MALWARE ANALYSIS IN WINDOWS SYSTEM

Dr.T.Arumuga Maria Devi [1], Saji K.S.[2], R.Satheesh Kumar [3]

[1]Assistant Professor, CITE, Manonmaniam Sundaranar University, Tirunelveli, India.
[2] Research Scholar, CITE, Manonmaniam Sundaranar University, Tirunelveli, India.
[3] PG Scholar, CITE, Manonmaniam Sundaranar University, Tirunelveli, India.
.

***Abstract**: Today malware threats represent the greatest challenge to information security. Malware can enter the system when it is connected to the network or when external drives are used. Viruses, worms, and the like often perform malicious acts, such as deleting files, accessing personal data, or using your computer to attack other computers. Windows platform is more vulnerable to these attacks. Techniques are needed to defend Information Technology services from access or stolen by unauthorized parties. This paper focuses on detecting and removing malware using static method in Windows environment. This paper is developed in Visual studio and tested in Vmware environment. Various files with different extension are tested for malicious files. Malware such as ransomware, virus, worm and botnet are detected and removed to enhance the security of the system.*

Keywords: *Malware, Virus*

I.INTRODUCTION

Nowadays Internet is used by all sorts of people. Malware can enter the system when it is connected to the network or when external drives are connected to the system. Mischievous youths seeking a thrill or hardened cybercriminals looking to exploit billion-dollar companies, are looking to find ways to commit fraud, causing widespread damage, or to just experience the rush of breaking into a computer.

Every intrusion into a computer is not meant to cause damage or steal valuable information, but the attack might be dangerous. All intrusions into a computer exploit what is known as a vulnerability, or a weakness in the computer's operating system that can act as an access point to an attack. Viruses, worms, and the like often perform malicious acts, such as deleting files, accessing personal data, or using your computer to attack other computers.

Antivirus software is an important tool to help prevent such attacks. Antivirus software's are programs that help protect your computer against most viruses, worms, Trojan horses, and other unwanted invaders that can make your computer "sick". Every type of cyber-attack cannot be prevented with antivirus software, but it can be a tool to prevent intrusion into a computer.

Windows operating system is more vulnerable compared to another OS like Mac and Linux. Hence, there is need to propose a work that detects the malware in the windows platform.

II. OBJECTIVE AND OVERVIEW

The main objectives of this paper are as follows

To scan specific files or directories for any malware. To schedule the scans to start automatically. To initiate a scan of a specific file in a computer / CD / flash drive. To remove the malicious file and notify the user.

This paper is developed using Visual studio. It is used to detect malware files in local or external drives. This work can scan specific directory or folder. It can detect malware such as ransomware, botnet, virus and worm in various files with extension like .txt, .exe, .pdf, .mp3, .mp4. Testing is done in Vmware environment for safety measures.

III.MALWARE

Malware or malicious software is any computer software intended to harm the host operating system or to steal sensitive data from users, organizations or companies. Malware may include software that gathers user information without permission. Malware analysis is the process of determining the functionality, origin and potential impact of a given malware. Some types of malware are given below.

Virus, botnet, Ransomware, Rootkit, Worm, Backdoor, Adware, Trojan horse, Spyware

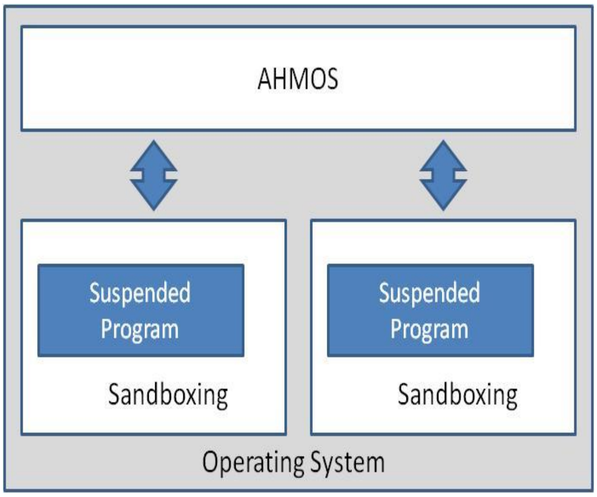


Figure 1 Existing System

Antivirus or anti-virus software (abbreviated as AV), sometimes known as anti-malware software, is computer software used to prevent, detect and remove malicious software. modern antivirus software can protect from: malicious browser helper objects (BHOs), browser hijackers, ransomware, keyloggers, backdoors, rootkits, trojan horses, worms, malicious LSPs, dialers, fraud tools, adware and spyware.

