Wf project

Name : ariel sapir

Project: windows forensics – analyzer

Wf project

Some information about the tools that were used in the project:

Bulk-Extractor: This tool is a C++ application that scans disk images, files, or directories to extract useful information, bypassing file system structures. The extracted data is saved in feature files, which can be easily inspected or processed by automated tools. Additionally, Bulk-Extractor generates histograms of the features it discovers, highlighting commonly found and potentially important features.

Binwalk: Binwalk is designed to identify embedded files and executable code within binary images, such as firmware. It searches the binary for known file signatures and embedded data. Binwalk leverages the libmagic library, making it compatible with magic signatures, similar to those used by the Unix file utility.

Foremost: Foremost is a forensic tool used to recover files by searching for file headers, footers, and internal data structures. It can operate on disk images created by utilities like dd or forensic tools such as Encase, or even work directly on drives.

Strings: The strings command in Linux is used to extract printable character sequences from binary files. This is useful for uncovering readable text embedded in executable files or memory dumps, which would be difficult to identify manually.

Volatility: Volatility is used for analyzing memory dumps to uncover critical system information during an investigation. It can reconstruct active and closed network connections, list running processes, retrieve command prompts, and extract screenshots or clipboard data from the memory at the time of the dump. This tool helps in piecing together system activity during an incident.

Tcpdump: A network traffic capture tool that captures and saves network packets for analysis. In forensic cases, it helps extract network data from memory or disk images to investigate network activity.

Bc: A command-line calculator that supports both basic and advanced arithmetic operations. In the script, it could be used for handling precise calculations during analysis.

Tree: Displays the directory structure of a path in a tree-like format, making it easy to visualize file locations within the analyzed memory or HDD image.

Ent: A tool for calculating the entropy of files to detect randomness or encryption. It helps identify encrypted or compressed files by analyzing their entropy levels.

```
# Check if the user is running the script as root
if [ "$(id -u)" -ne 0 ]; then
    echo -e "This script must be run as root!"
    exit 1
fi
```

Checking if root.

Wf project

```
# Check for necessary tools and install if missing
function check_tools {
    tools=("bulk_extractor" "binwalk" "foremost" "strings" "tcpdump" "bc" "tree" "ent")
    for tool in "${tools[@]}"; do
        if ! command -v $tool &> /dev/null; then
            echo -e "$tool is not installed. Installing.."
            sudo apt-get install -y $tool >/dev/null
        else
            echo -e "$tool is already installed."
        fi
    done
}

check_tools
```

Function to check if necessary tools are installed.

```
# Get the user input for the file and the analysis type
echo -e "Enter the file path (Memory or HDD):"
read FILE

if [ ! -f "$FILE" ]; then
    echo -e "File not found! Exiting..."
    exit 1
fi

# Ask user to select file type for analysis (Memory or HDD)
echo -e "Select M[Memory File] or H[HDD File]:"
read SEL

# Start a timer to track the analysis duration
START_TIME=$(date +%s)

# Initialize the report file
REPORT="analysis_report.txt"
echo "Analysis Report" > $REPORT
echo "File analyzed: $FILE" >> $REPORT
echo "Analysis started at: $(date)" >> $REPORT
```

This is a code that take th name and type as input from the user and also create the repoort file.

```
# Function to extract network traffic
function extract_network_traffic {
    echo -e "Attempting to extract network traffic..." | tee -a $REPORT
    tcpdump -r $FILE -w extracted_network.pcap 2>/dev/null
    if [ -f "extracted_network.pcap" ]; then
        echo -e "Network traffic extracted and saved to extracted_network.pcap" | tee -a $REPORT
        echo -e "Size of extracted traffic: $(du -h extracted_network.pcap | cut -f1)" | tee -a $REPORT
    else
        echo -e "No network traffic found in the file." | tee -a $REPORT
    fi
}
```

