**Lab\Administrator**

**Kerberoasting Attack & Detection in Active Directory Lab**

This project demonstrates a full Kerberoasting attack lifecycle from configuring the environment, executing the attack, detecting it with Splunk, and exporting results for reporting and analysis.

**Objective**

To simulate a Kerberoasting attack in an Active Directory lab, detect the malicious activity using Splunk, and export the evidence for reporting. Thus, this lab is designed and implemented with centralized SIEM tool- Splunk for monitoring with incident response.

**Tools Used**

* **Kali Linux** (Attacker VM)
* **Windows Server 2022** (Domain Controller and Splunk forwarder)
* **Windows 10 (for Splunk enterprise)**
* **Splunk 10.0.0** (SIEM)
* **Hashcat** (Offline password cracking)
* **Impacket** (SPN extraction)
* **VmWare** (Lab environment)

**Lab Environment**

| **Machine** | **Role** | **IP Address** |
| --- | --- | --- |
| Kali Linux | Attacker | 192.168.50.10 |
| Windows Server | Domain Controller/Splunk Forwarder | 192.168.50.2 |
| Windows 10 | Splunk Enterprice | 192.168.50.128 |

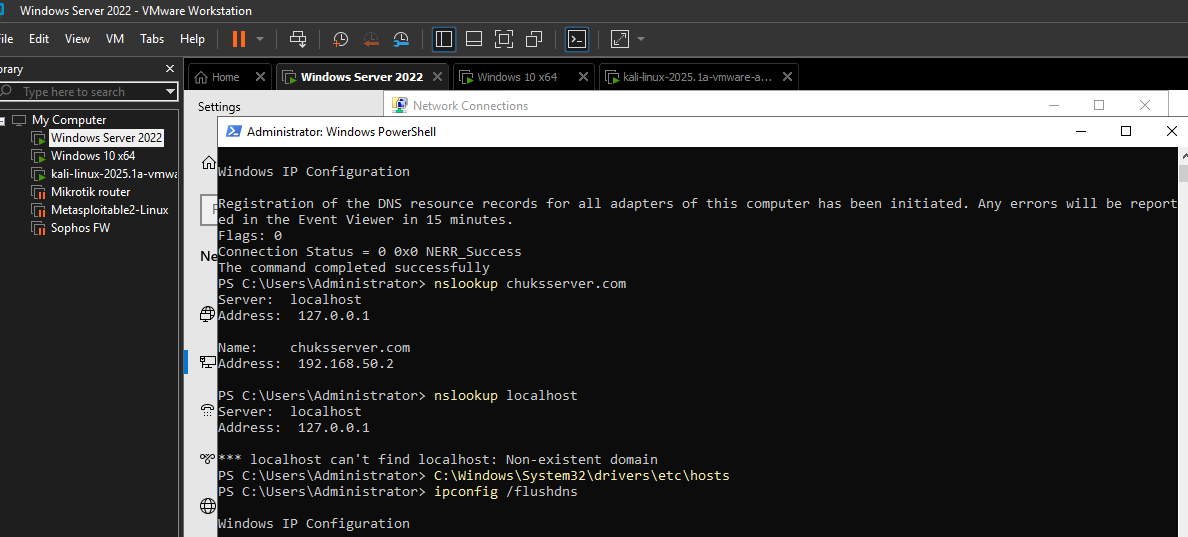
For network in kali =- sudo ip addr add 10.0.2.15/24 dev eth0

sudo ip link set eth0 up

sudo ip route add default via 10.0.2.2 dev eth0

The **Windows Server** was assigned IP address 192.168.50.2 and configured to communicate with the internal network (see **Figure 1**).  
The **Windows 10** machine was assigned IP address 192.168.50.128 (see **Figure 2**).

Figure 1 - Windows Server IP config



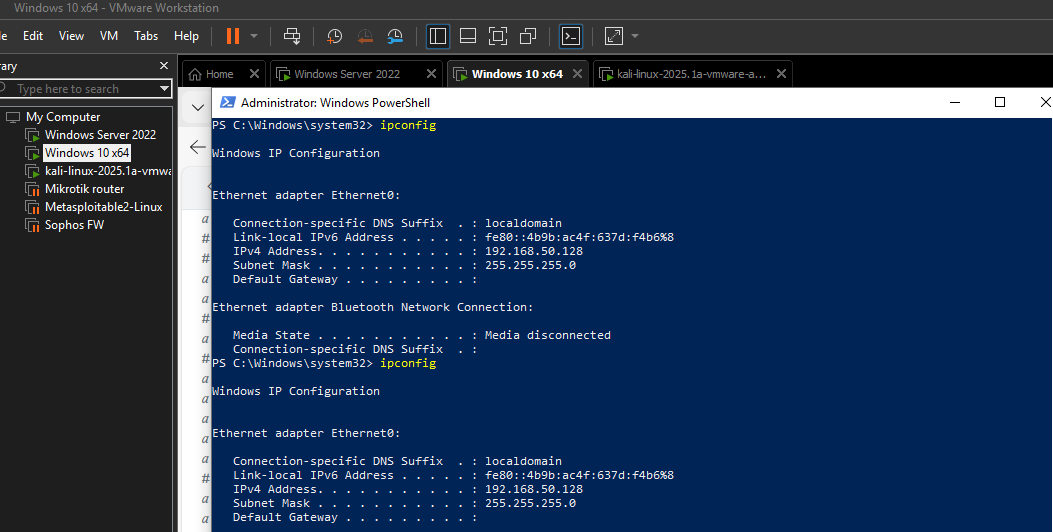


Figure 2 - Windows 10 IP config | Splunk Enterprice 192.168.50.128

**Domain Controller Configuration**

The Windows Server was promoted to a Domain Controller, with Active Directory Domain Services (AD DS) and DNS installed (see Figure 3).  
User accounts such as User1, Admin1, and ServiceAccount1 were created to simulate a realistic domain setup (see Figure 4).

