Computer Modeling in Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2024.052375

ARTICLE

Optimal Cyber Attack Strategy Using Reinforcement Learning Based on Common Vulnerability Scoring System

Bum-Sok Kim[1], Hye-Won Suk[1], Yong-Hoon Choi[2], Dae-Sung Moon[3] and Min-Suk Kim[2,*]

[1]Department of Electronic Information System Engineering, Sangmyung University, Cheonan, 31066, Republic of Korea

[2]Department of Human Intelligence & Robot Engineering, Sangmyung University, Cheonan, 31066, Republic of Korea

[3]Intelligent Convergence Research Laboratory, Electronics and Telecommunications Research Institute, Daejeon, 34129, Republic of Korea

*Corresponding Author: Min-Suk Kim. Email: minsuk.kim@smu.ac.kr

Received: 31 March 2024 Accepted: 15 August 2024 Published: 27 September 2024

ABSTRACT

Currently, cybersecurity threats such as data breaches and phishing have been on the rise due to the many different attack strategies of cyber attackers, significantly increasing risks to individuals and organizations. Traditional security technologies such as intrusion detection have been developed to respond to these cyber threats. Recently, advanced integrated cybersecurity that incorporates Artificial Intelligence has been the focus. In this paper, we propose a response strategy using a reinforcement-learning-based cyber-attack-defense simulation tool to address continuously evolving cyber threats. Additionally, we have implemented an effective reinforcement-learning-based cyber-attack scenario using Cyber Battle Simulation, which is a cyber-attack-defense simulator. This scenario involves important security components such as node value, cost, firewalls, and services. Furthermore, we applied a new vulnerability assessment method based on the Common Vulnerability Scoring System. This approach can design an optimal attack strategy by considering the importance of attack goals, which helps in developing more effective response strategies. These attack strategies are evaluated by comparing their performance using a variety of Reinforcement Learning methods. The experimental results show that RL models demonstrate improved learning performance with the proposed attack strategy compared to the original strategies. In particular, the success rate of the Advantage Actor-Critic-based attack strategy improved by 5.04 percentage points, reaching 10.17%, which represents an impressive 98.24% increase over the original scenario. Consequently, the proposed method can enhance security and risk management capabilities in cyber environments, improving the efficiency of security management and significantly contributing to the development of security systems.

KEYWORDS

Reinforcement learning; common vulnerability scoring system; cyber attack; cyber battle simulation

1 Introduction

The advancement of Artificial Intelligence (AI) has involved innovation and convenience in our lives and diverse industrial sectors, but it also created new challenges in cybersecurity [1,2]. With AI being employed for processing and managing sensitive data in cyberspace, the risk of encountering diverse security challenges such as data leakage, privacy breaches, and identity theft is increasing [3,4]. These problems lead to unauthorized access and leakage of personal and corporate information, posing a serious threat to organizations and creating new opportunities for cyber attackers [5,6].

In this evolving threat environment, traditional cybersecurity has revealed its constraints [7,8]. Attack detection approaches based on fixed data and supervised learning have been constrained in effectively addressing the complexity of continuously evolving attack types [9,10]. In this paper, we use Reinforcement Learning (RL) to develop adaptive attack strategies capable of responding dynamically to evolving threats. These strategies are simulated and evaluated using RL-based cyber-attack-defense simulation tools like Cyber Battle Simulation (CyberBattleSim), developed by Microsoft's Defender team [11,12]. CyberBattleSim is a useful and testable cyber-attack-defense simulation tool that accurately simulates real-world situations. It can also serve as a robust training platform that facilitates the red and blue team dynamics and enables continuous interaction with the environment. Integration of these dynamics is important for applying RL, as it effectively simulates real-world cybersecurity training scenarios, enabling the optimization of strategies that can adapt to evolving threats. This environment not only enhances the development and evaluation of RL-based strategies but also facilitates the training of behavioral strategies of attacker agents. It is also offering methods to effectively counter unknown threats and new types of attacks [13]. These strategies are essential for devising adaptive responses within a complex cyber threat environment [14].

In the rapidly evolving cyber threats, RL-based cyber-attack-defense simulation techniques are being investigated as a new approach to overcome the limitations of traditional security methods [15,16]. As a result, with the expansion of cyberspace and the increasing complexity of attacker's Tactics, Techniques, and Procedures (TTPs), attack-defense simulations are becoming essential tools for responding to diverse cyber-attacks [17]. The attack-defense simulations aim to mimic the TTPs of cyber attackers. By simulating cyber-attack scenarios resembling real-world situations, they can professionally assist security in developing strategies to effectively counter cyber-attacks [18]. However, such advanced training demands the expertise of cybersecurity professionals and entails considerable time and cost [19]. Additionally, the complexity of cybersecurity training is further increased due to the constraints and risks of real-world training environments [20].

The purpose of the paper is to develop and evaluate RL-based cyber-attack strategies using CyberBattleSim to effectively respond to cyber in the real world. CyberBattleSim provides a dynamic simulation environment for testing attack scenarios, allowing for a comprehensive evaluation of the generated strategies. In this process, we applied a new vulnerability assessment method based on the Common Vulnerability Scoring System (CVSS) to optimize attack strategies. This method enables tailored cost allocations for vulnerabilities, enhancing risk management and the prioritization of security measures. In this paper, we develop the 'ToyCTF (Capture the Flag) Alpha' scenario, which is an extension of the original 'ToyCTF' scenario provided by CyberBattleSim, to improve learning performance by analyzing vulnerabilities and scenario information of specific nodes. In addition, RL-based off-policy algorithms such as Q-Learning [21], Deep Q-Network (DQN) [22], and Dueling Deep Q-Network (DDQN) [23] and on-policy algorithms such as Advantage Actor-Critic (A2C) [24], Proximal Policy Optimization (PPO) [25], and REINFORCE [26] were selected to evaluate their effectiveness in predicting attack strategies within the CyberBattleSim environment.

2 Background and Related Work

2.1 Cybersecurity Challenges and Machine Learning

The field of cybersecurity is faced with complex and diverse challenges due to the rapid evolution of technology [27]. However, traditional rule-based security methods that depend on fixed data and predefined scenarios have shown limitations in adapting to the changing threat environment [28,29]. In particular, methods such as intrusion detection systems and antivirus tools using static file scanning are effective against known threats but struggle to counter new or modified attack patterns [30,31]. With advancements in digital technology, cyber-attack methods such as custom malware, social engineering attacks, and Distributed Denial of Service (DDoS) have become more sophisticated, capable of bypassing traditional security systems [32,33]. In particular, the advent of cloud computing and IoT devices has expanded the scope of security attacks, exposing organizations to new threats [34]. This necessitates the enhancement of cybersecurity strategies and the development of more dynamic and adaptive security solutions [35].

For that reason, advanced technologies such as machine learning (ML) have begun to be integrated into cybersecurity approaches, providing new effective approaches in an ever-changing threat environment [36]. Supervised ML has been used to train models to distinguish between malicious and normal behavior using labeled data, outperforming traditional methods in detecting unknown attacks and variants of attack patterns [37–39]. However, supervised ML still struggles to adapt quickly to new and evolving threats and there are limitations in its application to dynamic environments.

Recently, RL approaches garnered significant interest in overcoming these limitations [40,41]. RL is trained through continuous interaction with dynamic environments, developing strategies that adapt to unknown threats and changing attack patterns [42]. RL is an adaptive method to evaluate strategies in a realistic environment, and it can develop better response strategies for dynamic environments such as the effectiveness of cybersecurity systems. However, RL relies largely on underlying decision-making models based on the traditional Markov Decision Process (MDP), which has limitations in reflecting the uncertainty and complexity of real-world security environments [43]. Therefore, extending MDP methods to better capture various aspects of cybersecurity is essential. This will enable RL to respond more effectively to dynamic cybersecurity threats.

2.2 Cybersecurity Simulation Environment Based on Reinforcement Learning

Recently, cyber-simulation environments rely primarily on hand-crafted scenarios by experts, limiting the scalability and adaptability of training modules to different expertise levels and objectives [44]. These environments cannot dynamically and autonomously generate scenarios that accurately reflect the diverse requirements of real-world operations. Thus, there is a demand for useful and testable simulation environments that can automatically generate and adapt cyberattack scenarios to the evolving characteristics of cybersecurity threats [45]. These improvements reduce reliance on manual scenario creation and increase the authenticity and variability of the training environment.

In response to these challenges, RL-based methods are increasingly being used in the field of cybersecurity, particularly within simulation environments designed to model complex cyber-attack-defense scenarios [46]. RL-based cyber simulators, such as Network Attack Simulation (NASim) and CyberBattleSim, play an important role in simulating real-time network attacks and defense scenarios in complex network environments. Specifically, NASim allows users to configure and manipulate virtual computer networks to simulate various network attack scenarios but focuses primarily on attack strategies, lacking comprehensive defensive tactics [47–49]. Conversely, CyberBattleSim can simulate both attack and defense strategies in complex network environments using RL [50]. This tool enables the observation of the network's real-time response and the optimization of strategies through continuous learning, but it requires significant computational resources and expert knowledge to set up and maintain [51].

Integrating intelligent simulation environments such as NASim and CyberBattleSim can provide a more realistic and comprehensive educational experience. These are not only capable of modeling complex scenarios but also support the iterative testing and refinement of cyber defense tactics, providing important tools in the development of robust defense mechanisms against existing and emerging cyber threats [52].

