I. J. Computer Network and Information Security, **2018, 8, 58-66**
Published Online August 2018 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2018.08.07

Design and Implementation of Malware Detection Scheme

Sonali Sharma
Department of CSE & IT Global Institute of Technology and management, Gurugram, India
E-mail: sonalisharma.261993@gmail.com

Received: 15 March 2018; Accepted: 10 July 2018; Published: 08 August 2018

Abstract—Malware is a worldwide epidemic and studies suggest that with the evolution of internet it is getting even worse. There is a new virus implemented every minute and various relevant strategies and tactics have been proposed to alleviate and eradicate cyber threats. Therefore, major concern for the researchers today is to detect and mitigate such programs. In this paper an effort has been made to propose a system which will detect some dangerous viruses and some features of the recently emerged new form of malware in cyberspace known as Ransomware. Imposing serious threats to information assets protection ransomware victimizes the internet users by hijacking user files, encrypting them, then demanding a ransom to be paid in exchange of the decryption key. The proposed program aims to scan the system to get hold of all vulnerable files present on the system and to detect the malicious one's and remove them.

Index Terms—Virus, Ransomware, Anti-virus, System scan, vulnerable file extensions, Shadow volume copies.

I. Introduction

Computer virus is a program which infects the system by replicating its own copy to other non malicious programs without the user's knowledge or consent. It is often attached by the attacker to some software or documents which the victim receives. Some remain undetected due to lack of commands that trigger them other's remain undetected because of user negligence. For example if a keylogger is installed in the victim's computer then there are many chances that the user might not know about it and the information is constantly being shared and sent to the attacker who has the control and is operating the keylogger. There are many kinds of viruses present the attackers can even encrypt the virus body with some encryption algorithm to hide it from simple view and make it more difficult to analyze and detect by the antivirus [1]. Recently Ransomware is one kind of virus which is gaining popularity rapidly. It has 44 different variants and therefore different detection strategies are required for the detection of all. In this paper we have addressed the issues of the encryption and folder lock variants of ransomware which have affected many large organizations by demanding huge amount of ransom in exchange of decryption key. Apart from these two variants other viruses which are hard to detect are addressed and their detection mechanism is proposed.

The proposed anti-virus works in three phases. The first phase is the system scan where it scans the entire system for files with vulnerable file extensions, The second phase is the detection phase where the virus is detected and the third phase is log generation where logs are generated about the scanned files and in case the virus has caused any damage to the system the user will know about it. Thousands of anti-viruses have been proposed which detect viruses but no such program is present which ensures 100% safety for the computer system. For this, the prevailing anti-viruses should be trained and their databases should be updated regularly to avoid the new malwares from damaging the systems. For its successful implementation, viruses are also designed for ransom ware which is explained in details in the Sections ahead.

II. Related Work

Ankush R Kakad et al. [1] proposed that signature based methods are easiest to implement and detect major kinds of viruses but this method fails to detect the novel attacks. In their work they have classified viruses as transient and Resident. Transient means depending on the life of host i.e. these viruses terminate when the life of the host ends, whereas resident viruses attach themselves to the memory of the system and work as a standalone application even if the program terminates.

Savan Ghaiya and kaushal Bhavsar [4] in 2013 proposed that there are two ways of malware detection- Static and dynamic. Dynamic malware detection is the best way of malware analysis since it executes the malware code in controlled environment and then analyzes it. The basic emphasis is given on the sandboxing method of malware detection. Sandbox environment makes a virtual environment in order to isolate malicious program from rest of the system for its proper analysis.

Jing Liu et. al. [6] in 2009 proposed the concept of Botnet and their formation and detection. The paper laid major emphasis on IRC and P2P based botnets. In their

work they have mentioned that some viruses remain inactive until some activity triggers them. Thus their detection becomes difficult since the code is embedded and can only be detected if it executes and shows some abnormal behavior.

