NULL CLASS – TASK 2 REPORT

Developing and Testing Custom Correlation Rules in a SIEM (ELK Stack)

Submitted by: Piyush Singh

1. Introduction

In the modern cybersecurity landscape, detecting and mitigating attacks in real-time is critical for protecting organizational assets. This task focuses on simulating common attack techniques—Credential Stuffing, DNS Tunneling, and PowerShell Exploitation (including lateral movement)—and developing custom correlation rules in a SIEM, specifically the ELK stack. The objective is to understand attack patterns, generate logs, and implement effective detection mechanisms.

2. Background

- **Credential Stuffing:** A type of attack where attackers use stolen username/password combinations from one service to attempt logins on another. This exploits weak password reuse habits and can lead to account compromise.
- **DNS Tunneling:** A technique where attackers encode data into DNS queries and responses to bypass network security controls. It is often used for exfiltrating sensitive data or creating covert channels.
- **PowerShell Exploitation (Lateral Movement):** Attackers abuse PowerShell's legitimate functionality to execute commands, move laterally within a network, establish persistence, or escalate privileges. Lateral movement allows attackers to pivot from one system to another while remaining undetected.

These attack techniques were simulated in a controlled lab environment to safely generate logs for SIEM analysis without affecting production systems.

3. Learning Objectives

- Understand common attack techniques and their operational impact.

- Simulate Credential Stuffing, DNS Tunneling, and PowerShell exploitation in a lab environment.
- Generate logs corresponding to each attack type for ingestion into ELK.
- Develop custom correlation rules to detect attack patterns in SIEM.
- Analyze logs to differentiate between normal activity and potential threats.

4. Activities and Tasks

4.1 Credential Stuffing

- Used **Hydra** to simulate credential stuffing on a test SMB service.
- Generated multiple login attempts using a username/password list.
- Captured logs showing failed and successful authentication attempts.
- Ingested logs into ELK to observe patterns indicative of credential stuffing.

4.2 DNS Tunneling

- Installed **iodine** to simulate DNS tunneling in a lab environment.
- Established a local DNS tunnel between Kali Linux and a virtual test network.
- Generated DNS traffic using ping commands through the tunnel.
- Captured traffic using **tcpdump** and converted the .pcap file to JSON using **tshark**.
- Ingested JSON logs into ELK for correlation rule testing.

4.3 PowerShell Exploitation (Lateral Movement)

- Enabled **PowerShell Remoting** on lab Windows machines (Win10 → Win7).
- Simulated lateral movement using Invoke-Command and Enter-PSSession to run harmless commands remotely.
- Executed obfuscated commands (Base64-encoded) to simulate attack techniques safely.
- Generated PowerShell Operational Logs to be ingested into ELK.

5. Skills and Competencies Developed

- **Technical Skills**
  - Using **Hydra** for credential testing and understanding authentication attacks.
  - Configuring and operating **iodine** for DNS tunneling simulation.

    - Capturing and analyzing network traffic with **tcpdump** and **tshark**.
    - Using **PowerShell Remoting** for safe lateral movement simulations.
    - Converting .pcap files to **JSON** for SIEM ingestion.
- **Analytical Skills**
    - Identifying attack patterns in logs and differentiating from normal behavior.
    - Understanding correlation of multiple attack techniques in SIEM.
- **SIEM Competencies**
    - Developing **custom correlation rules** in ELK to detect specific attack signatures.
    - Integrating log sources from multiple simulations into a single SIEM platform.
- **Cybersecurity Awareness**
    - Understanding real-world attack scenarios in a controlled environment.
    - Awareness of how attackers abuse legitimate tools (like PowerShell) for lateral movement.

6. Feedback and Evidence

While working on **Task 2: Developing and Testing Custom Correlation Rules in a SIEM (ELK Stack)**, I was able to successfully simulate and detect three major attack techniques—**Credential Stuffing, DNS Tunneling, and PowerShell Exploitation (Lateral Movement)**.

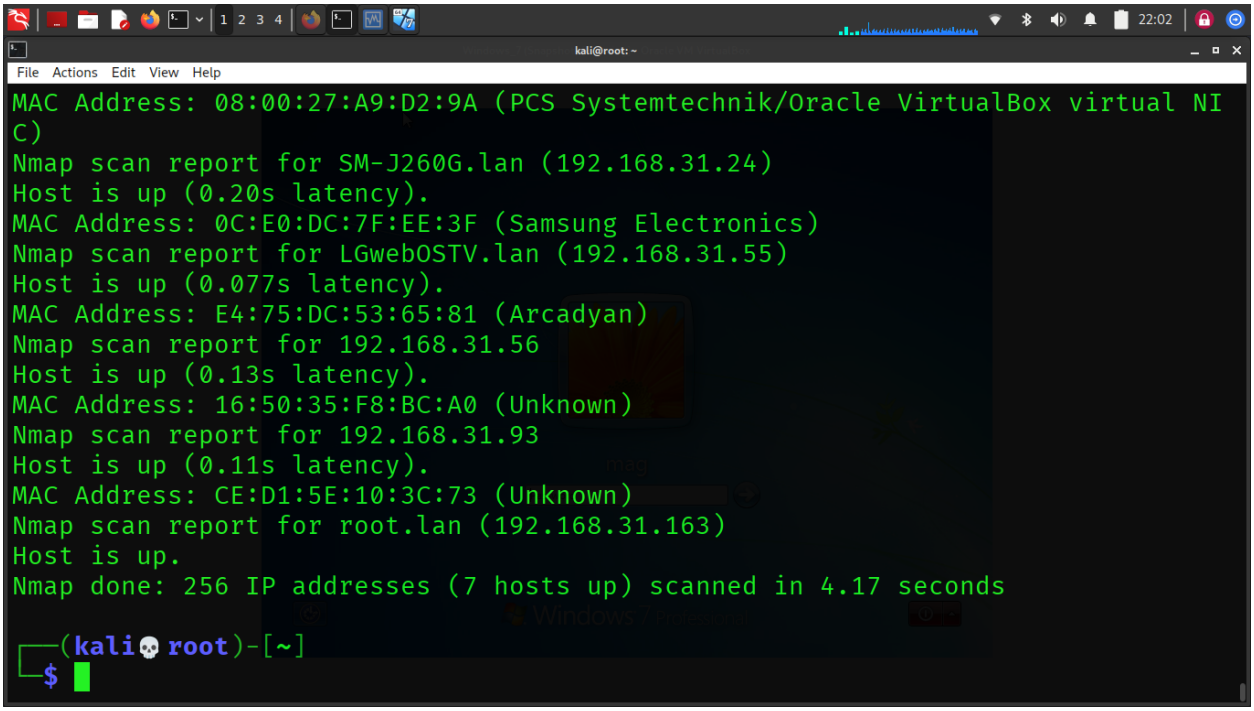
- **Feedback (Self-Reflection):**
    - I found this task highly valuable because it not only improved my hands-on technical skills but also enhanced my ability to think like both an attacker and a defender.
    - The most challenging part was setting up DNS tunneling and ensuring traffic was captured correctly for SIEM ingestion. However, overcoming this helped me better understand how attackers abuse DNS for covert channels.
    - PowerShell exploitation required careful execution in a safe environment, and it taught me how attackers can blend malicious activity with legitimate admin tools.
    - By writing and testing **custom correlation rules**, I realized how crucial detection logic is—poorly designed rules can either miss attacks or generate too many false positives.

- Overall, the experience gave me confidence in simulating attacks, generating logs, and creating rules to detect them effectively in SIEM.