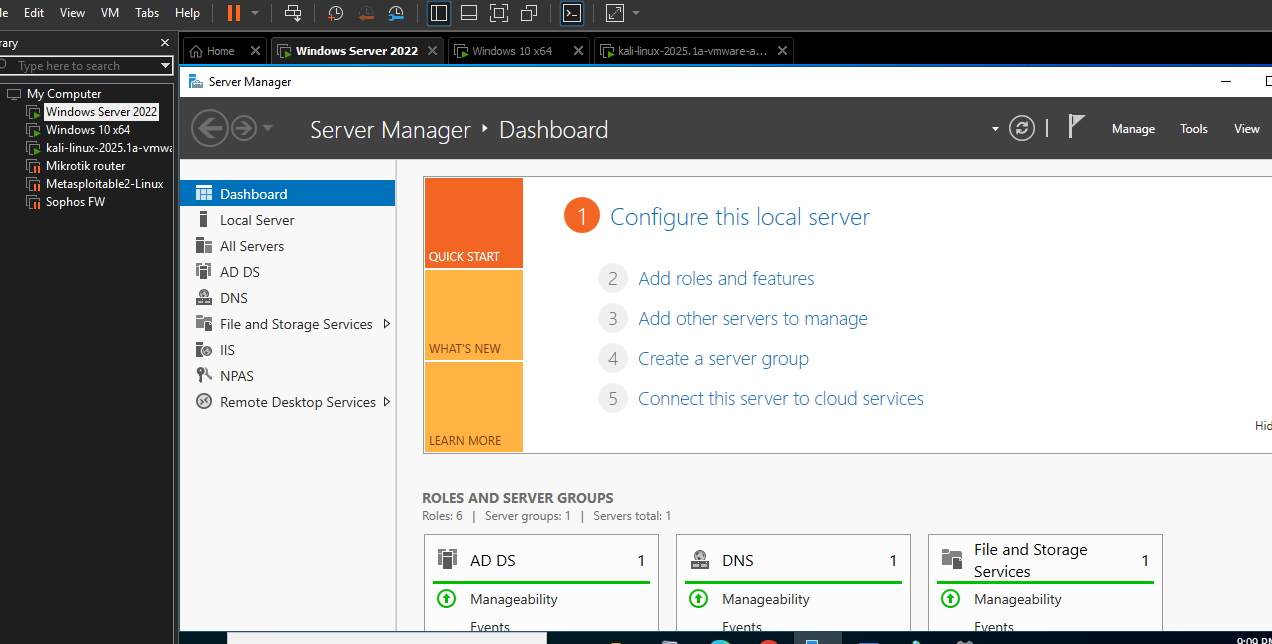
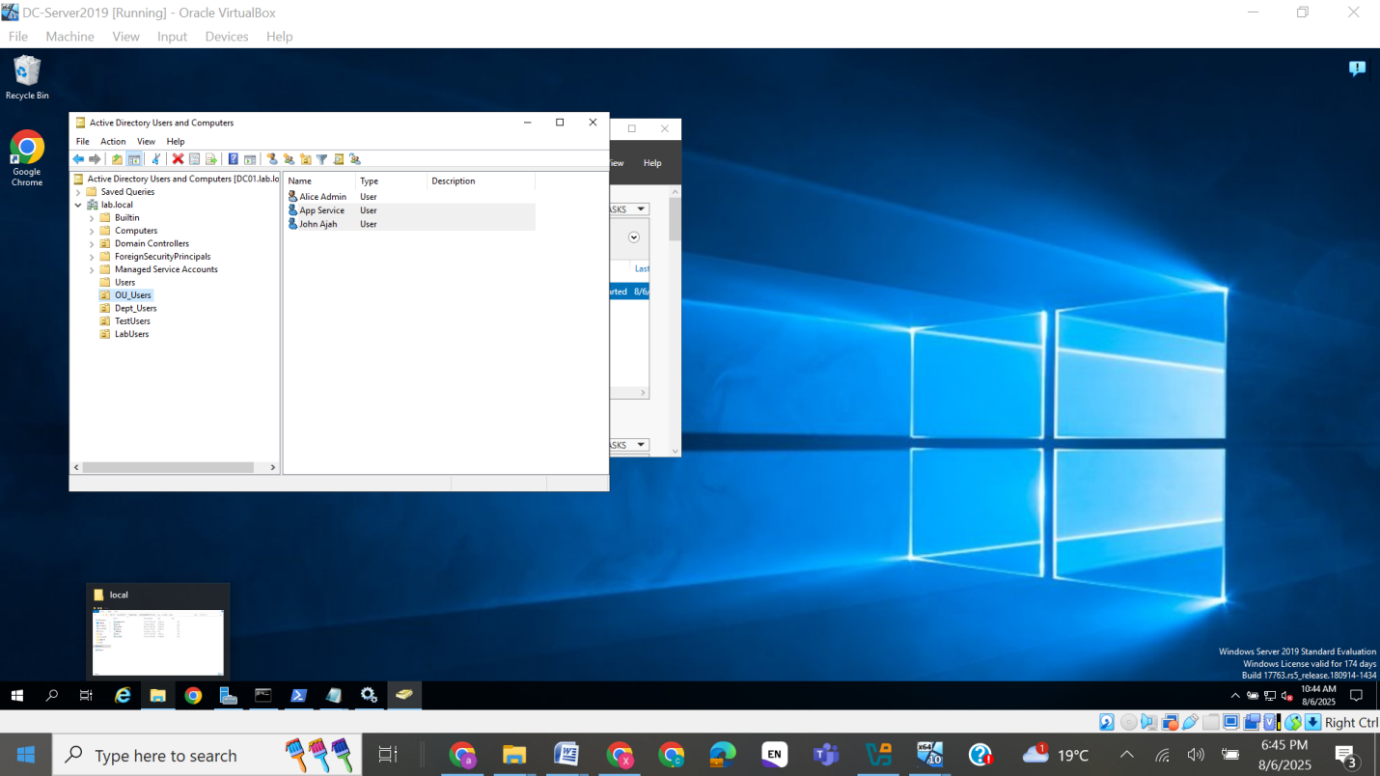
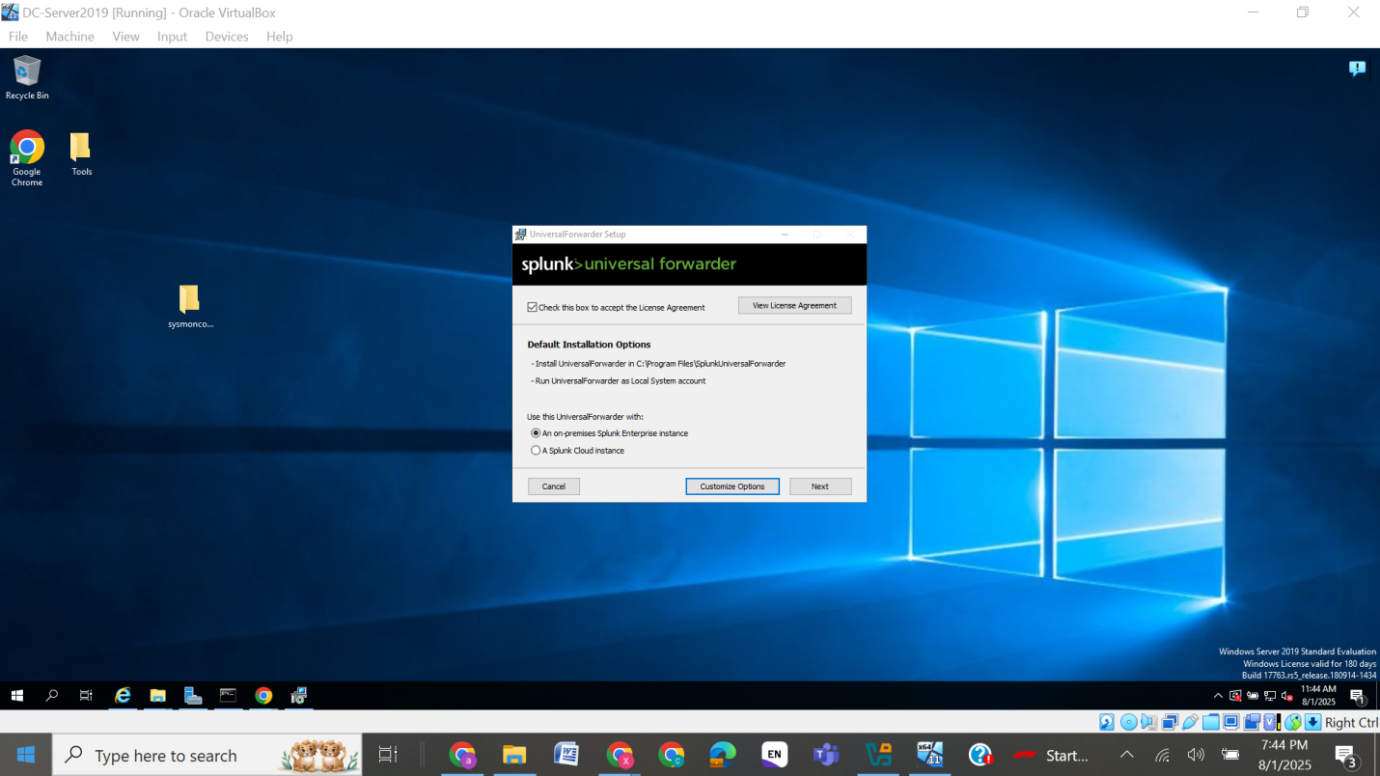


