DEPARTMENT OF INFORMATION TECHNOLOGY

DELHI TECHNOLOGICAL UNIVERSITY

Malware Analysis (IT-321)

Submitted By:

Name: Samarth Mittal Roll No: 2K18/IT/106

Name: Vivek Yadav Roll No: 2K18/IT/135

Analysis of Widows Payload embedded in Software

Abstract- **Malware detection is an essential factor in security of internet-oriented machines. The combinations of different features are used for malware analysis. The different combinations are generated from APIs, Machine Learning, Summary Information, DLLs and generated Registry Keys. These methods have their own strengths and weaknesses. Here we are going to analyze the type of malware, check how the system was infected and what these kinds of malware can do.**

Keywords- **Malwares, Backdoor, Payload, hashing, DLL, Executable, static analysis, Pestudio, Wireshark, Attacker, Pestudio, Strings.**

INTRODUCTION

Malware is a threat to every single computer user, regardless of their level of usage and skill proficiency. According to AV-Test, an independent IT-security institute, every year the number of malwares is increasing at an unprecedented rate despite using malware detection techniques.

Types of malware analysis:

- Static analysis: It is the process of analysing malware without executing it. Main objective is to extract metadata from the malware. Example: strings, file type, PE headers.
- Dynamic analysis: It is the process of analysing malware after executing it. To understand what and how the malware does during execution. It is mostly monitored in the debugger.
- Code analysis: It is the process of analysing and reverse engineering assembly code. This can be done both statically and dynamically.
- Behavior analysis: It is the process of analysing and monitoring malware right after execution of it. Network monitoring to check how and whom this malware sends the data, processes monitoring to check processes running on the system.

The malware can be a script, executable binary or any other piece of code, which has malicious intention. The main aims of malware are to gain access to the system, disrupt system services, denial of service, steal confidential information and destruction of resources. According to Cisco 2017 annual cybersecurity report, 95% of their analyzed malware were of age below 24 hours. This indicates the pace of malware evolution which stands as a challenge for malware research.

The problem of the signature-based detection system motivates the researchers to think about techniques to deal with new and unknown malware. Downloading legitimate software from any website may download malicious software itself. Mostly malwares are found in cracked software and pirated software. MA is the process of analyzing malware and extracting information from it. This information helps us understand the scope/ functionality of malware, how it can infect systems and how to defend against similar attacks in future.

A. *The purpose of static analysis*

In static analysis the executable file is being analyzed on a structure basis without executing it in a controlled environment. So, it could at least analyze an unknown binary whether or not it falls into a suspicious class of software or not because it does not depend on the malware signature, instead the malicious features that the malware has.

LITERATURE REVIEW

[Muhammad Ijaz, Maliha Ismail "Statical and dynamic malware analysis using machine learning"] worked to analyse the accuracy on the static analysis of malware. They proposed that due to the increasing intelligence of malware the static methods are not efficient now. Controlled environment for malware analysis is not very useful due to the tricky nature of malware, the virtualized and debugging modes are quickly detectable by malware. The virtualized environment is not as effective as the real system due to many traces, which are easily detectable. The tricky malware executes APIs to detect virtual environments.

[Om Prakash Samantray, Satya Narayan Tripathi "A study to understand Malware Behaviour through malware analysis"] work to analyse the behavior of malware. They proposed that most of the malware detection techniques use malware signatures for detection. A novel approach is required to represent malware features effectively to detect obfuscated, unknown, and mutated malware. This paper emphasizes malware behavior, characteristics and properties extracted by different analytic techniques and to decide whether to include them to create behavioral based malware signatures.

We got to know about the malware behavior and the detection techniques. Sometimes benign files may also contain similar operations as that of malicious files which allows the anti-malware system to erroneously detect benign files as malware. When a file performs the same task for which it was created, then no suspicion arises. Static analysis, as one of the malware analysis methods, assists in the detection of software maliciousness as well as the classification of malware families. They also emphasize malware behavior, characteristics and properties extracted by different analytic techniques and to decide whether to include them to create behavioral based malware signatures. We have made an attempt to understand the malware behavior using a few openly available tools for malware analysis.

