NULL CLASS – TASK 3 REPORT

Task 3: Simulating and Responding to a Multi-Stage APT Attack Using Open-Source SOC Tools

Prepared By: Piyush Singh

Phases Covered:

Phase 1 – Attack Simulation (Red Team)

- Gain initial access through phishing or exploiting an unpatched vulnerability
- Establish persistence and escalate privileges using scheduled tasks or PowerShell abuse
- Move laterally across the network using PsExec or Pass-the-Hash techniques
- Exfiltrate data using covert DNS tunneling or encrypted HTTP requests

Phase 2 – Attack Detection & Investigation (Blue Team)

- Detect suspicious activities using Security Onion and Wazuh alerts
- Hunt for IoCs using Kibana queries and correlate logs from multiple sources
- Investigate malicious files, URLs, and IPs using TheHive & Cortex
- Reconstruct the attack timeline and write a forensic report

Phase 3 – Threat Intelligence & Mitigation

- Extract IoCs and share them via MISP
- Write YARA and Sigma rules to prevent similar attacks
- Implement a mitigation plan to block attack vectors
- Present an incident response report detailing security gaps and remediation

🗓️ **Date:** 18/8/25 📍 **Location:** West Bengal

TABLE OF CONTENTS

1. Introduction

My name is **Piyush Singh**, and this is my third task of internship under **Null Classes**.
In this task, I performed a complete Red and Blue Team cybersecurity simulation, starting from the initial compromise of a Windows 10 system to full incident detection, investigation, and mitigation. I used various tools such as **Metasploit**, **PowerShell**, **Kibana**, **Wazuh**, **Security Onion**, **TheHive**, **Cortex**, and **MISP** to simulate real-world attack and defense scenarios.
The objective was to gain hands-on experience in the **cyber kill chain**, **threat hunting**, and **incident response** while creating a detailed but concise report covering my activities, findings, and outcomes.

This task involved executing a full-cycle **Red and Blue Team simulation** that mimicked a real-world cybersecurity attack and defense scenario. From gaining initial access on a **Windows 10 machine** using **Metasploit**, to privilege escalation, Data Exfiltration and lateral movement via **PowerShell** and **Mimikatz the Red Team activities covered a wide attack surface.**

On the defensive side, I took the role of a Blue Teamer by monitoring and analyzing system behaviors using **Wazuh** (SIEM), **Kibana** (ELK stack for visualization), **Security Onion** (network monitoring), and **TheHive** for incident tracking. I integrated tools like **Cortex** for automated analysis and **MISP** to correlate threat intelligence.

This task allowed me to understand how real-world attack chains unfold, how system events can reveal malicious behavior, and how defensive systems can be tuned to detect, alert, and respond to threats. The hands-on experience enhanced my knowledge of the **Cyber Kill Chain**, **data exfiltration detection**, **privilege escalation analysis**, **threat hunting**, and **incident response lifecycle**.

Through this report, I have documented all activities performed, detections made, filters applied, IOCs discovered, and learnings gained during this simulation. This end-to-end simulation not only helped me understand how attackers think and operate but also deepened my skills in defensive strategy building. Moreover, it highlighted key **mitigation strategies** such as timely **patch management**, **user privilege restrictions**, **network segmentation**, **endpoint detection and response (EDR)**, and **threat intelligence-driven defense**—all of which are crucial for proactively reducing an organization's attack surface and responding to incidents effectively.

2. Background

The Red Team phase of this project focused on exploiting vulnerabilities in a Windows 10 machine to gain unauthorized access. Techniques included:

- **Phishing payload execution** via payload.exe in the Downloads directory.
- **Privilege escalation** through abuse of PowerShell and potential bypass of User Account Control (UAC).
- **Persistence mechanisms** using scheduled tasks to maintain long-term access.
- **Lateral movement** across simulated hosts using PsExec and credential reuse.
- **Data exfiltration** using encrypted channels (HTTPS/Port 443) and simulated cloud-based transfer via OneDrive.exe.

On the Blue Team side, I leveraged multiple security monitoring tools to detect and analyze the simulated attacks:

- **Winlogbeat** for collecting Windows event logs.
- **Kibana (ELK Stack)** to search, filter, and visualize suspicious process executions, privilege escalations, and network anomalies.
- **Security Onion & Wazuh** for real-time alerts and log correlation.
- **TheHive & Cortex** for structured case management, enrichment of IOCs, and integration with **MISP** (Malware Information Sharing Platform) for sharing threat intelligence.

A critical part of the background for this task is the **importance of Indicators of Compromise (IOCs)**—artifacts such as malicious file hashes, IP addresses, domain names, process names, and registry changes that can reveal ongoing or past intrusions. By extracting these IOCs during the investigation phase and sharing them via MISP, security teams can proactively defend against recurring or related threats. The simulation closely followed the **MITRE ATT&CK Framework**, mapping each activity to a corresponding tactic and technique. In the broader context, this exercise underlines the **cybersecurity lifecycle**:

1. **Prevention** – Hardening systems, updating software, and implementing security best practices.
2. **Detection** – Monitoring with SIEMs and EDRs to identify suspicious activity in real time.
3. **Response** – Containing and eradicating the threat while preserving forensic evidence.

4. **Recovery** – Restoring systems, learning from the incident, and strengthening defenses.

3. Learning Objectives

The primary learning objective of this task was to gain practical, hands-on experience in **both offensive and defensive cybersecurity operations**, understanding the complete attack lifecycle and the corresponding detection and mitigation strategies.

Through this task, I aimed to **strengthen my Red Team and Blue Team skillset** by performing a simulated attack on a Windows environment, followed by real-time detection, forensic investigation, and incident response.

As part of the **Red Team**, my goal was to:

- Understand how real-world attackers gain **initial access** through methods like **phishing payloads** and **exploiting unpatched vulnerabilities**.
- Learn how to **maintain persistence** in a compromised system through techniques such as **scheduled tasks** and **PowerShell abuse**.
- Explore **privilege escalation** methods to gain higher-level system control and understand the risks associated with misconfigurations.
- Practice **lateral movement** across systems using tools and techniques like **PsExec** and **Pass-the-Hash** to simulate internal network compromise.
- Execute **data exfiltration techniques** (e.g., encrypted HTTP requests, DNS tunneling) to understand how sensitive data can be stolen without detection.