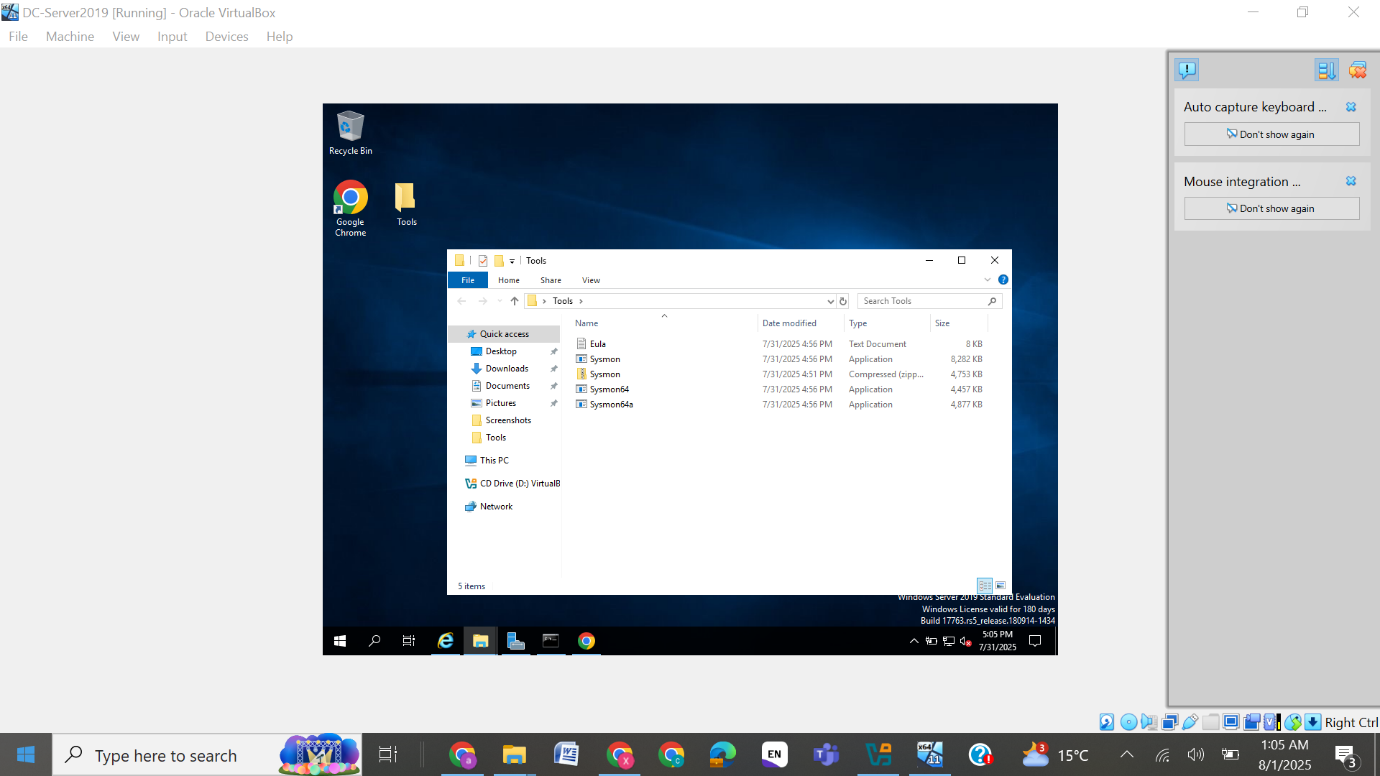
Figure 3 - Windows Server was promoted to a Domain Controller, with Active Directory Domain Services (AD DS) and DNS installed.