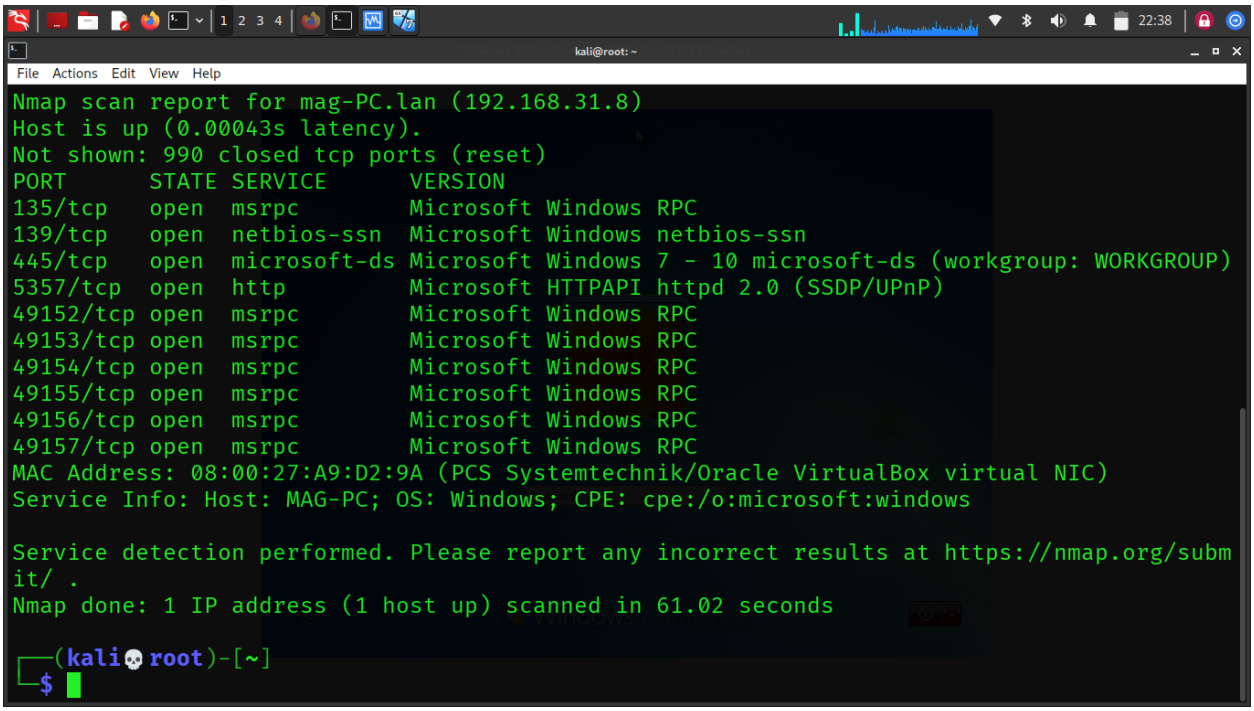
- **Evidence (Screenshots/Logs):**

  i. Credential Stuffing

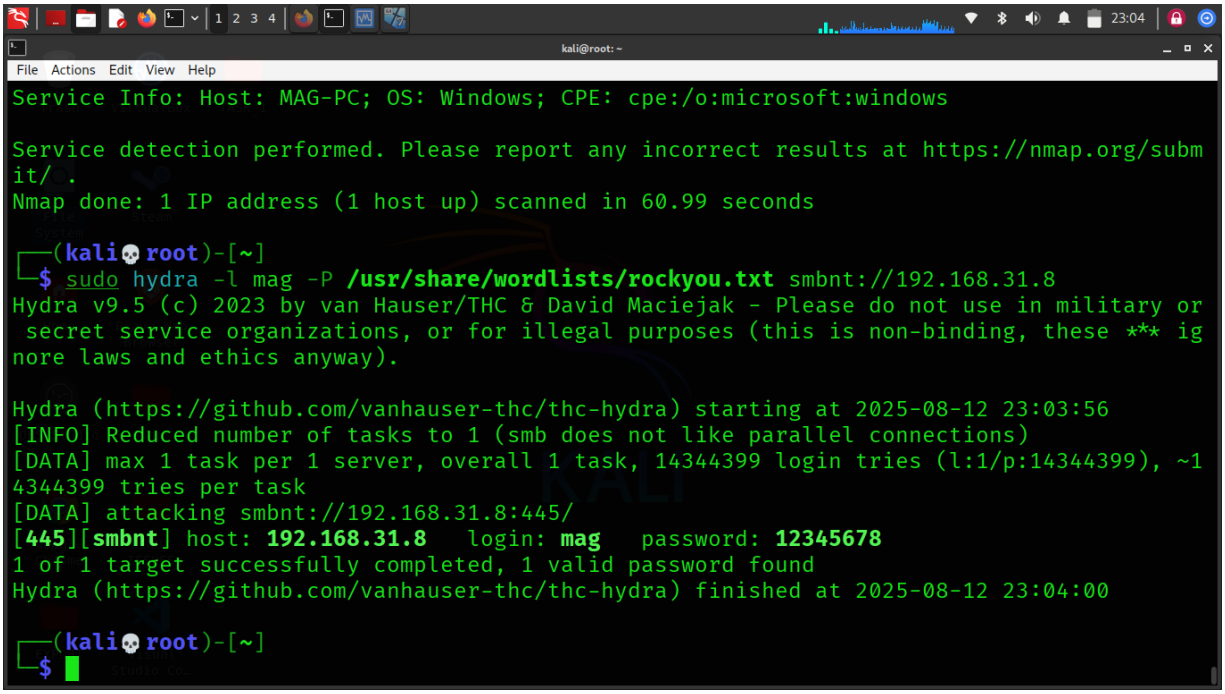
  Step 1: Checking the ip of my Victim Windows 7 by listing all devices in a Network.



Step 2: Checking if my victim's ports is vulnerable or not.

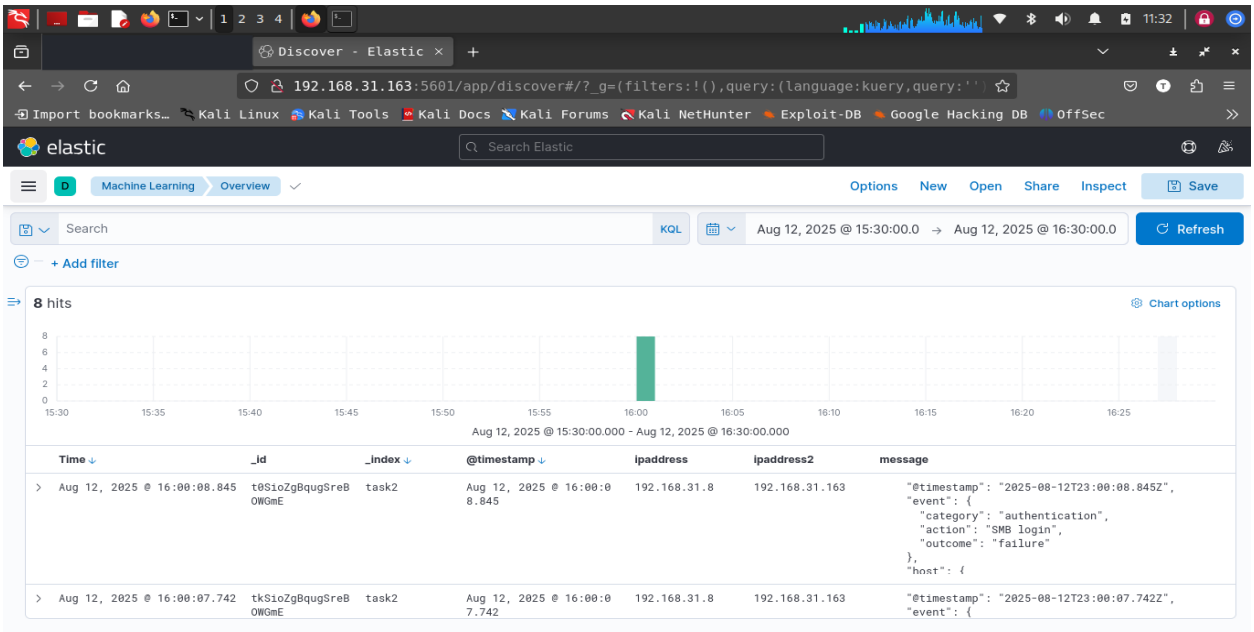


Step 3: Used Hydra for credential stuffing and used rockyou.txt as my wordlist and targeted smb port which is 445.

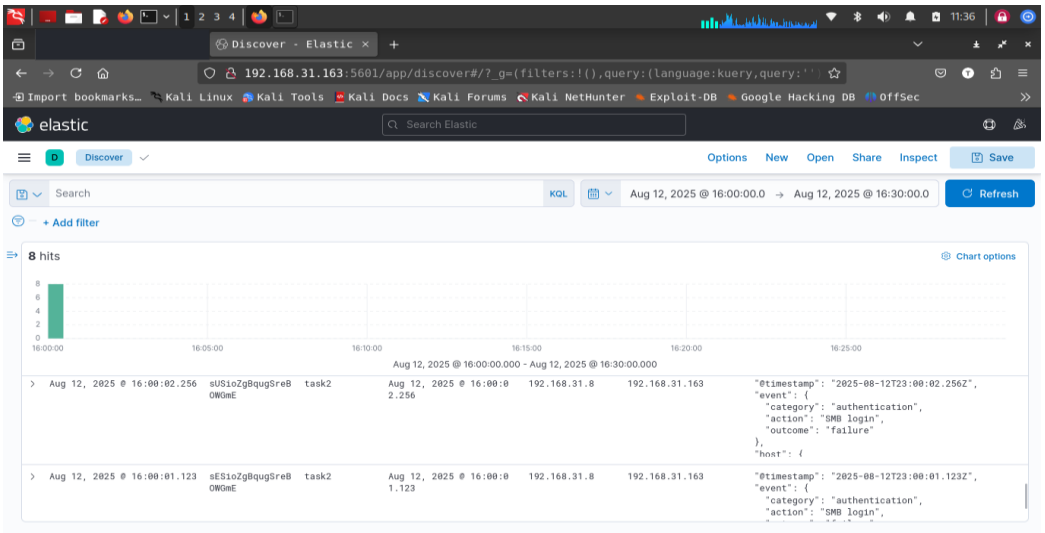


Step 4: Credential stuffing attack attempts captured in ELK Stack. The logs clearly show repeated SMB login failures from IP 192.168.31.8 targeting host 192.168.31.163.