3 Cyber Range Simulation Environment

CyberBattleSim is a simulation framework based on Open AI Gym that designs network environments and vulnerabilities, providing an essential tool for cybersecurity training [11]. This framework provides realistic cyber-attack/defense scenarios with a simulated environment that allows attacker and defender agents to strategize within a real-world network. The RL agent can be trained using the default scenarios or constructing new ones. In this section, we proposed an extended cyber-attack simulating scenario using CyberBattleSim in a newly developed cyber-attack scenario to provide a more effective RL-based cyber threat environment.

3.1 Scenario of ToyCTF Alpha

'ToyCTF' scenario provided by CyberBattleSim is based on the concept of Capture The Flag (CTF) and is designed to engage security professionals in strategic planning while owning and defending a range of nodes in a competitive environment [11]. This is simulated in security vulnerabilities and types of attacks through diverse network interactions including Know, Remote Exploit, and Lateral Movement reflecting a realistic security environment. However, the 'ToyCTF' has several limitations such as firewall settings on some nodes allowing excessive access and assigning the same cost to all vulnerabilities.

In this paper, we propose a new scenario, 'ToyCTF Alpha' which extends the original 'ToyCTF' scenario by incorporating CyberBattleSim to overcome its limitations. CyberBattleSim generally enables the simulation of attack strategies, which allows for a comprehensive evaluation of the RL agent's learning performance and adaptability. The 'ToyCTF Alpha' can provide an enhanced network environment compared to the original 'ToyCTF' scenario. As shown in Fig. 1, the network structure of the scenario with network topology is presented. While some node components remain the same across both scenarios, significant differences are found in enhancing the scenarios' realism and improving the RL agent's learning performance. The proposed scenario can overcome the limitations of the 'ToyCTF' by more comprehensively reflecting a range of vulnerabilities and types of attacks expected in a realistic security environment, while still incorporating the basic concept of 'ToyCTF' and providing advanced security strategies.

'ToyCTF Alpha' incorporates several improvements for effective learning over traditional simulation environments. First, we addressed the issue of excessive access permissions observed in the original 'ToyCTF' scenario by readjusting the firewall settings across all nodes, thus strictly limiting access to essential services such as Hypertext Transfer Protocol Secure (HTTPS), Secure Shell (SSH), and Global Information Tracker (GIT). These firewall settings create constraints that allow agents to strategically navigate and exploit the environment for the learning process. Second, we applied vulnerability assessment methods based on the CVSS [53,54] for more effective cost allocation within the 'ToyCTF Alpha' scenario, aiming to improve the RL-based learning process. This method distinguishes between the severity of vulnerabilities and the complexity of the attack on each node, allowing us to assign a tailored cost to each vulnerability based on its CVSS score. This adjustment addresses a problem in 'ToyCTF' scenarios, where all vulnerabilities were treated with the same level of risk, disregarding their severity or complexity, due to the same cost setting for all vulnerabilities. The cost represents the resources an agent spends to take an action, while the value represents the rewards gained from successfully attacking and gaining control of a node. These metrics are essential for the agent to learn optimal policies that prevent unnecessary actions and improve the overall effectiveness of the cyber-attack strategy. Finally, we implement dynamic simulations similar to real-world environments by assigning different values to each node based on their strategic importance. Node values influence the agent's prioritization process, guiding it towards more critical nodes. Action costs motivate the agent to optimize its resource usage and avoid high-cost actions. Services running on the nodes determine potential attack vectors and the complexity of compromising each node in the simulation. RL models can simulate a more precise and effective cyber-attack strategy, reflecting cybersecurity challenges and responses. In Table 1, the detailed node information and vulnerability assessment results are presented. The cost and value metrics for reward represent the assessment of nodes based on their CVSS scores. This is vital for RL agents to navigate and strategize effectively in a constrained environment.

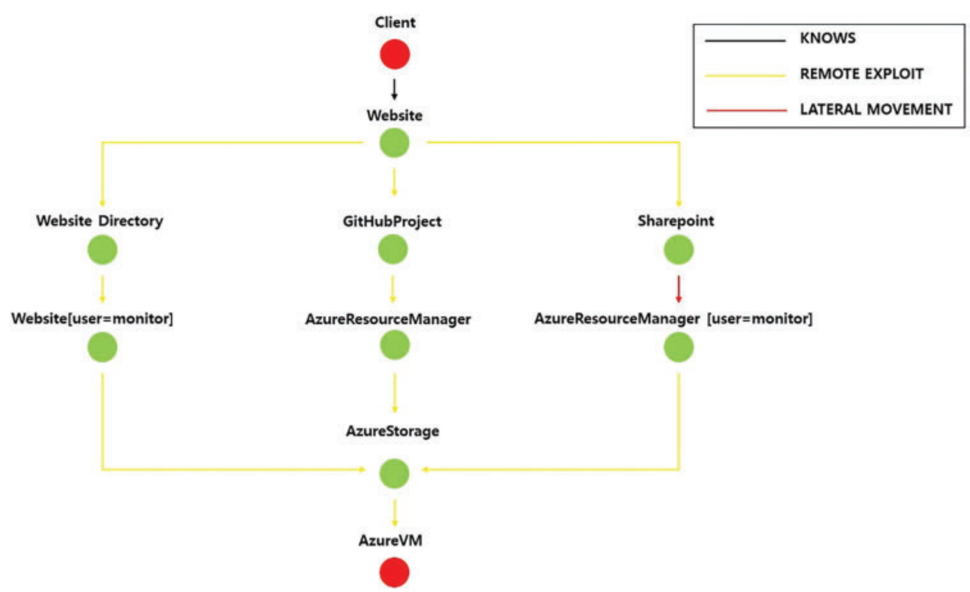

Figure 1: Network topology in ToyCTF Alpha

Table 1: Node components in ToyCTF Alpha

	Component				
Node	Service	Property	Firewall	Cost	Value
Client	None	None	None	1.5	0

(Continued)

Table 1 (continued)

		Component			
Node	Service	Property	Firewall	Cost	Value
Website	HTTPS SSH	MySql Ubuntu nginx/1.10.3	Incoming: SSH (ALLOW), HTTPS (ALLOW) Outgoing: default allow rules	1.5	135
WebsiteDirectory	HTTPS GIT	GitHub SasUrlInCommit	Incoming: GIT (ALLOW) Outgoing: default allow rules	2.0	160
Website [user=monitor]	HTTPS	Sharepoint-LeakingPassword	Incoming: HTTPS (ALLOW) Outgoing: default allow rules	2.0	130
GitHubProject	HTTPS SSH	Ubuntu nginx/1.10.3 CTFLAG: Readme.txt	Incoming: HTTPS (ALLOW) Outgoing: default allow rules	2.0	150
AzureStorage	HTTPS	MySql Ubuntu nginx/1.10.3	Incoming: HTTPS (ALLOW) Outgoing: default allow rules	2.0	140
Sharepoint	HTTPS SSH	CTFFLAG: LeakedCustomerData2	Incoming: HTTPS (ALLOW) Outgoing: default allow rules	1.5	155
AzureResourceManager	HTTPS	SensitiveAzureOp-erations	Incoming: HTTPS (ALLOW) Outgoing: default allow rules	2.0	140
AzureResoureManager [user=monitor]	HTTPS	CTFFLAG: VMPRIVATEINFO	Incoming: HTTPS (ALLOW) Outgoing: default allow rules	1.5	145
AzureVM	SSH	CTFFLAG: LeakedCustomer-Data	Incoming: SSH (BLOCK) Outgoing: default allow rules	2.0	140

As a consequence, the attacker agent's penetration process was designed in more detail by simulating the attacker's diverse strategies and behaviors from initial access to ultimate information theft. The penetration strategy of the attacker is depicted in Fig. 2, which presents a flowchart outlining the progression from phishing attack to sensitive data exfiltration. Algorithm 1 provides the corresponding pseudo-code, detailing each step of the process. In the initial stage, the attacker gains entry through a phishing attack to acquire user credentials. After gaining access, an attacker can use it to explore and exploit system vulnerabilities, starting with the 'Website' node to gather information within the system and expanding to the 'Website.Directory' node, 'GitHubProject' node, and 'Sharepoint' node. Afterward, in the advanced penetration phase, the attacker targets the 'AzureResourceManager' and 'AzureStorage' nodes to exfiltrate sensitive data [11]. In this process, the attacker exploits the 'DirectoryTraversal' vulnerability in the 'AzureResourceManager' node to gain vital credentials, which enable access to a broader range of Azure resources. Meanwhile, the 'AzureStorage' node is targeted through the 'InsecureBlobStorage' vulnerability, which enables attackers to expose sensitive stored data. Then, the attacker combines the data with sensitive information stored on the 'AzureVM' node to execute the exfiltration. In the end, the attacker secures privileged access to the 'AzureVM' nodes by exploiting the 'UnpatchedSSHService' vulnerability. This allows the attacker to access virtual machine instances within the cloud service and leak sensitive information.