IV. MALWARE TYPES

Ransomware (Wannacry)

WannaCry is a ransomware cryptoworm, which targeted computers running in Microsoft Windows operating system by encrypting data and demanding ransom payments in the Bitcoin cryptocurrency. It is considered as a network worm because it also includes a "transport" mechanism to automatically spread itself. This transport code scans for vulnerable systems, then uses the EternalBlue exploit to gain access, and the DoublePulsartool to install and execute a copy of itself

Botnet (Zeus)

Zeus, ZeuS, or Zbot is a Trojan horse malware package that runs on versions of Microsoft Windows. While it can be used to carry out many malicious and criminal tasks, it is often used to steal banking information by man-in-the-browser, keystroke logging and form grabbing. It is also used to install the CryptoLocker ransomware. Zeus is spread mainly through downloads and phishing schemes. First identified in July 2007, when it was used to steal information from the United States Department of Transportation, it became more widespread in March 2009.

In June 2009, security company Prevx discovered that Zeus had compromised over 74,000 FTP accounts on websites of such companies as the Bank of America, NASA, Monster.com, ABC, Oracle, Play.com, Cisco, Amazon, and Business Week. Similarly to Koobface, Zeus has also been used to trick victims of tech support scams into giving the scam artists money through pop-up messages that claim the user has a virus, when in reality they might have no viruses at all. The scammers may use programs such as Command prompt or Event viewer to make the user believe that their computer is infected.

Virus (Happy99)

Happy99 (also termed Ska or I-Worm) is a computer worm for Microsoft Windows. It first appeared in mid-January 1999, spreading through email and use net. The worm installs itself and runs in the background of a victim's machine, without their knowledge.

It is generally considered the first virus to propagate by email, and has served as a template for the creation of other self- propagating viruses. Happy99 has spread on multiple continents, including North America, Europe, and Asia.

Worm (Lime)

Lime ransomware could be the ransomware that encoded system files. Commonly, ransomware uses spam emails and dangerous downloads to invade users, and one of the two was the possible cause of the infection. Ransomware is an extremely harmful piece of damaging program because it encodes data, and requests for money in exchange for getting them back. If you regularly backup your files, or if malware specialists create a free decryptor, file-recovery would not be complicated. But otherwise, there is a big chance you will lose your files.

V. MALWARE ANALYSIS

Static Malware Analysis

Static or Code Analysis is usually performed by dissecting the different resources of the binary file without executing it and studying each component. The binary file can also be disassembled (or reverse engineered) using a disassembler such as IDA. The machine code can sometimes be translated into assembly code which can be read and understood by humans the malware analyst can then make sense of the assembly instructions and have an image of what the program is supposed to perform. Some modern malware is authored using evasive techniques to defeat this type of analysis, for example by embedding syntactic code errors that will confuse disassemblers but that will still function during actual execution.

Yara Static Analysis Method

YARA is the name of a tool primarily used in malware research and detection. It provides a rule-based approach to create descriptions of malware families based on textual or binary patterns. A description is essentially a Yara rule name, where these rules consist of sets of strings and a Boolean Expression. The language used has traits of Perl compatible regular expressions.

PDF Forensic Analysis System using YARA

This paper presents an important enhanced method to detect suspicious PDF files by applying two scanning methods (structure scan and YARA scan), which depend on extracting and pointing out malicious objects that are often used for attacks. This enhanced method will be a great assistant to forensic analysts in analysing PDF files and detecting malicious content in them. Testing both scanning methods was carried out through conducting several experiments on a real dataset. The results show an improvement for detecting malicious PDF files when applying both methods. The structure scan achieved an accuracy of 99.91% and the YARA scan achieved an accuracy of 98.05%.

