Cyber.AI : AI powered Cyber Security tool

Sparsh Verma
Student Number 220996233
Project Supervisor: Syed Rafee
MSc. Artificial Intelligence, QMUL

***Abstract*—Cyber.AI represents a novel convergence of artificial intelligence and cybersecurity, providing an innovative method for automated penetration testing. Utilising deep learning, this tool autonomously detects and leverages system vulnerabilities. While conventional penetration testing is invaluable, it is often limited by manual processes and human oversight. Cyber.AI mitigates these challenges, offering a swift, uniform, and evolving solution for vulnerability detection. This study explores the tool's design, its synergy with established penetration testing platforms, and its performance across varied environments. Additionally, the ethical ramifications and potential consequences of such automation in the realm of cybersecurity are critically assessed. Initial results indicate that, though Cyber.AI presents promising advancements in automation, ensuring its ethical and responsible use is crucial to maintaining robust cybersecurity protocols.**

Keywords—CyberAi, Artificial intelligence, Cyber Security, penetration testing.

I. INTRODUCTION (*HEADING 1*)

Cyber.AI embodies the seamless integration of cutting-edge artificial intelligence and the intricate domain of cybersecurity. The underlying codebase serves as the foundation for this tool's ability to autonomously identify and exploit system vulnerabilities, delivering a paradigm shift in the approach to penetration testing. At the core, Cyber.AI leverages the power of deep learning, utilising meticulously designed neural network architectures that adapt and refine their strategies based on the input data and feedback from penetration attempts. As a result, the tool evolves, becoming increasingly proficient in identifying vulnerabilities and executing exploits over time. This codebase encompasses a range of functionalities, from data preprocessing and neural network training to interfacing with renowned penetration testing platforms. Every module and function has been crafted to ensure optimal performance, accuracy, and scalability, catering to both novice users and seasoned cybersecurity professionals. Users delving into this code will find a well-structured and documented repository, designed for clarity, expendability, and efficient debugging. Whether you aim to understand the nuances of automated penetration testing, adapt the tool for a specific use-case, or contribute to its future iterations, this introduction aims to pave the way for a productive and enlightening exploration.

II. RELATED WORKS

The domain of automated penetration testing, which CyberAi occupies, has been a topic of research and development for many years. Numerous tools, frameworks, and research papers have emerged in this space. Below are some related works to CyberAi, each contributing to the advancement of automated cybersecurity measures:

- Metasploit: Perhaps one of the most renowned frameworks for penetration testing, Metasploit provides a platform for developing, testing, and executing exploit code. While it doesn't use deep learning like CyberAi, it is a pivotal tool in the cybersecurity arsenal and has modules that can be automated.[1]
- AutoSploit: As the name suggests, AutoSploit attempts to automate the exploitation process by gathering hosts that are vulnerable and then running Metasploit modules against them.[2]
- Nexpose: A product by Rapid7 (the same company behind Metasploit), Nexpose is an integrated vulnerability management tool. It scans networks to identify vulnerable devices and then integrates with Metasploit to coordinate and manage penetration tests against those vulnerabilities.[5]
- OpenVAS: The Open Vulnerability Assessment System (OpenVAS) is a free software that functions as a full-service vulnerability scanner, providing a framework to search for known vulnerabilities in scanned systems.[4]
- AI-driven cybersecurity research: Many researchers are now investigating the fusion of AI and cybersecurity. While not all of these endeavours lead to the creation of tools like CyberAi, the underlying research provides valuable insights and foundational knowledge that tools can later leverage. Topics might include neural networks for anomaly detection, machine learning for phishing detection, or reinforcement learning for penetration testing.[8]
- ZAP (Zed Attack Proxy): Developed by OWASP (Open Web Application Security Project), ZAP is a free security tool aimed at finding vulnerabilities in web applications. It's designed for both automated scanning and manual testing, making it a versatile tool in a pentester's toolkit.[9]
- Athena: A project aimed at automating penetration tests using machine learning. Athena learns from previous penetration tests and tries to adapt its strategy based on past successes and failures.[3]
- AVDS (Automated Vulnerability Detection System): This focuses on the automation of vulnerability detection. By using a range of scripts and tools, AVDS scans systems and networks for known vulnerabilities.[6]
- Research on adversarial machine learning: While this is broader than penetration testing, adversarial machine learning research is about understanding how machine learning models can be attacked and defended. Insights from this domain might influence how tools like CyberAi are developed or countered in the future.
- Hydra: An influential tool primarily used for brute-forcing login credentials. Hydra supports numerous protocols to attack, and while it doesn't incorporate

machine learning, its automation capabilities have made it a staple in penetration testing.[6]

- Burp Suite: An integrated platform for performing security testing on web applications, Burp Suite offers a range of tools, from an advanced proxy to automate the process of identifying and exploiting application vulnerabilities. Its automation capabilities highlight the move towards more hands-free cybersecurity measures.[7]
- MARA Framework: A mobile penetration testing toolkit, MARA is utilised to assist in the identification of threats in the mobile ecosystem. While its primary focus is on Android, its automated tools and scripts position it in the broader context of automated security testing.[11]
- Research on Genetic Algorithms in Cybersecurity: Genetic algorithms, inspired by the process of natural selection, have found their way into cybersecurity. Some projects explore the feasibility of using these algorithms to evolve new exploits or to identify vulnerabilities in software. This represents another facet of machine learning and AI-driven cybersecurity measures.[15]
- Machine Learning and Intrusion Detection Systems (IDS): There's a considerable body of research dedicated to integrating machine learning into IDS. Deep learning networks, in particular, have shown potential in detecting new, previously unidentified threats based on patterns.[10]
- Ghidra: Introduced by the U.S. National Security Agency's Research Directorate, Ghidra is an open-source software reverse engineering tool. While its core function isn't penetration testing, the insights gained from reverse engineering can inform and enhance automated penetration testing tools.[12]
- Binary Exploitation with AI: Recent research has delved into the possibility of using AI to exploit binary vulnerabilities. By analysing compiled binaries, some researchers are attempting to automate the process of finding and exploiting these vulnerabilities, bringing a new dimension to the landscape.
- Pentest Automation Frameworks: Tools like Faraday offer a collaborative platform for penetration test and vulnerability management. By integrating various tools under one roof, they reflect the industry's move towards more coordinated and automated cybersecurity measures.[17]