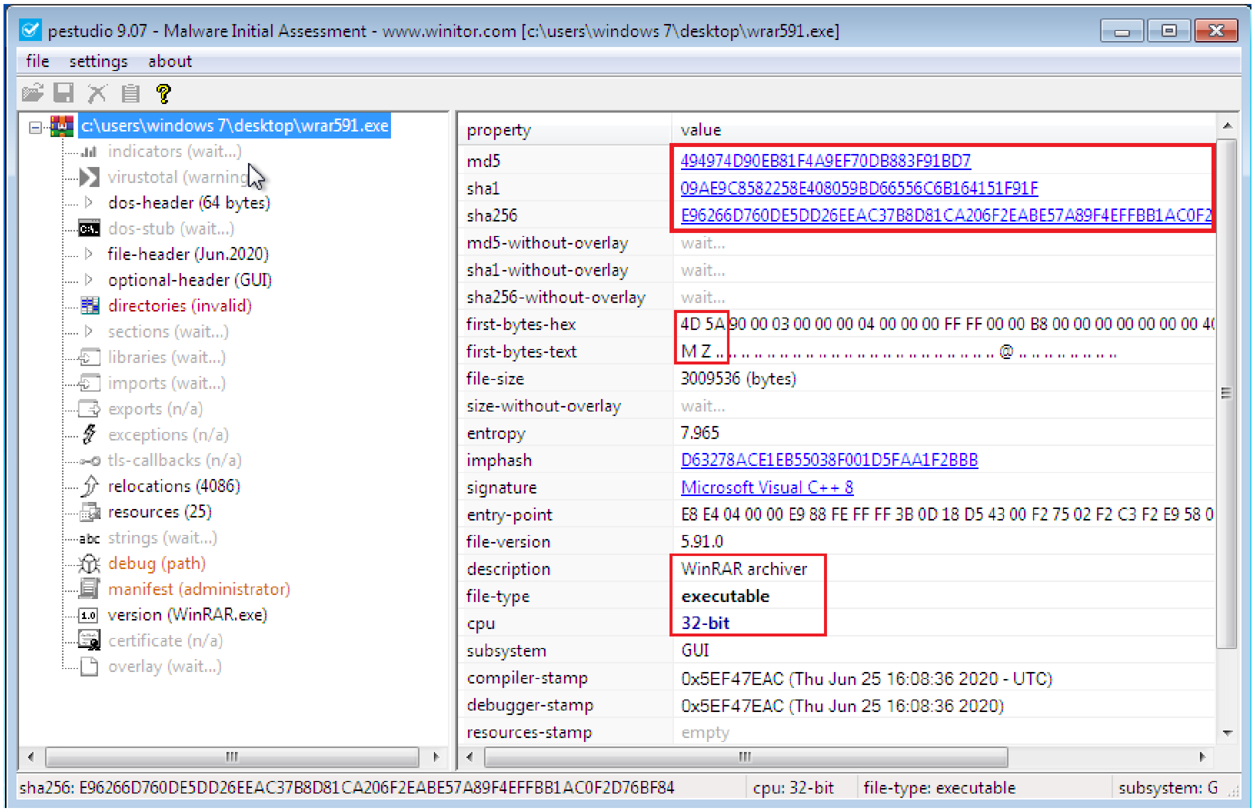
METHODOLOGY

Our analysis works with following component-:

1. **Identify the file type, target os and architecture (32 bit or 64 bit), file format executable or dynamic linked library.**
    - Identifying the file type is extremely important as it helps us identify the target OS and the corresponding architecture.
    - An example of a windows executable is the PE(Portable Executable)
    - A PE could be in the form; .exe, .dll etc.
    - Sometimes attackers use double extensions to avoid malware detection. That's why analysing file signature is important.
    - The file signature exists on the file header.
    - The file signatures for PE files are represented by hexadecimal values of 4D 5A or MZ in the first 2 bytes (0-1).
    - PE programs also have the notice "This program cannot be run in DOS mode"

Note: Attackers use double extensions to make it look like another file. It may look like an image, doc or pdf file. That's why identifying file type is the primary step.

We'll use pestudio to identify the file type:



pestudio 9.07 - Malware Initial Assessment - www.winitor.com [c:\users\windows 7\desktop\wrar591.exe]

property	value
md5	494974D90EB81F4A9EF70DB883F91BD7
sha1	09AE9C8582258E408059BD66556C6B164151F91F
sha256	E96266D760DE5DD26EEAC37B8D81CA206F2EABE57A89F4EFFBB1AC0F2
md5-without-overlay	wait...
sha1-without-overlay	wait...
sha256-without-overlay	wait...
first-bytes-hex	4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00 00 00 00 00 00 40
first-bytes-text	M Z .. .. @ .. ..
file-size	3009536 (bytes)
size-without-overlay	wait...
entropy	7.965
imphash	D63278ACE1EB55038F001D5FAA1F2BBB
signature	Microsoft Visual C++ 8
entry-point	E8 E4 04 00 00 E9 88 FE FF FF 3B 0D 18 D5 43 00 F2 75 02 F2 C3 F2 E9 58 0
file-version	5.91.0
description	WinRAR archiver
file-type	**executable**
cpu	**32-bit**
subsystem	GUI
compiler-stamp	0x5EF47EAC (Thu Jun 25 16:08:36 2020 - UTC)
debugger-stamp	0x5EF47EAC (Thu Jun 25 16:08:36 2020)
resources-stamp	empty

sha256: E96266D760DE5DD26EEAC37B8D81CA206F2EABE57A89F4EFFBB1AC0F2D76BF84 | cpu: 32-bit | file-type: executable | subsystem: G

Information extracted:

Md5: 494974D90EB81F4A9EF70DB883F91BD7

SHA256: 09AE9C8582258E408059BD66556C6B164151F91F

First bytes hex: 4D 5A 90 00 03 …….

First bytes text: M Z …..