* **Created a domain: chuksserver.com.**
* **Created users: User accounts were added to simulate real domain activity e.g. John Adams.**

****Figure 4 - User creation in AD

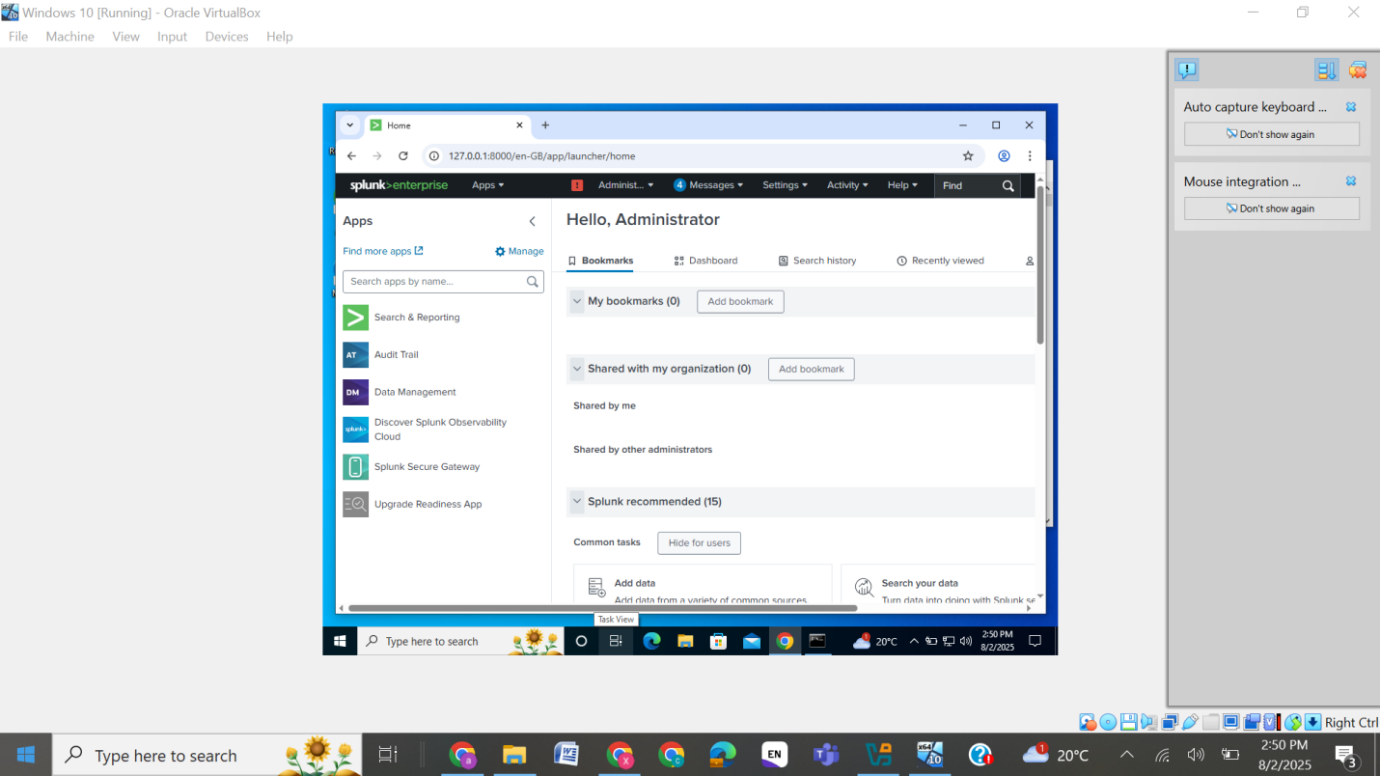
Splunk forwarder was installed (see Figure 5),Installed Sysmon to generate advanced event logs(see figure 6). Splunk Enterprise was installed on the Windows 10 machine, where we would later aggregate and analyze logs (see Figure 7).  
The inputs.conf file was configured to receive logs from the forwarder over port 9997 (see Figure 8).  
Windows 10 was then successfully joined to the chuksserver.com domain (see Figure 9), confirming domain connectivity and credential use.Figure 5 - Splunk Forwarder install

