# Abstract

The modern integration of Artificial Intelligence (AI) into Penetration Testing (PT) is rapidly becoming a transformative force. This report conducts a comprehensive review of five formative research papers, , highlighting the escalating demand for AI-driven PT tools, the necessity for further research, and the motivation for innovative approaches to outpace cyber threats. Building from these insights, a short seminar course titled "Harnessing Artificial Intelligence for Penetration Testing" is designed. It aims to provide students with a comprehensive understanding of PT and the vital role of AI in enhancing efficiency and efficacy. This course goes beyond theoretical knowledge; it imparts practical skills and empowers students to navigate the complex realm of cybersecurity, combat adversaries, and actively contribute to a more secure digital environment.

# Introduction

In the complex landscape of cybersecurity, the art of penetration testing (PT) has emerged as a critical and dynamic discipline. PT, often referred to as ethical hacking, is the “offensive approach” of probing and assessing computer systems, networks, and applications “to actively identify vulnerabilities and then exploit them in the same way as a genuine attacker.” [1] Therefore, as modern digital tools become increasingly intelligent and interconnected, organizations face mounting challenges to safeguard their assets and data. The need for skilled professionals who can “provide organizations with actionable information about their security posture, enabling them to identify and prioritize areas of risk and improve their overall security,” [2] adeptly has never been more pressing.

PT is a multifaceted endeavor, that approaches cybersecurity in a series of meticulous steps. It begins with Gathering Information, where information about the target system and potential vulnerabilities is meticulously compiled. This is followed by the scanning phase, where potential vulnerabilities are verified as exploitable. Once vulnerabilities are confirmed, the Exploitation phase leverages these weaknesses to gain access to the target system. The following three phases, which include establishing persistence, minimizing traces of the intrusion, and documenting any results, fall into post-exploitation category. For the sake of this report and this course, these three phases will be referred to as post-exploitation, to allow for more of a focus on the first three phases.

Given the complex nature and substantial workload inherent to traditional penetration testing (PT), it necessitates significant expertise from penetration testers (pentesters). [3] However, due to a current critical shortage of skilled experts in the field, there is a fundamental need to embrace the automation group of AI, Machine Learning (ML), and Reinforcement Learning (RL). First, the growing complexity of networks demands a more sophisticated and efficient approach to identifying vulnerabilities. Traditional manual penetration testing methods, while valuable, are often time-consuming, resource-intensive, and may struggle to keep pace with the dynamic nature of cyber threats. Additionally, the incorporation of ML into automated PT systems has the potential to reduce “recurrent human errors” resulting from factors such as “tiredness, omission, and pressure.” [1] Automated PT can significantly reduce the time and resources required for comprehensive testing, making it a practical choice for organizations aiming to bolster their cybersecurity defenses. As such, the evolving field of automated PT utilizes “uses advanced algorithms, machine learning, and AI to scan systems for vulnerabilities,” [2] and significantly improving multiple facets of PT.

This recent surge in AI's capabilities and its potential to revolutionize every facet of PT highlight the importance of spreading knowledge of ‘Harnessing Artificial Intelligence for Penetration Testing.’ From introducing the basics of AI to delving into advanced topics such as AI-driven scanning, vulnerability assessment, and post-exploitation techniques, this course offers a comprehensive exploration not only of the current AI landscape but also its future possibilities. The primary goal is to not just keep pace with evolving threats but to leap ahead, embracing the most cutting-edge technologies in modern AI, and empowering innovative cybersecurity professionals to navigate the complex digital frontier with confidence.

# Literature Review

## Review 1

### Introduction

During the upcoming European Software Engineering Conference proceedings, researchers Andreas Happe and Jürgen Cito [3] will present a compelling exploration of the integration of Large Language Models (LLMs) into the realm of PT. LLMs, such as ChatGPT, GPT3.5, and AutoGPT, have gained significant popularity recently due to their remarkable ability to predict missing data and generate human-like text. As a result of these pattern-recognition abilities, which are learned through extensive training, the authors recognized the potential for leveraging LLMs to identify vulnerabilities, execute custom exploits, and even acting as virtual sparring partners to their human counterparts. This integration could provide guidance to not only “empower existing human security testers,” but could also “counteract the lack of sufficiently educated security professionals,” addressing a current critical shortage of skilled experts in the field. [3]

### Summary

With the aim of determining to what extent security testing can be automated through LLMs, the authors framed their research question around the deployment of these models as virtual sparring partners for security professionals. To provide a structured framework for their investigation, they turned to MITRE ATT&CK, a comprehensive repository of knowledge concerning threat actors in the cybersecurity domain. Their goal was to produce a proficient sparring partner, capable of covering a diverse array of tactics, techniques, and procedures (TTP) summarized within ATT&CK.

To comprehensively explore their hypothesis, the authors led a series of experiments, where they conducted demonstrations with both high- and low-levels of guidance. These demonstrations vary in detail and specificity, with high-level addressing general PT aspects and low-level dealing with more detailed, practical actions. For their high-level demonstration, they employed LLMs to assist the planning phase of a PT. This involved tasking the LLM with designing the test itself, including determining TTPs and identifying potential vulnerabilities. [3] They then explored low-level guidance, during which they engaged the LLM to assist in the execution phase of the PT. As it is assumed that pentesters have completed their high-level analysis by the time they begin a low-level analysis, this stage is often in a step-by-step format and includes activities such as identifying systems, targeting specific vulnerabilities, executing custom commands and exploits, and providing information on how to escalate privileges.

