Introduction to Static Analysis

Reverse Engineering – Introduction to Static Analysis

Albert López - newlog@overflowedminds.net

Goals

What can you expect after this module?

Goals

1. Understand how to extract meaningful information from binaries

1. Get in touch with different well-known RE tools

Introduction

We are going to give a **first baby steps** towards static analysis.

Static analysis is the craft of extracting information from binaries without executing them.

Basically, we will get familiar with a couple of tools that can be used to extract information from binaries.

Enough writing our own tools, let's not reinvent the wheel.

The Tools

- PEView: PE parser. Allows you traverse PE structures
- CFF Explorer: PE parser. Allows you traverse PE structures with some extra help
- PEStudio: PE parser with other utilities
- Dependency Walker: PE imports parser
- Readelf: ELF parser. Linux native.
- IDA Pro: Reverse engineering framework
- Radare2: Open source reverse engineering framework
- (...)

Identify Binary

TrID

- URL: http://mark0.net/soft-trid-e.html
- Description: TrID is an utility designed to identify file types from their binary signatures.
- Available for Windows and Linux

Usage:

$.\trid <filename>

C:\Windows\System32\cmd.exe

```
C:\Users\IEUser\Desktop\tools>trid.exe PEview.exe

TrID/32 - File Identifier v2.24 - (C) 2003-16 By M.Pontello
Definitions found:  9541
Analyzing...

Collecting data from file: PEview.exe
 52.9% (.EXE) Win32 Executable (generic) (4508/7/1)
 23.5% (.EXE) Generic Win/DOS Executable (2002/3)
 23.5% (.EXE) DOS Executable Generic (2000/1)
```

file

- Preinstalled in Unix distros
- Description: A tool to “determine file type”.

Usage:

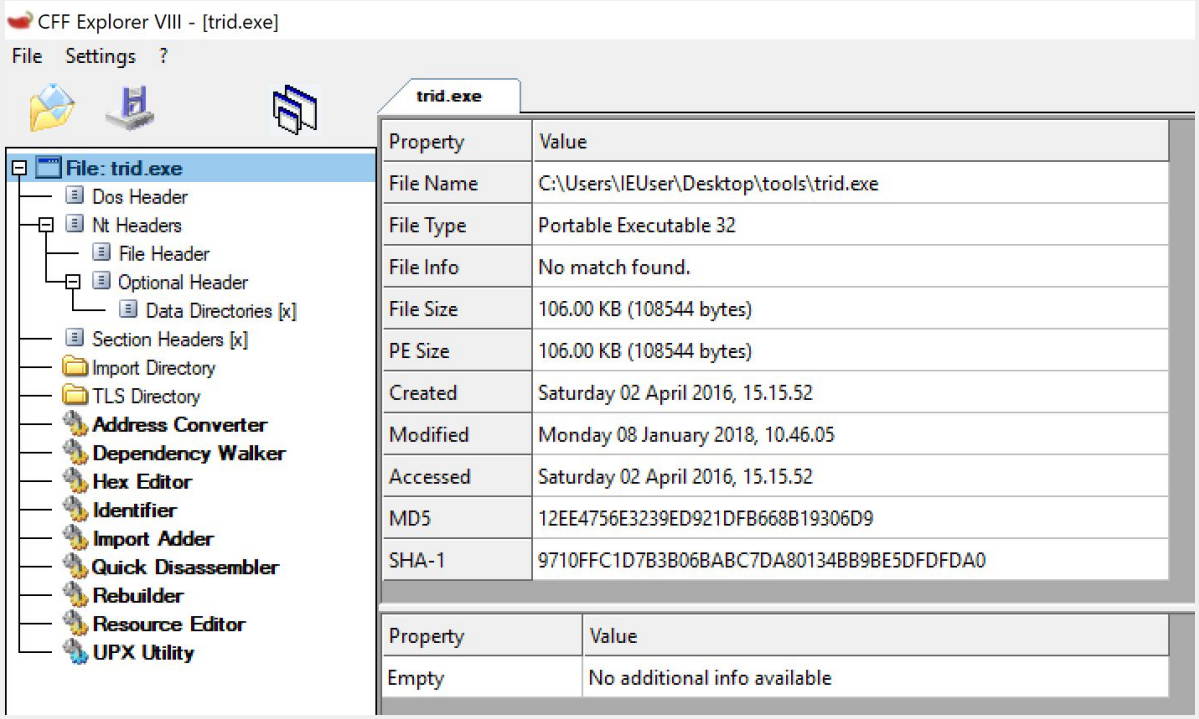
$ file <filename>

```
→ ~ file /bin/bash
/bin/bash: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked, i
nterpreter /lib64/ld-linux-x86-64.so.2, for GNU/Linux 2.6.32, BuildID[sha1]=17150535c
59ef39d9b6db94479a51fcd69942a0c, stripped
```

CFF Explorer

- URL: http://www.ntcore.com/exsuite.php
- Description: Designed to make PE editing as easy as possible.
- Available for Windows.