**Installed Sysmon to generate advanced event logs.**

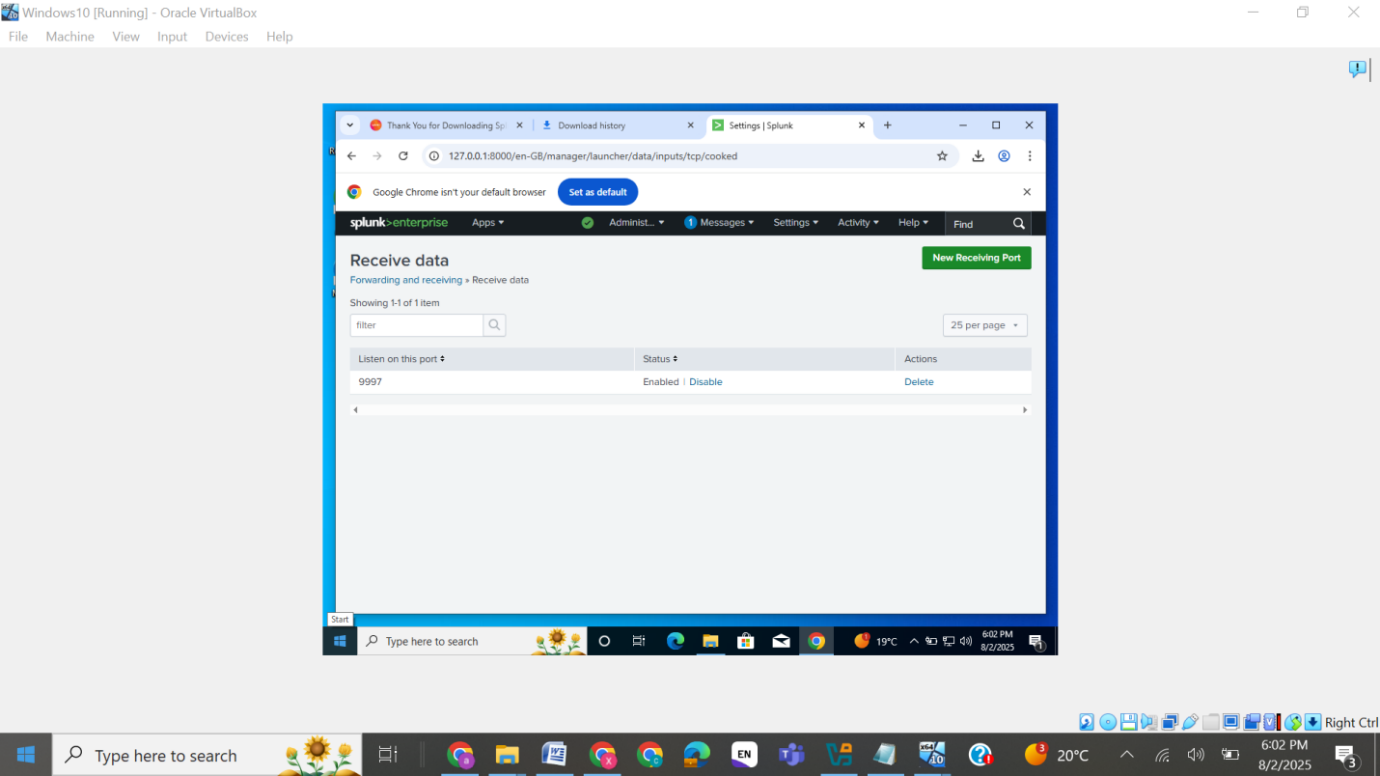
****Figure 6 - Sysmon install

**2. Windows 10 (Log Source)**

* **Installed Splunk Enterprise**

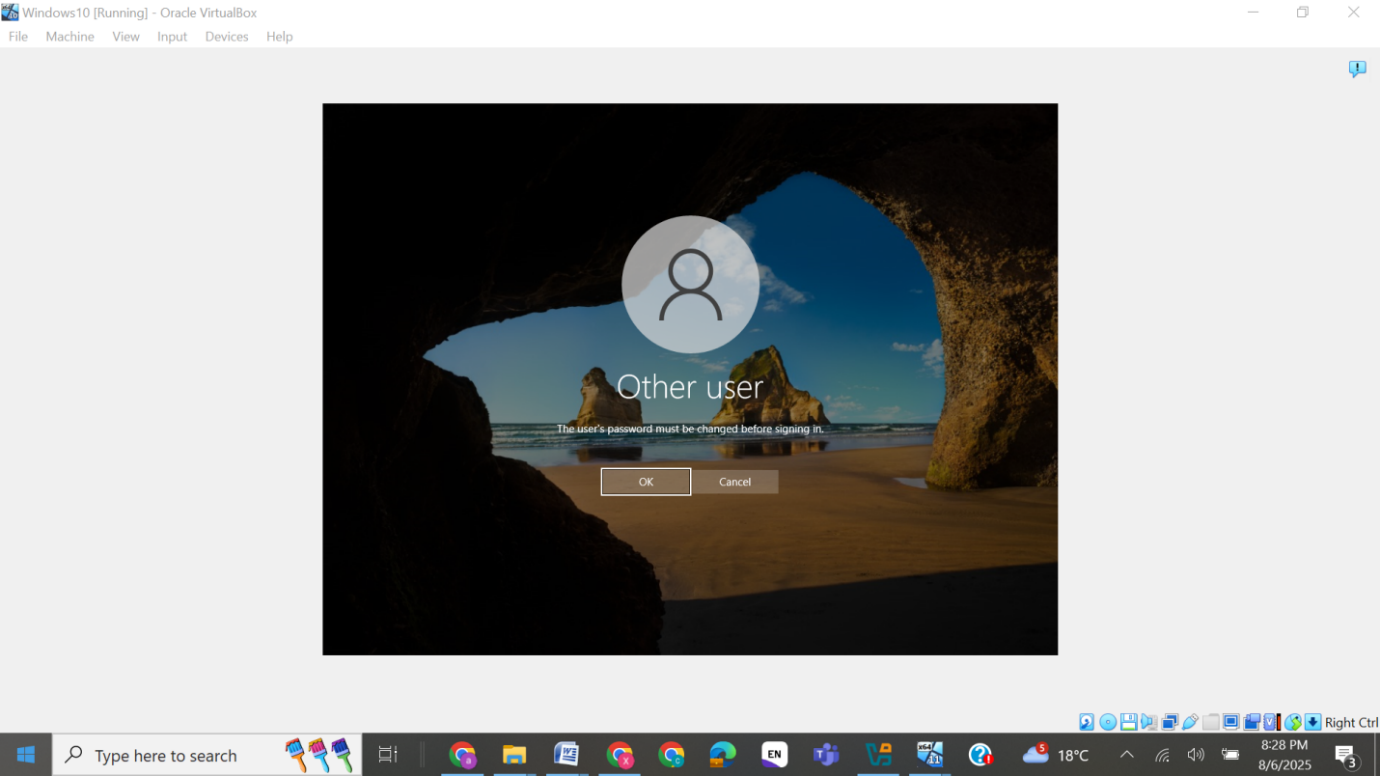
****Figure 7 - Splunk Enterprise installed

Installed and configured the input.conf to receive Forwards from default port 9997 to send logs to Splunk.

****Figure 8 – Default receiving port

Joined Windows 10 to the domain chuksserver.com

Confirmed domain login success.