### Methodologies

In pursuit of answering their research question and exploring their hypothesis further, the authors aimed to demonstrate the extent and effectiveness of deploying LLMs as virtual sparring partners. In order for the LLMs to meet the authors expectation of a success, not only must their models produce valid and “suitable tactics and corresponding techniques.” [3] To test the practicality of using LLMs as sparring partners, the authors built upon the framework established in their research question with carefully designed experiments that encompassed both levels of guidance. Their approach ranged from broad and theoretical to highly specific and practical, which allowed them to assess the capacity and applicability of these models.

While the traditional approach to leveraging LLMs in PT requires human testers to manually initiate conversations using prompts, the authors sought to automate this process by using pre-trained Autonomous AI Agents: AutoGPT and AgentGPT. Not only do these agents increase productivity, but the incorporation of "external knowledge and automated feedback” can mitigate the occurrence of fact inventing, known as hallucinations. [3] Each tool can operate independently, eliminating the need for constant human intervention. This is accomplished by automatically breaking down predefined tasks into smaller, specialized subtasks through the use of “self-prompts.” [4] However, despite their simulation similarities, AutoGPT is described as having more decision making capabilities than AgentGPT, while AgentGPT offers a more user-friendly experience that welcomes a wider range of users. [5] Regardless, as both AutoGPT and AgentGPT can successfully accomplish an assigned objective from a single directive, they are valid options for PT.

In the High-Level experiments, the authors focused on the LLMs' potential to provide strategic guidance for both a general and specific target using both autonomous agents. For the general scenario, they provided AgentGPT with the task of “becoming a domain admin in an Active Directory,” [3] while, for the specific target, they tasked AutoGPT with creating a PT plan. These experiments were considered successful as both AI agents provided responses which were “realistic, and feasible, and would give a penetration tester good feedback about potential attack vectors.” [3] However, it is important to note that while AutoGPTs functionality also enabled it to crawl the target’s website, it declined to perform certain actions, citing ethical concerns.

In contrast, the low-level guidance experiments focused on providing step-by-step guidance, offering detailed actions such as identifying and exploiting system-specific vulnerabilities, executing custom commands and exploits, and providing insights on privilege escalation. At this stage, it was assumed that pentesters had already completed their high-level analysis, obtained some basic level of access to the system, and simply required guidance to escalate to root. Therefore, the goal of this experiment was to achieve privilege escalation and gain root access on a deliberately vulnerable Linux Virtual Machine (VM). The authors used python to set up a connection between GPT3.5 and the vulnerable VM and asked the LLM to analyze the VM's state, generate commands or actions, and potentially control or influence the VM's behavior. The script operated in an infinite loop, instructing GPT3.5 to suggest Linux shell commands, execute them over SSH on the vulnerable VM, analyze the command and its output, identify potential security vulnerabilities, and finally provide steps on how to exploit them. The results showed that GPT3.5 successfully obtained root privileges, identified and exploited security vulnerabilities, and retrieved essential system files for privilege escalation.

### Main Findings

During the experiments, the researchers found that the LLM displayed signs of understanding causal relationships and exhibited a degree of logical thinking in its suggestions for PT tasks. These suggestions followed logical patterns, even when specific information about the target system's configuration or vulnerabilities were not provided. The authors highlighted that these suggestions, while “eerie”, were primarily generated “based upon pattern-matching and not on a deeper understanding” of the subject matter. [3]

The authors also found that, on a small scale, the performance of LLMs appeared unstable and inconsistent, and often produced a large variation in generated commands and identified vulnerabilities. During individual and short runs, the LLM would become too fixated and overly focused on a specific detail and lose sight of the broader picture, similar to “going down a rabbit hole.” [3] While extending or combing results from multiple runs led to more consistent outcomes, LLMs were deemed less predictable and consistent compared to traditional enumeration tools like linpeas.sh in their current state.

LLMs were also found to be limited by their ethical filters, which prevent the AI from generating responses or taking actions that could engage in unethical behaviors. This was shown during the experiments when AutoGPT refused to execute additional network scans or phishing attempts. The authors found that many of these restrictions could be bypassed by running the LLM locally or by using prompt engineering to test slight prompt variations and reduce triggering the ethical filters. The simplicity of engineering prompts was shown when the authors requested “verification commands for vulnerabilities” instead of “exploits for vulnerabilities” and when they instructed the AI not to “ask questions or provide judgments.” [3] While these techniques prove effective in reducing ethical denials, they also raise concerns about potential misuse. Due to the ease and accessibility of LLMs, they can be employed by both legitimate security professionals and malicious actors.

While the experiments with LLMs have showcased their potential in providing valuable PT guidance, there remains a pressing need for further refinement in their application. The findings indicate that LLMs, although proficient at pattern recognition and generating suggestions, still rely heavily on data-driven responses rather than true comprehension of security systems. Addressing the challenges of occasional hallucinations and variability in single runs, especially when overly focused on specific aspects, is crucial to ensure their reliability. However, the urgency to incorporate AI in PT is crucial. As the field faces a critical shortage of skilled security professionals, it becomes increasingly vital that the relationship between pentesters and AI is strengthened. As the cybersecurity landscape evolves, preparing for AI-driven attacks becomes not only a necessity but also an opportunity for the industry to stay ahead in the ongoing battle against emerging threats.

