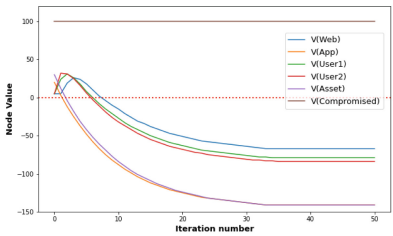
(c) Fixed defense, Median Attacker $c_a = 0.5$.



(d) Purple Teaming. Median Attacker $c_a = 0.5$.



(e) Fixed defense, Strong Attacker $c_a = 0.8$.



(f) Purple Teaming. Strong Attacker $c_a = 0.8$.

Fig. 5. Node values under different conditions. The x-axis is the number of iterations and y-axis is the value of the node for the attacker. We consider both fixed defense and purple teaming defense against three types of attackers ($c_a = \{0.2, 0.5, 0.8\}$).

ent capabilities: a weak attacker with $c_a = 0.2$, a median attacker with $c_a = 0.5$, and a strong attacker with $c_a = 0.8$. In the left column, we test the model under a fixed defense strategy. Specifically, at the web server, the probability of the defender granting access is 0.7. At the application server, the probability of the defender enforcing strict authorization policies is 0.3. Finally, at the critical asset, the probability of the defender executing the command is 0.6.

The value of each node represents the expected accumulated reward if the attacker starts penetration from that node. As observed in the figures, when the attacker is weak, even with a fixed defense strategy, no node in the system yields a positive reward. However, as the attacker's capabilities increase, certain states in the system can provide positive rewards. The stronger the attacker, the

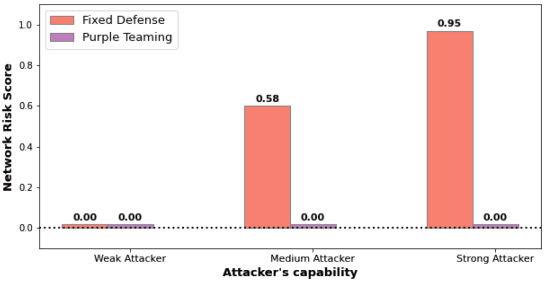


Fig. 6. Network risk score of the web server under different security schemes.

higher the maximum reward achievable. The potential system damage can be mitigated by adopting a purple teaming defense strategy. In Fig. 5, the right column demonstrates that when implementing purple teaming and adjusting the defense plan, scenarios where the attacker previously gained positive rewards turn into negative rewards as the strategy converges. The attacker can only achieve gains when the defense strategy is not yet converged. These results illustrate that our model effectively accommodates varying attacker capabilities, and purple teaming offers enhanced defense capabilities.

Figure 6 shows the network risk score of the web server under different security schemes. We consider $V_{max} = 100$ as the maximum damage the attacker could cause. With a fixed defense, the web server's network risk score increases as the attacker's capability increases. The web server is not risky only when the attacker is weak, and his capability is low ($c_a = 0.2$). With purple teaming defense, the web server is safe against all types of attackers, and the network risk score always remains zero. This indicates that purple teaming defense helps the system find a defense strategy that can reduce the network risk and prevent the system from being compromised.

4.2 Vulnerability Adaptability

We consider a scenario where the MTG changes within the network. In this experiment, we assume a change at the application server where no longer any information can lead to the critical asset. This change might occur if the system encrypts this information or restores the database, preventing the attacker from decrypting or accessing the data related to the critical asset. Consequently, the MTG at the application server changes, with the only outcome being staying at the same node after exploration.

We compute the meta penetration playbook under a fixed defense strategy before and after the local node change:

$$\text{Before: } \xi = \langle \Phi^{web}, \Phi^{app}, \Phi^{user1}, \Phi^{user2}, \Phi^{asset}, \pi^{g} \rangle;$$
$$\text{After: } \xi' = \langle \Phi^{web}, \Phi^{app,\prime}, \Phi^{user1}, \Phi^{user2}, \Phi^{asset}, \pi^{g,\prime} \rangle$$

Table 2. Global attack strategy π^g before vulnerability change.

	web	app	user	user	asset	final
web	0	0.7	0.3	0	0	0
app	0	0.37	0	0	0.63	0
user	0	0	0	0.5	0.5	0
user	0	0	0.3	0	0.7	0
asset	0	0	0	0	0.4	0.6
final	0	0	0	0	0	1

Table 3. Global attack strategy π^g after vulnerability change.

	web	app	user1	user2	asset	final
web	0	0	1	0	0	0
app	0	1	0	0	0	0
user1	0	0	0.5	0.5	0	0
user2	0	0	0.3	0	0.7	0
asset	0	0	0	0	0.4	0.6
final	0	0	0	0	0	1

It can be noticed that in the meta penetration playbook, the vulnerability change only affects the application server and the global attack strategy, while the other plans remain the same. This indicates that we only need to recompute the MTG at that node while retaining the original structure of the other nodes and updating the global attack policy.

