# Extended Literature Review

# Introduction

todo

## Review 1:

### Introduction

Summary

The blog emphasizes the crucial role of penetration testing in safeguarding businesses from cyber threats. It begins by stressing that successful businesses inevitably attract cybercriminals, making security paramount. Penetration testing (pentesting) is introduced as a method to identify system vulnerabilities by simulating cyberattacks. It helps businesses uncover weak points, understand potential attack scenarios, and prioritize security enhancements. The article breaks down penetration testing into several stages: information gathering, scanning, exploiting, maintaining access, covering tracks, and reporting. It also touches on the skillsets required for penetration testing, recommends training and practice, and discusses the importance of reporting findings for security improvements.

The authors continue on to show a full walkthrough on how to set up a test environment, as well as highlight various pentesting tools and their practical application, including Whois for information gathering, Shodan for searching reachable servers, Nmap for network scanning, Metasploit for exploiting vulnerabilities, Wireshark for packet analysis, John the Ripper for password cracking, and Burp Suite for web application testing. It stresses the need to prioritize vulnerabilities based on risk and common attack patterns, focusing on the most easily exploitable weaknesses. Overall, the article provides a comprehensive overview of penetration testing and its essential role in maintaining a secure business environment.

### Methodologies

The article outlines the methodologies involved in penetration testing, offering a structured approach to identifying and addressing security vulnerabilities. These methodologies are categorized into several stages, and each stage plays a crucial role in the penetration testing process.

The first stage is "Information Gathering." In this phase, testers gather as much data as possible about the target system. This includes collecting information about IP addresses, servers, frameworks, subdomains, and operating systems. A thorough understanding of the target is essential as it serves as the foundation for the subsequent testing phases. This stage is akin to reconnaissance in the world of cybersecurity, helping testers identify potential entry points.

The next stage is "Scanning." It involves the systematic exploration of the target system for potential weaknesses. Rather than haphazardly testing each possible vulnerability, scanning allows testers to intelligently create a list of weaknesses to focus on. Testers look for known vulnerabilities in versions of frameworks or tools and assess how the application responds to intrusion attempts. This stage significantly improves the efficiency of the testing process by narrowing down the areas of concern.

Following scanning, "Exploiting" comes into play. This stage is where actual penetration occurs. Testers attempt to exploit the security weaknesses they've identified. This could involve gaining access to a component or data, causing system failures, or modifying data. It's a critical phase that mimics the actions of a real attacker but with the goal of understanding the potential impact without causing actual harm. To replicate an attacker's actions accurately, testers often use virtual environments to safely perform these tests.

These methodologies enable penetration testers to thoroughly assess a system's security posture, from information gathering and vulnerability scanning to the exploitation of weaknesses. This structured approach is designed to help organizations proactively identify and address security flaws, ultimately strengthening their overall security.

### Main Findings

The main findings in the article revolve around the importance of penetration testing in the context of cybersecurity. The article highlights that businesses, especially those experiencing significant growth, are prone to attracting the attention of cybercriminals. As a result, cybersecurity becomes an imperative concern. The primary finding is that penetration testing, also known as pentesting, plays a pivotal role in addressing this challenge.

The article underscores that penetration testing is a multi-stage process with distinct stages, starting with information gathering, followed by scanning, exploiting, maintaining access, covering tracks, and concluding with reporting. It emphasizes the need for meticulous information gathering to understand the target system, followed by systematic scanning to identify potential vulnerabilities. The exploitation stage mimics actual cyberattacks to gauge the impact without causing harm, while maintaining access and covering tracks assess an attacker's ability to maintain a persistent presence and avoid detection. Finally, the reporting phase is crucial as it captures all the findings and serves as a reference for security fixes and updates.

The main finding is that penetration testing is a proactive, systematic, and effective approach to identify and rectify security vulnerabilities within an organization's systems and infrastructure. By replicating the actions of potential attackers, it helps organizations strengthen their security measures and protect their assets from cyber threats. The article underscores the significance of penetration testing as a foundational component of a robust cybersecurity strategy.

### Relevance to Your Course Content

As the specific phases of penetration testing vary depending on the source, their fundamental concept remains the same. Since my course is built specifically around these phases, it was important to find a credible source from which to structure my modules. This article not only provides a defined structure of these phases, but it also provides practical examples and context. This article is a great resource to provide additional context on the phases, information on how penetration testing works in practice, how to set up a testing environment, and even practical examples of various popular tools.