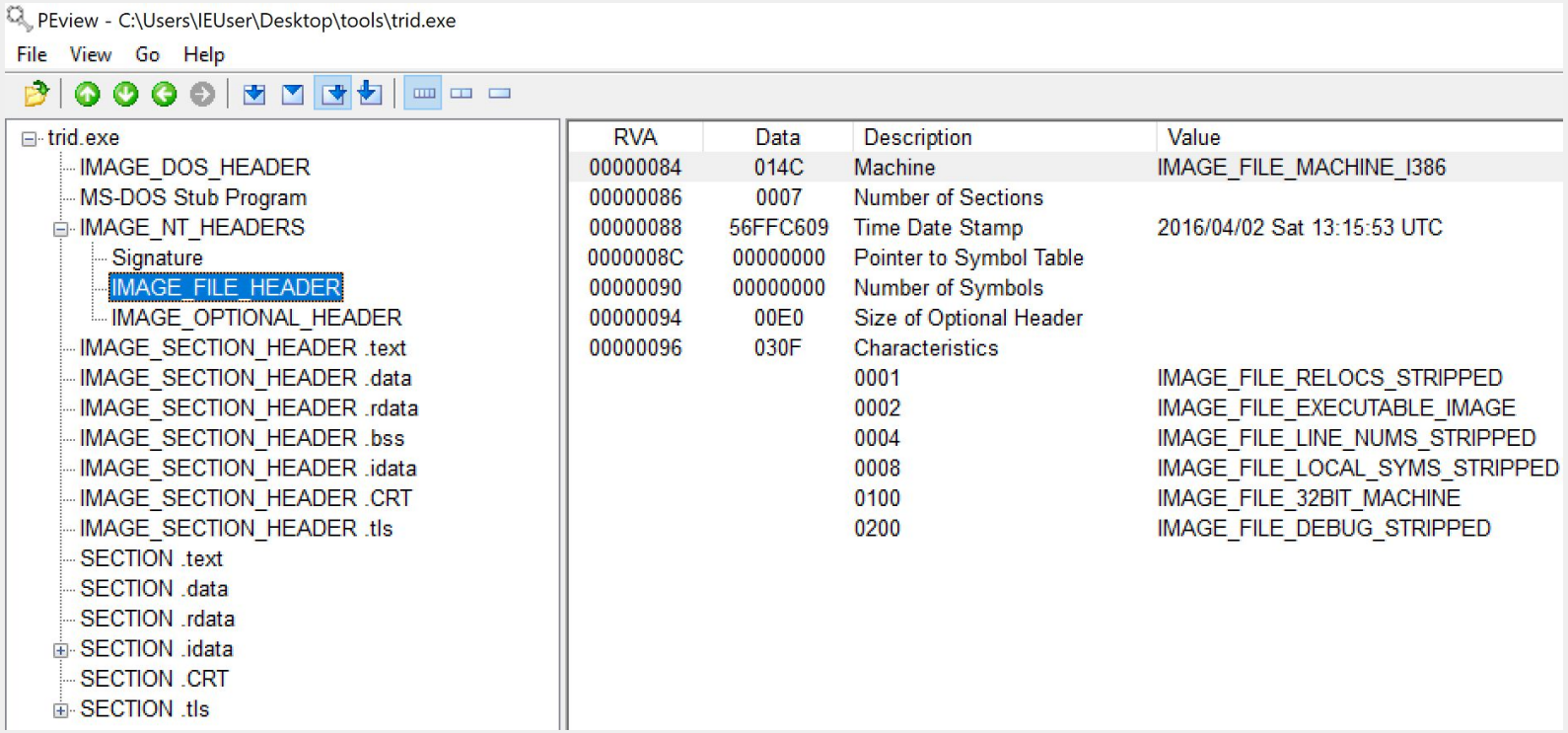
Usage:



PEView

- URL: http://wjradburn.com/software/
- Description: PEview provides a quick and easy way to view the structure and content of 32-bit PE and COFF files
- Available for Windows.

Usage:



Radare2

- URL: https://github.com/radare/radare2
- Description: Radare2 is a complete framework for reverse-engineering and analyzing binaries
- Available both for Windows and Linux (and others)

```
➜  ~ rabin2 -h
Usage: rabin2 [-AcdeEghHiIjlLMqrRsSUvVxzZ] [-@ at] [-a arch] [-b bits] [-B addr]
              [-C F:C:D] [-f str] [-m addr] [-n str] [-N m:M] [-P[-P] pdb]
              [-o str] [-O str] [-k query] [-D lang symname] | file
 -@ [addr]       show section, symbol or import at addr
 -A              list sub-binaries and their arch-bits pairs
 -a [arch]       set arch (x86, arm, .. or <arch,cpu,bits>)
 -b [bits]       set bits (32, 64 ...)
 -B [addr]       override base address (pie
 -c              list classes
 -cc             list classes in header form
 -C [fmt:C:D]    create [elf,mach0,pe] with
 -d              show debug/dwarf informatio
 -D lang name    demangle symbol name (-D al
 -e              entrypoint
```

```
➜  ~ r2 /bin/ls
 -- Add comments using the ';' key in visual mode or the 'CC' command from the radare2
[0x00005600]> i?
|Usage: i Get info from opened file (see rabin2's manpage)
| Output mode:
| '*'                 Output in radare commands
| 'j'                 Output in json
| 'q'                 Simple quiet output
| Actions:
| i|ij                Show info of current file (in JSON)
| iA                  List archs
| ia                  Show all info (imports, exports, sections..)
| ib                  Reload the current buffer for setting of the bin (use once only)
| ic                  List classes, methods and fields
| icc                 List classes, methods and fields in Header Format
| iC                  Show signature info (entitlements, ...)
| id[?]               Debug information (source lines)
| idp                 Load pdb file information
| iD lang sym         demangle symbolname for given language
| ie                  Entrypoint
| iE                  Exports (global symbols)
| ih                  Headers (alias for iH)
| iHH                 Verbose Headers in raw text
| ii                  Imports
| iI                  Binary info
```

Radare2

$ rabin2 -I <binary> or $ r2 /bin/ls
> i

```
➜  ~ rabin2 -I /bin/ls
arch     x86
binsz    127901
bintype  elf
bits     64
canary   true
class    ELF64
crypto   false
endian   little
havecode true
intrp    /lib64/ld-linux-x86-64.so.2
lang     c
linenum  false
lsyms    false
machine  AMD x86-64 architecture
maxopsz  16
minopsz  1
nx       true
os       linux
pcalign  0
pic      true
relocs   false
relro    full
rpath    NONE
static   false
stripped true
subsys   linux
va       true
```