The comparison of global attack strategies before and after this local node change is illustrated in Table 2 and Table 3. Prior to the vulnerability change, the attacker from the web server had a high probability ($\Pr = 0.7$) of transferring to the application server to further attack the system. However, after the change, since there is no longer a connection between the application server and the critical asset, our computed penetration plan adjusts its global attack strategy and no node would transfer to the application server anymore. The attacker would focus solely on the user devices to find any information that could compromise the operation. This result demonstrates that MEGA-PT can effectively adapt to local vulnerability changes.

4.3 Network-Level Scalability

In this scenario, we demonstrate the scalability of MEGA-PT by increasing the number of user devices within the subnet. We assume that all users share the same micro tactic tree, and the outcome leads to other users randomly transferring to another user device in the network with equal probability. Since MEGA-PT is modular, it allows us to compute each micro tactic tree in parallel. This means that if users share the same micro game, we can compute one instance and apply the result to all nodes in the network without recomputation. In contrast, traditional reinforcement learning-based methods treat each node's status in the system as a separate state. As the number of users increases, the state space grows exponentially, resulting in significantly increased computational time.

From a meta penetration playbook point of view, the network-level scale change results in the following change in the playbook:

$$\text{Before: } \xi = \langle \Phi^{web}, \Phi^{app}, \Phi^{user}, \Phi^{asset}, \pi^g \rangle;$$
$$\text{After: } \xi' = \langle \Phi^{web}, \Phi^{app}, \Phi^{user}, \Phi^{user}, \Phi^{user}, \ldots, \Phi^{asset}, \pi^{g,\prime} \rangle$$

Since the user devices are of the same type, each local penetration profile for the user device is the same. The only updated element in the playbook is the global attack strategy as it would consider more nodes in the system.

Figure 7 compares the computational time for finding the optimal strategy between our model and an RL-based model under different numbers of users. In the RL-based model, the state is the aggregation of all related information at each node, such as whether the web server has been discovered or whether the user credential has been found. The transition and reward in the RL-based model follow the same setting, but the state and transition space are enormous. We use Q-learning as the learning method under fixed defense and compare the computational time to find the optimal penetration strategy in the system. It is evident that as the number of users increases, the computational time for the RL-based method increases drastically, whereas our method shows minimal change. These results demonstrate that MEGA-PT scales effectively with large networks containing similar devices, providing robust scalability.

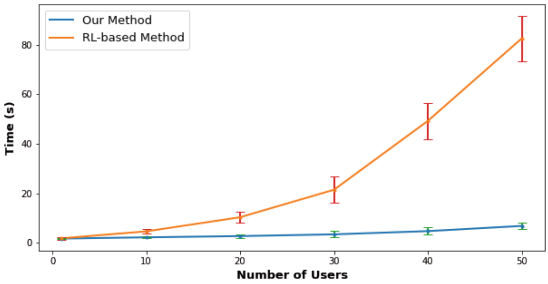


Fig. 7. Scalability comparison between our model and RL-based model.

5 Conclusion

In this work, we propose MEGA-PT, a meta-game agile penetration testing model for automated and effective penetration testing. This model features MTGs for local node interactions and a macro strategy process for network-wide attack chains. It adheres to the TTPs in real cyber security frameworks, allows distributed and modularized penetration testing, and adapts to changes at both local and network levels. Experiments show that using MEGA-PT's purple teaming, the system can find effective defense strategies to reduce the network risk score of each node. Compared to other RL-based automated penetration testing models, MEGA-PT's distributed features enable agile adaptation to both local-level vulnerability changes and network-level topology changes, allowing effective and scalable penetration testing in large network systems.

For future work, we plan to discuss global defense strategies at the macro level and explore partial information in the game. MEGA-PT is promising for

extension to different security schemes and can serve as a foundational framework for the next generation of automated penetration testing.