CyberAi, amidst this vast landscape of tools and research, underscores the evolving nature of cybersecurity. As threats become more sophisticated, so too must the tools that counter them. The integration of AI into this domain, as evidenced by CyberAi and other related works, is a testament to the industry's forward-thinking approach. While CyberAi is unique in its approach, especially with its integration of deep learning, it stands alongside these related works, each contributing towards a more secure digital landscape.

III. ARCHITECHTURE

The architecture of CyberAi's codebase provides a multifaceted understanding of how modern AI-driven cybersecurity tools operate. For those passionate about cybersecurity, this repository is more than just lines of code; it's a journey into the future of threat mitigation and system protection.

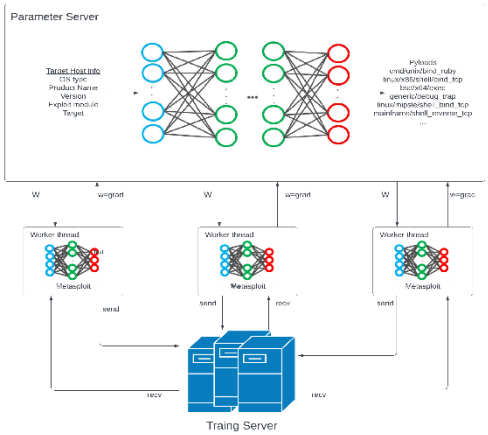
- Several key components form the bedrock of CyberAi:
- Data Integration: Harnessing data from various sources, this section is integral for feeding the machine learning algorithms with relevant information. The preprocessing functions ensure that the data is in its optimal format, priming it for effective analysis.
- Neural Network Framework: Herein lies the heart of CyberAi 's intelligence. Crafted using state-of-the-art frameworks, this component encapsulates the deep learning models responsible for making real-time decisions during penetration tests.
- User Interface and Interaction: CyberAi is not just about backend sophistication. The tool boasts a user-friendly interface, ensuring seamless interaction for end-users. Functions dedicated to result CyberAi and report generation enhance the user experience, translating complex findings into actionable insights.
- Integration Modules: Recognising the importance of synergy in cybersecurity ecosystems, CyberAi is equipped with modules allowing integration with widely-used penetration testing platforms. This ensures that the tool can be conveniently embedded into existing workflows.
- Documentation and Community: Beyond the primary code, the repository places a strong emphasis on comprehensive documentation. This aids in troubleshooting, feature exploration, and tool enhancement. Furthermore, a burgeoning community around CyberAi fosters collaboration, ensuring the tool's continual evolution in response to the dynamic landscape of cybersecurity.

CyberAi, as a unique tool that leverages machine learning for penetration testing, integrates various components to create its distinctive architecture.

- User Interface (UI):
  - Command Line Interface (CLI): The primary interaction point for users, allowing them to initiate scans, view results, and adjust settings.
  - Configuration Files: These allow users to set predefined configurations, such as target IP ranges, specific vulnerabilities to test for, or other parameters that CyberAi should consider during its operations.
- Data Handling Modules:
  - Data Acquisition: Gathers relevant cybersecurity data, such as logs of previous penetration tests, vulnerabilities from databases, etc.
  - Data Preprocessing: Cleans and processes the acquired data to make it suitable for machine learning training. This includes normalising and structuring the data.
- Machine Learning (ML) Module:

- Training Data Set: Historical data that the model uses to learn from. This could include known vulnerabilities, past successful exploits, and other related information.
- Deep Learning Model: The core of CyberAi, where predictions about vulnerabilities and potential exploits are made. The model's structure might include several layers of neural networks, trained using the training data set.
- Training & Evaluation: Functions that guide the model through the learning process, adjust its parameters for better performance, and evaluate its accuracy and efficiency.

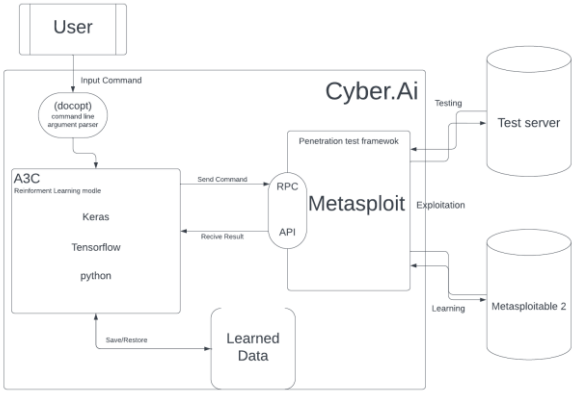
- Integration with Other Tools:
  - Metasploit Integration: One of the most significant features of CyberAi is its seamless integration with Metasploit, allowing it to use Metasploit's extensive database of exploits.
  - External Tool Connectivity: CyberAi can connect with other tools or platforms, broadening its capability to scan, detect, and exploit vulnerabilities.
- Result Presentation and Reporting:
  - Visualization Tools: Allows users to view graphical representations of scan results, vulnerabilities found, and other key metrics.
  - Report Generation: After a penetration test, CyberAi can generate detailed reports, offering insights into detected vulnerabilities, suggested countermeasures, and other relevant information.
- Continuous Learning & Feedback Loop:
  - Feedback System: Post-exploitation, feedback about the success or failure of exploits can be looped back into the system to improve future predictions and actions.
  - Model Retraining: Leveraging feedback, the deep learning model can be periodically retrained to adapt to new vulnerabilities, changes in the threat landscape, and any other pertinent factors.