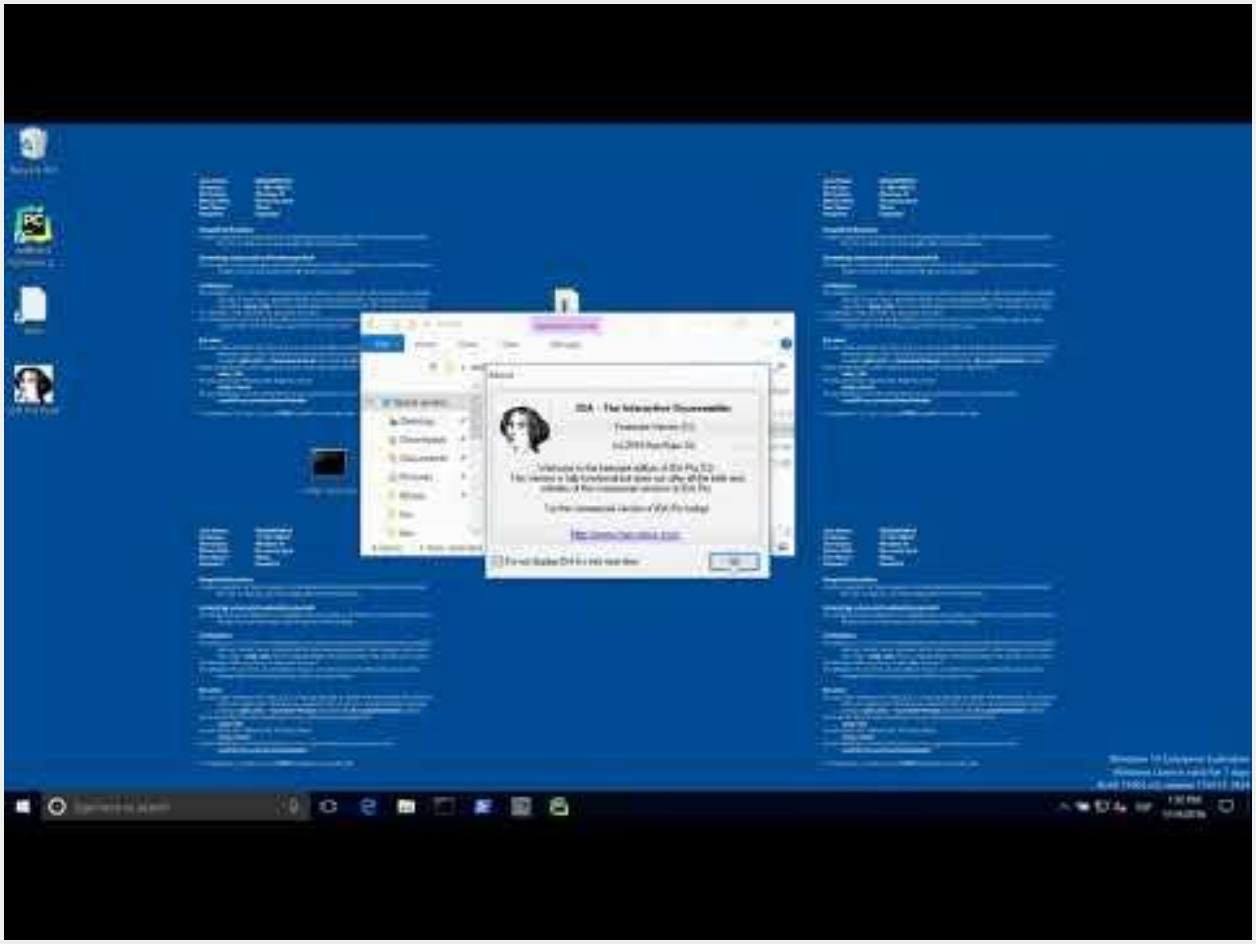
```
➜  ~ r2 /bin/ls
 -- Use 'e' and 't' in Visual mode to e
[0x00005600]> i
blksz    0x0
block    0x100
fd       3
file     /bin/ls
format   elf64
iorw     false
mode     -r-x
size     0x1faa0
humansz  126.7K
type     DYN (Shared object file)
arch     x86
binsz    127901
bintype  elf
bits     64
canary   true
class    ELF64
crypto   false
endian   little
havecode true
intrp    /lib64/ld-linux-x86-64.so.2
lang     c
linenum  false
lsyms    false
machine  AMD x86-64 architecture
maxopsz  16
minopsz  1
nx       true
os       linux
pcalign  0
pic      true
relocs   false
relro    full
rpath    NONE
static   false
stripped true
subsys   linux
va       true

[0x00005600]>
```

Imports

Windows

- PEView, CFF Explorer, IDA Pro



Linux

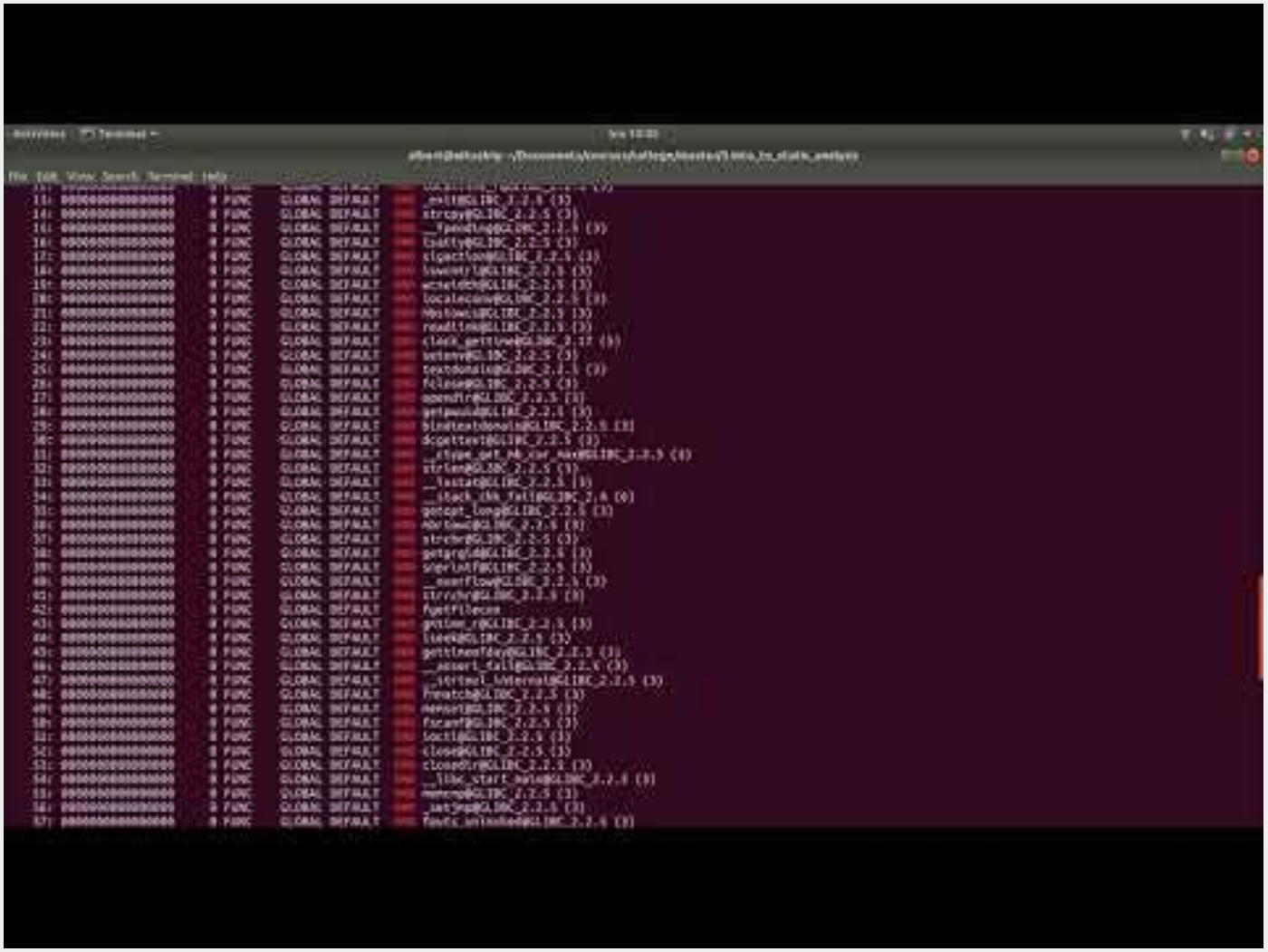
- Getting imported libraries

  - objdump -p <bin> | grep NEEDED
  - readelf -d <bin> | grep Shared
  - rabin2 -l <bin>