Dynamic Malware Analysis

Dynamic or Behavioural analysis is performed by observing the behaviour of the malware while it is actually running on a host system. This form of analysis is often performed in a sandbox environment to prevent the malware from actually infecting production systems. Many such sandboxes are virtual systems that can easily be rolled back to a clean state after the analysis is complete. The malware may also be debugged while running using a debugger such as GDB or WinDbg to watch the behaviour and effects on the host system of the malware step by step while its instructions are being processed. Modern malware can exhibit a wide variety of evasive techniques designed to defeat dynamic analysis including testing for virtual environments or active debuggers, delaying execution of malicious payloads, or requiring some form of interactive user input. Automated intelligent multinomial classification of malware species using dynamic behavioural analysis:

Malware detection has been widely explored in recent years due to an increased rate of information theft, ransom demands and money laundering cases. It is known that MS Windows Operating System family is susceptible to attacks due to the extensive number of found vulnerabilities across many outdated versions that are still in use. As a result, multiple malware categories and families have emerged. Many researchers stress the importance of multinomial malware classification rather than conventional malicious versus benign classification. However, multinomial detection has been neither sufficiently explored nor tested on recent malware samples. Authors believe that dynamic analysis can reveal information more relevant to classification characteristics in each malware category, and that cannot be done by static analysis.

In this ongoing research, Authors explore a novel application of Machine Learning classification on a multinomial malware using behavioural analysis in a controlled environment. A novel dataset containing recent malware samples was used to show a prospective application of the automated multinomial malware detection using different artifacts left in the system. Corresponding methodology was proposed to extract characteristics from the system artifacts. This work can facilitate a decision support when it is critical to narrow down the threat vectors and to find similarity in dealing with zero-day attacks.

Architecture for Automation of Malware Analysis:

Malware Analysis is the top trend in the security industry. The number of new malware samples and toolkits for automated malware generation are growing exponentially, whereas the analysis capacity and knowledge are going down.

This paper discusses the infrastructure we created for malware analysis, with network dissection of traffic, execution of samples on multiple virtual machines or in real ones if required.

The architecture performs fast analysis, comparing the results of multiple different anti-viruses and uses customized kernel-drivers, loaders and a clustered environment. New machines can be easily added to increase performance. Dispatchers, memory dumpers and dissectors are going to be discussed, as well as results got in live lab.

Malware Clearance for Secure Commitment of OS-Level Virtual Machines:

A virtual machine (VM) can be simply created upon use and disposed upon the completion of the tasks or the detection of error. The disadvantage of this approach is that if there is no malicious activity, the user has to redo all of the work in her actual workspace since there is no easy way to commit (i.e., merge) only the benign updates within the VM back to the host environment. In this work, Authors develop a VM commitment system called Secom to automatically eliminate malicious state changes when merging the contents of an OS-level VM to the host. Secom consists of three steps: grouping state changes into clusters, distinguishing between benign and malicious clusters, and committing benign clusters. Secom has three novel features.

First, instead of relying on a huge volume of log data, it leverages OS-level information flow and malware behaviour information to recognize malicious changes. As a result, the approach imposes a smaller performance overhead. Second, different from existing intrusion detection and recovery systems that detect compromised OS objects one by one, Secom classifies objects into clusters and then identifies malicious objects on a cluster by cluster basis. Third, to reduce the false-positive rate when identifying malicious clusters, it simultaneously considers two malware behaviours that are of different types and the origin of the processes that exhibit these behaviours, rather than considers a single behaviour alone as done by existing malware detection methods.

Authors have successfully implemented Secom on the feather-weight virtual machine system, a Windows- based OS-level virtualization system. Experiments show that the prototype can effectively eliminate malicious state changes while committing a VM with small performance degradation. Moreover, compared with the commercial antimalware tools, the Secom prototype has a smaller number of false negatives and thus can more thoroughly clean up malware side effects. In addition, the number of false positives of the Secom prototype is also lower than that achieved by the online behaviour-based approach of the commercial tools.

Malware Classification Using File System Footprints:

Automated analysis is useful in anti-malware research because it helps deal with large collections of samples and reduces the human effort. This paper describes an automated system that performs dynamic analysis by running new samples in a controlled environment and analysing the operations they perform on the filesystem. These operations are used to train a Support Vector Machine classifier that can proactively detect new malware samples. The experimental evaluation showed that automated system provides good results in terms of classification quality and in terms of performance. Being able to automatically decide if a file is clean or infected is very important in the antivirus industry, because based on this the file can be automatically blacklisted.

