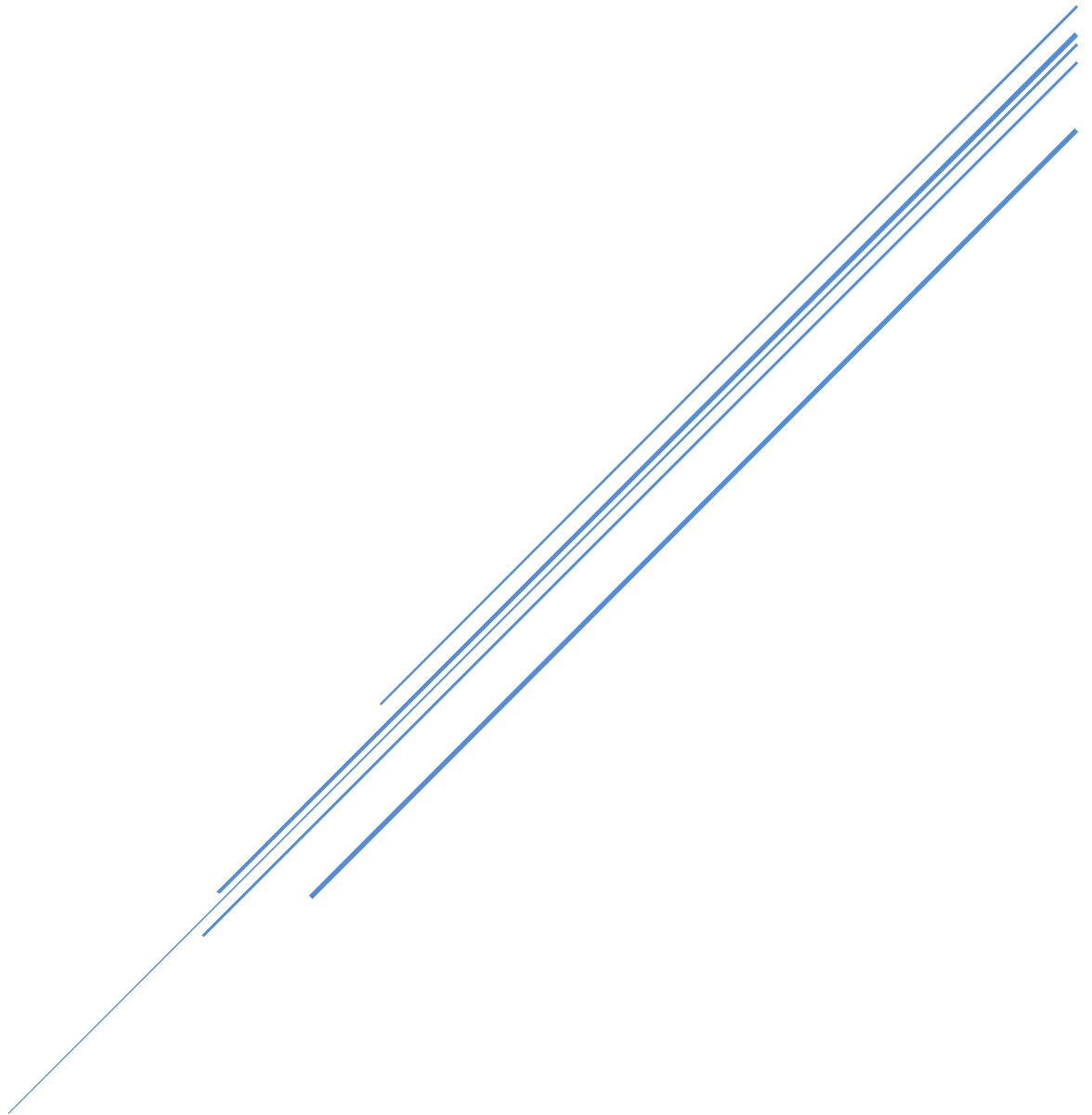


# Malware Analysis: Static and Dynamic



Information Security

## Table of Contents:

<b>Malware Analysis: Introduction</b>	<b>2</b>
<b>Static and Dynamic Malware Analysis using Existing Tools</b>	<b>3</b>
Static Analysis	3
Dynamic Malware Analysis	12
<b>Malware Analysis Prototype Development</b>	<b>16</b>
Static Malware Analysis	17
Dynamic Malware Analysis:	20
<b>Literature Review:</b>	<b>27</b>
• Methodologies and Tools	27
• Challenges and Limitations	28
• Comparative Analysis	28

## Malware Analysis: Introduction

There are two categories when we are talking about malware analysis:

The first category is **static analysis**, and the second one is dynamic analysis. Static analysis is a method of analyzing a sample or a file at the state it presents itself as without executing the file. So for example if we received an email from someone we don't know that includes a file in the email, what we can do is download the file and not execute it. Now, there are many ways to perform static analysis on a file, something like signature-based detection or permission-based detection, or a source code review.

**Dynamic analysis** essentially involves executing the sample or the file on your machine and then observing what is going to happen. This is, of course, not safe because this file could be a malware, or could be a ransomware that ends up encrypting all of your files. When we have received an email that includes a file, what we can do with dynamic analysis is to download this file, but this time we are going to execute it. And when we execute it, we are going to monitor how this file actually behaves on our machine. What IP address this file is trying to connect to, or what registry key this file is

trying to create or modify on our machine, or whether this file is trying to download something from the internet.

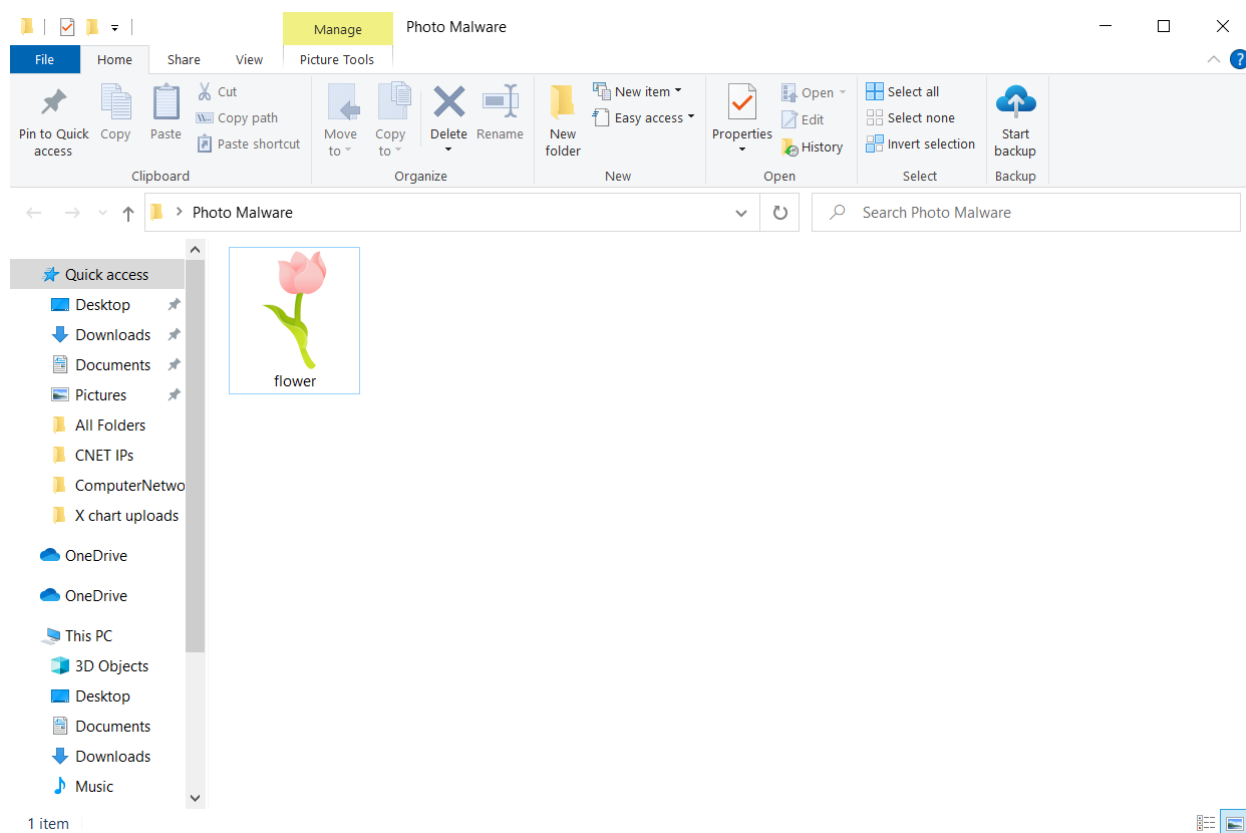
It's not always necessary to download and execute the payload on our machine. Instead, we can use some online services that are called online sandboxing that allow us to upload our file that we think is suspicious to its server and then it will perform the dynamic analysis for us and this is going to be much safer for us. We can see what IP addresses the file is trying to connect to, what registry keys are trying to create or modify, and so on.