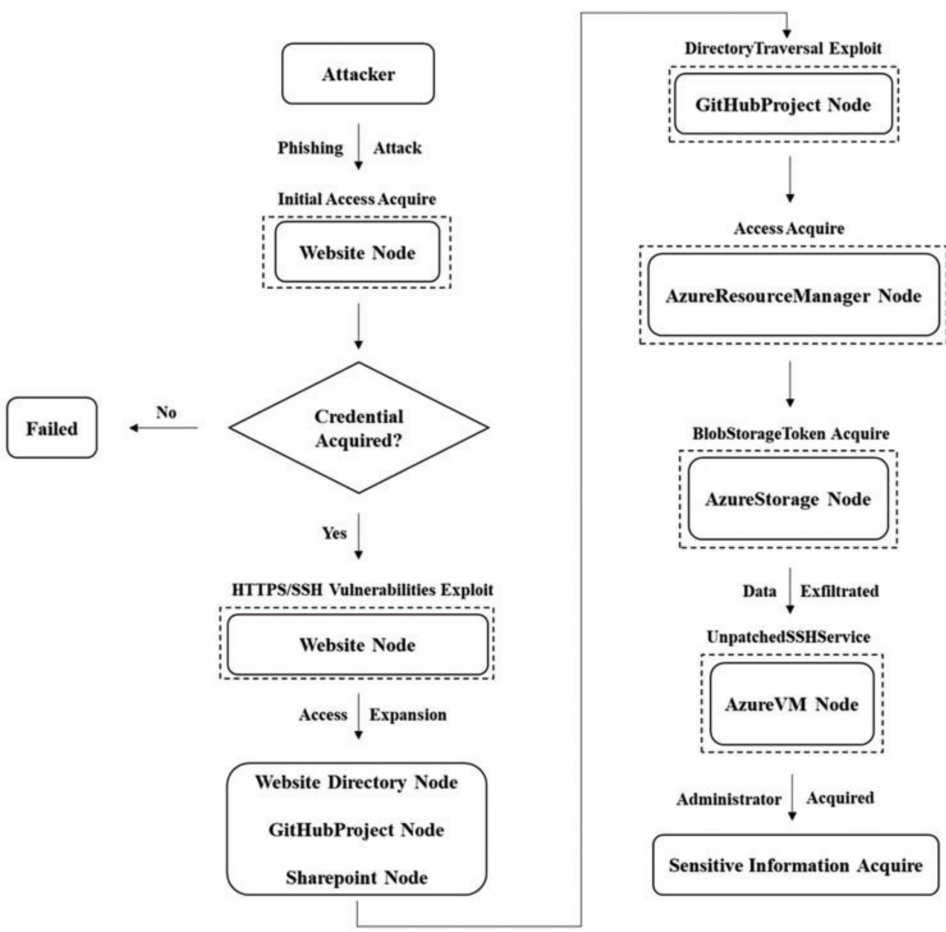

Figure 2: Flowchart of the penetration process in ToyCTF Alpha

Algorithm 1: Penetration process for ToyCTF Alpha

```
Input:  Initialize the network nodes, vulnerabilities
Output: Define Attack outcomes report
Set nodes to predefined vulnerabilities
privileged_access ← false
data_exfiltration ← false
for i ← 1 to 200,000 or success do
    Attempt phishing attack on 'Website'
    if success then
        privileged_access ← true
/* Initial Access
if privileged_access then
  if explore 'Website' then
    Scan Node
    GU ← GitHub URL
    DP ← Directory Paths
    SCE ← SSH Credentials Exploitation
    Node.info ← [GU, DP, SCE]
/* Information Gathering
while data_exfiltration = false do
  if access each node in ['Website.Directory', 'GitHubProject', 'Sharepoint'] then
    Retrieve data from .git directories or other sensitive sources
    Use SCE to bypass authentication
    Exploit directory traversal vulnerabilities for file access
    if 'DirectoryTraversal' is successful at 'GitHubProject' then
      Exploit 'AzureResourceManager':
      Use 'DirectoryTraversal' to obtain 'ResourceManagerAccess'
      Manipulate Azure resources and acquire 'BlobStorageToken'
/* Critical Resource Access
        if 'BlobStorageToken' is acquired then
          Exploit 'AzureStorage' to initiate data leaks
          Exploit SSH vulnerabilities in 'AzureVM' to access sensitive data
          if secure access to 'AzureVM' then
            Exploit unpatched SSH for sensitive data exfiltration
            data_exfiltration ← true
/* Data Exfiltration
    Apply strong authentication if required
/* Access Control
    while exploring nodes do
      Check misconfigured permissions and exposed credentials
      if finding misconfigurations then
        Escalate privileges
        Gain administrative access, particularly through 'Sharepoint'
```

(Continued)

Algorithm 1 (continued)

```
        end
    /∗ Privilege Escalation
    end
```

3.2 *Defining Cost Value Based on CVSS for Enhanced Learning Performance*

This paper aims to assess the cybersecurity risk of the scenario more realistically by utilizing CVSS for vulnerability assessment [55]. CVSS refers to a standardized system for quantifying the severity of vulnerabilities. It is used to assess the characteristics of security vulnerabilities such as confidentiality, integrity, and availability, as well as exploitability and attack complexity. In this paper, the Base Score Metrics from CVSS V2.0 to accurately assess the impact of vulnerabilities is used. In the 'ToyCTF', the cost was set to 1.0 for all vulnerabilities, regardless of severity or attack complexity, resulting in different vulnerabilities being assessed at the same level [11]. This revealed the limitations of risk management and security prioritization in real-world organizations. To overcome this limitation, in the 'ToyCTF Alpha', we divided the Base Score Metrics into Exploitability Metrics and Impact Metrics, then the severity score was calculated by the vulnerability by utilizing Attack Vector, Attack Complexity, Authentication, Confidentiality, Integrity Impact, and Availability Impact [56]. In addition, we assigned costs of 1.0, 1.5, and 2.0 for vulnerabilities rated as Low, Medium, and High respectively to help organizations better prioritize security measures. The approaches can make more precise adjustments in assessing the impact of vulnerabilities by distinguishing between different vulnerabilities, thereby enhancing realism. In this regard, the status of severity changes in CVSS V2.0 and the CVSS score of each vulnerability applied to the 'ToyCTF Alpha', along with their corresponding severity are presented in Tables 2 and 3, respectively.

Table 2: Severity level in CVSS V2.0

CVSS V2.0 score	
Severity	Score range
Low	0.0–3.9
Medium	4.0–6.9
High	7.0–10.0

Table 3: Node vulnerability assessment CVSS score and severity in ToyCTF Alpha

Vulnerability	CVSS score	Severity
ScanPageContent	5.0	Medium
ScanPageSource	5.0	Medium
StrongAuthRequirement	4.9	Medium
ExposedGitHistory	9.4	High
DirectoryTraversal	9.4	High
MisconfiguredPermissions	6.2	Medium

(Continued)

Table 3 (continued)

Vulnerability	CVSS score	Severity
MisconfiguredAccessControl	9.4	High
InsecureAPIEndpoint	9.4	High
PrivilegedOperationsExposure	6.0	Medium
InsecureBlobStorage	7.8	High
UnpatchedSSHService	10.0	High
PhishingVulnerability	6.3	Medium

3.3 Reassessing Node Value in ToyCTF Alpha

In this paper, we propose redefining the calculation of cost values for nodes by utilizing a range of components to redefine the value of a node, as detailed in Section 3.2. The existing 'ToyCTF' scenario cannot guarantee learning stability due to the lack of specific standards for node value. In contrast, reassessing node value can offer improved learning stability compared to the existing environment.

For one of the nodes in the 'ToyCTF', 'AzureVM' is assigned similar or lower values compared to less important nodes, despite being a critical infrastructure component [11]. This means that the value settings of some nodes do not fully reflect the actual importance of those nodes. To reflect the importance of nodes, we applied the value of each node to the 'ToyCTF Alpha', which comprehensively considers a range of components including the service importance of the node, firewall configuration, attribute importance, and identification of security vulnerabilities to enhance realism.

When calculating the final value (FV) of a node in this proposed method, the following equation is used:

$$FV = BV + SI + FC + AI - SV - RV \quad (1)$$

- *BV: Node Base Value*
- *SI: Node Service Importance*
- *FC: Node Firewall configuration*
- *AI: Node Attribute Importance*
- *SV: Node Security Vulnerability*
- *RV: Node Remote Vulnerability*

To determine the value of a node, we use the following step-by-step method. First, all nodes receive a base value (BV) of 100 by default. Next, the value is adjusted by evaluating the service importance (SI) of each node. To prevent each adjustment value from exceeding the importance of BV during the adjustment phase, the adjustment values have a maximum value of 50 by half the BV. Nodes that provide basic services such as HTTP or SSH have an SI value of +50 since they directly impact the accessibility and security of the system. In contrast, nodes that provide specialized services or additional functionality, such as the [monitor] tag, have an SI value of +40 due to a limited impact on the overall network. Subsequently, the value of a node is then adjusted based on the allowed access to the service in the node's firewall configuration (FC). It has an FC value of +10 since the risk to the overall system from a single service is lower than multiple services, and it is simpler to manage. In contrast, nodes that allow access to multiple services are given an FC value of +20 due to the increased complexity and risk associated with security management. Attribute importance (AI) value is calculated based on attributes related to a node's specific functionality or the management of important data. Attributes related to common operating systems or services such as MySQL, Ubuntu, and Nginx have a default AI value of +30 because they are directly related to the basic operation. On the other hand, specialized features or attributes responsible for managing sensitive data, such as GitHub and SensitiveAzureOperations, are assigned an AI value of +20 due to their relatively lower impact on the system compared to general operations. Additional scoring for SI, FC, and a default value based on AI are adjusted to reflect the security impact each node has on the system. Positive values are also added to account for factors that make it harder for attackers to target.

