R V COLLEGE OF ENGINEERING

Name: Dhanush M **USN:** 1RV18IS011 **Dept/Lab:** ISE/CSDF **Expt No.:** 06 b
Date: 08/12/2021 **Title:** FORENSICS TOOLS

---

b. BINWALK

INTRODUCTION

Firmware analysis is the process of recovering, extracting, and analyzing the contents of a firmware. A firmware here refers to a software or operating system running on an embedded device like a router, camera, refrigerator etc.
Binwalk is a tool for searching a given binary image for embedded files and executable code. Specifically, it is designed for identifying files and code embedded inside of firmware images. Binwalk uses the libmagic library, so it is compatible with magic signatures created for the Unix file utility.
Binwalk also includes a custom magic signature file which contains improved signatures for files that are commonly found in firmware images such as compressed/archived files, firmware headers, Linux kernels, bootloaders, filesystems, etc.

Objectives - To use a firmware image for forensics analysis.

EXECUTION STEPS

1. **Installing Binwalk from a package**
   Command - *sudo apt install binwalk*

2. **Basic Structure**
   Syntax - *binwalk [options] [file1] [file2] [file3] …*

Example cases

1. **Scanning Firmware / To scan and identify code, files, and other information**
   Binwalk can scan a firmware image for many different embedded file types and file systems just by giving it a list of files to scan
   Command - *binwalk firmware.bin*

   a. **Signature Analysis** (*-B*, *--signature*)
      Signature scanning is the most popular use of binwalk. This argument is used as default if no other analysis options are specified.
      Command - *binwalk -B firmware*

```
└─$ binwalk -B WhatsAppSetup.exe

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             Microsoft executable, portable (PE)
143216        0x22F70         Copyright string: "Copyright 1998 Gilles Vollant "
143248        0x22F90         CRC32 polynomial table, little endian
161579        0x2772B         mcrypt 2.5 encrypted data, algorithm: "sProcessorFeaturePresent", keysize:
702 bytes, mode: "G",
167104        0x28CC0         Zip archive data, at least v2.0 to extract, compressed size: 285468, uncomp
ressed size: 287824, name: background.gif
452616        0x6E808         Zip archive data, at least v2.0 to extract, compressed size: 83, uncompress
ed size: 82, name: RELEASES
452737        0x6E881         Zip archive data, at least v2.0 to extract, compressed size: 104981, uncomp
ressed size: 415922, name: setupIcon.ico
557761        0x882C1         Zip archive data, at least v2.0 to extract, compressed size: 688008, uncomp
ressed size: 1835728, name: Update.exe
1245809       0x130271        Zip archive data, at least v2.0 to extract, compressed size: 134112093, unc
ompressed size: 134705555, name: WhatsApp-2.2146.9-full.nupkg
135358263     0x8116737       End of Zip archive, footer length: 22
135725940     0x8170374       PNG image, 256 x 256, 8-bit/color RGBA, non-interlaced
```

2. **File Extraction**
   a. **Extract files from firmware** (*-e*, *--extract*)
   This option is used to find any files found in the firmware image.
   Command - *binwalk -e firmware.bin*

   b. **Extract files from firmware recursively** (*-M*)
   This option recursively extracts files during a *--signature* scan. Only valid when used with *--extract* or *--dd*
   Command - *binwalk -Me firmware.bin*

   c. **Extract specific signature types** (*-D*)
   Command - *binwalk -D 'png image:png' firmware.bin*