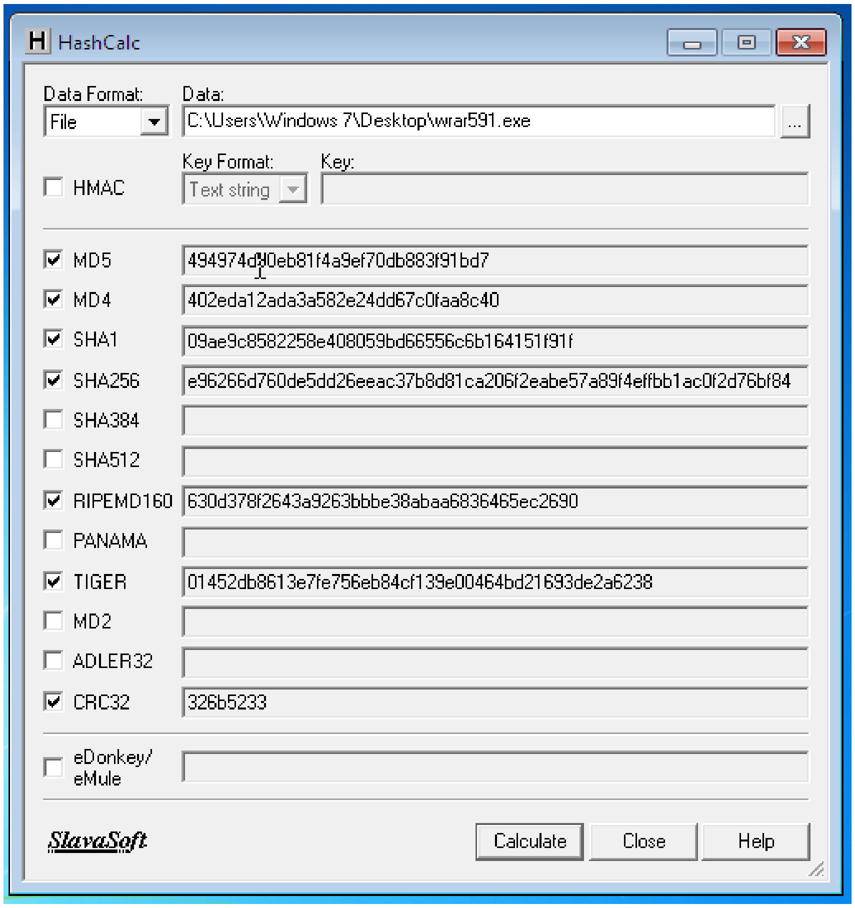
Meaning of this 4D 5A or MZ indicates that it is a Portable Executable (PE) Header.

Description: WinRAR archiver

File type: executable (exe file)

Architecture: 32 bit

2. **Identify the malware by generating a hash of it and check online if someone has already analysed this malware.**
    - Hash is like a fingerprint of a digital file. The **fingerprints** are usually 128-bit or 160-bit numbers that are displayed as a sequence of hexadecimal digits.



HashCalc

Data Format: File

Data: C:\Users\Windows 7\Desktop\wrar591.exe

□ HMAC Key Format: Text string Key:

☑ MD5 494974d90eb81f4a9ef70db883f91bd7

☑ MD4 402eda12ada3a582e24dd67c0faa8c40

☑ SHA1 09ae9c8582258e408059bd66556c6b164151f91f

☑ SHA256 e96266d760de5dd26eeac37b8d81ca206f2eabe57a89f4effbb1ac0f2d76bf84

□ SHA384

□ SHA512

☑ RIPEMD160 630d378f2643a9263bbbe38abaa6836465ec2690

□ PANAMA

☑ TIGER 01452db8613e7fe756eb84cf139e00464bd21693de2a6238

□ MD2

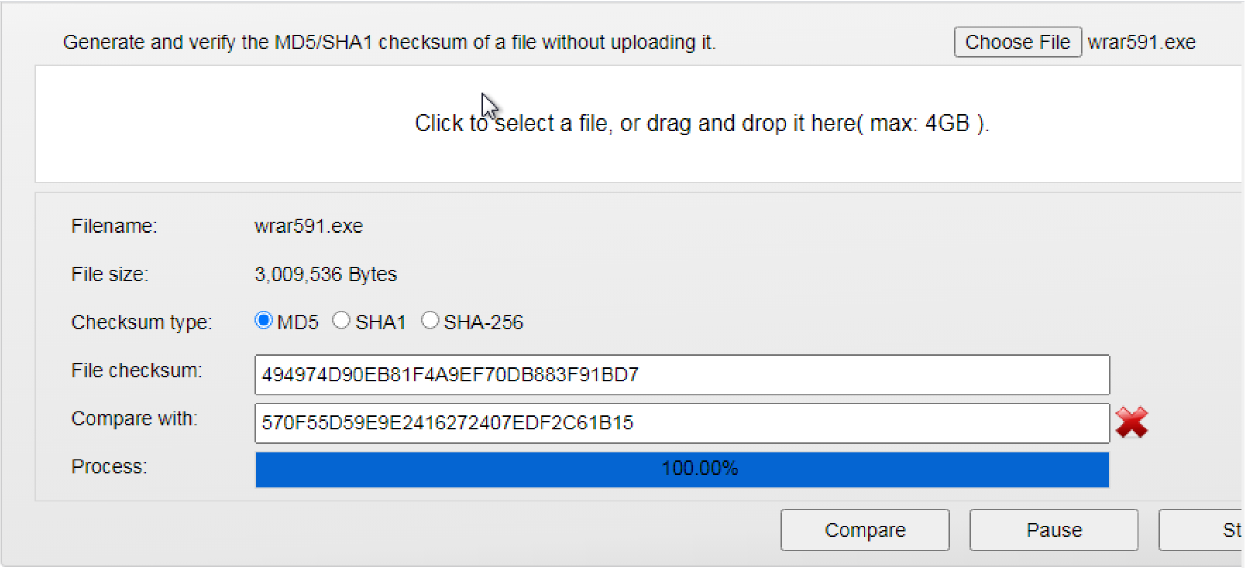
□ ADLER32

☑ CRC32 326b5233

□ eDonkey/ eMule

Generated hash Md5: **494974D90EB81F4A9EF70DB883F91BD7**

Compare generated hash with original winrar software.



Generate and verify the MD5/SHA1 checksum of a file without uploading it. Choose File wrar591.exe

Click to select a file, or drag and drop it here(max: 4GB).

Filename: wrar591.exe

File size: 3,009,536 Bytes

Checksum type: ● MD5 ○ SHA1 ○ SHA-256

File checksum: 494974D90EB81F4A9EF70DB883F91BD7

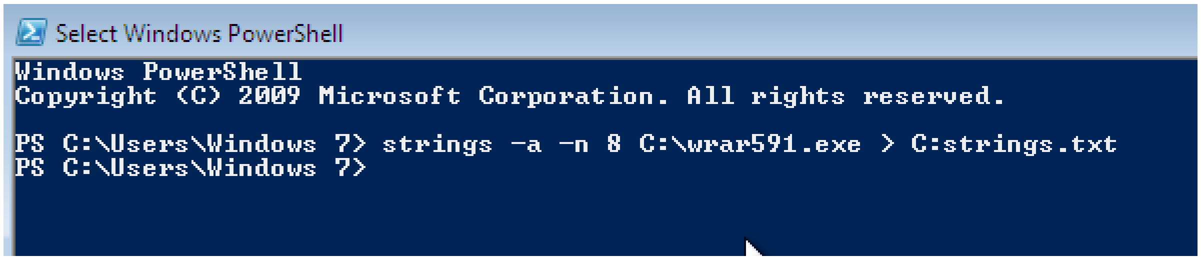
Compare with: 570F55D59E9E2416272407EDF2C61B15 ✗

Process: 100.00%

As the screenshot shows, the generated hash is differ from the hash of the original software.

3. **Strings- Malware creates files, information of that file stored as a string in binary. Extracting strings gives a lot of information about the functionality.**
    - Strings analysis is a process in which we extract readable data/ information from the malware.
    - Strings are in ASCII and Unicode format. Readable data such as file names, registry keys, URLs to which malware is communicating with, ip addresses.
    - For extracting strings, we use strings command line utility and pestudio.
    - Using Windows PowerShell, we can extract strings from the malware.

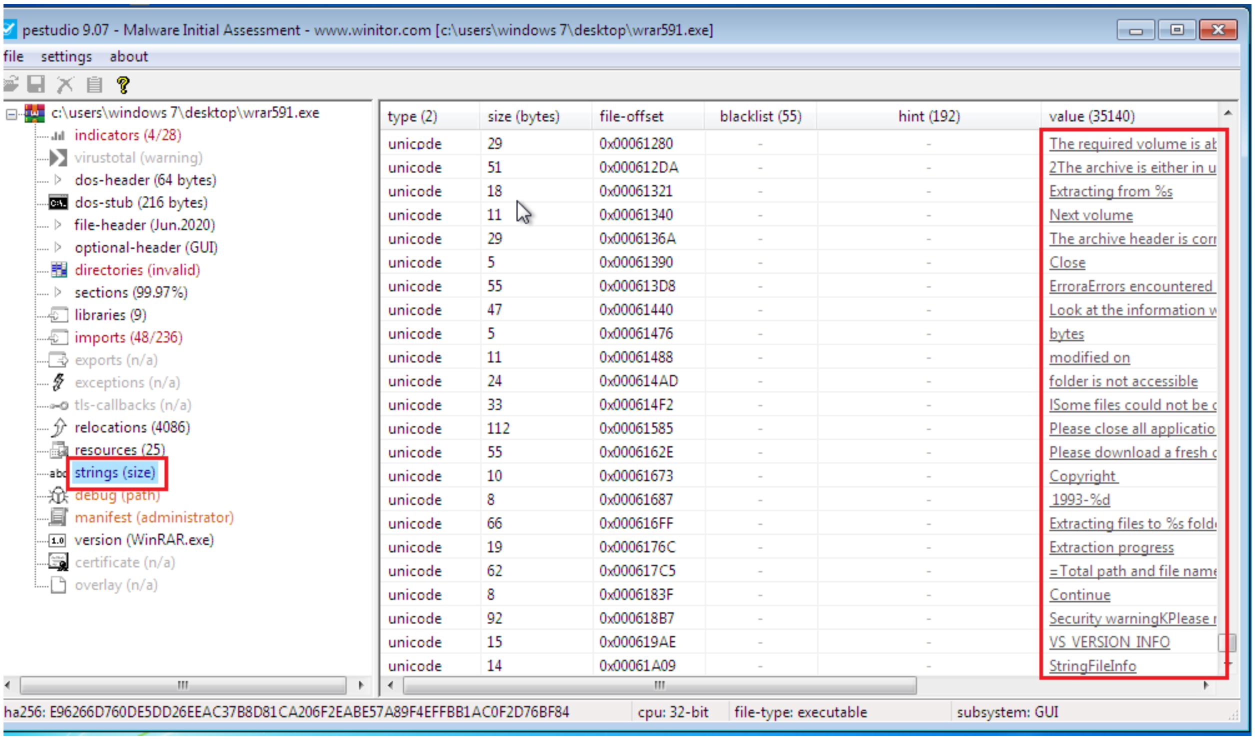
Command:



```
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS C:\Users\Windows 7> strings -a -n 8 C:\wrar591.exe > C:strings.txt
PS C:\Users\Windows 7>
```