From a security vulnerability (SV), we set an additional score on top of the default value. The score can provide an additional negative value to reflect on each node. If it is an additional security risk or is high in severity, the attacker is more likely to be attacked. By analyzing SVs, the value of the node is lowered according to the severity of each vulnerability. This adjustment further decreases the value of the node when vulnerabilities that pose additional security risks are remotely available. A low-severity vulnerability has an SV value of −10 since it is less likely to be attacked and the damage is limited. Conversely, vulnerabilities with higher severity are assigned SV values of −15, −20, etc., indicating the heightened potential for system damage by an attacker. This allows an attacker to distinguish between vulnerable and non-exploited nodes, thus enhancing their ability to effectively respond to real-world security threats.

Table 4 shows the criteria for value readjustment, aiming to make accurate value judgments for all nodes in the scenario based on diverse components such as service importance, firewall configuration, attribute importance, and security vulnerability identification. The following approach is important for enhancing security levels and enhancing the protection of critical resources against real-world security threats. Moreover, it can accurately reflect a realistic scenario where an attacker would target vulnerable nodes preferentially, thus enhancing the security of the entire system.

Table 4: Calibration based on security component

Component	Condition	Calibration
Service importance	Basic service provision	+50
	Specialized service provision	+40
Firewall configuration	Access allowed for a single service	+10
	Access allowed for multiple services	+20
Attribute importance	General operations and service support	+30
	Special function or data management	+20
Security vulnerability	Penalty for each vulnerability (CVSS severity: Low)	−10
	Penalty for each vulnerability (CVSS severity: Medium)	−15
	Penalty for each vulnerability (CVSS severity: High)	−20
	Remote vulnerability threat	−20

4 Reinforcement-Learning-Based Cyber Attack Strategies

This paper aims to improve attack and defense strategies in complex security environments by using CyberBattleSim based on RL. With CyberBattleSim, adversary agents learn optimal behavioral strategies by interacting with a dynamically changing network environment in real time. This tool is used to simulate the behavior of adversary agents in cyberattack scenarios.

4.1 Markov Decision Process in Cyber Battle Simulation

In this paper, we select CyberBattleSim to develop effective attack and defense strategies based on real-time dynamic network characteristics and vulnerability information. As shown in Fig. 3, CyberBattleSim has a structure that connects the attacker agent's action, state, observation, environment information, and reward [57].

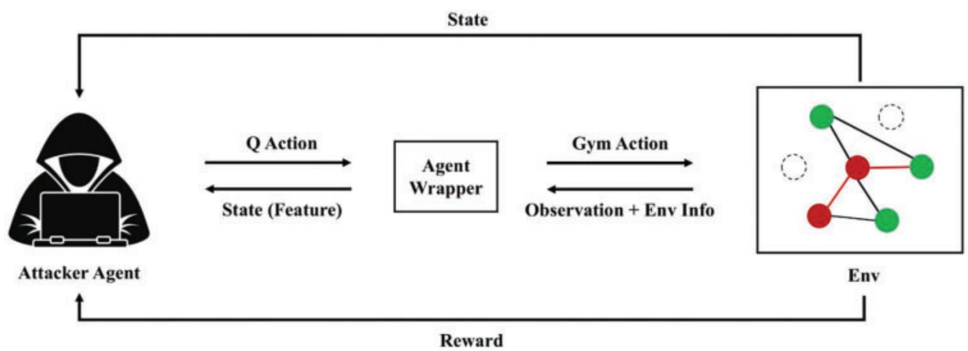

Figure 3: CyberBattleSim architecture overview

The state and action of the attacker agent are passed to the RL-based simulation environment including observation spaces. The results of actions in the environment are returned as rewards used to determine the attacker agent's next action. All of the RL-based interactions are used to continuously improve its strategy. With the environment, the attacker agent to perform a variety of behaviors, including local and remote attacks, network connection attempts, and more.

Table 5 shows the action space of the CyberBattleSim environment. It consists of three main action spaces to select effective actions. Table 6 shows the types of actions in the "ToyCTF Alpha". In this scenario, the agent distinguishes between local and remote attacks to consider strategies for specific states and target nodes. Through strategic planning, the agent performs effective actions in a given scenario, and the success of these actions depends on the state of the agent.

Table 5: Action space in CyberBattleSim

Action space	Required information for execution
Local vulnerability	Source node × local vulnerability to exploit
Remote vulnerability	Source node × target node × remote vulnerability to exploit
Connect	Source node × target node × credential index from cache

Table 6: Attack types in ToyCTF Alpha scenario

Attack type	ToyCTF Alpha scenario
Local attack	PhishingVulnerablity

(Continued)

Table 6 (continued)

Attack type	ToyCTF Alpha scenario
Remote attack	MisconfiguredPermission PrivilegedOperationsExposure ScanPageContent ScanPageSource StrongAuthRequirement ExposedGitHistory MisconfiguredAccessControl DirectoryTraversal InsecureBlobStorage InsecureAPIEndpoint UnpatchedSSHService

As shown in Table 7, the state space consists of a combination of observation space and node information, including information collected by the agent during an episode. According to the state space, the agent continuously adapts its policy to select the optimal action at a particular state in the scenario. The reward is important for the agent to determine its actions. When an agent performs a valid action, it receives a positive reward, while performing an invalid action in a negative reward. The rewards are determined by the CyberBattleSim environment and the CVSS score, which evaluates the vulnerability of the node. We provide a more precise and detailed assessment of node vulnerabilities, resulting in more effective rewards. For invalid actions, a penalty cost derived from the CVSS score is applied, motivating the agent to avoid invalid actions. Thus, RL-based cyber-attack strategies enable agents to effectively respond to diverse cybersecurity threats and play an important role in developing autonomous learning abilities in realistic environments.

Table 7: State space in CyberBattleSim

No.	Observation components	Node info components
1	Discovered node count	Tried at node
2	Owned node count	Active node properties
3	Discovered not owned node count	Active node age
4	Discovered ports sliding	None
5	Discovered node properties sliding	None

4.2 Optimizing Attack Strategies Based on CVSS

The RL in the CyberBattleSim environment is to motivate the agent to efficiently achieve a given goal based on rewards. CyberBattleSim's dynamic and interactive environment allows the RL agent to continuously adapt its strategies in response to evolving threats, thus optimizing attack strategies more effectively. In this paper, we propose the CVSS-based reward method to learn the optimal path in which an agent needs to maximize reward in each state. We also incorporate a more detailed analysis of vulnerability severity and complexity, allowing the agent to prioritize and optimize attack strategies more effectively. In particular, the optimal path can enable the attacking agent to prioritize targeting specific vulnerable nodes in cyber-attacks to optimize the penetration process. Fig. 4 shows the optimal paths discovered by the agent in the learning process. The agent follows the optimal policy represented by the red path, starting from the 'Client' node and sequentially exploring the 'Website', 'Sharepoint', and 'AzureResourceManager' nodes. The decision at each step reflects a strategy that prioritizes attacking the nodes posing the highest threat based on their CVSS scores. The value of each node is assigned weights to optimize the agent's performance and stabilize the learning process.