A Appendix: Micro Tactic Game Trees

(See Figs. 8, 9 and 10)

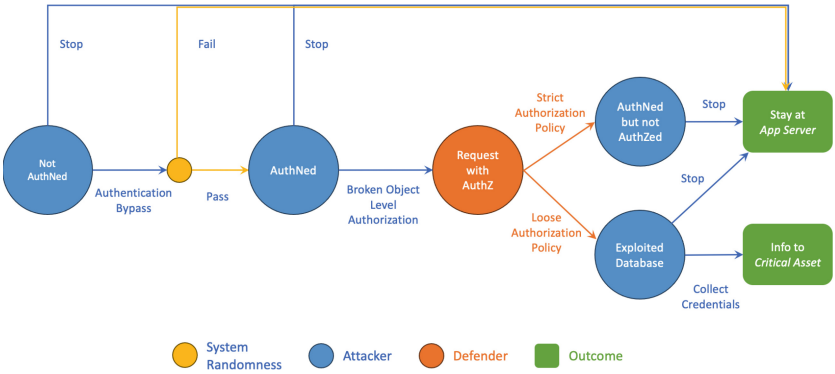


Fig. 8. MTG tree at the application server.

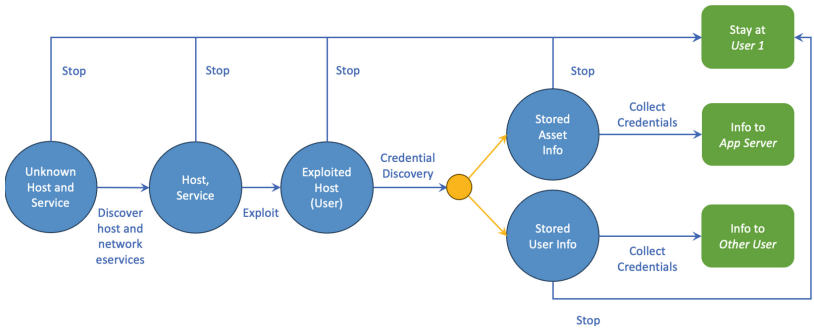


Fig. 9. MTG tree at the user device.

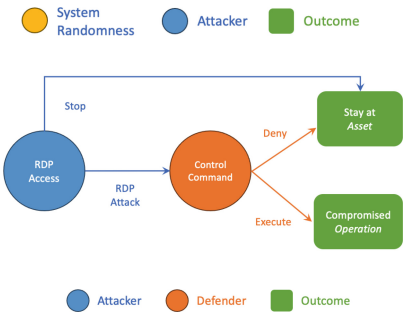


Fig. 10. MTG tree at the critical asset.

References

1. Aumann, R.J.: Mixed and Behavior Strategies in Infinite Extensive Games. Princeton University, Princeton (1961)
2. Ge, Y., Zhu, Q.: Gazeta: Game-theoretic zero-trust authentication for defense against lateral movement in 5G IoT networks. IEEE Trans. Inf. Forensics Secur. (2023)
3. Ghanem, M.C., Chen, T.M.: Reinforcement learning for efficient network penetration testing. Information **11**(1), 6 (2019)
4. Hu, Z., Beuran, R., Tan, Y.: Automated penetration testing using deep reinforcement learning. In: 2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), pp. 2–10. IEEE (2020)
5. Kuhn, H.W.: Extensive games and the problem of information. Contrib. Theory Games **2**(28), 193–216 (1953)
6. Maschler, M., Zamir, S., Solan, E.: Game Theory. Cambridge University Press, Cambridge (2020)
7. MITRE: Mitigations enterprise MITRE ATT&CK (2020). https://attack.mitre.org/mitigations/enterprise/
8. Shmaryahu, D., Shani, G., Hoffmann, J., Steinmetz, M.: Partially observable contingent planning for penetration testing. In: Iwaise: First International Workshop on Artificial Intelligence in Security, vol. 33 (2017)
9. Zhao, Y., Ge, Y., Zhu, Q.: Combating ransomware in internet of things: a games-in-games approach for cross-layer cyber defense and security investment. In: Bošanský, B., Gonzalez, C., Rass, S., Sinha, A. (eds.) GameSec 2021. LNCS, vol. 13061, pp. 208–228. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-90370-1_12