Penetration Testing Project : Vulner

Made by : daniel kovalevsky
Class : 7736/21
Teacher : arel regev

Quest Function:

The goal of this function is to act as menu where u can pick from the options given (basic scan , full scan , zip , exit)

```
function Quest()
{
printf ${LP}
figlet Network Scanning Script
printf ${E}

    echo -e "${LC}Made by Daniel Kov 🏴‍☠️ 🚩 ${E}"

echo "                    .-.-.-.-.-.-.          "
echo "                .-'-.-.-.-.-.-.-'-.        "
echo "              .'.-'.-.-.-.-.-.-.-'.'-.     "
echo "             /.'.-.-.-.-.-.-.-.-.-.'.\    "
echo "            /.'.-.-.-.-.-.-.-.-.-.-.'.\   "
echo "            |.'.-.-.-.-.-.-.-.-.-.-.'.|   "
echo "            \.'.-.-.-.-.-.-.-.-.-.-.'./   "
echo "             '.-.-.-.-.-.-.-.-.-.-.-.'    "
echo "              '-.-.-.-.-.-.-.-.-.-.-'     "
echo "                '-.-.-.-.-.-.-.-.-'       "
echo "          .-.-.-.-.-.-.-.-.-.-.-.-.       "
echo "        .-.-.-.-.-.-.-.-.-.-.-.-.-.-.     "
echo "      .-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.   "
echo "    .-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-. "
echo "   .-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-."
echo "  .-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-."
echo "  '-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-'  "

    echo -e "[?] Pick Your Choice : \n{${O}basic${E}} scan , {${O}full${E}} scan , {${O}zip${E}} the results , {${R}e${E}}xit"
        read OPTION
```

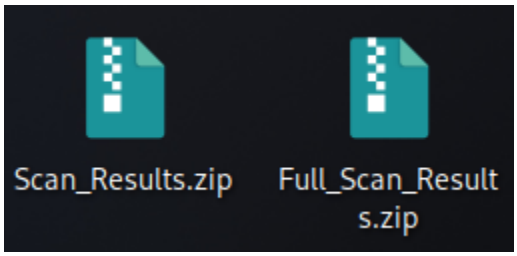
Case $Option in:

```
case $OPTION in

    basic)
    #calls For (Basic) Function
        BASIC

    ;;


    full)
    #calls for (FULL) Function
        FULL

    ;;

    zip)
        echo""
        echo -e "[${G}+${E}] Zipping The Scan Results :"
        #Zips the result.
        zip -r Full_Scan_Results.zip ./F_Scan &> /dev/null && find . -name Full_Scan_Results.zip
        zip -r Scan_Results.zip ./Scan &> /dev/null && find . -name Scan_Results.zip

    ;;
    e)
            echo -e "${LR}Exiting${E}"
            exit
    ;;

    *)
        echo -e "${LR}wrong input please pick again${E}"
        Quest
    ;;
esac
}
#calls for function "Quest"
Quest
```

Shows you the menu for which after receiving the output u choose calls for function **basic scan** or **full scan**

If you picked **zip** then it will zip all the information you gathered into a single and compressed zip by the scan you choose than using && if the zipping worked tell's u the name of the the zip

```
{basic} scan , {full} scan , {zip} the results , {e}xit
zip

[+] Zipping The Scan Results :
./Full_Scan_Results.zip
./Scan_Results.zip
```