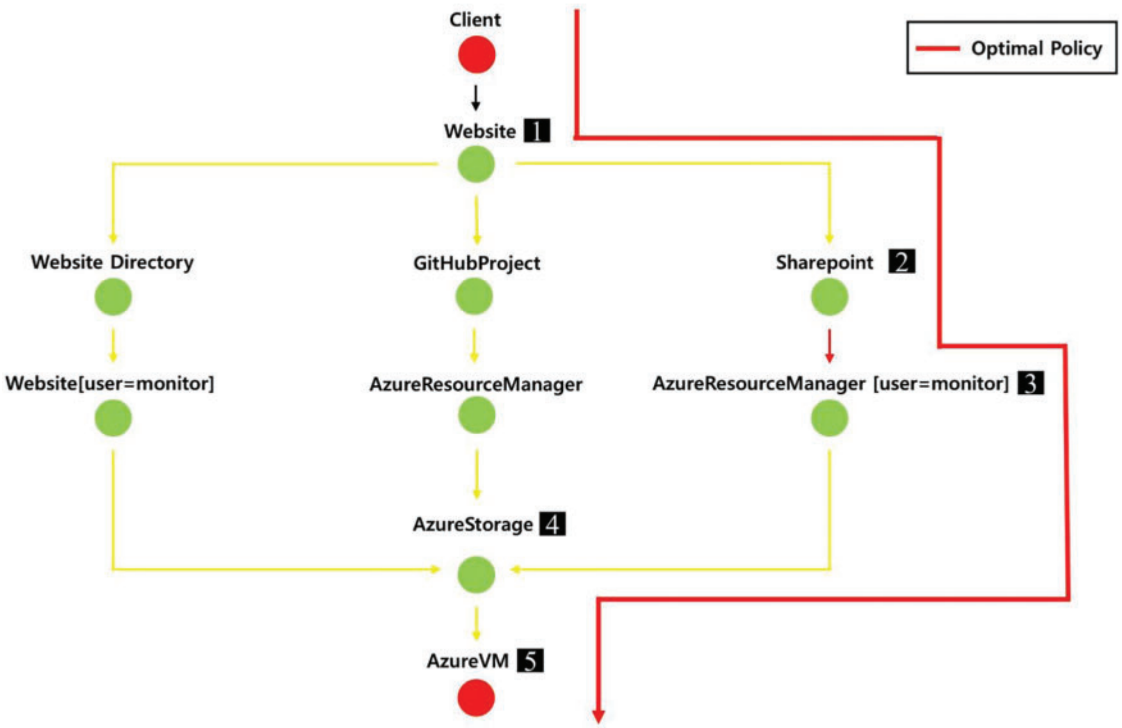

Figure 4: Optimal attack strategy in ToyCTF Alpha

In the 'ToyCTF', the cost has a consistent value of 1 for all local and remote attacks, and the range of negative rewards consists of values between 0 and -50 [11]. The large range of negative rewards accumulates over time steps, limiting the ability to learn optimal attack strategies. In the absence of a negative reward, there is a lack of a baseline to evaluate the validity of agent actions. If the negative rewards range from -50 to 0, it can slow down the calculation or increase the compensation variance, potentially disrupting the learning process. To solve this problem, we create the 'ToyCTF Alpha', and the scenario involves the value and cost to increase the stability of the learning process and to improve the learning performance of the agent. In addition, the CVSS-based reward method enhances learning accuracy for security vulnerability criticality and prioritizes attacking nodes posing the highest threat. Therefore, the agent can discover optimal policies within the simulated environment more rapidly and select the most appropriate action in each situation, enabling it to make decisions to achieve strategic goals in the cyber environment.

5 Experimental Result and Analysis

In this section, we validate the effectiveness of the above-proposed methods and present a comparative analysis of training performance using RL-based off-policy (Q-Learning, DQN, and DDQN) and on-policy (REINFORCE, A2C, and PPO) methods on the proposed scenarios ('ToyCTF Alpha', 'ToyCTF Alpha' with CVSS Reward). The comparative result and analysis also include a moving average graph of the cumulative reward, the time step at the end of the episode, and the success rate. The success rate is a ratio of valid actions to the number of steps performed by the agent in an episode, and it is critical to the comparative analysis since it is directly related to the accuracy of the attack strategies configured by the RL. We also present tabulated average values of the success rate and calculate the percentage improvement. In addition, since the performance of the RL-based methods can vary depending on the environment, we analyzed the learning results to find algorithms with high applicability. Through RL-based methods analysis based on policy update methods, our goal is to further validate the proposed scenario and find a learning method that can be applied efficiently within the cybersecurity environment.

5.1 Case of Scenario without CVSS Reward

In this section, we aim to demonstrate the effectiveness of the newly defined 'ToyCTF Alpha' that was proposed by incorporating the vulnerability assessment method. In Fig. 5, experimental results present a graph of the cumulative rewards for the 'ToyCTF', as provided by CyberBattleSim. REINFORCE, A2C, and DQN can maximize rewards for the optimal policy. In contrast, Q-Learning, DDQN, and PPO produced low reward values.

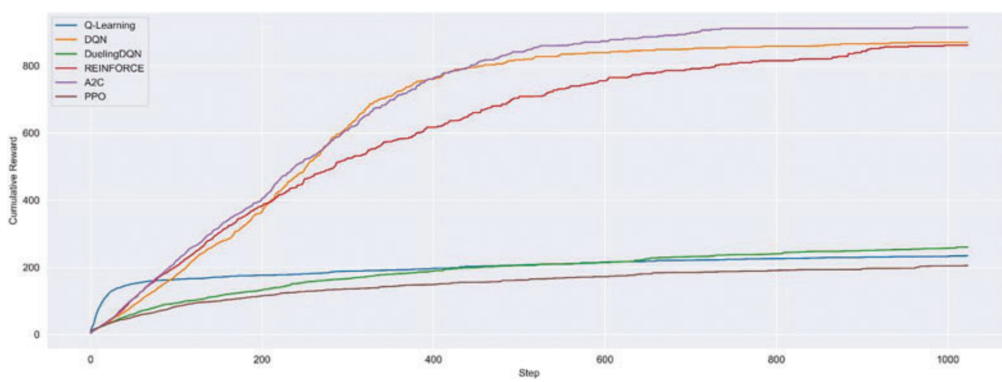

Figure 5: Moving average of cumulative reward in ToyCTF

Fig. 6 shows the number of steps at the end of the episode and the success rate in 'ToyCTF'. In this result, REINFORCE, A2C, and DQN showed a gradual decrease in the number of steps, which means that the RL-based model is stabilizing during the learning process. In contrast, the number of steps Q-Learning, DDQN, and PPO in the success rate graph decreases due to the absence of penalty rewards as explained in the previous Section 4.2. For that reason, the on-policy algorithm fails to learn a policy to avoid erroneous actions.

As shown in Fig. 7, a graph of the cumulative reward for the 'ToyCTF Alpha', proposed by our paper, shows a high reward value for the on-policy algorithms. In contrast, the off-policy algorithms returned low reward values. In particular, the off-policy-based DQN seems to have underperformed since the scenario of giving 0 as a penalty reward did not provide clear guidance for evaluating the validity of actions. However, both Q-Learning and DDQN are more stable in convergence compared to the previous experiments. In contrast, the on-policy algorithms maintained stable performance through the effective current policy. In particular, the PPO in the 'ToyCTF Alpha' demonstrates remarkable experimental performance when compared to the previous 'ToyCTF'.

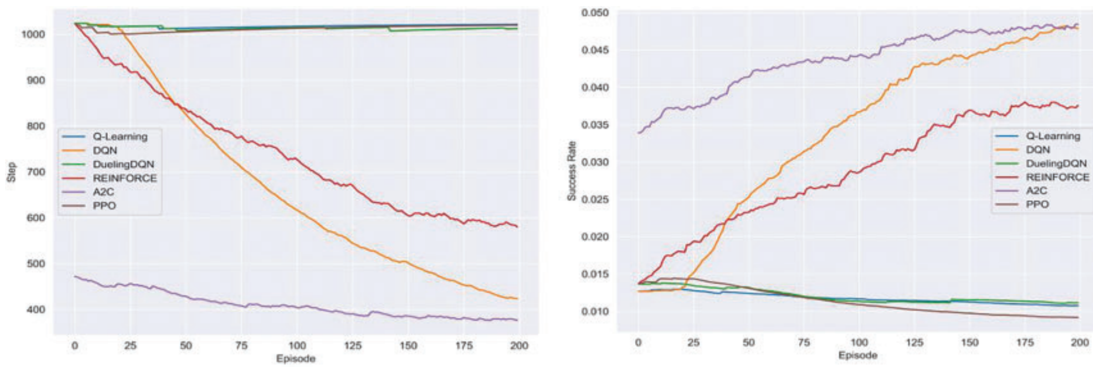

Figure 6: Moving average of steps and success rate in ToyCTF

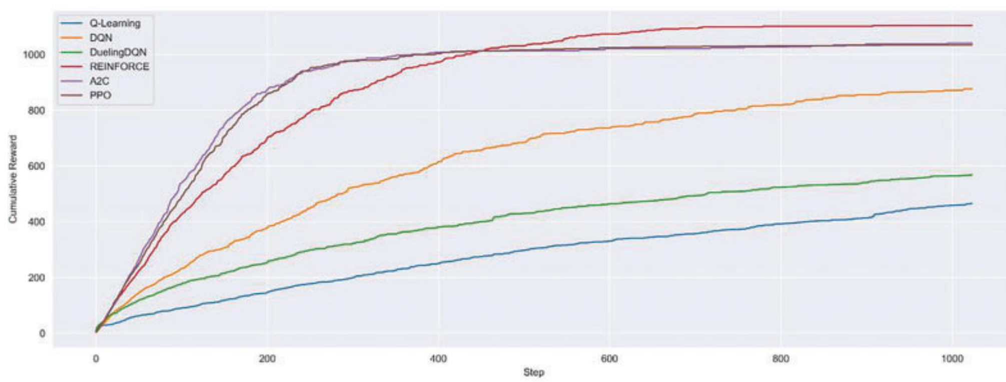

Figure 7: Moving average of cumulative reward in ToyCTF Alpha

In Fig. 8, the result shows the number of steps at the end of the episode and the success rate in 'ToyCTF Alpha'. Similar to the 'ToyCTF' scenario, the penalty rewards at 0 led to instability in the learning process of off-policy algorithms such as REINFORCE and PPO. Although REINFORCE achieved the highest reward value, it performed less than A2C in both the number of steps and the success rate. It appears that REINFORCE owns fewer critical nodes in the process of achieving the goal. The number of steps performed gradually increased for the PPO algorithm, accompanied by a high success rate. As for A2C, lower step values and stable convergence contributed to an improved success rate compared to the previous iteration.

In general, on-policy RL methods demonstrate high rewards and success rates by selecting actions based on the current policy and receiving immediate feedback on dynamic changes in the environment. This approach is excellent in environments characterized by frequent changes and high complexity. In contrast, off-policy RL methods exhibit lower rewards and success rates. These separate the target policy from the behavior policy, utilizing diverse experiential information to learn policies. However, this approach is less efficient in environments with significant variability, leading to slower convergence rates and difficulties in achieving the optimal policy. Therefore, 'ToyCTF Alpha' provides favorable conditions for the on-policy approach, and it can be seen that redefining node values has had a positive impact on the agent's policy.

