Oleksandr Shevchuk

This project was completed as part of the "Cyber Security" (7736/37) course — Information Security and Corporate Network Protection, at John Bryce College.

```
#+-------------------------------------------------------------------------------------+
#|                                                                                     |
#|                     Bash Script for Penetration Testing                             |
#|                                                                                     |
#|   ▸ Checks for required utilities and internet connection                           |
#|   ▸ Automatically installs and starts 'nipe' to route traffic through the TOR network |
#|   ▸ Masks your MAC address and requests a new IP address                            |
#|   ▸ Gets user input: target network, output directory, and scan mode (Basic/Full)   |
#|   ▸ Basic mode: scans TCP/UDP, detects service versions, and checks weak passwords  |
#|   ▸ Full mode: includes NSE scripts and vulnerability mapping with Searchsploit     |
#|   ▸ Tests weak credentials for SSH, RDP, FTP, and TELNET                            |
#|   ▸ Logs results, allows searching inside them, and saves everything to a ZIP archive |
#|   ▸ Metasploitable2 was used as a testbed for development and debugging             |
#|                                                                                     |
#|                              Requires root privileges!                              |
#|                                                                                     |
#+-------------------------------------------------------------------------------------+
```

Project Structure

1. Getting the User Input

1.1 Get from the user a network to scan.
1.2 Get from the user a name for the output directory.
1.3 Allow the user to choose 'Basic' or 'Full'.
1.3.1 Basic: scans the network for TCP, including the service version and weak passwords.
1.3.2 Full: include Nmap Scripting Engine (NSE), weak passwords, and vulnerability analysis.
1.4 Make sure the input is valid.

2. Weak Credentials

2.1 Look for weak passwords used in the network for login services.
2.1.1 Have a built-in password.lst to check for weak passwords.
2.1.2 Allow the user to supply their own password list.
2.2 Login services to check include: SSH, RDP, FTP, and TELNET.

3. Mapping Vulnerabilities

3.1 Mapping vulnerabilities should only take place if Full was chosen.
3.2 Display potential vulnerabilities via NSE and Searchsploit.

4. Log Results

4.1 During each stage, display the stage in the terminal.
4.2 At the end, show the user the found information.
4.3 Allow the user to search inside the results.
4.4 Allow to save all results into a Zip file.

1. Getting the User Input

1.1 Get from the user a network to scan.

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

if [[ ! "$network" =~ ^[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+/[0-9]+$ ]]; then
    echo -e "\e[91m[!] Wrong format. Example: 192.168.0.0/24\e[0m\n"
    SELECT_SCAN_METOD
fi
```

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

1.2 Get from the user a name for the output directory.

```
# Prompt the user to enter a folder name where scan results will be stored
read -p $'\e[31m[!]\e[0m\e[34m Enter folder name for scan results: \e[0m' working_dir
```

```
[!] Enter folder name for scan results: Horns&HoovesLLC
```

1.3 Allow the user to choose 'Basic' or 'Full'.

```
while true; do
    # Prompt the user to choose scan mode: Basic or Full
    read -p $'\e[31m[?]\e[0m\e[34m Choose scan mode: [B]asic or [F]ull: \e[0m' scan_mode_raw
    # Normalize scan mode to readable form
    if [[ "$scan_mode_raw" =~ ^[Bb]$ ]]; then
        scan_mode="Basic"
        break
    elif [[ "$scan_mode_raw" =~ ^[Ff]$ ]]; then
        scan_mode="Full"
        break
    else
        echo -e "\e[91m[!] Wrong choice. Example: B or F\e[0m"
    fi
done
```

```
[?] Choose scan mode: [B]asic or [F]ull: f
```

1.3.1 Basic: scans the network for TCP including the service version and weak passwords.

```
# BASIC_SCAN
# Runs a basic scan: TCP, version detection, and basic brute-force.
BASIC_SCAN()
{
    # Print starting message with color formatting
    echo -e "\n\e[31m[!]\e[0m\e[32m Starting BASIC scan...\e[0m"
```

```
[!] Starting BASIC scan...
[*] Scanning: 192.168.29.2
[*] Scanning: 192.168.29.129
```

```
# Run nmap scan with service/version detection and OS detection, output XML to the IP directory
nmap -Pn -sS -sV ${ip} -oX "$host_dir/res_tcp_${ip}.xml" -oN "$host_dir/res_tcp_${ip}.txt" > /dev/null 2>&1 &
SPINNER $!
```