By using pestudio



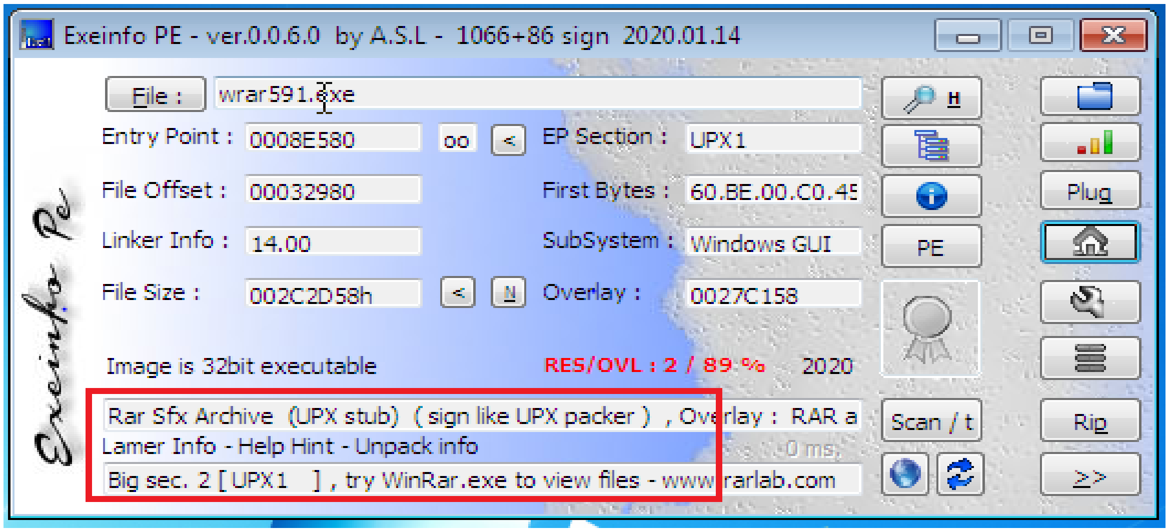
pestudio 9.07 - Malware Initial Assessment - www.winitor.com [c:\users\windows 7\desktop\wrar591.exe]

type (2)	size (bytes)	file-offset	blacklist (55)	hint (192)	value (35140)
unicode	29	0x00061280	-	-	The required volume is ak
unicode	51	0x000612DA	-	-	2The archive is either in u
unicode	18	0x00061321	-	-	Extracting from %s
unicode	11	0x00061340	-	-	Next volume
unicode	29	0x0006136A	-	-	The archive header is corr
unicode	5	0x00061390	-	-	Close
unicode	55	0x000613D8	-	-	ErroraErrors encountered
unicode	47	0x00061440	-	-	Look at the information w
unicode	5	0x00061476	-	-	bytes
unicode	11	0x00061488	-	-	modified on
unicode	24	0x000614AD	-	-	folder is not accessible
unicode	33	0x000614F2	-	-	lSome files could not be c
unicode	112	0x00061585	-	-	Please close all applicatio
unicode	55	0x0006162E	-	-	Please download a fresh c
unicode	10	0x00061673	-	-	Copyright
unicode	8	0x00061687	-	-	1993-%d
unicode	66	0x000616FF	-	-	Extracting files to %s fold
unicode	19	0x0006176C	-	-	Extraction progress
unicode	62	0x000617C5	-	-	=Total path and file name
unicode	8	0x0006183F	-	-	Continue
unicode	92	0x000618B7	-	-	Security warningKPlease r
unicode	15	0x000619AE	-	-	VS_VERSION_INFO
unicode	14	0x00061A09	-	-	StringFileInfo

ha256: E96266D760DE5DD26EEAC37B8D81CA206F2EABE57A89F4EFFBB1AC0F2D76BF84 cpu: 32-bit file-type: executable subsystem: GUI

4. Packer

- Lots of time attackers packed the malware to prevent detection. In such cases, strings are not useful and Information becomes unreadable.
- So, to detect if a malware has been packed or not, we use a tool called exeinfope. This tool will also give information about the packer which has been used to pack the malware.