IV. BACKGROUND

CyberAi identifies the status of all opened ports on the target server and executes the exploit at pinpoint using Machine Learning. It's key features are following.

- Efficiently execute exploit.
  CyberAi can execute exploits at pinpoint (minimum 1 attempt) using Machine Learning.
- Deep penetration.
  If CyberAi succeeds the exploit to the target server, it further executes the exploit to other internal servers.
- Self-learning.
  CyberAi can learn how to exploitation by itself (uses Reinforcement Learning).It is not necessary for humans to prepare learning data.
- Learning time is very fast.
  Generally, reinforcement learning takes a lot of time. So, CyberAi uses distributed learning by multi agents. I adopted an advanced machine learning model called A3C.
- Powerful intelligence gathering
  To gather the information of software operated on the target server is very important for successful the exploitation. CyberAi can identify product name and version using following methods.
  - Port scanning
  - Machine Learning (Analyze HTTP responses gathered by Web crawling)
  - Contents exploration



A. What is Penetration testing and how it is implemented in CyberAi?

Penetration Testing:

Penetration testing, often referred to as "pentesting" or "ethical hacking", is a simulated cyber attack on a computer system, network, or application. The primary purpose is to identify vulnerabilities that a malicious actor could exploit. By mimicking the actions of an adversary, pentesters can assess the strength of a system's security measures and provide insights into potential weaknesses.[14]

The main steps in a penetration test include:

- Planning & Reconnaissance: Outline the scope of the attack, gather as much information as possible about the target, and identify potential entry points.[16]
- Scanning: Use tools to identify vulnerabilities. This could be via static analysis (reviewing the

application's code) or dynamic analysis (analysing the code while it's running).[18]

- Gaining Access: This involves exploiting the identified vulnerabilities to see if they can be leveraged to breach the system or network.[19]
- Maintaining Access: Here, the goal is to create a situation mimicking an advanced persistent threat—a malicious actor who remains in the system undetected for an extended period.[19]
- Analysis: After the test, the findings are documented, potential impacts are assessed, and remedial suggestions are made.[18]

Implementation of Penetration Testing in CyberAi:

- CyberAi leverages the power of artificial intelligence, specifically deep learning, to automate and enhance various stages of the penetration testing process:
- Automated Targeting: CyberAi can intelligently identify target systems and services, reducing the manual effort traditionally required in the reconnaissance phase.
- Deep Learning-Driven Vulnerability Detection: By training on historical data of known vulnerabilities and successful exploits, CyberAi machine learning model predicts potential vulnerabilities in the target system. This dynamic approach means the tool can potentially identify novel vulnerabilities or be more efficient in its detection compared to traditional tools.
- Exploitation: Integrated with the Metasploit framework, CyberAi can automatically attempt to exploit detected vulnerabilities. This automation drastically speeds up the penetration test process.
- Feedback Loop for Model Refinement: One of the standout features of CyberAi is its ability to learn from its actions. If an attempted exploit is successful or unsuccessful, this information is fed back into the system, allowing the model to adjust and improve over time.
- Reporting: Post-exploitation, CyberAi can automatically generate reports. These reports provide detailed insights into the vulnerabilities detected, the exploits attempted, and their outcomes, giving system administrators a clear understanding of their system's security posture.
- Continuous Learning: CyberAi's architecture supports continuous retraining of its model. As new vulnerabilities emerge and as the system encounters varied environments, it can learn and adapt, making it more effective over time.

CyberAi represents a paradigm shift in penetration testing. By fusing traditional pentesting techniques with the predictive and adaptive capabilities of machine learning, it offers a potent, efficient, and continually improving tool for cybersecurity professionals.

Advantages of Integrating AI in Penetration Testing with CyberAi:

- Scalability: Traditional penetration tests can be resource-intensive and time-consuming, especially when conducted across large networks or multiple applications. CyberAi's automation capabilities allow it to scan, detect, and exploit vulnerabilities across vast networks with relative ease, saving time and resources.[18]
- Adaptability: The cyber threat landscape is constantly evolving with new vulnerabilities and attack vectors emerging regularly. CyberAi 's machine learning component enables it to learn from new data and adapt its strategies, ensuring it remains effective against the latest threats.
- Precision: With machine learning, CyberAi can refine its techniques based on past successes and failures, reducing false positives and ensuring that detected vulnerabilities are genuine and exploitable.
- Consistency: Human pentesters might have off days, overlook details, or be inconsistent in their testing methodologies. CyberAi, on the other hand, operates with a consistent methodology every time, ensuring uniformity in testing.
- Comprehensive Reporting: Automated, detailed reporting ensures that stakeholders receive thorough insights into the security posture of their systems, making it easier to take corrective actions and make informed decisions.

Challenges and Considerations:

- Ethical Concerns: As with all tools that can potentially exploit vulnerabilities, there's a responsibility to ensure that CyberAi is used ethically and legally. It's essential to have proper authorisation before conducting any penetration tests.
- Reliance on Data: The efficacy of CyberAi's machine learning model is heavily reliant on the quality and quantity of training data. If the tool is trained with outdated or irrelevant data, its predictions and performance could be compromised.
- Potential for Misuse: In the wrong hands, CyberAi could be used maliciously. Its automation and efficiency make it a potent tool, emphasising the need for secure handling and stringent access controls.
- Over-reliance on Automation: While CyberAi offers impressive automation capabilities, human intuition, experience, and judgement remain crucial in the realm of cybersecurity. It's essential to view the tool as a complement to human expertise, rather than a complete replacement.

