Reinforcement learning based self-adaptive moving target defense against DDoS attacks

Chungang Gao\textsuperscript{1,2} and Yongjie Wang\textsuperscript{1,2}

\textsuperscript{1}College of Electronic Engineering, National University of Defense Technology, Hefei, Anhui 230037, China
\textsuperscript{2}Anhui Province Key Laboratory of Cyberspace Security Situation Awareness and Evaluation, Hefei, Anhui 230037, China

E-mail: wangyongjie17@nudt.edu.cn

Abstract. In the research of moving target defense technology, the formulation and evaluation of network offensive and defensive game strategy have become a current research hotspot, but there are still some urgent problems to be solved, such as how to adjust the defense if the attacker’s strategy is constantly changing during the offensive and defensive game strategies to deal with its changes and how to balance system security and system performance when system resources are limited. Aiming at DDoS attacks, this paper proposes a moving target defense adaptive strategy based on reinforcement learning, which can adaptively adjust defense strategies according to changes in environmental conditions, while balancing system security and system performance through parameter adjustments to adapt to different scenarios. The simulation experiment results show that the proposed model and method are feasible and effective.

1. Introduction
Distributed denial of service (DDoS) attacks are an increasingly serious problem facing network systems. Attackers use botnets to send a large number of illegal requests to the target application's server, making its resources unbearable and eventually causing the system to crash. Traditional network defense technologies (such as intrusion detection, firewall) are static, and the attacker has the time advantage to organize the attack, that is, the attacker can perform repeated vulnerability analysis and penetration testing on the inherent vulnerability of the target for a long time, until the final Goal 0. Frequent network security incidents have proved that traditional passive defense technologies cannot cope with unknown network security threats, and cannot take measures to effectively defend against attacks before the network and information systems are attacked or before serious losses occur.

As a new type of active defense technology that can "change the rules of the game", Moving Target Defense (MTD) 0 is also a response to DDoS attacks. Focus direction. The idea is to make the system dynamic, by increasing the diversity, dynamics, and randomness of the system 0, increasing the difficulty and cost of the attacker, and improving the security of the system 0. In the existing research on MTD strategy, the use of game theory 0 to build an MTD offensive and defensive model, analyze the offensive and defensive process in different scenarios and find the corresponding optimal defense strategy has become a current research focus. In the process of offensive and defensive games, the
attacker's strategy is not static. How to adaptively adjust the defense strategy to deal with the change of the attacker's strategy has become the focus of research. At the same time, MTD will inevitably increase the defense cost of the system, because the dynamic changes of the system will reduce the performance of the system to a certain extent, which will harm the system. Therefore, in the MTD network offensive and defensive confrontation, how the defender reasonably selects strategies to balance system security and system performance is a new challenge for mobile target defense technology.

The outstanding performance of machine learning in large-scale data processing, classification, and intelligent decision-making has attracted widespread attention in various fields. The use of machine learning can also solve many problems in the field of network security, such as malware detection, intrusion detection, and other fields that have achieved good results. Reinforcement learning is a branch of machine learning, emphasizing how to act based on the environment, through continuous interaction with the environment, dynamic decision management according to changes in the environment.

This paper proposes a mobile target defense adaptive strategy based on reinforcement learning. The defender can adaptively optimize the defense strategy according to the change of the attack strategy, and at the same time can adjust the strategy during the defense process to balance the system security and system performance.

2. Related work
In the conventional defense against DDoS, defenders will try to filter attack packets but not normal traffic, or block the IP address of suspicious users and other methods. However, these methods cannot effectively prevent complex DDoS attacks, because the attacker may use many IP addresses as the source of their information packets, or construct attack information packets to imitate legitimate traffic. Therefore, the traditional defense can only partially mitigate DDoS attacks.

In the defense of moving targets, there is a type of technology that the defender uses random configuration methods to make the attack process difficult. Through continuous reconfiguration, MTD can significantly reduce the attacker's intelligence-gathering ability and delay the attack. In previous studies, researchers have proposed many MTD methods to deal with DDoS attacks. Khattab et al. proposed "active server roaming", which is a defensive system in which only one server processes client requests at a time and periodically rotates the servers in the server pool. Shi et al. proposed "IP jump", that is, the defender periodically changes the IP address of his server in the option pool, following a predefined path. Most of these strategies have the following problems: First, in the offensive and defensive process, the defenders are reconfigured in a fixed cycle, which is easy for the attacker to find the law, rendering the defense ineffective. Second, the simulation research on these strategies assigns a fixed strategy to the attacker and defender to implement, but in reality, the attacker's attack strategy is constantly changing. At the same time, when evaluating the effectiveness of the MTD strategy, the cost of the defender is not considered, and only paying attention to system security and not paying attention to the consumption of system performance by defense actions may cause a waste of resources.