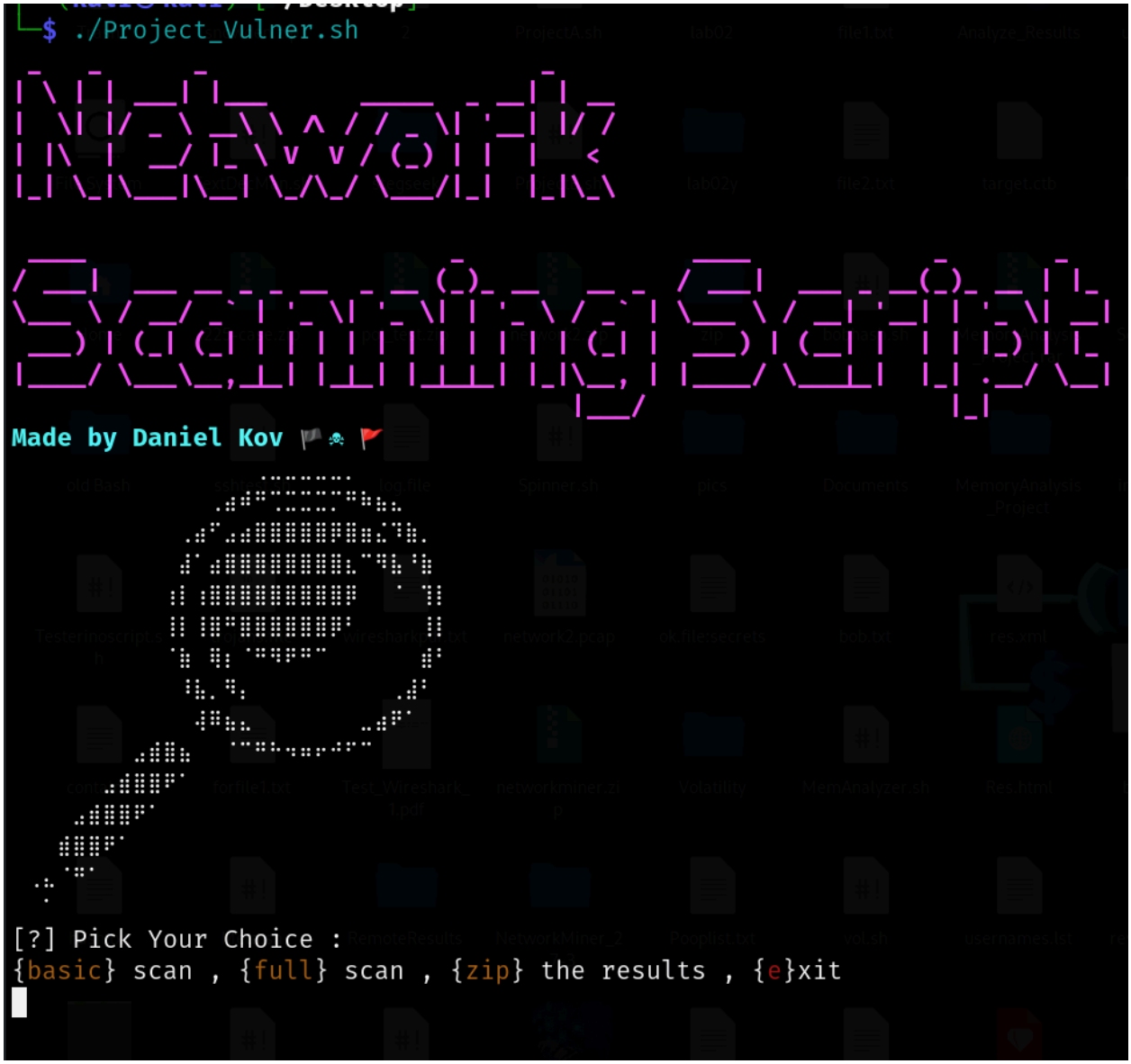
and if u will pick something that's not part of the menu it will ask you to pick again

```
{basic} scan , {full} scan , {zip} the results , {e}xit
test
wrong input please pick again
```

Exit will exit the script

```
{basic} scan , {full} scan , {zip} the results , {e}xit
e
Exiting
```

CLI Demonstration:

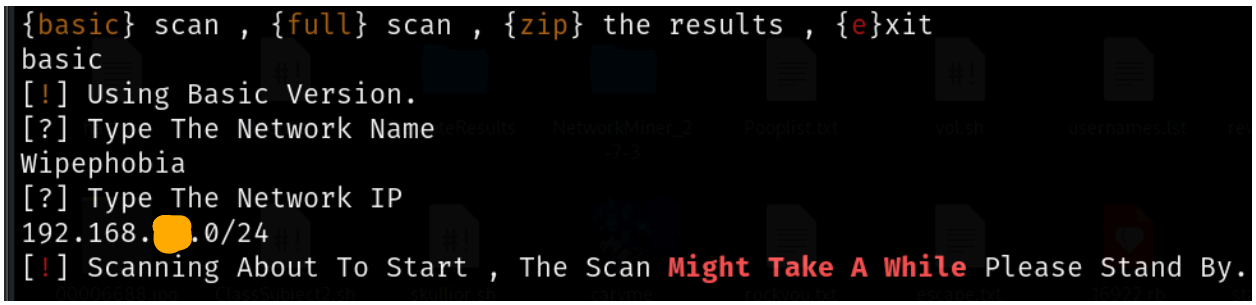


Function Basic:

The goal of this function is to scan your network , check which devices are active and what services are open , it also tries to find from a list of predetermined services which of these services have weak passwords

```
#scans the network and looking for weak passwords
function BASIC()
{
    echo -e "[${O}!${E}] Using Basic Version."
sleep 1
    echo "[?] Type The Network Name"
        read NAME
    echo "[?] Type The Network IP"
        read NETWORK

    #Makes The Folder
    mkdir -p $HOME/Scan/$NAME

    echo -e "Scan Date :\n $(date)" >> $HOME/Scan/$NAME/Scan_Date.txt

    #scans hosts that are up
    nmap $NETWORK -sn | grep for | awk '{print $5}' >> $HOME/Scan/$NAME/Active_Hosts.txt
    GATEWAY=$(route -n | awk '{print $2}' | head -3 | tail -1)
    grep -v "$GATEWAY" $HOME/Scan/$NAME/Active_Hosts.txt > $HOME/Scan/$NAME/Live_Hosts.txt

    echo -e "[${R}!${E}] Scanning About To Start , The Scan ${LR}Might Take A While${E} Please Stand By."
    echo ""
    #TCP Port Scanning
    for N in $(cat $HOME/Scan/$NAME/Live_Hosts.txt | awk '{print $1}')
    do

    mkdir $HOME/Scan/$NAME/$N
    echo ""

    echo -e "[!] Scanning IP : ${LC}$N${E}"
    nmap $N -sV -T3 -p- >> $HOME/Scan/$NAME/$N/Results_$N.txt

    cat $HOME/Scan/$NAME/$N/Results_$N.txt | grep -w "open" >> $HOME/Scan/$NAME/$N/OP_$N.txt
        echo -e "[${G}+${E}] Checking Common Services."
```

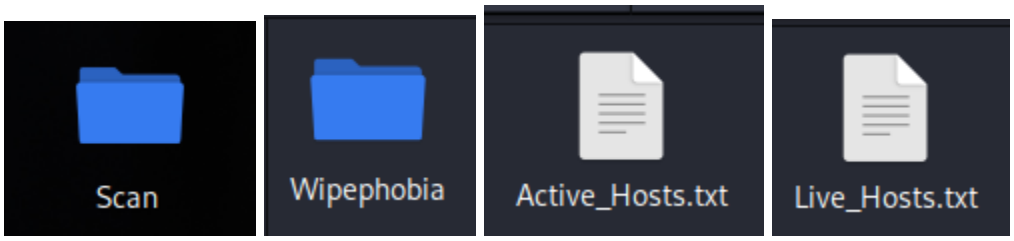
The function ask for your name , and network ip and removing your route Ip from the scan, Creating a folder base on the Network name u picked, than it informs you that the scan might take some time

