REMnux Tutorial-1: Statically Examine Portable Executables(PE) Files

Rhydham Joshi

M.S. in Software Engineering, San Jose State University

Phone : (+1) 408-987-1991 | Email : rhydham.joshi@yahoo.com

Blog : malwareforensics1.blogspot.com | Linkedin : www.linkedin.com/in/rhydhamjoshi

Contents:

REMnux: A Linux Toolkit for Reverse-Engineering and Analyzing Malware

- **REMnux is a free, lightweight Linux (Ubuntu distribution) toolkit for reverse-engineering malicious software.**
- REMnux provides the collection of some of the most common and effective tools used for reverse engineering malwares in categories like:

    1) Investigate Linux malwares
    2) Statically analyze windows executable file
    3) Examine File properties and contents
    4) Multiple sample processing
    5) Memory Snapshot Examination
    6) Extract and decode artifacts
    7) Examine Documents
    8) Browser Malware Examination
    9) Network utilities

- For more information about REMnux, navigate to my blog at: http://malwareforensics1.blogspot.com/2015/04/the-power-of-remnux-linux-toolkit-for.html

Entropy

- Entropy is a measure of the unpredictability of an information stream. A perfectly consistent stream of bits (all zeroes, or all ones) is totally predictable (has no entropy). A stream of completely unpredictable bits has maximum entropy.
- Range of entropy for 8 byte value : 0 (No entropy) -> 8(Maximum entropy)
- better is the encryption or packing, Higher will be the entropy level.

$$H = -\sum_{i=0}^{255} P_i \log_2(P_i)$$

Use of Entropy in detecting malware:

- Entropy can be used is many different way, but quite commonly to detect encryption and compression, since truly random data is not common in typical user data.
- Encrypted or packed data prevents an AV engine from seeing "inside" the executables and so the level of entropy will be high which may trigger an signal to malware investigator about it.
- It is also very helpful in identifying files that have a high-amount of randomness which could indicate an encrypted container/volume that may go otherwise unnoticed.

Unpacking -> UPX

- UPX is distributed with full source code under the GNU General Public License v2+ for **packing/unpacking** executables.
- UPX provides **excellent compression ratio** and **very fast decompression**.
- It provides **in-place decompression** so the executable suffers from no memory overhead.
- It is **safe, universal** and **portable** and supports various executable formats.
- It command line utility available for win32/linux and it is used to pack the malware executables.

```
C:\Users\Rhydham\Downloads\upx391w\upx391w>upx.exe
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2013
UPX 3.91w       Markus Oberhumer, Laszlo Molnar & John Reiser   Sep 30th 2013

Usage: upx [-123456789dlthVL] [-qvfk] [-o file] file..

Commands:
  -1     compress faster                    -9    compress better
  -d     decompress                         -l    list compressed file
  -t     test compressed file               -V    display version number
  -h     give more help                     -L    display software license
Options:
  -q     be quiet                           -v    be verbose
  -oFILE write output to 'FILE'
  -f     force compression of suspicious files
  -k     keep backup files
file..   executables to (de)compress

Type 'upx --help' for more detailed help.

UPX comes with ABSOLUTELY NO WARRANTY; for details visit http://upx.sf.net

C:\Users\Rhydham\Downloads\upx391w\upx391w>
```

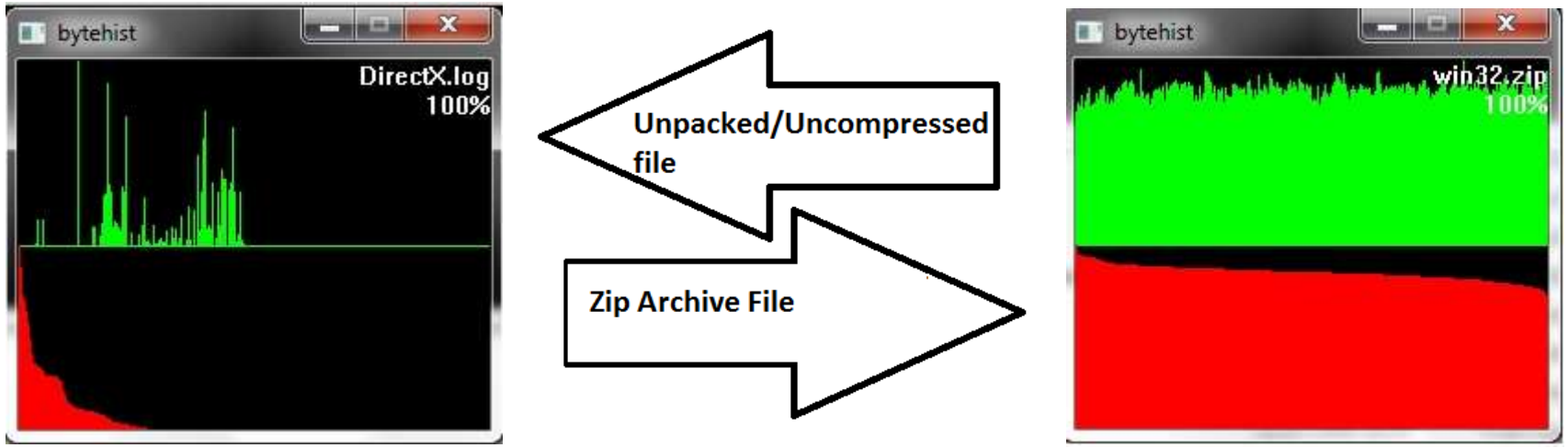
Unpacking -> ByteHist

- A tool for generating byte-usage-histograms for all types of files with a special focus on binary executables in PE-format (Windows).

