Automated Penetration Testing Using Deep Reinforcement Learning

1910088 HU Zhenguo

Penetration testing is a new network security technology developed in recent years, which has great practical application for computer network and security. It is a security testing and evaluation method that uses hacker techniques and methods to discover the security vulnerabilities of the target machine. By following this process, one can gain control of the system, access the confidential data, and discover the possible security risks that may affect the continued operation of the business. The ethical hackers, which are called pentesters, will conduct an in-depth security detection of the target network to find out the vulnerable links, and use various realistic attack methods to detect the possible vulnerabilities.

The biggest difference between penetration testing and hacker intrusion is that penetration testing is authorized by the customer. It uses controllable and non-destructive methods to find weaknesses in the target and network equipment, helps managers to know the problems located in their own network, and provides security suggestions to help improve the security of their system. However, currently penetration testing is performed mostly manually and relies mostly on the pentesters' experience, and there are no tools to intelligently analyze the network situation and discover the potential attack paths in a network system.

In order to solve this problem, many researchers try to find some methods that mimick the penetration testing process to discover the attack paths. One of the representative methods is called PDDL, which means Planning Domain Definition Language. It uses planning algorithms, such as graph planning and partial-order planning, to convert attack paths into PDDL expressions. But these methods do not perform well in the discovery of potential paths, and the degree of automation is not high. Another representative method is FF-Replan, which is a dynamic reprogramming algorithm that decomposes the probabilistic programming problem. The algorithm transforms the non-deterministic programming problem into a deterministic programming problem, then uses the FF (Fast Forward) programming algorithm to achieve automated attack path planning.

However, these methods have some shortages, as they need to delete non-deterministic information for planning, and have difficulties in dealing with path finding problems with multiple uncertain attack paths. For this reason, we present another method by which we reinforcement learning techniques into the field of cybersecurity, and use the powerful analysis capabilities of the neural networks to automatically plan the attack paths. Since reinforcement

learning does not require data to have precise labels, it is very suitable for attack path analysis.

In this thesis, we propose an automated penetration testing framework named AutoPentest-DRL, which we designed and implemented to address the shortcomings mentioned so far. The key idea is to employ deep reinforcement learning (DRL) to plan the attack path, and employ other penetration tools to automate the process of penetration testing. To realize this goal, we built a DQN Decision Engine to select the correct attack path according to network and vulnerability information. The input for the decision engine is the matrix representation of the attack tree, and the output is the most feasible attack path. We also employ a topology generator to create enough network topologies in order to increase the model's adaptability. Furthermore, the Depth-First Search (DFS) algorithm is used to simplify the input matrix. In this way, the AutoPentest-DRL can automatically give the corresponding attack path according to the input network information.

In order to make it possible to conduct penetration testing in a real network environment, we adopted Nmap to retrieve the necessary information by scanning the real network environment. The information of vulnerabilities is extracted from the scanning report of Nmap, and is combined with the network topology data in order to send them to the DQN Decision Engine. Penetration tools such as Metasploit have also been integrated in order to to execute automated attack commands.

We first discuss the basic architecture of this framework and the implementation of AutoPentest-DRL. Then, we demonstrate the efficiency of this framework by evaluating it both in logical and real network environments. In logical networks, the average accuracy of the two different topologies we studied is 0.932. In real networks, the framework can employ three different vulnerabilities to get the root privilege of the corresponding serves, and finally, it can copy the test Trojan to the target machine successfully. Our results show that AutoPentest-DRL is suited well for both environments and it is possible to apply it to real systems.

In the future, we plan to do more research on fast attack path discovery techniques in large-scale network scenarios to support penetration testing on complex networks. We also plan to incorporate the generated feedback information during the process of penetration testing into the attack path discovery algorithm to dynamically adjust the attack path. Finally, we will improve the compatibility of our system by integrating other penetration testing tools, such as Nessus and Cobalt Strike. We really hope that our research can promote the development of automated penetration testing, and also inspire other researchers.