B. What is Reinforcement learning and how it has been implemented in CyberAi?

Reinforcement Learning:

Reinforcement Learning (RL) is a type of machine learning paradigm where an agent learns to behave in an environment by performing certain actions and receiving rewards or penalties in return. The primary objective of RL is for the agent to figure out the best strategy, or policy, to obtain the maximum cumulative reward for its actions over time.[4]

Key Concepts:

- Agent: The decision-maker or learner.
- Environment: What the agent interacts with and where it operates.
- Action: A move made by the agent which affects the environment.
- Reward: Feedback from the environment following an action, indicating the value of the action.
- State: The current situation or configuration of the environment.
- Policy: The strategy used by the agent to determine the next action based on the current state.[4]

The agent learns by exploring the environment, taking actions based on its current policy, and observing the rewards or penalties. Over time, by learning from its experiences, the agent refines its policy to optimise for actions that will accumulate the highest rewards.

Implementation of Reinforcement Learning in CyberAi:
CyberAi utilises Reinforcement Learning, specifically Deep Reinforcement Learning (which combines deep learning and RL), to optimise its penetration testing approach.

- Agent and Environment: In CyberAi's context, the agent is the CyberAi system itself, and the environment is the target system or network it is testing.
- State: This represents the current configuration or status of the target system, including open ports, running services, detected vulnerabilities, etc.
- Action: Actions in this scenario would be the various exploitation attempts CyberAi makes against the target system.
- Reward: After each exploitation attempt, the environment (target system) provides feedback. A successful exploit would give a positive reward, while a failed exploit or detection would give a negative reward or penalty.
- Learning and Policy Refinement: Using the feedback from its actions, CyberAi refines its policy to make better-informed decisions in future penetration tests. For example, if certain exploits consistently provide high rewards (successful penetrations), CyberAi will prioritise them in similar future situations.
- Efficiency and Exploration: One of the challenges in RL is balancing between exploration (trying new actions) and exploitation (relying on known successful actions). CyberAi uses its learning mechanism to dynamically make this balance, ensuring that while it leverages known vulnerabilities effectively, it also explores potential new vulnerabilities or exploitation methods.

In summary, Reinforcement Learning provides CyberAi with a dynamic, adaptable framework that continuously refines its penetration testing strategies based on feedback from the environment. This iterative learning approach ensures that CyberAi not only can exploit known vulnerabilities efficiently but also has the capability to adapt and respond to new or unexpected configurations in the target systems.

C. How A3C is used in CyberAi?

A3C - Asynchronous Advantage Actor-Critic:
A3C, which stands for Asynchronous Advantage Actor-Critic, is an advanced reinforcement learning algorithm. It's a combination of two key ideas: the Actor-Critic architecture and asynchronous updates.[6]

- Actor-Critic Architecture:
  - Actor: Determines which action to take given a certain state. It essentially defines the current policy of the agent.
  - Critic: Estimates the value of taking a certain action in a given state. In other words, it evaluates the actor's policy.[7]
- The Actor-Critic method combines the benefits of value-based approaches (where we estimate a value function) and policy-based approaches (where we directly optimise the policy). The critic assesses the action taken by the actor and gives feedback to improve the policy.[6]
- Asynchronous Updates:
  The asynchronous aspect of A3C means that multiple agent-environment instances run in parallel on separate threads. Each of these agents explores its environment independently. The benefit of this asynchrony is that it helps in decorrelating the experiences of the agents, leading to a more stable and faster learning process.[6]

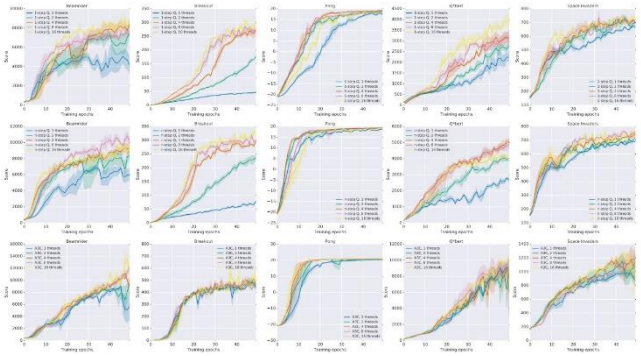
Implementation of A3C in CyberAi:
CyberAi incorporates the A3C algorithm to enhance its decision-making capabilities during penetration testing.

- Parallel Learning:
  CyberAi can run multiple penetration testing routines in parallel (asynchronous agent-environment instances). Each of these routines explores different strategies and vulnerabilities independently.[8]
- Action Selection:
  The "Actor" part of A3C helps CyberAi decide on which exploit or vulnerability to target based on its current knowledge and policy. Given a state (e.g., a detected open port or service on the target system), the actor chooses an action (an exploit attempt).[20]
- Policy Evaluation and Improvement:
  After the action is taken, the "Critic" evaluates the result (whether the exploit was successful or not). Based on this feedback, the critic guides the actor to adjust and improve its policy for future decisions. This feedback loop ensures that CyberAi gets better over time, refining its exploit strategies based on past experiences.[22]
- Exploration and Exploitation:

The A3C framework inherently balances between exploration (probing new exploits or vulnerabilities) and exploitation (utilising known successful exploits). This ensures that CyberAi is both innovative in discovering new vulnerabilities and efficient in exploiting known ones.

- Continuous Learning:
  Due to the asynchronous nature of A3C, CyberAi is constantly updating its knowledge base and policy from multiple sources of feedback. This allows for a faster and more comprehensive learning process.[6]

In essence, the integration of the A3C algorithm provides CyberAi with a robust, adaptive, and efficient learning mechanism. It empowers the tool to continually refine its penetration testing strategies, ensuring it remains effective against an ever-evolving cybersecurity landscape.