```
if [[ "$protocol" == "ftp" ]]; then
    # Run FTP brute-force attack using nmap
    nmap -p 21 "$ip" --script ftp-brute --script-args "$script_args" -oN "$host_dir/nmap_ftp_brute_${ip}.txt" > /dev/null 2>&1 &
    SPINNER $!
```

```
[*] Searching for weak passwords on 192.168.29.129 ...
[>>] Trying ftp on 192.168.29.129 ...
[+] Weak FTP credentials found for 192.168.29.129 (see: WidgetsInc/192.168.29.129/nmap_ftp_brute_192.168.29.129.txt)
[>>] Trying ssh on 192.168.29.129 ...
[+] Weak SSH credentials found for 192.168.29.129 (see: WidgetsInc/192.168.29.129/nmap_ssh_brute_192.168.29.129.txt)
[>>] Trying telnet on 192.168.29.129 ...
[+] Weak Telnet credentials found for 192.168.29.129 (see: WidgetsInc/192.168.29.129/nmap_telnet_brute_192.168.29.129.txt)
```

1.3.2 Full: include Nmap Scripting Engine (NSE), weak passwords, and vulnerability analysis.

```
# FULL SCAN
# Performs a complete scan for each host in live_hosts.txt
FULL_SCAN() {

    # Print starting message in colored format
    echo -e "\n\e[31m[!]\e[0m\e[32m Starting FULL scan...\e[0m"
```

```
[!] Starting FULL scan...
[*] Scanning: 192.168.29.2
```

```
# TCP scan: service/version detection, OS detection
nmap -Pn -sC -sV -O "${ip}" -oX "$host_dir/res_tcp_${ip}.xml" -oN "$host_dir/res_tcp_${ip}.txt" > /dev/null 2>&1 &
SPINNER $!
wait $!

# UDP scan: top 10 ports, service detection
nmap -Pn -sU --top-ports 10 -sV "${ip}" -oX "$host_dir/res_udp_${ip}.xml" -oN "$host_dir/res_udp_${ip}.txt" > /dev/null 2>&1 &
SPINNER $!
wait $!
```

```
[*] Searching for weak passwords on 192.168.29.129 ...
[>>] Trying ftp on 192.168.29.129 ...
[+] Weak FTP credentials found for 192.168.29.129 (see: Horns&HoovesLLC/192.168.29.129/nmap_ftp_brute_192.168.29.129.txt)
[>>] Trying ssh on 192.168.29.129 ...
[+] Weak SSH credentials found for 192.168.29.129 (see: Horns&HoovesLLC/192.168.29.129/nmap_ssh_brute_192.168.29.129.txt)
[>>] Trying telnet on 192.168.29.129 ...
[+] Weak Telnet credentials found for 192.168.29.129 (see: Horns&HoovesLLC/192.168.29.129/nmap_telnet_brute_192.168.29.129.txt)
```

1.4 Make sure the input is valid.

```
# Show summary and ask for confirmation
echo -e "\n\e[31m[*]\e[0m\e[32m Please verify the entered data:\e[0m"
echo ""
echo -e "    Target network : $network"
echo -e "    Scan mode      : $scan_mode"
echo -e "    Output folder  : $working_dir"

while true; do
    read -p $'\n\e[31m[?]\e[0m\e[34m Is everything correct? (Y/N): \e[0m' validation
    if [[ "$validation" =~ ^[Yy]$ ]]; then
```

```
[*] Please verify the entered data:

    Target network : 192.168.29.0/24
    Scan mode      : Full
    Output folder  : Horns&HoovesLLC

[?] Is everything correct? (Y/N): y
```

2. Weak Credentials

2.1 Look for weak passwords used in the network for login services.

```
# WEAK_CREDENTIALS
# Attempts to discover weak credentials on open services using Nmap scripts and Hydra.
# SSH, FTP, and Telnet are scanned using Nmap scripts.
# RDP is scanned using Hydra because Nmap lacks RDP brute-force support.
WEAK_CREDENTIALS() {
```

```
# Common script arguments for nmap brute-force scripts
script_args="userdb=$data/usernames.lst,passdb=$passwd_lst,brute.threads=10"

# Process each detected protocol individually
for protocol in $(cat "$host_dir/protocol_for_scan.txt"); do
    echo -e "\e[33m[>>] Trying $protocol on $ip...\e[0m"
```

2.1.1 Have a built-in password.lst to check for weak passwords.