## **Static and Dynamic Malware Analysis using Existing Tools**

### **Static Analysis**

#### **File Extension**

First we are going to start with static analysis on Windows and then we can move and use static or dynamic analysis on Linux or Windows. Here we have a file that's called **flower**



It has an icon that seems like an image. Instead, it could be something tempting so that you can click on it. As you know that by default, Windows would hide the extension of the files. This is status analysis so you don't actually have to open the file but just for the sake of demonstration, if you open the file,

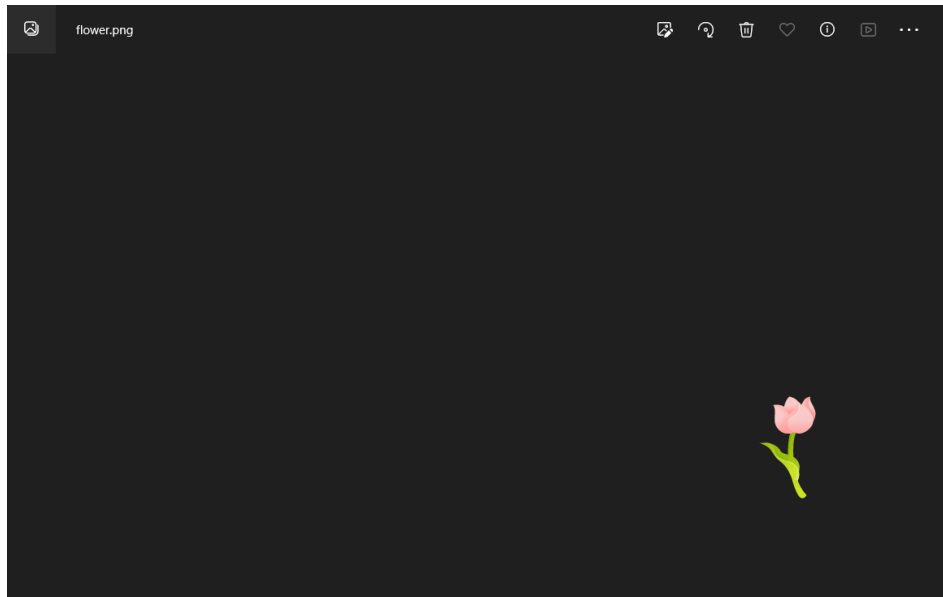
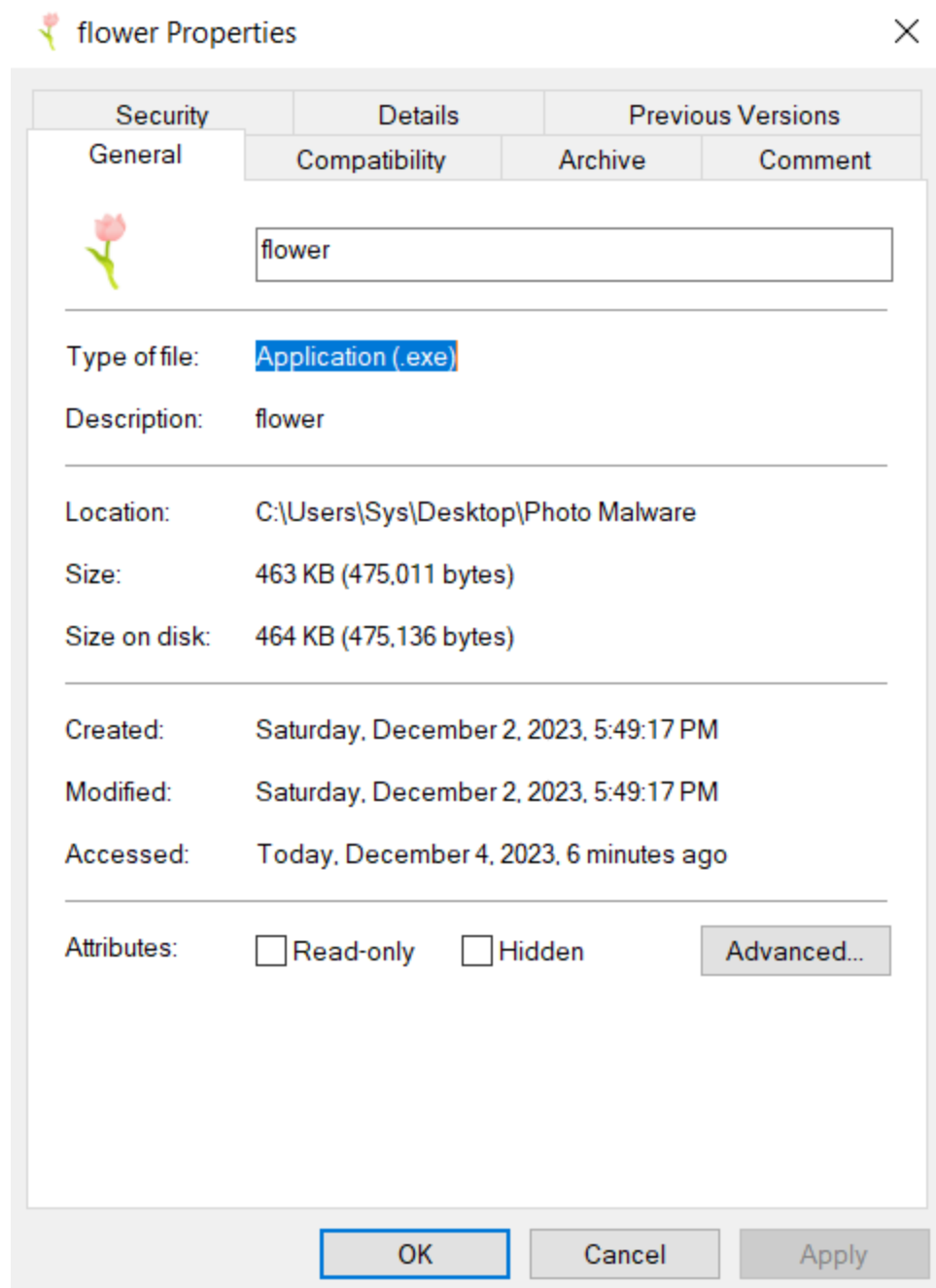


Image of a flower is displayed and at the top left you can see the **png extension**.

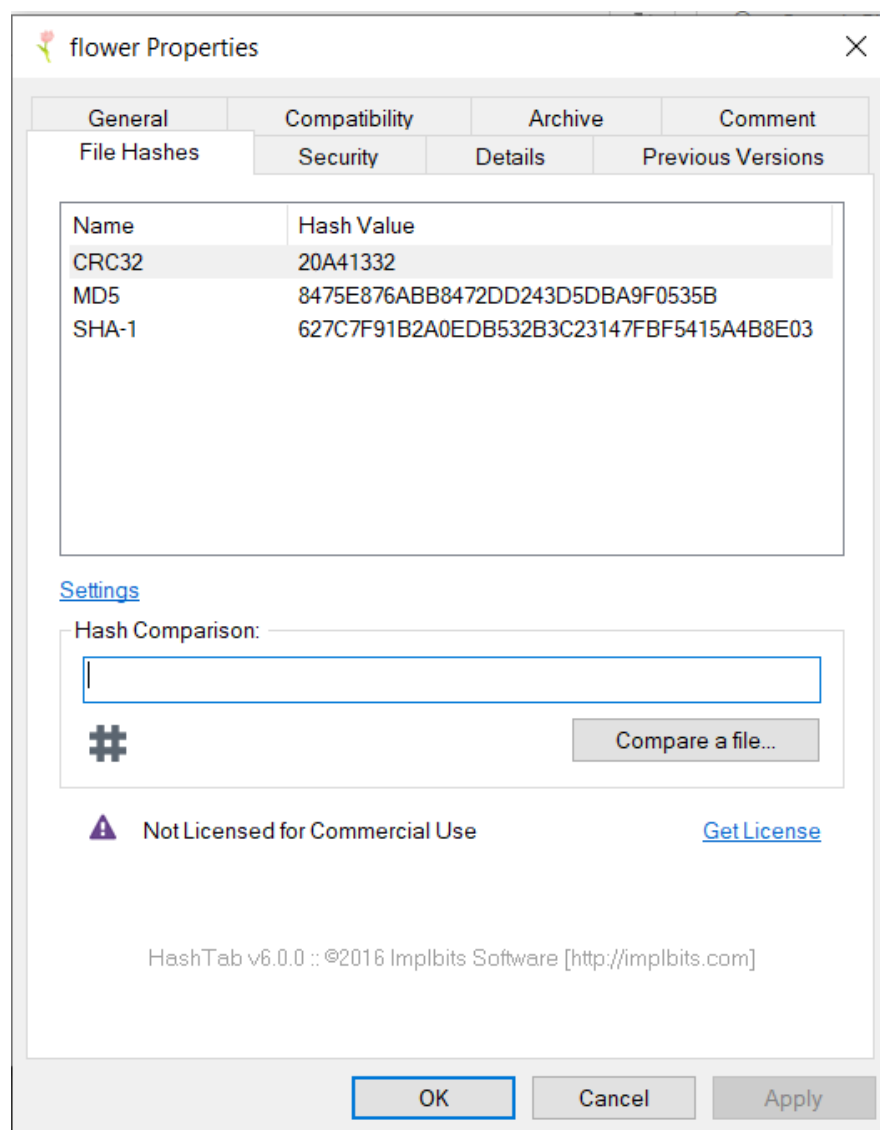
You can either go and view the hidden extensions by changing the settings or you can simply right-click on it and go to Properties, and you can see that the **type of the file is EXE**.



Another thing we can do is to extract the strings that are inside this file. We are going to do this in Linux. I'm not going to do it in Windows because I haven't installed the string tool that will allow me to extract all the strings that are within this executable file or this image.

## Extract the Hash

Another thing you can do is that you can extract the hash of the file and then check online whether this is a malicious file or not. I have installed a tool called **Hashtab** When you go to properties, you're going to see the file hashes tab here, and when you click on it, you will get the **md5 hash of this file and the sha1 hash** of this file.