V. EXPERIMENTS

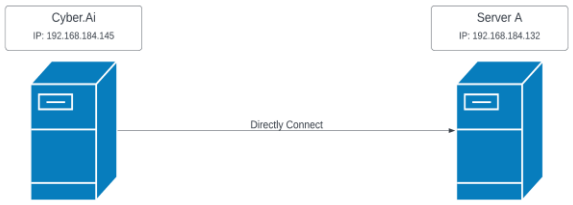
CyberAi is fully automated penetration test tool linked with Metasploit.

CyberAi is intricately linked with Metasploit, one of the most renowned open-source penetration testing frameworks. The integration of these two tools significantly amplifies CyberAi's capabilities, allowing it to utilise the vast array of exploits and payloads present in the Metasploit framework. Here's how CyberAi is connected with Metasploit and leverages its features:

- Framework Integration:
  CyberAi has been designed to work seamlessly with the Metasploit framework. It uses the Metasploit's RPC (Remote Procedure Call) server to communicate with it. This allows CyberAi to harness Metasploit's functionalities programmatically and in an automated fashion.
- Exploit Database:
  - Metasploit contains a vast and continually updated database of exploits. When CyberAi identifies a potential vulnerability in a target system, it queries this database to fetch suitable exploits.[25]
  - CyberAi's machine learning model then assesses which exploit has the highest probability of success, based on the gathered information about the target and past feedback.
- Payload Selection:
  - Once an exploit is chosen, CyberAi can also utilise Metasploit's extensive range of payloads. A payload is a piece of code executed after successful exploitation.[23]
  - Depending on the objective (e.g., establishing a reverse shell, executing a specific command, etc.), CyberAi selects the most appropriate payload from Metasploit.[23]
- Information Gathering:
  Metasploit provides several auxiliary modules and scanners for information gathering. CyberAi can utilise these modules to fetch valuable information about a target system, which in turn informs the exploit and payload selection process.
- Post-Exploitation Modules:
  Upon successful exploitation, CyberAi can leverage Metasploit's post-exploitation modules to carry out various tasks, from data extraction to privilege escalation.
- Feedback Loop Integration:
  - After each exploit attempt, CyberAi receives feedback on whether the exploit was successful or not. This feedback is crucial for the tool's reinforcement learning model.[24]
  - Metasploit provides detailed results of each exploit attempt, enabling CyberAi to refine its strategy and improve its predictive accuracy over time.[24]
- Reporting:
  Using data from both its internal operations and Metasploit's detailed logs, CyberAi can generate comprehensive reports outlining the vulnerabilities detected, exploits attempted, success rates, and more.

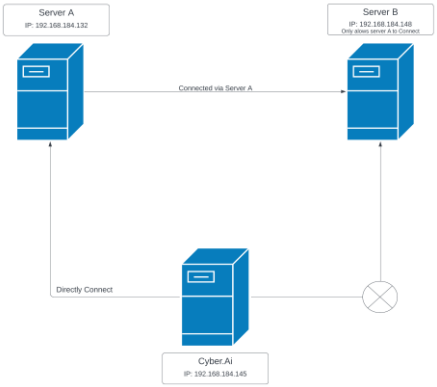
Experiment 1: Single target server

This case is very simple. In this case the CyberAi execute the exploit to directly reachable server.



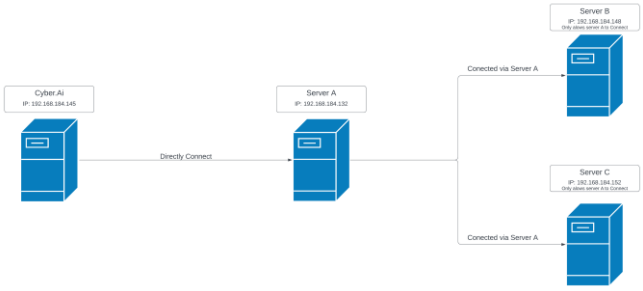
Experiment 2: Exploitation via compromised server

The CyberAi can directly connect to the Server-A, but cannot directly connect to the Server-B. So, the CyberAi must execute the exploit to the Server-B via the Server-A.



Experiment 3: Deep Penetration

The CyberAi can directly connect to the Server-A, but cannot directly connect to the Server-B and C. So, the CyberAi must execute the exploit to the Server-B and C via Server-A.




```
root@kali:~/ Cyber.AI# python3 CyberAi.py -t 192.168.184.129 -m test
```

-t: This is an argument flag, which typically stands for "target" in penetration testing tools.
192.168.184.129: This is the IP address of the target system on which CyberAi will attempt its testing. Essentially, you're instructing CyberAi to target this specific machine.
-m: This is another argument flag, which typically stands for "mode".
test: This indicates the mode in which CyberAi will operate. In this context, the "test" mode likely means that CyberAi will run in a testing or trial manner, perhaps to ensure that everything is working correctly or to perform a non-invasive check.

You can prohibit scanning of specific server using config.ini. If you want to prohibit scanning specific server, please add IP address of target server to prohibited_list in config.ini as following.

```
[Metasploit]
lport          : 4444
proxy_host     : 127.0.0.1
proxy_port     : 1080
prohibited_list : 192.168.220.1@192.168.220.2@192.168.220.254
path_collection : path@uri@dir@folder@file
```

VI. CONCLUTION

CyberAi, as illuminated throughout this research paper, emerges as a pioneering step in the fusion of artificial intelligence with cybersecurity. Its profound ability to utilise machine learning, particularly reinforcement learning, marks a significant advancement in automated penetration testing. While traditional tools often rely on static methods and predefined vulnerability lists, CyberAi adapts, learns, and refines its techniques based on outcomes, making it a dynamic and evolving solution.