Exeinfo PE - ver.0.0.6.0 by A.S.L - 1066+86 sign 2020.01.14

File : wrar591.exe

Entry Point : 0008E580 oo < EP Section : UPX1

File Offset : 00032980 First Bytes : 60.BE.00.C0.45

Linker Info : 14.00 SubSystem : Windows GUI

File Size : 002C2D58h < N Overlay : 0027C158

Image is 32bit executable RES/OVL : 2 / 89 % 2020

Rar Sfx Archive (UPX stub) (sign like UPX packer) , Overlay : RAR a

Lamer Info - Help Hint - Unpack info 0 ms.

Big sec. 2 [UPX1] , try WinRar.exe to view files - www.rarlab.com

In above example UPX software has been used to pack the malware.

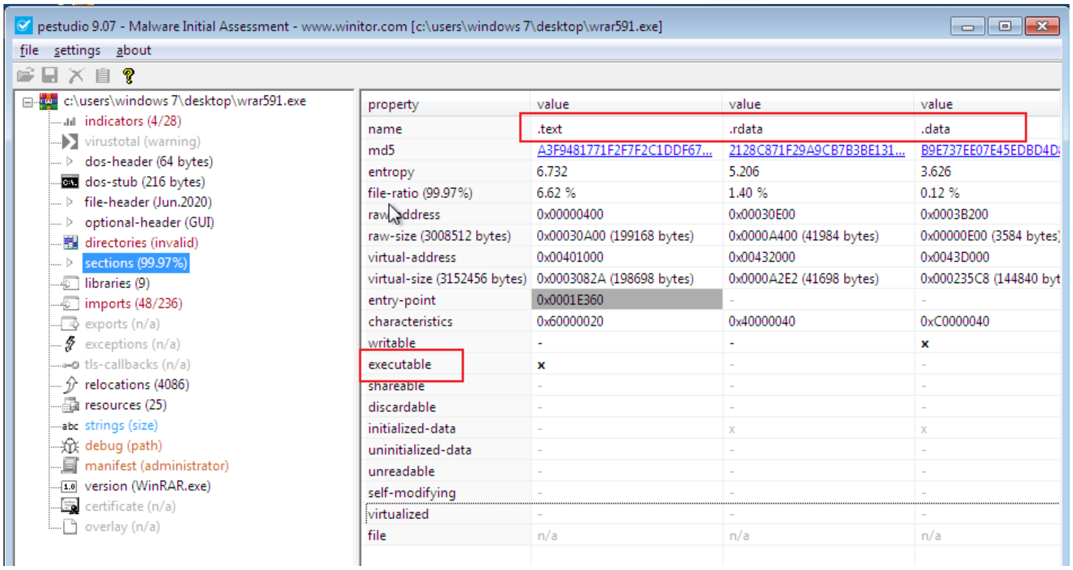
5. PE Header

- OS requires informations to run any executable file, PE header contains that information.
- Informations like
  - how the malware interacts with OS
  - Where the executable needs to be loaded in the memory
  - Libraries that the executable require to be loaded
  - From where the execution begins

These informations can be extracted from the PE header.

Sections in PE header

Section Name	Function
.code / .text	Executable code
.data	Stores Data (Readable/Writable)
.rdata	Stores Data (Read Only data)
.idata	Stores the Import Data
.edata	Stores Export Data
.rsrc	Stores Resources (Strings, icons)

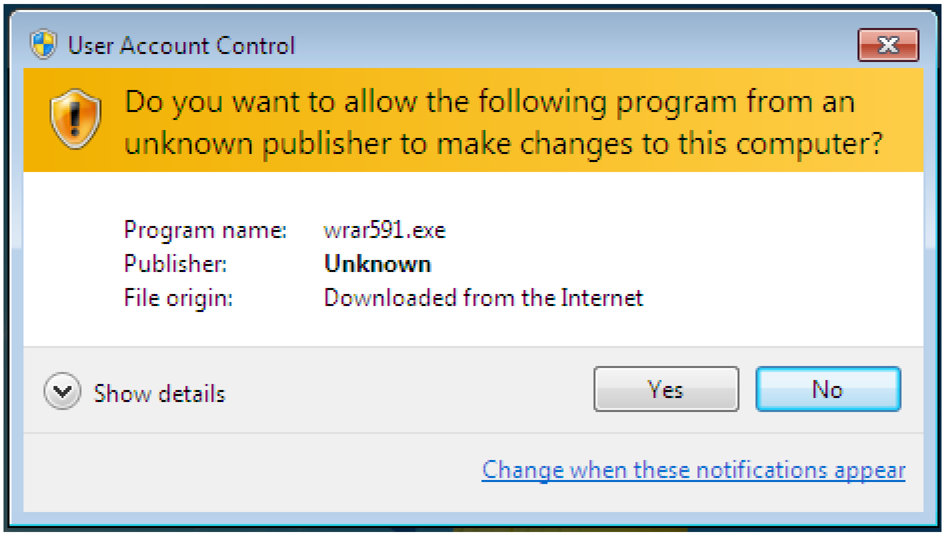

pestudio 9.07 - Malware Initial Assessment - www.winitor.com [c:\users\windows 7\desktop\wrar591.exe]
file
settings
about