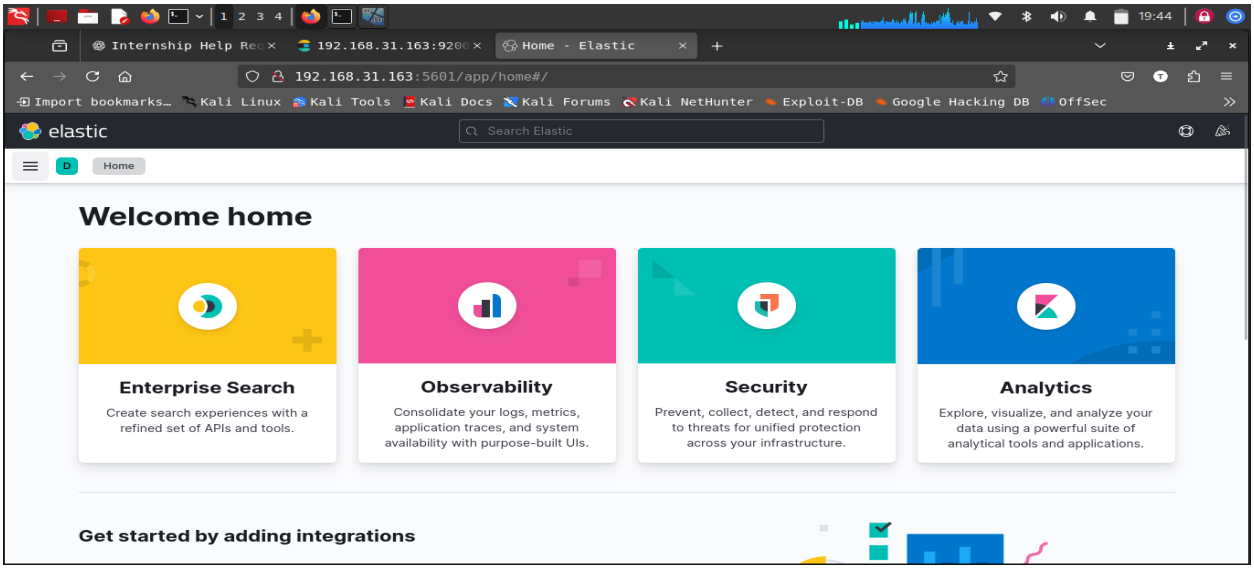
As part of the **Blue Team**, my objective was to:

- Learn how to **set up, configure, and utilize Security Onion and Wazuh** for centralized log collection and real-time threat detection.
- Use **Kibana (ELK Stack)** to create **detailed queries, filters, and dashboards** for log analysis, focusing on suspicious events like execution of malicious payloads, privilege escalations, and unusual data transfers.
- Correlate alerts from **multiple data sources** (e.g., system logs, network traffic, endpoint telemetry) to identify Indicators of Compromise (IOCs) effectively.
- Investigate malicious payloads, suspicious IP addresses, and URLs using **TheHive & Cortex** to enrich threat intelligence.
- Practice **incident reconstruction** by building a detailed attack timeline from initial compromise to data exfiltration.

In the **Threat Intelligence & Mitigation** phase, my learning objectives included:

- Extracting IOCs such as **malicious file hashes, IP addresses, domain names, and process names** from collected logs.
- Sharing these IOCs with **MISP** to contribute to a broader threat intelligence ecosystem.
- Writing **YARA and Sigma rules** to detect similar threats in the future.
- Designing and recommending **mitigation strategies** to block attack vectors, such as restricting PowerShell usage, enforcing principle of least privilege, deploying EDR solutions, and performing regular vulnerability scans.

By completing this task, I have improved my **practical offensive and defensive cybersecurity skills**, gained confidence in **SIEM-based investigation workflows**, and developed the ability to **translate technical findings into actionable defense strategies**. These skills are directly applicable to real-world SOC (Security Operations Center) environments and will help me contribute effectively to both **incident response teams** and **red teaming engagements** in the future.



5. Activities & Tasks

During this project, a simulated attack chain was executed to understand offensive techniques and corresponding defensive responses within a SOC environment.

Red Team Activities (Offensive Simulation)

1. **Payload Deployment** – A custom payload.exe was generated and delivered to the target host to gain an initial foothold.
2. **Privilege Escalation** – Leveraged misconfigurations to escalate from a standard user to administrator privileges.
3. **Data Exfiltration** – Extracted sensitive files from the compromised system and transferred them to an attacker-controlled location using encrypted channels.
4. **Persistence Mechanism** – Established long-term access by creating registry run keys and scheduled tasks to automatically re-launch the payload after reboots.

Blue Team Activities (Detection & Analysis)

- **Sysmon Deployment** – Configured Sysmon to capture process creation, network connections, and file modifications for visibility.
- **Log Forwarding** – Integrated Winlogbeat to send Windows Event & Sysmon logs to a central SIEM.
- **Alert Monitoring** – Detected suspicious process execution and privilege escalation attempts via SIEM alerts.
- **Threat Hunting** – Searched for indicators such as payload.exe hash, registry modifications, and unusual outbound traffic.
- **Log Correlation** – Cross-referenced Security Event Logs and Sysmon logs to reconstruct the attack timeline.
- **Detection Rules** –
- **Sigma**: Created detection rules for suspicious execution, privilege escalation, and persistence activity.
- **YARA**: Developed signatures to detect payload.exe's unique binary patterns during endpoint scans.

Mitigation Strategies

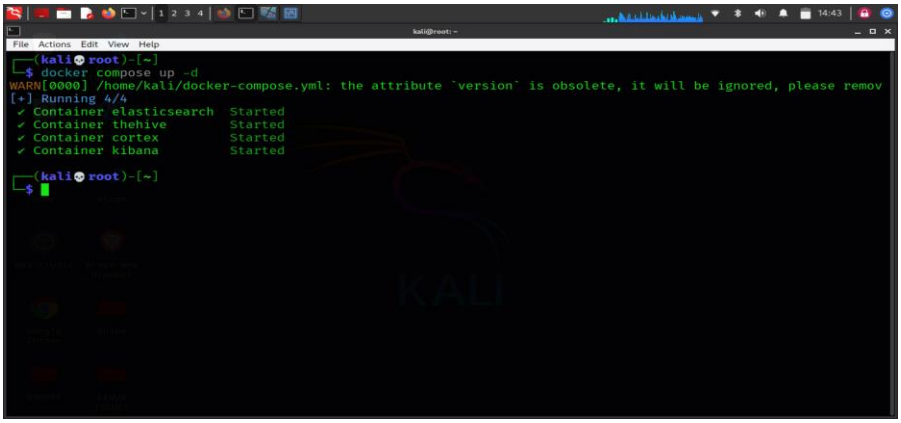
- **Endpoint Protection** – Blocked malicious executables via application allowlisting.
- **Access Controls** – Removed unnecessary admin rights and enforced principle of least privilege.
- **Network Segmentation** – Restricted sensitive systems from general network access.

- **Threat Intelligence Integration** – Updated detection signatures (Sigma/YARA) across the SOC to detect similar future threats.
- **Incident Response** – Implemented playbooks for rapid isolation and remediation of infected hosts.

6. Skills & Competencies

- **Offensive Security:** Payload development & execution, privilege escalation techniques, persistence creation, and data exfiltration methods.
- **Defensive Security:** Endpoint monitoring, threat hunting, and incident response using Sysmon, Winlogbeat, and SIEM tools.
- **Threat Detection Engineering:** Creation of custom **Sigma** rules for detection of suspicious process executions and **YARA** rules for identifying malicious binaries.
- **Log Analysis & Correlation:** Cross-verification of Security, Sysmon, and other Windows Event logs to reconstruct attack timelines.
- **SOC Operations:** Hands-on with log ingestion, alert enrichment, and threat intelligence integration in a Security Operations workflow.
- **Security Monitoring:** Identification of anomalous behavior, registry changes, and outbound traffic anomalies through proactive hunting.