The integration of this tool with Metasploit offers an expansive playground, capitalising on a vast repository of exploits and payloads. Such a merger ensures that while the tool thinks and learns like a contemporary AI model, its roots remain firmly entrenched in proven cybersecurity frameworks.

However, with cutting-edge technologies come challenges and responsibilities. CyberAi's complexity, the imperative nature of its training, and the potential ethical dilemmas it poses, are areas requiring careful consideration. Ensuring that it is used ethically, is kept updated, and is complemented by human expertise will be essential to realising its full potential.

In conclusion, CyberAi showcases the future of penetration testing – a blend of human expertise and machine intelligence. While it is not a panacea for all cybersecurity challenges, it is undeniably a leap forward, offering a more adaptive, efficient, and comprehensive approach to vulnerability assessment. As cybersecurity threats grow and evolve, tools like CyberAi will undoubtedly play a pivotal role in fortifying digital defences and ensuring a safer cyber landscape.

VII. FUTURE WORK

As the landscape of cybersecurity continuously evolves, so too must the tools designed to safeguard it. With respect to CyberAi, several potential avenues can be explored to further enhance its capabilities and address its current limitations. Here are some proposed future directions:

- Integration with Other Frameworks: Beyond Metasploit, integrating CyberAi with other cybersecurity frameworks and tools can expand its reach and versatility. This would allow it to tap into a broader range of vulnerabilities and solutions.[21]
- Enhanced Machine Learning Models: While reinforcement learning has proven effective for CyberAi, the exploration of other machine learning models or even hybrid models could offer improved prediction accuracy and reduced false positive/negative rates.[19]
- Real-time Learning: Incorporating real-time learning capabilities would enable CyberAi to adapt instantly to new threats, especially zero-day vulnerabilities, making its response more immediate and effective.[18]
- Scalability: Designing the tool to efficiently handle larger network infrastructures or cloud environments would increase its applicability in enterprise-scale scenarios.[22]

- User-friendly Interface: Developing a more intuitive graphical user interface (GUI) would make CyberAi more accessible to professionals who might not be well-versed in machine learning, ensuring broader adoption.[16]
- Ethical and Safe Use Mechanisms: Implementing features that prevent or deter malicious use and ensure that CyberAi is used primarily for ethical hacking and legitimate penetration testing purposes.[22]
- Collaborative Learning: Incorporating a system where multiple instances of CyberAi can learn collaboratively from each other's experiences, pooling knowledge, and refining techniques.[17]
- Customised Training Scenarios: Allowing users to design specific training scenarios tailored to their unique environment or industry-specific threats.[10]
- Enhanced Reporting: Developing more detailed and actionable reporting mechanisms, offering insights not just into vulnerabilities but also suggesting potential remediation strategies.[14]
- Community-driven Updates: Establishing a platform where the cybersecurity community can contribute to CyberAi's training data, ensuring it remains updated with the latest real-world threats and vulnerabilities.[9]

In conclusion, while CyberAi has made substantial strides in automated penetration testing, the journey is far from over. As cyber threats continue to diversify and intensify, the tool's evolution will be essential in ensuring robust and state-of-the-art defence mechanisms. The proposed future directions aim to capitalise on this potential, fostering a more secure and resilient digital world.

REFERENCES

[1] *2020 IEEE 11th International Conference on Software Engineering and Service Science (ICSESS)*

[2] *Y.Y. Ye, T. Li, Y. Chen et al., "Automatic malware categorization using cluster ensemble"*, *Washington: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2010.

[3] *X.Y. Zhao and S.F. Ding, "A review of deep reinforcement learning research"*, Computer Science, *vol. 45, no. 7, pp. 1-6, 2018.*

[4] *2022 IEEE 22nd International Conference on Communication Technology (ICCT)*

[5] *G Gerogiannis, M Birbas, A Leftheriotis et al., "Deep Reinforcement Learning Acceleration for Real-Time Edge Computing Mixed Integer Programming Problem"*, IEEE Access, *pp. 18526-18543, 2022.*

[6] *V Mnih, A P Badia, M Mirza et al., "Asynchronous methods for deep reinforcement learning"*, International conference on machine learning, *pp. 1928-1937, 2016.*

[7] *2023 IEEE 24th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*

[8] *V. Mnih et al., "2016. asynchronous methods for deep reinforcement learning"*, International Conference on Machine Learning 19281937, *pp. 19281937, 2016.*

[9] *T. Huang and L. Sun, "Deepmpc: A mixture abr approach via deep learning and mpc"*, 2020 IEEE International Conference on Image Processing (ICIP), *pp. 1231-1235, 2020.*

[10] *IEEE Transactions on Information Forensics and Security (Volume: 18)*

[11] *M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks"*, Proc. 36th Int. Conf. Mach. Learn. (ICML), *pp. 6105-6114, 2019.*

[12] *J. Kodovský, J. Fridrich and V. Holub, "Ensemble classifiers for steganalysis of digital media"*, IEEE Trans. Inf. Forensics Security, *vol. 7, no. 2, pp. 432-444, Apr. 2012.*

[13] *Van Hasselt, Hado, Guez, Arthur, and Silver, David. Deep reinforcement learning with double q-learning. arXiv preprint arXiv:1509.06461, 2015.*

[14] *Tsitsiklis, John N. Asynchronous stochastic approximation and q-learning. Machine Learning, 16(3):185–202, 1994.*

[15] *Lillicrap, Timothy P, Hunt, Jonathan J, Pritzel, Alexander, Heess, Nicolas, Erez, Tom, Tassa, Yuval, Silver, David, and Wierstra, Daan. Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971, 2015*

[16] *"Metasploit Unleashed", 2014*

[17] *K. Park, Y. Song, and Y.-G. Cheong, "Classification of Attack Types for Intrusion Detection Systems Using a Machine Learning Algorithm," in 2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (BigDataService), Bamberg, pp. 282–286,2018.*