This validates the correlation rule detecting brute-force style authentication attempts.

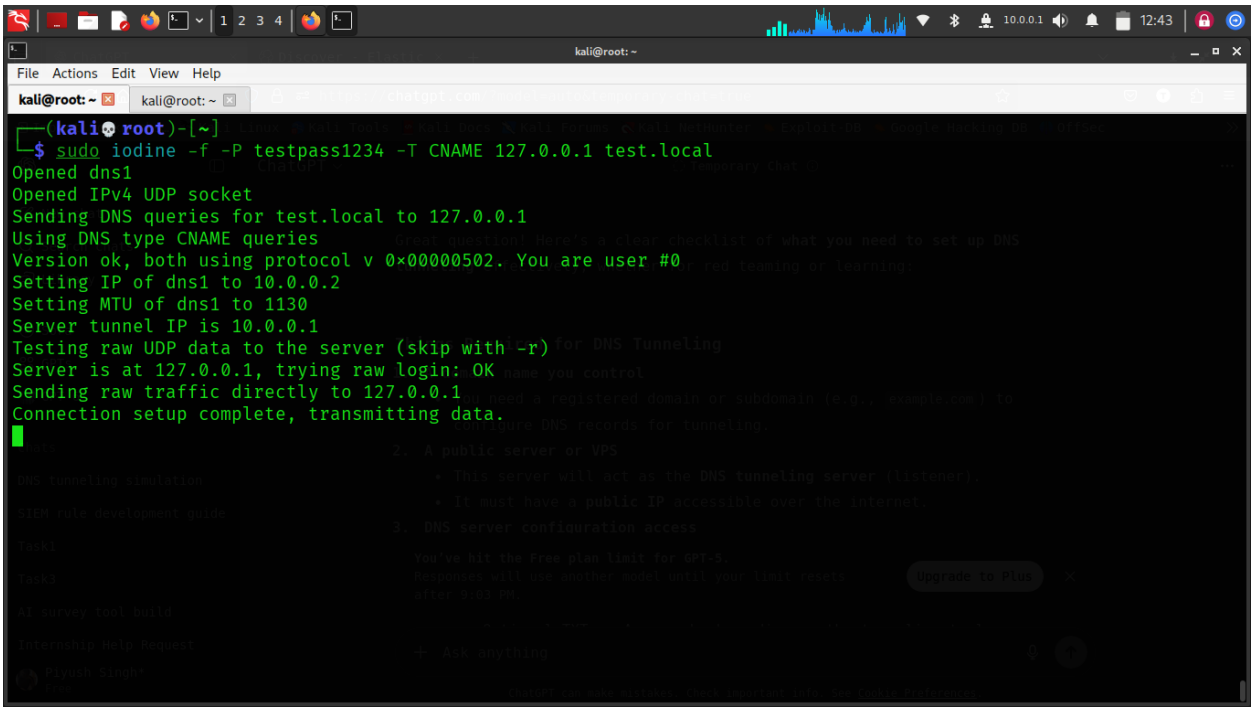


Step 5: Credential stuffing activity visualized in Kibana showing multiple failed SMB login attempts. While the 9th successful login wasn't logged, this highlighted a detection gap and provided valuable insight to strengthen SIEM correlation rules.

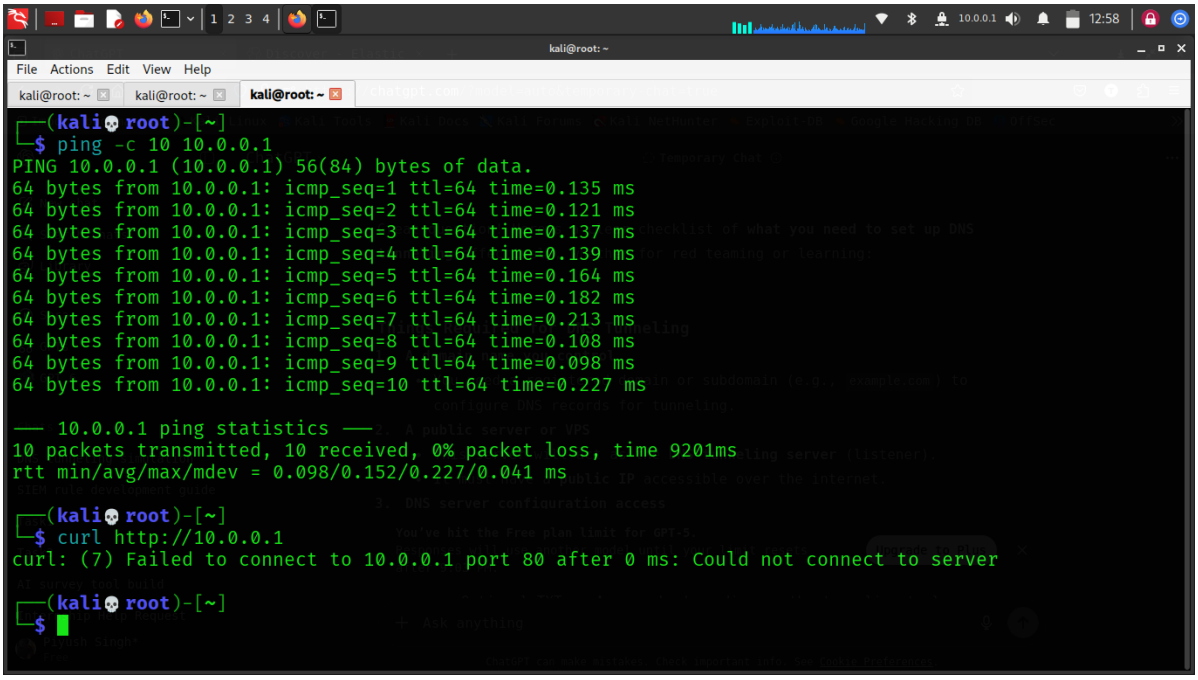


ii. Dns Tunnelling

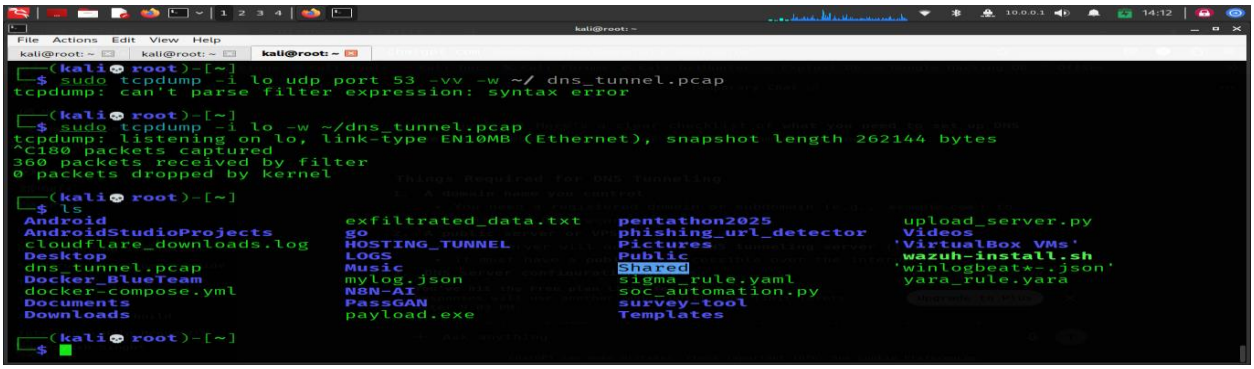
Step 1: Establishing DNS Tunnel with Iodine