What you can basically do is copy the MD5 hash and then go to a website like Virustotal



Analyse suspicious files, domains, IPs and URLs to detect malware and other breaches, automatically share them with the security community.

go to search and paste the md5 hash.

A screenshot of the VirusTotal website's search interface. The browser address bar shows 'virustotal.com/gui/home/search'. The page has tabs for 'FILE', 'URL', and 'SEARCH', with 'SEARCH' being the active tab. A search input field contains the MD5 hash '8475E876ABB8472DD243D5DBA9F0535B'. Below the input field, there is a disclaimer about sharing data with the security community. A 'flower Properties' dialog box is open in the foreground, showing the 'File Hashes' tab. It lists the Name, Hash Value, CRC32, MD5, and SHA-1 for the file. The MD5 value matches the one in the search field. The dialog also shows a 'Hash Comparison' section with the same MD5 hash and a 'Compare a file...' button. The footer of the page includes links for VirusTotal, Community, Tools, Premium Services, and Documentation.

virustotal.com/gui/home/search

Analyse suspicious files, domains, IPs and URLs to detect malware and other breaches, automatically share them with the security community.

FILE URL **SEARCH**

Search for a hash, domain, IP address, URL or gain additional context and threat landscape visibility with [VT ENTERPRISE](#).

8475E876ABB8472DD243D5DBA9F0535B

By submitting data above, you are agreeing to our [Terms of Service](#) and [Privacy Policy](#), and to the **sharing of your sample submission with the security community**. Please do not submit any personal information; VirusTotal is not responsible for the contents of your submission. [Learn more](#).

Want to automate submissions? [Check our API](#), or access your [API key](#).

**flower Properties**

General	Compatibility	Archive	Comment
<b>File Hashes</b>	<b>Security</b>	<b>Details</b>	<b>Previous Versions</b>
Name	Hash Value		
CRC32	20A41332		
MD5	8475E876ABB8472DD243D5DBA9F0535B		
SHA-1	627C7F91B2A0EDB532B3C23147FBF5415A48BE03		

**Settings**

Hash Comparison:

8475E876ABB8472DD243D5DBA9F0535B

MD5

Not Licensed for Commercial Use [Get License](#)

HashTab v6.0.0 - ©2016 Implants Software (<http://implbits.com>)

OK Cancel Apply

Now, of course, this is only a file that I have created. Nobody scanned this file, that's why it's not suspicious.





## No matches found

Alternatively, do you want to locate your threat based on static, dynamic, content, attribution or other advanced IoC context? VT Intelligence allows you to search across VirusTotal's entire threat corpus using a [myriad of modifiers](#), [learn more](#).

[Try out VT Enterprise](#)

[Try a new search](#)


So this is not always a reliable way to check whether this is malicious or not.

But assuming that you have received an email that could be a spam email, and this email has been sent to hundreds, thousands of people, and then people downloaded it and executed it, and then the antivirus reported that it's a malware, then all the results will also be shared with Virustotal. So when you copy its MD5 hash and then put it in VirusTotal, you will get a result because it's a famous malware that's been detected by various antiviruses. For example, when we search for the **WannaCry** hash and we put it in VirusTotal,

FILE

URL

SEARCH



Search for a hash, domain, IP address, URL or gain additional context and threat landscape visibility with [VT ENTERPRISE](#).

db349b97c37d22f5ea1d1841e3c89eb4

By submitting data above, you are agreeing to our [Terms of Service](#) and [Privacy Policy](#), and to the **sharing of your Sample submission with the security community**. Please do not submit any personal information; VirusTotal is not responsible for the contents of your submission. [Learn more](#).

Want to automate submissions? [Check our API](#), or access your [API key](#).

we will see a completely different result. Let me share it for it. This is the service that runs when you execute the WannaCry. When we copy the MD5 hash and then go to virus total and they put it here and then hit Enter, and as you can see

70

172

Community Score

70 security vendors and 5 sandboxes flagged this file as malicious

Reanalyze

Similar

More

24d004a104d4d54034dbcffc2a4b19a11f39008a575aa614ea04703480b1022c

Size

3.55 MB

Last Analysis Date

1 hour ago

EXE

lhdfgrui.exe

peexe

malware

macro-create-ole

runtime-modules

detect-debug-environment

checks-network-adapters

exploit

cve-2017-0147

long-sleeps

direct-cpu-clock-access

checks-user-input

cve-2017-0144

DETECTION

DETAILS

RELATIONS

BEHAVIOR

COMMUNITY 30 +

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to [automate checks](#).

Popular threat label

trojan.wannacry/wanna

Threat categories

trojan ransomware worm

Family labels

wannacry wanna wannacryptor

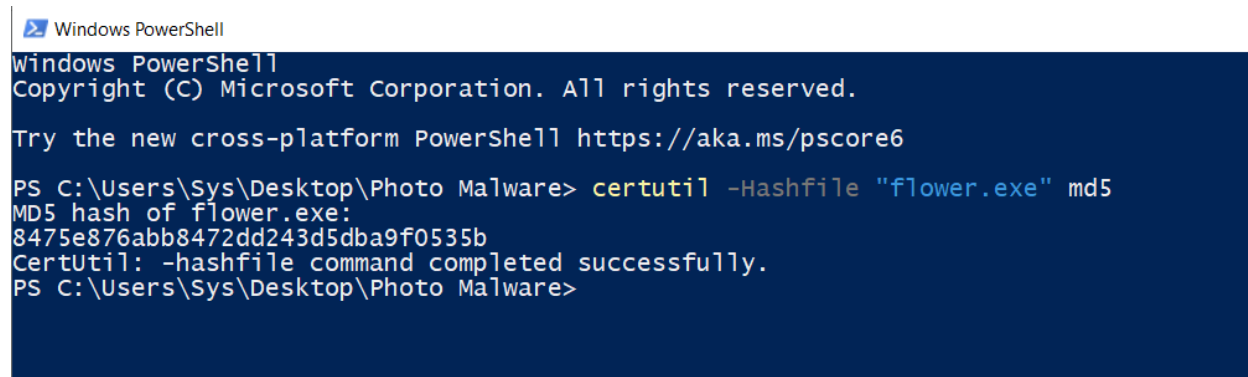
Security vendors' analysis

Do you want to automate checks?

Acronis (Static ML)	Suspicious	AhnLab-V3	Trojan/Win32.WannaCryptor.R200572
Alibaba	Ransom:Win32/WannaCry.398	ALYac	Trojan.Ransom.WannaCryptor
Antiy-AVL	Trojan[Ransom]/Win32.Wanna	Arcabit	Trojan.Ransom.WannaCryptor.H
Avast	Sf:WNCryLdr-A [Trj]	AVG	Sf:WNCryLdr-A [Trj]
Avira (no cloud)	TR/Ransom.IZ	Baidu	Win32.Worm.Rbot.a
BitDefender	Trojan.Ransom.WannaCryptor.H	BitDefenderTheta	Genc:NN.ZexaF.36608.Jt0@aePsbmpi

**It's been detected by 70 antiviruses out of 72.** You can also see some more information like, what it is, its exploit, long-sleeps, macro-create-ole etc, . You can read much more about it and what the antiviruses categorized this file as.

You don't have to use a tool to extract it for you, although it's much easier than using the command line. To do so, I'm going to open powershell in this directory. The comment that we are going to use is: **certutil -Hashfile "flower.exe" md5**. We have to specify the file hash and then the filename, which is called flower.exe, and then the type of the hash we want to extract, whether it's md5, sha1, sha256, etc. I'm going to say md5 and hit Enter. As you can see, this is the MD5 hash of the file without using any tools.

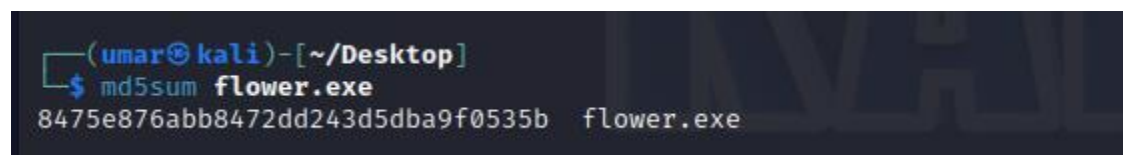


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/powershell

PS C:\Users\Sys\Desktop\Photo Malware> certutil -Hashfile "flower.exe" md5
MD5 hash of flower.exe:
8475e876abb8472dd243d5dba9f0535b
CertUtil: -hashfile command completed successfully.
PS C:\Users\Sys\Desktop\Photo Malware>
```

**Now on Linux**, and what we are going to analyze is this file on the desktop, which is called test.exe or the same flower.exe. Now let's see how we can extract the hash of this file on Linux. So to do so, it's very simple and easy. All we have to do in order for us to extract the MD5 hash is to type in the command **md5sum [filename]**



```
(umar@kali)-[~/Desktop]
$ md5sum flower.exe
8475e876abb8472dd243d5dba9f0535b flower.exe
```

This is the MD5 hash of the file. Now if you want to extract the sha1 hash, you can just type sha1sum, and then the filename, and here is the hash. You can copy this hash and

put it in VirusTotal to check whether it's malicious or not. You can, of course, upload it if you don't want to extract the hash.

## Extracting the strings

Another command that is useful is by extracting the strings. We can just type strings and then the name of the file. **strings test.exe**