Features:

- Makes byte-usage-histograms of any file of any size
- Histograms are generated as sorted and unsorted diagrams
- Sub-histograms for each section of binary executables (PE)
- Quick overview with GUI navigation in case of sub-histograms
- Percentage for the share in the total filesize for sub-histograms

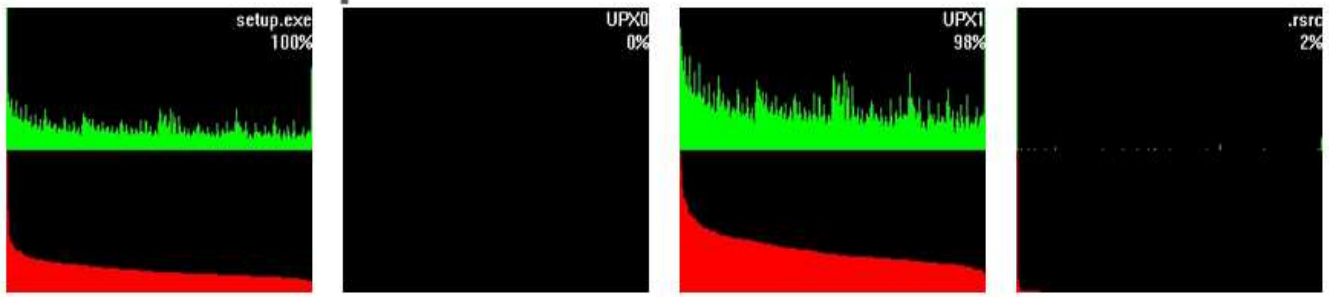
“The byte-distribution of unencrypted and unpacked clear text, database-files and executable binaries differs massively as compared to the encrypted and/or packed ones. By putting this "phenomenon" into a picture this difference can be easily visualized by histograms.”



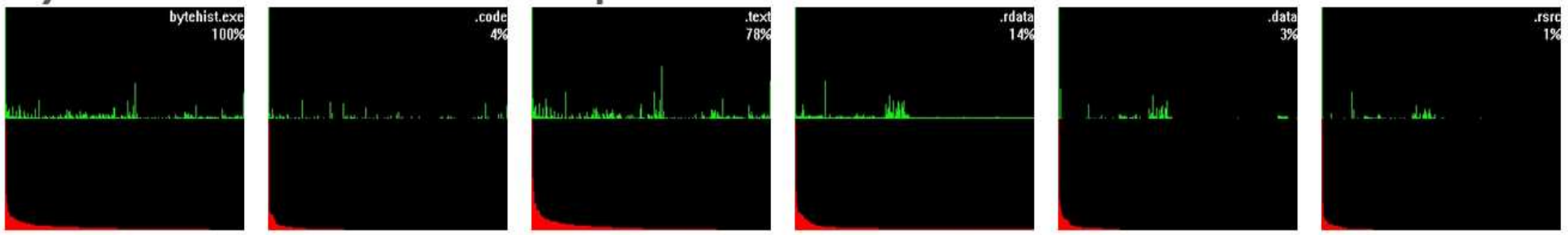
Unpacking -> ByteHist

- The tall green bar on the most left side tells represents pixel-column for **0h byte-code** and the most right side represents **FFh byte-code.**
- Red section arranges the pixel-columns in descending order.
- Section-wise distribution tells us about which section we need to analyze. This feature gives a reverser the possibility to instantly find out the section that's containing (if so) packed/encrypted data.

An UPX packed executable:



bytehist itself - unpacked:

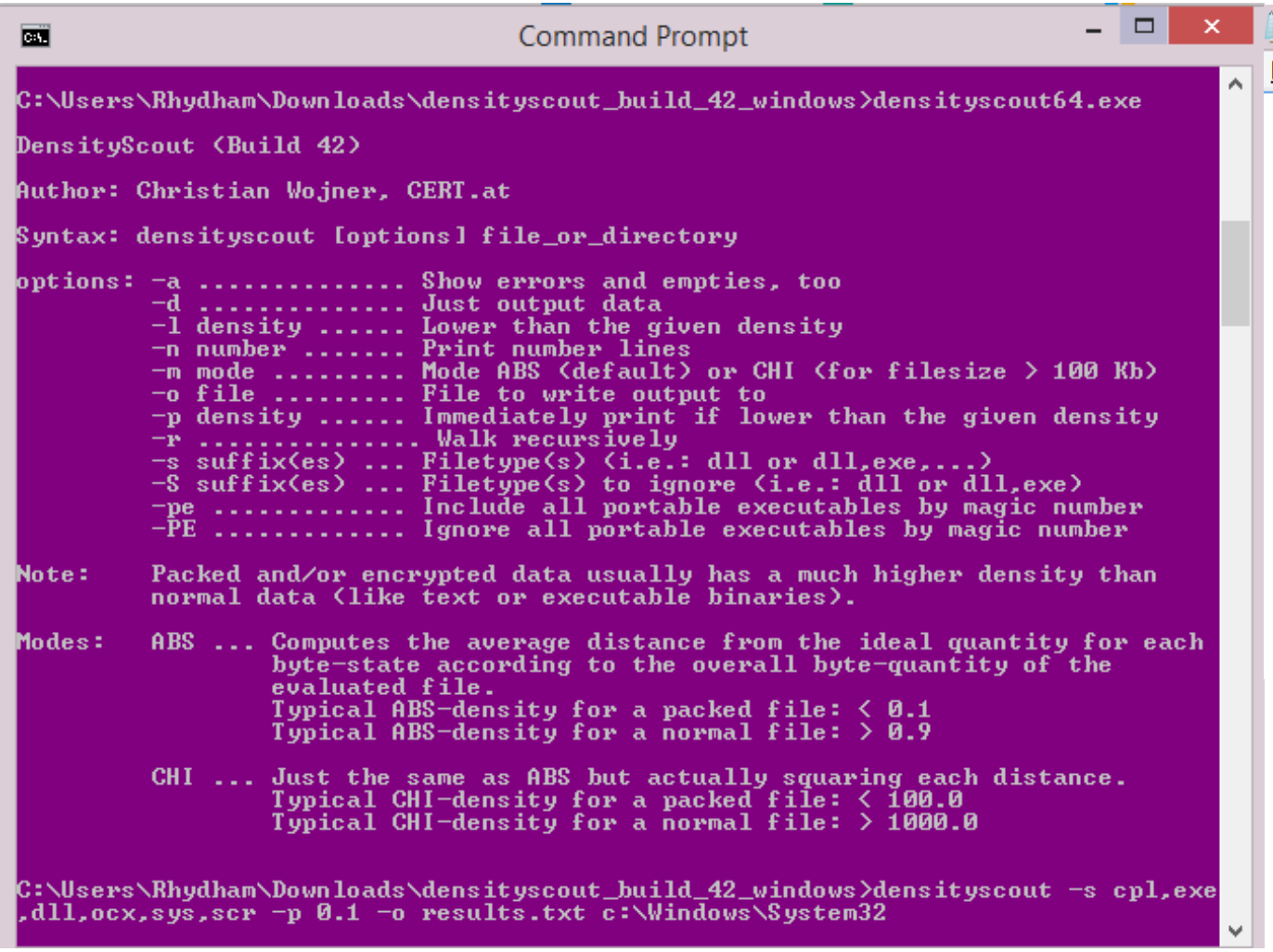


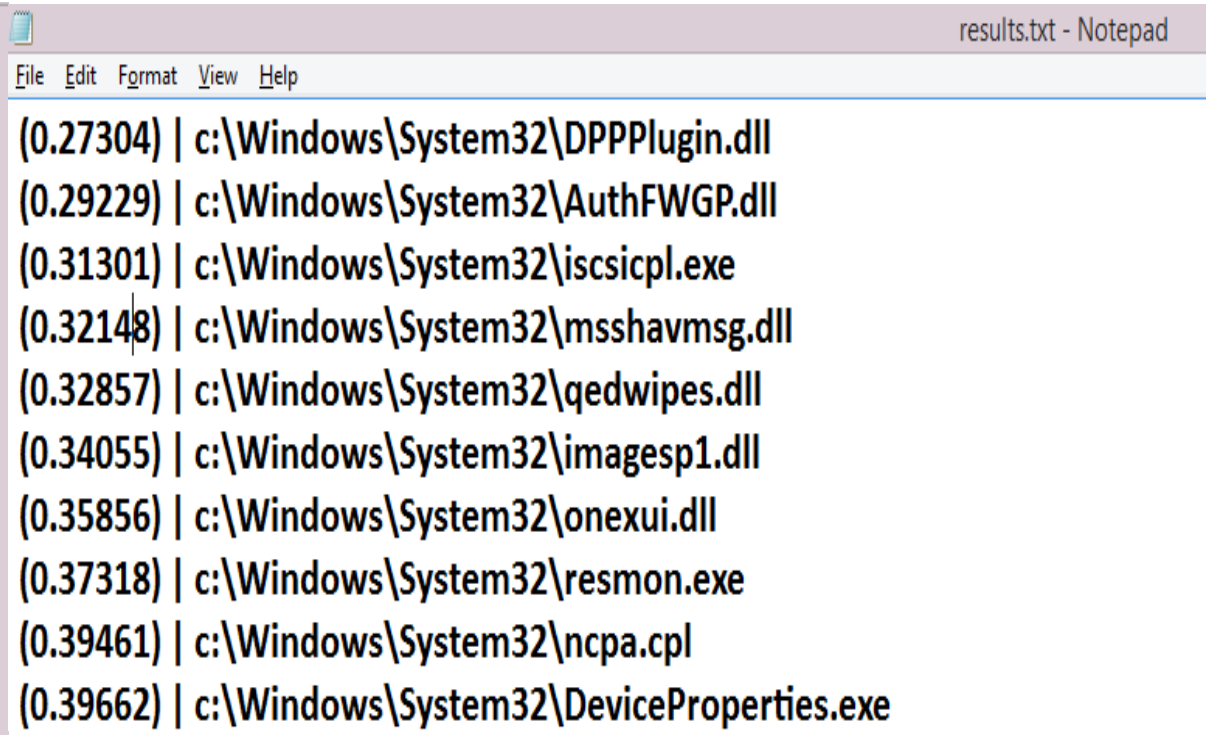
Unpacking -> Density Scout

- Density Scout is a tool that has been written for one purpose: finding (eventually unknown) malware on a potentially infected system.
- Therefore it takes advantage of the typical approach of malware authors to protect their "products" with obfuscation like run-time-packing and -encryption.
- It is based on the concept of Bytehist.
- Density Scout's main focus is to scan a desired file-system-path by calculating the density of each file to finally print out a descending list.
- Usually Microsoft Windows executables are not packed or encrypted in any way which throws the hits of malicious executables to the top of the list where you can easily focus on.

Unpacking -> Density Scout

Max Entropy for C:\Windows\System32 folder in my system is 1.84094





Anomaly detection -> PEScanner

pescanner.py is a PE analyzer written in python.

The script has the ability to detect:

- Files with TLS entries
- Files with resource directories
- Suspicious IAT entries
- Suspicious entry point sections
- Sections with zero-length raw sizes
- Sections with extremely low or high entropy
- Invalid timestamps
- File version information

Among other things, this script is helpful to:

- understand the behavior of an executable
- classify malwares (UPX packed, trojan downloader, trojan dropper, ...)

Anomaly detection -> PEScanner

```
remnux@remnux:~$ pescanner /home/remnux/Desktop/malwares/kroker.exe
################################################################################
Record 0
################################################################################

Meta-data
================================================================================
File:     /home/remnux/Desktop/malwares/kroker.exe
Size:     172032 bytes
Type:     PE32 executable for MS Windows (GUI) Intel 80386 32-bit
MD5:      7d5fe48df650bb98454d1613b4f9444a
SHA1:     6c313a3f8f3ded918dda034a47c4b0be05c6884c
ssdeep:   3072:YBntkCwu0I8Y0I5RgINhMcPR0kdzAdC0+14gfG2h5lB2y7UgzIQ7:LFu+EHgIFR0eAAV1B0YDEre
Date:     0x4AAB9BE7 [Sat Sep 12 13:02:31 2009 UTC]
EP:       0x4073a0 .text 0/5
CRC:      Claimed: 0x0, Actual: 0x345f5 [SUSPICIOUS]
Packers:  Xtreme-Protector v1.05

Signature scans
================================================================================

Clamav:  /home/remnux/Desktop/malwares/kroker.exe: Trojan.Packed-182 FOUND

Suspicious IAT alerts
================================================================================
VirtualAllocEx

Sections
================================================================================
Name        VirtAddr     VirtSize     RawSize      Entropy
--------------------------------------------------------------------------------
.text       0x1000       0xab6c       0xac00       4.524299
.data       0xc000       0xe74        0x1000       4.560540
BSS         0xd000       0x3908d      0x1d200      7.465702   [SUSPICIOUS]
DATA        0x47000      0x384        0x400        0.000000
.rdata      0x48000      0x6ee        0x800        0.204859   [SUSPICIOUS]

remnux@remnux:~$
```

