Autonomous Cyber Defense: A Multi-Agent Decision-Making Framework

Mathematical Formulation of the Cyberwheel Environment

Contents

We consider a multi-agent decision-making system for autonomous cyber defense, where a Red agent (attacker) attempts to compromise network assets while a Blue agent (defender) deploys defensive measures including cyber deception techniques. The Red agent follows the MITRE ATT&CK framework with structured kill-chain phases, while the Blue agent strategically places decoy hosts (honeypots) and isolates compromised systems to minimize damage and gather threat intelligence.

1 Environment

1.1 Decision-making problem overview

We consider an episodic reinforcement learning problem where each episode represents a complete cyber attack scenario. Episodes are of finite length H, representing the maximum number of decision steps before the environment resets. We use T to denote the total number of training episodes.

The Red and Blue agents operate in a shared network environment but have distinct state and action spaces reflecting their asymmetric roles. We use $\mathcal{S}^{(r)}$ and $\mathcal{S}^{(b)}$ to denote the state space of the red and blue agents respectively. We similarly use $\mathcal{A}^{(r)}$ and $\mathcal{A}^{(b)}$ to denote the action space of the two agents.

The agents operate in a turn-based fashion within each timestep. Formally, for each decision time $h \in [1 : H]$ within episode $t \in [1 : T]$, the red agent observes the network state and executes an attack action first, followed by the blue agent observing alerts and taking a defensive action.

In episode t at decision time h, the red and blue agents observe their respective states $S_{t,h}^{(r)} \in \mathcal{S}^{(r)}$ and $S_{t,h}^{(b)} \in \mathcal{S}^{(b)}$. After observing their states, the agents select actions $A_{t,h}^{(r)} \in \mathcal{A}^{(r)}$ and $A_{t,h}^{(b)} \in \mathcal{A}^{(b)}$ according to their respective policies $\pi^{(r)}$ and $\pi^{(b)}$.

The environment provides immediate rewards $R_{t,h}^{(r)}$ and $R_{t,h}^{(b)}$ to each agent based on the outcomes of their actions and the current network state. These rewards are generally adversarial - successful red actions that compromise real assets provide negative rewards to the blue agent, while successful deception (red agent attacking decoys) provides positive rewards to the blue agent.

The decision-making objective of the red agent is to maximize its expected cumulative reward:

$$J^{(r)}(\pi^{(r)}) = \mathbb{E}\left[\sum_{t=1}^{T}\sum_{h=1}^{H}\gamma^{h-1}R_{t,h}^{(r)} \mid \pi^{(r)}\right] \tag{1}$$

The decision-making objective of the blue agent is to maximize its expected cumulative reward:

$$J^{(b)}(\pi^{(b)}) = \mathbb{E}\left[\sum_{t=1}^{T}\sum_{h=1}^{H}\gamma^{h-1}R_{t,h}^{(b)} \mid \pi^{(b)}\right] \tag{2}$$

where $\gamma \in [0, 1]$ is the discount factor that determines the relative importance of immediate versus future rewards.

1.2 Red Agent (Attacker)

The red agent represents a sophisticated cyber adversary following the MITRE ATT&CK framework. Its behavior is structured around kill-chain phases that model realistic attack progression.

1.2.1 State Space

The red agent's state space $\mathcal{S}^{(r)} \subset \mathbb{R}^{d_r}$ is a d_r-dimensional vector encoding:

$$S_{t,h}^{(r)} = \begin{bmatrix} \text{pos}_{t,h} \\ \text{knowledge}_{t,h} \\ \text{phase}_{t,h} \\ \text{capabilities}_{t,h} \end{bmatrix} \tag{3}$$

where:

- $\text{pos}_{t,h} \in \{0,1\}^{|H|}$ is a one-hot encoding of the red agent's current compromised host position
- $\text{knowledge}_{t,h} \in \{0,1\}^{|H|+|S|}$ represents discovered network information (hosts and subnets)
- $\text{phase}_{t,h} \in \{0,1\}^4$ is a one-hot encoding of the current kill-chain phase: {discovery, reconnaissance, privilege-escalation, impact}
- $\text{capabilities}_{t,h} \in \{0,1\}^{|\mathcal{T}|}$ indicates available techniques from the set $\mathcal{T}$ of MITRE ATT&CK techniques

The total dimensionality is $d_r = |H| + |H| + |S| + 4 + |\mathcal{T}|$, where $|H|$ is the number of hosts, $|S|$ is the number of subnets, and $|\mathcal{T}|$ is the number of available attack techniques.