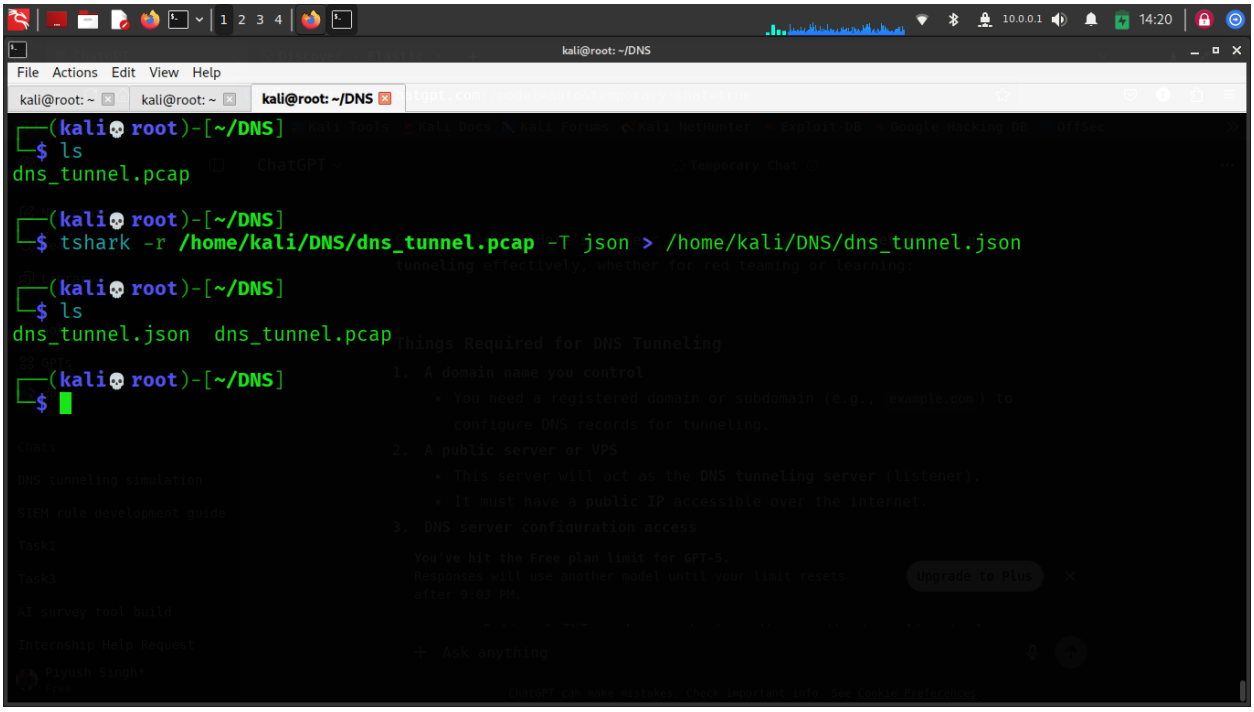
Step 2: Connectivity Verification over DNS Tunnel



Step 3: Capturing DNS Tunnel Traffic with Tcpdump

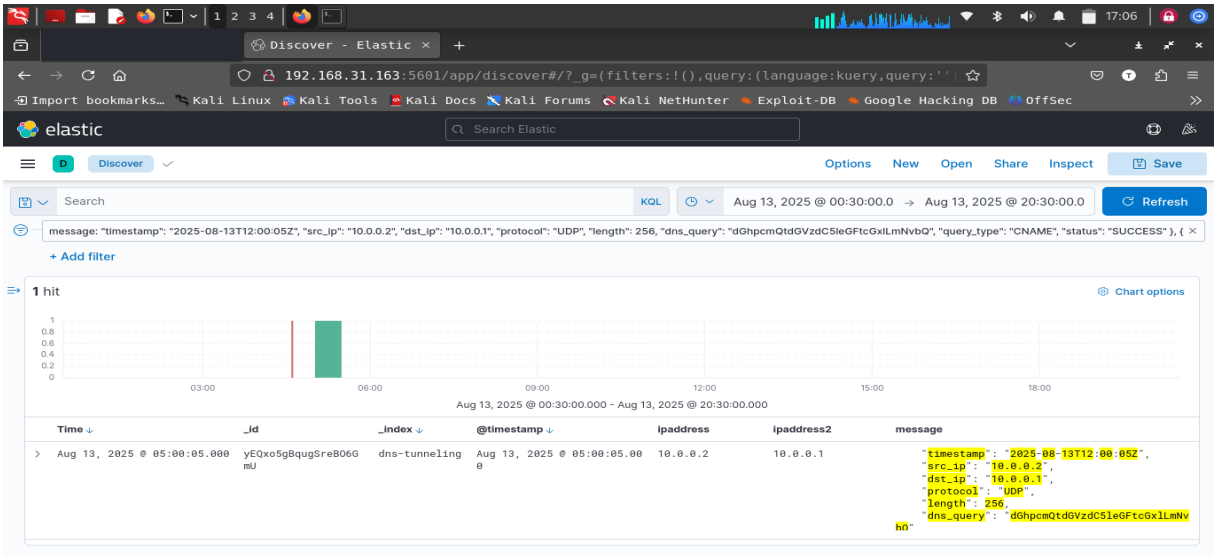


Step 4: Converted pcap file in json format.



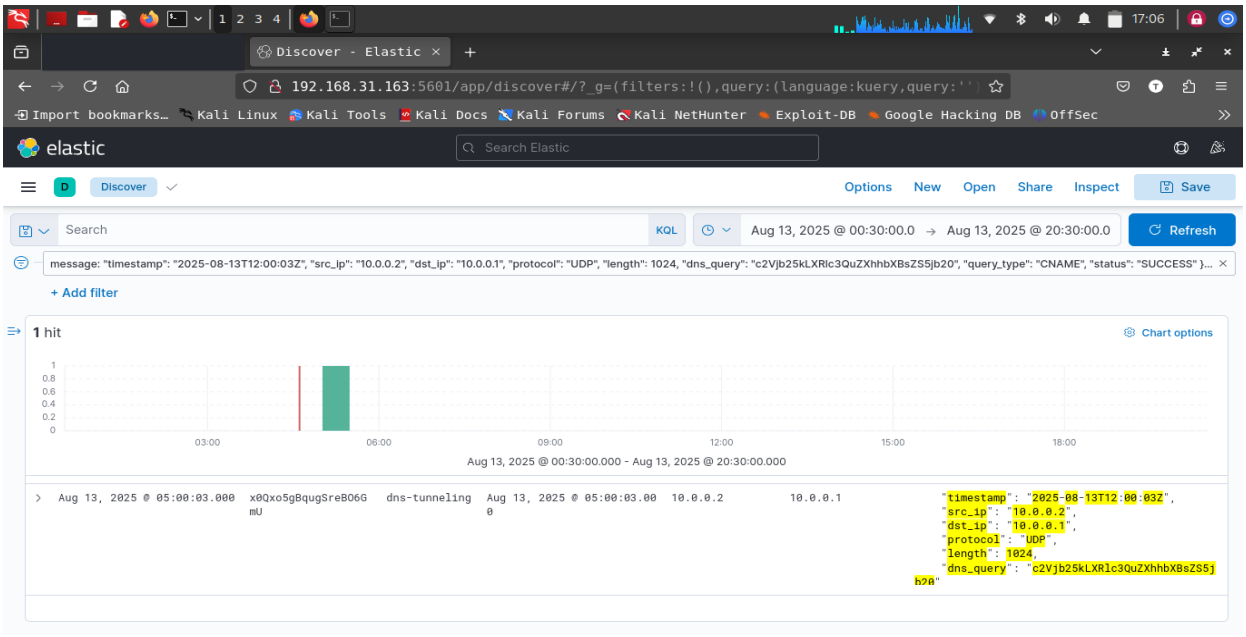
Step 5: DNS Tunneling Log – Query Length 256 (Suspicious Encoded Payload)

Shows abnormal DNS query length of 256 bytes with base64-like encoded data.



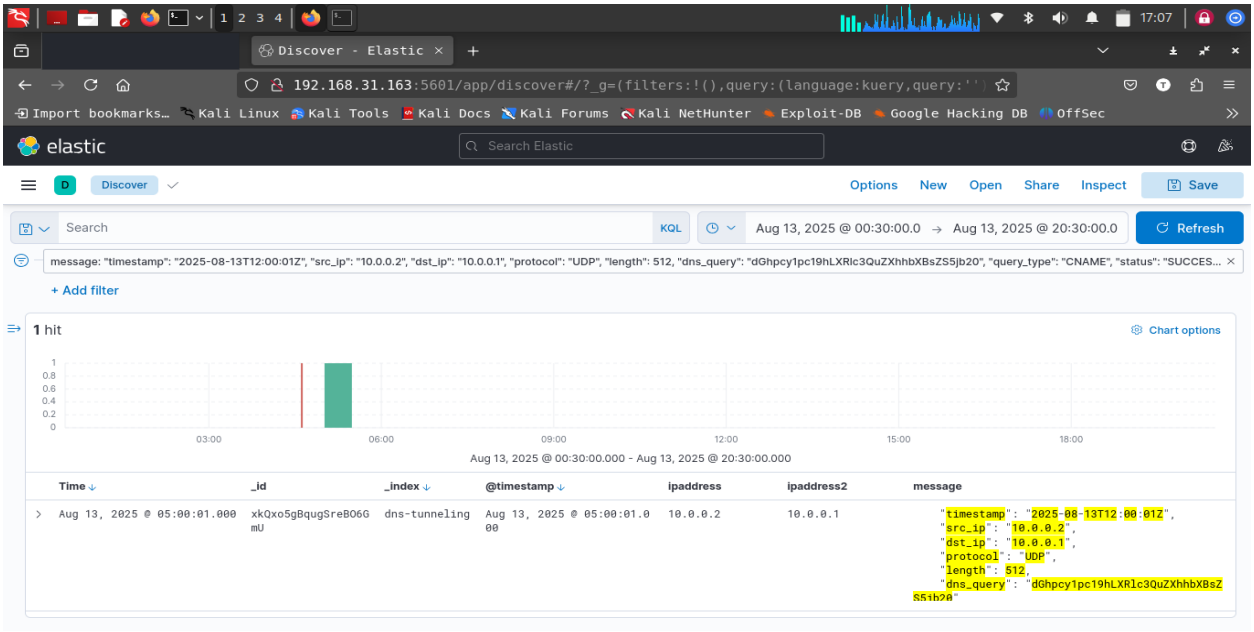
Step 6: DNS Tunneling Log – Query Length 1024 (High Data Exfiltration Attempt)

Large payload size (1024 bytes) transmitted via DNS query, indicating possible data exfiltration.