Wf project

This is a function that is designed to extract network traffic from the file the user provided.

```
# Functions for Memory analysis
function BULK() {
    echo "Running Bulk-Extractor on Memory file..." | tee -a $REPORT
    bulk_extractor $FILE -o BulkMEM 1>/dev/null
    echo "Bulk-Extractor completed." | tee -a $REPORT
}

function BINWALK() {
    echo "Running Binwalk on Memory file..." | tee -a $REPORT
    binwalk $FILE > BinwalkMEM 2>/dev/null
    echo "Binwalk completed." | tee -a $REPORT
}

function FOREMOST() {
    echo "Running Foremost on Memory file..." | tee -a $REPORT
    foremost $FILE -o ForemostMEM 2>/dev/null
    echo "Foremost completed." | tee -a $REPORT
}

function STRINGS() {
    echo "Running Strings on Memory file..." | tee -a $REPORT
    strings $FILE > StringsMEM 2>/dev/null
    echo "Strings completed." | tee -a $REPORT
}

function VOL() {
    echo "Running Volatility on Memory file..." | tee -a $REPORT
    python3 vol.py -f $FILE imageinfo > VolMEM_profile.txt 2>/dev/null
    profile=$(grep "Suggested Profile" VolMEM_profile.txt | cut -d ':' -f2 | xargs)

    if [ -z "$profile" ]; then
        echo "Could not determine memory profile." | tee -a $REPORT
    else
        echo "Using profile: $profile" | tee -a $REPORT
        python3 vol.py -f $FILE --profile=$profile pslist > VolMEM_pslist.txt 2>/dev/null
        python3 vol.py -f $FILE --profile=$profile netscan > VolMEM_netscan.txt 2>/dev/null
        python3 vol.py -f $FILE --profile=$profile hivelist > VolMEM_registry.txt 2>/dev/null
        echo "Volatility analysis completed." | tee -a $REPORT
    fi
}
```

That’s all the functions that were used to analyze the memory file with wide set of tools.

Wf project

```
# Functions for HDD analysis
function BULK2() {
    echo "Running Bulk-Extractor on HDD file..." | tee -a $REPORT
    bulk_extractor $FILE -o BulkHDD 1>/dev/null
    echo "Bulk-Extractor completed." | tee -a $REPORT
}

function BINWALK2() {
    echo "Running Binwalk on HDD file..." | tee -a $REPORT
    binwalk $FILE > BinwalkHDD 2>/dev/null
    echo "Binwalk completed." | tee -a $REPORT
}

function FOREMOST2() {
    echo "Running Foremost on HDD file..." | tee -a $REPORT
    foremost $FILE -o ForemostHDD 2>/dev/null
    echo "Foremost completed." | tee -a $REPORT
}

function STRINGS2() {
    echo "Running Strings on HDD file..." | tee -a $REPORT
    strings $FILE > StringsHDD 2>/dev/null
    echo "Strings completed." | tee -a $REPORT
}

# Function to calculate entropy (to find encrypted/compressed files)
function calculate_entropy {
    echo -e "Calculating entropy of the files..." | tee -a $REPORT
    ent $FILE >> $REPORT
}
```

This is the functions that were used to analyze the HDD file with wide set of tools.

```
# log function for Memory analysis
function LOGMEM() {
    mkdir -p memory_results
    mv BulkMEM BinwalkMEM ForemostMEM StringsMEM VolMEM* memory_results 2>/dev/null

    echo "Generating statistics for memory analysis..." | tee -a $REPORT

    # Calculate more detailed statistics
    echo "Number of text files: $(find memory_results -name '*.txt' | wc -l)" | tee -a $REPORT
    echo "Number of executable files: $(find memory_results -name '*.exe' | wc -l)" | tee -a $REPORT
    echo "Number of zip files: $(find memory_results -name '*.zip' | wc -l)" | tee -a $REPORT
    echo "Number of image files: $(find memory_results -name '*.jpg' -or -name '*.png' -or -name '*.gif' | wc -l)" | tee -a $REPORT
    echo "Number of PDF files: $(find memory_results -name '*.pdf' | wc -l)" | tee -a $REPORT
    echo "Number of video files (mp4, avi): $(find memory_results -name '*.mp4' -or -name '*.avi' | wc -l)" | tee -a $REPORT

    # Largest files
    echo "Top 5 largest files:" | tee -a $REPORT
    find memory_results -type f -exec du -h {} + | sort -rh | head -5 | tee -a $REPORT

    echo "Calculating entropy for memory file..." | tee -a $REPORT
    calculate_entropy

    echo "Saving the results into a zip file..." | tee -a $REPORT
    zip -r memory_results.zip memory_results 1>/dev/null

    echo "Memory analysis results saved." | tee -a $REPORT
```