### Relevance to Your Course Content

This paper aligns with my course content by exploring the integration of a familiar AI model, specifically LLMs, into the field PT. As the LLMs discussed in this paper are among the most recognizable AI models, they provide an ideal starting point for introducing the central theme of my course. Their familiarity offers a comfortable and approachable introduction to ‘Harnessing Artificial Intelligence (AI) for Penetration Testing.’ Furthermore, this report not only engages in theoretical discussions, but also delves into the practical application of popular prompt-response techniques within PT. Through tangible examples, it illustrates how AI can enhance various facets of PT, offering both a relevant and captivating perspective to witness firsthand the transformative potential of AI from the outset.

Additionally, the paper introduces important ethical dilemmas that arise when AI is used as a tool in PT. Ethical considerations hold great significance in the cybersecurity domain, and addressing these issues early on is crucial. The report explores the effectiveness of prompt engineering, raising questions of its acceptability and ethical boundaries. It also addresses the accessibility of these powerful tools to both security professionals and malicious actors, prompting considerations about the distinctions between their respective ethical codes. Exploring and understanding these ethical complexities is a vital step to exploring PT.

## Review 2

### Introduction

The report by Ghanem and Chen, focuses on the initial step of PT, known as Gathering Information, and how the integration of AI, particularly RL, can revolutionize this critical phase. RL has quickly become one of the most important PT advancements, resulting from the recent integration of AI and cybersecurity. This transformative approach to ML enables systems to learn through experiences from interactions with their environments. The incorporation of RL into automated PT techniques not only increases productivity, but also limits “recurrent human errors due to tiredness, omission, and pressure.” [1] However, existing automation systems have limitations in scope and optimization that result in their inability to comprehensively address all potential threats while efficiently managing resources. Recognizing these challenges, Ghanem and Chen's research paper sets forth to employ ML techniques in the development of an Intelligent Automated Penetration Testing System (IAPTS) that will be “capable of imitating human PT experts in performing an intelligent and automated pen test.” [1]

### Summary

Ghanem and Chen's research delves into the complexities of PT, an area that humans themselves often find challenging. They emphasize that blind automation, which entails complete automation without any human intervention, is impractical, particularly during the initial phases of PT. Since these early phases often yield incomplete results, they result in uncertainty and continuous revisitation of tasks and changes in approach. However, the authors suggest that by using RL to automate these phases intelligently, automation can more closely resemble a human expert’s decision-making process.

The challenges associated with automation in PT are not new, as autonomous systems are regularly used in the industry. However, these current systems often require substantial hands-on guidance, extensive time, and resources, limiting their applicability to smaller networks. Especially considering “PT should be repeated and performed on a regular basis to ensure continuous security,” [1] Ghanem and Chen's work suggests that automation holds the key to significantly improve various aspects of PT. Automation would not only reduce the cost of manual, repetitive, and methodical testing but also make PT more efficient and targeted, alleviating the strain on tested assets. This streamlining and automation of repetitive tasks reduces testing time, fostering adaptability and facilitates the exploration of innovative and unconventional techniques.

The authors advocate for the use of RL in PT, noting that RL aligns well with the goal-directed learning and decision-making processes required in the PT context. Unlike analyst-driven solutions with predetermined rules, RL learns through the consequences of its interactions, focusing on long-term goals rather than short-term fixes. This emphasis on RL represents a crucial step in addressing the challenges posed by PT automation and is converted into a formal computational model known as a Partially Observed Markov Decision Process (POMDP).

### Methodologies

The methodologies employed in Ghanem and Chen's research revolve around the innovative application of RL within the framework of POMDP. This approach seeks to address real-world PT challenges by translating them into a formal mathematical model. The POMDP model encompasses essential elements such as state observations, actions, rewards, and transition dynamics. Within this framework, an RL agent learns to make decisions based on its observations, aiming to maximize cumulative rewards. The strategies acquired by the RL agent that return the largest reward value are then stored in memory for similar cases in the future, thus enabling it to tackle complex PT problems systematically and autonomously.

One key methodology employed is the Policy Search method, which involves determining optimal sequences of actions, known as policies, that lead to the highest cumulative rewards over time within the POMDP framework. To find these optimal strategies, the researchers combined the POMDP model with specific algorithms. Notable among these is the “randomized point-based value iteration” algorithm known as PERSEUS, designed for solving POMDPs in scenarios with limited resources, making it suitable for addressing PT challenges in large networks. [1] Additionally, the PEGASUS algorithm plays a crucial role by estimating and seeking high-value sets of policies that determine the actions taken by the RL agent, simplifying the search for optimal decision-making strategies.

Initially, the learning process for their proposed system, IAPTS, relies on human input, as experts teach the system and provide knowledge. However, over time, the system evolves, with the potential to develop autonomous learning modules that reduce the need for manual interventions. This evolution aligns with the various operational modes of IAPTS ranging from fully autonomous (Level 4) to learning mode (Level 1), where a human expert performs PT while the system observes and learns.

