

Assignment Title: Research Proposal (***Automated Penetration Testing Using Reinforcement Learning***)

Student Name: Raju Molla

Student ID: 2421791

Module: Research Methods and Professional Development

Module code: ACFA7004

Table of Contents

1. Introduction

1.1 Context and Background

1.2 Research Problem

1.3 Research Aim and Objectives

1.4 Significance of the Research

2. Literature Review

2.1 Traditional and Automated Penetration Testing Approaches

2.2 Machine Learning in Penetration Testing and Security Automation

2.3 Reinforcement Learning in Cybersecurity

2.4 Simulation-Based Environments for Penetration Testing Research

2.5 Research Gap and Positioning of the Current Study

2.6 Literature Summary

3. Research Design and Methodology

3.1 Research Design

3.2 Research Methods and Materials

4. Expected Outcomes

5. Timeline

6. Conclusion

6. References

1. Introduction

1.1 Context and Background

The rapid advancement of digital technology has significantly transformed modern society, enabling businesses and individuals to rely heavily on interconnected information systems, cloud computing platforms, and Internet of Things (IoT) devices. These technologies have made operations more efficient, scalable, and accessible, but they have also made it easier for bad actors to launch cyberattacks. As systems get more complex and connected, it gets harder to find and fix security holes, which leads to more attacks that are both more frequent and more advanced. Recent global cybersecurity reports show that cyber threats are always changing and targeting sensitive data, critical infrastructure, and key services in the healthcare, banking, government, and industrial sectors [1]. Cybersecurity has become a very important area of study and practice that is focused on keeping systems, networks, and data safe from unauthorized access, use, and interference.

Proactive security assessment is an important part of cybersecurity. It looks for and fixes weaknesses before attackers can use them. A lot of people use penetration testing as a proactive way to check their security. This means pretending to be a real cyberattack to see how well security measures work and find weaknesses in systems, apps, and networks. There are several steps in a standard penetration testing process: reconnaissance, vulnerability assessment, exploitation, privilege escalation, and post-exploitation reporting [2]. Penetration testing, which mimics how an attacker would act, gives important information about possible attack paths and system weaknesses.

Historically, penetration testing has been done by hand by very skilled security experts. Manual penetration testing is a good way to find complex and contextual vulnerabilities, but it is time-consuming, requires a lot of resources, and depends a lot on the tester's skill. Because of this, many companies only do penetration tests every now and then, often just to meet regulatory requirements. This leaves systems open to new threats that are found between tests. The global shortage of skilled cybersecurity professionals further limits the scalability of manual penetration testing methodologies [3].

To solve these problems, automated penetration testing tools and vulnerability scanners have been made. Tools like Nessus, OpenVAS, and Metasploit use established protocols, vulnerability repositories, and exploit frameworks to automate parts of finding and using vulnerabilities [4]. Compared to manual testing, these tools make things much more efficient and cover a wider range of issues. Even so, they are mostly static and rely on known vulnerability signatures. Because of this, they can't adapt to new situations, learn from past evaluations, or come up with complex multi-step attack plans. This limitation is particularly problematic in modern dynamic contexts where system configurations and threat scenarios change rapidly.

Recent progress in Artificial Intelligence (AI) and Machine Learning (ML) has opened up new ways to make cybersecurity automation better. Reinforcement Learning (RL), a branch of

Machine Learning (ML), has attracted significant attention due to its ability to model sequential decision-making and adaptive behavior.

In reinforcement learning, an autonomous agent interacts with an environment by performing actions, assessing outcomes, and receiving rewards, with the objective of maximizing cumulative reward over time [5]. This learning paradigm closely resembles the conduct of a human attacker, who continuously observes system responses, modifies techniques, and combines multiple actions to achieve goals such as unauthorized access or privilege escalation.

At the same time, research environments for cybersecurity that use simulations have become more popular. Metasploitable2 and other intentionally weak platforms offer realistic but safe settings for penetration testing research and experimentation without breaking any laws or morals [6]. Researchers can do experiments over and over again and objectively judge security assessment methods in these environments. Combining reinforcement learning with simulation-based penetration testing environments is a promising way to make vulnerability assessment frameworks that are smart, adaptable, and self-sufficient.