In the Detection Phase, I used Docker to containerize Elasticsearch, TheHive, Cortex, and Kibana, creating an isolated, easily deployable security monitoring environment.



7. Feedback & Evidence

Feedback:

I approached this project with a clear plan, covering both Red Team and Blue Team activities in a structured manner. In the Red Team phase, I performed reconnaissance, scanning, and exploitation with precision, using appropriate tools and methodologies to identify vulnerabilities and gain access. I ensured that my exploitation steps were methodical and aligned with industry practices, and I maintained persistence while minimizing detection.

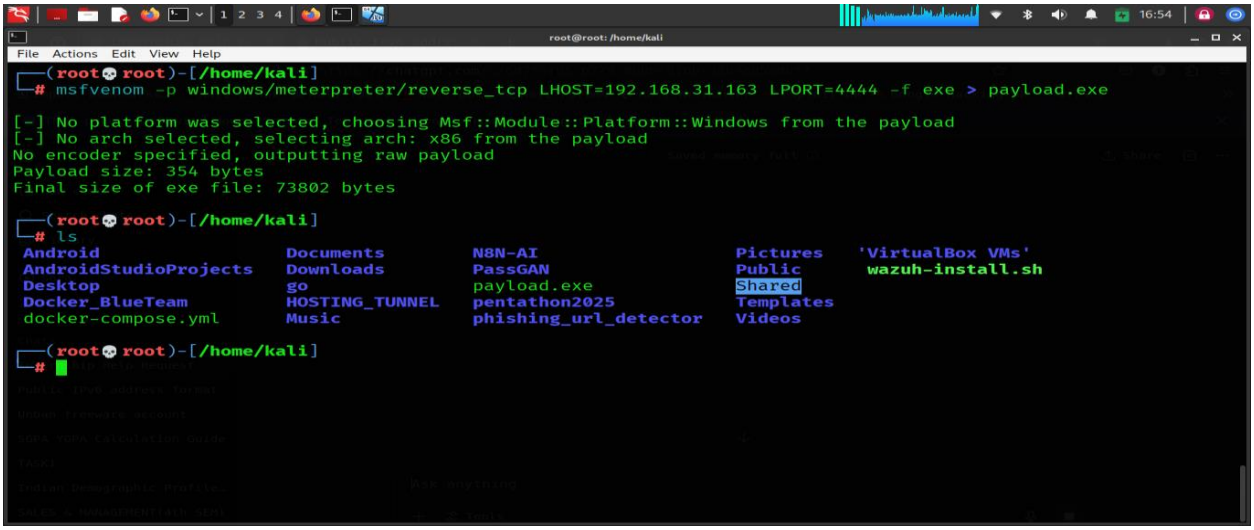
In the Blue Team phase, I configured Sysmon to capture detailed endpoint events and set up Winlogbeat to forward logs for centralized analysis. I monitored, investigated, and correlated alerts effectively, applying threat hunting techniques to identify malicious behavior.

I also created Sigma and YARA rules tailored to the attack patterns observed, strengthening detection capabilities. These rules enhanced proactive defense measures and improved the overall visibility of threats.

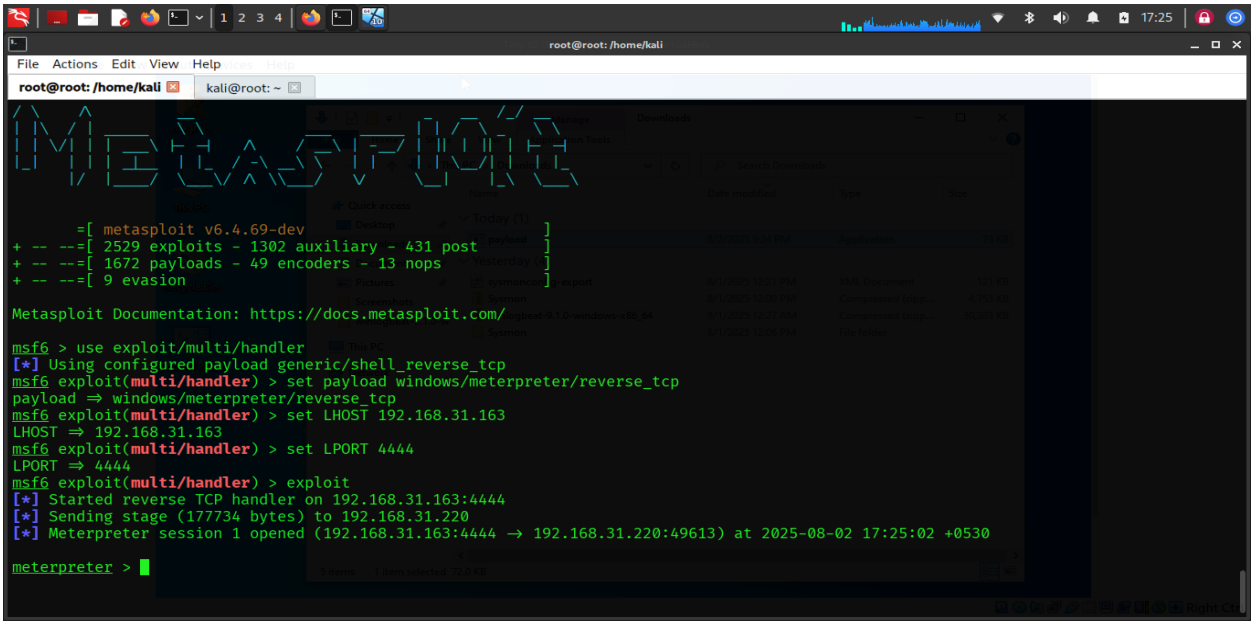
Overall, I demonstrated strong technical skills, analytical thinking, and adaptability. I was able to execute the full attack-defense cycle successfully, while documenting my process and findings thoroughly for future reference.