- Getting imported symbols

  - objdump -T <bin> | grep UND
  - readelf -s <bin> | grep UND
  - rabin2 -i <bin>

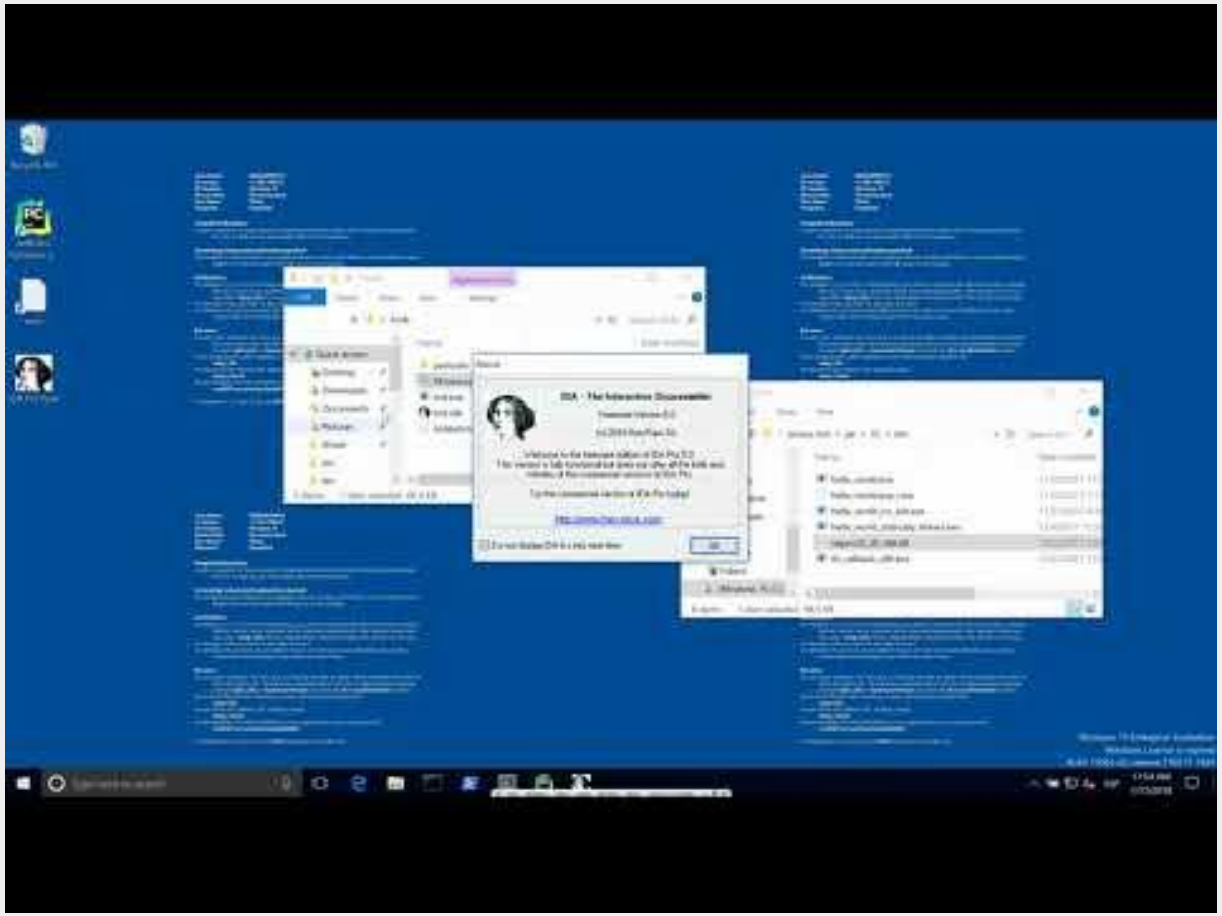
Linux



Exports

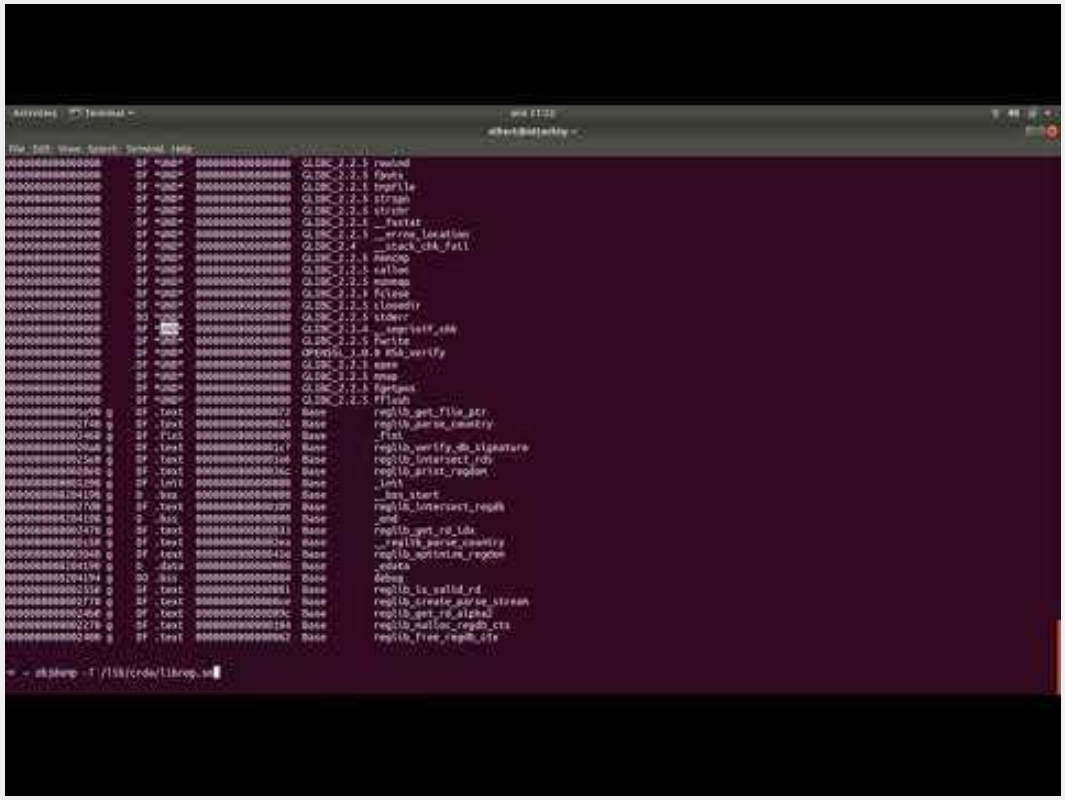