1.2.2 Action Space

The red agent's action space $\mathcal{A}^{(r)}$ consists of kill-chain phase actions:

$$\mathcal{A}^{(r)} = \mathcal{A}_{\text{discovery}} \cup \mathcal{A}_{\text{recon}} \cup \mathcal{A}_{\text{privesc}} \cup \mathcal{A}_{\text{impact}} \tag{4}$$

where:

- $\mathcal{A}_{\text{discovery}} = \{\text{ping-sweep}, \text{port-scan}, \text{network-scan}\}$
- $\mathcal{A}_{\text{recon}} = \{\text{gather-host-info}, \text{enumerate-services}, \text{identify-vulns}\}$
- $\mathcal{A}_{\text{privesc}} = \{\text{exploit-vulnerability}, \text{lateral-movement}, \text{escalate-privileges}\}$
- $\mathcal{A}_{\text{impact}} = \{\text{data-exfiltration}, \text{service-disruption}, \text{system-compromise}\}$

Each action is parameterized by a target host $h \in H$, giving total action space size $|\mathcal{A}^{(r)}| = 12 \times |H|$.

1.2.3 Reward Function

The red agent receives rewards based on successful attack progression and impact on network assets:

$$R_{t,h}^{(r)} = \sum_i \alpha_i \cdot \mathbf{1}[\text{technique}_i \text{ successful}] + \beta \cdot |\text{assets compromised}| - \lambda \cdot \mathbf{1}[\text{detected}] \tag{5}$$

where $\alpha_i > 0$ are rewards for successful technique execution, $\beta > 0$ rewards asset compromise, and $\lambda > 0$ penalizes detection by blue team defenses.

1.3 Blue Agent (Defender)

The blue agent implements defensive strategies focused on cyber deception and network isolation to counter red agent attacks.

1.3.1 State Space

The blue agent's state space $\mathcal{S}^{(b)} \subset \mathbb{R}^{d_b}$ is a d_b-dimensional vector with dual structure:

$$S_{t,h}^{(b)} = \begin{bmatrix} \text{alerts}_{t,h}^{\text{current}} \\ \text{alerts}_{t,h}^{\text{history}} \\ \text{decoys}_{t,h} \\ \text{metadata}_{t,h} \end{bmatrix} \tag{6}$$

where:

- $\text{alerts}_{t,h}^{\text{current}} \in \{0,1\}^{|H|}$ encodes current timestep alerts for each host
- $\text{alerts}_{t,h}^{\text{history}} \in \{0,1\}^{|H|}$ maintains cumulative alert history (sticky memory)
- $\text{decoys}_{t,h} \in \{0,1\}^{|H|}$ indicates current decoy host deployments
- $\text{metadata}_{t,h} \in \mathbb{R}^2$ contains [padding constant, total decoy count]

The total dimensionality is $d_b = 3|H| + 2$. The dual alert structure allows the agent to react to immediate threats while learning long-term attack patterns.

1.3.2 Action Space

The blue agent's action space $\mathcal{A}^{(b)}$ consists of defensive actions across network subnets:

$$\mathcal{A}^{(b)} = \mathcal{A}_{\text{deploy}} \cup \mathcal{A}_{\text{remove}} \cup \mathcal{A}_{\text{isolate}} \cup \{\text{nothing}\} \tag{7}$$

where:

- $\mathcal{A}_{\text{deploy}} = \{(\text{deploy}, s_j, d_k) : s_j \in S, d_k \in \mathcal{D}\}$ deploys decoy type d_k on subnet s_j
- $\mathcal{A}_{\text{remove}} = \{(\text{remove}, s_j, d_k) : s_j \in S, d_k \in \mathcal{D}\}$ removes decoy from subnet
- $\mathcal{A}_{\text{isolate}} = \{(\text{isolate}, h_i) : h_i \in H\}$ isolates compromised host h_i
- nothing represents taking no defensive action

The total action space size is $|\mathcal{A}^{(b)}| = 2|S||\mathcal{D}| + |H| + 1$, where $|S|$ is the number of subnets and $|\mathcal{D}|$ is the number of decoy types.

1.3.3 Reward Function

The blue agent reward emphasizes successful deception and asset protection:

$$R_{t,h}^{(b)} = R_{\text{deception}} + R_{\text{protection}} + R_{\text{cost}} \tag{8}$$

where:

$$R_{\text{deception}} = \begin{cases} 10 \cdot |R_{\text{red}}^{\text{base}}| & \text{if red attacks decoy successfully} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

$$R_{\text{protection}} = \begin{cases} -|R_{\text{red}}^{\text{base}}| & \text{if red attacks real host successfully} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

$$R_{\text{cost}} = -c_{\text{deploy}} \cdot N_{\text{decoys}} - c_{\text{maintain}} \cdot \sum_i \text{decoy}_i \tag{11}$$

The 10× multiplier for successful deception creates strong incentives for effective decoy placement, while deployment and maintenance costs prevent trivial strategies.