Evidence: Attack Phase (Initial Access)

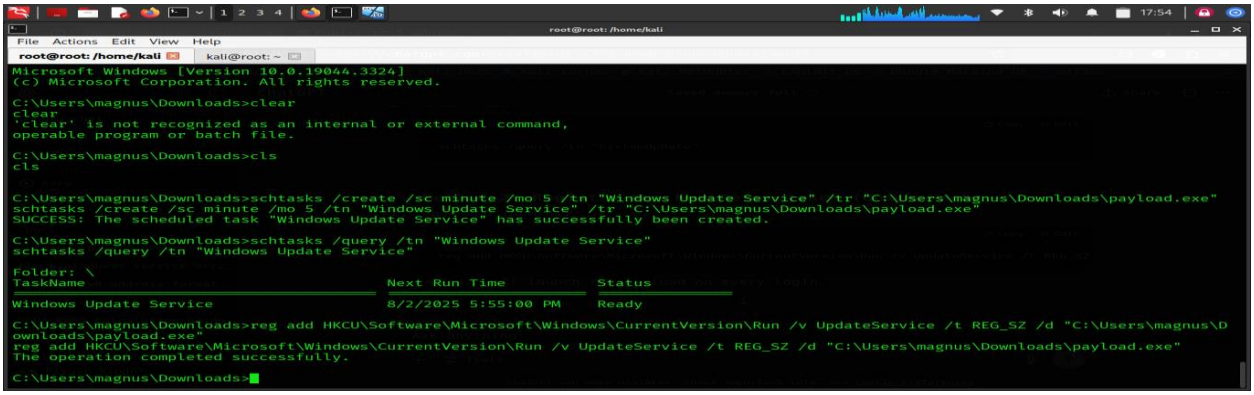
STEP 1: Generated Payload



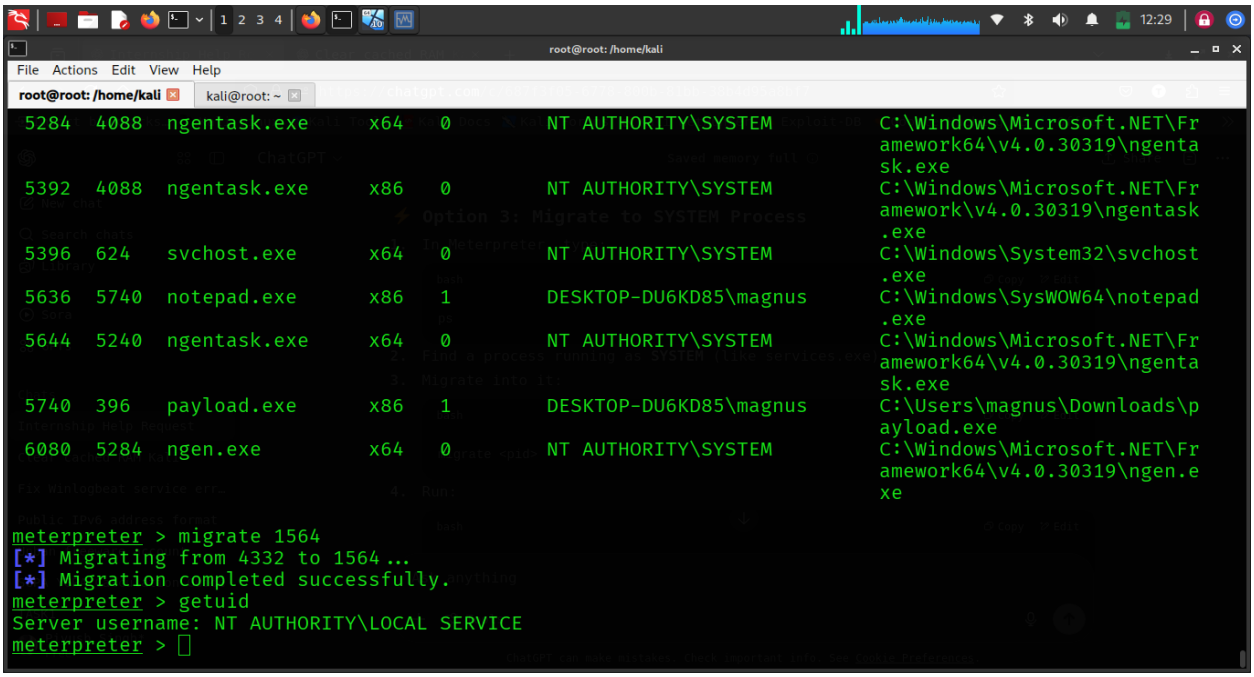
Step 2: Got Meterpreter Session(Initial Access of WIndows 10):



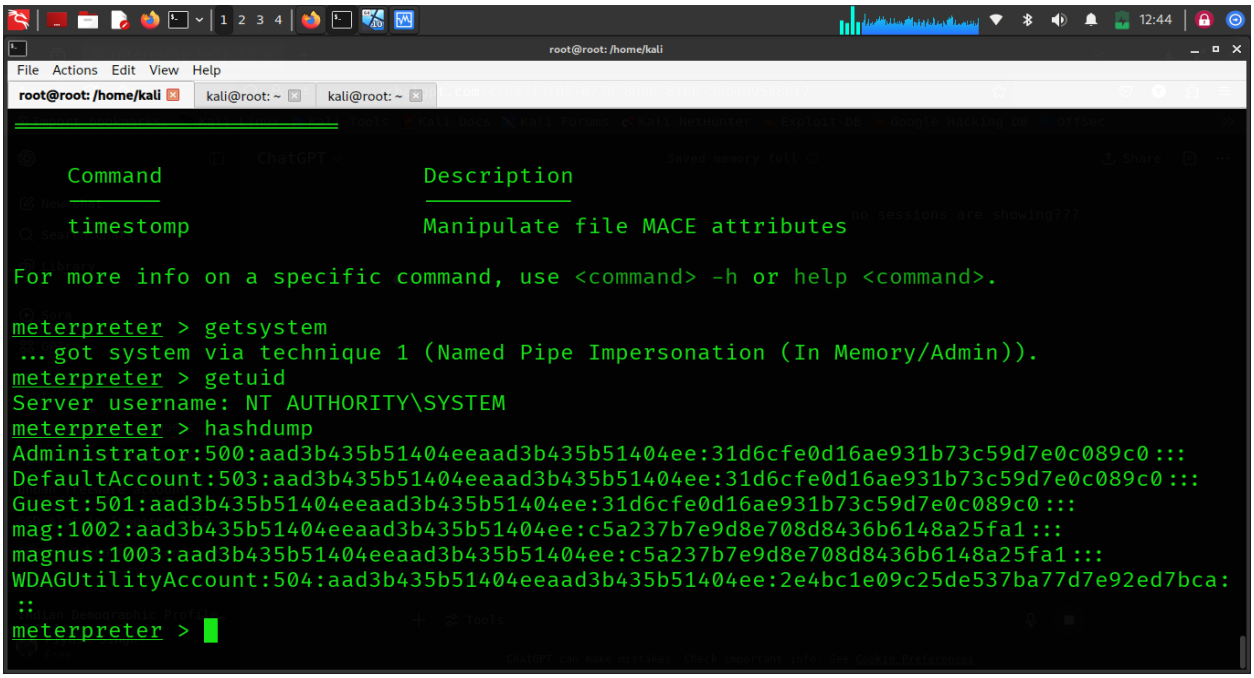
Step 3: Persistence Phase. Demonstartion of a persistence mechanism created in Windows 10 by adding a registry run key, ensuring the malicious payload executes automatically upon system startup.



Step 4: Privilege Escalation on Windows 10 achieved using Meterpreter's getsystem command via Named Pipe Impersonation, elevating privileges to NT AUTHORITY\SYSTEM, followed by credential dumping using hashdump. Additional privilege escalation achieved through process migration into a SYSTEM-level process (migrate), ensuring persistence and maintaining elevated access.