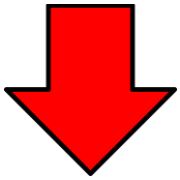
Step 7: DNS Tunneling Log – Query Length 512 (Moderate Payload Transfer)

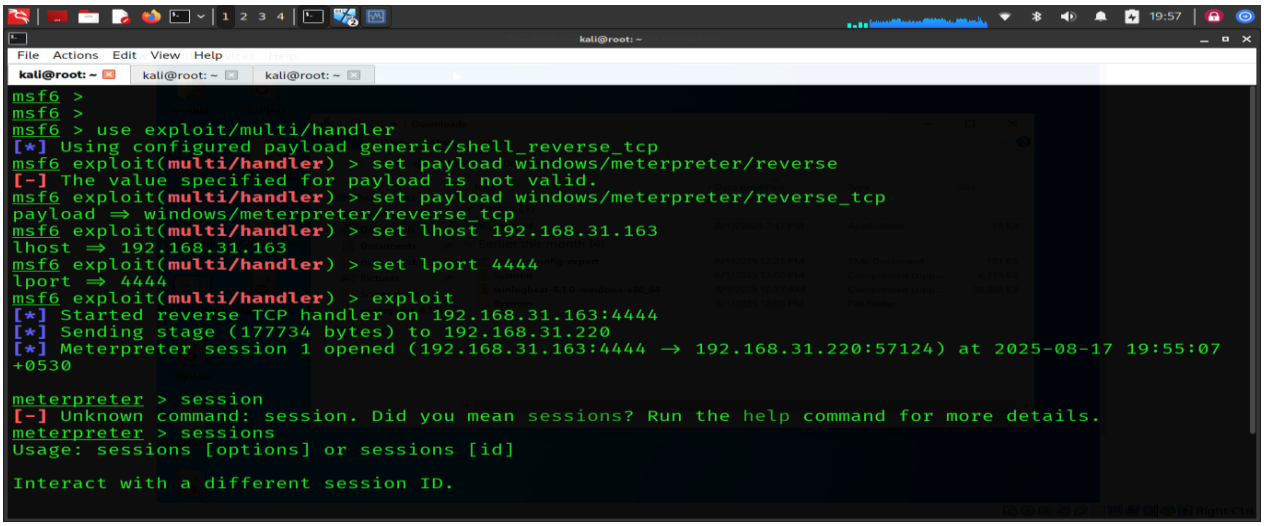
Medium-sized (512 bytes) DNS query detected, consistent with tunneling activity.



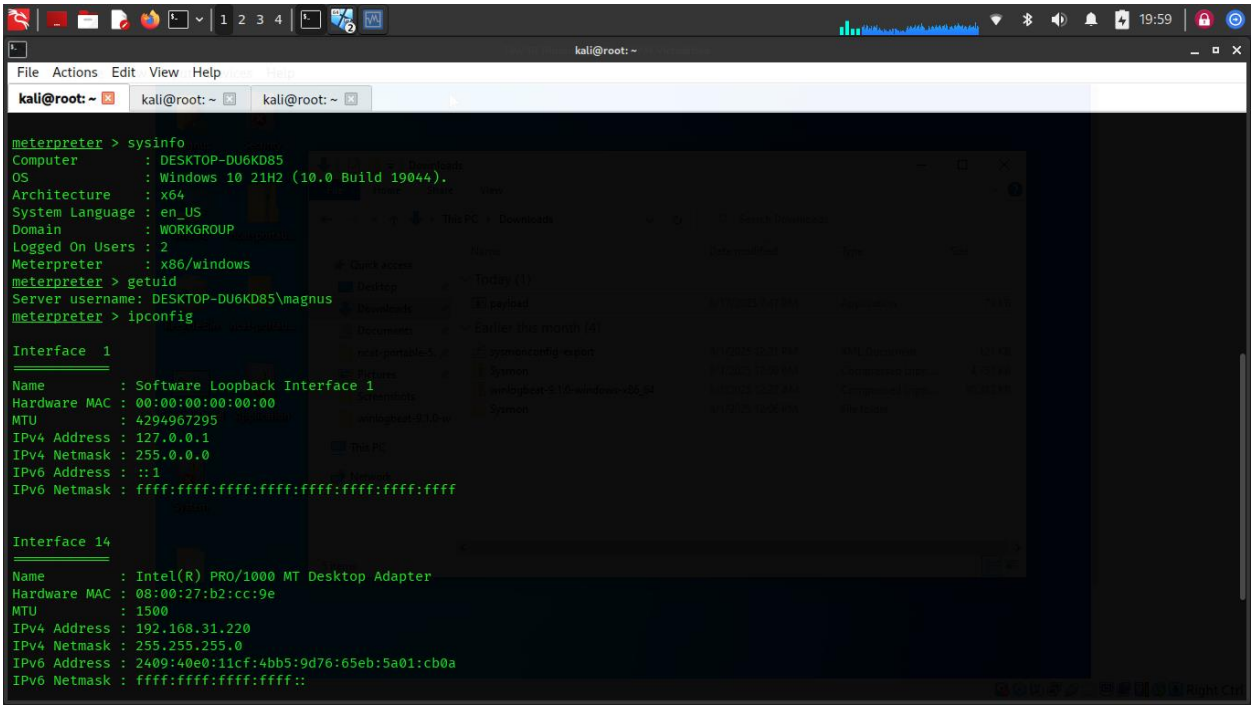
iii. Powershell Exploitation: PowerShell exploitation refers to the abuse of Windows PowerShell, a powerful command-line shell and scripting language, by attackers to gain initial access, privilege escalation, persistence, lateral movement, or data exfiltration within a target system/network.

Step 1: Got Initial Access in Win 10 by using a payload.exe which i made using metasploit. As view screenshot down