c:\users\windows 7\desktop\wrar591.exe
- indicators (4/28)
- virustotal (warning)
- dos-header (64 bytes)
- dos-stub (216 bytes)
- file-header (Jun.2020)
- optional-header (GUI)
- directories (invalid)
- sections (99.97%)
- libraries (9)
- imports (48/236)
- exports (n/a)
- exceptions (n/a)
- tls-callbacks (n/a)
- relocations (4086)
- resources (25)
- strings (size)
- debug (path)
- manifest (administrator)
- version (WinRAR.exe)
- certificate (n/a)
- overlay (n/a)

property	value	value	value
name	.text	.rdata	.data
md5	A3F9481771F2F7F2C1DDF67...	2128C871F29A9CB7B3BE131...	B9E737EE07E45EDBD4D
entropy	6.732	5.206	3.626
file-ratio (99.97%)	6.62 %	1.40 %	0.12 %
raw-address	0x00000400	0x00030E00	0x0003B200
raw-size (3008512 bytes)	0x00030A00 (199168 bytes)	0x0000A400 (41984 bytes)	0x00000E00 (3584 bytes)
virtual-address	0x00401000	0x00432000	0x0043D000
virtual-size (3152456 bytes)	0x0003082A (198698 bytes)	0x0000A2E2 (41698 bytes)	0x000235C8 (144840 byt
entry-point	0x0001E360	-	-
characteristics	0x60000020	0x40000040	0xC0000040
writable	-	-	x
executable	x	-	-
shareable	-	-	-
discardable	-	-	-
initialized-data	-	x	x
uninitialized-data	-	-	-
unreadable	-	-	-
self-modifying	-	-	-
virtualized	-	-	-
file	n/a	n/a	n/a

Optional work

Usually genuine software has a publisher name but if someone has embedded a virus in it, its publisher changes to **Unknown** and hash will change as well. While executing this software, we got this message.

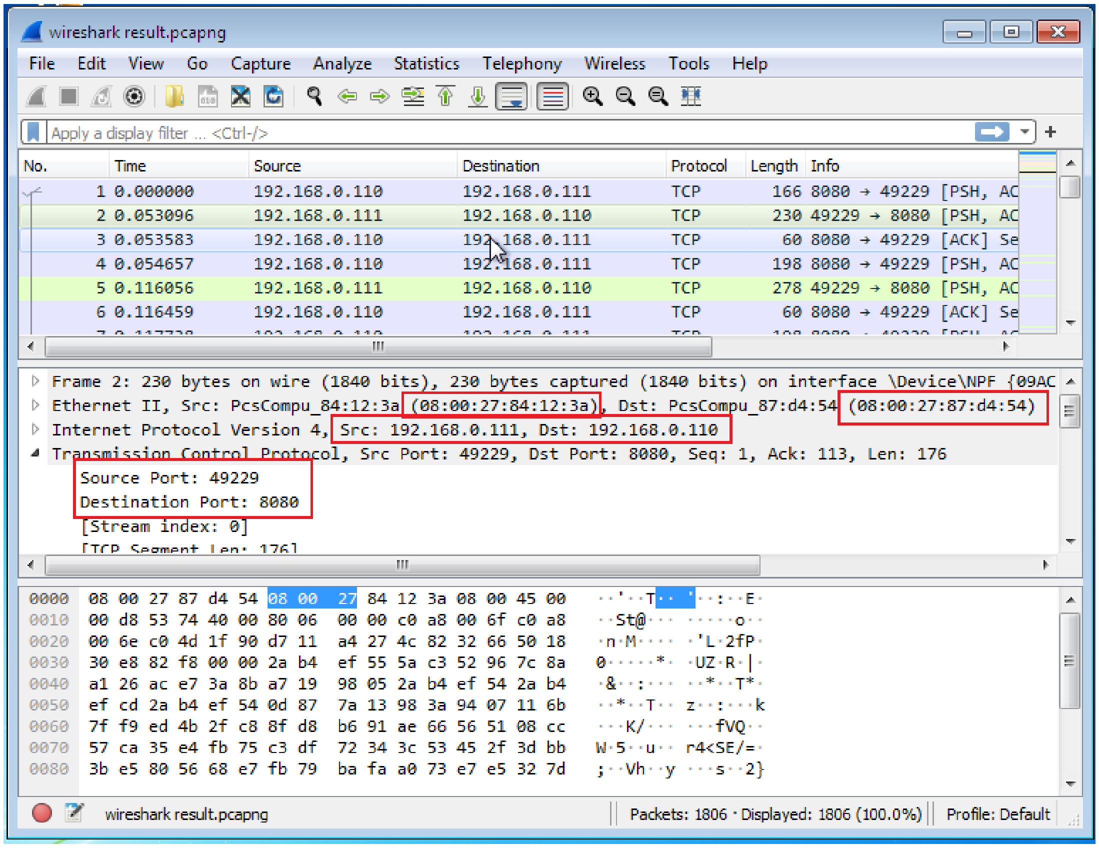


Do you want to allow the following program from an unknown publisher to make changes to this computer?