Windows

- PEView, CFF Explorer, IDA Pro



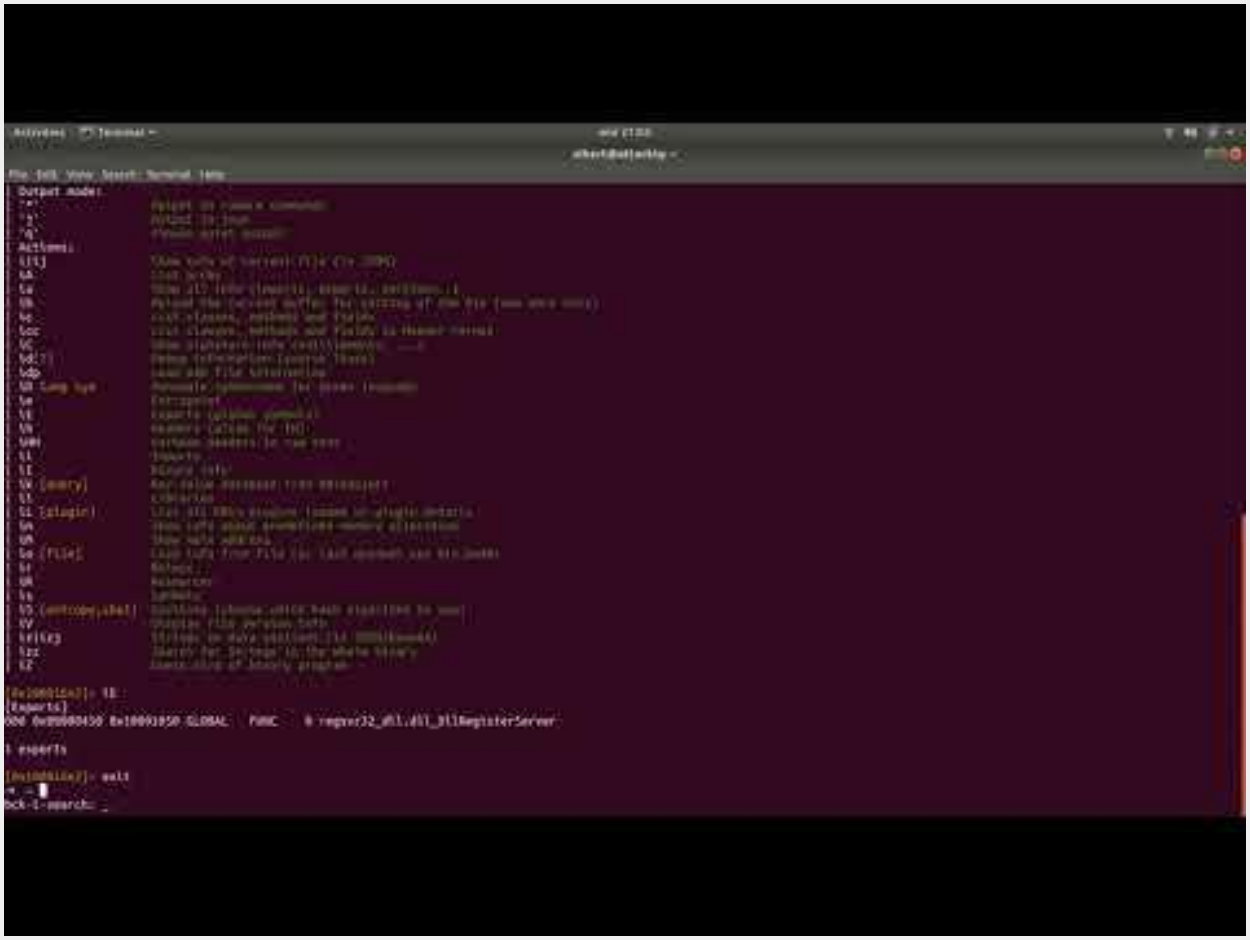
Linux

- objdump -T <bin> | grep -v UND | grep FD
- readelf -s <bin> | grep -v UND | grep F
- r2 <bin>; iE~FUNC