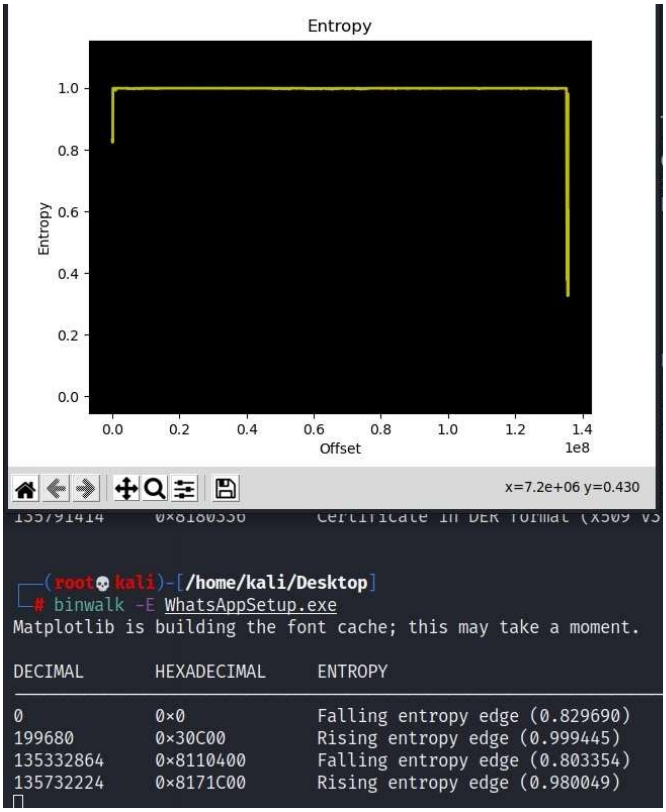
```
└─$ binwalk -D 'png image:png' WhatsAppSetup.exe

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             Microsoft executable, portable (PE)
143216        0x22F70         Copyright string: "Copyright 1998 Gilles Vollant "
143248        0x22F90         CRC32 polynomial table, little endian
161579        0x2772B         mcrypt 2.5 encrypted data, algorithm: "sProcessorFeaturePresent", keysize: 702 bytes, mode: "G",
167104        0x28CC0         Zip archive data, at least v2.0 to extract, compressed size: 285468, uncompressed size: 287824,
name: background.gif
452616        0x6E808         Zip archive data, at least v2.0 to extract, compressed size: 83, uncompressed size: 82, name: RE
LEASES
452737        0x6E881         Zip archive data, at least v2.0 to extract, compressed size: 104981, uncompressed size: 415922,
name: setupIcon.ico
557761        0x882C1         Zip archive data, at least v2.0 to extract, compressed size: 688008, uncompressed size: 1835728,
 name: Update.exe
1245809       0x130271        Zip archive data, at least v2.0 to extract, compressed size: 134112093, uncompressed size: 13470
5555, name: WhatsApp-2.2146.9-full.nupkg
135358263     0x8116737       End of Zip archive, footer length: 22
135725940     0x8170374       PNG image, 256 x 256, 8-bit/color RGBA, non-interlaced
135777207     0x817CBB7       XML document, version: "1.0"
135784968     0x817EA08       Object signature in DER format (PKCS header length: 4, sequence length: 8895
135785137     0x817EAB1       Certificate in DER format (x509 v3), header length: 4, sequence length: 1321
135786462     0x817EFDE       Certificate in DER format (x509 v3), header length: 4, sequence length: 951
135787417     0x817F399       Certificate in DER format (x509 v3), header length: 4, sequence length: 1328
135788749     0x817F8CD       Certificate in DER format (x509 v3), header length: 4, sequence length: 1328
135790081     0x817FE01       Certificate in DER format (x509 v3), header length: 4, sequence length: 1329
135791414     0x8180336       Certificate in DER format (x509 v3), header length: 4, sequence length: 1278
```

   d. **Extracted undetected files** (*-r*)
   Any file signatures that couldn't be extracted or those that resulted in 0-size files will be automatically deleted
   Command - *binwalk -Mre firmware.bin*

3. **Entropy Analysis**

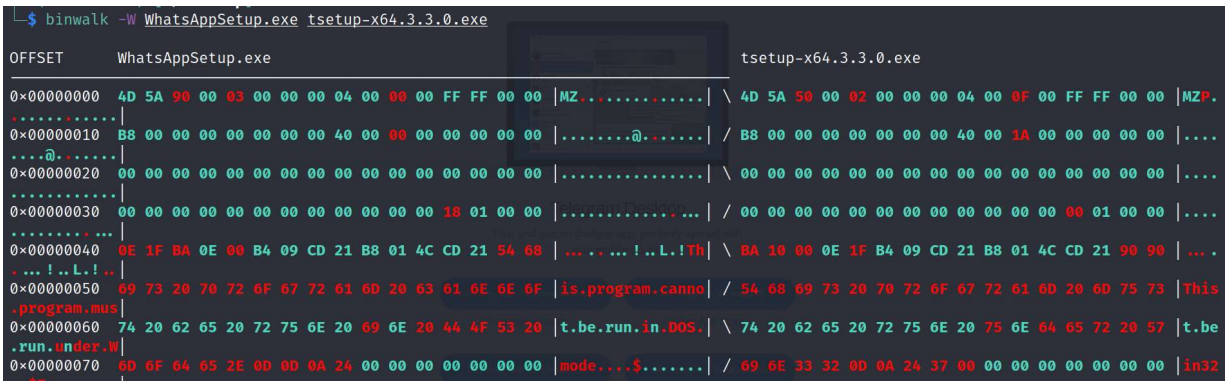
Entropy analysis can help identify interesting sections of data inside a firmware image. Low entropy signifies encryption mechanisms may not be implemented while high entropy signifies the availability of an encryption mechanism.