That is a function that show some statics from the memory file analysis including saving them to the report file.

Wf project

```
# log function for HDD analysis
function LOGHDD() {
    mkdir -p hdd_results
    mv BulkHDD BinwalkHDD ForemostHDD StringsHDD hdd_results 2>/dev/null

    echo "Generating statistics for HDD analysis..." | tee -a $REPORT

    # Calculate more detailed statistics
    echo "Number of text files: $(find hdd_results -name '*.txt' | wc -l)" | tee -a $REPORT
    echo "Number of executable files: $(find hdd_results -name '*.exe' | wc -l)" | tee -a $REPORT
    echo "Number of zip files: $(find hdd_results -name '*.zip' | wc -l)" | tee -a $REPORT
    echo "Number of image files: $(find hdd_results -name '*.jpg' -or -name '*.png' -or -name '*.gif' | wc -l)" | tee -a $REPORT
    echo "Number of PDF files: $(find hdd_results -name '*.pdf' | wc -l)" | tee -a $REPORT
    echo "Number of video files (mp4, avi): $(find hdd_results -name '*.mp4' -or -name '*.avi' | wc -l)" | tee -a $REPORT

    # Largest files
    echo "Top 5 largest files:" | tee -a $REPORT
    find hdd_results -type f -exec du -h {} + | sort -rh | head -5 | tee -a $REPORT

    echo "Calculating entropy for HDD file..." | tee -a $REPORT
    calculate_entropy

    echo "Saving the results into a zip file..." | tee -a $REPORT
    zip -r hdd_results.zip hdd_results 1>/dev/null

    echo "HDD analysis results saved." | tee -a $REPORT
}
```

That is a function that show some statics from the HDD file analysis including saving them to the report file.

```
# Execute analysis based on user selection
case $SEL in
    M)
        BULK
        BINWALK
        FOREMOST
        STRINGS
        VOL
        extract_network_traffic
        LOGMEM
        ;;
    H)
        BULK2
        BINWALK2
        FOREMOST2
        STRINGS2
        extract_network_traffic
        LOGHDD
        ;;
    *)
        echo -e "Invalid selection. Exiting..."
        exit 1
        ;;
esac
```

This code is for execute the analysis based on the user selection.

Wf project

```
# Calculate the time taken for analysis
END_TIME=$(date +%s)
ELAPSED_TIME=$(($END_TIME - $START_TIME))
echo -e "Analysis completed in $ELAPSED_TIME seconds." | tee -a $REPORT

# Save general statistics into the report file
echo "Analysis Summary" >> $REPORT
echo "Time taken: $ELAPSED_TIME seconds" >> $REPORT
echo "Results saved as: memory_results.zip or hdd_results.zip" >> $REPORT

echo -e "mAll results and report have been saved successfully."
```