MD5 Values can be used to check executable in Virus Total/ThreatExpert /etc

Difference in CRC arises Suspicious

Packed executable might be suspicious

ClamAV scan reports malware

Suspicious Internet Address Table calls

PEScanner marks as Suspicious

PEScanner marks as Suspicious

Anomaly Detection -> ExeScan

- ExeScan is the FREE command-line tool to detect anomalies in PE (Portable Executable or EXE/DLL) files.
- It instantly scans EXE/DLL file and reports all kind of abnormalities in the PE header fields such as checksum differences, header field sizes, non-ascii/empty section names, improper size of raw data etc.
- Typically Malwares use packers/protectors to pack their EXE.
- These packers modify PE header fields in EXE file to make reverse engineering of these malwares difficult.
- E.g. : These anomalies in PE header can crash debugger thus preventing any attempt to reversing. Exe-scan becomes handy in such situations.

- Features of EXE-Scan :

  * Instantly detect all kind of abnormalities in EXE/PE file.
  * Detect the type of Compiler/Packer used in the PE file.
  * Scan for commonly used malware APIs
  * Great for automation
  * Displays PE header and Import table structures
  * Generate detailed analysis report

Anomaly Detection -> ExeScan

```
remnux@remnux:~/Desktop/ExeScan$ exescan.py -a /home/remnux/Desktop/kroker.exe

[+] File: /home/remnux/Desktop/kroker.exe

        [*] MD5         : 7d5fe48df650bb98454d1613b4f9444a
        [*] SHA-1       : 6c313a3f8f3ded918dda034a47c4b0be05c6884c
        [*] SHA-256     : 3d15678be304895209258d1ab7af77fb65d78ca8685d8dc3027b2c8a1dd473c4

[+] File Type: EXE

[+] Signature [Compiler/Packer]

        ['Xtreme-Protector v1.05']

[+] Address of entry point      : 0x000073a0

[+] Image Base Address          : 0x00400000

[+] Sections
        Name: .text     Virtual Address: 0x00001000     Size: 0x0000ab6c        Entropy: 4.524299
        Name: .data     Virtual Address: 0x0000c000     Size: 0x00000e74        Entropy: 4.560540
        Name: BSS       Virtual Address: 0x0000d000     Size: 0x0003908d        Entropy: 7.465702
        Name: DATA      Virtual Address: 0x00047000     Size: 0x00000384        Entropy: 0.000000
        Name: .rdata    Virtual Address: 0x00048000     Size: 0x000006ee        Entropy: 0.204859

[+] Anomalies Check

        [*] Based on the sections entropy check! file is possibly packed
        [*] Header Checksum is zero!
```

```
[+] Anomalies Check

        [*] Based on the sections entropy check! file is possibly packed
        [*] Header Checksum is zero!

[+] Following expected Malware APIs are Detected

        [-] Import Table

                IA: 0x0040c168  CreateThread
                IA: 0x0040c140  GetFileSize
                IA: 0x0040c194  GetProcAddress
                IA: 0x0040c1d8  GetTickCount
                IA: 0x0040c1e8  GetModuleFileNameA
                IA: 0x0040c1f4  GetStartupInfoA
                IA: 0x0040c1c8  LockResource
                IA: 0x0040c178  Sleep
                IA: 0x0040c16c  VirtualAllocEx

        [-] Entire Executable

                1 times         CreateThread
                1 times         FindWindow
                1 times         GetCommandLine
                1 times         GetFileSize
                1 times         GetModuleHandle
                1 times         GetProcAddress
                1 times         GetTickCount
                1 times         GetModuleFileNameA
                1 times         GetStartupInfoA
                1 times         LoadLibrary
                1 times         LockResource
                1 times         Sleep
                1 times         VirtualAlloc
                1 times         VirtualAllocEx
remnux@remnux:~/Desktop/ExeScan$
```

Anomaly Detection -> PEFrame

- Dowload PEFrame in linux using following command:

  pip install https://github.com/guelfoweb/peframe/archive/master.zip

- **It lists different sections for anomaly detection like :**

  - **MD5 Hash Value,**
  - **XOR Discovered**
  - **Digital Signature**
  - **Packers**
  - **Anti Debug code**
  - **Anti VM tricks**
  - **Suspicious API**
  - **Suspicious Sections**
  - **Files, URL and Metadata names**
  - **Imports/exports of file**
  - **Strings**
  - **Dump**

```
remnux@remnux:~$ peframe
PEframe v.4.2 - Open Source Project
Author: Gianni 'guelfoweb' Amato
Github: https://github.com/guelfoweb/peframe

Usage
     peframe.py malware.exe
     peframe.py [--option] malware.exe

Option
     --json          Output in json

     --import        Imported DLL and functions
     --export        Exported functions

     --dir-import    Import directory
     --dir-export    Export directory
     --dir-resource  Resource directory
     --dir-debug     Debug directory
     --dir-tls       TLS directory

     --strings       Get all strings
     --sections      Sections information
     --dump          Dump all information
remnux@remnux:~$
```

