# Extended Literature Review

# Introduction

In an age defined by the relentless increase of technology, the growin digital landscape has become both a playground for innovation and a battleground for cyber threats. As businesses and organizations increasingly rely on technology to operate and store critical data, the significance of safeguarding digital assets against potential adversaries becomes more crucial each day. This realization has given rise to Penetration Testing (PT) as a vital and proactive strategy that allows organizations to simulate cyberattacks on their systems, IN AN ATTEMPT to discover and eliminate dangerous vulnerabilities

This literature review delves into the core concepts and methodologies of penetration testing, dissecting the stages involved, exploring the tools utilized, and highlighting the practical implications …..2nd article… in real-world cybersecurity scenarios.

Acronyms: PT, RL

## Review 1: Penetration Testing: Practical Introduction & Tutorials

### **Introduction**

The “Penetration Testing: Practical Introduction & Tutorials" blog was published by Splunk, a leading authority in the realm of cybersecurity, in order to emphasize the [??] of digital defense. [why should we care about splunk]. This post introduces PT as an offensive strategy designed to identify system vulnerabilities by simulating cyberattacks. The primary goal of this process is to uncover weak points, understand potential attack scenarios, and analyze the severity of vulnerabilities. By acting like a genuine attacker, the target systems are given an opportunity to prioritize their most critical security enhancements before hit by a real attack. Along with exploring how PT reinforces security, this blog also explores the intricate stages involved in the process, illustrates real-world examples of PT tasks, and provides practical [??] of PT tools. Overall, this post not only explores essential foundational themes but also builds a solid foundation for the PT process by providing a structured outline of the PT phases; It offers a comprehensive understanding of the importance of PT in cybersecurity and its practical relevance to safeguarding businesses in an age where digital threats are both inevitable and evolving.

**Summary**

This article provides an in-depth exploration of the stages involved in PT. By providing a structured approach to the testing process, these stages ensure the PT process is methodical, ethical, and comprehensive evaluation of a system's security. They help in replicating real-world attack scenarios, identifying vulnerabilities, assessing their severity, and providing a clear roadmap for strengthening cybersecurity measures. While various sources may present slightly different PT stage breakdowns, this blog distinguishes itself by offering not only a comprehensive breakdown of these stages but also by providing concrete examples, clear definitions, and practical insights into their importance. The stages outlined in this Splunk article include information gathering, scanning, exploiting, maintaining access, covering tracks, and reporting. This structured approach allows security professionals to systematically evaluate vulnerabilities and prioritize security enhancements effectively

The article also emphasizes the essential skills required for effective PT and underscores the significance of continuous training and practice in mastering these skills. These skills encompass a wide range of technical and practical competencies, ranging from “knowledge of operating systems and networking”, an understanding of “authentication and authorization mechanisms”, to a strong foundation in programming. [1] The author emphasizes the significance of continuous learning and skill development, whether through self-driven efforts like reading and exploring online resources or a formal education. Whichever method is used, the need for hands-on practice in emphasized as it allows individuals to apply their knowledge and techniques in a controlled environment.

The authors then continue on to practical demonstations, beginning with how to set up a controlled test environment featuring an attacker machine and a victim machine using VMWare. This hands-on approach showcases the practical application of various PT tools throughout the stages, ensuring a comprehensive PT process. The initial steps involve obtaining crucial information about network-related aspects. For instance, they delve into domain-specific intelligence gathering with Whois, identifying vulnerable devices exposed on the internet through Shodan, and conducting a comprehensive network topology scan with Nmap. This foundational phase assists security professionals and network administrators in obtaining a comprehensive understanding of network infrastructure and potential security vulnerabilities.

After obtaining a thorough visuliation of the network, the article explores exploitation tools, such as password cracking with John the Ripper and web traffic interception and manipulation with Burp Suite. Additionally, it highlights the use of Metasploit exploits to establish a a “backdoor shell that [will allow them] to run commands on the victim system.” [1]

### **Methodologies**

This article outlines the methodologies involved in PT and offers a structured approach for identifying and addressing security vulnerabilities. These methodologies are categorized into several stages, and each stage plays a crucial role in the PT process. From information gathering and scanning to exploiting vulnerabilities and maintaining access, each step simulates real-world attack scenarios, helping organizations comprehend their system's weak points. By conducting PT, businesses can prioritize security enhancements effectively, thereby safeguarding their digital assets and sensitive data.

The first stage as defined by this article is ‘Information Gathering.’ This initial phase is akin to conducting reconnaissance, where the tester collects essential data about the target system or organization. The primary objective is to accumulate as much pertinent data as to better understand the surface area available for testing, and potentially pinpoint areas where security weaknesses may exist. Some common categories of information that penetration testers aim to collect in this stage include IP addresses, details about servers, knowledge about the frameworks or software in use, identification of subdomains, and information about the operating systems running on the target's network. Essentially, this phase is about building a comprehensive profile of the target to serve as the foundation for subsequent stages of the testing process

After identifying potential entry points or vulnerabilities in the target system during the previous phase, the tester begins to assess these points for possible weaknesses during the ‘Scanning’ phase. Its important that exploration of the target system is conducted systematically, rather than haphazardly testing each possible vulnerability. Not only is the blind approach more time-consuming, it is also significantly less effective. Therefore, to increase the chances of success in identifying significant vulnerabilities. testers look for known vulnerabilities in versions of frameworks or tools and assess how the application responds to intrusion attempts.[2] This stage significantly improves the efficiency of the testing process as it refines the list of concerning potential vulnerabilities to focus on.