1.2 Research Problem

Even though there are both manual and automated ways to test for security holes, it is still very hard to get scalable, adaptable, and smart security evaluations. Manual penetration testing requires specialized knowledge, a lot of experience, and a lot of time, making it impractical for ongoing security evaluation in large or quickly changing environments [3]. Additionally, organizations often lack the resources to conduct regular penetration tests, which delays the discovery of vulnerabilities and makes them more vulnerable to cyber threats.

Automated penetration testing tools try to fix these problems, but they also come with their own problems. Most of the tools we use today rely on pre-set vulnerability signatures, rule-based reasoning, and static exploit databases. These tools are good at finding known vulnerabilities, but they can't analyze system states on their own, adapt to new settings, or come up with multi-step attack plans that look like real attacker behavior [7]. As a result, they often fail to see complex vulnerabilities that come from chained attacks, lateral movement, or privilege escalation.

Reinforcement learning has been successfully used in many areas of cybersecurity, such as intrusion detection and network defense. However, it is still limited in its use in automated penetration testing. Contemporary research often relies on overly simplified environments or theoretical frameworks that do not integrate genuine penetration testing tools or realistic attack scenarios [8]. Furthermore, concerns regarding state representation, reward function development, training efficacy, and assessment metrics largely remain unresolved. This paper tackles the core research problem of the lack of an intelligent, adaptive, and autonomous penetration testing framework that employs reinforcement learning for efficient vulnerability assessment and attack-path optimization in realistic yet secure settings.

1.3 Research Aim and Objectives

The goal of this research is to create, build, and test an automated penetration testing framework that uses reinforcement learning to intelligently find weaknesses and adaptively find attack paths in a simulated environment.

This project will look at current ways of doing penetration testing and how reinforcement learning can be used in cybersecurity to find gaps in research and limits. A penetration testing environment based on simulation will be created on a platform that is purposely vulnerable, such as Metasploitable2. A reinforcement learning agent will be developed to perform reconnaissance, detect vulnerabilities, exploit them, and escalate privileges through interaction with the environment. The framework's effectiveness will be evaluated through objective metrics such as exploit success rate, efficiency, and learning convergence, and compared against traditional automated penetration testing techniques. We will also look at ethical issues, limitations, and possible future research paths.

1.4 Significance of the Research

This research is significant from both academic and pragmatic perspectives. This study contributes to the body of knowledge on AI-driven cybersecurity by demonstrating the use of reinforcement learning in offensive security operations. The study links theoretical reinforcement learning models to actual penetration testing procedures, providing insights into advanced attacker simulation and dynamic vulnerability assessment.

The suggested architecture can be used in controlled environments for cybersecurity education, training, and research experimentation. It might also help with the design of future automated penetration testing systems that offer ongoing and flexible security assessments. This research could lead to better security for organizations, lower the risk of cyber attacks, and help build stronger digital infrastructures in the long run.

2. Literature Review

The goal of this literature review is to critically look at the most recent research on automated penetration testing, reinforcement learning in cybersecurity, and security assessment settings that use simulations. The evaluation aims to identify key trends, theoretical foundations, technical approaches, and research gaps related to the proposed subject. The literature includes studies published between 2008 and 2024 that look at traditional penetration testing tools, security automation driven by machine learning, and reinforcement learning methods used in both offensive and defensive cybersecurity areas.

2.1 Traditional and Automated Penetration Testing Approaches

Penetration testing is widely recognized as a crucial component of proactive cybersecurity assessment. The first steps in penetration testing were mostly done by hand and followed systematic methods set out by NIST and OWASP [2]. Manual penetration testing helps find

complex vulnerabilities and security gaps that depend on the situation, but it is limited by high costs, long execution times, and the need for specialized knowledge. As digital infrastructures get bigger and more complicated, manual methods don't provide ongoing and complete security coverage [3].

To get around these problems, automated vulnerability scanners and penetration testing tools were made. Tools like Nessus and OpenVAS automate the process of finding vulnerabilities by comparing system settings to known vulnerability databases [4]. Metasploit enhances automation by providing exploit modules that can be utilized against known vulnerabilities. These technologies significantly improve efficiency and standardization; however, they rely on fixed logic and predefined signatures. Because of this, they can't change quickly to new situations or think about multi-step attack pathways [7].