Program name:	wrar591.exe
Publisher:	**Unknown**
File origin:	Downloaded from the Internet

Monitoring the traffic using wireshark

Wireshark is a traffic analyzer. It captures the traffic and extract information like mac addresses, ip addresses, type or request, connection type, port numbers, size of headers and their messages.



```
No.      Time        Source           Destination      Protocol  Length Info
       1 0.000000    192.168.0.110    192.168.0.111    TCP          166 8080 → 49229 [PSH, AC
       2 0.053096    192.168.0.111    192.168.0.110    TCP          230 49229 → 8080 [PSH, AC
       3 0.053583    192.168.0.110    192.168.0.111    TCP           60 8080 → 49229 [ACK] Se
       4 0.054657    192.168.0.110    192.168.0.111    TCP          198 8080 → 49229 [PSH, AC
       5 0.116056    192.168.0.111    192.168.0.110    TCP          278 49229 → 8080 [PSH, AC
       6 0.116459    192.168.0.110    192.168.0.111    TCP           60 8080 → 49229 [ACK] Se
```

```
▷ Frame 2: 230 bytes on wire (1840 bits), 230 bytes captured (1840 bits) on interface \Device\NPF_{09AC
▷ Ethernet II, Src: PcsCompu_84:12:3a (08:00:27:84:12:3a), Dst: PcsCompu_87:d4:54 (08:00:27:87:d4:54)
▷ Internet Protocol Version 4, Src: 192.168.0.111, Dst: 192.168.0.110
◢ Transmission Control Protocol, Src Port: 49229, Dst Port: 8080, Seq: 1, Ack: 113, Len: 176
    Source Port: 49229
    Destination Port: 8080
    [Stream index: 0]
```

```
0000  08 00 27 87 d4 54 08 00  27 84 12 3a 08 00 45 00   ··'··T·· '··:··E·
0010  00 d8 53 74 40 00 80 06  00 00 c0 a8 00 6f c0 a8   ··St@··· ·····o··
0020  00 6e c0 4d 1f 90 d7 11  a4 27 4c 82 32 66 50 18   ·n·M···· ·'L·2fP·
0030  30 e8 82 f8 00 00 2a b4  ef 55 5a c3 52 96 7c 8a   0·····*· ·UZ·R·|·
0040  a1 26 ac e7 3a 8b a7 19  98 05 2a b4 ef 54 2a b4   ·&··:··· ··*··T*·
0050  ef cd 2a b4 ef 54 0d 87  7a 13 98 3a 94 07 11 6b   ··*··T·· z··:···k
0060  7f f9 ed 4b 2f c8 8f d8  b6 91 ae 66 56 51 08 cc   ···K/··· ···fVQ··
0070  57 ca 35 e4 fb 75 c3 df  72 34 3c 53 45 2f 3d bb   W·5··u·· r4<SE/=·
0080  3b e5 80 56 68 e7 fb 79  ba fa a0 73 e7 e5 32 7d   ;··Vh··y ···s··2}
```

wireshark result.pcapng | Packets: 1806 · Displayed: 1806 (100.0%) | Profile: Default

Information extracted from analysing the traffic

Ip address of attacker: 192.168.0.110

MAC address of attacker: 08:00:27:87:d4:54

Port at which attacker is listening connection: 8080

Type of protocol: TCP

RESULT ANALYSIS

Sample software contains a backdoor (malware)

- File type: executable (It is winrar archiver software)
- Target OS: windows
- Architecture: 32 bit
- Information extracted from the strings.txt
  - It is executable
  - Can download and upload files
  - Has access to read and write other files
  - Can send messages/requests to another host/service on internet
  - Monitor keystrokes
  - Has access to mic and webcam
- Generated hash
  - Md5: 494974D90EB81F4A9EF70DB883F91BD7
  - Compare with hash md5: 570f55d59e9e2416272407edf2c61b15
- This malware has been packed using UPX packer
- Result from Wireshark
  - Ip address of attacker: 192.168.0.110
  - MAC address of attacker: 08:00:27:87:d4:54
  - Port at which attacker is listening connection: 8080
  - Type of protocol: TCP

REFERENCES

[1] Wang, T. Y., Wu, C. H., & Hsieh, C. C. (2009, August). Detecting unknown malicious executables using portable executable headers. In INC, IMS and IDC, 2009. NCM'09. Fifth International Joint Conference on (pp. 278-284). IEEE.

[2] Jain, A., & Singh, A. K. (2017, August). Integrated Malware analysis using machine learning. In 2017 2nd International Conference on Telecommunication and Networks (TEL-NET) (pp. 1- 8). IEEE

[3] Muhammad Ijaz, Muhammad Hanif Durad, Maliha Ismail.Static and Dynamic Malware Analysis Using Machine Learning. Conference on Applied Sciences & Technology (IBCAST) Islamabad, Pakistan.

[4] Om Prakash Samantray, Satya Narayan Tripathy, Susanta Kumar Das. A study to Understand Malware Behavior through Malware Analysis. Conference on system computation Automation and Networking 2019.

[5] Tools used and their official links:

- Kali Linux: https://www.kali.org/
- Pestudio: https://www.winitor.com/
- Hashcalc: https://www.slavasoft.com/hashcalc/
- Exeinfo PE: http://exeinfo.atwebpages.com/
- Windows PowerShell: https://www.microsoft.com/
- Window 7 for testing: https://www.microsoft.com/
- Wireshark: https://www.wireshark.org/