Therefore, the MTD defender needs a more reasonable strategy. If the attacker's strategy can change over time, the defender can adaptively adjust the strategy according to the opponent's expected actions, and the reconfiguration is no longer a fixed cycle, but with the defense strategy At the same time, system security and system performance should be considered comprehensively when evaluating MTD strategy.

3. Models and algorithms
This section will introduce a mobile target defense adaptive algorithm based on reinforcement learning. By running this algorithm, the defender can automatically optimize the defense strategy according to changes in the environment, improve defense performance, and achieve a balance between security and performance.
3.1. Task modeling
The offensive and defensive strategy scenarios studied in this article are aimed at DDoS attacks. Attackers usually need a more complicated process to implement DDoS attacks. First of all, the attacker must capture enough puppet hosts and install the corresponding attack program. The next step is to understand the target. The attacker has a comprehensive understanding of the target through scanning, social engineering, and other means. What the attacker needs to master The content includes the IP address, port, configuration, performance, bandwidth, etc. of the attack target. The last stage is the actual attack process. The attacker can issue an attack command to the puppet host through the master computer, or according to the set time And the target, keep sending a large number of attack packets to the target to achieve the purpose of the denial of service.

For defenders, it is more difficult to detect and take countermeasures against the first stage of a DDoS attack, because it is difficult to find the captured puppet machine in the entire network, and in the actual offensive and defensive confrontation, the puppet machine is The attacker already has the conditions, so this article will establish an offensive and defensive game model for the latter two stages (understanding the attack target and the actual attack). In the general offensive and defensive game model, the attack process of a DDoS attacker is shown in Figure 1. For the implementer of a DDoS attack, the time to understand the target of the attack is much longer than the actual attack time. At the stage when the attacker understands the target of the attack because the attacker does not perform destructive actions on the system, it is difficult for the defender to detect the presence of the attacker, and the attacker can carry out an effective attack relatively easily. Only when the defender passes the attack detection When the method finds that the system is under a DDoS attack, it will immediately take measures to block the attack (such as blocking the attacker’s IP), and then the attacker’s attack will be invalid. Know the target of the attack.

![Figure 1. DDoS attack process in the general offensive and defensive game model.](image)

When the system deploys MTD technology, each time the defender detects a DDoS attack, it will use MTD methods (such as IP change, port change) to block the attack, and because the system environment changes, the attacker needs to re-understand the attack target. The attack process is shown in Figure 2.

![Figure 2. DDoS attack process in the MTD offensive and defensive game model.](image)

By deploying MTD technology, greatly increases the difficulty for a DDoS attacker to successfully attack the target, but it can be found that each time the attacker re-understands the target, it is still possible to successfully attack the target. Our ideal situation is shown in Figure 3 below. The defender can successfully block the attack before the attacker implements an actual attack. That is, the defender should reconfigure the system when the attacker knows the target of the attack. In the actual attack stage, the attacker cannot achieve the attack effect, so that DDoS attacks can be truly avoided.
Figure 3. DDoS attack process in the ideal MTD offensive and defensive game model.

If the defender wants to block the attack every time the attacker knows the target of the attack, the interval $L_{MTD}$ for implementing the MTD technology needs to be less than the duration of the attacker knows the target of the attack. Because the duration of the attacker's understanding of the target will be different each time, $L_{MTD}$ should be the shortest duration. This can intercept every DDoS attack, but because the attacker's attack strategy will change, the attack intensity will be very high in a period, and the attack intensity will be smaller in another period. If $L_{MTD}$ is fixed, it will make a lot of ineffective defenses and waste system performance.

As shown in Figure 4, when the system is deployed with MTD technology, the service process of the system is divided into cycles, the length of the cycle is LMTD, and each cycle contains two processes, namely the normal service phase and the decision execution phase. Based on the above analysis, a mobile target defense adaptive strategy based on reinforcement learning is proposed. The change in attack intensity is inferred based on the state of the system feedback, and the defense strategy is adaptively adjusted, that is, whether to implement defense actions in the decision-making execution stage to reduce unnecessary The production of defensive actions improves the hit rate of defense strategies.

Figure 4. System service process.

3.1.1. Game participant

The set of players in the game is $P=\{1,2\}$, where 1 represents the defender and 2 represents the attacker. The two types of participants are briefly described below:

Attacker: The attack type is a DDoS attack. Attackers continue to implement DDoS attacks on the target system. During the attack, the attacker usually constantly adjusts the attack strategy, such as attack type and attack intensity.