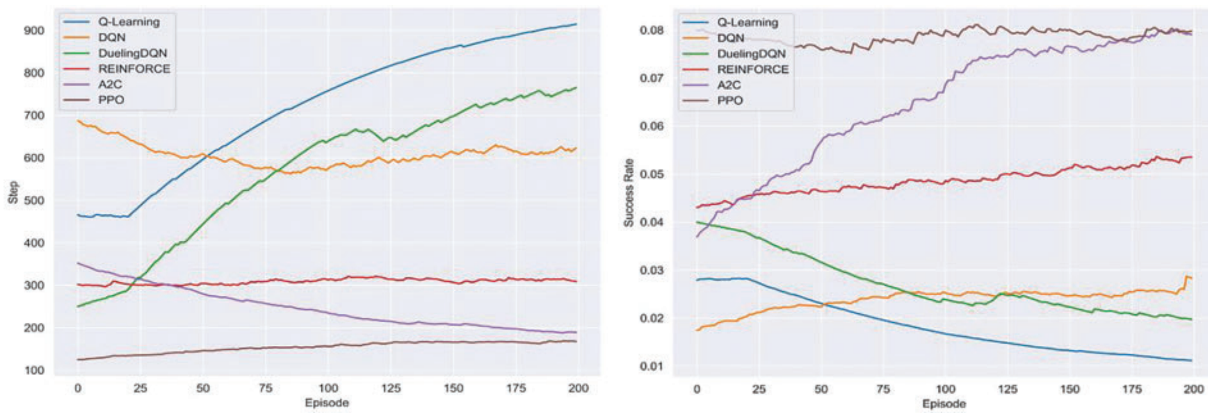

Figure 8: Moving average of steps and success rate in ToyCTF Alpha

5.2 Case of Scenario with CVSS Reward

In this section, we analyze the impact of the proposed CVSS reward methods on the learning efficiency of the RLs. Fig. 9 presents the cumulative reward for 'ToyCTF Alpha' based on the CVSS reward. While on-policy RLs demonstrated quick convergence in the initial stages, they ultimately achieved lower rewards compared to the off-policy-based DQN. This can be possibly attributed to on-policy algorithms relying on the current policy, leading to being trapped in local optima. However, the performance is expected to gradually improve due to the continuous upward trend. In addition, DQN and Q-Learning can also improve performance compared to the prior results without CVSS rewards. Q-Learning also shows a more stable learning curve although it has a lower reward value. In particular, DQN demonstrated a pattern of converging to the highest reward value.

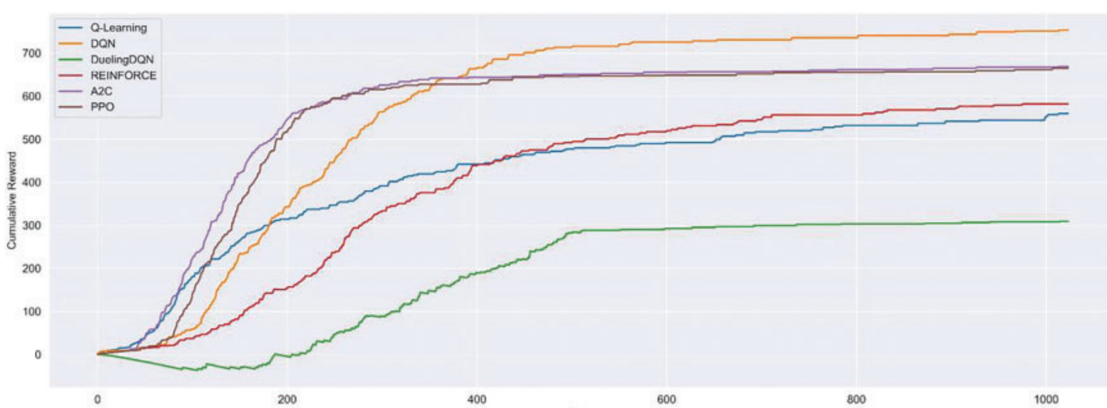

Figure 9: Moving average of cumulative reward in ToyCTF Alpha with CVSS reward

Fig. 10 presents a graph of the step and success rate for 'ToyCTF Alpha' with CVSS reward. In this result, the off-policy RLs showed a high number of steps and gradually converged to a low value, but A2C and PPO immediately updated the policy using the performed trajectory and converged to a lower number of steps. However, REINFORCE showed an unstable graph to be compared with the previous experiment. This instability is likely due to the increased variance in rewards after the adoption of penalty rewards. Overall, the success rate of the RLs improved, which indicates that the adoption of CVSS rewards contributed to the generation of efficient attack strategies by the agents. To be compared with the prior experiments, DQN generally returned the highest cumulative reward even though its success rate was lower than the other RLs. This is attributed to a significant portion of penalty rewards returning zero, which means that the scaling of penalty rewards could be an important factor in improving learning performance. In our experiments, we observed that the on-policy RLS performed well in both scenarios, with A2C exhibiting the highest reward and success rate.

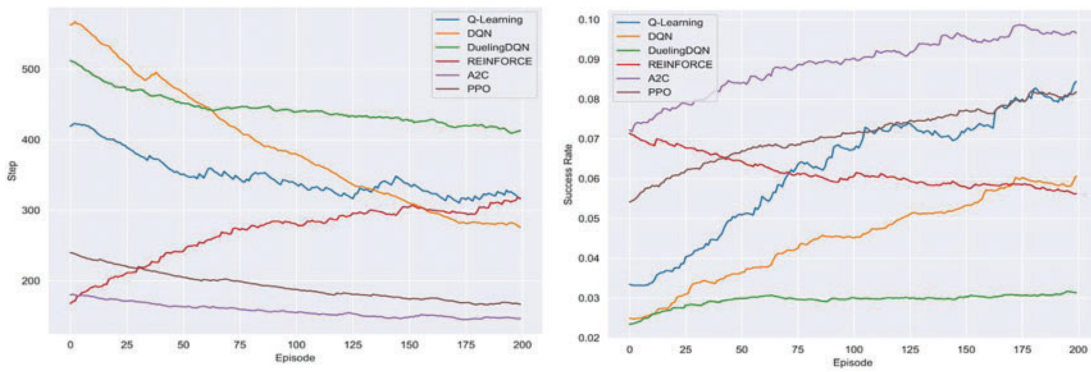

Figure 10: Moving average of steps and success rate in ToyCTF Alpha with CVSS reward

The success rate is directly linked to the accuracy of the attack techniques generated by the RLs. Fig. 11 is a graph comparing the success rate of the A2C, which outperformed in all scenarios. It offers an intuitive understanding of the performance improvement following the proposed methods such as the 'ToyCTF Alpha' and CVSS rewards. The success rate is higher in the 'ToyCTF Alpha' compared to the ToyCTF, with consistent increases. With CVSS rewards in the 'ToyCTF Alpha', significant improvements in learning performance were observed. When comparing the prior value of the moving average graph, the success rate percentage increased by 16.77%. The overall learning performance of the RLs improved in the scenario with CVSS rewards, demonstrating that the CVSS reward method effectively enhanced the training performance of the RLs in the cybersecurity simulation environment.

Figure 11: Moving average of the success rate of the A2C in both ToyCTF and ToyCTF Alpha

Table 8 presents the average success rate for 40 episodes extracted from each scenario. When analyzing the average values from episodes 161 to 200 for the ToyCTF and ToyCTF Alpha. In the 'ToyCTF Alpha' scenario, the success rate of Q-Learning and DQN decreased by 0.23% and 1.95 percentage points, respectively, representing a decrease of 23.00% and 36.38% compared to the ToyCTF. In contrast, DDQN, REINFORCE, and A2C increased by 0.68%, 1.67%, and 3.52 percentage points, respectively, representing to improvements of 64.76%, 40.83%, and 68.64% over the 'ToyCTF'. In particular, PPO, which did not learn in the 'ToyCTF' increased by 7.17 percentage points, with a remarkable growth rate of 853.57% compared to the ToyCTF. These results indicate improved training stability and performance in the 'ToyCTF Alpha'. When comparing the 'ToyCTF Alpha' with the addition of CVSS rewards to the basic 'ToyCTF Alpha', we examined the average success rate from episodes 161 to 200. Overall, most algorithms exhibited an increase in success rates. Compared to 'ToyCTF Alpha', Q-Learning improved by 9.91 percentage points, DQN by 3.31 percentage points, and DDQN by 1.63 percentage points. The results show increases in success rate of 1287.01%, 97.07%, and 94.21%, respectively, over the previous scenario. The off-policy RLs, which showed low success rates in 'ToyCTF Alpha', demonstrated improvement with CVSS rewards. The REINFORCE algorithm decreased by 0.55 percentage points to 9.55%, which is relatively high compared to the 'ToyCTF' scenario. The success rates of A2C and PPO increased by approximately 1.52%p and 1.28 percentage points, respectively, resulting in improvements of 15.98% and 17.53% over the previous scenario. Finally, the scenario that incorporated all proposed methods based on A2C increased from 5.13% to 10.17%, an increase of about 98.24% compared to 'ToyCTF'.