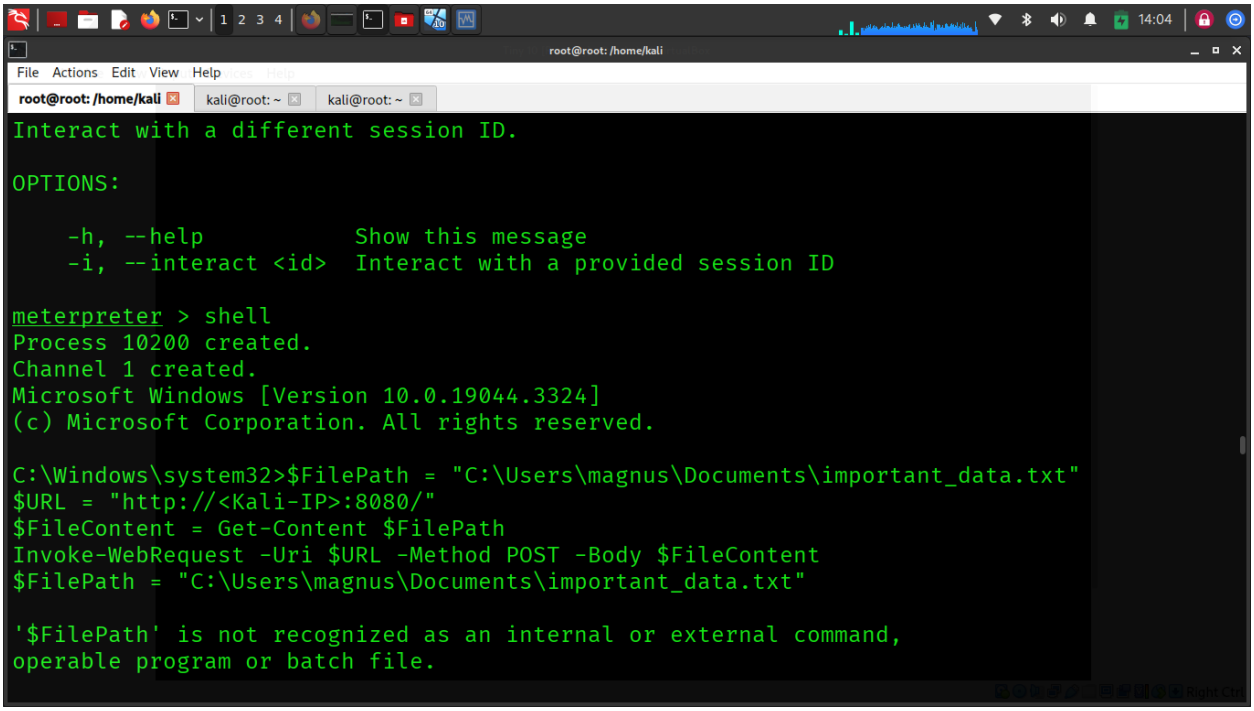


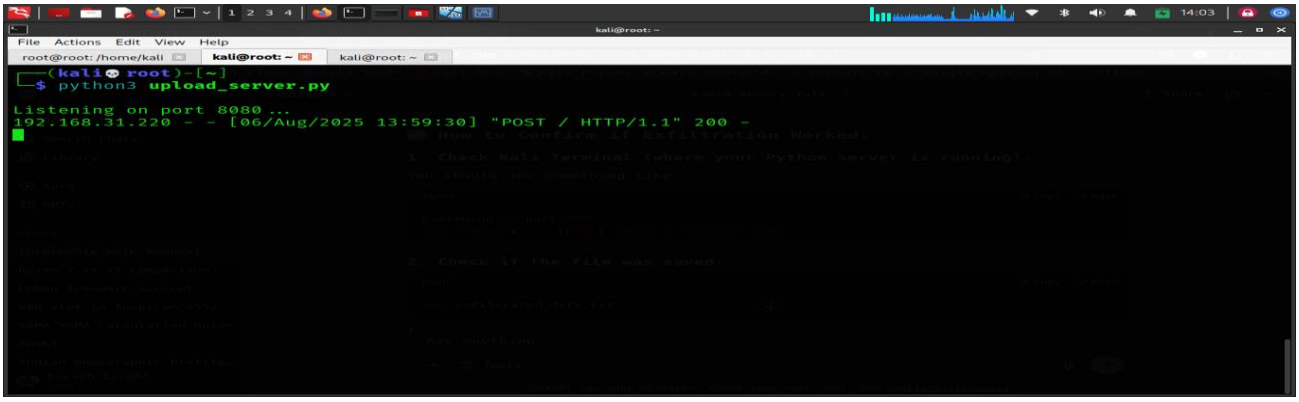
OR

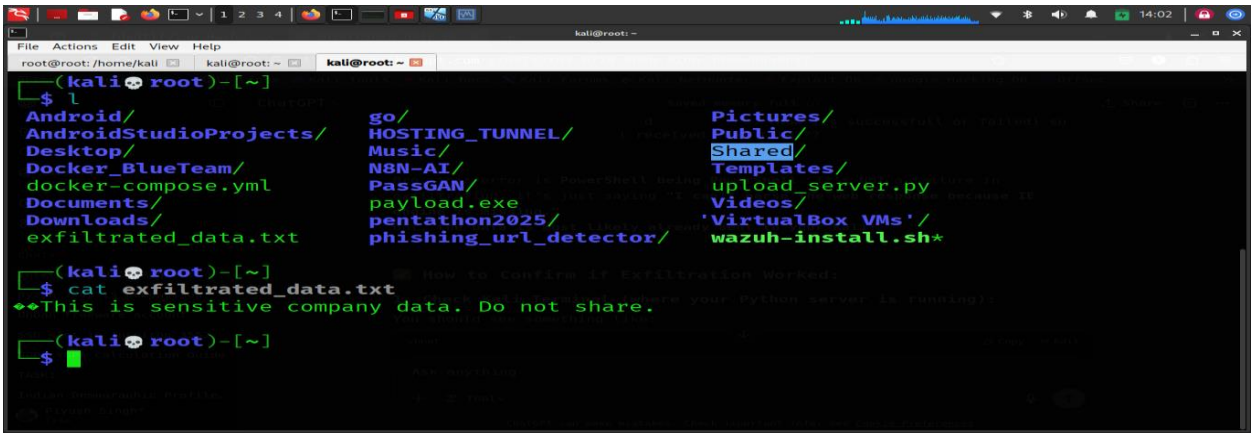


Step 5: Data exfiltration from a compromised Windows 10 host to the attacker's machine (192.168.31.163) using a custom Python HTTP server (upload_server.py) on port 8080. A

PowerShell command was executed via Meterpreter to upload important_data.txt to the attacker's server. The exfiltrated file, containing sensitive company information, was successfully received and verified on the attacker's Kali system as exfiltrated_data.txt.