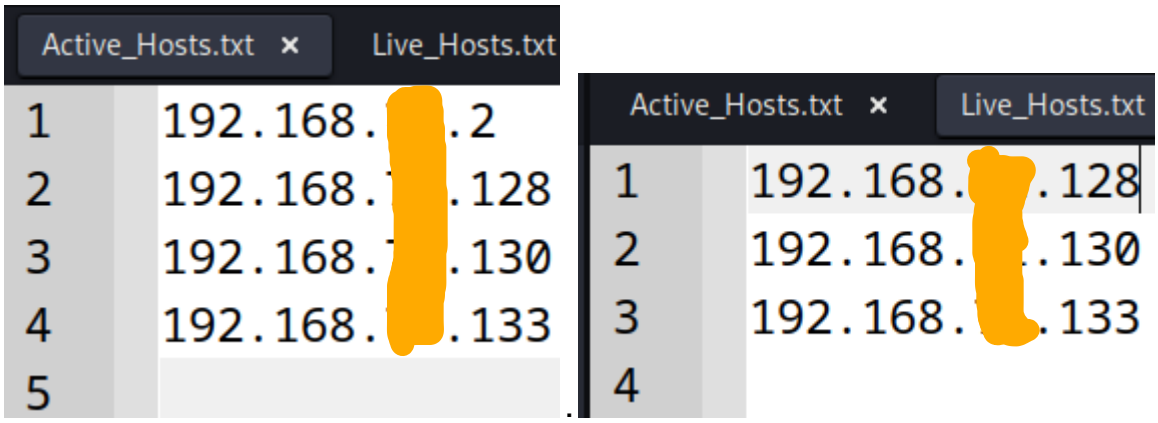


Removing The Route ip and continues the script with Live_Hosts.txt

```
mkdir -p $HOME/Scan/$NAME
```

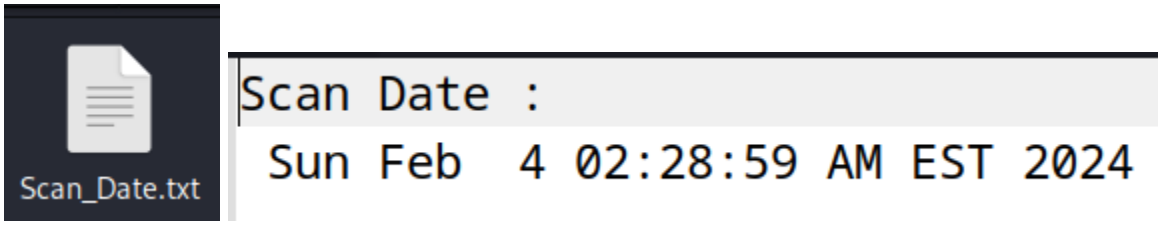
```
└─$ ls -l Scan
total 4
drwxr-xr-x 5 kali kali 4096 Feb  4 02:39 Wipephobia
```





Also creates a notepad with the date of when the scan occurred

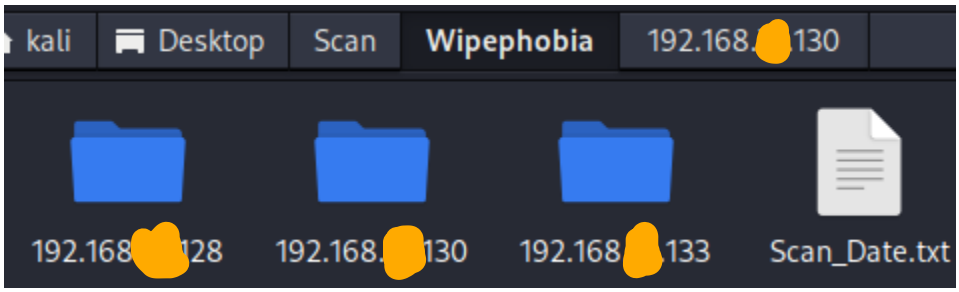
```
echo -e "Scan Date :\n $(date)" >> $HOME/Scan/$NAME/Scan_Date.txt
```