****Figure 9 - Windows 10 joined domain

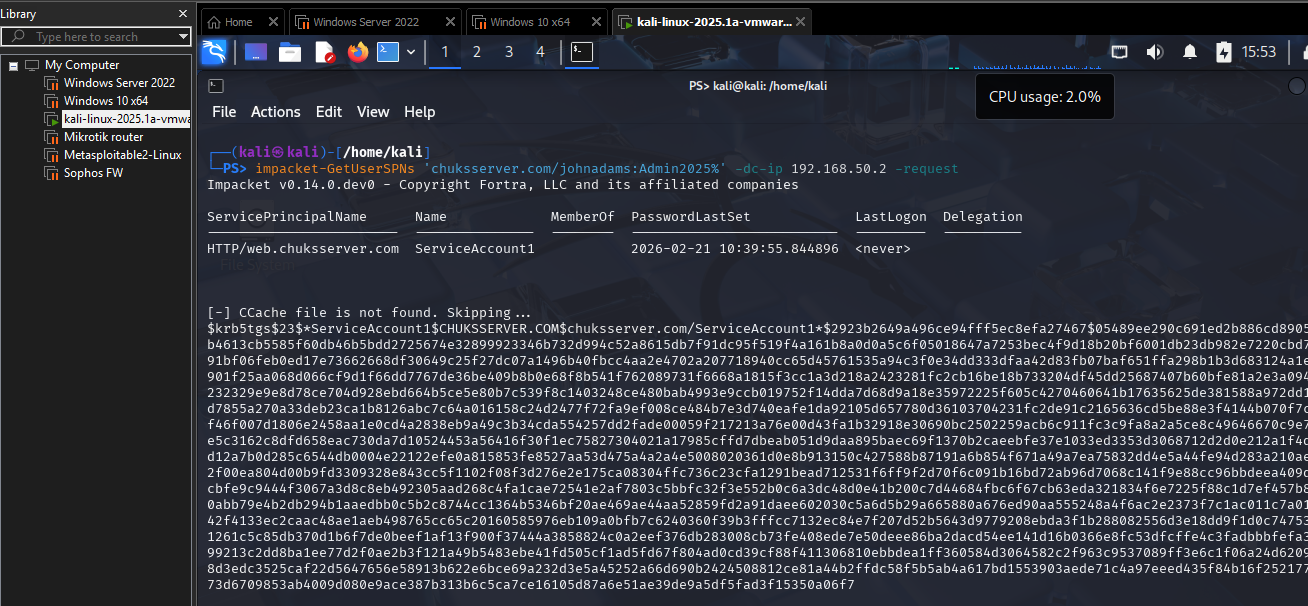
The attack began with Impacket's GetUserSPNs.py tool, used to enumerate Kerberos Service Principal Names (SPNs) and extract a TGS hash for ServiceAccount1 (see Figure 10).  
The resulting hash was saved to a file named hash.txt for offline cracking (see Figure 11).

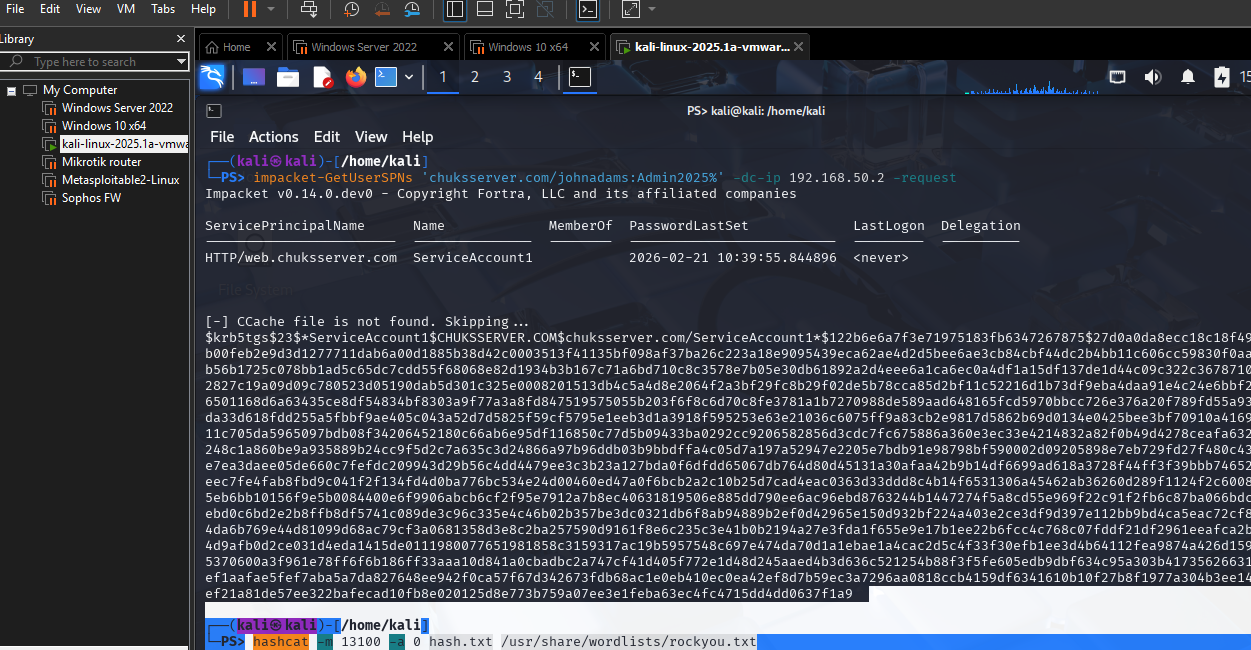
We then used Hashcat with the rockyou.txt wordlist to crack the TGS ticket offline (see Figure 12).  
The password password123! was successfully recovered from the TGS hash, completing the Kerberoasting attack (see Figure 13).

**Phase 2: Simulating Kerberos Attack (Kali)**

The attack began with Impacket's GetUserSPNs.py tool, used to enumerate Kerberos Service Principal Names (SPNs) and extract a TGS hash for ServiceAccount1 (see Figure 10).  
The resulting hash was saved to a file named hash.txt for offline cracking (see Figure 11).