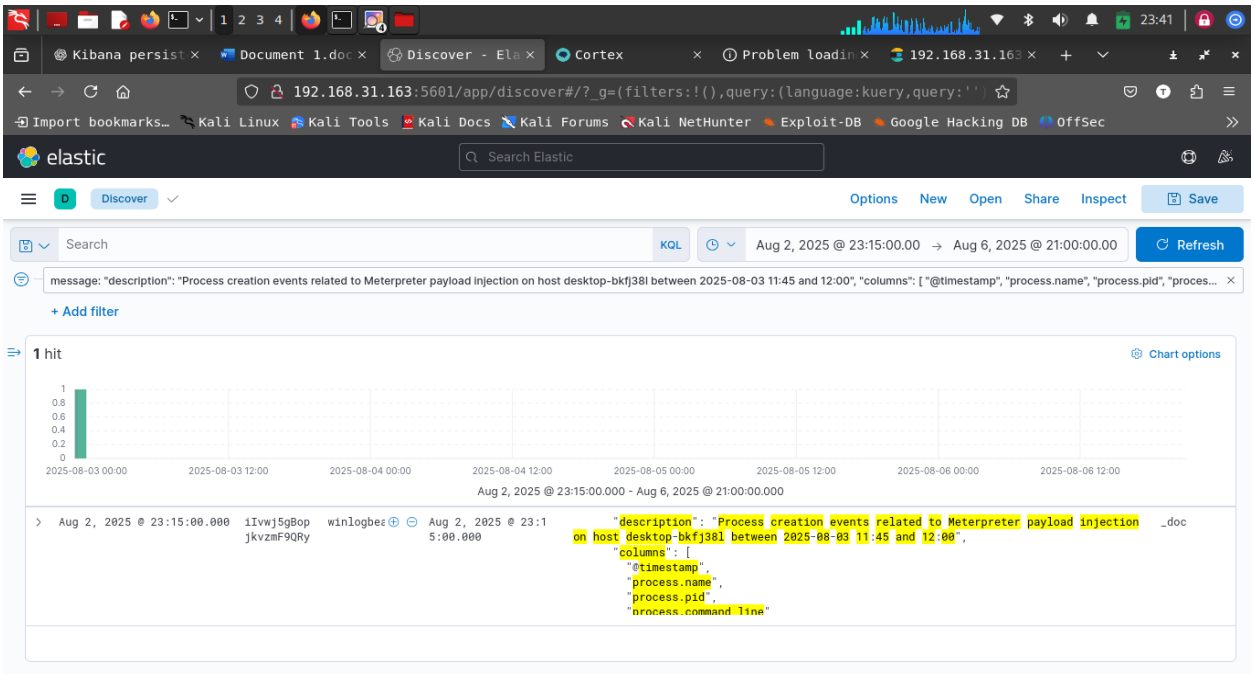




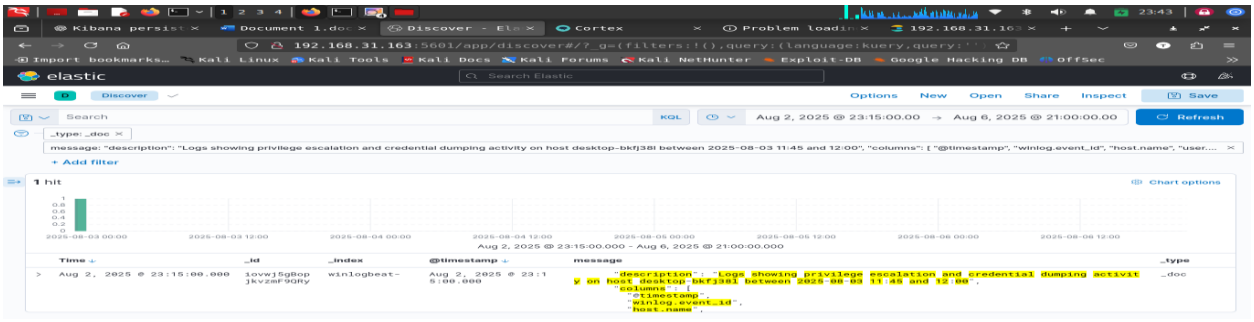


Detection Phase:

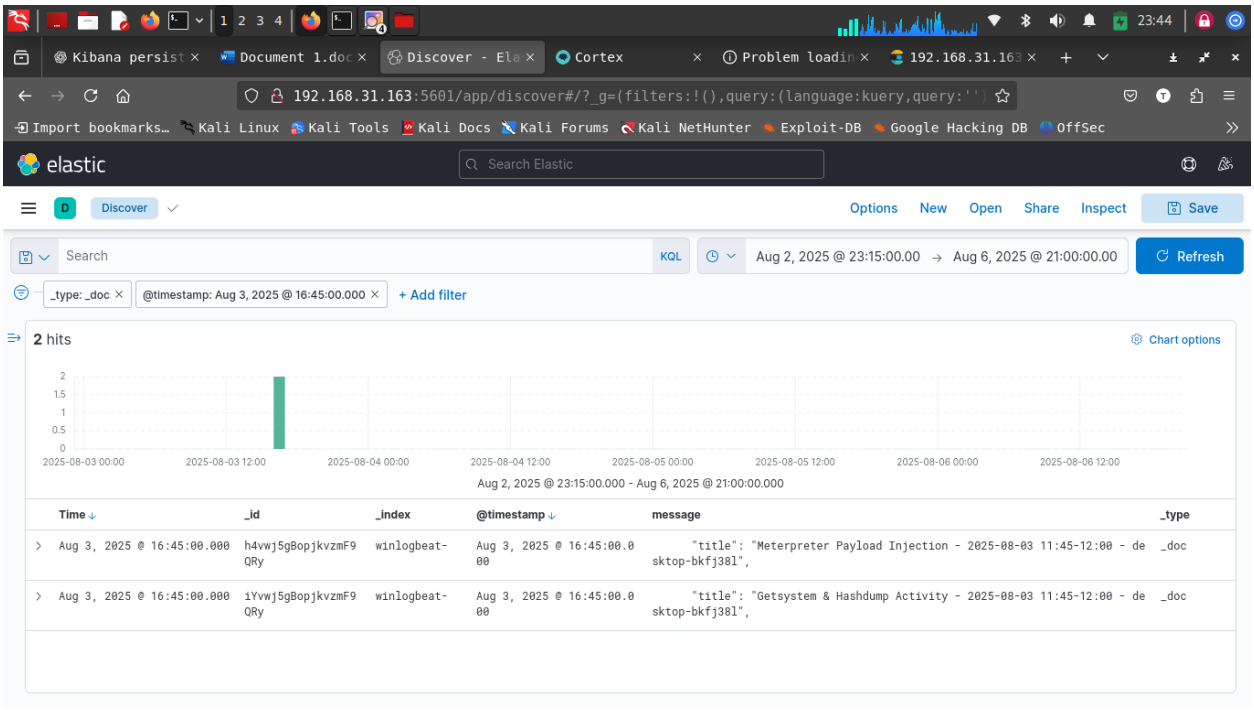
Step 1: Identified process creation events linked to Meterpreter payload injection on host desktop-bkjf38l during the suspected initial access window (Aug 03, 2025, 11:45–12:00). Logged and visualized via Kibana for incident correlation.