A lot of studies have shown that automated vulnerability scanners don't work well in real life. Wang et al. assert that automated scanners frequently produce false positives and inadequately identify chained attacks associated with privilege escalation and lateral movement [7]. These limitations demonstrate that traditional automation does not possess the necessary intelligence to genuinely emulate attacker behavior.

2.2 Machine Learning in Penetration Testing and Security Automation

Machine learning techniques have been increasingly employed in cybersecurity tasks, encompassing intrusion detection, malware classification, and anomaly detection [5]. In penetration testing, machine learning has mostly been used to improve how vulnerabilities are classified, how exploits are prioritized, and how attacks are predicted.

DeepExploit, introduced by Shin et al. [9], was the first attempt to use machine learning to automate penetration testing. DeepExploit uses machine learning to pick exploits based on the results of scans. This makes it more efficient than methods that only use rules. The method relies on supervised learning and requires training data that has been labeled. Consequently, it cannot autonomously adjust to new environments or devise long-term offensive strategies. This limitation emphasizes the need for reinforcement learning techniques proficient in modeling sequential decision-making.

Further research has examined attack graph-based models for automated security assessment. Attack graphs provide a structured representation of potential attack paths; however, they often require significant human configuration and prior knowledge of system vulnerabilities [8]. Attack graphs are also mostly static, meaning they don't change when the system setup or the attacker's behavior changes.

2.3 Reinforcement Learning in Cybersecurity

Reinforcement learning has attracted significant attention in cybersecurity due to its ability to learn optimal behavior through interaction with the environment rather than relying on labeled datasets [5]. Reinforcement learning has been successfully applied in defensive cybersecurity

domains, such as intrusion detection, adaptive firewall configuration, and automated incident response [10]. These studies demonstrate that reinforcement learning agents can adapt to evolving threats and improve defensive strategies over time.

In offensive cybersecurity research, reinforcement learning has been utilized to replicate intelligent attacker behavior and autonomous cyber operations. Ghanem and Chen demonstrated that reinforcement learning agents can recognize effective attack strategies and adapt to changing network conditions [10]. Nonetheless, numerous RL-based studies rely on abstract or overly simplified systems that do not integrate genuine penetration testing tools or realistic vulnerabilities. As a result, they are not very useful for real-world penetration testing.

There are many technological problems that come up when trying to use reinforcement learning in penetration testing. This includes coming up with important state representations, making reward functions that encourage real attack advancement, and making training more efficient in complex situations [11]. Poorly designed reward functions can cause agents to act in ways that aren't realistic or aren't important, and large state spaces can make computing costs much higher.

2.4 Simulation-Based Environments for Penetration Testing Research

Simulation-based environments are crucial in penetration testing research, particularly for testing offensive security strategies. Metasploitable2, DVWA, and OWASP Juice Shop are examples of intentionally vulnerable platforms that provide real vulnerabilities in safe settings, making it possible to test safely and ethically [6]. These settings make it easier to do experiments that can be repeated and make it possible to objectively judge security assessment methods.

Metasploitable2 is frequently employed in academic research due to its diverse range of exploitable services and misconfigurations [6]. Studies show that these environments are good for training and testing automated attack frameworks, especially those that use reinforcement learning. However, contemporary research often lacks standardized evaluation metrics and benchmarking methodologies, making it challenging to compare outcomes across different RL-based penetration testing methods [11].

2.5 Research Gap and Positioning of the Current Study

The literature reviewed reveals numerous critical shortcomings. At first, traditional automated penetration testing methods are fixed and don't have any adaptive intelligence. Second, penetration testing methods that use machine learning are often limited to supervised learning and can't come up with multi-step attacks on their own. Third, studies on reinforcement learning in cybersecurity frequently rely on abstract environments that do not accurately reflect real-world penetration testing workflows. In the end, there aren't any standardized ways to evaluate reinforcement learning-based penetration testing systems that work in real-world situations.

This research addresses these shortcomings by providing an automated penetration testing system grounded in reinforcement learning, integrating genuine penetration testing tools within

a realistic simulation environment. This study aims to improve intelligent vulnerability assessment by conceptualizing penetration testing as a sequential decision-making problem and evaluating the framework using quantitative metrics.