```
# Download usernames and passwords lists to local directory
wget --no-check-certificate -O "$data/usernames.lst" "https://drive.google.com/uc?export=download&id=1mPSHpfd-P35FNv4r4nQVxfJmAesVLER_" >/dev/null 2>&1
wget --no-check-certificate -O "$data/passwords.lst" "https://drive.google.com/uc?export=download&id=1dCdCBwTg15ZeUmTyZ7YyU7oxb3mccGAu" >/dev/null 2>&1
```

```
read -p $'\n\e[31m[?]\e[0m\e[34m Choose password list: [S]tandard or [C]ustom: \e[0m' passwd_lst_raw
if [[ "$passwd_lst_raw" =~ ^[Ss]$ ]]; then
    passwd_lst="$data/passwords.lst"
    break
```

```
[?] Choose password list: [S]tandard or [C]ustom: s
```

2.1.2 Allow the user to supply their own password list.

```
# CONFIGURE_WORDLISTS
# Downloads or sets up required wordlists for brute-force operations.
CONFIGURE_WORDLISTS() {
    while true; do
        # Prompt the user to choose between standard or custom password list
        read -p $'\n\e[31m[?]\e[0m\e[34m Choose password list: [S]tandard or [C]ustom: \e[0m' passwd_lst_raw
```

```
elif [[ "$passwd_lst_raw" =~ ^[Cc]$ ]]; then
    # Ask user to provide full path to their custom password list
    read -p $'\e[31m[!]\e[0m\e[34m Enter the full path to your custom password list: \e[0m' custom_passwd_lst
    echo ""
    passwd_lst="$custom_passwd_lst"
    break
```

```
[?] Choose password list: [S]tandard or [C]ustom: c
[!] Enter the full path to your custom password list: /home/kali/Desktop/data/passwords.lst
```

2.2 Login services to check include: SSH, RDP, FTP, and TELNET.

```
# WEAK_CREDENTIALS
# Attempts to discover weak credentials on open services using Nmap scripts and Hydra.
# SSH, FTP, and Telnet are scanned using Nmap scripts.
# RDP is scanned using Hydra because Nmap lacks RDP brute-force support.
WEAK_CREDENTIALS() {
    # Iterate over each IP address from the list of live hosts
    for ip in $(cat "$working_dir/live_hosts.txt"); do
        host_dir="$working_dir/$ip"
        tcp_file="$host_dir/res_tcp_${ip}.txt"

        echo -e "\e[31m[*]\e[0m\e[32m Searching for weak passwords on $ip...\e[0m"
```

```
[*] Searching for weak passwords on 192.168.29.2...
[*] Searching for weak passwords on 192.168.29.129...
[>>] Trying ftp on 192.168.29.129...
[+] Weak FTP credentials found for 192.168.29.129 (see: Horns&HoovesLLC/192.168.29.129/nmap_ftp_brute_192.168.29.129.txt)
[>>] Trying ssh on 192.168.29.129...
[+] Weak SSH credentials found for 192.168.29.129 (see: Horns&HoovesLLC/192.168.29.129/nmap_ssh_brute_192.168.29.129.txt)
[>>] Trying telnet on 192.168.29.129...
[+] Weak Telnet credentials found for 192.168.29.129 (see: Horns&HoovesLLC/192.168.29.129/nmap_telnet_brute_192.168.29.129.txt)
```

3. Mapping Vulnerabilities

3.1 Mapping vulnerabilities should only take place if Full was chosen.

```
# MAPPING
# Processes scan results, extracts open services, and runs vulnerability scan
MAPPING() {

    # Receive the IP argument passed from FULL_SCAN
    local host="$1"
    local host_dir="$working_dir/$host"

    echo -e "\e[33m[>>] Mapping for: $host\e[0m"
```

```
[!] Starting FULL scan...
[*] Scanning: 192.168.29.2
[>>] Mapping for: 192.168.29.2
[-] No vulnerabilities detected on 192.168.29.2
[*] Scanning: 192.168.29.129
[>>] Mapping for: 192.168.29.129
[+] Vulnerable services found on 192.168.29.129 (see: Horns&HoovesLLC/192.168.29.129/mapping_tcp_192.168.29.129.html)
```

3.2 Display potential vulnerabilities via NSE and Searchsploit.

```
# If open ports exist, run nmap vulnerability scripts
if [ -n "$ports" ]; then
    nmap -p"$ports" --script vuln "$host" -oX "$host_dir/mapping_tcp_${host}.xml" -oN "$host_dir/mapping_tcp_${host}.txt" > /dev/null 2>&1 &
    SPINNER $!
    wait $!
```

```
[+] Vulnerable services found on 192.168.29.129 (see: Horns&HoovesLLC/192.168.29.129/mapping_tcp_192.168.29.129.html)
```

Port		State (toggle closed [0] | filtered [0])	Service	Reason
21	tcp	open	ftp	syn-ack
	ftp-vsftpd-backdoor	VULNERABLE: vsFTPd version 2.3.4 backdoor State: VULNERABLE (Exploitable) IDs: BID:48539 CVE:CVE-2011-2523 vsFTPd version 2.3.4 backdoor, this was reported on 2011-07-04. Disclosure date: 2011-07-03 Exploit results: Shell command: id Results: uid=0(root) gid=0(root) References: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-2523 https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/unix/ftp/vsftpd_234_backdoor.rb http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backdoored.html https://www.securityfocus.com/bid/48539		

4. Log Results

4.1 During each stage, display the stage in the terminal.