Table 8: Average success rate per 40 episodes in both ToyCTF and ToyCTF Alpha

Scenario	Algorithms	Episodes				
		40	80	120	160	200
ToyCTF without CVSS reward	Q-Learning	1.26	1.04	1.06	1.05	1.00
	DQN	3.72	5.36	5.86	5.37	5.36
	DDQN	1.19	1.00	0.99	1.18	1.05
	REINFORCE	3.79	3.62	4.12	4.57	4.09
	A2C	4.97	5.27	5.06	5.01	5.13
	PPO	1.42	0.77	0.77	0.80	0.84
ToyCTF Alpha without CVSS reward	Q-Learning	1.98	0.73	0.74	0.80	0.77
	DQN	3.25	2.92	2.48	2.54	3.41
	DDQN	2.15	1.18	1.74	1.70	1.73
	REINFORCE	5.26	4.89	5.27	5.70	5.76
	A2C	7.83	8.71	9.80	7.90	8.65
	PPO	6.94	7.87	8.38	8.02	8.01
ToyCTF Alpha with CVSS reward	Q-Learning	6.57	10.34	9.19	7.17	10.68
	DQN	5.28	6.12	6.12	7.12	6.72
	DDQN	4.06	3.16	3.09	3.03	3.36
	REINFORCE	5.54	5.02	5.80	5.50	5.21
	A2C	10.18	10.38	9.65	10.12	10.17
	PPO	8.64	7.88	8.36	8.14	9.29

In conclusion, the performance of DQN and Q-Learning has significantly improved in an environment that reflects CVSS Rewards. Off-policy methods such as DQN and Q-Learning learn optimal policies by repeatedly training on various state-action pairs stored in a replay buffer, suggesting that the improved reward signals have contributed to their learning process. This explicitly demonstrates that the improved reward system via CVSS has captured the complexity of the environment, thus enhancing learning stability. The proposed method, which is based on a scenario that defines the information and importance of nodes can be used not only in the 'ToyCTF Alpha' scenario but also can be utilized in other reinforcement learning-based cyber training environments and cyber ranges. This provides significant results for the future selection and development of RL models and algorithms in cybersecurity.

6 Conclusion

In this paper, we proposed a new approach to address the rapidly changing cyber threat environment by leveraging CyberBattleSim, which is an RL-based cyber-attack-defense simulation tool. We aimed to enhance security by developing our proposed scenario, ToyCTF Alpha', an extension of the original 'ToyCTF' based on the CyberBattleSim. Furthermore, by applying the CVSS-based vulnerability assessment method and intricately designing the attacker's penetration process, we have bridged the gap with reality and enhanced realism. These enhanced assessment methods are instrumental in assisting organizations to more effectively prioritize actual risk assessment and vulnerability management. We validated the effectiveness of the developed system by conducting experiments with RL-based off-policy algorithms such as Q-Learning, DQN, and DDQN and RL-based on-policy algorithms such as REINFORCE, PPO, and A2C, and we also compared and analyzed the results within ToyCTF and ToyCTF Alpha environmental scenarios. The experimental results demonstrated an overall improvement in the learning stability of the proposed methods. In particular, there was a significant increase in both the reward and success rate for Q-Learning and PPO that exhibited low learning effectiveness in 'ToyCTF'. In each scenario, the A2C outperformed the other RLs. Furthermore, the CVSS-based reward method improves the success rate of A2C on ToyCTF Alpha improved to 10.17%, representing a 98.24% increase compared to 'ToyCTF'. These findings suggest that focusing on specific vulnerable nodes has contributed to the performance enhancement in the cyber-attack scenario. The results of the performance improvement and algorithm analysis also show that the proposed methods in this research have a positive impact on enhancing learning efficiency and strategic planning capabilities of RL agents responding to cybersecurity challenges. Additionally, the vulnerability assessment method and CVSS-based reward can contribute to the design of reward systems for cyber-attack-defense simulation environments based on RL in the future.

In future work, we aim to improve the learning process of RL algorithms within CyberBattleSim and other RL-based cyber training environments. Specifically, our objective is to refine the environment to facilitate the derivation of more effective attack strategies by the agents. Furthermore, we plan to focus on developing multi-agent-based cyber-attack and defense scenarios aiming to develop an intelligent cyber-attack defense system capable of handling complex and dynamic real-world cyber-attack-defense scenarios.

Acknowledgement: The authors wish to express their appreciation to the reviewers for their helpful suggestions which greatly improved the presentation of this paper.

Funding Statement: This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea Government (MSIT) (No. RS-2022-II220961).

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Bum-Sok Kim, Min-Suk Kim; data collection: Hye-Won Suk, Yong-Hoon Choi; analysis and interpretation of results: Bum-Sok Kim, Hye-Won Suk; draft manuscript preparation: Bum-Sok Kim. Hye-Won Suk; writing: Bum-Sok Kim, Hye-Won Suk, Yong-Hoon Choi; writing/review: Dae-Sung Moon, Min-Suk Kim; editing supervision: Min-Suk Kim; project administration: Dae-Sung Moon. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: All data generated or analyzed during this study are included in this published article.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Bharadiya J. Machine learning in cybersecurity: techniques and challenges. Eur J Technol. 2023;7(2):1–14. doi:10.47672/ejt.1486.
2. Zheng Y, Li Z, Xu X, Zhao Q. Dynamic defenses in cyber security: techniques, methods and challenges. Dig Commun Netw. 2022;8(4):422–35. doi:10.1016/j.dcan.2021.07.006.
3. De Azambuja AJG, Plesker C, Schützer K, Anderl R, Schleich B, Almeida VR. Artificial intelligence-based cyber security in the context of Industry 4.0—a survey. Electronics. 2023;12(8):1920. doi:10.3390/electronics12081920.
4. Alhayani B, Mohammed HJ, Chaloob IZ, Ahmed JS. Effectiveness of artificial intelligence techniques against cyber security risks in the IT industry. Mater Today: Proc. 2021;65(10):531. doi:10.1016/j.matpr.2021.02.531.
5. Bandari V. Enterprise data security measures: a comparative review of effectiveness and risks across different industries and organization types. Int J Bus Intell Big Data Anal. 2023;6(1):1–11. doi:10.17613/9fer-nw82.
6. Bhadouria AS. Study of: impact of malicious attacks and data breach on the growth and performance of the company and few of the world's biggest data breaches. Int J Sci Res Publ. 2022. doi:10.29322/IJSRP.X.2022.p091095.
7. Sarker IH. CyberLearning: effectiveness analysis of machine learning security modeling to detect cyber-anomalies and multi-attacks. Internet of Things. 2021;14(5):100393. doi:10.1016/j.iot.2021.100393.
8. Kilincer IF, Ertam F, Sengur A. Machine learning methods for cyber security intrusion detection: datasets and comparative study. Comput Netw. 2021;188:107840. doi:10.1016/j.comnet.2021.107840.
9. Zhang J, Pan L, Han QL, Chen C, Wen S, Xiang Y. Deep learning based attack detection for cyber-physical system cybersecurity: a survey. IEEE/CAA J Autom Sinica. 2022;9(3):377–91. doi:10.1109/JAS.2021.1004261.
10. Inayat U, Zia MF, Mahmood S, Khalid HM, Benbouzid M. Learning-based methods for cyber attacks detection in IoT systems: a survey on methods, analysis, and future prospects. Electronics. 2022;11(9):1502. doi:10.3390/electronics11091502.
11. Microsoft. CyberBattleSim. Available from: https://github.com/microsoft/CyberBattleSim. [Accessed 2021].