2.6 Literature Summary

In summary, human penetration testing is very thorough but can't be done on a large scale. Automated methods are quick but not very flexible. Machine learning techniques improve some parts of security automation, but they can't fully replicate the behavior of advanced attackers. Reinforcement learning is a feasible choice due to its capacity for adaptive, sequential decision-making; however, its application in automated penetration testing remains insufficiently explored in practical contexts. The identified deficiencies in the literature substantiate the proposed research and clarify its contribution to AI-driven cybersecurity.

3. Research Design and Methodology

3.1 Research Design

This research utilizes a simulation-based experimental methodology to assess the feasibility and effectiveness of reinforcement learning in automated penetration testing. A simulation-based methodology is particularly appropriate for this research, as penetration testing involves invasive and potentially harmful actions that cannot be conducted ethically or legally on live production systems. Conducting research in a controlled virtual environment allows researchers to safely recreate real-life assault situations without putting real infrastructure at risk or breaking legal and ethical rules [2].

Simulation-based research is widely employed in cybersecurity studies, especially in offensive security and penetration testing, as it enables reproducible experimentation and regulated evaluation of attack methodologies. Using a purposely vulnerable platform like Metasploitable2 makes the study environment very similar to real-world system vulnerabilities while still giving full control over the state and configuration of the system [6]. This method makes it easier to repeat experiments in the same conditions, which is important for testing the learning behavior and performance stability of reinforcement learning agents.

The experimental design makes it easier to develop and test things over and over again. The reinforcement learning agent trains over many episodes, during which it interacts with the simulated environment, watches how the system reacts, and changes its attack plan based on what it learns. This method makes it easier to look at how the agent's behavior changes over time in a systematic way, including how well it learns, how flexible it is, and how efficient it is. The experimental methodology enables a direct comparison between the proposed reinforcement learning-based framework and traditional automated penetration testing methods, providing empirical evidence of their comparative performance.

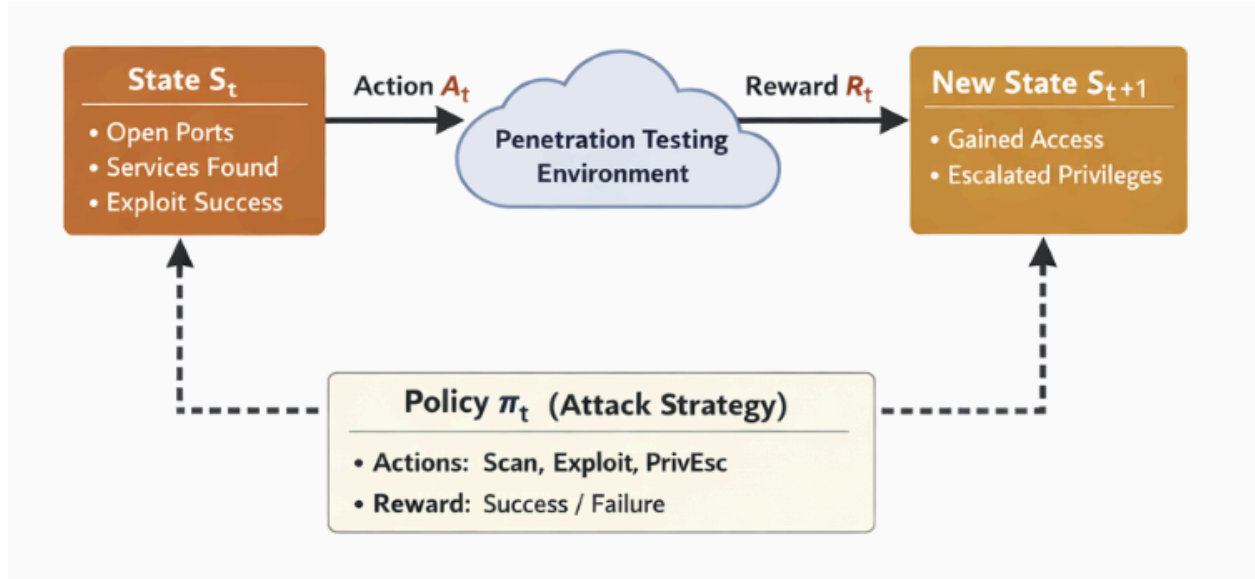


Figure 1: Modelling Penetration Testing as a Markov Decision Process

3.2 Research Methods and Materials

This study employs a quantitative research methodology, allowing for the assessment of the proposed automated penetration testing framework's efficacy through objective and measurable performance criteria. Quantitative methods are appropriate for this study because essential outcomes, such as exploit success rate, time to compromise, learning convergence, and attack efficiency, can be quantitatively evaluated and statistically analyzed. This methodology enables a structured comparison between reinforcement learning-based penetration testing and traditional automated tools [5].