```
(umar@kali)-[~/Desktop]
$ strings test.exe
!This program cannot be run in DOS mode.
.text
P`.data
.rdata
00/4
00.bss
.idata
.CRT
.tls
0B/20
```

Here are all the strings that we can view in this file. There is a lot of text and if you have noticed anything suspicious, you can search for it online to see if this file is legit or not. The test.exe file I have over here is actually an executable of C++ code I myself write that just creates a simple .txt file

## Dynamic Malware Analysis

Now let's assume that we have received a file which is the malware.exe and I know that it looks very suspicious so I'm not going to open it. It is actually a malware I downloaded from a website called **Malware Bazaar** and I renamed it to **malware.exe**. There are other websites like Virus Share, VX Heaven etc where you can download malwares as well. Be sure to download and run them in a controlled environment such as Virtual Box.

MALWAREbazaar  
by ABUSE[REDACTED]
Browse Upload Hunting API Export Statistics FAQ About Login

## Browse Database

See search syntax see below, example: tag:TrickBot
Search

Search Syntax ⓘ

Search: 29

Date (UTC)	SHA256 hash	Type	Signature	Tags	Reporter	DL
2023-12-04 12:29	eb2697299983caf35d74d...	exe	RiseProStealer	exe RiseProStealer	andretavare5	
2023-12-04 11:58	2922fdceb5d931d1492d...	exe		exe	andretavare5	
2023-12-04 11:29	e6438f002d7e462be296...	exe		exe	Anonymous	
2023-12-04 11:11	b1ffc4c87c2210706d...	exe	RiseProStealer	exe RiseProStealer	andretavare5	
2023-12-04 11:11	5fb3a526d0e47584efd56...	exe		exe	TeamDreier	

First I did static analysis on it using VirusTotal using its md5 hash and it was deemed as malicious by 19/72 vendors

19  
/ 72

Community Score

19 security vendors and 1 sandbox flagged this file as malicious

2922fdceb5d931d1492d4d8d7c7206d36aa18659cad221667fcb0b53ee0ae4dc  
wm.exe

Size  
3.04 MB

peexe malware checks-cpu-name detect-debug-environment checks-user-input persistence

As I said before, dynamic analysis involves executing the file and this is what we are going to do. Although this is dangerous and I do not recommend it if you are not very familiar with malware and malware analysis. Now we have a file that is suspicious. We might copy it or download it in a virtual machine so that we are out of any sort of danger from this file. Another thing we can do instead of executing the payload or the program or the suspicious file on our machine is that we can upload it to online sandboxing services that will execute this file and analyze how it behaves and then it will give us a report. One of the most famous websites are **Any Run** and a website called **Hybrid Analysis**. So here is the first website Hybrid Analysis.


[File/URL](#)
[File Collection](#)
[Report Search](#)
[YARA Search](#)
[String Search](#)

This is a free malware analysis service for the community that detects and analyzes unknown threats using a unique Hybrid Analysis technology.

You can just upload your file and then say, Analyze. It will take like 5-10 minutes until it gives you the full report.

After running this file on a Windows 10 Operating System, Hybrid Analysis has deemed this file as Malicious with a **threat score of 100/100**

The screenshot shows the "Analysis Overview" page for a file named "malware.exe". The page includes a navigation bar with the Hybrid Analysis logo, a search bar, and a "Request Info" link. The main content area displays various file details: Submission name (malware.exe), Size (3MiB), Type (peexe, executable), Mime (application/x-dosexec), SHA256 (2922fdceb5d931d1492d4d8d7c7206d36aa18659cad221667fcb0b53ee0ae4dc), Operating System (Windows), Last Anti-Virus Scan (12/04/2023 12:59:23 (UTC)), and Last Sandbox Report (12/04/2023 12:59:23 (UTC)). On the right side, there is a red "malicious" badge, a "Threat Score: 100/100", "AV Detection: 34%", and a label "Win/malicious\_confidence\_90%". At the bottom right, there are links for "Link", "Twitter", and "E-Mail".

Analysis Overview [Request Report Deletion](#)

Submission name:	malware.exe	<b>malicious</b>
Size:	3MiB	
Type:	peexe executable	Threat Score: 100/100
Mime:	application/x-dosexec	AV Detection: 34%
SHA256:	2922fdceb5d931d1492d4d8d7c7206d36aa18659cad221667fcb0b53ee0ae4dc	Labeled as:
Operating System:	Windows	Win/malicious_confidence_90%
Last Anti-Virus Scan:	12/04/2023 12:59:23 (UTC)	<a href="#">Link</a> <a href="#">Twitter</a>
Last Sandbox Report:	12/04/2023 12:59:23 (UTC)	<a href="#">E-Mail</a>

Furthermore, it also tells us that the malware tried to contact a host with IP address 195.20.16.45

The screenshot displays the HYBRID network analysis tool interface. A modal window titled "Network Analysis Overview" is open, showing a table of "Contacted Hosts". The table has four columns: "IP Address", "Port/Protocol", "Associated Process", and "Details". One host is listed with IP 195.20.16.45, Port/Protocol -, Associated Process -, and Details "Country n/a". A "Close" button is in the bottom right of the modal. Below the modal, the main interface shows a "Network Behavior" section with a "View all details" button.

**Network Analysis Overview**

Contacted Hosts

Login to Download Contacted Hosts (CSV)

IP Address	Port/Protocol	Associated Process	Details
195.20.16.45	-	-	Country n/a

Close

**Evasive** Possibly tries to evade analysis by sleeping many times

**Spreading** Opens the MountPointManager (often used to detect additional infection locations)

**Network Behavior** Contacts 1 host: [View all details](#)

And this is the second website that I think is also very good. You can register on it and then you can use it.

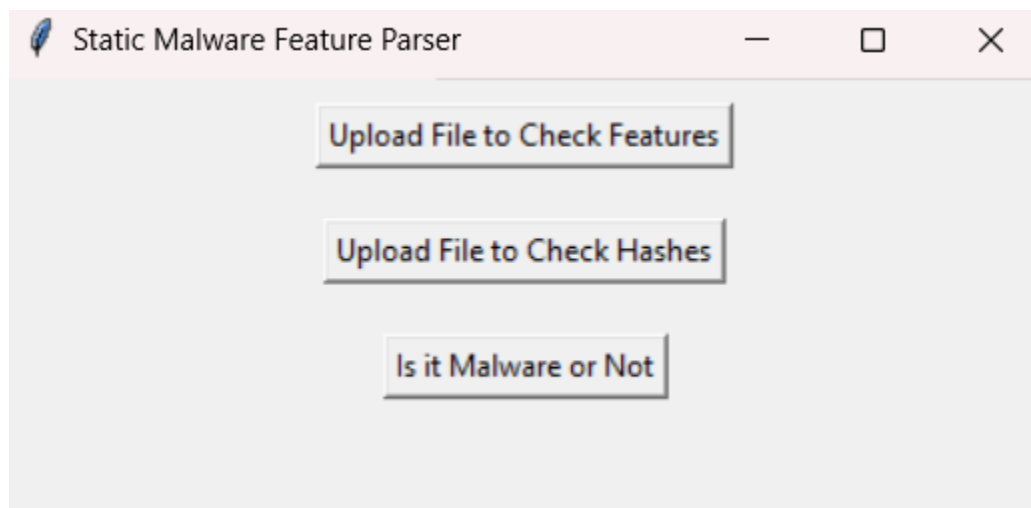


This was a basic introduction on how we can analyze a file that we have received by email or we have got in a USB drive. We have learned the basics of static analysis and dynamic analysis.

## Malware Analysis Prototype Development



## Static Malware Analysis



**Extracting Features from Executable Files:** The script extracts various features from executable files (.exe) for analysis. This includes parsing specific attributes of the file using libraries like pefile.

```
import itertools
import pefile
import math
import hashlib

file_path = 'e4dcfe552c8f34aa797aeb9b68988edb50ebf185e67f0b173ef7e8c57685b0.exe'

def get_entropy(data):
    if not data:
        return 0

    entropy = 0
    for x in range(256):
        p_x = float(data.count(chr(x)))/len(data)
        if p_x > 0:
            entropy += - p_x*math.log(p_x, 2)
    return entropy

def EXTRACT_OPTIONAL_HEADER_INFO(LResource,Resources_header,file_path,fileName):

    pefile_info = pefile.PE(file_path)

    LResource[Resources_header['Name']] = fileName
    LResource[Resources_header['MajorLinkerVersion']] = pefile_info.OPTIONAL_HEADER.MajorLinkerVersion
    LResource[Resources_header['MinorLinkerVersion']] = pefile_info.OPTIONAL_HEADER.MinorLinkerVersion
    LResource[Resources_header['SizeOfCode']] = pefile_info.OPTIONAL_HEADER.SizeOfCode
    LResource[Resources_header['SizeOfInitializedData']] = pefile_info.OPTIONAL_HEADER.SizeOfInitializedData
    LResource[Resources_header['SizeOfUninitializedData']] = pefile_info.OPTIONAL_HEADER.SizeOfUninitializedData
```

**Hashing for Malware Identification:** It uses hashing techniques (like SHA256) to generate unique identifiers for files, aiding in identifying and comparing malware samples.