12. Walter E, Ferguson-Walter K, Ridley A. Incorporating deception into CyberBattleSim for autonomous defense. In: IJCAI 1st International Workshop on Adaptive Cyber Defense (IJCAI-ACD), 2021; Montreal, QC, Canada.
13. Khetarpal K, Riemer M, Rish I, Precup D. Towards continual reinforcement learning: a review and perspectives. J Artif Intell Res. 2022;75:1401–76. doi:10.1613/jair.1.13673.
14. Nguyen TT, Reddi VJ. Deep reinforcement learning for cyber security. IEEE Trans Neural Netw Learn Syst. 2021;34(8):3779–95. doi:10.1109/TNNLS.2021.3121870.
15. Pandey AB, Tripathi A, Vashist PC. A survey of cyber security trends, emerging technologies and threats. Cyber Secur Intell Comput Commun. 2022;1007:19–33. doi:10.1007/978-981-16-8012-0_2.
16. Safitra MF, Lubis M, Fakhrurroja H. Counterattacking cyber threats: a framework for the future of cybersecurity. Sustainability. 2023;15(18):13369. doi:10.3390/su151813369.
17. Aslan Ö., Aktuğ SS, Ozkan-Okay M, Yilmaz AA, Akin E. A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions. Electronics. 2023;12(6):1333. doi:10.3390/electronics12061333.
18. Armenia S, Angelini M, Nonino F, Palombi G, Schlitzer MF. A dynamic simulation approach to support the evaluation of cyber risks and security investments in SMEs. Decis Support Syst. 2021;147(2):113580. doi:10.1016/j.dss.2021.113580.
19. Chowdhury N, Katsikas SK, Gkioulos V. Modeling effective cybersecurity training frameworks: a delphi method-based study. Comput Secur. 2022;113(3):102551. doi:10.1016/j.cose.2021.102551.
20. Kweon E, Lee H, Chai S, Yoo K. The utility of information security training and education on cybersecurity incidents: an empirical evidence. Inf Syst Front. 2021;23:361–73. doi:10.1007/s10796-019-09977.
21. Watkins C, Dayan P. Technical note: Q-learning. Mach Learn. 1992;8:279–92.
22. Mnih V, Kavukcuoglu K, Silver D, Graves A, Antonoglou I, Wierstra D, et al. Playing Atari with deep reinforcement learning. In: NIPS Deep Learning Workshop, 2013 Dec 5–10; NV, USA.
23. Wang Z, Schaul T, Hessel M, Hasselt HV, Lanctot M, Freitas ND. Dueling network architectures for deep reinforcement learning. In: Proceedings of the 33rd International Conference on Machine Learning, 2016 Jun 20–22; New York, NY, USA.
24. Mnih V, Badia AP, Mirza M, Graves A, Harley T, Lillicrap TP, et al. Asynchronous methods for deep reinforcement learning. In: Proceedings of the 33rd International Conference on International Conference on Machine Learning, 2016 Jun 24–26; New York, NY, USA.
25. Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal policy optimization algorithms. Comput Res Rep. 2017. doi:10.48550/arXiv.1707.06347.
26. Sutton RS, McAllester D, Singh S, Mansour Y. Policy gradient methods for reinforcement learning with function approximation. In: Proceedings of the Advances in Neural Information Processing Systems 12, 1999 Nov 29–Dec 26; Denver, CO, USA.
27. Djenna A, Harous S, Saidouni DE. Internet of Things meet internet of threats: new concern cyber security issues of critical cyber infrastructure. Appl Sci. 2021;11(10):4580. doi:10.3390/app11104580.
28. Nassar A, Kamal M. Machine learning and big data analytics for cybersecurity threat detection: a holistic review of techniques and case studies. J Artif Intell Mach Learn Manag. 2021;5(1):51–63.
29. Bouchama F, Kamal M. Enhancing cyber threat detection through machine learning-based behavioral modeling of network traffic patterns. Int J Bus Intell Big Data Anal. 2021;4(9):1–9.
30. Radivilova T, Kirichenko L, Alghawli AS, Ageyev D, Mulesa O, Baranovskyi O, et al. Statistical and signature analysis methods of intrusion detection. In: Information Security Technologies in the Decentralized Distributed Networks, Cham: Springer International Publishing; 2022; p. 115–31. doi:10.1007/978-3-030-95161-0_5.
31. Saputra IP, Utami E, Muhammad AH. Comparison of anomaly based and signature based methods in detection of scanning vulnerability. In: 2022 9th International Conference on Electrical Engineering,

Computer Science and Informatics (EECSI), 2022 Oct 6–7; Jakarta, Indonesia: IEEE Xplore; p. 221–5. doi:10.23919/EECSI56542.2022.9946485.

32. Li Q, Huang H, Li R, Lv J, Yuan Z, Ma L, et al. A comprehensive survey on DDoS defense systems: new trends and challenges. Comput Netw. 2023;233(4):109895. doi:10.1016/j.comnet.2023.109895.
33. Ferdous J, Islam R, Mahboubi A, Islam MZ. A review of state-of-the-art malware attack trends and defense mechanisms. IEEE Access. 2023;11:121118–41. doi:10.1109/ACCESS.2023.3328351.
34. Tahirkheli AI, Shiraz M, Hayat B, Idrees M, Sajid A, Ullah R, et al. A survey on modern cloud computing security over smart city networks: threats, vulnerabilities, consequences, countermeasures, and challenges. Electronics. 2021;10(15):1811. doi:10.3390/electronics10151811.
35. Khalil MI, Abdel-Rahman M. Advanced cybersecurity measures in IT service operations and their crucial role in safeguarding enterprise data in a connected world. Eigenpub Rev Sci Technol. 2023;7(1):138–58.
36. Kumar S, Gupta U, Singh AK, Singh AK. Artificial intelligence: revolutionizing cyber security in the digital era. J Comput, Mech Manage. 2023;2(3):31–42. doi:10.57159/gadl.jcmm.2.3.23064.
37. Apruzzese G, Laskov P, Montes de Oca E, Mallouli W, Rapa LB, Grammatopoulos AV, et al. The role of machine learning in cybersecurity. Digital Threats: Res Pract. 2022;4(1):1–38. doi:10.48550/arXiv.2206.09707.
38. Abdallah EE, Eleisah W, Otoom AF. Intrusion detection systems using supervised machine learning techniques: a survey. Procedia Comput Sci. 2022;201(1):205–12. doi:10.1016/j.procs.2022.03.029.
39. Aljabri M, Aljameel SS, Mohammad RMA, Almotiri SH, Mirza S, Anis FM, et al. Intelligent techniques for detecting network attacks: review and research directions. Sensors. 2021;21(21):7070. doi:10.3390/s21217070.
40. Adawadkar AMK, Kulkarni N. Cyber-security and reinforcement learning—a brief survey. Eng Appl Artif Intell. 2022;114:105116. doi:10.1016/j.engappai.2022.105116.
41. Sewak M, Sahay SK, Rathore H. Deep reinforcement learning for cybersecurity threat detection and protection: A review. In: International Conference On Secure Knowledge Management In Artificial Intelligence Era (SKM), 2021 Oct 8–9; San Antonio, TX, USA; p. 51–72. doi:10.1007/978-3-030-97532-6_4.
42. Huang Y, Huang L, Zhu Q. Reinforcement learning for feedback-enabled cyber resilience. Annu Rev Control. 2022;53(2):273–95. doi:10.1016/j.arcontrol.2022.01.001.
43. Ghanem MC, Chen TM. Reinforcement learning for efficient network penetration testing. Information. 2020;11(1):6. doi:10.3390/info11010006.
44. van Geest RJ, Cascavilla G, Hulstijn J, Zannone N. The applicability of a hybrid framework for automated phishing detection. Comput Secur. 2024;139(9):103736. doi:10.1016/j.cose.2024.103736.
45. Kavak H, Padilla JJ, Vernon-Bido D, Diallo SY, Gore R, Shetty S. Simulation for cybersecurity: state of the art and future directions. J Cybersecur. 2021;7(1):tyab005. doi:10.1093/cybsec/tyab005.
46. Sheikh ZA, Singh Y, Singh PK, Ghafoor KZ. Intelligent and secure framework for critical infrastructure (CPS): current trends, challenges, and future scope. Comput Commun. 2022;193:302–31.
47. Jjschwartz. NetworkAttackSimulator (version 0.9.1). Available from: https://github.com/Jjschwartz/NetworkAttackSimulator. [Accessed 2019].
48. Franco J, Aris A, Canberk B, Uluagac AS. A survey of honeypots and honeynets for internet of things, industrial internet of things, and cyber-physical systems. IEEE Commun Surv Tutorials. 2021 Aug;23(4):2351–83. doi:10.1109/COMST.2021.3106669.
49. Reti S, Elzer K, Fraunholz D, Schneider D, Schotten HD. Evaluating deception and moving target defense with network attack simulation. In: Proceedings of the 9th ACM Workshop on Moving Target Defense, 2022; Los Angeles, CA, USA: ACM Digital Library; p. 43–53. doi:10.1145/3560828.3564006.
50. Oh S, Jeong M, Kim H, Park J. Applying reinforcement learning for enhanced cybersecurity against adversarial simulation. Sensors. 2023;23(6):3000. doi:10.3390/s23063000.

51. Oh S, Kim J, Nah J, Park J. Employing deep reinforcement learning to cyber-attack simulation for enhancing cybersecurity. Electronics. 2024;13(3):555. doi:10.3390/electronics13030555.
52. AL-Hawamleh A. Cyber resilience framework: strengthening defenses and enhancing continuity in business security. Int J Comput Dig Syst. 2024;15(1):1315–31.
53. Anand P, Singh Y, Selwal A, Singh PK, Ghafoor KZ. IVQFIoT: an intelligent vulnerability quantification framework for scoring internet of things vulnerabilities. Expert Syst. 2022;39(5):e12829. doi:10.1111/exsy.12829.
54. Gangupantulu R, Cody T, Park P, Rahman A, Eisenbeiser L, Radke D, et al. Using cyber terrain in reinforcement learning for penetration testing. In: Proceedings of the 2022 IEEE International Conference on Omni-Layer Intelligent Systems (COINS), 2022 Aug 1–3; Barcelona, Spain; p. 1–8. doi:10.1109/COINS54846.2022.9855011
55. Chatterjee S, Thekdi S. An iterative learning and inference approach to managing dynamic cyber vulnerabilities of complex systems. Reliabil Eng Syst Saf. 2020;193(2):106664. doi:10.1016/j.ress.2019.106664.
56. Kekül H, Ergen B, Arslan H. Estimating vulnerability metrics with word embedding and multiclass classification methods. Int J Inf Secur. 2024;23(1):247–70. doi:10.1007/s10207-023-00734-7.
57. Kim B, Suk H, Kim J, Jeong J, Kim M. Cyber attack and defense strategies using reinforcement learning via simulated network environment. In: The 7th International Conference on Mobile Internet Security (MobiSec), 2023 Dec 19–21; Okinawa, Japan.