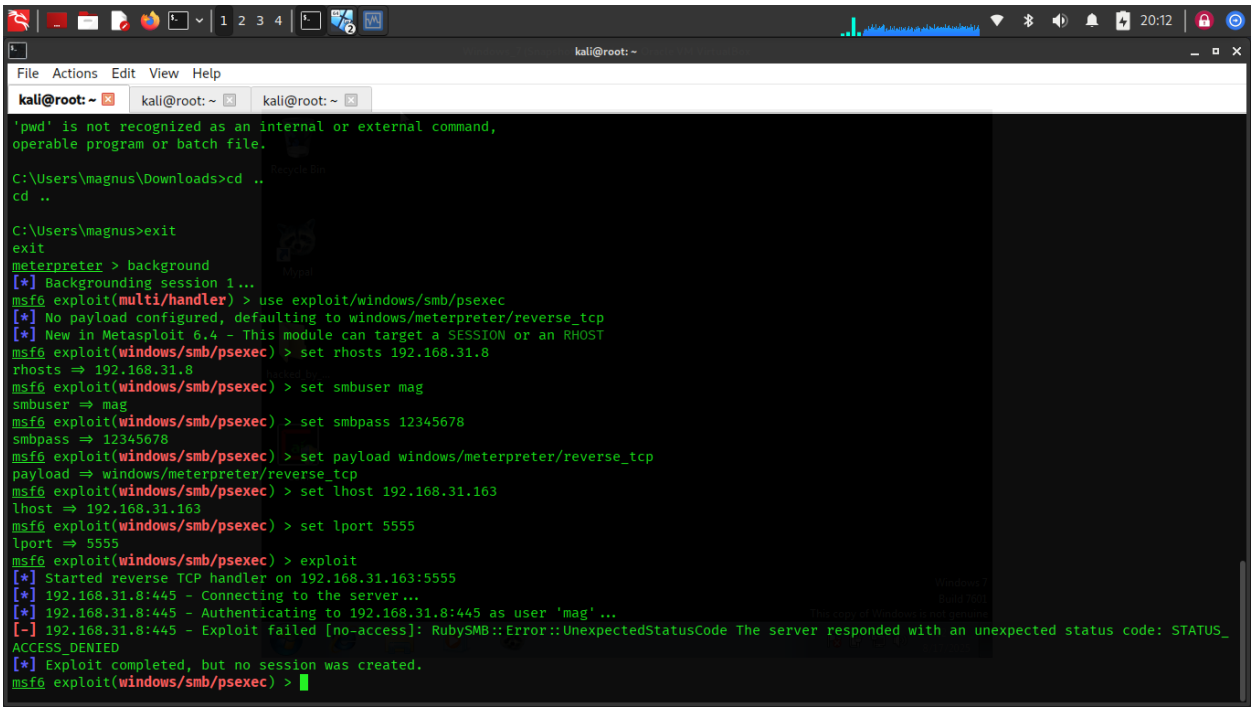




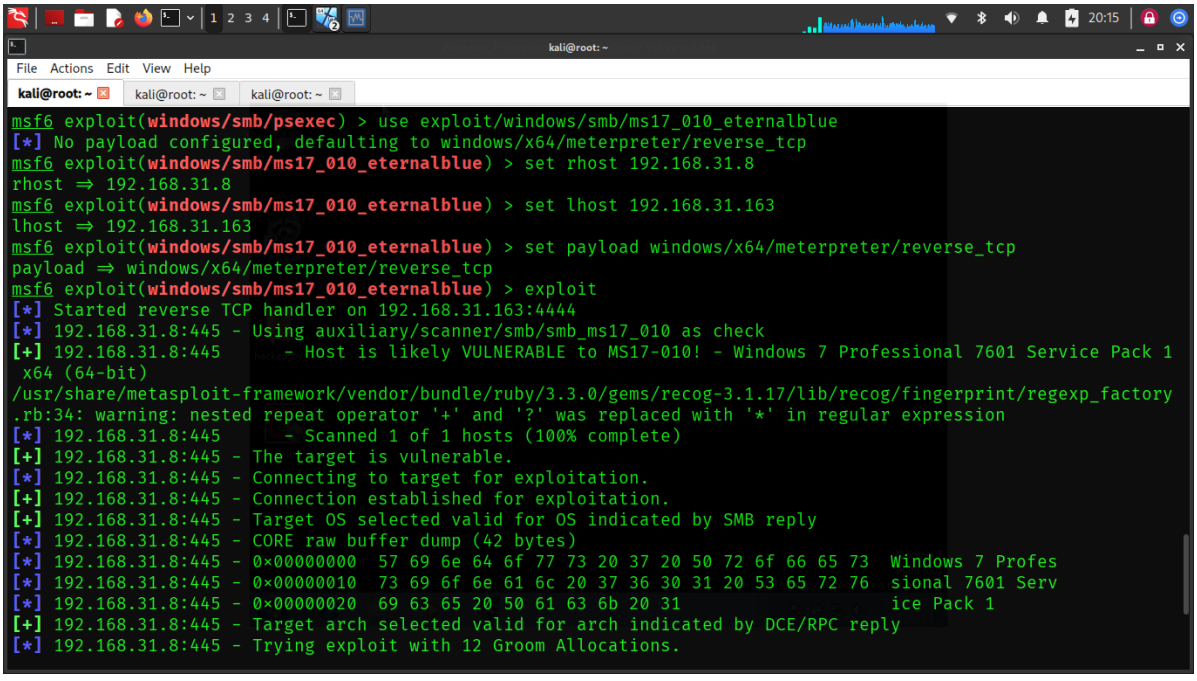
Step 2: Showing Windows 10 information by sysinfo command as we got the meterpreter session successfully.



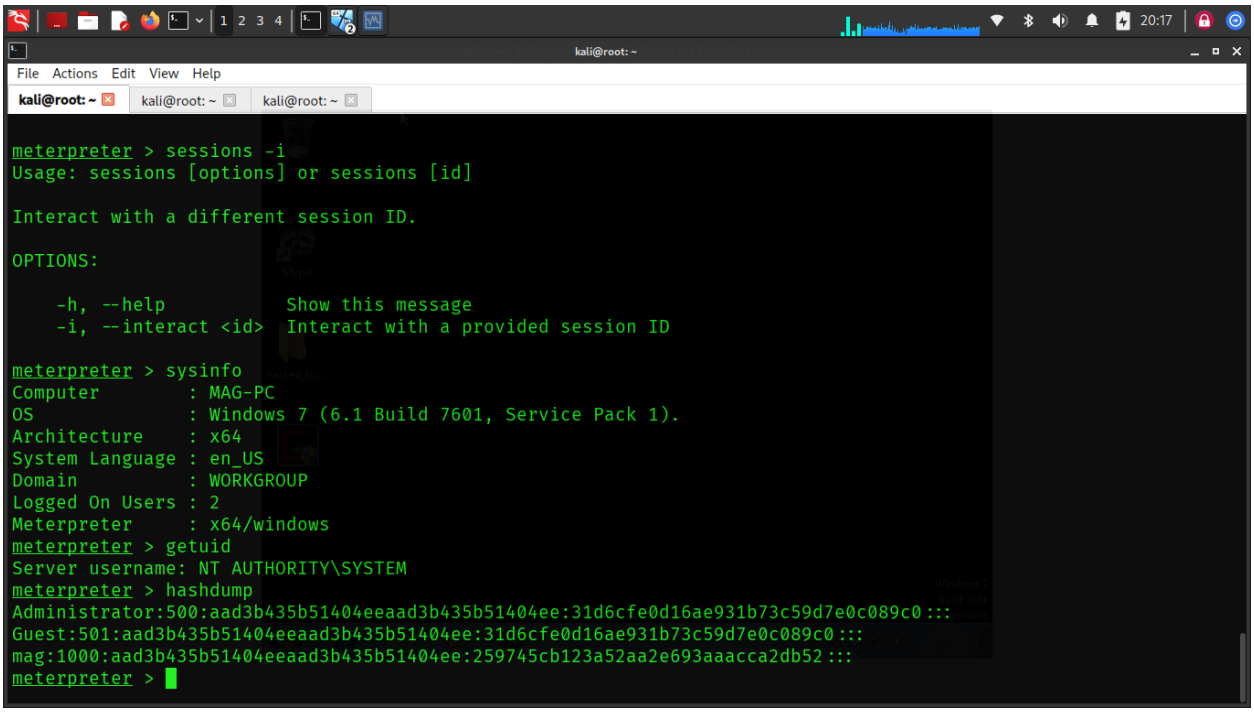
Step 3: Then i tried Lateral Movement between win 10 and win 7 using Psexec but it failed as which is captured in screenshot.

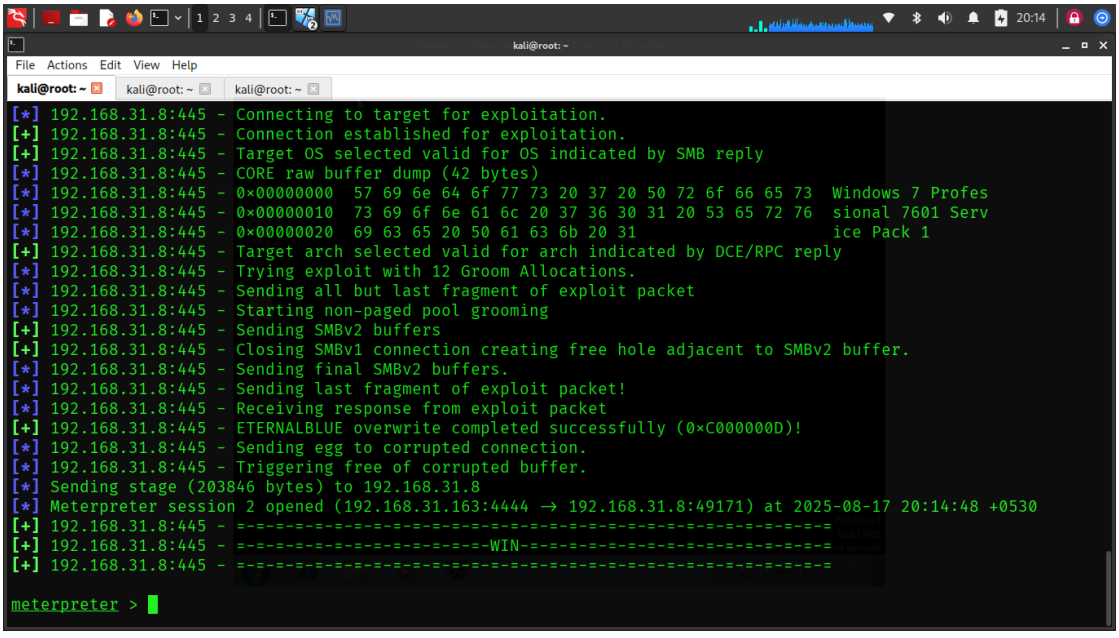


Step 4: Then in a new session I used Eternala Blue Vulnerability to gain access in Win 7 which is done in session 2.