```
import hashlib

def calculate_hashes(file_path):
    # Create hash objects
    md5_hash = hashlib.md5()
    sha1_hash = hashlib.sha1()
    sha256_hash = hashlib.sha256()

    # Open the file in binary mode and read it in chunks
    with open(file_path, "rb") as file:
        for chunk in iter(lambda: file.read(4096), b''):
            md5_hash.update(chunk)
            sha1_hash.update(chunk)
            sha256_hash.update(chunk)

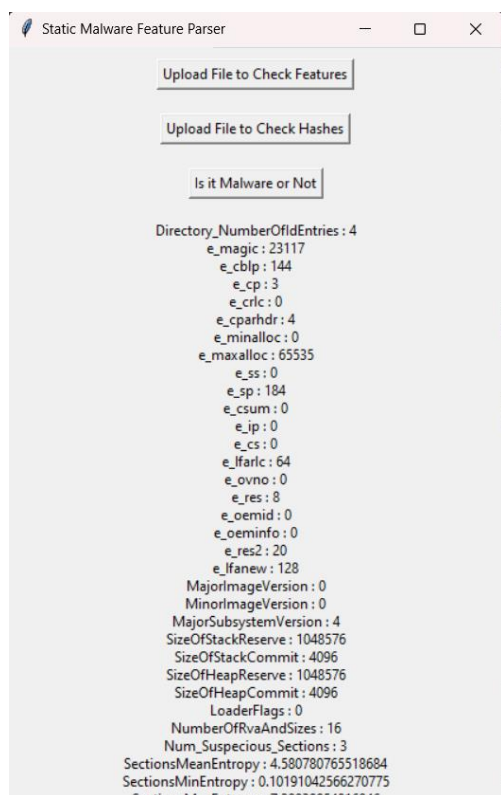
    # Return the hexadecimal representations of the hashes
    return (
        md5_hash.hexdigest(),
        sha1_hash.hexdigest(),
        sha256_hash.hexdigest()
    )

# Example usage:
file_path = 'e4dcfe552c8f34aa797aeb9b68988edb50ebf185e67f0b173ef7e8c57685b0.exe' # Replace with your actual file path
md5_hash, sha1_hash, sha256_hash = calculate_hashes(file_path)