Anomaly Detection -> PEFrame

```
$ peframe malware.exe

Short information
-------------------------------------------------------------
File Name          malware.exe
File Size          935281 byte
Compile Time       2012-01-29 22:32:28
DLL                False
Sections           4
Hash MD5           cae18bdb8e9ef082816615e033d2d85b
Hash SAH1          546060ad10a766e0ecce1feb613766a340e875c0
Imphash            353cf96592db561b5ab4e408464ac6ae
Detected           Xor, Sign, Packer, Anti Debug, Anti VM
Directory          Import, Resource, Debug, Relocation, Security

XOR discovered
-------------------------------------------------------------
Key length         Offset (hex)       Offset (dec)
1                  0x5df4e            384846
2                  0x5df4e            384846
4                  0x5df4e            384846
8                  0x5df4e            384846

Digital Signature
-------------------------------------------------------------
Virtual Address    12A200
Block Size         4813 byte
Hash MD5           63b8c4daec26c6c074ca5977f067c21e
Hash SHA-1         53731a283d0c251f7c06f6d7d423124689873c62
```

```
Packer matched [4]
-------------------------------------------------
Packer             Microsoft Visual C++ v6.0
Packer             Microsoft Visual C++ 5.0
Packer             Microsoft Visual C++
Packer             Installer VISE Custom

Anti Debug discovered [9]
-------------------------------------------------
Anti Debug         FindWindowExW
Anti Debug         FindWindowW
Anti Debug         GetWindowThreadProcessId
Anti Debug         IsDebuggerPresent
Anti Debug         OutputDebugStringW
Anti Debug         Process32FirstW
Anti Debug         Process32NextW
Anti Debug         TerminateProcess
Anti Debug         UnhandledExceptionFilter

Anti VM Trick discovered [2]
-------------------------------------------------
Trick              Virtual Box
Trick              VMware trick

Suspicious API discovered [35]
-------------------------------------------------
Function           CreateDirectoryA
Function           CreateFileA
Function           CreateFileMappingA
Function           CreateToolhelp32Snapshot
```

```
Suspicious Sections discovered [2]
-------------------------------------------------------------
Section            .data
Hash MD5           b896a2c4b2be73b89e96823c1ed68f9c
Hash SHA-1         523d58892f0375c77e5e1b6f462005ae06cdd0d8
Section            .rdata
Hash MD5           41795b402636cb13e2dbbbec031dbb1a
Hash SHA-1         b674141b34f843d54865a399edfca44c3757df59

File name discovered [43]
-------------------------------------------------------------
Binary             wiseftpsrvs.bin
Data               ESTdb2.dat
Data               Favorites.dat
Data               History.dat
Data               bookmark.dat
Data               fireFTPsites.dat
Data               quick.dat
Data               site.dat
Data               sites.dat
Database           FTPList.db
Database           sites.db
Database           NovaFTP.db
Executable         unleap.exe
Text               signons2.txt
Text               signons3.txt

Url discovered [2]
-------------------------------------------------------------
Url                RhinoSoft.com
Url                http://0uk.net/zaaqw/gate.php

Meta data found [4]
-------------------------------------------------------------
CompiledScript     AutoIt v3 Script
FileVersion        3, 3, 8, 1
FileDescription
Translation        0x0809 0x04b0
```

Anomaly Detection -> Pev tool

- Pev is a multiplatform toolkit to work with PE binaries.
- Its main goal is to provide feature-rich tools for proper analyze binaries, specially the suspicious ones.

Features:

- Based on own PE library, called libpe
- Support for PE32 and PE32+ (64-bit) files
- Formatted output in text and CSV (other formats in development)

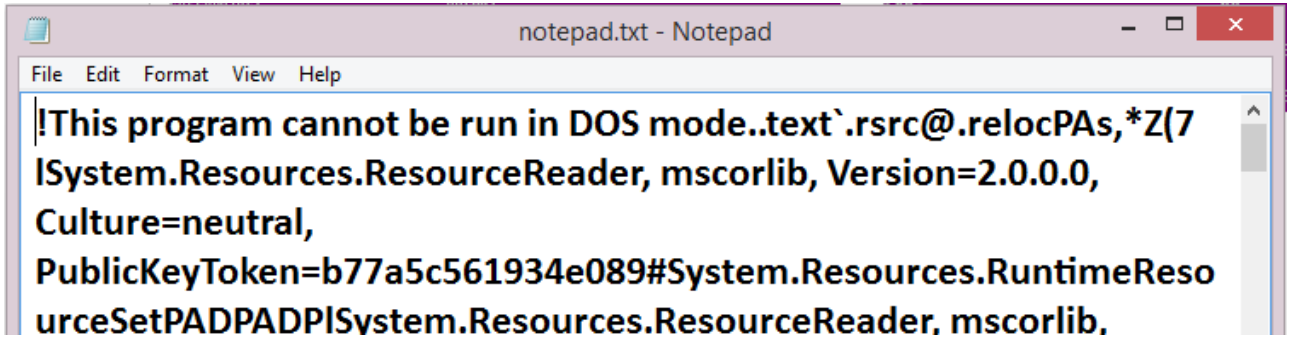
Tools:

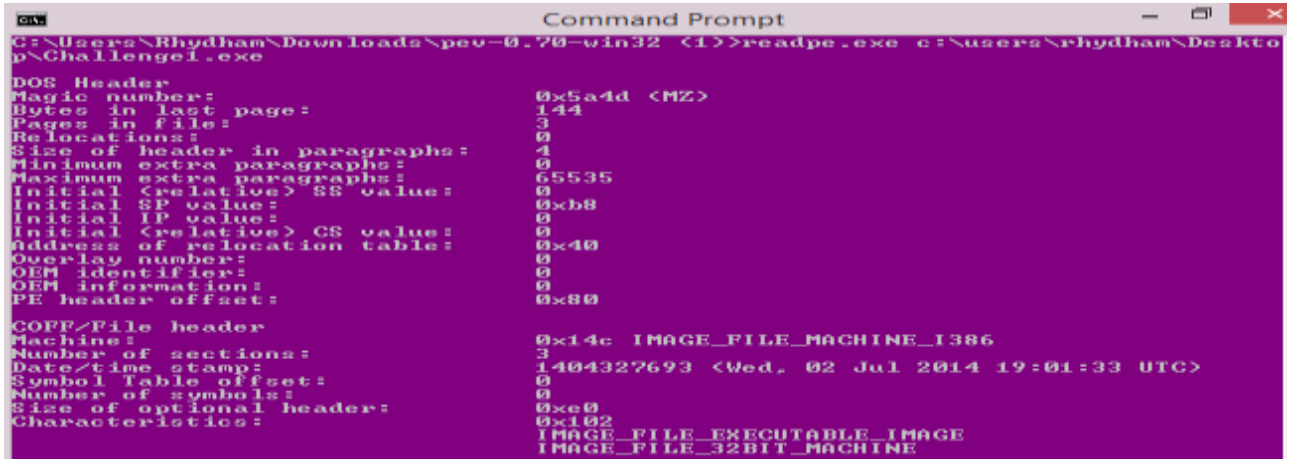
- **pesec**: check security features in PE files, extract certificates and more
- **readpe**: parse PE headers, sections, imports and exports
- **pescan**: detect TLS callback functions, DOS stub modification, suspicious sections and more
- **pedis**: disassembly a PE file section or function with support for Intel and AT&T syntax
- **pehash**: calculate PE file hashes
- **pepack**: detect if an executable is packed or not
- **pestr**: search for hardcoded Unicode and ASCII strings simultaneously in PE files
- **peres**: show and extract PE file resources

Anomaly Detection -> Pev tool

```
C:\Users\Rhydham\Downloads\pev-0.70-win32 (1)>pescan c:\users\rhydham\Desktop\Ch
allenge1.exe
file entropy:                   7.877885 (probably packed)
fpu anti-disassembly:           no
imagebase:                      normal
entrypoint:                     normal
DOS stub:                       normal
TLS directory:                  not found
section count:                  3
.text:                          normal
.rsrc:                          normal
.reloc:                         small length
timestamp:                      normal

C:\Users\Rhydham\Downloads\pev-0.70-win32 (1)>pepack c:\users\rhydham\Desktop\Ch
allenge1.exe
warning: without valid database file, pepack will search in generic mode only
packer:                         no packer found

C:\Users\Rhydham\Downloads\pev-0.70-win32 (1)>pehash c:\users\rhydham\Desktop\Ch
allenge1.exe
file:                           c:\users\rhydham\Desktop\Challenge1.exe
md5:                            66692c39aab3f8e7979b43f2a31c104f
sha1:                           5f7d1552383dc9de18758aa29c6b7e21ca172634
ssdeep:                         3072:vaL7nzo5UC2ShGACS3XzX1/ZPYHLy7argeZX:uUUC2
SHjpurG

header:                         IMAGE_DOS_HEADER
md5:                            919f1c12cf0d1cd93d7f1077f13ac374
sha1:                           3d43712e5606b4640b85f5f0e25e9db8ed552074
ssdeep:                         3:WlWUqt/vllnln:idqP

header:                         IMAGE_COFF_HEADER
md5:                            a29a1e3b447778ca29c473336432caef
sha1:                           1f5a860410687731c425582644fb13cad03e70d9
ssdeep:                         3:HKkn:H/n

header:                         IMAGE_OPTIONAL_HEADER
md5:                            705cb2c333df355dc0e6a7fd079d3fb7
sha1:                           5c6267c71169a39747ecb636bbe932a639c6e4a8
ssdeep:                         3:6jEl/h/lFlUB//llrlltldglPtl19tllH:zv/IlN

section:                        .text
md5:                            e4c64d5b55603ecef3562099317cad76
sha1:                           a2d3a5aee6372f578ff19e0b19fe5bf4e6d5394b
ssdeep:                         3072:MaL7nzo5UC2ShGACS3XzX1/ZPYHLy7ar:nUUC2SHjp
urG

section:                        .rsrc
md5:                            6adbd3818087d9be58766dccc3f5f2bd
sha1:                           6362762d14fcd2b18dcedaa9cc873bfe3045cea9
ssdeep:                         24:nTDRw8ZWq8YZhNwxEZp35frPN8qgdt4+lEbNFjMyi0:/
5ZWq8Yl3ZZ5fBFWSfbNtm

section:                        .reloc
md5:                            34db3eafce34815286f13c2ea0e2da70
sha1:                           968b406252dd04ee359f4d45fdcc5746498126fe
ssdeep:                         3:kLlg:kLO

C:\Users\Rhydham\Downloads\pev-0.70-win32 (1)>
```

```
C:\Users\Rhydham\Downloads\pev-0.70-win32 (1)>pesec.exe c:\users\rhydham\Desktop
\Challenge1.exe
ASLR:                                   yes
DEP/NX:                                 yes
SEH:                                    no
Stack cookies (EXPERIMENTAL):           yes

C:\Users\Rhydham\Downloads\pev-0.70-win32 (1)>pestr.exe c:\users\rhydham\Desktop
\Challenge1.exe > notepad.txt

C:\Users\Rhydham\Downloads\pev-0.70-win32 (1)>notepad.txt
```





Investigation -> Pyew

Pyew is very useful tool to investigate the file or executables for malwares.
It helps in finding : Imported files, exported files, URLS, Call graph, Anti-vm tricks, Shell Code presence, Packers, MD5 etc.
It also searches the file at threat-expert and virus total.