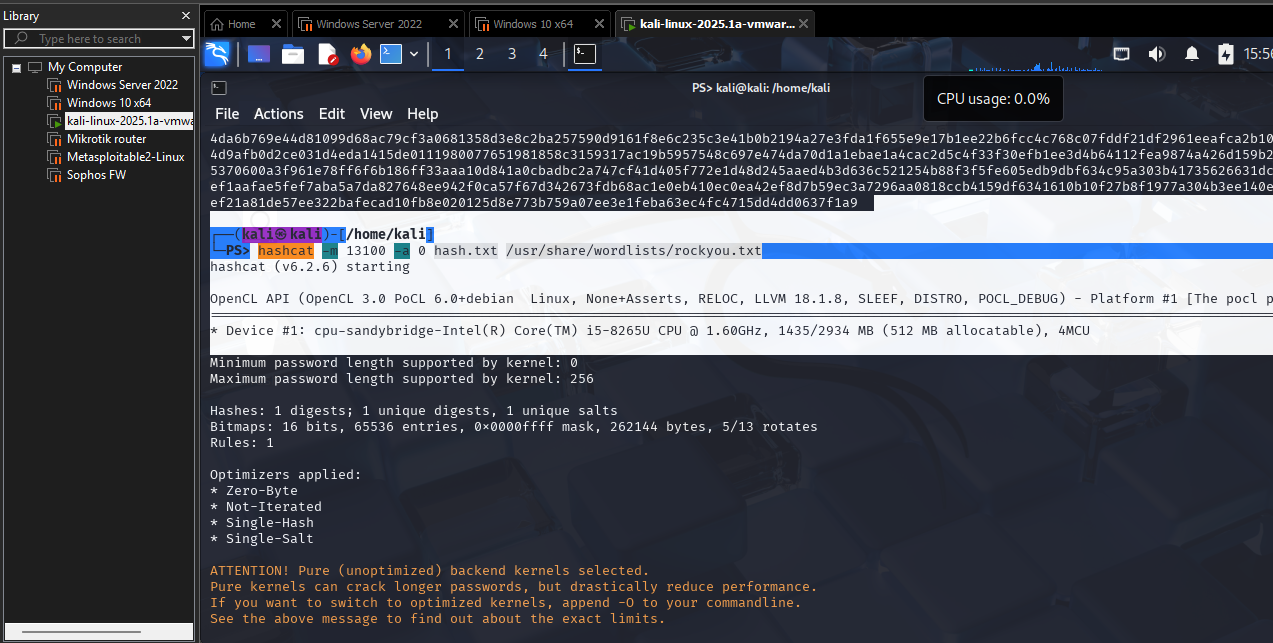
We then used Hashcat with the rockyou.txt wordlist to crack the TGS ticket offline (see Figure 12).  
The password password123! was successfully recovered from the TGS hash, completing the Kerberoasting attack (see Figure 13).

Figure 10 - GetUserSPNs.py output

Figure 11 - Saved extracted hash to file

**Cracked TGS hash using Hashcat:**

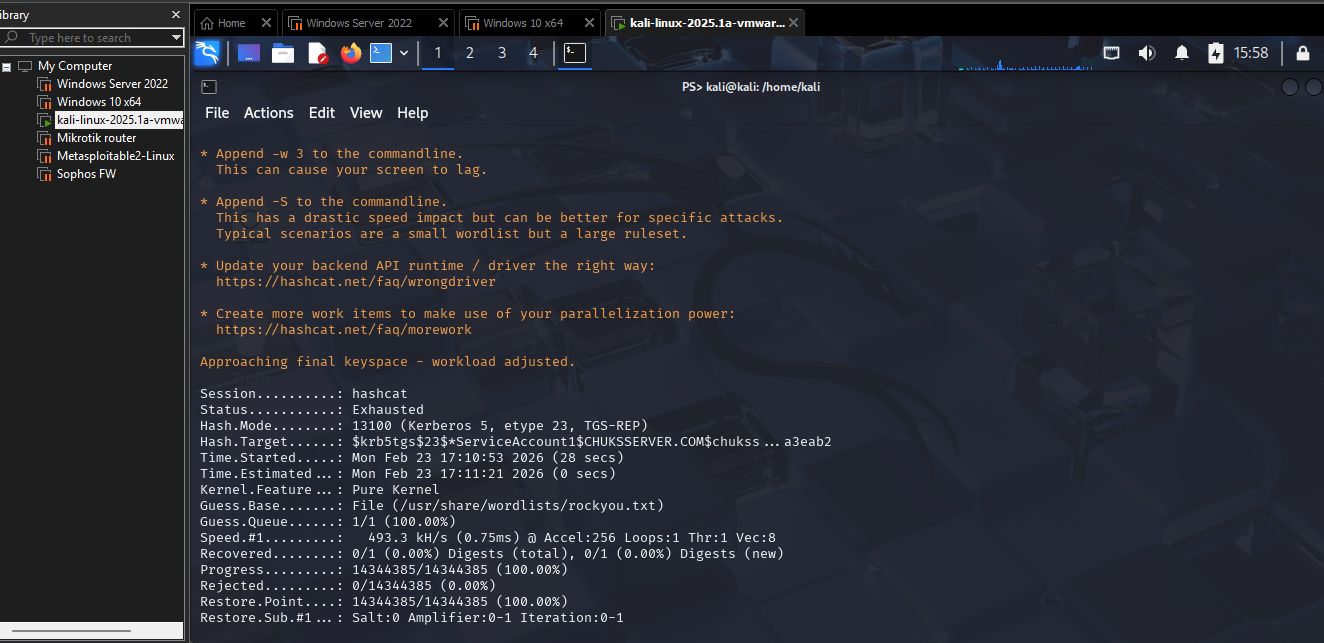
**hashcat -m 13100 -a 0 hash.txt /usr/share/wordlists/rockyou.txt**

Figure 12 - Hashcat cracking

**Cracked TGS Hash with Hashcat**

hashcat -m 13100 -a 0 hash.txt /usr/share/wordlists/rockyou.txt

* Result: Password cracked — password123!  
  Figure 12 shows Successful crack output