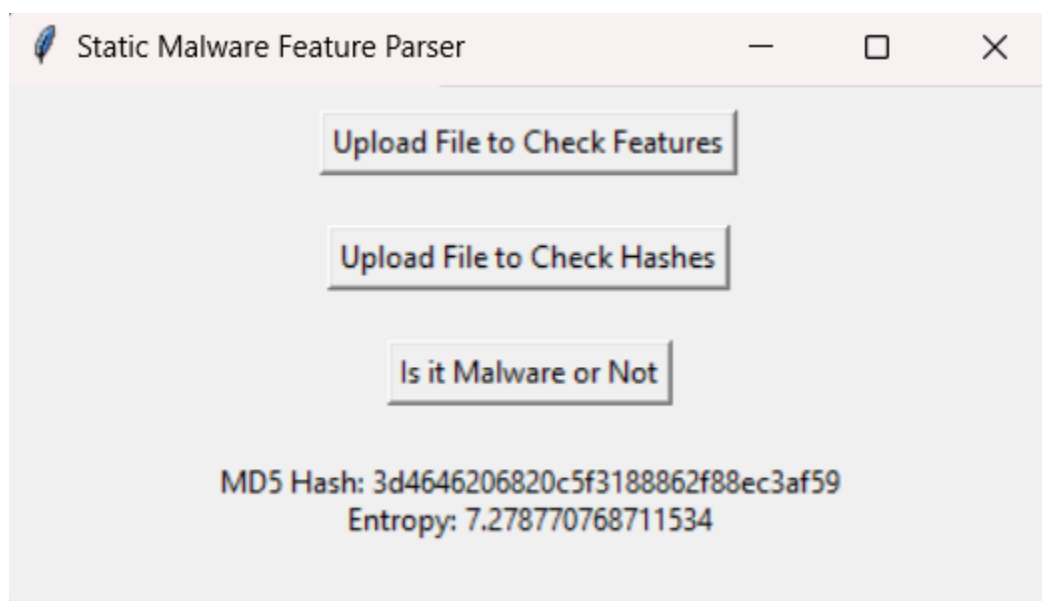
print(f"MD5 Hash of {file_path}: {md5_hash}")
print(f"SHA-1 Hash of {file_path}: {sha1_hash}")
print(f"SHA-256 Hash of {file_path}: {sha256_hash}")
```

MD5 Hash of e4dcfe552c8f34aa797aeb9b68988edb50ebf185e67f0b173ef7e8c57685b0.exe: 3d4646206820c5f3188862f88ec3af59  
SHA-1 Hash of e4dcfe552c8f34aa797aeb9b68988edb50ebf185e67f0b173ef7e8c57685b0.exe: fecc92d2e0bdfdd5abe90548dd5ea3235bc7daeb  
SHA-256 Hash of e4dcfe552c8f34aa797aeb9b68988edb50ebf185e67f0b173ef7e8c57685b0.exe: e4dcfe552c8f34aa797aeb9b68988edb50ebf185e67f0b173ef7e8c57685b0

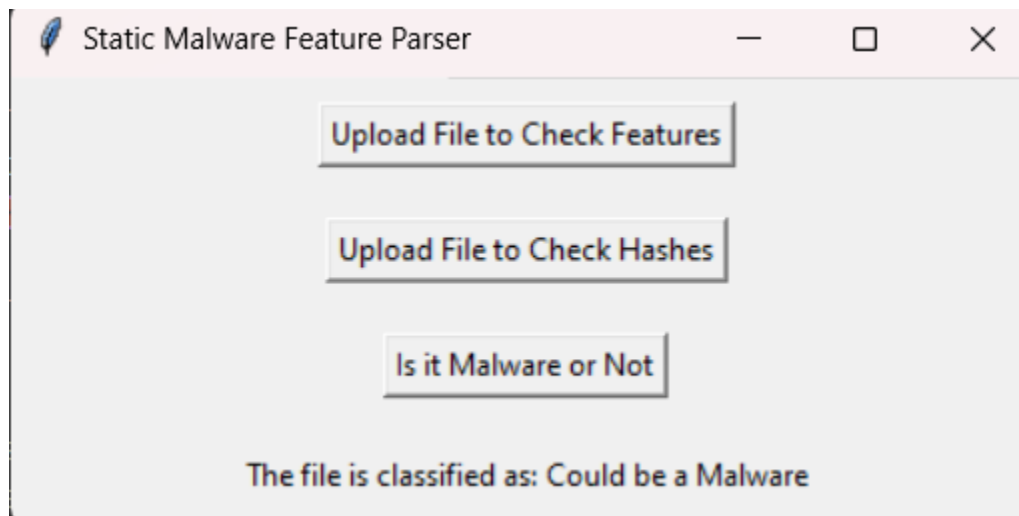
**Using External Tools for Analysis:** The script likely integrates with external tools or databases (e.g., VirusTotal) for enhanced analysis and comparison with known malware signatures.



**GUI for Analysis:** The code includes a graphical user interface (GUI) setup, allowing users to upload files for feature extraction and hash checking. This suggests an interactive platform for malware analysis.



**Malware Classification:** There's functionality for classifying a file as malicious or not, possibly based on the extracted features and comparison against known malware characteristics.

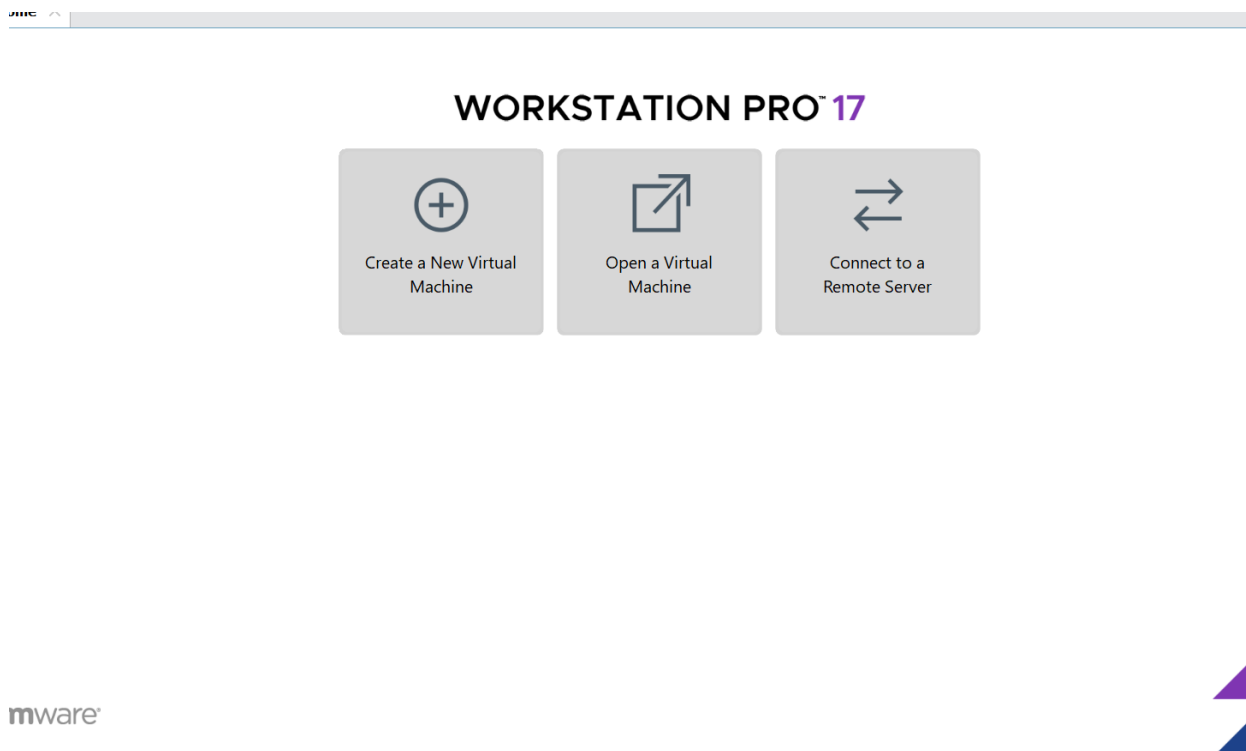


**Error Handling and User Feedback:** The script handles exceptions and provides feedback to the user, indicating a user-friendly approach to malware analysis.

**Main Loop for GUI Interaction:** The script runs a main loop for the GUI, allowing continuous user interaction and analysis execution.

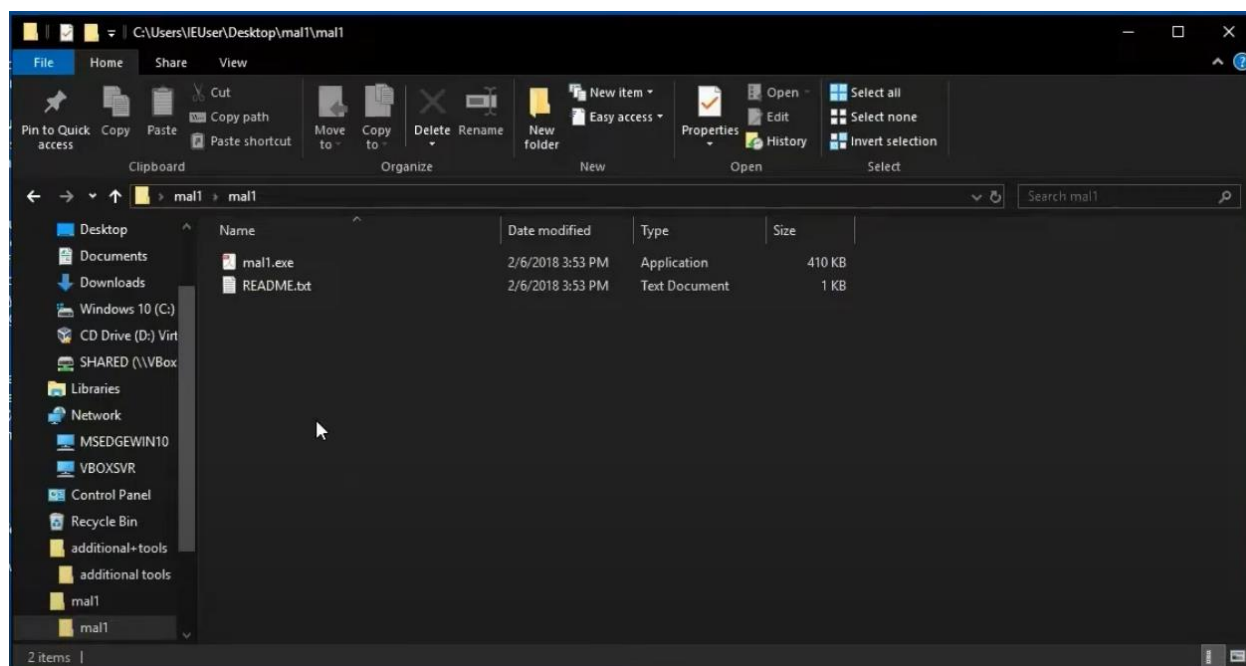
## Dynamic Malware Analysis:

In dynamic malware analysis first we will install virtualbox or vmware station then we will install windows 7/10/11.



After installing windows 7/10/11 we will open it and then install three different software for our task. Fakenet, procmonitor and then regshot. It will help in our work. First we will download the malware from the website. We will execute it before we need to on the process monitor to monitor the processes, and then fake one in order to show the malware the internet is connected to the system in reality it is not. Then regshot which takes the screenshot of the system. We take regshot before executing it and then after executing then we will compare it and analyze the output.

## 1. Malware file



## 2. Process Monitor

Process Monitor - Sysinternals: www.sysinternals.com

File Edit Event Filter Tools Options Help

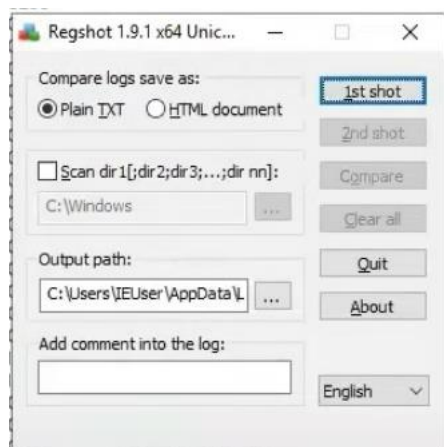
Time ...	Process Name	PID	Operation	Path	Result	Detail
10:31:...	Malware.exe	4392	Thread Create		SUCCESS	Thread ID: 3772
10:31:...	Explorer.exe	3084	QueryNameInfo...	C:\Users\IEUser\AppData\Local\Temp...	SUCCESS	Name: \Users\IEU...
10:31:...	Explorer.exe	3084	CreateFile	C:\Users\IEUser\AppData\Local\Temp...	SUCCESS	Desired Access: R...
10:31:...	Explorer.exe	3084	QueryBasicInfo...	C:\Users\IEUser\AppData\Local\Temp...	SUCCESS	CreationTime: 5/3/...
10:31:...	Explorer.exe	3084	CloseFile	C:\Users\IEUser\AppData\Local\Temp...	SUCCESS	
10:31:...	Explorer.exe	3084	CreateFile	C:\Users\IEUser\AppData\Local\Temp...	SUCCESS	Desired Access: R...
10:31:...	Explorer.exe	3084	QueryDirectory	C:\Users	SUCCESS	FileInformationClas...
10:31:...	Explorer.exe	3084	CloseFile	C:\Users	SUCCESS	
10:31:...	Explorer.exe	3084	CreateFile	C:\Users	SUCCESS	Desired Access: R...
10:31:...	Explorer.exe	3084	QueryDirectory	C:\Users\IEUser	SUCCESS	FileInformationClas...
10:31:...	Explorer.exe	3084	CloseFile	C:\Users	SUCCESS	
10:31:...	Explorer.exe	3084	CreateFile	C:\Users\IEUser	SUCCESS	Desired Access: R...
10:31:...	Explorer.exe	3084	QueryDirectory	C:\Users\IEUser\AppData	SUCCESS	FileInformationClas...
10:31:...	Explorer.exe	3084	CloseFile	C:\Users\IEUser	SUCCESS	
10:31:...	Explorer.exe	3084	CreateFile	C:\Users\IEUser\AppData	SUCCESS	Desired Access: R...
10:31:...	Explorer.exe	3084	QueryDirectory	C:\Users\IEUser\AppData\Local	SUCCESS	FileInformationClas...
10:31:...	Explorer.exe	3084	CloseFile	C:\Users\IEUser\AppData	SUCCESS	
10:31:...	Explorer.exe	3084	CreateFile	C:\Users\IEUser\AppData\Local	SUCCESS	Desired Access: R...
10:31:...	Explorer.exe	3084	QueryDirectory	C:\Users\IEUser\AppData\Local\Temp	SUCCESS	FileInformationClas...
10:31:...	Explorer.exe	3084	CloseFile	C:\Users\IEUser\AppData\Local	SUCCESS	
10:31:...	Explorer.exe	3084	RegQueryValue	HKCU\Software\Classes	SUCCESS	Query: Name
10:31:...	Explorer.exe	3084	RegQueryValue	HKCU\Software\Classes	SUCCESS	Query: HandleTag...
10:31:...	Explorer.exe	3084	RegQueryValue	HKCU\Software\Classes	SUCCESS	Query: HandleTag...
10:31:...	Explorer.exe	3084	RegOpenKey	HKCU\Software\Classes\CLSID\{56AD...	NAME NOT FOUND	Desired Access: R...
10:31:...	Explorer.exe	3084	RegOpenKey	HKCR\CLSID\{56AD4C5D-B908-4F85-...	NAME NOT FOUND	Desired Access: R...
10:31:...	Explorer.exe	3084	QueryNameInfo...	C:\Users\IEUser\AppData\Local\Temp...	SUCCESS	Name: \Users\IEU...
10:31:...	Explorer.exe	3084	CreateFile	C:\Users\IEUser\AppData\Local\Temp...	SUCCESS	Desired Access: R...
10:31:...	Explorer.exe	3084	QueryBasicInfo...	C:\Users\IEUser\AppData\Local\Temp...	SUCCESS	CreationTime: 5/3/...
10:31:...	Explorer.exe	3084	CloseFile	C:\Users\IEUser\AppData\Local\Temp...	SUCCESS	
10:31:...	Explorer.exe	3084	QueryDirectory	C:\Users	SUCCESS	Desired Access: R...
10:31:...	Explorer.exe	3084	CloseFile	C:\Users	SUCCESS	FileInformationClas...
10:31:...	Explorer.exe	3084	CreateFile	C:\Users	SUCCESS	Desired Access: R...
10:31:...	Explorer.exe	3084	QueryDirectory	C:\Users\IEUser	SUCCESS	FileInformationClas...

Showing 28,726 of 164,433 events (17%) Backed by virtual memory

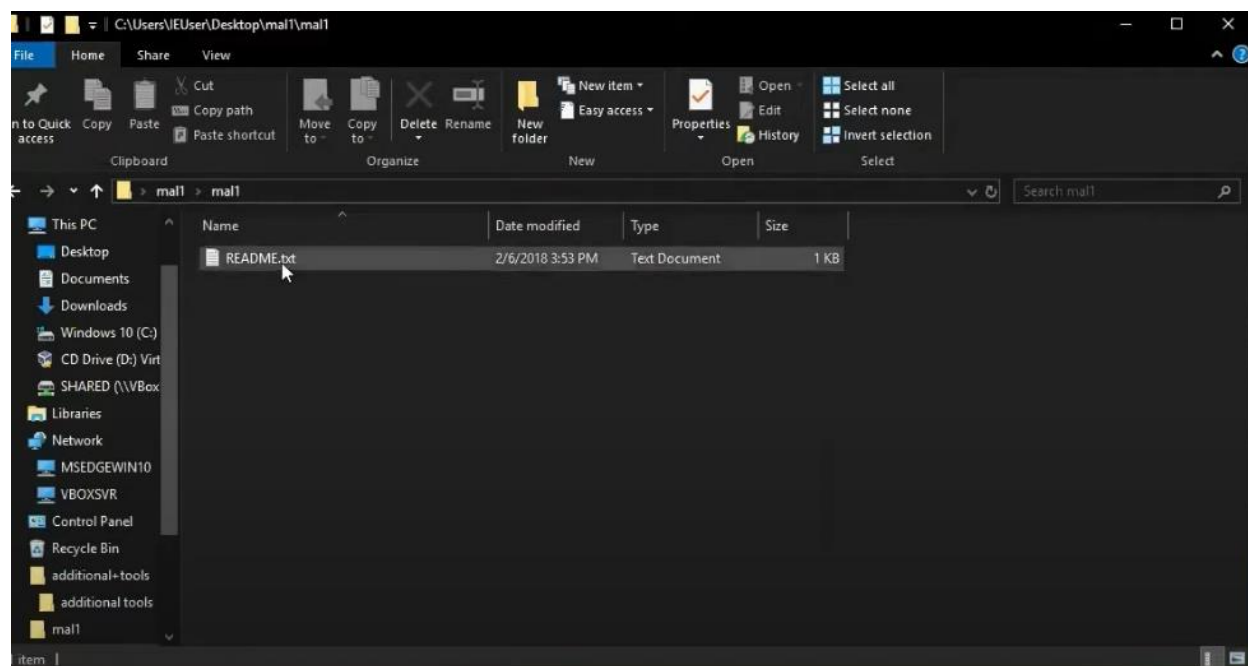
## 3. Fake net



#### 4. Regshot



We have taken the screenshot. Now we will execute the malware and after that we will take screenshot again and compare the outputs.



After executing the malware the file has been removed and now we have taken a screenshot again. Now we will compare the first and second shot to compare the out.