The primary goal of testing IAPTS was not only to evaluate its capabilities but also to demonstrate the suitability and effectiveness of applying RL to PT. The researchers conducted two main types of tests: Simple Simulation and Experience Replay, both within controlled environments. In the Simple Simulation, they set up a simulated network consisting of seven machines (M0 to M6) to mimic real-world PT scenarios. This allowed them to gain insights into how IAPTS would perform under various conditions, measure its performance metrics, assessed execution times, and identify potential weaknesses. In the Experience Replay tests, the researchers simulated scenarios in which the same network underwent updates and upgrades. These tests aimed to evaluate how well IAPTS learned and adapted to changes in the network, further confirming its potential for enhancing penetration testing processes.

### Main Findings

The main findings of Ghanem and Chen's research paper provide valuable insights into the field of PT. In their Experience Replay tests, they discovered that the system successfully learned and stored knowledge from previous tests, with policies being effectively reused in most instances. This highlights the system's adaptability and ability to learn from past experiences, a crucial feature proving highly beneficial in the PT field. When compared to traditional manual methods, which rely on human expertise, and the blind automation approach, where tasks are automated but lack intelligent decision-making, IAPTS, significantly reduces the time required for testing while outperforming both approaches in terms of efficiency and effectiveness. This not only saves time and resources but also generates alternative attack strategies that humans may overlook.

The RL-generated attack policies also proved to be highly relevant and accurate, especially when targeting the most secure machine in the network. These policies were deemed plausible and realistic, mirroring how actual attackers might approach and execute an attack on the target system. Additionally, IAPTS was intentionally designed with flexibility in mind, permitting the seamless incorporation of new features and functionalities in the future. This forward- thinking design ensures IAPTS remains a versatile and evolving tool in the field of penetration testing, continually enhancing its capabilities and performance.

### Relevance to Your Course Content

The research paper by Ghanem and Chen offers a comprehensive overview of PT, including its purpose, advantages, disadvantages, and the intricate challenges involved, especially as it pertains to the first step in PT - Gathering Information. They emphasize the extensive data collection and assessment required during this phase and set the stage for discussions on automation and AI as potential solutions to address the growing complexity of threats. Through practical simulations, the authors demonstrate how these solutions can significantly reduce human effort, enhance accuracy, improve adaptability, and expedite tasks, ultimately proving that automation can make the PT process more efficient.

The authors introduce advanced techniques such as Reinforcement Learning (RL) and Partially Observable Markov Decision Processes (POMDPs) within the context of PT. RL, as a subset of artificial intelligence (AI), is particularly relevant in the context of automating PT phases. By incorporating RL and POMDPs, the paper demonstrates the direction of future research, showing that automated PT is an evolving field with ongoing developments. The paper not only demonstrates what is currently achievable but also serves as an eye-opener to the vast possibilities within the field, expanding the perspective of potential advancements in automated PT.

Incorporating this paper into my seminar course can provide a well-rounded understanding of PT, especially in the initial Gathering Information phase, and explain how advanced AI-driven techniques like RL are transforming the field. It serves as an excellent resource to introduce the challenges of PT, automation as a solution, and the application of AI in enhancing cybersecurity practices.

## Review 3

### Introduction

The research paper “Penetration Testing Procedure using Machine Learning” focuses on the second phase of PT - the scanning phase, with a particular focus on assessing the effectiveness of GyoiThon. GyoiThon is a penetration testing tool integrated with ML capabilities, specifically leveraging the Naïve Bayes algorithm. This integration represents a significant advancement within the field of cybersecurity, not only enhancing the speed and efficiency of vulnerability detection, but also introducing the potential for more precise identification of security weaknesses. By leveraging other PT tools to enhance its capabilities, GyoiThon extends its utility beyond traditional methods. It automates the process of gathering data from target URLs, thus streamlining the scanning phase while reducing the time and effort required by penetration testers.

### Summary

In this study, the researchers set out to address the fundamental research question: 'How effective is the GyoiThon tool in detecting vulnerabilities?' [6] The hypothesis guiding this exploration speculates that PT tools integrating ML algorithms will exhibit greater effectiveness in searching for and identifying vulnerabilities compared to their non-ML counterparts. To highlight this fact, the paper briefly examines common ML-based PT methods used in the field, including tools known for simulating real-world attacks, detecting vulnerabilities, and addressing security weaknesses. Their analysis provided valuable context and benchmarks for evaluating GyoiThon's performance and offers support for their hypothesis regarding the efficacy and capabilities of using ML in penetration testing.

This exploratory study places particular emphasis on comparing GyoiThon's default mode with its Machine Learning mode, executing them within controlled environments. The research delves into the capabilities of GyoiThon, showcasing its ability to enhance the scanning phase of PT by automating data acquisition from target URLs. By exploring the effectiveness of GyoiThon, the authors assess its efficiency in detecting known vulnerabilities, identifying software components, discovering configuration weaknesses, highlighting authentication issues, and pinpointing general web application vulnerabilities. [6] These capabilities emphasize GyoiThon's pivotal role in the scanning phase of PT, highlighting the demand for advanced tools and techniques to navigate the complex landscape of cybersecurity.