Defender: The defender is responsible for ensuring the safe operation of the system. Define defense actions as IP address change and port change. The goal of the defender is to use limited resources to maximize system security.

3.1.2. State space

The state space of the system is $S = \{s_1, s_2, \ldots, s_n\}$. The defender updates the state of the system during the execution of the decision. In the offensive and defensive game model, we only consider whether the system is attacked. So the system status can only be attacked and not attacked. During the execution of the decision, the defender needs to perform actions based on the system state. If we only consider the current attack situation of the system, it is difficult to make a correct decision. Therefore, we define the system state obtained by the defender in the decision execution stage as the system state of the past five cycles. The attacked situation, ie $s_n = (s_{n1}, s_{n2}, s_{n3}, s_{n4}, s_{n5})$.

3.1.3. Action space

The action space of the defender is $A1 = \{a_{11}, a_{12}\}$, where $a_{11}$ indicates that the defender performs defensive actions, and $a_{12}$ indicates that the defender does not perform defense actions.

The action space of the attacker is $A2 = \{a_{21}, a_{22}\}$, where $a_{21}$ means that the attacker performs an attack action, and $a_{22}$ means that the attacker does not perform an attack action.
3.1.4. Decision return function

To measure the rewards of different actions taken by defenders in different states, several decision variables are defined. Define the benefit $P$ as the defense benefit obtained by the defender successfully intercepting the attack. Define consumption $C$ as the consumption of system performance by defenders performing defense actions. The loss $L$ is defined as the loss of the system caused by the defender not successfully intercepting the attack.

The decision reward function $R$ can be expressed as:

$$ R = P - C - L $$

To obtain greater returns, the defensive actions taken should have greater benefits, less consumption, and less loss.

3.2. Revenue quantification

In the process of offensive and defensive games, the more random, dynamic, and uncertain the defense strategy selection is, the higher the security of the system; the more defense actions are executed, the greater the impact on system performance. From the perspective of system security, the defense process ensures the randomness, dynamics, and uncertainty of the system; from the perspective of system performance, the defense action interrupts the normal service process of the system and reduces the performance of the system. We quantify the benefits of defense strategies.

Definition 1: System security refers to the proportion of the system’s safe operation time in the entire game time. System security is the main goal pursued by defenders.

Definition 2: System performance refers to the ability of the system to provide normal services. Because the defense process will interrupt the normal service process of the system and reduce the availability of the system, the more defense actions are performed, the lower the system performance.

Definition 3: Defense efficiency refers to the ratio of effective defense to all defense actions. In the offensive and defensive process, if the attacker did not implement the attack, but the defender took defensive actions, then these defensive actions are not necessary. Therefore, the definition of defense efficiency checks the accuracy of the defense actions.

3.3. Algorithm framework

Figure 5. Structure model of the defense system.

Figure 5 shows the overall structure model of the defense system proposed in this article. The environment of the defense system includes a web server, an attack detection system, and a decision execution system. The defensive agent uses reinforcement learning algorithms as the framework to interact with the environment through three variables: state, action, and reward.

Use the Q-learning algorithm to continuously iteratively optimize the future action strategy through the income feedback function $Q$ Table. The iterative update formula is as follows:

$$ Q(s_t, a_t) = Q(s_t, a_t) + \alpha [R + \gamma \max Q(s_{t+1}, a) - Q(s_t, a_t)] $$
Among them, R is the reward function, \( \alpha \) is the learning rate, and \( \gamma \) is the discount factor. Reinforcement learning introduces a discount factor \( \gamma \) to adjust the importance of future returns. When it is close to 0, the agent is more concerned about short-term returns; when it is closer to 1, long-term returns become more important. At the same time, the learning rate is introduced to control the update range of the income feedback function Q. The larger the learning rate \( \alpha \), the larger the proportion of results obtained by new attempts, and the smaller the proportion of maintaining the old results.

The pseudo-code of the mobile target defense adaptive algorithm based on reinforcement learning is as follows:

\begin{algorithm}
1. for k = 1 → K do
2. Initialization status information
3. if \( \varepsilon \neq 0 \) then
4. \( \varepsilon = \varepsilon - \theta \)
5. end if
6. for t = 1 → T do
7. use \( \varepsilon \)-greedy method to select actions according to Q Table and state s
8. get attack information
9. get a reward
10. update system status s
11. Update Q Table
12. end for
13. end for
\end{algorithm}

In the above algorithm, we divide the entire offensive and defensive game process into K rounds, and each round is divided into T steps. At the beginning of each round, the state information is initialized and the exploration factor is decreased. In each step, use the \( \varepsilon \)-greedy method to select defense actions according to Q Table and state s, and update system state s according to state transition matrix W according to the attack information and defense action and obtain rewards, and finally update the revenue feedback function Q Table according to the iterative update formula.