```
Windows10-MalwareAnalysis (FlareVM-NotInfected) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
~res-x64.txt - Notepad
File Edit Format View Help
Regshot 1.9.1 x64 Unicode (beta r321)
Comments:
Datetime: 2022-05-04 05:34:58, 2022-05-04 05:42:26
Computer: MSEDEGEWIN10, MSEDEGEWIN10
Username: IEUser, IEUser

-----
Keys deleted: 3
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\MSHist012022021520220216
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\MSHist012022021620220217
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\MSHist012022021720220218

-----
Keys added: 12
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Group Policy\ServiceInstances
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Group Policy\ServiceInstances\d94c195e-97fa-430f-b3e6-5027b29a25e0
HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Group Policy\ServiceInstances
HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Group Policy\ServiceInstances\d94c195e-97fa-430f-b3e6-5027b29a25e0
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\SessionInfo\1\ApplicationViewManagement\W32:0000000000503C2
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\SessionInfo\1\ApplicationViewManagement\W32:0000000000160324
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\SessionInfo\1\ApplicationViewManagement\W32:000000000017037E
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\SessionInfo\1\ApplicationViewManagement\W32:00000000001F045C
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\SessionInfo\1\ApplicationViewManagement\W32:0000000000200394
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\SessionInfo\1\ApplicationViewManagement\W32:0000000000240394
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\MSHist012022050320220504
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Search\JumpListData

-----
Values deleted: 24
HKLM\SYSTEM\ControlSet001\Services\Tcpip\Parameters\DhcpNameServer: "125.99.61.254 116.72.253.254"
HKLM\SYSTEM\ControlSet001\Services\Tcpip\Parameters\Interfaces\{4aa86136-917b-45d2-be98-087b589b8ca0}\DhcpNameServer: "125.99.61.254 116.72.253.254"
HKLM\SYSTEM\ControlSet001\Services\Tcpip\Parameters\Interfaces\{4aa86136-917b-45d2-be98-087b589b8ca0}\DhcpDefaultGateway: "10.0.2.2"
HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\DhcpNameServer: "125.99.61.254 116.72.253.254"
HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{4aa86136-917b-45d2-be98-087b589b8ca0}\DhcpNameServer: "125.99.61.254 116.72.253.254"
HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{4aa86136-917b-45d2-be98-087b589b8ca0}\DhcpDefaultGateway: "10.0.2.2"
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\MSHist012022021520220216\CachePrefix: ":2022021520220216"
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\MSHist012022021520220216\CachePath: "C:\Users\IEUser\AppData
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\MSHist012022021520220216\CacheRelativePath: "Microsoft\Win
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\MSHist012022021520220216\CacheOptions: 0x00000000
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\MSHist012022021520220216\CacheRepair: 0x00000000
```

if we see in the fake net the malware try to make connection with some ip