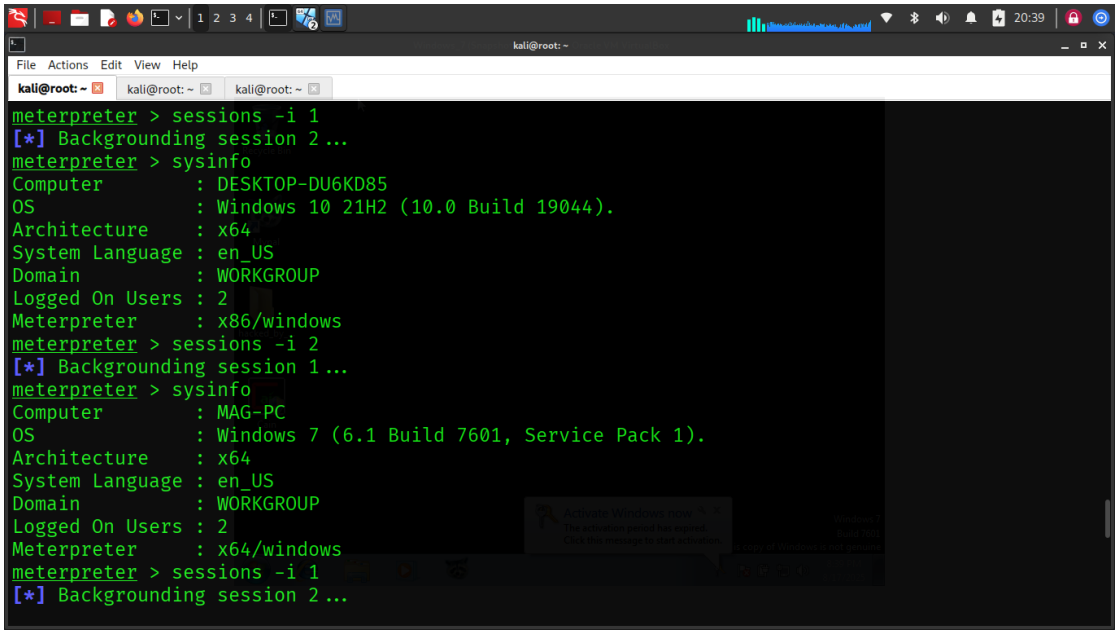


Step 5: This was when i got up the meterpreter session in Windows 7 and got the system information along with hashes and all



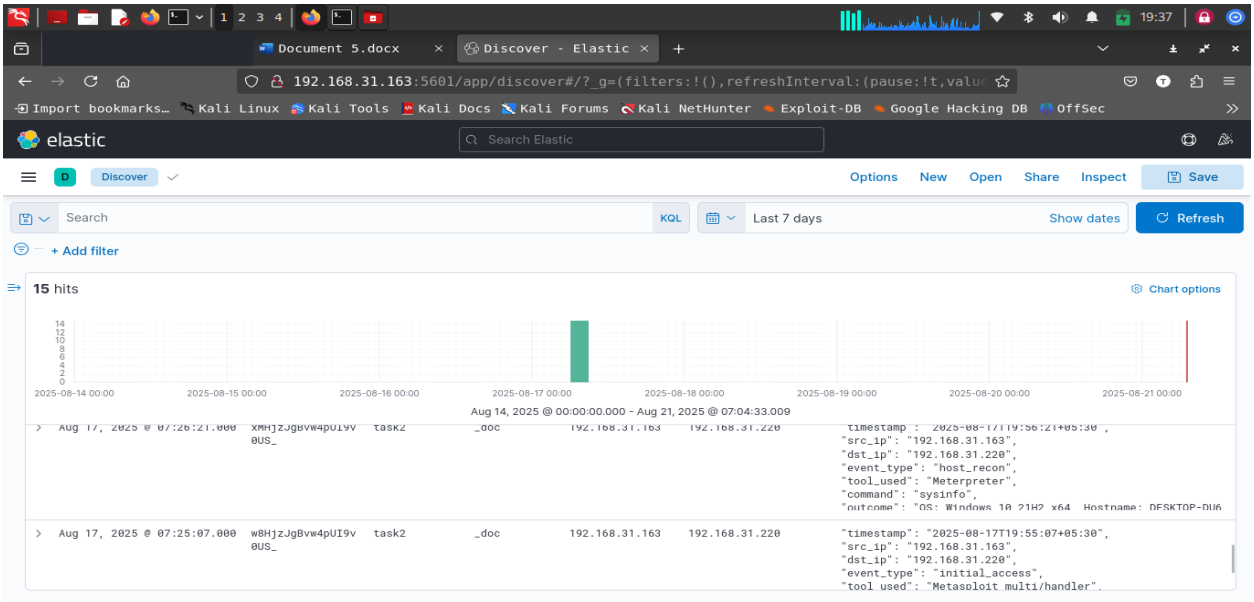


Step 6: This is when Lateral Moevement Achieved as two sessions are running in which Session 1 consist of Windows 10 and Session 2 consist of Windows 7. So here i can jump between sessions.