### 4. Evaluation

#### 4.1. Experimental environment

This section analyzes and evaluates the adaptive mobile target defense algorithm based on reinforcement learning introduced in the previous section through simulation experiments. Arrange the offensive and defensive environment, including attacker parameters and defender parameters. According to the analysis in the previous section, we know that an attack cycle includes adjustments, understanding of attack targets, and actual attacks. The duration of an attack cycle, that is, the time it takes for an attacker to prepare for an attack to implement an attack. The shorter the attack cycle, the greater the attack intensity. Reflect the constant change of attack intensity by setting attack cycles of different durations. We have set four attack intensities, and the duration of the attack cycle of each attack intensity is shown in Table 1 below.
Table 1. Different levels of attack intensity.

| Attack intensity | Minimum duration (s) | Maximum duration (s) |
|------------------|----------------------|----------------------|
| Level 1          | 1000                 | 2000                 |
| Level 2          | 2000                 | 3000                 |
| Level 3          | 3000                 | 4000                 |
| Level 4          | 4000                 | 5000                 |

In the actual DDoS attack process, the attacker’s preparation time is much longer than the actual attack time, so we ignore the actual attack time. Set the defense period of the defender to 1000s to ensure that the interval for implementing the MTD technology is less than the duration of the attacker's understanding of the attack target.

4.2. Algorithm parameter adjustment

At the same time, adjust the value of income, consumption, and loss. Their value range is 1-4, a total of 64 situations, representing 64 types of defenders. Different defenders have different biases. Some are biased towards greater gains, some are biased towards smaller consumption, and some are biased towards smaller losses. Quantify the benefits of all defenders. Some defenders are not suitable for actual scenarios. For example, the system security is too low or the system performance is too low. Some defenders that are not in line with the actual situation are screened out, and three defenders are found among the remaining defenders. The most representative types of defenders are security defenders, performance defenders, and balanced defenders. The parameters of these three types of defenders are shown in Table 2 below.

Table 2. Three types of defenders.

| category   | profit | cost | lost |
|------------|--------|------|------|
| security   | 4      | 2    | 2    |
| performance| 2      | 3    | 2    |
| balance    | 4      | 3    | 2    |

The simulation experiment was repeated 20 times, and the income quantification of the three types of defenders was shown in Figure 6. Security defenders have high system security but low system performance, performance defenders have high system performance but low system security, and balanced defenders ensure system performance under the premise of ensuring higher system security. These three different defenders are suitable for different scenarios. The security defender is suitable for systems with high-security requirements and good enough performance; the performance defender is suitable for systems with low performance and only need to achieve basic security; balance The type defender is suitable for systems that have certain requirements for security and have limited performance. In actual scenarios, the parameters of the defender are adjusted to meet the system requirements.
Figure 6. Quantification of the benefits of different types of defenders.

4.3. Algorithm convergence verification
Set the strength of the attacker to be strong, the type of defender to be safe, initialize the environment state and defense strategy, verify the convergence of the mobile target defense adaptive algorithm, that is, whether the defender can adaptively optimize the defense strategy and finally learn the optimal strategy. There are 20 rounds of experiment, 400 steps per round. The average safe time is the average of the time the system is in a safe state for each round (if the system is successfully attacked, the safe state will be broken), and the maximum safe time is 400 steps.

Figure 7. Convergence of adaptive strategy.

It can be seen from Figure 7 that as the training continues, the average security duration of the system gradually increases, and finally converges to 400 cycles. The defense strategy is optimal, and the attack is completely blocked.

4.4. Algorithm performance comparison
The experiment compares the adaptive mobile target defense strategy based on reinforcement learning and the random strategy. Random strategy means that the defender randomly implements defensive actions during the offensive and defensive game. On the premise of ensuring the same system performance, compare the system security and defense efficiency of the two strategies.
It can be seen from Figure 8 that whether in terms of system security or defensive efficiency, the mobile target defense adaptive strategy based on reinforcement learning is much higher than the random strategy.

Comprehensive analysis of the above experiments, first of all, we can obtain different types of defenders by adjusting the defender parameters, balance system security and system performance, and meet the needs of different scenarios; through the analysis of the algorithm convergence and effectiveness, it is proved that the algorithm can be self-contained Adaptive optimized defense strategy, and achieve the best within a limited number of steps; through the comparison with the random strategy, the effectiveness of the algorithm is proved.