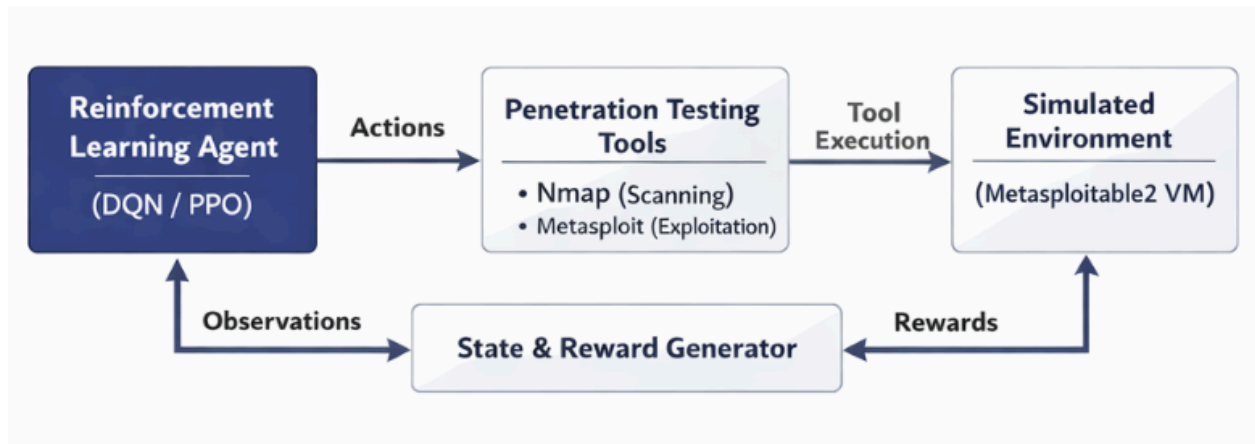


Figure 2: Architecture of the Reinforcement Learning-Based Automated Penetration Testing Framework

The research strategy combines experimental research with the creation of software. The experimental part focuses on teaching and testing a reinforcement learning agent in a simulated

penetration testing environment. The software development part involves creating and putting into use a custom reinforcement learning environment that includes penetration testing tools and an agent that makes decisions based on what it learns. This integrated approach is suitable as the research aims to both conceptualize an intelligent penetration testing framework and to develop and empirically validate a functional prototype.

The penetration testing process is organized as a Markov Decision Process (MDP), which gives reinforcement learning problems about making decisions in order a formal mathematical basis [5]. In this method, the environment stands for the system being tested, and the reinforcement learning agent stands for a smart attacker. At every step, the agent sees the current state of the system, picks an action, and gets a reward based on how well that action worked. Over time, the agent learns a policy that maximizes the total reward.

In the proposed framework, states represent the agent's comprehension of the target system. This could include information like open ports, active services, vulnerability assessment results, and the results of previous exploits. Network scanning, service enumeration, exploitation attempts, and privilege escalation are all actions that are part of penetration testing. Rewards are meant to encourage big improvements in attack strategies, like finding services, getting shell access, or getting higher privilege levels. To encourage effective investigation, unproductive or unnecessary actions are given few or no rewards.