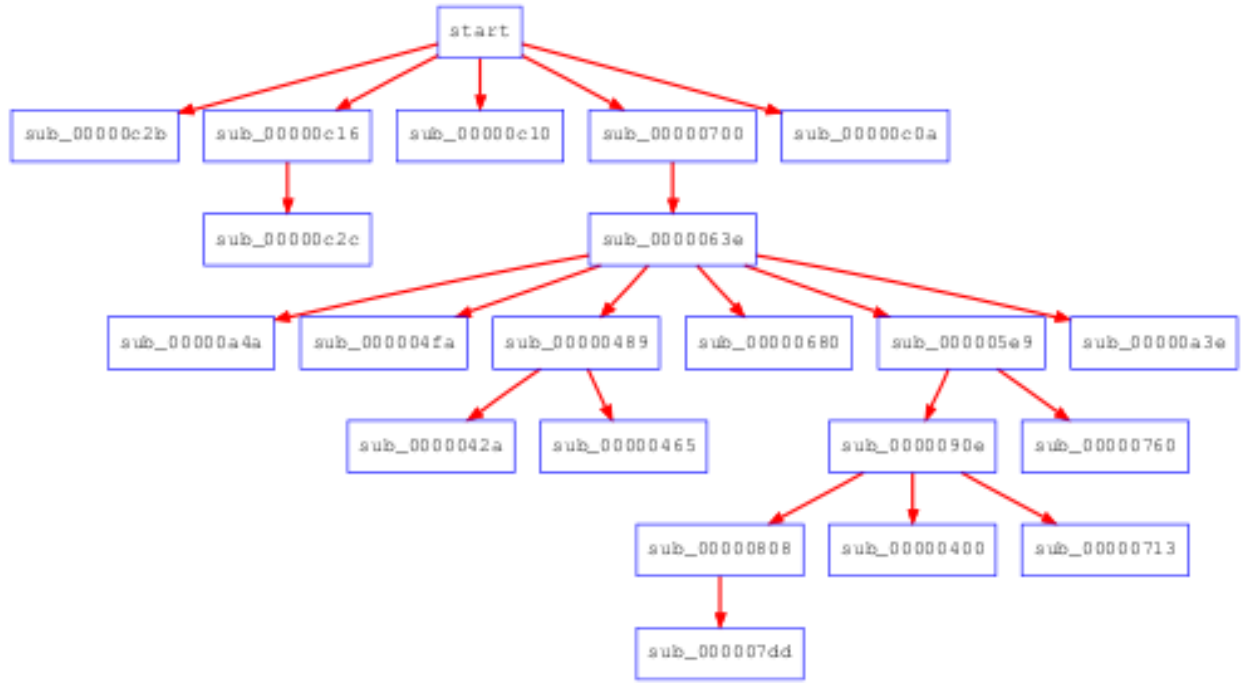
```
remnux@remnux: ~
File  Edit  Tabs  Help
premnux@remnux:~$ pyew /home/remnux/Desktop/kiwi.exe
PE Information

Sections:
   .text 0x1000 0x834 2560
   .rdata 0x2000 0x344 1024
   .data 0x3000 0x608 512
   .rsrc 0x4000 0x8cb8 36352

Entry Point at 0xaa6
Virtual Address is 0x4016a6
Code Analysis ...
Searching typical function's prologs...
Found 0 possible function(s) using method #1
Found 15 possible function(s) using method #2


Searching function's starting at the end of known functions...


0000   4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00    MZ..............
0010   B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00    ........@.......
0020   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
0030   00 00 00 00 00 00 00 00 00 00 00 00 E8 00 00 00    ................
0040   0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68    ........!..L.!Th
0050   69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F    is program canno
0060   74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20    t be run in DOS
0070   6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00    mode....$.......
0080   53 73 D0 0D 17 12 BE 5E 17 12 BE 5E 17 12 BE 5E    Ss.....^...^...^
0090   94 0E B0 5E 14 12 BE 5E 78 0D B4 5E 1C 12 BE 5E    ...^...^x..^...^
00A0   78 0D B5 5E 16 12 BE 5E 78 0D BA 5E 13 12 BE 5E    x..^...^x..^...^
```

```
Pyew Plugins:

fgraph    Show the flowgraph of the specified function or the current on
cgraph    Show the callgraph of the whole program or the specified function
binvi     Show an image representing the current opened file
pdfilter  Get information about the streams
pdfstream  Show streams list
vt        Search the sample in Virus Total
chkurl    Check URLs of the current file
pdf       Get the information about the PDF
pdfvi     Show decoded streams
pdfobj    Show object's list
pdfss     Seek to one stream
pdfview   Show decoded streams (in a GUI)
ole2      Get the OLE2 directory
url       Search URLs in the current document
chkbad    Check for known bad URLs
packer    Check if the PE file is packed
threat    Search in Threat Expert for the behavior's report
sc        Search for shellcode
antivm    Search for common antivm tricks
pdfso     Seek to one object

Any other expression will be evaled as a Python expression

[0x00000000:0x00400000]>
```