Command - *binwalk -E firmware.bin*



- Generate differences between firmware images

  Command - *binwalk -W firmware1.bin firmware2.bin*



- Verbose output

  Command - *binwalk --verbose firmware.bin*

- Capturing log files
  Command - *binwalk -f file.log firmware.bin*

```
*/home/kali/Desktop/file.log - Mousepad
File  Edit  Search  View  Document  Help
=
DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             Microsoft executable, portable (PE)
143216        0x22F70         Copyright string: "Copyright 1998 Gilles Vollant "
143248        0x22F90         CRC32 polynomial table, little endian
161579        0x2772B         mcrypt 2.5 encrypted data, algorithm: "sProcessorFeSSaturePresent", keysize: 702 bytes, mode: "G",
167104        0x28CC0         Zip archive data, at least v2.0 to extract, compressed size: 285468, uncompressed size: 287824, name: back
452616        0x6E808         Zip archive data, at least v2.0 to extract, compressed size: 83, uncompressed size: 82, name: RELEASES
452737        0x6E881         Zip archive data, at least v2.0 to extract, compressed size: 104981, uncompressed size: 415922, name: setu
557761        0x882C1         Zip archive data, at least v2.0 to extract, compressed size: 688008, uncompressed size: 1835728, name: Upd
1245809       0x130271        Zip archive data, at least v2.0 to extract, compressed size: 134112093, uncompressed size: 134705555, name
135358263     0x8116737       End of Zip archive, footer length: 22
135725940     0x8170374       PNG image, 256 x 256, 8-bit/color RGBA, non-interlaced
135777207     0x817CBB7       XML document, version: "1.0"
135785137     0x817EAB1       Certificate in DER format (x509 v3), header length: 4, sequence length: 1321
135786462     0x817EFDE       Certificate in DER format (x509 v3), header length: 4, sequence length: 951
135787417     0x817F399       Certificate in DER format (x509 v3), header length: 4, sequence length: 1328
135788749     0x817F8CD       Certificate in DER format (x509 v3), header length: 4, sequence length: 1328
135790081     0x817FE01       Certificate in DER format (x509 v3), header length: 4, sequence length: 1329
135791414     0x8180336       Certificate in DER format (x509 v3), header length: 4, sequence length: 1278
```

- To display file system of a binary
  Command - *binwalk -y 'filesystem' firmware.bin*

- To extract the firmware recursively and decompress the file
  Command - *binwalk -reM firmware.bin*

- To display CPU architecture
  Command - *binwalk --disasm firmware.bin*

```
└─$ binwalk --disasm WhatsAppSetup.exe

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
18            0x12            ARM executable code, 16-bit (Thumb), big endian, at least 515 valid instructions
```

CONCLUSION

1. This forensics tool is used to analyze and extract firmware images and help in identifying code, files, and other information embedded in the binary image of firmware.
2. Binwalk uses a libmagic library and custom magic signature file, which makes it more effective in analyzing executable binaries.

REFERENCES

1. BinWalk - https://www.kali.org/tools/binwalk/
2. BinWalk - https://en.kali.tools/?p=1634
3. Analyzing Firmware image using Binwalk - https://blog.pentesteracademy.com/analyzing-firmware-image-using-binwalk-a6e8277310dc
4. Tutorial: Firmware Analysis Tool using Binwalk - https://allabouttesting.org/short-tutorial-firmware-analysis-tool-binwalk/