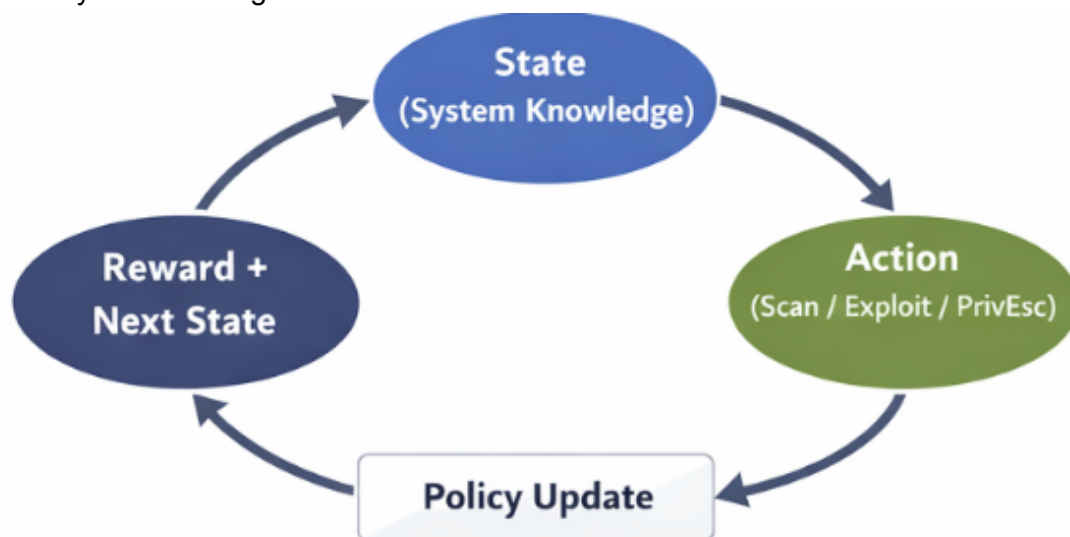


Figure 3: Reinforcement Learning Interaction Cycle Applied to Penetration Testing

The reinforcement learning agent will first be built using Deep Q-Networks (DQN). This is a good choice for environments with discrete action spaces and has been used a lot in problems where decisions have to be made in order. To make training more stable and speed up convergence, experience replay and target networks will be used. Based on experimental results, various reinforcement learning algorithms, including Proximal Policy Optimization (PPO), may be examined to tackle more complex decision spaces.

The framework includes real penetration testing tools, like Metasploit for exploitation and Nmap for network scanning, in the simulated environment. The outputs from the tools are looked at and changed into state representations that the reinforcement learning agent can use. This integration ensures that the agent operates on genuine information rather than abstract or synthetic data, thereby augmenting the practical relevance of the research.

This study, unlike traditional data-driven research, does not rely on external databases. Agent-environment interaction creates data in real time. During training and evaluation, detailed logs are kept that include the actions taken, the moods observed, the incentives received, and the results of each episode. This information is used to look at how people learn, how well they do, and how well the system works. This method is appropriate for reinforcement learning, which relies on experiential learning rather than supervised datasets [5].

Different quantitative measures will be used to evaluate the proposed framework. These include the exploit success rate, which measures how often the agent successfully breaches the target system; the time to compromise, which measures how many actions it takes to get a successful exploit; and learning convergence, which shows how quickly the agent's performance improves during training. We will also look at other factors, such as how well the system can adapt to changes in the environment and how well it works.

Ethical considerations are a core component of this research. All investigations are conducted exclusively within isolated virtual machines that lack external network connectivity. There are no real systems, personal data, or private information involved. The research complies with ethical cybersecurity principles and guarantees that the produced techniques are utilized exclusively for defensive, educational, and research objectives [2].

The implementation environment uses Python, which has full support for reinforcement learning and the use of cybersecurity tools. Stable Baselines3 and OpenAI Gym make it easier to improve reinforcement learning by giving you standard ways to make environments and train agents. VirtualBox and VMware are two examples of virtualization platforms that can be used to host the emulated target environment. The hardware prerequisites consist of a normal workstation capable of operating virtual machines, with optional GPU support to enhance training if necessary.

```

Input:
    Simulated penetration testing environment E
    Reinforcement learning agent A
    Maximum number of episodes N
    Maximum steps per episode T

Output:
    Trained penetration testing policy  $\pi^*$ 

1: Initialize reinforcement learning agent A with random policy  $\pi$ 
2: for episode = 1 to N do
3:     Reset environment E
4:     Observe initial state  $s_0$ 
5:     for step = 1 to T do
6:         Select action  $a_t$  based on policy  $\pi(s_t)$ 
7:         Execute action  $a_t$  in environment E
8:         Observe next state  $s_{t+1}$  and reward  $r_t$ 
9:         Store transition  $(s_t, a_t, r_t, s_{t+1})$ 
10:        Update policy  $\pi$  using reinforcement learning algorithm
11:        if terminal condition reached then
12:            break
13:        end if
14:         $s_t \leftarrow s_{t+1}$ 
15:    end for
16: end for
17: Return learned policy  $\pi^*$ 

```

Structure of system: Reinforcement Learning-Based Automated Penetration Testing

4. Expected Outcomes

The goal of this project is to make an automated penetration testing framework that uses reinforcement learning to find vulnerabilities and adaptively follow attack paths in a simulated environment. Compared to traditional rule-based automated testing tools, the framework is expected to be smarter and more flexible, especially in situations where there are multiple steps in an attack.