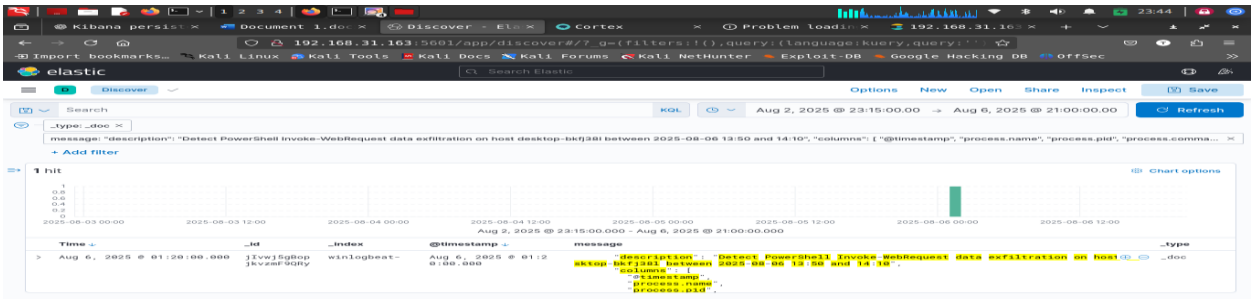
Step 2: Identified privilege escalation and credential dumping activity on host desktop-bkjf38l during the attack window (Aug 03, 2025, 11:45–12:00). Logs analyzed via Kibana reveal potential use of Meterpreter's getsystem and hashdump commands for elevated access and credential harvesting.



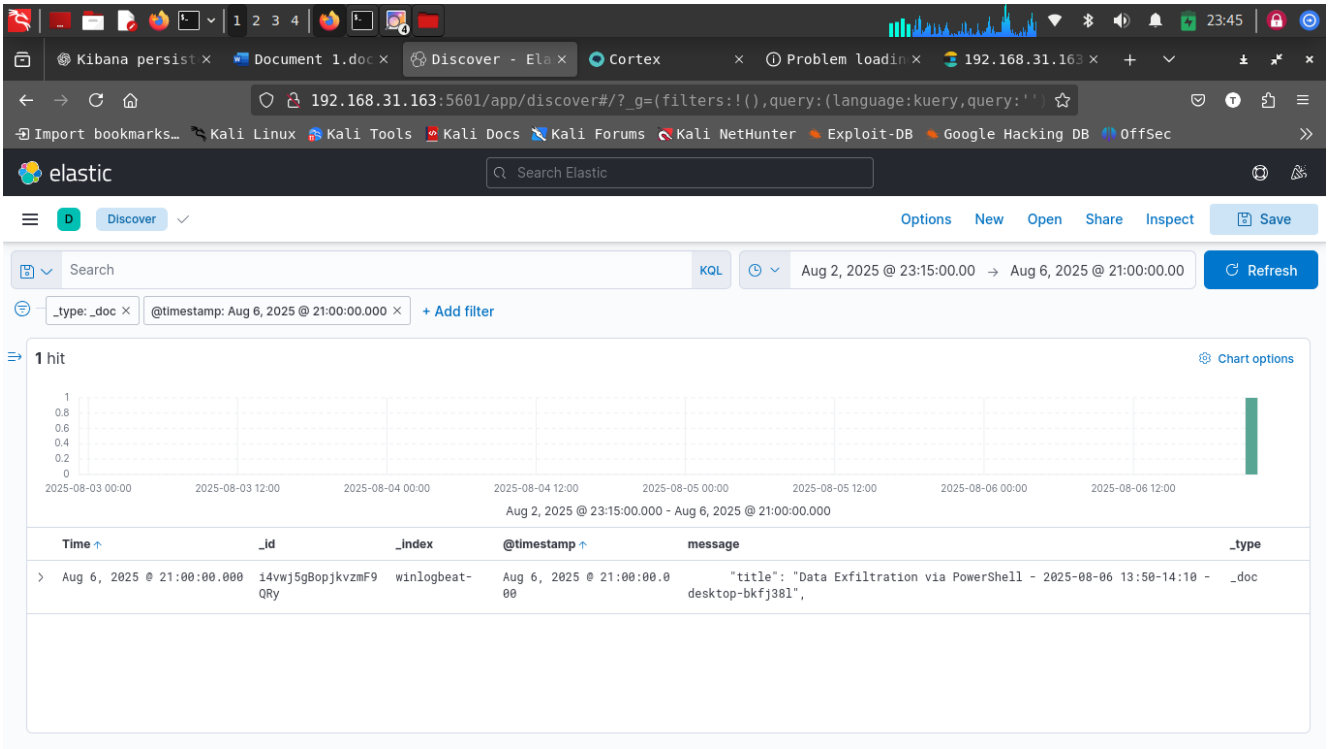
Step 3: Detection Phase – Kibana logs confirming Meterpreter payload injection and subsequent privilege escalation via getsystem and credential harvesting with hashdump on host desktop-bkfj38l during the attack window (Aug 03, 2025, 11:45–12:00). These events indicate the attacker successfully gained SYSTEM-level privileges and extracted password hashes.



Step 4: Kibana log entry showing malicious PowerShell Invoke-WebRequest activity used for data exfiltration from host desktop-bkfj38l to remote IP during Aug 06, 2025 (13:50–14:10). This confirms the attacker's final stage of the intrusion, transferring stolen data out of the compromised system.



Step 5: Kibana log entry indicating confirmed data exfiltration via PowerShell on Aug 06, 2025 (13:50–14:10). This log validates the successful execution of the attacker's exfiltration technique as part of the final stage in the attack chain.



Step 6: Sigma Rule Creation Phase – I developed Sigma detection rules to identify malicious activities observed during my attack simulation, translating my earlier detection findings into structured YAML rules for automated log analysis.

title: Multi-Stage Attack Detection on desktop-bkfj38l id: e8f1c1d9-3b6f-4c8a-99f6-0a94b7a1d123 status: stable description: | Detects multi-stage attack on host desktop-bkfj38l involving:

1. **Initial access and payload.exe execution from Downloads folder via scheduled tasks (Win event ID 4698),**

2. **Persistence creation through scheduled tasks and registry Run key modifications referencing payload.exe,**
3. **Privilege escalation with Meterpreter payload injection, getsystem command, and hashdump activity (Win event IDs 1, 4672),**
4. **Data exfiltration via PowerShell Invoke-WebRequest sending data to suspicious external URL. author: Cybersecurity Analyst date: 2025-08-09 references:**