### Methodologies

The methodology employed in this report is particularly significant as it delves into a novel area of interest within cybersecurity. It's worth noting that this comprehensive study of GyoiThon represents a unique endeavor, with the only prior study conducted by the tool's developer. As such, the researchers had the distinct advantage of operating within a flexible framework that lacks predefined steps, which enabled them to create new procedures to address their research question. [6]

As for execution, the researchers established an isolated testing environment using the Kali Linux operating system within VirtualBox. Within this controlled environment, GyoiThon was utilized to detect data exchange vulnerabilities in both unencrypted HTTP over Port 80 and encrypted HTTP over Port 443. The target websites were hosted on a server provided by OWASP and accessed via a locally hosted environment.

The procedure was conducted using two distinct modes: Default Mode and ML Mode. The Default Mode encompassed various steps, including parsing HTTP responses, identifying product/version information, assessing vulnerabilities using Common Vulnerabilities and Exposures (CVE) numbers, examining HTML and JavaScript comments, analyzing debug messages, and assessing login pages. [7] In contrast, the Machine Learning (ML) Mode incorporated all the steps from the Default Mode but added the utilization of the Naïve Bayes algorithm for product/version identification. [7] It's essential to note that the full scope of GyoiThon was not tested in this study, given that the tool offers a total of nine modes for PT. Researchers deliberately focused on the Default and Machine Learning modes to assess the effectiveness of the Naïve Bayes algorithm in the realm of penetration testing, aligning with their hypothesis.

### Main Findings

The analysis of the PT procedure conducted using GyoiThon revealed several significant insights. First, it was observed that Port 80, commonly associated with unencrypted HTTP data, exhibited a higher number of vulnerabilities in both the Default andML modes. This finding aligns with expectations, as Port 80's lack of encryption renders it less secure compared to HTTPS (Port 443). This absence of encryption makes Port 80 more susceptible to vulnerabilities and potential attacks, which was reflected in the test results. However, the difference in vulnerability frequencies was reduced when Machine Learning mode was used, indicating improved effectiveness. When ML mode was used, utilizing the Naïve Bayes algorithm, it outperformed the Default Mode in vulnerability detection by detecting three additional vulnerabilities in Port 80. Not only does this outcome highlight the potential of GyoiThon, it also supports the hypothesis that integrating machine learning into PT tools enhances their effectiveness in identifying vulnerabilities.

Despite these results, it is crucial to note that GyoiThon's depends on external vulnerability data sources, such as the National Vulnerability Database (NVD), for identification. This reliance is a limitiation of the tools capabilities as it may be unable to identify vulnerabilities that have not been previously recorded in the NVD. This potential blind spot highlights the importance of staying updated with emerging threats. As such, while GyoiThon showcases promise as a valuable penetration testing tool, the researchers identified areas for improvement. Future testing against real websites and an assessment of all nine modes could provide a more comprehensive understanding of its capabilities. These findings contribute to the ongoing development of AI-driven penetration testing tools and emphasize the need for continuous refinement to stay ahead of evolving cyber threats.

### Relevance to Your Course Content

This report extensively explores the application of ML in PT, using GyoiThon as a focal point. This alignment with the central theme of my course, which aims to provide a thorough understanding of AI techniques for cybersecurity, makes it a valuable resource. In particular, the article provides a sturdy foundation for the second phase of PT, scanning, which is the primary focus of my second module and an important aspect of cyber operations. Through real-world applications, the authors demonstrate how AI-driven tools, such as GyoiThon, can effectively identify vulnerabilities by scanning web pages. This practical case provides invaluable insights for pentesters and resonates with the primary course objective, emphasizing the harnessing of AI for PT.

Through a comparative analysis between default penetration testing methods and those augmented with AI, this article showcases the effectiveness of AI-driven approaches through direct evidence. This novel and practical study not only highlights the superiority of AI-enhanced techniques but also underscores the potential transformative power of AI within the cybersecurity domain. By offering compelling empirical evidence, it encourages further exploration into the integration of AI, positioning it as a fundamental fortification of cybersecurity strategies. This practical validation not only encourages exploration but also invites active engagement with AI-driven security solutions, recognizing them as a cornerstone of modern cybersecurity practices.

Lastly, the article delves into the discussion of common vulnerabilities found in web applications and the various tools used to detect them. This practical understanding of vulnerabilities and the AI-driven tools and techniques available for their detection and mitigation is essential for effectively navigating the complex cybersecurity landscape.

## Review 4

### Introduction

In the paper titled “Vulnerability Exploitation Using Reinforcement Learning,” the authors leverage modern PT techniques, specifically ML and RL, to automate one of the most critical phases in cybersecurity: exploitation. By prioritizing actions that maximize rewards, RL underscores the importance of developing tools that not only identify vulnerabilities but also utilize ML to efficiently exploit them. However, their focus extends from just automation and highlights the necessary evolution of PT in response to the complex field of cyber security.

The intelligent agent created in the report can be trained on a wide array of vulnerabilities and operating systems, offering a tailored and intelligent approach to exploitation. While traditional methods often involve resource-intensive, brute-force techniques that are time-consuming, this agent leverages Metasploit, a well-known PT tool with a wide range of payloads for various purposes. This not only accelerates the penetration testing process but also ensures a more targeted and efficient approach to identifying and exploiting vulnerabilities. What further sets this approach apart is the agents ability to archive successful exploits as states—combinations of operating systems and vulnerabilities—alongside their corresponding payloads with high success probabilities. The agent then intelligently leverages this payload repository, known as a Q-Table, to execute exploitation with precision and efficiency.