[18] *K. Sadhukhan, R. A. Mallari, and T. Yadav, "Cyber Attack Thread: A control-flow based approach to deconstruct and mitigate cyber threats," in 2015 International Conference on Computing and Network Communications (CoCoNet), Trivandrum, India, pp. 170– 178, 2015.*

[19] *Ö. Aslan and R. Samet, "Mitigating Cyber Security Attacks by Being Aware of Vulnerabilities and Bugs," 2017 International Conference on Cyberworlds (CW), Chester, pp.222-225, 2017.*

[20] *F. Holik, J. Horalek, O. Marik, S. Neradova and S. Zitta, "Effective penetration testing with Metasploit framework and methodologies," 2014 IEEE 15th International Symposium on Computational Intelligence and Informatics (CINTI), Budapest, pp. 237-242, 2014.*

[21] *A. Ghafarian, "Using Kali Linux Security Tools to Create Laboratory Projects for Cybersecurity Education," in Proceedings of the Future Technologies Conference (FTC) 2018, vol. 881, Cham: Springer International Publishing, pp. 358–367, 2019.*

[22] *Bellemare, Marc G., Ostrovski, Georg, Guez, Arthur, Thomas, Philip S., and Munos, Rémi. Increasing the action gap: New operators for*

reinforcement learning. In Proceedings of the AAAI Conference on Artificial Intelligence, 2016

[23] 2014 IEEE 15th International Symposium on Computational Intelligence and Informatics (CINTI)

[24] "OWASP Testing Guide", 2008.

[25] S. Kalia and M. Singh, "Masking approach to secure systems from Operating system Fingerprinting", Tencon 2005 2005 Ieee Region 10, *Dec. 2005.*

MSc Project - Reflective Essay

Project Titile:	Cyber.Ai : AI powered Cyber Security tool
Student Name:	Sparsh Verma
Student Number:	220996233
Supervisor Name:	Syed Rafee
Programme of Study:	MSc. Artificial Intelligence

1. Advantages of CyberAi:

- Automated Penetration Testing: CyberAi automates the process of penetration testing, reducing the time and effort required by security professionals to find and exploit vulnerabilities.
- Learning Capability: Powered by machine learning, specifically reinforcement learning, CyberAi can learn from previous attempts and improve its efficiency over time[1]. This adaptability is absent in traditional penetration testing tools.
- Integration with Metasploit: CyberAi's integration with the renowned Metasploit framework provides access to a vast repository of exploits and payloads, enhancing its capability to test across a myriad of vulnerabilities.
- Efficiency: By learning from past attempts, CyberAi can more efficiently decide which exploits are likely to succeed, reducing the number of unnecessary or failed attempts.
- Adaptability: CyberAi can be trained on new vulnerabilities or environments, making it adaptable to the ever-evolving landscape of cybersecurity.
- Comprehensive Reporting: It provides detailed reports of its testing process, enabling security professionals to understand vulnerabilities, exploit attempts, and results more comprehensively.

Disadvantages of CyberAi:

- Complexity: Implementing machine learning in cybersecurity tools introduces a layer of complexity. Setting up, configuring, and understanding CyberAi might be challenging for those unfamiliar with machine learning concepts.
- Dependence on Training: The efficacy of CyberAi is significantly dependent on its training. If not properly trained or updated, its efficiency can decrease.
- Potential False Positives/Negatives: Like all automated tools, CyberAi might have instances of false positives (identifying a non-existent vulnerability) or false negatives (missing a real vulnerability).
- Resource Intensive: Machine learning models, especially deep learning models, can be computationally intensive. Running CyberAi might require more computational resources compared to more straightforward penetration testing tools.
- Ethical Concerns: There's always a concern that such tools can fall into the wrong hands. If used maliciously, they could be utilised for illegal hacking attempts, given their automation and efficiency.
- Over-reliance: Security professionals might become overly reliant on tools like CyberAi and underplay the importance of human intuition and expertise in the penetration testing process.

In summary, while CyberAi brings automation and adaptability to the table, making penetration testing more efficient and comprehensive, it also introduces complexities and challenges that need to be addressed by its users and developers. Proper training, understanding of its operations, and regular updates are crucial to harness its benefits fully.

2. Future work possibility

As the landscape of cybersecurity continuously evolves, so too must the tools designed to safeguard it. With respect to CyberAi, several potential avenues can be explored to further enhance its capabilities and address its current limitations. Here are some proposed future directions:

- Integration with Other Frameworks: Beyond Metasploit, integrating CyberAi with other cybersecurity frameworks and tools can expand its reach and versatility. This would allow it to tap into a broader range of vulnerabilities and solutions.[1]
- Enhanced Machine Learning Models: While reinforcement learning has proven effective for CyberAi, the exploration of other machine learning models or even hybrid models could offer improved prediction accuracy and reduced false positive/negative rates.[2]
- Real-time Learning: Incorporating real-time learning capabilities would enable CyberAi to adapt instantly to new threats, especially zero-day vulnerabilities, making its response more immediate and effective.[3]
- Scalability: Designing the tool to efficiently handle larger network infrastructures or cloud environments would increase its applicability in enterprise-scale scenarios.[4]

- User-friendly Interface: Developing a more intuitive graphical user interface (GUI) would make CyberAi more accessible to professionals who might not be well-versed in machine learning, ensuring broader adoption.[5]
- Ethical and Safe Use Mechanisms: Implementing features that prevent or deter malicious use and ensure that CyberAi is used primarily for ethical hacking and legitimate penetration testing purposes.[5]
- Collaborative Learning: Incorporating a system where multiple instances of CyberAi can learn collaboratively from each other's experiences, pooling knowledge, and refining techniques.[8]
- Customised Training Scenarios: Allowing users to design specific training scenarios tailored to their unique environment or industry-specific threats.[7]
- Enhanced Reporting: Developing more detailed and actionable reporting mechanisms, offering insights not just into vulnerabilities but also suggesting potential remediation strategies.[6]
- Community-driven Updates: Establishing a platform where the cybersecurity community can contribute to CyberAi's training data, ensuring it remains updated with the latest real-world threats and vulnerabilities.[9]

In conclusion, while CyberAi has made substantial strides in automated penetration testing, the journey is far from over. As cyber threats continue to diversify and intensify, the tool's evolution will be essential in ensuring robust and state-of-the-art defence mechanisms. The proposed future directions aim to capitalise on this potential, fostering a more secure and resilient digital world.