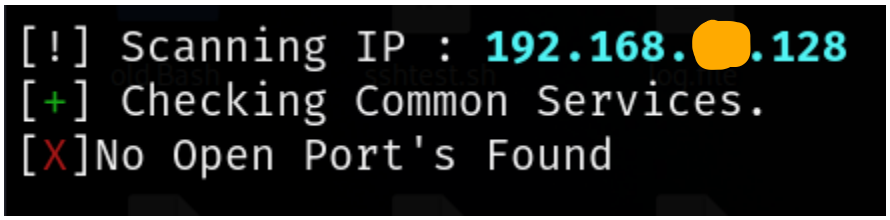
Then start's a loop that contains another two loops

```
for N in $(cat $HOME/Scan/$NAME/Live_Hosts.txt | awk '{print $1}')
do

mkdir $HOME/Scan/$NAME/$N
echo ""

echo -e "[!] Scanning IP : ${LC}$N${E}"
nmap $N -sV -T3 -p- >> $HOME/Scan/$NAME/$N/Results_$N.txt

cat $HOME/Scan/$NAME/$N/Results_$N.txt | grep -w "open" >> $HOME/Scan/$NAME/$N/OP_$N.txt
    echo -e "[${G}+${E}] Checking Common Services."

        if [ ! -s "$HOME/Scan/$NAME/$N/OP_$N.txt" ]; then
        echo -e "[${R}X${E}]No Open Port's Found"
    fi
```

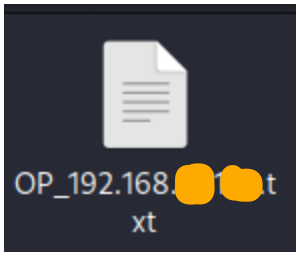
The First Loop that running till the end of the function
The loop running on Live_Hosts.txt which contains the scanned ip's
It creates subfolder named as the IP that is begin scanned



And inform the user if the ip contains open services ,



Then if the scanned ip contains open services , creates inside the folder
Txt file with the clean results after text manipulation of the opened services



OP_192.168 txt ×

```
1   21/tcp    open  ftp         vsftpd 2.3.4
2   22/tcp    open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
3   23/tcp    open  telnet      Linux telnetd
4   25/tcp    open  smtp        Postfix smtpd
```

And using it to initiate the next loop

```
            if [ ! -s "$HOME/Scan/$NAME/$N/OP_$N.txt" ]; then
            echo -e "[${R}X${E}]No Open Port's Found"
        fi

        # Initialize an array to keep track of found services
                SERVICES_FOUND=()

#Check if the following host got the common services
for SER in $SERVICES
do
    # Check if the service is present
    if grep -q "$SER" "$HOME/Scan/$NAME/$N/OP_$N.txt"; then
        echo -e "[${G}✓${E}] $SER was found" | tee -a  $HOME/Scan/$NAME/$N/open_services.txt
        # Keep track of the services and perform brute-forcing outside the loop
        SERVICES_FOUND+=("$SER")
    fi

done
```