In essence, this paper marks a significant milestone in PT, demonstrating the potential of RL to leverage an award system continuously refine and enhance exploitation strategies using AI. It's provides a look into the future of PT, where customization, adaptability, and intelligence combine to not only identify vulnerabilities but to masterfully exploit them.

### Summary

In this report, the authors utilize ML to create an RL agent that makes decisions by interacting with an environment. The primary focus of this RL agent lies in the exploitation phase, the third and crucial step in PT. At its core, this RL agent departs from conventional PT tools by leverages the Metasploit framework, a well-known PT tool recognized for its extensive array of payload options, to target a wide range of vulnerabilities and operating systems.

Through an extensive training process, the agent interacts with a simulated environment, dynamically adapting its exploitation strategies by analyzing various factors, including the environment configuration. This adaptive approach is made possible by representing the environment as states, each defined by a unique combination of operating system and vulnerability. These states are then linked to payloads that have demonstrated a high likelihood of success and are stored in a Q-Table. Due to the variability in payload effectiveness based on these states, the authors reward successful attempts, which they define as "the establishment of a reverse shell session following payload execution." [2] Therefore, even in instances where the payload is not successful, the RL agent adjusts its decision-making based on the rewards it receives and learns to prioritize actions that result in positive rewards.

Once the RL agent is trained, it is deployed in a real-world scenario where it encounters target systems with specific operating systems and vulnerabilities. Metasploit serves as a valuable resource as the RL agent selects and utilizes payloads based on its learned strategies, facilitating effective delivery of exploits to compromised target systems. The extensive payload options offered by Metasploit enhance the agent's versatility during the exploitation process. This integration contributes to the authors primary goal of creating a versatile “general agent that is capable of exploiting any/general task and making the appropriate decision.” [2]

This combination of ML, RL, and established PT tools represents a significant advancement in the merging of AI and cybersecurity. Through the incorporation of RL algorithms and their integration with established tools like Metasploit, this report demonstrates an evolution of PT. This innovative approach showcases the potential of AI-driven agents to optimize and streamline exploitation tasks, ultimately benefiting cybersecurity professionals in identifying and addressing vulnerabilities in a more efficient and effective manner.

### Methodologies

The methodologies employed in this study consist of two important phases, the Training Phase and the Exploitation Phase. During the Training Phase, an intelligent agent is developed through the application of RL techniques, using a guess-and-reward system. This phase involves the agent navigating a simulated environment, in which it uses an “epsilon greedy strategy” to make informed decisions by balancing exploration (delivering a randomly selected payload) and exploitation (selecting a specific payload that will yield the highest expected reward based on its learning so far). The agent then receives rewards based on the success or failure of a particular payload, from which it builds a valuable repository of previous exploits and their results. The training phase is then repeated for a certain number of iterations, with a gradual decrease of exploration.

To motivate its decision-making, a point-based reward system is employed, offering substantial rewards for success and imposing penalties for failures. These rewards are maximized by leveraging the Q-learning algorithm, to “determine the best series of actions to take based on the agent’s current state.” [8] This approach often results in the agent executing calculated and cautious actions to minimize risks.

Gradually decreasing epsilon encourages the agent to focus on exploiting the best-known actions. The training phase is computationally intensive and spans multiple trials, with the agent executing 500 attempts to exploit vulnerabilities over an average of 2.5 hours, a deliberate choice made during experimentation.

In the experiment, the learning phase honed its exploitation skills across seven trials. Over the course of these trials, the agent spent an average of 2.5 hours executing 500 attempts to exploit vulnerabilities. During this phase, the agent's primary focus was on continuous learning and strategy refinement. It actively experimented with different actions, assessing their success or failure, and served to provide insight into valuable tuning parameters from controlling the importance of new versus old information, long-term versus short-term rewards, to exploration vs exploitation. [8] An assessment of the agent's performance is then calculated to determine how effective it is at establishing a reverse shell. This process was computationally intensive, reflecting the agent's ability to actively learn and adapt its exploitation techniques, by making informed decisions.

In the exploitation phase, the RL agent takes advantage of its learned strategies, drawing insights from its repository, to effectively select payloads from the Metasploit framework. To simulate real-world scenarios, it was deployed on multiple vulnerable machines with a "remote code execution" vulnerability found in Apache CouchDB, specifically Version 3.1.0. [2] The agent's primary objective was to establish a reverse shell, which it achieved with remarkable efficiency by leveraging the payloads with the “highest rank in the Q-Table.” [2] Impressively, it accomplished this goal in an average of just 8.26 seconds across the tested systems. This remarkable performance indicates that the training phase prepared the agent well, proving its ability and effectively execute its learned strategies against real-world systems, confirming its readiness for practical cybersecurity tasks.

### Main Findings

The study's main findings underscore the remarkable effectiveness of the RL agent in automating exploitation tasks, particularly within the realm of PT. As the RL agent gains experience through training, it exhibits a gradual shift from exploration to exploitation, becoming more discerning in its actions. Initially, it explores new actions to gather information, but over time, it prioritizes actions it has found effective in achieving its goals. This transition, combined with the selection of optimal parameters, consistently resulted in an average success rate of 83.64% and an average exploit time of 8.26 seconds. These notable statistics highlight the potential of the RL approach to significantly reduce the time and resources required for PT, presenting a novel and cost-effective solution to the challenges of vulnerability exploitation.