Amin kharaz et. al. [7] in 2016 presented a system UNVEIL: a novel approach to detecting and analyzing ransomware. This system identifies typical behavior of ransomware such as encryption and folder lock. It correctly detected 13,637 ransomware samples from multiple families in a real-world data feed with zero false positives. The prototype of UNVEIL was implemented in windows on top of the popular open source malware analysis framework Cuckoo sandbox.

Xin Luo and Qinyu Liao [8] in 2007 proposed a framework which involves 4 steps in ransomware prevention: policy regulation, access control and management, exposure analysis and awareness and training. The approach stated that the key is to proactively detect ransomware attacks and spread awareness at the management, information technology and end-user level.

In October, 2017 Sonali Sharma and shilpa mahajan [17] published a study about various viruses and how they can damage the system by exploiting the vulnerabilities. The recent attacks of ransomwares like the "wannacry" ransomware exploited many vulnerabilities and caused devastating damage to money and property of people. The paper gives knowledge and motivation about creating a system which can keep the system safe by constantly updating and keeping awareness about such vulnerable files.

In the upcoming sections, the information about vulnerable files has been used to design a security system for the protection from already known viruses and also safeguard the files on the computer, from the new viruses exploiting the vulnerabilities.

III. Motivation

National institute of standards and technology (NIST) in 2015 presented a report on national vulnerability database (NVD). The database gives statistics on number of vulnerabilities in different operating systems [10]. These vulnerabilities can easily be targeted by the attackers and the system can be hijacked to get any sort of confidential information from the victim. Some of them have been shown in Table 1.

From the above table, it can be inferred that even the most popular operating systems are not secure and can be compromised easily. To avoid this regular check on existing as well as new viruses should be done so that the anti-virus is trained to avoid attacks on the system.

Malware continues to be a major concern and the most important security threat on the internet today. Recently a specific form of malware called ransomware has become very popular with cyber criminals. Although the concept of ransomware is not new such attacks have been registered in 1980's, but the recent success of these attacks due to growing vulnerabilities of operating systems and web browsers have resulted in increasing number of new families in past few years. For example CryptoWall 3.0 made headlines around the world as a highly profitable ransomware family causing an estimated $325M in damages. Another example of Sony ransomware attack received large media attention and the U.S government even took the official position that North Korea is behind the attack [7].

Table 1. NVD Statistics Report

S.no.	Operating System	Number of vulnerabilities
1.	Apple OS X	384
2.	Microsoft Windows server 2012	155
3.	Canonical Ubuntu Linux	152
4.	Microsoft windows 8.1	151
5.	Microsoft Windows server 2008	149
6.	Microsoft Windows 7	147
7.	Microsoft Windows 8	146
8.	Microsoft windows vista	135
9.	The linux kernel	77
10.	Microsoft windows 10	53
11.	Microsoft windows 2003	36

IV. Proposed Work

To test the functioning of the proposed framework, the entire implementation process has been divided broadly into number of phases.

The broad categorization of the entire implementation of the scheme on which the system has been designed is as follows:

A. Creation of Virus

Viruses are designed to damage or to infiltrate a computer system without the user consent. Here, Visual studio 2013 framework is used to create the viruses mentioned in Table1. These include ransomwares as well. As discussed earlier there are 44 behaviors of ransomware and many anti viruses can still not detect all of them. Folder lock and encryption of files are the two features of ransomware implemented in this paper. This is implemented so as to test the functioning of the proposed antivirus.

B. System Scan

The proposed method of system scan can effectively detect vulnerable files present in the system which can have viruses present in them on the basis of the vulnerable file extensions as listed in Table 2. A log file will be generated and maintained which can be further used to train the system about new malwares and viruses.

C. Detection

Every minute a new virus is generated over the internet to infect many systems, but many remain undetected due to either by negligence of the user or remain deactivated due to lack of command which triggers them.

The flow diagram of the entire implemented work has

been explained in Fig.1.