Windows and Linux

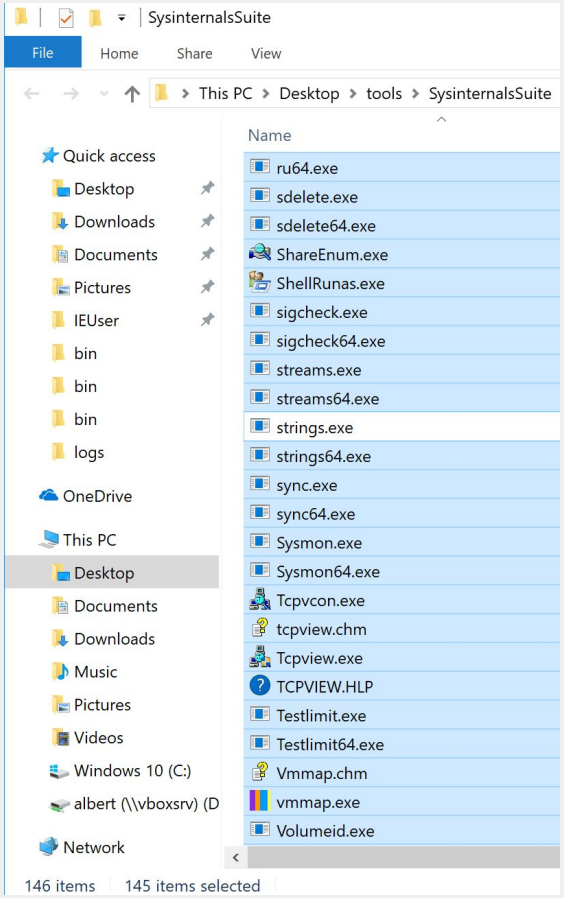
- Getting exported symbols with Radare2



Strings

Sysinternals Suite

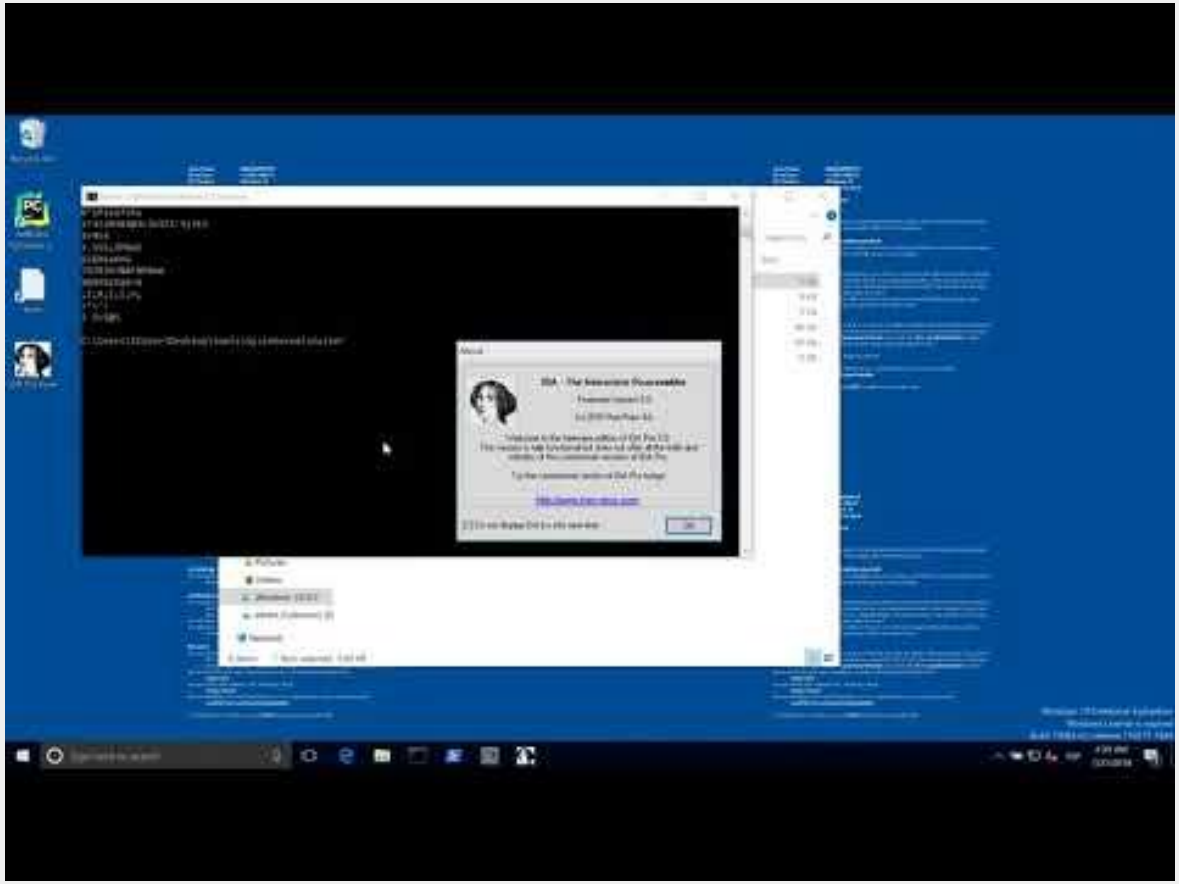
- URL: https://docs.microsoft.com/en-us/sysinternals/downloads/sysinternals-suite
- Description: A bunch of tools to analyze a bunch of things!
- Available for Windows.



- Strings
- Process Monitor
- PSExec
- Process Explorer
- And another 141 tools...

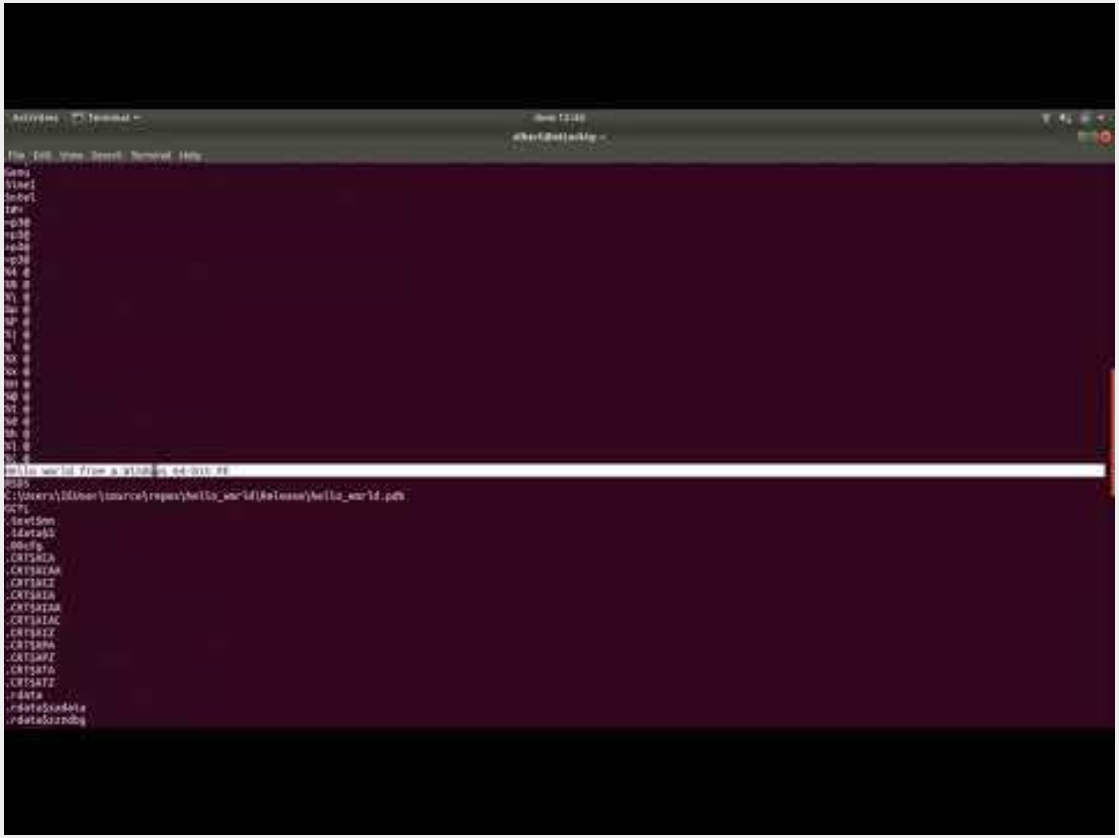
Windows

- strings.exe, IDA Pro



Linux

- strings <bin>
- r2 <bin>; iz



Others

File Entropy

What is entropy?

Formal definition:

> "Information entropy is defined as the average amount of information produced by a stochastic source of data."

Definition that makes sense in our case:

> "~~In thermodynamics,~~ entropy is commonly associated with the amount of order, disorder, or chaos ~~in a thermodynamic system.~~"

The entropy value for a chunk of data can be used in order to infer if:

- The data is encrypted
- The data is compressed
- The data follows predictable patterns

However, entropy is just a tool, it won't tell you anything for sure.