Dynamic malware detection and recording using virtual machine introspection:

Detecting and collecting malware samples are considered to be a milestone in computer security. Recording entire Virtual Machine (VM) activities requires considerable resources and it is not the wiser choice too. This approach is combination of Virtual machine introspection (VMI), file system clustering, malware activity recording. The proposed framework consists of four steps. In initial step possible executable codes are detected using clustering algorithm.

Next step monitors process and information flow for these executables using VMI. The data and information flow graph are generated for such processes. Malicious graphs among all are detected in next step. Final step comprises of recording malicious graphs and commitment of VM. Authors have implemented the prototype of this framework on leading hypervisor for Windows OS. Experimental results show that it is better way to detect and store malicious processes than storing entire VM. This is a cost-effective intelligent solution for malware recording.

Automated Malware Classification based on Network Behaviour:

Over the past decade malware, i.e., malicious software, has become a major security threat on the Internet. Today anti-virus companies receive thousands of malicious samples every day. However, the vast majority of these samples are variants of the existing malware. Due to the sheer number of malware variants it is important to accurately determine whether a sample belongs to a known malware family or exhibits a new behaviour and thus requires further analysis and separate detection signature. Despite of the importance of network activity, the existing research on malware analysis does not fully leverage the malware network behaviour for classification. This paper, proposes an automated malware classification system that focuses on network behaviour of malware samples. This approach employs behavioural profiles that summarize the network behaviour of malware samples. The proposed approach is applied to a real-world malware corpus. Experimental results show the effectiveness of the proposed approach in classifying malware samples only based on the network activity exhibited by the samples.

AHMDS: Advanced Hybrid Malware Detector System:

Malware development has become a serious activity lately. Furthermore, the purpose of malware development is getting worse as time goes by. Today, malware has been used as weapon, known as "Cyber weapon". Malware detector system is the frontline in war against malware. However, traditional malware detection systems that mainly use signature-based detection and API Call analysis are susceptible by obfuscations used by malwares. This paper presented the design, implementation, and evaluation of AHMDS:- a hybrid malware detection system that uses behavioural malware detection technique and signature-based detection technique. AHMDS uses a virtual environment which runs above the operating system, making it safe to execute and analyse malware's behaviour.

To increase performance, AHMDS also uses signature-based detection that based on known malwares. AHMDS also uses whitelist filtering mechanism to decrease false positive rate. Evaluation on Microsoft Windows AHMDS Implementation shows that AHMDS is able to detect more than 99% of malware samples or 15.89% more than current market leading antivirus. In addition, AHMDS also detected 'special designed malware that the other antivirus did not.

VI. EXISTING SYSTEM DRAWBACKS

There will be new security holes that exploit the operating system, networking software which would give the malware another entry point that bypasses the anti-virus software. Today's Anti-virus software is fairly effective only if it is updated regularly and the user takes the precautions. Anti-virus is not a firewall & it will not prevent from getting hacked, sometimes user need to install a firewall or internet security suite to completely protect the system. Anti-virus software slows down PC or network, Installing and running anti-virus software can use a lot of computer memory & hard disk space and slows down your PC.

VII. SYSTEM DESCRIPTION

This proposed work is used to scan the given specific drive. It helps the user to scan the particular location. Once malware is found this work will automatically delete the infected file in the location show in figure

This proposed system is used to scan the given specific drive. It helps the user to scan the particular location. Once malware is found this proposed system will automatically delete the infected file in the location show in Figure 2.

VIII. FUNCTIONAL DESCRIPTION

This Proposed work contains has the following functionalities.

Booting, Defining path, Validating, Scanning, Deleting, Update result

In the scanning function, the specified path/location is been scanned if the validation of the path is successful or else the scan is not pursued. If the validation process is successfully completed the scanning of the file for malware is started. Scan involves each of the file in the folder with the comparing of hexadecimal and offset value of the malware and the file specified through the driver. Scan phase involves searching of four cases of malwares such as WannaCry, Zeus botnet, LimeWorm, Happy99 virus.

In the deleting action phase, the scanned result is been passed to the action where default process should delete the malware if it is detected and display the result to the user with report of malware detected. If the scan is been completed with the detected malware it throws a message to the user as Malware detected and repairs the location by removing the malware from the machine.

After repairing the specified location, the folder has been updated, refreshed and status is changed dynamically as protected. Application is also ready for the next process scan.