Investigation -> Pyew

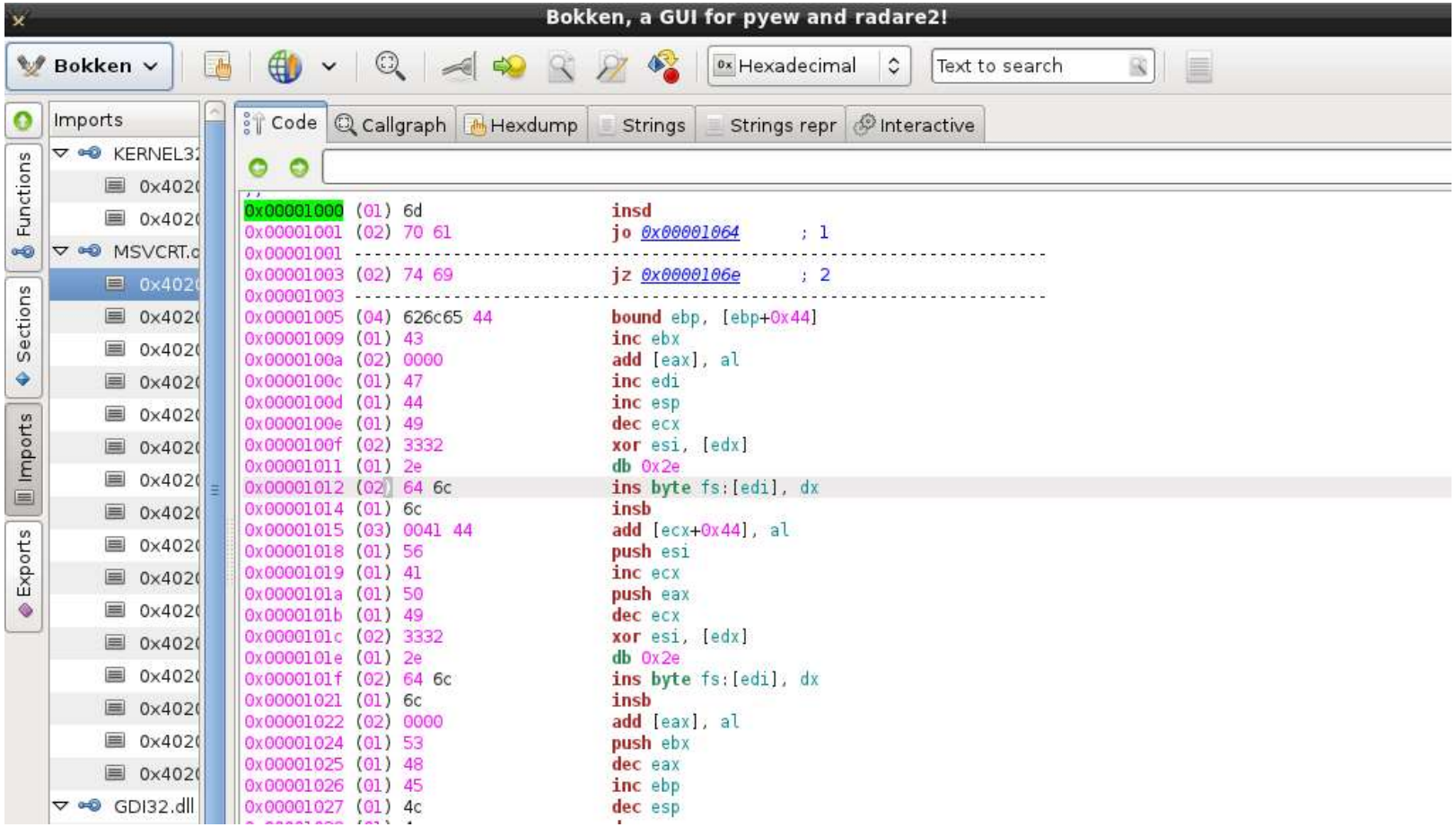
```
File  Edit  Tabs  Help
Any other expression will be evaled as a Python expression

[0x00000000:0x00400000]> imports
KERNEL32.dll
        0x40201c GetModuleHandleA
        0x402020 GetStartupInfoA
USER32.dll
        0x40206c LoadImageA
GDI32.dll
        0x402000 GetObjectA
        0x402004 GetBitmapBits
        0x402008 TextOutA
        0x40200c CreateCompatibleDC
        0x402010 SelectObject
        0x402014 DeleteObject
MSVCRT.dll
        0x402028 _stricmp
        0x40202c _except_handler3
        0x402030 free
        0x402034 _exit
        0x402038 _XcptFilter
        0x40203c exit
        0x402040 _acmdln
        0x402044 __getmainargs
        0x402048 _initterm
        0x40204c __setusermatherr
        0x402050 _adjust_fdiv
        0x402054 __p__commode
        0x402058 __p__fmode
        0x40205c _controlfp
        0x402060 ??2@YAPAXI@Z
        0x402064 __set_app_type
[0x00000000:0x00400000]> url
ASCII URLs

http://crl.thawte.com/ThawtePremiumServerCA.crl0
http://cs-g2-crl.thawte.com/ThawteCSG2.crl0
http://ocsp.thawte.com0
http://www.usertrust.com1
http://www.comodogroup.com/repository0B
http://crl.usertrust.com/UTN-USERFirst-Object.crl0
[0x00000000:0x00400000]>
```

```
[0x00000000:0x00400000]> md5
md5: d7549732c7e9446bdeb7cf93a08b0eeb
[0x00000000:0x00400000]> sc
***No shellcode detected via emulation
[0x00000000:0x00400000]> antivm
[0x00000000:0x00400000]> pdf
PDFiD 0.0.11 /home/remnux/Desktop/kiwi.exe
 Not a PDF document

[0x00000000:0x00400000]> cgraph
[0x00000000:0x00400000]> threat
[0x00000000:0x00400000]> vt
File /home/remnux/Desktop/kiwi.exe with MD5 d7549732c7e9446bdeb7cf93a08b0eeb
------------------------------------------------------------------------------

[0x00000000:0x00400000]>
```



Investigation -> Bokken

Bokken describes all the task done by Pyew and Radare in GUI Interactive mode



Disassembler vs Debugger vs Decompiler

Disassembler:

> A disassembler is a software tool which transforms machine code into a human readable mnemonic representation called assembly language.

Debugger:

> Debuggers allow the user to view the running state of a program.

Decompiler:

> Software used to revert the process of compilation. Decompiler takes a binary program file as input and output the same program expressed in a structured higher-level language.

Reversing tools for Windows (some may work on other platforms):

- Jclasslib (bytecode viewer)
- FrontEnd Plus (java bytecode decompiler)
- Jad (java bytecode decompiler)
- Fernflower (java bytecode decompiler)
- OllyDbg (machine code debugger-disassembler)
- IdaPro (machine code debugger-disassembler)
- PEBrowse (machine code debugger-disassembler)
- Boomerang (machine code decompiler)

References:

- Entropy for files: http://www.forensickb.com/2013/03/file-entropy-explained.html
- UPX: http://upx.sourceforge.net/
- Bitehist: https://www.cert.at/downloads/software/bytehist_en.html
- DensityScout : http://digital-forensics.sans.org/blog/2012/04/26/finding-unknown-malware-with-densityscout
- PEScanner : http://sourceforge.net/projects/pescanner/
- EXE-Scan : http://securityxploded.com/exe-scan.php
- PEFrame : http://www.tekdefense.com/news/2013/3/17/tektip-ep25-static-malware-analysis-with-peframe.html
- PEV : http://pev.sourceforge.net/
- PYEW : https://github.com/joxeankoret/pyew
- Bokken : https://inguma.eu/projects/bokken/wiki/Installation

Thank you.!