Once vulnerabilities have been identified during the previous stages, pentesters actively exploit them during the ‘Exploitation’ phase. The goal is to simulate a real penetration or intrusion into the target system by either accessing data within the target system, intentionally trigger system failures, or making unauthorized changes to data or system configurations. [1] While this critical phase mimics the actions of a real attacker, the authors emphasize it is essential to maintain ethical and controlled PT practices, where the primary focus is on understanding the vulnerabilities rather than causing actual damage.

After successful exploitation, the testing focus shifts from gaining initial access to the 'Post-Exploitation' phases: 'Maintaining Access,' 'Covering Tracks,' and 'Reporting.' These phases, encompassing Steps 4 to 6, align with this primary purpose of assessing and improving security measures rather than engage in malicious actions.

Unlike malicious actions, these phases aim to evaluate and enhance security by examining the system's ability to not only detect unauthorized access but also log and store data related to security incidents. [2] By reporting their insights into findings in detail, pentesters aim to the enable organizations to fortify their defenses against real-world cyber threats.

### **Main Findings**

The main findings in the article revolve around the importance of PT in the context of cybersecurity. The article highlights that businesses, especially those experiencing significant growth, are prone to attracting the attention of cybercriminals and highlights PT as a foundational approach to safeguarding their digital assets. It is introduced as a proactive method for identifying system vulnerabilities by simulating cyberattacks to uncover weak points, anticipate potential attack scenarios, and assess the severity of vulnerabilities, ultimately enabling organizations to prioritize the most critical security enhancements.

The article also emphasizes that PT is a multi-stage process, each stage playing a crucial role in systematically evaluating a system's security and replicating real-world attack scenarios. This approach offers a methodical and ethical route, prioritizing a deep understanding of vulnerabilities without causing actual harm during the exploitation phase. Each stage in Splunks PT process contributes significantly to the overall efficiency and efficacy, ensuring a comprehensive assessment of an organization's defenses. Information gathering provides critical insights into the target, while scanning refines the focus by identifying specific vulnerabilities, thus preventing inefficient testing of unrelated weaknesses. The exploitation phase, although simulating an actual breach, strictly adheres to ethical principles to help rather than harm the target system. Then, following successful exploitation, the post-exploitation phases, maintaining access and covering tracks, assess an attacker's ability to maintain a persistent presence and avoid detection. Finally, the reporting phase is crucial for outlining and prioritizing vulnerabilities, guiding businesses to address easily exploitable weaknesses first.

Overall, the post highlights that PT is a proactive, systematic, and highly effective approach for identifying and addressing security vulnerabilities within an organization's system. By simulating the actions of potential attackers, it enables organizations to fortify their security measures and safeguard their valuable assets from complex cyber threats. . The article not only emphasizes the significance of PT but also provides a foundational framework for conducting PT through well-defined phases, all while upholding the essential ethical considerations in the process.

This structured and ethical approach ensures that PT not only identifies and exploits vulnerabilities but utilizes this data to equipt organizations with practical strategies to improve their overall security.

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

While the specific phases of PT may exhibit variations in terminology depending on the source, their fundamental organization and underlying concepts remain the same. As my course is specifically developed around these phases, it was important to find a credible source from which to structure the course modules. As such, the choice to adopt Splunk's phases into the curriculium is substantiated by their extensive expertise and experience within the field of cybersecurity. My goal is that by integrating these well-structured phases into the course, will ensure that students receive a solid and reliable foundation necessary to navigate the complex landscape of cybersecurity

This article bridges the gap between theoretical knowledge and practical implementation, making it an invaluable resource for my course content. It not only offers an in-depth exploration of PT stages, it also provides additional context, insights into practical applications, guidance on setting up testing environments, and real-world examples of popular tools. These practical examples vividly demonstrate the impact of PT tools in identifying vulnerabilities, creating custom exploits, and enhancing offensive strategies. By providing a blend of theoretical insights and hands-on experience, this article equips students with a solid foundation in PT, serving as the launchpad from which my course can seamlessly delve into the realm of AI-enhanced cybersecurity strategies.

Some of these tools discussed in this report, including Nmap, Metasploit, and Burp Suite, are commonly used in the field and are essential for carrying out PT and identifying vulnerabilities. While these tools are not AI tools themselves, they can be integrated with AI and Machine Learning (ML) 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 PT tools that leverage ML for improved threat identification and exploitation. While I will provide some background into discussed tools, this article serves as a great tool to provide additional context.

## 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