1.4 Distribution of state transitions and rewards

The environment state transitions are governed by the joint actions of both agents and the underlying network dynamics. Let $\mathcal{N}_{t,h}$ denote the complete network state at time (t,h), including host compromise status, active decoys, and network topology.

The transition probability is defined as:

$$\mathbb{P}(\mathcal{N}_{t,h+1}, S_{t,h+1}^{(r)}, S_{t,h+1}^{(b)} \mid \mathcal{N}_{t,h}, S_{t,h}^{(r)}, S_{t,h}^{(b)}, A_{t,h}^{(r)}, A_{t,h}^{(b)}) \tag{12}$$

This can be decomposed as:

$$\mathbb{P}(\mathcal{N}_{t,h+1} \mid \mathcal{N}_{t,h}, A_{t,h}^{(r)}, A_{t,h}^{(b)}) \cdot \tag{13}$$

$$\mathbb{P}(S_{t,h+1}^{(r)} \mid \mathcal{N}_{t,h+1}, S_{t,h}^{(r)}, A_{t,h}^{(r)}) \cdot \tag{14}$$

$$\mathbb{P}(S_{t,h+1}^{(b)} \mid \mathcal{N}_{t,h+1}, S_{t,h}^{(b)}, A_{t,h}^{(r)}, A_{t,h}^{(b)}) \tag{15}$$

The network state transitions are deterministic given the actions: - Red actions modify host compromise status based on vulnerability exploitation - Blue actions add/remove decoys and modify network isolation - Alert generation follows probabilistic detection models based on MITRE ATT&CK techniques

2 Algorithm

An algorithm takes in the complete interaction history and outputs a policy distribution over next actions. We define the history at time (t,h) as:

$$\mathcal{H}_{t,h} = \{(S_{t',h'}^{(r)}, A_{t',h'}^{(r)}, S_{t',h'}^{(b)}, A_{t',h'}^{(b)}, R_{t',h'}^{(r)}, R_{t',h'}^{(b)})\}_{(t',h')<(t,h)} \tag{16}$$

2.1 Red Agent

2.1.1 Baseline: Deterministic Kill-Chain Agent

The baseline red agent follows a deterministic policy based on the current kill-chain phase:

2.1.2 Adaptive Campaign Agent

An enhanced red agent that adapts strategy based on observed blue agent behavior:

Algorithm 1 Deterministic Red Agent Policy

Input: Current state $S_{t,h}^{(r)}$, network knowledge
Extract current phase ϕ and position p from state
if $\phi =$ discovery **then**
    Select ping-sweep or port-scan action on current subnet
    if sufficient hosts discovered **then**
        Transition to reconnaissance phase
    end if
else if $\phi =$ reconnaissance **then**
    Gather information on discovered hosts
    if vulnerable server found **then**
        Transition to privilege-escalation phase
    end if
else if $\phi =$ privilege-escalation **then**
    Attempt lateral movement to server
    if server compromised **then**
        Transition to impact phase
    end if
else if $\phi =$ impact **then**
    Execute impact actions on compromised servers
end if
return Action $A_{t,h}^{(r)}$

$$\pi^{(r)}(a \mid s, \mathcal{H}) = \text{softmax}(\beta \cdot Q^{(r)}(s,a) + \alpha \cdot \text{adaptation\_bonus}(a, \mathcal{H})) \tag{17}$$

where adaptation_bonus increases probability of actions that counter observed blue patterns.

2.2 Blue Agent

2.2.1 Baseline: Random Decoy Placement

The baseline blue agent randomly deploys decoys with uniform probability across subnets:

$$\pi_{\text{baseline}}^{(b)}(a \mid s) = \begin{cases} \frac{1}{|S||\mathcal{D}|} & \text{if } a \in \mathcal{A}_{\text{deploy}} \\ 0.1 & \text{if } a = \text{nothing} \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

2.2.2 PPO Algorithm

The main blue agent is trained using Proximal Policy Optimization (PPO) with the following objective:

$$L^{\text{PPO}}(\theta) = \mathbb{E}_t \left[\min \left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right] \tag{19}$$

where:

- $r_t(\theta) = \frac{\pi_\theta(A_{t,h}^{(b)} | S_{t,h}^{(b)})}{\pi_{\theta_{\text{old}}}(A_{t,h}^{(b)} | S_{t,h}^{(b)})}$ is the probability ratio

- $\hat{A}_t$ is the generalized advantage estimate
- $\epsilon = 0.2$ is the clipping parameter

The advantage is computed using Generalized Advantage Estimation (GAE):

$$\hat{A}_{t,h} = \sum_{l=0}^{H-h} (\gamma\lambda)^l \delta_{t,h+l} \tag{20}$$

where $\delta_{t,h} = R_{t,h}^{(b)} + \gamma V(S_{t,h+1}^{(b)}) - V(S_{t,h}^{(b)})$ and $\lambda = 0.95$ is the GAE parameter.