This project aims to provide a basic model that can be built on to create a corporate cybersecurity solution, in addition to its academic value. The proposed architecture could help businesses by making it easier to do ongoing, smart, and cheap vulnerability assessments, which would improve the way penetration testing is done now. The results, obtained from a

controlled environment, are intended to inform future real-world application and commercialization, subject to further validation and ethical considerations.

5. Timeline

The research will take place over eight weeks, following a structured and staged timeline to ensure that progress is made in a systematic way and that the project is managed well. During Week 1, an extensive review of the literature will be undertaken to evaluate contemporary studies on penetration testing, reinforcement learning, and automation in cybersecurity. This phase will focus on identifying key theoretical foundations, current methodologies, and research gaps related to the proposed subject.

The research environment will be built and the reinforcement learning framework will be made during Weeks 2 and 3. In this phase, a simulated penetration testing environment is set up with a platform that is meant to be vulnerable. Virtual machines are built, and the reinforcement learning components, such as the state representation, action space, and reward function, are defined.

From Weeks 4 to 6, the implementation and training phase will take place. In this stage, the reinforcement learning agent will be built using the right algorithms and libraries and will learn by interacting with the simulated environment. To improve learning stability and performance, iterative refinement and parameter optimization will be used.

In Week 7, the trained model will be tested and analyzed using quantitative performance metrics like the success rate of exploits, learning convergence, and efficiency. A comparative analysis employing standard automated penetration testing techniques will be performed to assess the effectiveness of the proposed framework.

In the end, Week 8 will be all about writing the final thesis and recording the research results. This includes writing, reviewing, and improving the dissertation to make sure it is clear, makes sense, and follows academic and institutional rules.

6. Conclusion

This research proposal delineates a framework for the integration of reinforcement learning in automated penetration testing within simulated environments. Traditional manual and rule-based penetration testing methods have problems with scalability, adaptability, and intelligence, especially in modern dynamic systems. The proposed work conceptualizes penetration testing as a sequential decision-making process and employs reinforcement learning to enable autonomous vulnerability identification and attack-path exploration.

The research utilizes a simulation-based experimental framework to ensure ethical and secure experimentation, integrating genuine penetration testing tools and vulnerable platforms. Quantitative evaluation metrics assess the effectiveness, efficiency, and adaptability of the proposed framework.

The study seeks to advance academic research by furthering the application of reinforcement learning in offensive cybersecurity and to enhance practical security evaluation by directing the development of intelligent automated penetration testing tools. This research presents a feasible and ethically sound approach to enhancing vulnerability assessment through AI-driven automation.

7. References

- [1] ENISA, "ENISA Threat Landscape Report," 2023.
- [2] K. Scarfone, P. Mell, and M. Souppaya, Guide to Penetration Testing and Vulnerability Assessment, NIST, 2008.
- [3] S. Gupta, "The Cybersecurity Skills Gap," Cyber Defense Review, 2020.
- [4] R. Beale, OpenVAS: Open Vulnerability Assessment System, Springer, 2017.
- [5] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, MIT Press, 2018.
- [6] Rapid7, "Metasploitable2: Vulnerable Linux VM," 2020.
- [7] J. Wang et al., "Limitations of Automated Vulnerability Scanners," IEEE Transactions on Information Security, 2020.
- [8] H. M. Almohri, D. Yao, and S. Pan, "Automated Security Assessment: Challenges and Opportunities," 2020.
- [9] Y. Shin et al., "DeepExploit: Fully Automatic Penetration Test Framework," BlackHat Asia, 2019.
- [10] M. C. Ghanem and T. M. Chen, "Reinforcement Learning for Autonomous Cyber Operations," 2020.
- [11] L. Xie and D. Yao, "Challenges in RL-based Cybersecurity Simulations," IEEE Security & Privacy, 2022.