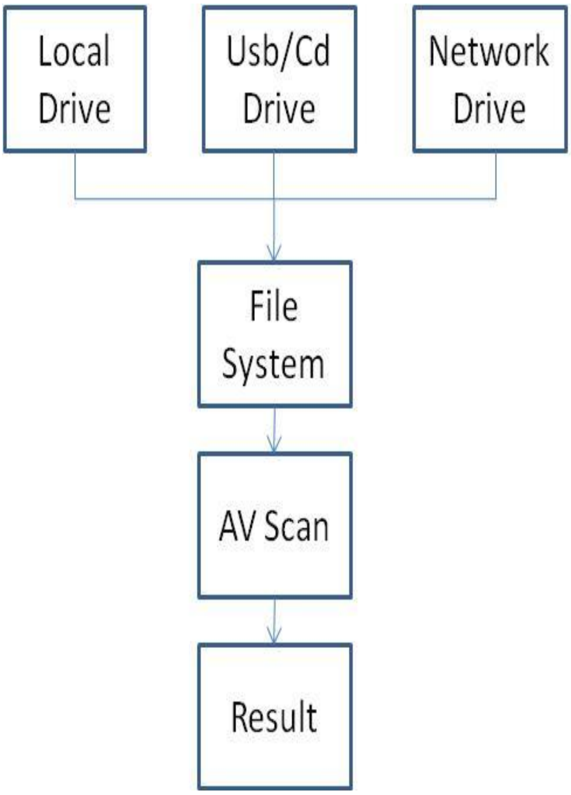


Figure 2 – New System

IX. FUNCTIONAL DESIGN

Functional Design is a paradigm used to simplify the design of hardware and software devices such as computer software and increasingly, 3D models. A functional design assures that each modular part of a device has only one responsibility and performs that responsibility with the minimum side effects on other parts. Functionally designed modules tend to have low coupling. The standard way to assure functional design is to review the description of a module. If the description includes conjunctions such as “and” or “or”, then the design has more than one responsibility, and is therefore likely to have side effects. The responsibilities need to be divided into several modules in order to achieve a functional design.

Activity Diagram

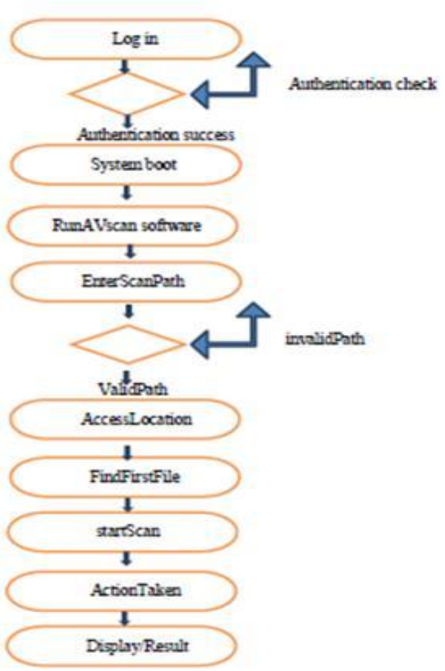


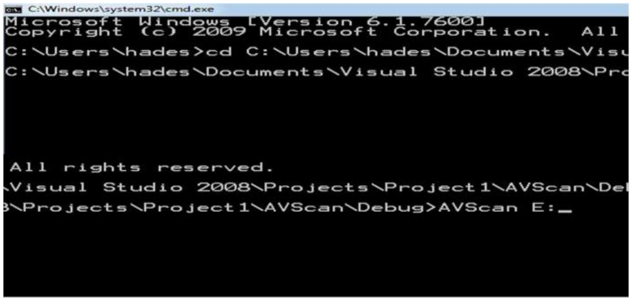
Figure 3 – Activity Diagram

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. Activity diagram is basically a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system. Activity diagram is flow with emphasis on the sequence and conditions of the flow. The actions coordinated by activity models can be initiated because other actions finish executing, because objects and data become available shown in figure

X. Implementation and Results

This proposed work is developed using visual studio and implemented in console terminal in windows operating system. This proposed work can detect all infected files in a specific location. It is tested this proposed work in different systems like desktops and laptop. Uses VMware to test the proposed work due to spreading of malware. It is tested different local drives in the system and also external storage like pen drive, cd drive and external hard disk.