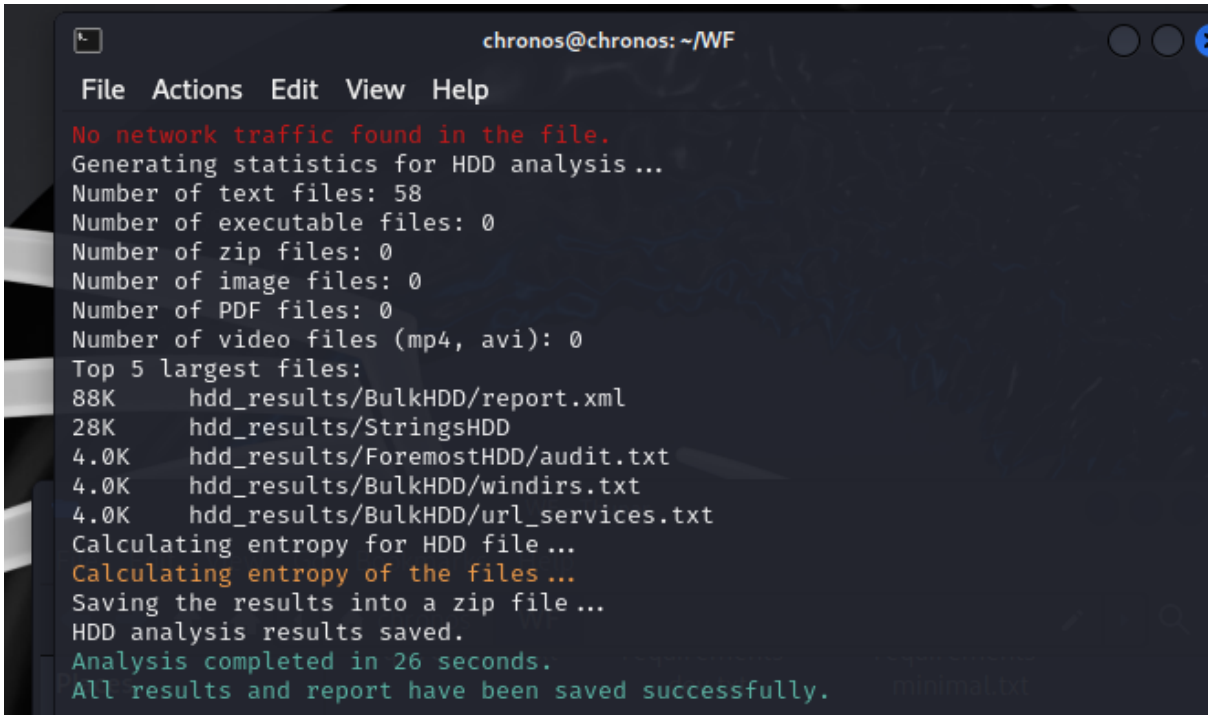
That's code is to calculate the time it took and save the general statistics.

Prooof of function:

```
chronos@chronos: ~/WF
File  Actions  Edit  View  Help

┌──(chronos㉿chronos)-[~/WF]
└─$ sudo ./WF.sh
bulk_extractor is already installed.
binwalk is already installed.
foremost is already installed.
strings is already installed.
tcpdump is already installed.
bc is already installed.
tree is already installed.
ent is already installed.
Enter the file path (Memory or HDD):
NTFS.E01
Select M[Memory File] or H[HDD File]:
H
Running Bulk-Extractor on HDD file ...
Bulk-Extractor completed.
Running Binwalk on HDD file ...
Binwalk completed.
Running Foremost on HDD file ...
Foremost completed.
Running Strings on HDD file ...
Strings completed.
Attempting to extract network traffic ...
No network traffic found in the file.
Generating statistics for HDD analysis ...
Number of text files: 58
```

Wf project


chronos@chronos: ~/WF
File Actions Edit View Help
No network traffic found in the file.
Generating statistics for HDD analysis ...
Number of text files: 58
Number of executable files: 0
Number of zip files: 0
Number of image files: 0
Number of PDF files: 0
Number of video files (mp4, avi): 0
Top 5 largest files:
88K hdd_results/BulkHDD/report.xml
28K hdd_results/StringsHDD
4.0K hdd_results/ForemostHDD/audit.txt
4.0K hdd_results/BulkHDD/windirs.txt
4.0K hdd_results/BulkHDD/url_services.txt
Calculating entropy for HDD file ...
Calculating entropy of the files ...
Saving the results into a zip file ...
HDD analysis results saved.
Analysis completed in 26 seconds.
All results and report have been saved successfully.