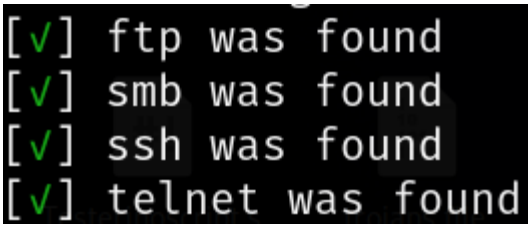
```
# Perform brute-forcing for each service found
for SERVICE_FOUND in "${SERVICES_FOUND[@]}"
do
    case "$SERVICE_FOUND" in
        "ftp")
            echo -e "[${G}+${E}] Using ${O}ftp-brute${E} script against the target..."
            nmap $N -Pn --script=$SCRIPT1 | grep -v "NSE" | grep -v "open" >> $HOME/Scan/$NAME/$N/Ftp_Cred.txt
            ;;
        "ssh")
            echo -e "[${G}+${E}] Using ${O}ssh-brute${E} script against the target..."
            nmap $N -Pn --script=$SCRIPT2 | grep -v "NSE" | grep -v "open" >> $HOME/Scan/$NAME/$N/Ssh_Cred.txt
            ;;
        "smb")
            echo -e "[${G}+${E}] Using ${O}smb-brute${E} script against the target..."
            nmap $N -Pn --script=$SCRIPT3 | grep -v "NSE" | grep -v "open" >> $HOME/Scan/$NAME/$N/Smb_Cred.txt
            ;;
        "telnet")
            echo -e "[${G}+${E}] Using ${O}telnet-brute${E} script against the target..."
            nmap $N -Pn --script=$SCRIPT4 | grep -v "NSE" | grep -v "open" >> $HOME/Scan/$NAME/$N/Telnet_Cred.txt
            ;;
    esac

done
done
echo -e "[${G}✓${E}] ${B}Scan Complete${E}"
echo""
Quest
}
```

Second loop *For SER*

With variable `SERVICES="ftp smb ssh telnet"`

If the txt file contains the open services from the variable then

It informs the user



And stores them to a new variable SERVICES_FOUND

Which Will start the third loop of the Brute Forcing

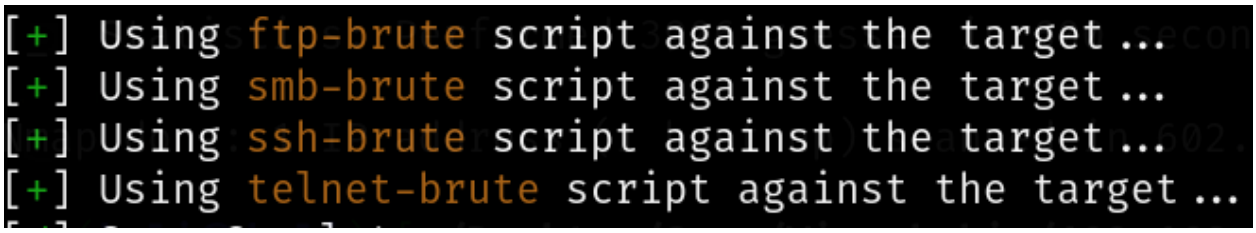
```
for SERVICE_FOUND in "${SERVICES_FOUND[@]}"
do
    case "$SERVICE_FOUND" in
        "ftp")
            echo -e "[${G}+${E}] Using ${O}ftp-brute${E} script against the target..."
            nmap $N -Pn --script=$SCRIPT1 | grep -v "NSE" | grep -v "open" >> $HOME/Scan/$NAME/$N/Ftp_Cred.txt
            ;;
        "ssh")
            echo -e "[${G}+${E}] Using ${O}ssh-brute${E} script against the target..."
            nmap $N -Pn --script=$SCRIPT2 | grep -v "NSE" | grep -v "open" >> $HOME/Scan/$NAME/$N/Ssh_Cred.txt
            ;;
        "smb")
            echo -e "[${G}+${E}] Using ${O}smb-brute${E} script against the target..."
            nmap $N -Pn --script=$SCRIPT3 | grep -v "NSE" | grep -v "open" >> $HOME/Scan/$NAME/$N/Smb_Cred.txt
            ;;
        "telnet")
            echo -e "[${G}+${E}] Using ${O}telnet-brute${E} script against the target..."
            nmap $N -Pn --script=$SCRIPT4 | grep -v "NSE" | grep -v "open" >> $HOME/Scan/$NAME/$N/Telnet_Cred.txt
            ;;
    esac

done
```