3. Legal and Ethical guidelines

Legal Considerations:

- Permission: Always ensure you have explicit permission before testing any system or network. Unauthorized access, even if the intent is benign, can lead to severe legal repercussions.[6]
- Jurisdiction: Cybersecurity laws vary across countries. Before undertaking any testing, familiarise yourself with the laws and regulations of the jurisdiction you are operating within.[6]
- Data Protection and Privacy: If during testing, you encounter personal or sensitive data, handle it with care. Breaching data protection laws, even inadvertently, can result in significant legal penalties. This includes GDPR in Europe, CCPA in California, and other regional data protection laws.[5]
- Disclosure: If you identify a vulnerability in a system, the manner in which you disclose this information is crucial. Ideally, privately inform the concerned entity and give them adequate time to address the issue before any public disclosure.[3]

Ethical Considerations:

- Intent: Always use CyberAi for its intended purpose – to identify and rectify vulnerabilities, not exploit them maliciously.[3]
- Scope: Adhere strictly to the scope of the test. If you're given permission to test certain systems or parts of a network, do not exceed those boundaries.[9]
- Confidentiality: Maintain strict confidentiality of any information or data you come across during your testing.[7]
- Respect: Systems and networks are the products of individuals' or organisations' hard work. Respect their integrity and avoid causing unnecessary harm or disruption.[2]
- Transparency: If you're conducting a test, be transparent about your methods, tools, and findings, especially if working as a part of a team or for a client.[7]
- Continuous Learning: The cybersecurity landscape is ever-evolving. Regularly update your skills and knowledge and stay informed about the ethical considerations in the field.[9]

While CyberAi can automate many aspects of penetration testing, it can't automate ethical decision-making. Being aware of and adhering to legal and ethical guidelines is imperative not just for the reputation of individual testers, but for the broader acceptance and development of cybersecurity practices. As with any powerful tool, with great power comes great responsibility.

4. Conclusion

CyberAi, as illuminated throughout this research paper, emerges as a pioneering step in the fusion of artificial intelligence with cybersecurity. Its profound ability to utilise machine learning, particularly reinforcement learning, marks a significant advancement in automated penetration testing. While traditional tools often rely on static methods and predefined vulnerability lists, CyberAi adapts, learns, and refines its techniques based on outcomes, making it a dynamic and evolving solution.

The integration of this tool with Metasploit offers an expansive playground, capitalising on a vast repository of exploits and payloads. Such a merger ensures that while the tool thinks and learns like a contemporary AI model, its roots remain firmly entrenched in proven cybersecurity frameworks.

However, with cutting-edge technologies come challenges and responsibilities. CyberAi's complexity, the imperative nature of its training, and the potential ethical dilemmas it poses, are areas requiring careful consideration. Ensuring that it is used ethically, is kept updated, and is complemented by human expertise will be essential to realising its full potential.

In conclusion, CyberAi showcases the future of penetration testing – a blend of human expertise and machine intelligence. While it is not a panacea for all cybersecurity challenges, it is undeniably a leap forward, offering a more adaptive, efficient, and comprehensive approach to vulnerability assessment. As cybersecurity threats grow and evolve, tools like CyberAi will undoubtedly play a pivotal role in fortifying digital defences and ensuring a safer cyber landscape.

5. Reference

[1] *G Gerogiannis, M Birbas, A Leftheriotis et al., "Deep Reinforcement Learning Acceleration for Real-Time Edge Computing Mixed Integer Programming Problem",* IEEE Access, *pp. 18526-18543, 2022*

[2] *K. Sadhukhan, R. A. Mallari, and T. Yadav, "Cyber Attack Thread: A control-flow based approach to deconstruct and mitigate cyber threats," in 2015 International Conference on Computing and Network Communications (CoCoNet), Trivandrum, India, pp. 170– 178, 2015.*

[3] *Lillicrap, Timothy P, Hunt, Jonathan J, Pritzel, Alexander, Heess, Nicolas, Erez, Tom, Tassa, Yuval, Silver, David, and Wierstra, Daan. Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971, 2015*

[4] *"Metasploit Unleashed", 2014*

[5] *K. Park, Y. Song, and Y.-G. Cheong, "Classification of Attack Types for Intrusion Detection Systems Using a Machine Learning Algorithm," in 2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (BigDataService), Bamberg, pp. 282– 286,2018.*

[6] *Y.Y. Ye, T. Li, Y. Chen et al., "Automatic malware categorization using cluster ensemble", Washington: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* 2010.

[7] *X.Y. Zhao and S.F. Ding, "A review of deep reinforcement learning research",* Computer Science, *vol. 45, no. 7, pp. 1-6, 2018.*

[8] *2022 IEEE 22nd International Conference on Communication Technology (ICCT)*

[9] *G Gerogiannis, M Birbas, A Leftheriotis et al., "Deep Reinforcement Learning Acceleration for Real-Time Edge Computing Mixed Integer Programming Problem",* IEEE Access, *pp. 18526-18543, 2022.*