Target Location


```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All
C:\Users\hades>cd C:\Users\hades\Documents\Visu
C:\Users\hades\Documents\Visual Studio 2008\Pro



 All rights reserved.
\Visual Studio 2008\Projects\Project1\AVScan\De
8\Projects\Project1\AVScan\Debug>AVScan E:_
```

Figure 4 – Specific Drive Gave by User

SCANNING

In this phase scanning process is started and it check all the files in that location and also all the directories and sub-directories from that location shown in Figure 5.

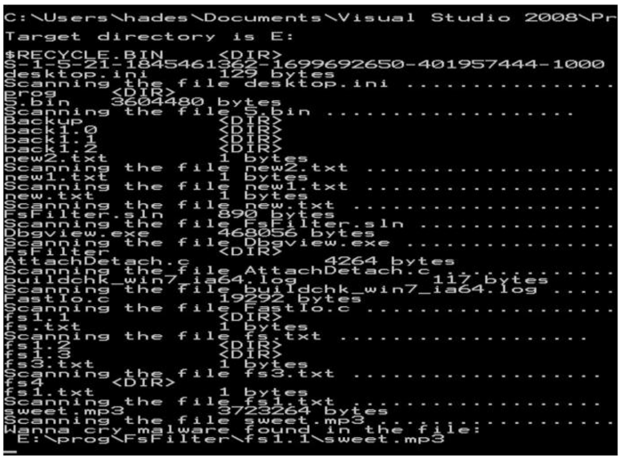

```
C:\Users\hades\Documents\Visual Studio 2008\Pr
Target directory is E:
$RECYCLE.BIN      <DIR>
S-1-5-21-1845461362-1699692650-401957444-1000
desktop.ini       129 bytes
Scanning the file desktop.ini ...............
prog      <DIR>
5.bin     3604480 bytes
Scanning the file 5.bin ....................
Backup            <DIR>
back1.0           <DIR>
back1.1           <DIR>
back1.2           <DIR>
new2.txt          1 bytes
Scanning the file new2.txt ..................
new1.txt          1 bytes
Scanning the file new1.txt ..................
new.txt           1 bytes
Scanning the file new.txt ...................
FsFilter.sln      890 bytes
Scanning the file FsFilter.sln ..............
Dbgview.exe       468056 bytes
Scanning the file Dbgview.exe ...............
FsFilter          <DIR>
AttachDetach.c            4264 bytes
Scanning the file AttachDetach.c ............
buildchk_win7_ia64.log            117 bytes
Scanning the file buildchk_win7_ia64.log ....
FastIo.c          19292 bytes
Scanning the file FastIo.c ..................
fs1.1             <DIR>
fs.txt            1 bytes
Scanning the file fs.txt ..................
fs1.2             <DIR>
fs1.3             <DIR>
fs3.txt           1 bytes
Scanning the file fs3.txt ..................
fs4       <DIR>
fs1.txt           1 bytes
Scanning the file fs1.txt ..................
sweet.mp3         3723264 bytes
Scanning the file sweet.mp3 .................
Wanna cry malware found in the file:
 E:\prog\FsFilter\fs1.1\sweet.mp3
_
```