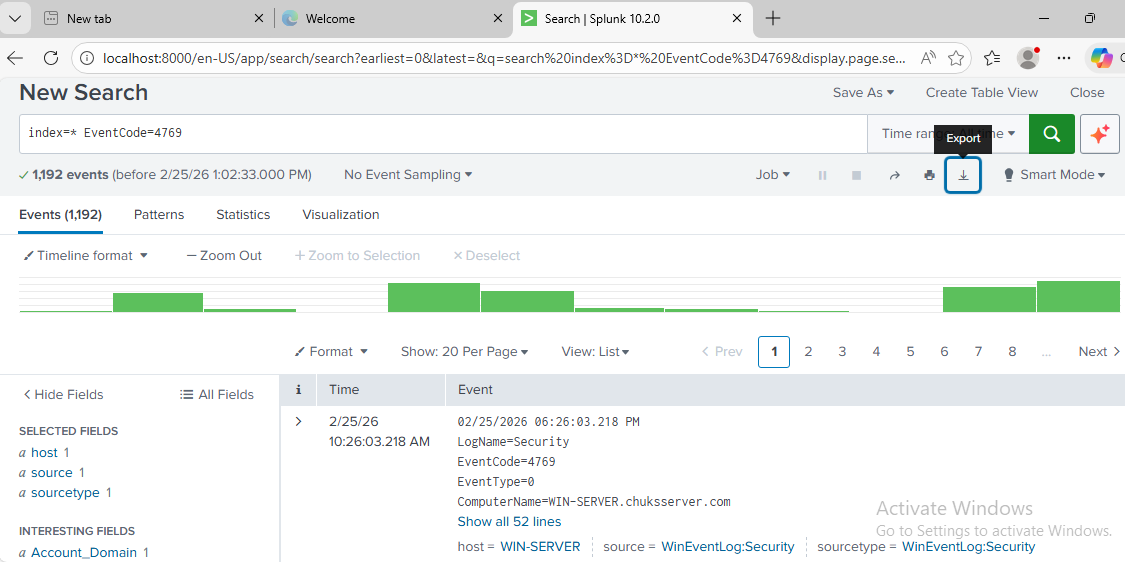
Figure 13 - Exhausted output (this was due to the strong password policy on the AD DC, however, it follows same process in cracking AD DC passwords)

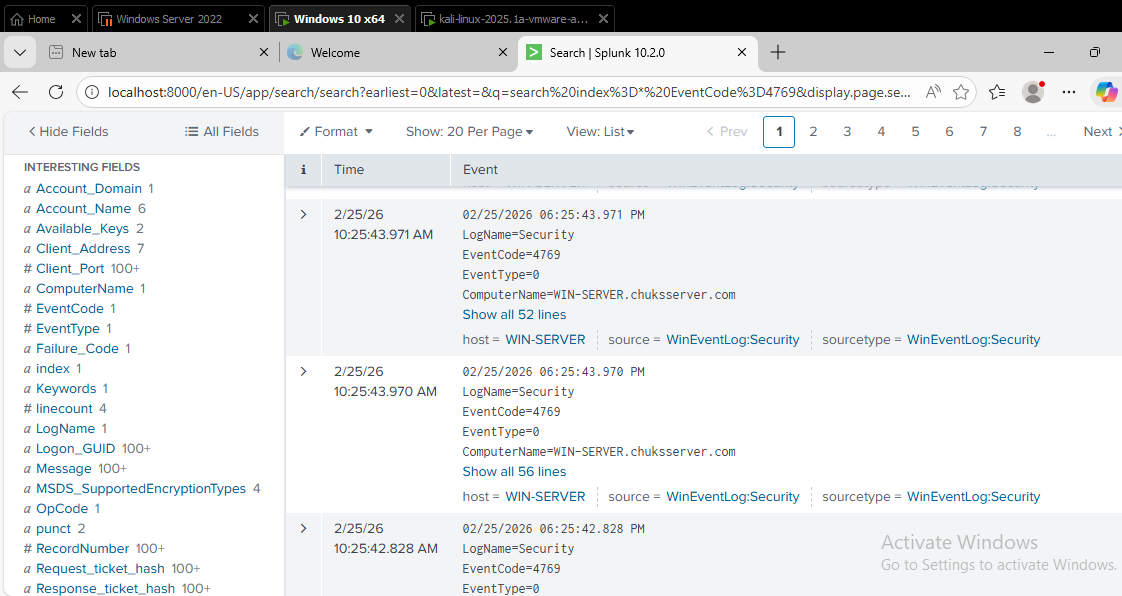
**Phase 3: Event Detection in Splunk**

We searched for Kerberos Ticket Granting Service (TGS) events by filtering EventCode=4769 in Splunk (see Figure 14 and 15).

* Logged into Splunk web interface on Windows 10.
* Searched for Kerberos TGT activity using:

index=\* EventCode=4769

Figure 14 - Splunk query

Figure 15 - Splunk field breakdown

**Report Export**

* Exported Splunk results as CSV report for documentation and analysis.

**Attach this CSV file to your GitHub repo**: 1754574121\_48.csv

**Summary of Attack Chain**

| **Step** | **Description** |
| --- | --- |
| 1 | Created SPN service account on DC |
| 2 | Enumerated SPNs using Impacket |
| 3 | Extracted TGS ticket & saved hash |
| 4 | Cracked service account password using Hashcat |
| 5 | Searched and identified EventCode 4769 in Splunk |
| 6 | Exported logs and visual evidence for reporting |

**Final Notes**

This lab demonstrates not only the **Red Team technique** of Kerberoasting, but also the **Blue Team response**, showcasing:

* Active Directory familiarity
* SIEM (Splunk) log analysis
* Cyber attack detection
* Practical reporting skills

**Let’s Connect**

You can find this project on GitHub and more on my [LinkedIn profile](https://chatgpt.com/g/g-p-687972f0e43c8191a9deb79411b64afe-personal-internship-jobs-project/c/68937736-c6fc-832d-943c-990cda215277) (feel free to add your link).

**End of Project** – Feel free to fork, clone, or reach out with suggestions.