```
[!] Starting FULL scan...
[*] Scanning: 192.168.29.2
```

```
[!] Opening a shell in 'Horns&HoovesLLC'.
```

```
[?] Choose password list: [S]tandard or [C]ustom: s
```

```
[*] Searching for weak passwords on 192.168.29.129...
```

```
[>>] Mapping for: 192.168.29.129
```

```
[!] Scan completed.
```

```
[+] Results saved to: Horns&HoovesLLC.zip
```

4.2 At the end, show the user the found information.

```
Horns&HoovesLLC
├── 192.168.29.129
│   ├── mapping_tcp_192.168.29.129.html
│   ├── mapping_tcp_192.168.29.129.txt
│   ├── mapping_tcp_192.168.29.129.xml
│   ├── nmap_ftp_brute_192.168.29.129.txt
│   ├── nmap_ssh_brute_192.168.29.129.txt
│   ├── nmap_telnet_brute_192.168.29.129.txt
│   ├── protocol_for_scan.txt
│   ├── res_tcp_192.168.29.129.html
│   ├── res_tcp_192.168.29.129.txt
│   ├── res_tcp_192.168.29.129.xml
│   ├── res_udp_192.168.29.129.html
│   ├── res_udp_192.168.29.129.txt
│   └── res_udp_192.168.29.129.xml
├── 192.168.29.2
│   ├── mapping_tcp_192.168.29.2.txt
│   ├── protocol_for_scan.txt
│   ├── res_tcp_192.168.29.2.html
│   ├── res_tcp_192.168.29.2.txt
│   ├── res_tcp_192.168.29.2.xml
│   ├── res_udp_192.168.29.2.html
│   ├── res_udp_192.168.29.2.txt
│   └── res_udp_192.168.29.2.xml
└── live_hosts.txt
```

4.3 Allow the user to search inside the results.

```
# Start a temporary interactive shell with a custom intro message
bash --rcfile <(echo "echo -e '\e[33m[!] This shell is opened inside the SEARCH_RESULTS function for search and post-scan result processing.\e[0m';
                     echo -e '\e[33m[!] When finished, type exit and press Enter to return to the main program.\e[0m'\n")
```

```
[!] Opening a shell in 'Horns&HoovesLLC'.
[!] This shell is opened inside the SEARCH_RESULTS function for search and post-scan result processing.
[!] When finished, type exit and press Enter to return to the main program.n
root@kali:/home/kali/Desktop/PenetrationTesting/Horns&HoovesLLC# ls -la
total 20
drwxr-xr-x 4 root root 4096 Aug 16 10:38 .
drwxrwxr-x 8 kali kali 4096 Aug 16 10:36 ..
drwxr-xr-x 2 root root 4096 Aug 16 10:47 192.168.29.129
drwxr-xr-x 2 root root 4096 Aug 16 10:46 192.168.29.2
-rw-r--r-- 1 root root   28 Aug 16 10:36 live_hosts.txt
root@kali:/home/kali/Desktop/PenetrationTesting/Horns&HoovesLLC# exit
exit
```

4.4 Allow to save all results into a Zip file.

```
# ZIP_RESULTS
# Offers the user an option to archive all scan results into a ZIP file named after the working directory.
# Provides a confirmation message when the archive is created.
ZIP_RESULTS() {
    while true; do
        # Ask user if they want to archive results
        read -p $'\n\e[31m[?]\e[0m\e[34m Save all results into a ZIP file? (Y/N): \e[0m' zip_choice
```

```
[?] Save all results into a ZIP file? (Y/N): y

[+] Results saved to: Horns&HoovesLLC.zip
```

```
┌──(root㉿kali)-[/home/kali/Desktop/PenetrationTesting]
└─# ll
total 544
drwxrwxr-x 2 kali kali   4096 Aug 13 08:58  data
drwxr-xr-x 4 root root   4096 Aug 16 10:38 'Horns&HoovesLLC'
-rw-r--r-- 1 root root  56876 Aug 16 10:48 'Horns&HoovesLLC.zip'
```

GitHub Repository