5. Conclusion
This paper proposes an adaptive strategy for mobile target defense based on reinforcement learning for DDoS attacks. This strategy uses reinforcement learning to adaptively adjust defense strategies based on changes in the attacker’s strategy, successfully intercepting them before the actual attack is implemented and passing Adjust defender parameters to balance system security and system performance to meet the needs of different scenarios. Simulation experiments and comparison of random strategies prove the effectiveness of the algorithm. At the same time, three types of defenders are obtained by adjusting the algorithm parameters. The current stage of the research in this article is mainly aimed at DDoS attacks. The next step is to apply it to the defense against more complex attacks such as APT attacks.

References
[1] Bowers, K. D. , Dijk, M. V. , Griffin, R. , Juels, A. , Oprea, A. , & Rivest, R. L. , et al. (2012). Defending against the Unknown Enemy: Applying FlipIt to System Security. International Conference on Decision and Game Theory for Security. Springer, Berlin, Heidelberg.
[2] NITRD.Cybersecurity game-change research & development recommendations[EB/OL].(2014-03-21). http://www.nitrd.gov/pubs/CSIA_IWG_%20Cybersecurity_%20GameChange_RD_%20Recommendations_20100513.pdf
[3] Prevelakis, V. , Keromytis, A. D. , & Kc, G. S. . (2003). Countering code-injection attacks with instruction-set randomization. Proceedings of the 10th ACM Conference on Computer and Communications Security, CCS 2003, Washington, DC, USA, October 27-30, 2003. ACM.
[4] Jun Xu, Pinyao Guo, Mingyi Zhao, Robert F. Erbacher, Minghui Zhu, and Peng Liu. 2014. Comparing Different Moving Target Defense Techniques. In Proceedings of the First ACM Workshop on Moving Target Defense (MTD '14). Association for Computing Machinery, New York, NY, USA, 97–107.

[5] Hamilton, Samuel & Miller, Wendy & Ott, Allen & Saydjari, O.. (2002). The Role of Game Theory in Information Warfare.

[6] R. K. C. Chang, "Defending against flooding-based distributed denial-of-service attacks: a tutorial," in IEEE Communications Magazine, vol. 40, no. 10, pp. 42-51, Oct. 2002, doi: 10.1109/MCOM.2002.1039856.

[7] He, X. , Dai, H. , & Ning, P. . (2015). Improving learning and adaptation in security games by exploiting information asymmetry. Computer Communications. IEEE.

[8] Briefs, & Henry, W. . (1983). Goetz briefs on capitalism and democracy: an introduction. Review of Social Economy, 41(3), 212-227.

[9] Mirkovic, J. , & Reiher, P. . (2004). A taxonomy of ddos attack and ddos defense mechanisms. ACM SIGCOMM Computer Communication Review.

[10] Gupta, B. B. , Joshi, R. C. , & Misra, M. . (2012). Distributed denial of service prevention techniques. International Journal of Computer and Electrical Engineering.

[11] Jia, Q. , Sun, K. , & Stavrou, A. . (2013). MOTAG: Moving Target Defense against Internet Denial of Service Attacks. Computer Communications and Networks (ICCCN), 2013 22nd International Conference on. IEEE.

[12] Keromytis, A. D. , Misra, V. , & Rubenstein, D. . (2002). Sos: secure overlay services. Proc Acm Sigcomm Pittsburgh Pa Aug, 32(4), 61-72.

[13] Khattab, S. M. , Sangpachatanaruk, C. , Melhem, R. , Mosse, D. L. , & Znati, T. . (2003). Proactive server roaming for mitigating denial-of-service attacks. Information Technology: Research and Education, 2003. Proceedings. ITRE2003. International Conference on. IEEE.

[14] Shi, L. , Jia, C. , Shuwang Lü , & Liu, Z. . (2007). Port and address hopping for active cyber-defense.

[15] Khattab, S. M. , Sangpachatanaruk, C. , Melhem, R. , Mosse, D. L. , & Znati, T. . (2003). Proactive server roaming for mitigating denial-of-service attacks. Information Technology: Research and Education, 2003. Proceedings. ITRE2003. International Conference on. IEEE.

[16] Shi, Elaine & Stoica, Ion & Andersen, David & Perrig, Adrian. (2006). OverDoSe: A Generic DDoS Protection Service Using an Overlay Network.

[17] Kewley, D. , Fink, R. , Lowry, J. , & Dean, M. . (2001). Dynamic approaches to thwart adversary intelligence gathering. DARPA Information Survivability Conference & Exposition II, 2001. DISCEX '01. Proceedings. IEEE.