In contrast to traditional exhaustive testing methods, which often follow rigid approaches, the RL agent's adaptability and capacity for fine-tuning its strategies prove advantageous. By focusing on maximizing overall rewards and balancing learning and randomness, the RL approach proves more efficient and effective in verifying exploitable vulnerabilities. In summary, the main findings of this study emphasize the RL agent's aptitude for automating exploitation tasks, its proficiency in achieving PT objectives, and its potential to revolutionize vulnerability assessment practices. The agent's adaptability and parameter customization position it as a valuable and promising tool within the cybersecurity landscape.

### Relevance to Your Course Content

The study's exploration of modern techniques in cybersecurity, particularly the innovative use of RL algorithms for vulnerability exploitation, emphasizes the dynamic nature of the field. It serves as a compelling example of how RL can offer unique approaches to solving complex security challenges, showcasing its versatility and broad applicability. This adaptability extends to parameter tuning, where the impact of variables like learning rate and exploration rate on RL algorithm performance is studied, emphasizing the importance of fine-tuning and optimizing security tools to keep pace with evolving threats. Additionally, since RL consistently selects the most effective actions to maximize rewards, it directly addresses a critical aspect of Penetration Testing, particularly in Step 3 - Exploitation. By prioritizing the actions that yield the highest rewards, RL showcases the importance of developing similar tools that not only identify vulnerabilities but utilize ML to efficiently exploit them. Overall, this research broadens perspectives on the possibilities within the field of cybersecurity and highlights its crucial role in staying current with the dynamic landscape of digital threats.

## Review 5

### Introduction

In the report, “Automated Post-Breach Penetration Testing through Reinforcement Learning,” the authors introduce the concept of using RL, a subset of ML, to automate the post-breach phases of PT. Theses phases occurs after the initial breach of a system and focus on privilege escalation, maintaining persistence, and further exploration. [9] This approach aims to automate and enhance the capabilities of an AI agent, allowing it to navigate and interact with diverse network environments effectively. The agent is trained through interactions with various network environments and prioritizes the balance of exploration and exploitation. The importance of this study is emphasized as there is still a substantial lack of testing for automation in the post-exploitation phase. [10] This lack of research is particularly dangerous as current practices are rapidly growing in complexity and resource consumption. In an attempt to mitigate these challenges, researchers explore the application of AI techniques, specifically the Deep Reinforcement Learning (DRL) subset, Deep Q-learning (DQ). The hope that by leveraging neural networks to directly map input states to action-Q-value pairs, [8] their agent will excel in navigating complex environments without requiring a detailed model of the environment.

### Summary

This report highlights the current limitations in the field, pointing out that even with the use of automated tools, current PT practices remain complex and resource-intensive. In response to the limitations of the popular RL algorithm, Q-learning, and driven by recent advancements in deep Q-learning (DQ) algorithms, the authors made a deliberate choice to adopt DQ as their training model. Q-learning struggles when tasked with handling intricate systems or environments as it becomes computationally expensive to maintain the Q-table [9]. To overcome these challenges, the authors explore the implementation of alternative approaches such as DRL.

DRL has emerged to address these challenges and offer more solutions for RL in larger and more intricate environments. Instead of relying on a detailed model of the environment, DQ learns directly from interactions with the environment by utilizing neural networks. These networks take the current state as input, produce estimated reward values known as Q-values for all possible actions in that state, and associate each action with a unique Q-value, where higher values indicate more favorable actions. In other words, DQ distinguishes itself from traditional Q-learning by replacing the Q-table with a neural network that directly produces recommendations for actions based on the current state. [8] The agent then employs a decision-making process that involves comparing these Q-values and selects the action linked to the largest reward.

These rewards are used to adjust and refine the Q-values in both traditional Q-learning and DQ. Just as it does in the traditional method, he Q-values become more accurate over time and enhance the agent's ability to make decisions.

Overall, while the paper primarily presents a conceptual framework, it lays the groundwork for practical applications of AI in post-exploitation cybersecurity. It introduces key concepts such as Q-value estimation, exploration-exploitation balance, and the importance of realistic training environments. The paper's focus on future research and development suggests its potential to shape the future of AI-driven penetration testing.

### Methodologies

For their research, the authors propose an architecture which involves employing a Deep Q Network with TF Agents, constructed on the TensorFlow library as the fundamental framework for training the RL agent. TF Agents is a specialized, modular software that leverages the capabilities of the TensorFlow framework. [11] By building upon TensorFlow and incorporating TF Agents, they ensure a solid foundation for training the agent, harnessing the power and flexibility of these frameworks to achieve their research objectives effectively.

To create a realistic PT environment, the authors plan to deploy the agent in virtualized Linux and Windows servers. These environments were crafted to simulate authentic cybersecurity scenarios, effectively mirroring computer networks specifically tailored for cybersecurity training and competitions. Emulating target data, these environments included critical elements commonly found in these networks such as password files, shadow files, and system configurations. The agent's actions are confined to a predefined list of terminal commands, with the specifics of these commands being adapted based on the agent's observations within the environment. The resulting performance will then be gauged based on its adeptness at exploration and exploitation within these environments, with scores serving as rewards critical for reinforcement. The balance between exploiting actions that appear promising based on its current knowledge (Q-values) and exploring new actions to discover potentially better strategies, are carefully managed to optimize the agent's learning process and overall performance.