The Third Loop

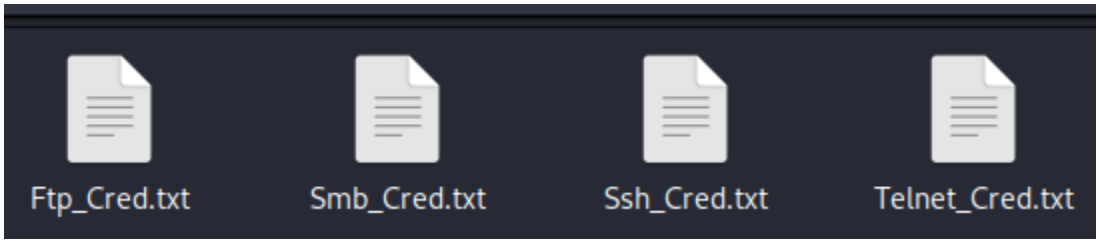
Using The Previous saved variable (SERVICES_FOUND)

And runs the Nmap With NSE script based on the saved services in the variable

```
SCRIPT1="ftp-brute.nse"
SCRIPT2="ssh-brute.nse"
SCRIPT3="smb-brute.nse"
SCRIPT4="telnet-brute.nse"
```



Then it save the results to a txt file with the output



Ftp_Cred.txt

```
1  | ftp-brute:
2  |   Accounts:
3  |     user:user - Valid credentials
4  |_  Statistics: Performed 3735 guesses in 601 seconds, average tps: 6.0
```

The Function ends with

```
echo -e "[${G}✓${E}] ${LP}Scan Complete${E}"
echo""
Quest
}
```

Informing that the scan ended
Which then redirect the user back to the menu by calling the *Quest* function where he can choose to scan again zip or exit the script

Function FULL

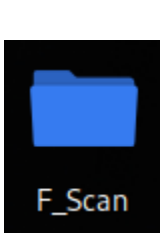
```
function FULL()
{
    echo -e "[${O}!${E}] Using Full Version."

sleep 1

    echo "[?] Type The Network Name"
        read NAME
    echo "[?] Type The Network IP"
        read NETWORK
    #makes the folder
    mkdir -p $HOME/F_Scan/$NAME

    echo -e "Scan Date :\n $(date)" >> $HOME/F_Scan/$NAME/Scan_Date.txt

    #scans hosts that are up and than removing your own route ip
    nmap $NETWORK -sn | grep for | awk '{print $5}' >> $HOME/F_Scan/$NAME/Active_Hosts.txt
    GATEWAY=$(route -n | awk '{print $2}' | head -3 | tail -1)
    grep -v "$GATEWAY" $HOME/F_Scan/$NAME/Active_Hosts.txt > $HOME/F_Scan/$NAME/Live_Hosts.txt

    echo "[!] Scanning About To Start , The Scan Might Take A While Please Stand By."
    echo ""
```