```

FakeNet-NG - fakenet.exe
05/03/22 10:37:49 PM [ RawUDPListener] 0230: 3E 77 73 64 70 3A 44 65 76 69 63 65 3C 2F 77 73 >wsdp:Device</ws
05/03/22 10:37:49 PM [ RawUDPListener] 0240: 64 3A 54 79 70 65 73 3E 3C 2F 77 73 64 3A 50 72 d:Types></wsd:Pr
05/03/22 10:37:49 PM [ RawUDPListener] 0250: 6F 62 65 3E 3C 2F 73 6F 61 70 3A 42 6F 64 79 3E obe></soap:Body>
05/03/22 10:37:49 PM [ RawUDPListener] 0260: 3C 2F 73 6F 61 70 3A 45 6E 76 65 6C 6F 70 65 3E </soap:Envelope>
05/03/22 10:37:49 PM [ RawUDPListener] 0000: 3C 3F 78 6D 6C 20 76 65 72 73 69 6F 6E 3D 22 31 <?xml version="1
05/03/22 10:37:49 PM [ RawUDPListener] 0010: 2E 30 22 20 65 6E 63 6F 64 69 6E 67 3D 22 75 74 .0" encoding="ut
05/03/22 10:37:49 PM [ RawUDPListener] 0020: 66 2D 38 22 3F 3E 3C 73 6F 61 70 3A 45 6E 76 65 f-8"><soap:Enve
05/03/22 10:37:49 PM [ RawUDPListener] 0030: 6C 6F 70 65 20 78 6D 6C 6E 73 3A 73 6F 61 70 3D lope xmlns:soap=
05/03/22 10:37:49 PM [ RawUDPListener] 0040: 22 68 74 74 70 3A 2F 2F 77 77 77 2E 77 33 2E 6F "http://www.w3.o
05/03/22 10:37:49 PM [ RawUDPListener] 0050: 72 67 2F 32 30 30 33 2F 30 35 2F 73 6F 61 70 2D rg/2003/05/soap-
05/03/22 10:37:49 PM [ RawUDPListener] 0060: 65 6E 76 65 6C 6F 70 65 22 20 78 6D 6C 6E 73 3A envelope" xmlns:
05/03/22 10:37:49 PM [ RawUDPListener] 0070: 77 73 61 3D 22 68 74 74 70 3A 2F 2F 73 63 68 65 wsa="http://sche
05/03/22 10:37:49 PM [ RawUDPListener] 0080: 6D 61 73 2E 78 6D 6C 73 6F 61 70 2E 6F 72 67 2F mas.xmlsoap.org/
05/03/22 10:37:49 PM [ RawUDPListener] 0090: 77 73 2F 32 30 30 34 2F 30 38 2F 61 64 64 72 65 ws/2004/08/addre
05/03/22 10:37:49 PM [ RawUDPListener] 00A0: 73 73 69 6E 67 22 20 78 6D 6C 6E 73 3A 77 73 64 ssing" xmlns:wsd
05/03/22 10:37:49 PM [ RawUDPListener] 00B0: 3D 22 68 74 74 70 3A 2F 2F 73 63 68 65 6D 61 73 ="http://schemas
05/03/22 10:37:49 PM [ RawUDPListener] 00C0: 2E 78 6D 6C 73 6F 61 70 2E 6F 72 67 2F 77 73 2F .xmlsoap.org/ws/
05/03/22 10:37:49 PM [ RawUDPListener] 00D0: 32 30 30 35 2F 30 34 2F 64 69 73 63 6F 76 65 72 2005/04/discover
05/03/22 10:37:49 PM [ RawUDPListener] 00E0: 79 22 20 78 6D 6C 6E 73 3A 77 73 64 70 3D 22 68 y" xmlns:wsdp="h
05/03/22 10:37:49 PM [ RawUDPListener] 00F0: 74 74 70 3A 2F 2F 73 63 68 65 6D 61 73 2E 78 6D ttp://schemas.xm
05/03/22 10:37:49 PM [ RawUDPListener] 0100: 6C 73 6F 61 70 2E 6F 72 67 2F 77 73 2F 32 30 30 soap.org/ws/200
05/03/22 10:37:49 PM [ RawUDPListener] 0110: 36 2F 30 32 2F 64 65 76 70 72 6F 66 22 3E 3C 73 6/02/devprof"><s
05/03/22 10:37:49 PM [ RawUDPListener] 0120: 6F 61 70 3A 48 65 61 64 65 72 3E 3C 77 73 61 3A oap:Header><wsa:
05/03/22 10:37:49 PM [ RawUDPListener] 0130: 54 6F 3E 75 72 6E 3A 73 63 68 65 6D 61 73 2D 78 To>urn:schemas-x
05/03/22 10:37:49 PM [ RawUDPListener] 0140: 6D 6C 73 6F 61 70 2D 6F 72 67 3A 77 73 3A 32 30 mlsoap-org:ws:20
05/03/22 10:37:49 PM [ RawUDPListener] 0150: 30 35 3A 30 34 3A 64 69 73 63 6F 76 65 72 79 3C 05:04:discovery<
05/03/22 10:37:49 PM [ RawUDPListener] 0160: 2F 77 73 61 3A 54 6F 3E 3C 77 73 61 3A 41 63 74 /wsa:To><wsa:Act
05/03/22 10:37:49 PM [ RawUDPListener] 0170: 69 6F 6E 3E 68 74 74 70 3A 2F 2F 73 63 68 65 6D ion>http://schem
05/03/22 10:37:49 PM [ RawUDPListener] 0180: 61 73 2E 78 6D 6C 73 6F 61 70 2E 6F 72 67 2F 77 as.xmlsoap.org/w
05/03/22 10:37:49 PM [ RawUDPListener] 0190: 73 2F 32 30 30 35 2F 30 34 2F 64 69 73 63 6F 76 s/2005/04/discov
05/03/22 10:37:49 PM [ RawUDPListener] 01A0: 65 72 79 2F 50 72 6F 62 65 3C 2F 77 73 61 3A 41 ery/Probe</wsa:A
05/03/22 10:37:49 PM [ RawUDPListener] 01B0: 63 74 69 6F 6E 3E 3C 77 73 61 3A 4D 65 73 73 61 ction><wsa:Messa
05/03/22 10:37:49 PM [ RawUDPListener] 01C0: 67 65 49 44 3E 75 72 6E 3A 75 75 69 64 3A 34 30 geID>urn:uuid:40
05/03/22 10:37:49 PM [ RawUDPListener] 01D0: 37 32 35 61 37 39 2D 65 38 31 36 2D 34 35 32 39 725a79-e816-4529
05/03/22 10:37:49 PM [ RawUDPListener] 01E0: 2D 61 30 65 31 2D 64 31 34 39 63 63 65 38 66 37 -a0e1-d149cce8f7
05/03/22 10:37:49 PM [ RawUDPListener] 01F0: 63 30 3C 2F 77 73 61 3A 4D 65 73 73 61 67 65 49 c0</wsa:MessageI
05/03/22 10:37:49 PM [ RawUDPListener] 0200: 44 3E 3C 2F 73 6F 61 70 3A 48 65 61 64 65 72 3E D></soap:Header>
05/03/22 10:37:49 PM [ RawUDPListener] 0210: 3C 73 6F 61 70 3A 42 6F 64 79 3E 3C 77 73 64 3A <soap:Body><wsd:
05/03/22 10:37:49 PM [ RawUDPListener] 0220: 50 72 6F 62 65 3E 3C 77 73 64 3A 54 79 70 65 73 Probe><wsd:Types
05/03/22 10:37:49 PM [ RawUDPListener] 0230: 3E 77 73 64 70 3A 44 65 76 69 63 65 3C 2F 77 73 >wsdp:Device</ws
05/03/22 10:37:49 PM [ RawUDPListener] 0240: 64 3A 54 79 70 65 73 3E 3C 2F 77 73 64 3A 50 72 d:Types></wsd:Pr
05/03/22 10:37:49 PM [ RawUDPListener] 0250: 6F 62 65 3E 3C 2F 73 6F 61 70 3A 42 6F 64 79 3E obe></soap:Body>
05/03/22 10:37:49 PM [ RawUDPListener] 0260: 3C 2F 73 6F 61 70 3A 45 6E 76 65 6C 6F 70 65 3E </soap:Envelope>
05/03/22 10:37:49 PM [ Divertor] System (4) requested UDP 169.254.255.255:137
05/03/22 10:37:50 PM [ Divertor] System (4) requested UDP 169.254.255.255:138
05/03/22 10:37:50 PM [ Divertor] System (4) requested UDP 169.254.255.255:137
05/03/22 10:37:50 PM [ Divertor] svchost.exe (5056) requested UDP 239.255.255.250:1900
05/03/22 10:37:50 PM [ Divertor] svchost.exe (6060) requested TCP 192.168.56.1:2869

```

Now we have the output after the malware execution. We analyze it using the code.

We have code to analyze it

```

import json, itertools

def Remove_PC_Name(path_name):
    New_Path_Name=[]
    if 'test' in path_name:
        r_split=path_name.split("\\")
        for strings in r_split:
            if 'test' in strings:
                strings="guest"
            New_Path_Name.append(strings)
        New_Path_Name="\\".join(New_Path_Name)
        return New_Path_Name
    return path_name

def Remove_File_name_Reg(reg):
    if 'RASMANCE' in reg:
        r_split=reg.split("\\")
        new_reg=[]
        for m in r_split:
            if 'RASMANCE' in m:
                m='RASMANCE'
            new_reg.append(m)
        new_reg="\\".join(new_reg)
        return new_reg
    if 'RASAPI32' in reg:
        r_split=reg.split("\\")
        new_reg=[]
        for m in r_split:
            if 'RASAPI32' in m:
                m='RASAPI32'
            new_reg.append(m)
        new_reg="\\".join(new_reg)
        return new_reg
    return reg

def WriteDictToCSV(dict_data, csv_header, header):
    try:
        with open('Dataset.csv', 'a') as csvfile:
            writer = csv.DictWriter(csvfile, fieldnames=csv_header)
            if int(header)==1:
                writer.writeheader()
            writer.writerow(dict_data)
    except IOError as err:
        errno, strerror = err.args
        print("I/O error({0}): {1}".format(errno, strerror))
    return

def Extract_Registry(Resource, Resources_header, data , file_list ):
    file_name=file_list
    for name in file_name:

```

After executing this we will be able to know what kind of harm malware did to our system.

## Literature Review:

- Methodologies and Tools

### Static Analysis:

- 1) **Hashing Techniques:** Used to generate unique checksums for files, aiding in identifying malware variants. SHA256 is a common hashing algorithm used for this purpose.<sup>1</sup>
- 2) **Fuzzy Hashing:** ssdeep, a fuzzy hashing algorithm, is employed to address hashbusting by creating similarity digests. This method is less sensitive to small changes in a file compared to standard cryptographic hashing.
- 3) **VirusTotal:** A widely used tool in static analysis, VirusTotal scans malware samples against multiple antivirus engines and reports findings, helping analysts leverage existing knowledge on malware variants.

### Dynamic Analysis:

- 1) **Common Tools:** Tools like IDA Freeware, IDA Pro, x64 debugger, pestudio, OllyDbg, and VMware virtual machines are commonly used for dynamic analysis.<sup>2</sup>
- 2) **Tool Selection:** The choice of tools depends on the malware's programming language and format. Analysts often choose tools that support multiple formats and focus on making the malware readable for analysis.

---

<sup>1</sup> <https://www.techtarget.com/searchsecurity/feature/Top-static-malware-analysis-techniques-for-beginners>

<sup>2</sup> <https://www.techtarget.com/searchsecurity/feature/Understanding-malware-analysis-and-its-challenges>

## ● Challenges and Limitations

- 1) **Time and Resource Constraints:** Analysts often face limitations in terms of time and resources, affecting the depth of their research. The choice of malware analysis strategy is influenced by these constraints and the specific questions that need answering.
- 2) **Mismatch of Expectations:** There is often a gap between stakeholders' expectations and the technical scope of work required for malware analysis. This can result in unrealistic time frames and workloads for analysts.
- 3) **Lack of Fundamental Computing Knowledge:** Analysts without a strong foundation in computing may struggle to select appropriate tools and fully understand malware samples.
- 4) **Difficulty in Code Analysis:** Analysts often face challenges in translating low-level assembly language to high-level logic, finding evidence in code, and explaining their findings to non-technical stakeholders.

## ● Comparative Analysis

### **Static vs. Dynamic Analysis:**

Static analysis allows for collecting data from a suspicious file without execution, providing a preliminary understanding of the malware. It includes techniques like hashing and fuzzy hashing and leverages tools like VirusTotal.

3

---

<sup>3</sup> Imam Riadi, 2015, p.2 <https://core.ac.uk/download/pdf/85136527.pdf>

Dynamic analysis, on the other hand, involves executing the malware and observing its behavior on the system, using tools like IDA Pro, x64 debugger, and VMware virtual machines.

The choice between static and dynamic analysis often depends on the specific objectives of the analysis, available resources, and the nature of the malware being investigated.<sup>4</sup>

<sup>5</sup>In summary, both static and dynamic malware analysis have distinct methodologies and tools, each with its own set of challenges and limitations. The choice of methodology often depends on the specific goals of the analysis, the nature of the malware, and the resources available. Understanding these aspects is crucial for effective malware analysis.

---

<sup>4</sup><https://core.ac.uk/download/pdf/85136527.pdf>

<sup>5</sup> Imam Riadi, 2015, p6 <https://core.ac.uk/download/pdf/85136527.pdf>