Some of these tools, including Nmap, Metasploit, and Burp Suite, are commonly used in the field and are essential for carrying out penetration tests and identifying vulnerabilities. While these tools are not AI tools themselves, they can be integrated with AI and machine learning techniques to enhance their capabilities. For example, AI can be applied to automate the detection of vulnerabilities or streamline the exploitation of weaknesses using data-driven approaches. My course will explore how AI can augment the functionalities of various tools and discuss AI-driven penetration testing tools that leverage machine learning for improved threat identification and exploitation. While I will provide some background, this course serves as a great tool to provide additional context. This article provides a foundational understanding of penetration testing, which can serve as a strong starting point for students to dive deeper into the AI-driven aspects of the course.

## Review 2:

### **Introduction**

The article "Autonomous Security Analysis and Penetration Testing" from Arizona State University introduces an innovative framework, designed to address the growing challenge of evaluating network security amidst the complexity of expanding networks and a shortage of cybersecurity professionals. By leveraging advanced Reinforcement Learning (RL) techniques based on DeepQ Networks (DQN), this framework integrates vulnerability information into penetration testing (PT) processes. It associates RL reward values with Common Vulnerability Scoring System (CVSS) scores, enabling prioritization of the most critical vulnerabilities. The result is a highly efficient, automated PT system that can significantly reduce assessment time and improve overall efficiency.

**Summary**

Previous research in the field of automating PT through RL has predominantly focused on smaller networks and often failed to harness vulnerability information effectively. These traditional AI models have struggled to grasp the intricacies of real-world networks, falling short in accounting for the specific network structure, distribution of vulnerabilities, or correlation between vulnerabilities and exploitation probabilities. This limitation has led to difficulty in prioritizing vulnerabilities, resulting in less accurate and efficient security assessments. In order to obtain essential information about a target and its associated vulnerabilities, these methods often rely on known sources, scans, or manual analysis for identification. This overall failure of traditional AI models unable to comprehensively understand the nuances of the dynamic and complex security landscapes of modern networks has resulted in a desperate need for a more comprehensive approach to PT.

Recognizing these limitations, the authors introduced ASAP as an innovative approach to security analysis and PT. This autonomous system not only understands the interconnectedness of vulnerabilities and their relation to a network's structure, but it also leverages an RL reward system based on vulnerability severity and exploitability. The approach adopted by the authors emphasizes domain-specific modeling, integrating the Common Vulnerability Scoring System (CVSS) to quantify known vulnerabilities. This system tracks the severity of vulnerabilities and the ease or difficulty of exploiting them, allowing for a more comprehensive understanding of the network's security landscape. This modeling approach harnesses state-transition diagrams to visualize the most optimal penetration testing policy for the network. These diagrams represent different network states and the associated actions, including probability values derived from the vulnerability's access complexity (AC). By generating autonomous attack plans and validating them against real-world networks, ASAP creates a comprehensive map of security threats and potential attack paths.This approach ensures efficiency not only in smaller networks but also in large-scale environments, demonstrating its scalability and exceptional performance.

To enable autonomous PT, the authors adopt a RL-based AI algorithm to identify the optimal “attack path that maximizes the reward value for the pentester.” [2] Reinforcement learning is a concept where an agent learns through the consequences of its interactions within an environment, focusing on long-term objectives, not unlike security professionals experimenting with attack strategies against vulnerabilities until successful exploitation. However, what sets their RL model apart from other traditional AI models in the PT domain, the authors propose using a Deep-Q Network (DQN) based RL model. As a DQ model learns directly from interactions with the environment by utilizing neural networks, it is more equipt handle diverse network conditions, including those that may not have been encountered during training. As such, their RL approach involves a dynamic interaction with the environment by considering current user privilege level, actions linked to vulnerability exploitation, the difficulty and probability of a successful action, reward values, and the decision-making process for the next action based on the current state. The outcome of this process is a carefully designed attack plan that “guides the security professional” through their next actions based on their user privilege and progression strategy. [2]

### **Methodologies**

The methodologies of ASAP involve a structured series of steps which enable efficient and effective PT. First, researchers employ popular scanners such as Nessus and OpenVAS to scan for vulnerabilities in the target network. The obtained scan information about the availability and accessibility of network services (e.g., open ports, protocols) and vulnerabilities within those services are then generated into an attack graph. This graph creates a visual representation of potential attack paths and relationships between different elements of a network “and dependencies between the vulnerabilities.” [2] Key Information from the attack graph is then converted into a structured format, known as a State Graph, and passed to the RL algorithm for further analysis.

The State Graph is designed to represent how privileges transition within the network and if a specific vulnerability leads to an exploit. If a vulnerability is found to directly link to an exploit, attributes such as the CVSS and the AC are extracted and stored for reference. The reward value corresponds to the vulnerability CVSS score – higher severity vulnerabilities that could have a greater potential impact if exploited get a higher reward. This information is critical for calculating exploit success probabilities and is used to define and build the previously mentione components needed for the RL algorithm including state of user privilege, actions, transition probability, reward values, and the agents decision policy. After confirming success of their exploits through log analysis, the state graph and any relevant threat information is generated into an attack plan.