Figure 5 – Scanning

Malware Detection

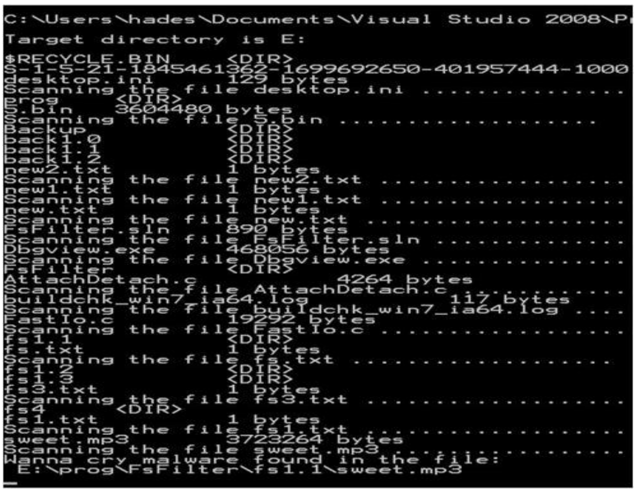

```
C:\Users\hades\Documents\Visual Studio 2008\P
Target directory is E:
$RECYCLE.BIN      <DIR>
S-1-5-21-1845461362-1699692650-401957444-1000
desktop.ini       129 bytes
Scanning the file desktop.ini ...............
prog      <DIR>
5.bin     3604480 bytes
Scanning the file 5.bin ....................
Backup            <DIR>
back1.0           <DIR>
back1.1           <DIR>
back1.2           <DIR>
new2.txt          1 bytes
Scanning the file new2.txt ..................
new1.txt          1 bytes
Scanning the file new1.txt ..................
new.txt           1 bytes
Scanning the file new.txt ...................
FsFilter.sln      890 bytes
Scanning the file FsFilter.sln ..............
Dbgview.exe       468056 bytes
Scanning the file Dbgview.exe ...............
FsFilter          <DIR>
AttachDetach.c            4264 bytes
Scanning the file AttachDetach.c ............
buildchk_win7_ia64.log            117 bytes
Scanning the file buildchk_win7_ia64.log ....
FastIo.c          19292 bytes
Scanning the file FastIo.c ..................
fs1.1             <DIR>
fs.txt            1 bytes
Scanning the file fs.txt ..................
fs1.2             <DIR>
fs1.3             <DIR>
fs3.txt           1 bytes
Scanning the file fs3.txt ..................
fs4       <DIR>
fs1.txt           1 bytes
Scanning the file fs1.txt ..................
sweet.mp3         3723264 bytes
Scanning the file sweet.mp3 .................
Wanna cry malware found in the file:
 E:\prog\FsFilter\fs1.1\sweet.mp3
_
```

Figure 6 – Malware Detection

Figure 6 shows that file sweet.mp3 is originally a malware changed by the attacker into some other format and sent it to the user.

Malware Removal

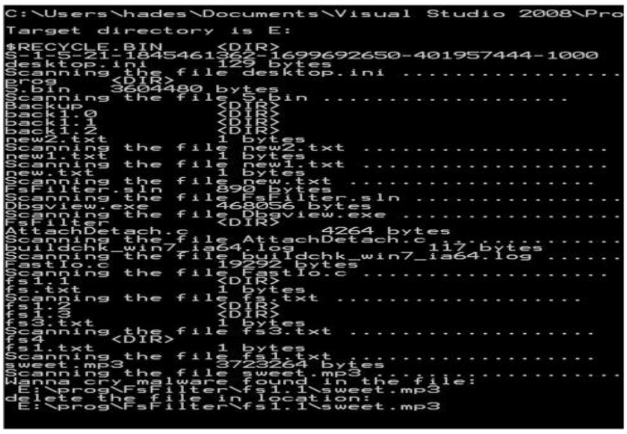

```
C:\Users\hades\Documents\Visual Studio 2008\Pro
Target directory is E:
$RECYCLE.BIN      <DIR>
S-1-5-21-1845461362-1699692650-401957444-1000
desktop.ini       129 bytes
Scanning the file desktop.ini ...............
prog      <DIR>
5.bin     3604480 bytes
Scanning the file 5.bin ....................
Backup            <DIR>
back1.0           <DIR>
back1.1           <DIR>
back1.2           <DIR>
new2.txt          1 bytes
Scanning the file new2.txt ..................
new1.txt          1 bytes
Scanning the file new1.txt ..................
new.txt           1 bytes
Scanning the file new.txt ...................
FsFilter.sln      890 bytes
Scanning the file FsFilter.sln ..............
Dbgview.exe       468056 bytes
Scanning the file Dbgview.exe ...............
FsFilter          <DIR>
AttachDetach.c            4264 bytes
Scanning the file AttachDetach.c ............
buildchk_win7_ia64.log            117 bytes
Scanning the file buildchk_win7_ia64.log ....
FastIo.c          19292 bytes
Scanning the file FastIo.c ..................
fs1.1             <DIR>
fs.txt            1 bytes
Scanning the file fs.txt ..................
fs1.2             <DIR>
fs1.3             <DIR>
fs3.txt           1 bytes
Scanning the file fs3.txt ..................
fs4       <DIR>
fs1.txt           1 bytes
Scanning the file fs1.txt ..................
sweet.mp3         3723264 bytes
Scanning the file sweet.mp3 .................
Wanna cry malware found in the file:
 E:\prog\FsFilter\fs1.1\sweet.mp3
delete the file in location:
 E:\prog\FsFilter\fs1.1\sweet.mp3
```

Figure 7 – Malware Removal

Figure 7 shows File sweet.mp3 is the wannacry malware is deleted from the system

XI. CONCLUSION

This proposed work is developed to detect the malware files that is present in the local or other external drive like pen drive, cd rom, and external hard disks of the system. It can detect and remove the malicious files like wannacry, ransomware, happy99 virus, LimeWorm, and zeus botnet. This proposed work is tested in different system with different local drives. Malicious files are detecting and removed. In c drive detected files are not removed.