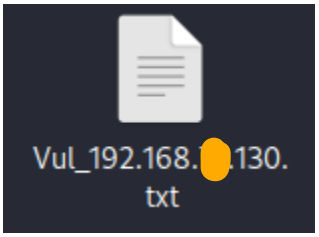
```
    for N in $(cat $HOME/F_Scan/$NAME/Live_Hosts.txt | awk '{print $1}')
    do

    mkdir $HOME/F_Scan/$NAME/$N

echo ""

    echo "[!] Scanning $N"

    nmap $N -sV -p- --script=vulners.nse >> $HOME/F_Scan/$NAME/$N/Vul_$N.txt

    cat $HOME/F_Scan/$NAME/$N/Vul_$N.txt | grep -w "open" >> $HOME/F_Scan/$NAME/$N/OP_$N.txt
    cat $HOME/F_Scan/$NAME/$N/Vul_$N.txt | grep -iw "CVE" >> $HOME/F_Scan/$NAME/$N/CVE_$N.tx
    echo -e "[${G}+${E}] Checking Common Services."

        if [ ! -s "$HOME/F_Scan/$NAME/$N/OP_$N.txt" ]; then
            echo -e "[${R}x${E}]No Open Port's Found"
        fi
        # Initialize an array to keep track of found services
                SERVICES_FOUND=()
```

```
#Check if the following host got the common services
for SER in $SERVICES
do
    # Check if the service is present
    if grep -q "$SER" "$HOME/F_Scan/$NAME/$N/OP_$N.txt"; then
        echo -e "[${G}✓${E}] $SER was found" | tee -a $HOME/F_Scan/$NAME/$N/open_services.txt
        # Keep track of the services and perform brute-forcing outside the loop
        SERVICES_FOUND+=("$SER")
    fi

done
```

```
# Perform brute-forcing for each service found
for SERVICE_FOUND in "${SERVICES_FOUND[@]}"
do
    case "$SERVICE_FOUND" in
        "ftp")
            echo -e "[${G}+${E}] Using ${O}ftp-brute${E} against the target..."
            nmap $N -Pn --script=$SCRIPT1 | grep -v "NSE" | grep -v "open" | grep "|" >> $HOME/F_Scan/$NAME/$N/Ftp_Cred.txt
            ;;
        "ssh")
            echo -e "[${G}+${E}] Using ${O}ssh-brute${E} against the target..."
            nmap $N -Pn --script=$SCRIPT2 | grep -v "NSE" | grep -v "open" | grep "|" >> $HOME/F_Scan/$NAME/$N/Ssh_Cred.txt
            ;;
        "smb")
            echo -e "[${G}+${E}] Using ${O}smb-brute${E} against the target..."
            nmap $N -Pn --script=$SCRIPT3 | grep -v "NSE" | grep -v "open" | grep "|" >> $HOME/F_Scan/$NAME/$N/Smb_Cred.txt
            ;;
        "telnet")
            echo -e "[${G}+${E}] Using ${O}telnet-brute${E} against the target..."
            nmap $N -Pn --script=$SCRIPT4 | grep -v "NSE" | grep -v "open" | grep "|" >> $HOME/F_Scan/$NAME/$N/Telnet_Cred.txt
            ;;
    esac

done
done
echo -e "[${G}✓${E}] ${LP}Scan Complete${E}"
echo""
Quest
}
```

The Full Function runs almost the same way as the basic version
With preforming Network scan , looking for open services , and trying to find weak passwords for the open services in addition the function also scans avabillable vulnarbilities and saves it to txt file

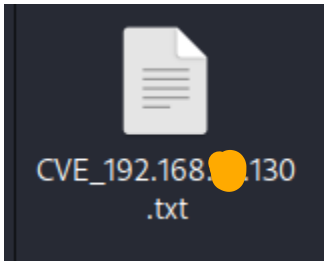





```
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-04 05:53 EST
Nmap scan report for 192.168.  .130
Host is up (0.00054s latency).
Not shown: 65534 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
| vulners:
|   cpe:/a:vsftpd:vsftpd:3.0.3:
|       PRION:CVE-2021-3618 5.8 https://vulners.com/prion/PRION:CVE-2021-3618
|_      PRION:CVE-2021-30047    5.0 https://vulners.com/prion/PRION:CVE-2021-30047
Service Info: OS: Unix

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.77 seconds
```

Then runs text manipulation and saves to new txt file:

```
cat $HOME/F_Scan/$NAME/$N/Vul_$N.txt | grep -iw "CVE" >> $HOME/F_Scan/$NAME/$N/CVE_$N.txt
```




```
1   |       PRION:CVE-2021-3618 5.8 https://vulners.com/prion/PRION:CVE-2021-3618
2   |_      PRION:CVE-2021-30047    5.0 https://vulners.com/prion/PRION:CVE-2021-30047
```