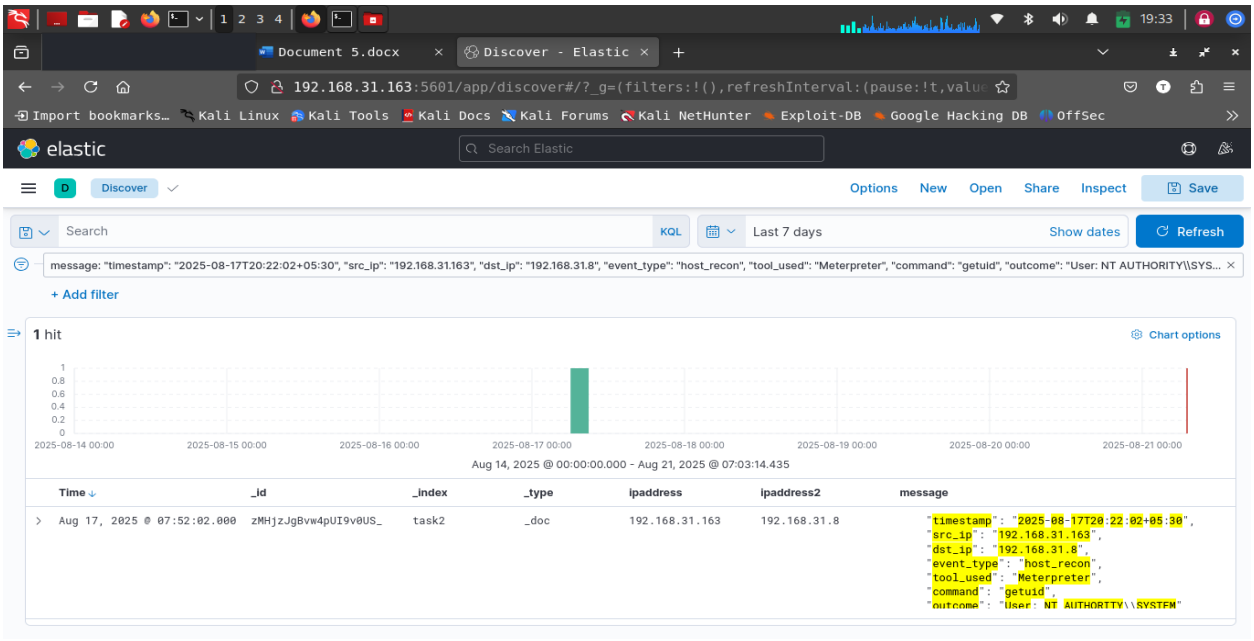


LOGS

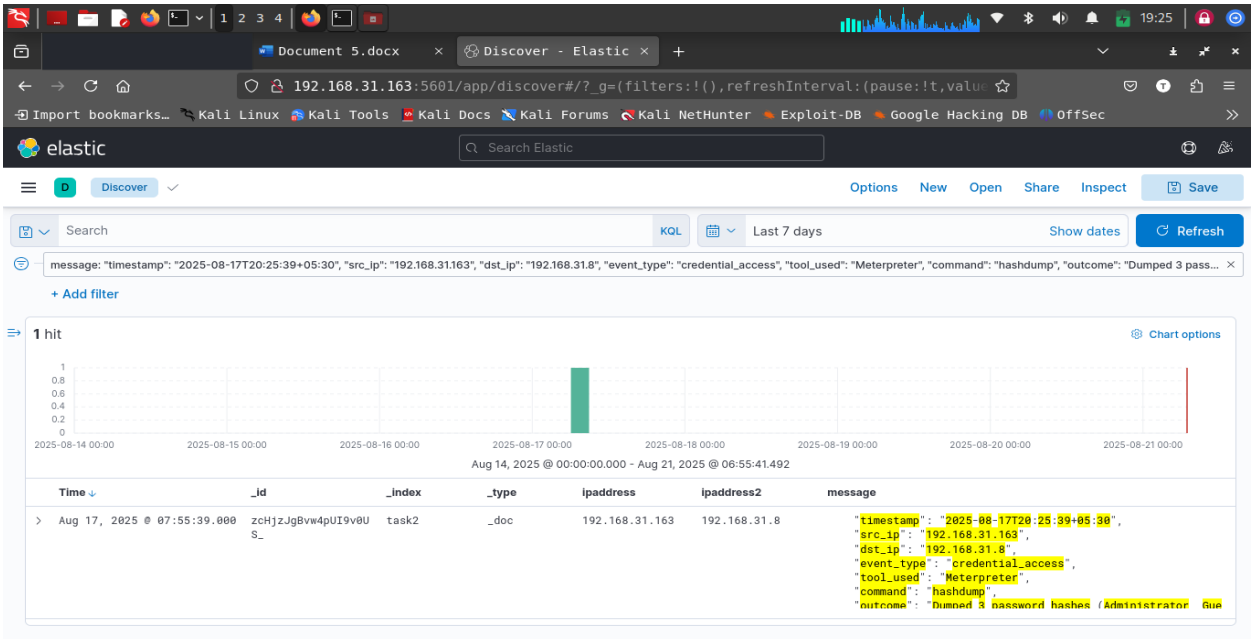
Got logs of Initial Access of Windows 10 using that payload.exe



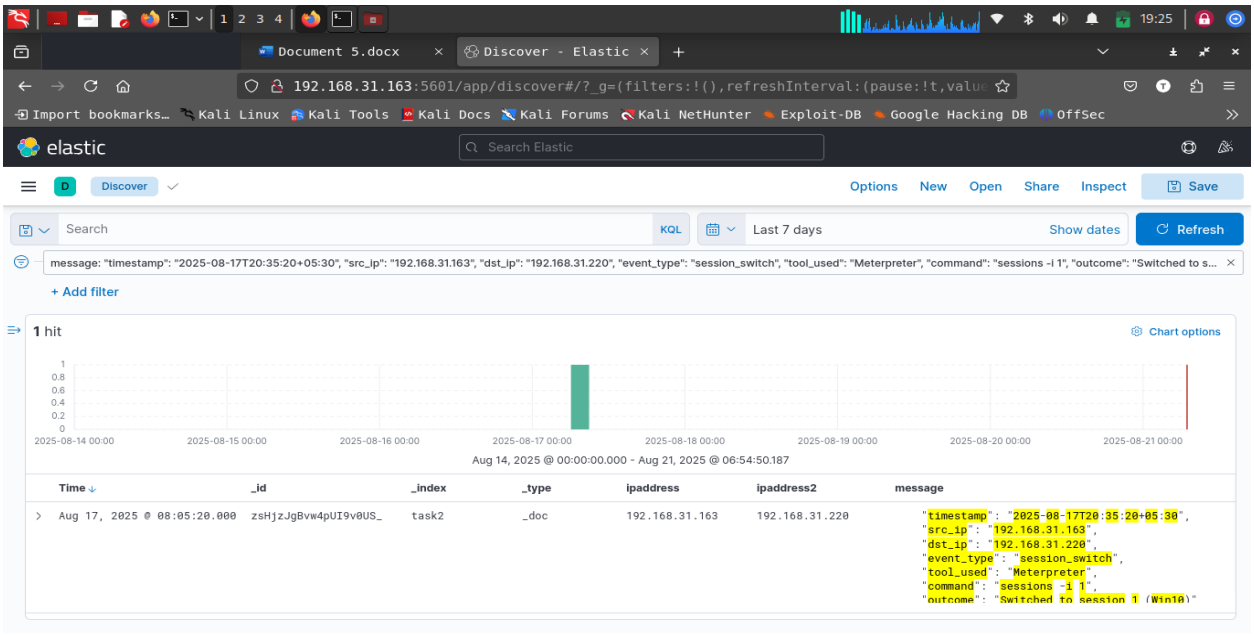
As used Eternal Blue Vulnerability to get into Windows 7 in session 2



Done Windows 7 hashdump which got logged.



Powershell Exploitation done by doing Lateral movement between two sesions as in which Session1 consist of Windows 10 and Session2 consist of Windows 7 so here i am in Session 1 Windows 10



Challenges and Solutions

1. Credential Stuffing

Challenge:

While simulating credential stuffing with Hydra against SMB login, multiple failed authentication attempts were captured in Kibana. However, the **successful 9th login attempt was not logged** in Elasticsearch, leading to incomplete visibility.

Solution:

Enhanced logging rules were tuned in the SIEM to ensure both **failed and successful authentication events** are captured. This gap highlighted the importance of correlation rules that track login attempts holistically (failed + successful) to detect brute-force/credential stuffing activity effectively.

2. DNS Tunneling

Challenge:

Detecting DNS tunneling was tricky because malicious traffic was **hidden inside normal DNS queries**. The volume of legitimate DNS requests created noise, making it hard to differentiate between benign and malicious queries.

Solution:
Developed custom correlation rules in the SIEM to detect:

- Unusually **high frequency of DNS queries** to a single domain.
- Presence of **suspicious encoded strings** (base64-like patterns) within DNS requests.
  This helped reduce false positives and allowed DNS tunneling detection with higher accuracy.

3. PowerShell Exploitation (Lateral Movement)

Challenge:
While performing exploitation on Windows 10 using a payload, **PSExec failed** for lateral movement to Windows 7, restricting attacker pivoting. The challenge was to achieve reliable movement across machines.

Solution:
As an alternative, **EternalBlue exploit** was used to compromise Windows 7. After gaining a new session, successful **lateral movement** was achieved — enabling interaction with both sessions (Win10 and Win7) simultaneously. This exercise demonstrated how attackers adapt by chaining multiple exploitation techniques when direct tools fail.

Outcomes and Impact

1. Improved SIEM Visibility

The exercise highlighted both the **strengths and limitations** of SIEM logging. During credential stuffing, failed attempts were logged correctly, but the successful login attempt was missed. This gap demonstrated the need to refine event ingestion and correlation rules. As a result, visibility into both successful and failed authentications was improved, reducing the chance of an attacker bypassing detection after brute-forcing credentials.

2. Enhanced Detection Capabilities

By creating and testing **custom correlation rules**, the SIEM was able to detect:

- Abnormal login patterns (credential stuffing).
- Suspiciously frequent DNS queries that matched tunneling behavior.
- Indicators of malicious PowerShell exploitation and privilege escalation.

This proved the effectiveness of tuning detection logic to catch not only known attack signatures but also behavioral anomalies.

3. Realistic Adversary Simulation

The hands-on activity provided a practical view of how adversaries adapt when initial methods fail. For instance, when PSExec did not work for lateral movement, EternalBlue was successfully leveraged to compromise the second host. This demonstrated the **flexibility of real-world attackers** and emphasized the importance of layered detection across different tactics and techniques.

4. Strengthened Defensive Posture

Through testing these attack scenarios, the organization gains actionable insights:

- Which log sources require enrichment.
- Where detection rules need fine-tuning.
- How attackers can move laterally if initial defenses are bypassed.

The impact of this work is a **stronger, more resilient detection environment**, better prepared to handle credential abuse, covert tunneling, and PowerShell-based exploitation attempts.

Conclusion

This task provided a valuable opportunity to simulate real-world attack techniques and evaluate the effectiveness of defensive measures in detecting and responding to them. By working through scenarios such as credential stuffing, DNS tunneling, and PowerShell exploitation with lateral movement, I was able to replicate the tactics an adversary might use and assess how well the SIEM environment captured and correlated those events.

The exercise highlighted both the strengths and gaps in log visibility—such as the missed detection of a successful brute-force attempt—which underscored the importance of refining event collection and correlation logic. At the same time, the ability of the SIEM to identify abnormal behaviors across authentication, DNS activity, and PowerShell execution confirmed that properly tuned detection rules can significantly strengthen security monitoring.

Overall, this activity not only enhanced my technical understanding of adversary tradecraft but also reinforced the critical role of continuous tuning and validation in building a resilient detection strategy. The knowledge gained here will directly support better security operations by ensuring that potential threats are identified earlier and responded to more effectively.