- Entropy using radare2

With radare2 you can obtain the entropy of many things:

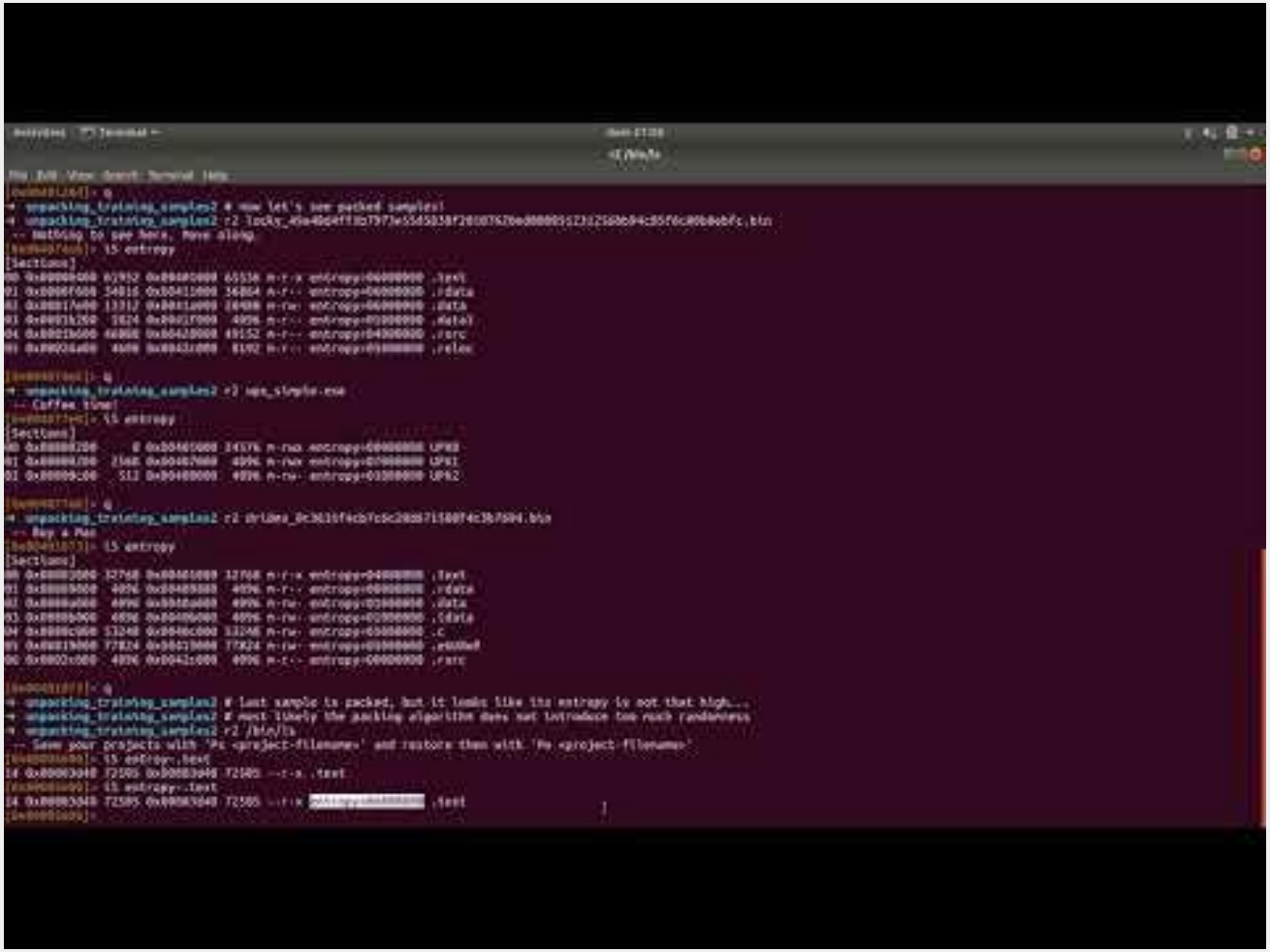
- Binary sections (r2 <bin>; > iS entropy)
- Whole files (rahash2 -a entropy <bin>)
- Chunks of data (r2 <bin>; > p=e)
- Entropy in the form of code jump (r2 <bin>; >p=j)
- (...)

Checking binary section entropy can be useful to understand if the binary is encrypted or compressed.

The .text section can be checked, but the .data and .rdata sections entropy will be quite significative!

Checking the jumps in .text might help to understand if a binary is obfuscated.

- Entropy using radare2



- Entropy using radare2

With radare2, it looks like:

- 04(...) values are low entropies (text files, ascii data, natural language).
- 07(...) values are high entropies (UPX packed sections).

These values can be used as a comparative reference for further analysis.

Having a 07(...) entropy does not necessarily mean that the data is encrypted/random. We have seen that a perfectly normal media (video) file has that entropy.

Packers

Yara

- URL: https://github.com/VirusTotal/yara
- Description: "YARA is a tool aimed at (but not limited to) helping malware researchers to identify and classify malware samples."
- Available for Linux, Windows, macOS

"With YARA you can create descriptions of malware families (or whatever you want to describe) based on textual or binary patterns."

And example of a rule:

```
import pe

rule EXE_Stealth_25: PEiD
{
   strings:
      $a = { 60 90 EB 22 45 78 65 53 74 65 61 6C 74 68 20 2D 20 77 77 77 2E 77 65 62 74
6F 6F 6C 6D 61 73 74 65 72 (...) 09 00 00 8D BD 88 1E 40 00 8B F7 AC }
   condition:
      $a at pe.entry_point

}
```

Yara

In Ubuntu you can install yara with:

- sudo apt-get install yara

As of January 2018, version installed from repository is 3.6.3, whereas last version is 3.7.2 (only 2 versions old).

For Yara to be “consumable”, you need to get rules to apply to your files.

A great project to get rules and contribute rules is here:

Yara-Rules:

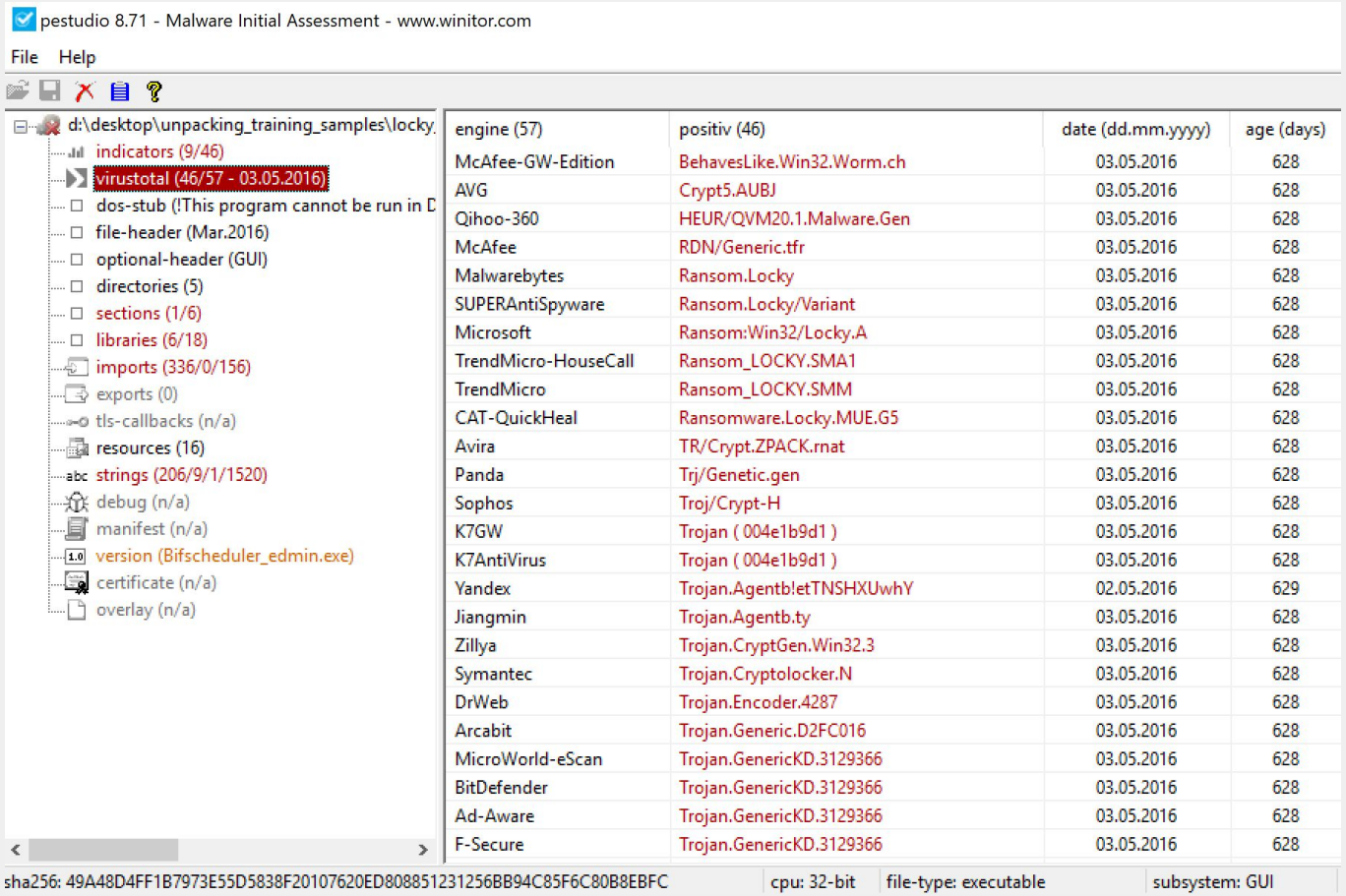
- https://github.com/Yara-Rules/rules/tree/master/Packers

Yara

Malicious Indicators

PEStudio

- URL: https://www.winitor.com/binaries.html
- Description: Spot these artifacts so as to ease the malware initial assessment.
- Available for Windows



engine (57)	positiv (46)	date (dd.mm.yyyy)	age (days)
McAfee-GW-Edition	BehavesLike.Win32.Worm.ch	03.05.2016	628
AVG	Crypt5.AUBJ	03.05.2016	628
Qihoo-360	HEUR/QVM20.1.Malware.Gen	03.05.2016	628
McAfee	RDN/Generic.tfr	03.05.2016	628
Malwarebytes	Ransom.Locky	03.05.2016	628
SUPERAntiSpyware	Ransom.Locky/Variant	03.05.2016	628
Microsoft	Ransom:Win32/Locky.A	03.05.2016	628
TrendMicro-HouseCall	Ransom_LOCKY.SMA1	03.05.2016	628
TrendMicro	Ransom_LOCKY.SMM	03.05.2016	628
CAT-QuickHeal	Ransomware.Locky.MUE.G5	03.05.2016	628
Avira	TR/Crypt.ZPACK.rnat	03.05.2016	628
Panda	Trj/Genetic.gen	03.05.2016	628
Sophos	Troj/Crypt-H	03.05.2016	628
K7GW	Trojan (004e1b9d1)	03.05.2016	628
K7AntiVirus	Trojan (004e1b9d1)	03.05.2016	628
Yandex	Trojan.Agentb!etTNSHXUwhY	02.05.2016	629
Jiangmin	Trojan.Agentb.ty	03.05.2016	628
Zillya	Trojan.CryptGen.Win32.3	03.05.2016	628
Symantec	Trojan.Cryptolocker.N	03.05.2016	628
DrWeb	Trojan.Encoder.4287	03.05.2016	628
Arcabit	Trojan.Generic.D2FC016	03.05.2016	628
MicroWorld-eScan	Trojan.GenericKD.3129366	03.05.2016	628
BitDefender	Trojan.GenericKD.3129366	03.05.2016	628
Ad-Aware	Trojan.GenericKD.3129366	03.05.2016	628
F-Secure	Trojan.GenericKD.3129366	03.05.2016	628

PEStudio

pestudio 8.71 - Malware Initial Assessment - www.winitor.com

File Help

- d:\desktop\unpacking_training_samples\locky.
  - indicators (9/46)
  - virustotal (46/57 - 03.05.2016)
  - dos-stub (!This program cannot be run in D
  - file-header (Mar.2016)
  - optional-header (GUI)
  - directories (5)
  - sections (1/6)
  - libraries (6/18)
  - imports (336/0/156)
  - exports (0)
  - tls-callbacks (n/a)
  - resources (16)
  - strings (206/9/1/1520)
  - debug (n/a)
  - manifest (n/a)
  - version (Bifscheduler_edmin.exe)
  - certificate (n/a)
  - overlay (n/a)

indicator (46)	severity
The file modifies the Registry	1
The file enumerates the list of running processes	1
The file enumerates the list of loaded modules	1
The file enumerates the list of registered windows	1
The file references the protection of the Virtual Address space	1
The file is scored (46/57) by virustotal	1
The debug directory is invalid	1
The file references (6) blacklisted library	1
The section (name:.data3) is blacklisted	1
The file references the Remote Desktop Session Host Server	2
The file references the Service Control Manager (SCM)	2
The file references the Desktop window	2
The file references the Internet Protocol Helper API	2
The file spawns another process	2
The file enumerates files	2
The file references the Global Atom Table	2
The file references the Event Log	2
The file installs an Exception Handler	2
The file references (206) blacklisted string(s)	2
The file imports (156) blacklisted function(s)	2
The file opts for Data Execution Prevention (DEP)	3
The file references the Authorization API	5
The file references the Registry API	5
The file references the Memory Management API	5
The file references the Tool Help API	5

sha256: 49A48D4FF1B7973E55D5838F20107620ED808851231256BB94C85F6C80B8EBFC | cpu: 32-bit | file-type: executable

External Services

Check services such as:

- VirusTotal
- Hybrid Analysis

Homework

No Homework today! Yay!

(Just play a little bit with the tools and get familiar with them...)