To establish a performance baseline for their research, the authors created a Python script designed to locate files within simulated environments. This script servs as a reference point to assess the AI agent's performance, enabling a direct comparison between the agent and the script's capabilities. This python script is tested across the servers to determine its effectiveness in locating files, particularly focusing on configurations and log files. To assess the trained agent's effectiveness, a comparison will be drawn between the script’s performance and that of an RL agent “created using randomized policy,” [10] ultimately providing insights into the efficiency of the trained agent's policy.

As such, the methodologies discussed in this report offer a comprehensive plan for training and evaluating a RL agent for automated post-breach penetration testing. The authors encompass the choice of Deep Q-Learning, the design of a specialized architecture, training in simulated environments, definition of agent actions and rewards, and a rigorous preliminary evaluation involving baseline testing and performance comparison. These methodologies are integral to the successful development and assessment of the proposed automated penetration testing approach.

### Main Findings

One of the primary findings of the report is the recognition of the applicability of RL, specifically DQ, in the domain of PT. The report emphasizes that RL offers a promising avenue for automating the post-breach phases of PT, a field where modern practices are lacking research, and traditional practices are quickly becoming obsolete. While this report focuses on conceptual feasibility of this incorporation, it directly addresses the beginning steps required to train an RL agent to perform tasks in a compromised network environment. This discovery is significant as it not only validates the role of ML in enhancing cybersecurity measures but also opens doors to the development of more efficient and effective penetration testing methodologies.

Through the analysis of automating the post-exploitation phase, the authors sets the stage for ongoing and future investigations. The report communicates the researchers' intention to implement the proposed approach, train the RL agent, and expand the model's applicability to a broader spectrum of network environments. While this report primarily focuses on conceptual feasibility, it is also practical and action-oriented, with the aim of making meaningful contributions to the cybersecurity field by advancing the automation of PT practices.

### Relevance to Your Course Content

This report highlights the critical application of AI, specifically RL, in automating and optimizing the post-breach stages of PT. Notably, it offers practical examples showcasing how AI can be effectively deployed during this critical phase and advances uncharted territory in doing so. This novel discussion addresses the imperative need for cybersecurity professionals to remain current and adaptive in the face of evolving threats and encourages innovation.

Additionally, this study provides insights into the advanced technologies that power AI- driven PT, notably highlighting the significance of DQ and its role in improving traditional Q-learning methods. As ML and deep learning are at the core of AI's capabilities, delving into these technological intricacies establishes a robust foundation to comprehend how AI models are constructed, trained, and effectively deployed. This comprehensive understanding is essential as it empowers individuals to make informed decisions, adapt AI tools to specific cybersecurity challenges, and innovate within the field, ultimately contributing to the ongoing evolution and effectiveness of AI-driven PT practices.

# Incorporation of Findings into the Course

* Discussion on how the findings from the literature review inform your course development
* Preliminary Ideas for Course Content based on the Literature Review

# Table of Contents for the Course

## Introduction

Through a structured curriculum, students will delve into AI-driven penetration testing tools, machine learning, and advanced deep learning techniques.

Each module will provide hands-on demonstrations, illustrating how AI enhances offensive strategies, from vulnerability identification to custom exploit creation. Real-world scenarios will showcase AI's transformative power in the world of cybersecurity.

## Module 1: Introduction to AI and ML in Penetration Testing

### Module Overview and Brief Description

### Submodule 1.1: Introduction to AI and ML in Penetration Testing

### Submodule 1.2: Gathering Information and Reconnaissance with AI

## Module 2: Machine Learning for Vulnerability Assessment

### Module Overview and Brief Description

### Submodule 2.1: Scanning and Vulnerability Assessment with ML

### Submodule 2.2: Exploiting and AI-Enhanced Techniques

## Module 3: Post-Exploitation AI and ML Techniques

### Module Overview and Brief Description

### Submodule 3.1: Maintaining Connection, Covering Tracks, and Reporting

### Submodule 3.2: AI-Enhanced Post-Exploitation and Privilege Escalation

## Module 4: Deep Learning and Advanced Techniques

### Module Overview and Brief Description

### Submodule 4.1: Deep Learning and Advanced Techniques

### Submodule 4.2: Review/Conclusion

## Conclusion

## References and Further Reading

# Key Components of the Newly Developed Course

* Learning Objectives
* Target Audience
* Assessment Strategies
* Supplementary Materials

# Conclusion

* Summary of Key Insights
* Potential Impact of the Course

# References

* [4]
  + <https://autogpt.net/auto-gpt-vs-chatgpt-how-do-they-differ-and-everything-you-need-to-know/>
* [5]
  + <https://openaimaster.com/auto-gpt-vs-agentgpt/>
* [1]
  + Reinforcement Learning for Intelligent Penetration Testing
  + Mohamed C. , Thomas M. Chen
  + <https://ieeexplore.ieee.org/document/8611595>
* [7]
  + <https://github.com/gyoisamurai/GyoiThon>
* [2]
  + <https://ieeexplore.ieee.org/document/10185700>
* [8]
  + <https://www.baeldung.com/cs/q-learning-vs-deep-q-learning-vs-deep-q-network>
* [11]

https://www.tensorflow.org/learn

Notes: switch some AI to intelligent agent to break monotony

Cybersecurity or cyber security, pick one