XII FUTURE ENHANCEMENT

Features such as modifying and repairing the malicious files can be added to this proposed work. GUI option which helps to increase the human interaction can be included. Different detection techniques are used to find other malware like spyware, adware, backdoor and etc can be included. Automated malware analysis function can be also added.

REFERENCE

[1] Ali Hadi, Jala Atoum and Suleiman (2017) ‘PDF Forensic Analysis System using YARA’ in IJCSNS International Journal of Computer Science and Network Security, Vol.17, No.5.

[2] Andrii Shalaginov, Katrin Franke (2017) ‘Automated intelligent multinomial classification of malware species using dynamic behavioural analysis’ in Norwegian Information Security Laboratory Center for Cyber and Information Security Norwegian University of Science and Technology.

[3] Rodrigo Rubira Branco and Udi Shamir (2010) ‘Architecture for Automation of Malware Analysis’ in 5th International Conference on Malicious and Unwanted Software.

[4] Tzi-cker Chiueh, Xin Wang and Zhiyong Shan (2013) ‘Malware Clearance for Secure Commitment of OS-Level Virtual Machines’ in IEEE transactions on dependable and secure computing, Vol. 10, NO. 2.

[5] Ciprian Oprisa, George Cab˘au and Magda Buhu (2016) ‘Malware Classification Using Filesystem Footprints’in IEEE Technical University of Cluj-Napoc.

[6] Asset More and Shashikala Tapaswi (2015) ‘Dynamic malware detection and recording using virtual machine introspection’ in Indian Institute of Information Technology & Management, Gwalior

[7] Ali A. Ghorbani and Saeed Nari (2013) ‘Automated Malware Classification based on Network Behavior’ in International Conference on Computing, Networking and Communications, Communications and Information Security Symposium

[8] Garima Aggarwal, Lini Mathew, S.Chatterji, “MATLAB/Simulink Based Simulation of a Hybrid Power Flow Controller”, 2014 Fourth International Conference on Advanced Computing & Communication Technologies, 2014.

[9] J.Rajeswari, M.Jagannath, “Advances in Biomedical signal and image processing – A Systematic Review”, Elsevier, Informatics in Medicine Unlocked 8 (2017) 13-19, Journal Homepage: www.elsevier.com/locate/imu,https://doi.org/10.1016/j.imu. 2017.04.002

[10] T. Arumuga Maria Devi Aruna Jeyalakshmi, Kumar Parasuraman, 2015, “Graph Cut Based Method for Automatic Lung Segmentation for Tuberculosis by using Screening Method in Chest”, CiiT International Journal of Digital Image Processing

[11] T. Arumuga Maria Devi, 2012, “A Novel Technique in Fingerprint Identification using Relaxation labelling and Gabor Filtering”

[12] Chein-I Chang, Hyperspectral Data Processing: Algorithm Design and Analysis, March 2013, ISBN 978-0-471-69056-6. DOI:10.1002/9781118269787.

[13] Hahn, Brian H., and Daniel T.Valentine.“SIMULINK®Toolbox”, Essential Matlab for Engineers and Scientists, 2013.

[14] Dr.T.Arumuga Maria Devi, “Performance Comparison on Various Bio-Electrical Signals of MRI,CT and HIS in Human Abnormal conditions Using Hyerspectral Signal Analysis based 3D Visualization”, , International Journal for Research in Engineering Application & Management (IJREAM), ISSN: 24549150 Vol.04, Issue-04, July 2018, Page No. 698-704. DOI : 10.18231/2454-9150.2018.0567

[15] Andre Susantol and Munawar Ahmad Z.A (2016) ‘AHMDS: Advanced Hybrid Malware Detector System’ in IEEE School of Electrical Engineering and InformaticsInstitut Teknologi Bandung Bandung, Indonesia