After the attack plan is generated, it undergoes validation using a Python wrapper for the Metasploit framework, which assesses the effectiveness of the PT. If vulnerabilities and weaknesses are found in the target organization's network or systems, the findings are used to recommend actions to improve the security of the organization. These actions may include applying patches or making changes to the network based on the vulnerabilities and weaknesses discovered during the test. Once these changes are implemented, the attack graph, which represents the network's vulnerabilities and potential attack paths, can be updated to reflect the new security measures. The system can then be retested to ensure that the implemented changes have effectively addressed the identified security issues and that the network's security posture has improved. This cyclical process of testing, improving, and retesting to enhance the organization's security is vital in cybersecurity, as it ensures that security measures remain robust, modern, and adaptive to evolving threats.

### **Main Findings**

Overall, the main findings of the article emphasize that the ASAP framework, with its use of RL and attack graphs, offers a more efficient and effective approach to penetration testing. It not only reduces the manual effort and time required but also uncovers latent attack paths that manual testing might miss, ultimately improving the overall security assessment process.

A case study involving the pentesting of an enterprise network with an industrial control system and IoT devices was conducted. The network consisted of 16 hosts distributed across three networks and offered a mix of Windows and Linux systems. The goal was to compromise email information by exploiting vulnerabilities on the SMTP service and infiltrating the IoT subsystem through a vulnerability in the gateway machine. The study explored the impact of variations in the discount factor to decide how much importance to give to future rewards in the decision-making process, with values closer to 0 prioritizing immediate rewards and values closer to 1 prioritizine future rewards and batch size to explore the amount of interactions the AI system uses at once to learn and improve its policy. Variations in These parameters were explored to assess their impact on learning the optimal policy for the RL agent.

The findings revealed that the DQN algorithm reached an effective solution quickly for different varaitions in discount factor, with the optimal value being around 0.8. Higher variations in discount factor, 0.9 and 0.99, caused the agent to take more time to learn and make decisions as it spent more time exploring potential future outcomes. Similarly, reward value diminishes considerably the more the agent prioritizes long-term rewards, regardless of the number of interactions the system uses at once to learn and improve. The agent's reward value was highest for the optimal discount around 0.8, especially with a batch size of 16. The study showed that with larger batch sizes like 32 or 64, the AI's performance was not as good. However, the reasearchers note that this observation might not hold true for very large networks due to the increased complexity and scale of such networks which can lead to different dynamics in the learning process for the AI system.

Additionally, a scalability experiment was conducted on a simulated flat network with 300 hosts and three vulnerabilities. The results demonstrated the framework's ability to provide an optimal attack plan within a short time, even in scenarios where the decision between exploration and exploitation is not straightforward. What sets the ASAP framework apart from manual methods is its distinct strategy for penetration testing. Unlike traditional manual testing, AI-based approaches, like ASAP, prioritize the exploitation of certain vulnerabilities before others, resulting in more efficient and effective PT. This data-driven approach involves adapting to the unique network environment and the characteristics of vulnerabilities. Sometimes, starting with less challenging vulnerabilities can lead to a more efficient overall penetration test. The ASAP framework's adaptability and its consideration of vulnerability characteristics make it a valuable asset in the field of cybersecurity and offers a significant reduction in the time and effort required over traditional manual approaches.

### **Relevance to Your Course Content**

This article is highly relevant to my course as it aligns with the central theme of harnessing AI techniques for offensive strategies in PT. In the article, an ASAP framework provides a practical example of various aspects touched on in my course, including AI-driven PT tools, RL, DQ, and real-world applications of AI in security assessments. This in-depth exploration forms the foundational knowledge for the course's focus on AI-driven penetration testing techniques and demonstrates the efficiency and effectiveness of AI in identifying security vulnerabilities.

Furthermore, the article's application of DQN as a deep reinforcement learning technique serves as a practical example of how machine learning models can be used for identifying vulnerabilities and threats. This is in line with my course's content, which covers the training of machine learning models for this purpose. The ASAP framework showcased in the article also highlights the scalability of AI-driven techniques, making it suitable for large-scale networks, which is pertinent to provide context of real-world scenarios. It demonstrates the transformative power of AI in penetration testing and how it can uncover latent attack paths, offering valuable insights about AI's practical applications in identifying vulnerabilities and optimizing security assessments. The article provides a comprehensive example of how advanced AI techniques can be applied in the context of penetration testing. This can provide students with a concrete example of AI-driven penetration testing offers a comprehensive understanding of how AI techniques can be practically applied to enhance security assessments.

# Conclusion

todo