- **https://attack.mitre.org/techniques/T1053/005/**
- **https://attack.mitre.org/techniques/T1547/001/**
- **https://attack.mitre.org/techniques/T1078/**
- **https://attack.mitre.org/techniques/T1003/**
- **https://attack.mitre.org/techniques/T1041/ logsource: product: windows service: security definition:**
    - **winlog.event_id**
    - **process.name**
    - **process.command_line**
    - **host.name detection: selection_scheduled_task_creation: winlog.event_id: 4698 host.name: desktop-bkfj38l process.command_line|contains: "payload.exe" @timestamp|gte: "2024-08-01T14:20:00" @timestamp|lte: "2024-08-01T14:40:00" selection_registry_persistence: winlog.event_id: 13 # Registry value set (Sysmon) host.name: desktop-bkfj38l process.command_line|contains:**
        - **"reg add"**
        - **"payload.exe" @timestamp|gte: "2024-08-01T14:20:00" @timestamp|lte: "2024-08-01T14:40:00" selection_meterpreter_injection: winlog.event_id: 1 # Process creation (Sysmon) host.name: desktop-bkfj38l process.parent.name: meterpreter.exe process.name|contains: "notepad.exe" @timestamp|gte: "2025-08-03T11:45:00" @timestamp|lte: "2025-08-03T12:00:00" selection_privilege_escalation: host.name: desktop-bkfj38l @timestamp|gte: "2025-08-03T11:45:00" @timestamp|lte: "2025-08-03T12:00:00" any:**
        - **winlog.event_id: 4672**
        - **message|contains: "hashdump"**
        - **message|contains: "getsysteem" selection_data_exfiltration: host.name: desktop-bkfj38l @timestamp|gte: "2025-08-06T13:50:00" @timestamp|lte: "2025-08-06T14:10:00" process.name: powershell.exe process.command_line|contains:**
        - **"Invoke-WebRequest"**
        - **"192.168.31.163:8080" condition: selection_scheduled_task_creation or selection_registry_persistence or selection_meterpreter_injection or selection_privilege_escalation or selection_data_exfiltration fields:**

- **'@timestamp'**
- **'winlog.event_id'**
- **'host.name'**
- **'user.name'**
- **'process.name'**
- **'process.parent.name'**
- **'process.command_line'**
- **'message' level: high tags:**
- **attack.initial_access**
- **attack.persistence**
- **attack.privilege_escalation**
- **attack.exfiltration**
- **mitre.t1053.005**

- **mitre.t1547.001**
- **mitre.t1078**
- **mitre.t1003**
- **mitre.t1041**
- **windows**
- **sysmon**
- **winlogbeat**

Step 7: YARA Rule Phase – YARA is a pattern-matching tool used to identify and classify malware by scanning files, memory, or processes for specific byte patterns, strings, or behavioral signatures. In this phase, I created custom YARA rules based on the malicious artifacts from my attack simulation, enabling precise detection of the malware and persistence mechanisms observed during the engagement.

```
rule Payload_DesktopBKFJ38L_MultiStageAttack { meta: description = "Detects malicious payload.exe used in multi-stage attack (initial access, persistence, privilege escalation, exfiltration) on Windows 10 host desktop-bkfj38l" author = "Your Name" date = "2025-08-10" md5 = "b8e65d5320b676741a7d3988904ffe75" sha1 = "7f6436096fa1df98059a96e796d3a212ba3c6926" sha256 = "a78acea453c68a684917b967162a599f9881eba69471c202c071b86e72066037" file_size = "73802 bytes"

strings:
  $s1 = "!This program cannot be run in DOS mode."
  $s2 = "` .rdata"
  $s3 = "@.data"
  $s4 = "@2P7DF"
  $s5 = "@<ALhK"

condition:
  filesize == 73802 and
  (all of ($s*) or
  hash.md5(0, filesize) == "b8e65d5320b676741a7d3988904ffe75")


}
```

8. Challenges & Solutions

Challenges:

1. **From an Attacker's Perspective**

   a. **Bypassing Detection Mechanisms**: Modern security solutions like EDRs, YARA rules, and Sigma rules make evasion increasingly difficult.
   b. **Limited Reconnaissance Windows**: Strict network segmentation and monitoring reduce the available time for information gathering.
   c. **Payload Delivery Restrictions**: Advanced email filtering, sandboxing, and web proxies often block or detonate malicious files before execution.
   d. **Privilege Escalation Barriers**: Implementation of strict privilege management (e.g., PAM) and system hardening limit escalation opportunities.
2. **From a Blue Teamer's Perspective**
   a. **Detecting Sophisticated Threats**: Advanced persistent threats (APTs) use stealth techniques to blend in with normal activity.
   b. **Alert Fatigue**: High volume of alerts from multiple sources can lead to missed or delayed responses.
   c. **Rule Fine-tuning**: Creating YARA and Sigma rules without generating false positives while still catching threats is challenging.
   d. **Rapid Response Time**: Coordinating between teams and acting before attackers achieve persistence is difficult under time pressure.

Solutions (with Mitigation Strategies):

1. **From an Attacker's Perspective**
   a. **Challenge Mitigation (Offensive)**:
      i. Develop custom obfuscated payloads to evade signature-based detection.
      ii. Utilize living-off-the-land binaries (LOLBins) to operate without dropping suspicious files.
      iii. Leverage social engineering to bypass technical barriers.
2. **From a Blue Teamer's Perspective**
   a. **Challenge Mitigation (Defensive)**:
      i. **Strengthen Threat Detection**: Regularly update and test YARA and Sigma rules to identify both known and emerging threats.
      ii. **Reduce False Positives**: Continuously refine detection rules by analyzing legitimate activity patterns.

iii. **Improve Incident Response Speed**: Implement SOAR (Security Orchestration, Automation, and Response) solutions to automate repetitive tasks.
iv. **Enhance Threat Intelligence Sharing**: Use platforms like MISP to stay informed on attacker TTPs (Tactics, Techniques, Procedures).

9. Outcomes and Impact

Through the simulation and analysis phases of this project, I successfully demonstrated both offensive and defensive perspectives in a Windows 10 environment.

From the **attacker's perspective**, I was able to implement and validate multiple persistence mechanisms, followed by the creation of detection rules (Sigma and YARA) to evade and bypass standard security measures. This provided a clear understanding of real-world adversarial tactics and their technical execution.

From the **blue team perspective**, I developed and deployed effective detection and mitigation strategies to identify and neutralize these threats. The process included designing behavioral detection rules, implementing continuous monitoring, and applying incident response workflows to minimize dwell time.

The impact of this project lies in enhancing threat detection capabilities, strengthening security posture, and bridging the knowledge gap between red and blue teams. By experiencing both perspectives, I gained a 360° understanding of the attacker–defender dynamic, enabling a proactive and adaptive cybersecurity approach.

10. Conclusion

This project provided an in-depth, hands-on understanding of the attack–defense lifecycle, bridging both the red team (offensive) and blue team (defensive) perspectives. From reconnaissance and exploitation to detection engineering using Sigma rules and malware identification with YARA, I gained valuable insights into adversarial tactics and corresponding defensive measures.

By simulating real-world threat scenarios, analyzing vulnerabilities, and applying mitigation strategies, the exercise reinforced the importance of proactive security monitoring, threat intelligence integration, and incident response readiness. Ultimately, the experience highlighted that effective cybersecurity lies in the synergy between offensive awareness and defensive resilience—ensuring systems are not only tested against attacks but are continuously improved to withstand evolving threats.