Algorithm 2 PPO Training for Blue Agent

```
Phase 1 – Experience Collection
for t = 1 to T do
    for h = 1 to H do
        Observe state S_{t,h}^{(b)}
        Sample action A_{t,h}^{(b)} ~ π_θ(S_{t,h}^{(b)})
        Execute action and observe reward R_{t,h}^{(b)}
        Store transition (S_{t,h}^{(b)}, A_{t,h}^{(b)}, R_{t,h}^{(b)}, S_{t,h+1}^{(b)})
    end for
end for
Phase 2 – Advantage Computation
Compute advantages {Â_{t,h}} using GAE
Compute returns {R_{t,h}^{total}}
Phase 3 – Policy Update
for k = 1 to K epochs do
    Compute PPO loss L^{PPO}(θ)
    Update parameters θ ← θ − α∇_θ L^{PPO}(θ)
end for
```

3 Evaluation

We define several key evaluation metrics to assess the performance of blue agent policies and the overall security of the defended network.

3.1 Primary Security Metrics

3.1.1 Deception Effectiveness

The rate at which red agents are successfully deceived into attacking honeypots:

$$\text{Deception Rate} = \frac{\sum_{t,h} \mathbf{1}[\text{red attacks decoy at } (t,h)]}{\sum_{t,h} \mathbf{1}[\text{red attacks any host at } (t,h)]} \tag{21}$$

3.1.2 Asset Protection

The fraction of real network assets that remain uncompromised:

$$\text{Protection Rate} = \frac{|H_{\text{real}}| - |\{h \in H_{\text{real}} : \text{compromised}(h)\}|}{|H_{\text{real}}|} \tag{22}$$

where H_{real} is the set of non-decoy hosts.

3.1.3 Attack Detection Latency

The expected time between attack initiation and blue agent awareness:

$$\text{Detection Latency} = \mathbb{E}\left[\min_h \{h : \text{alert generated at timestep } h\} - \text{attack start time}\right] \tag{23}$$

3.2 Operational Metrics

3.2.1 Resource Efficiency

The effectiveness of defensive resource allocation:

$$\text{Resource Efficiency} = \frac{\text{Successful Deceptions}}{|\text{Active Decoys}| + c \cdot |\text{Isolation Actions}|} \tag{24}$$

where $c > 0$ weights the cost of isolation actions relative to decoy maintenance.

3.2.2 False Positive Rate

The rate of false alerts generated by detection systems:

$$\text{False Positive Rate} = \frac{\sum_{t,h} \mathbf{1}[\text{false alert at } (t,h)]}{\sum_{t,h} \mathbf{1}[\text{any alert at } (t,h)]} \tag{25}$$

3.3 Strategic Metrics

3.3.1 Total Expected Reward

The fundamental RL objective for both agents:

$$J^{(b)} = \mathbb{E}\left[\sum_{t=1}^{T} \sum_{h=1}^{H} \gamma^{h-1} R_{t,h}^{(b)}\right] \tag{26}$$

$$J^{(r)} = \mathbb{E}\left[\sum_{t=1}^{T} \sum_{h=1}^{H} \gamma^{h-1} R_{t,h}^{(r)}\right] \tag{27}$$

3.3.2 Attack Success Rate

The fraction of attempted attacks that achieve their intended effect:

$$\text{Attack Success Rate} = \frac{\sum_{t,h} \mathbf{1}[\text{red action successful at } (t,h)]}{\sum_{t,h} \mathbf{1}[\text{red action attempted at } (t,h)]} \tag{28}$$

3.3.3 Strategic Adaptation Index

A measure of how well the blue agent adapts to changing red agent strategies:

$$\text{Adaptation Index} = \frac{\text{Performance in final 10\% episodes}}{\text{Performance in first 10\% episodes}} \tag{29}$$

where performance is measured by deception rate or protection rate.

3.4 Network-Specific Metrics

3.4.1 Coverage Quality

The strategic value of decoy placement across network topology:

$$\text{Coverage Quality} = \sum_{s \in S} w_s \cdot \frac{\text{decoys in subnet } s}{\text{total hosts in subnet } s} \tag{30}$$

where w_s represents the strategic importance weight of subnet s.

3.4.2 Mean Time to Compromise (MTTC)

Expected time for red agent to achieve primary objectives:

$$\text{MTTC} = \mathbb{E}\left[\min_h \{h : \text{critical asset compromised at timestep } h\}\right] \tag{31}$$

These metrics provide a comprehensive evaluation framework for comparing blue agent policies and assessing the security posture of defended networks under various attack scenarios.