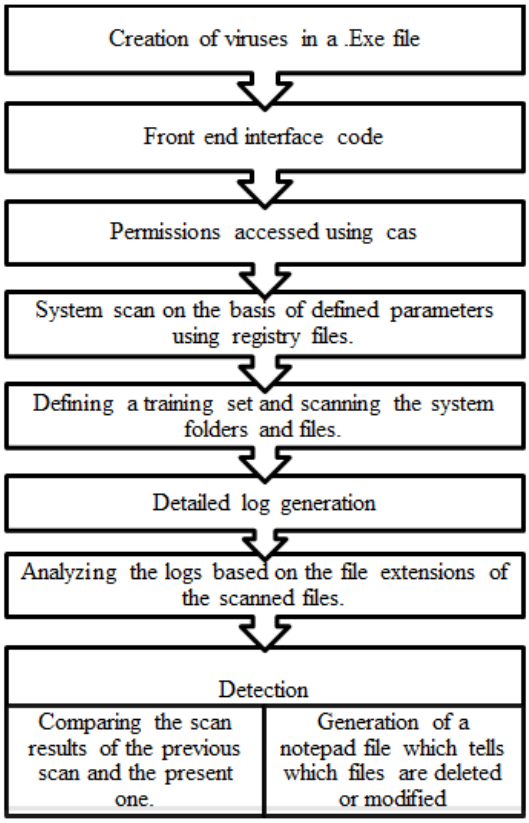

Fig.1. Implementation Procedure

This flow diagram gives an abstract view of the overall system that has been designed. The sections ahead contain detailed implementation regarding the same.

Table 2. Viruses and their Description

TYPE	EXPLAINATION
Too many DNS requests	Comes under Network behavior of ransomware. It prevents the detection of command and control server.
Encryption of files	As the virus runs some of the files get encrypted. The decryption key however, will be provided to the victim only when the ransom is paid.
Folder lock	The folders in the system can be locked by setting some password and the victim will not know the password so he will not be able to access that folder.
Delete directory	Comes under behavioral ransom ware. Deletes the directory or folders when it runs on the system.
WIFI and sound disabled	WIFI and Sound of the computer system gets disabled.
Services disabled	The services which are running currently on the system get disabled and the ones which are disabled get enabled.

V. Creation of Viruses

Computer Virus is a program which can replicate its own copy to other non-malicious programs by modifying them or by deleting them without the user's permission. It spreads at a geometric rate eventually infecting the entire system and affecting the other connected systems as well [1].

There are many kinds of viruses, which can be categorized as resident or transient i.e. memory based and host based. There is a new virus found in every single minute. The main motive is on implementing and working with the viruses which are not so common and cannot be easily detected but are gaining popularity slowly. The following viruses are implemented in .NET language using Visual studio 2013 framework.

A. Ransomware

Ransomware is a type of malware which locks the victims desktop to render the system inaccessible to the user by encrypting, deleting or overwriting the user's files unless a ransom is paid [11]. More advanced ransomware may also encrypt the Master file table (MFT) or the entire hard drive. Therefore, a ransomware can be considered to be a denial-of-access attack that prevents the victim from accessing the system or files. There are basically 44 behaviors that have been shown by the ransomware and its variants which are classified under 4 categories:-

- Network
- System
- Static
- Behavioral

The 3 variants of ransomware implemented are as follows:

Too many DNS requests

This virus will open number of tabs, of the URL specified in the browser window. A timer can also be used to set after which each tab will open. The overall performance of the system will degrade as CPU will be busy and was unable to perform some useful activities. Moreover, it is not possible to detect the command which has triggered the virus to run. The pseudo code is defined below and its possible impact is also discussed.

Initialize component;
Set timer =1sec;
Process.start (http://google.com);
End

Outcomes:

- If this code runs then many google.com tabs will appear on the browser window.
- The CPU utilization is increased to 90-95%
- Due to which the system does not respond.

Folder lock

Folder lock is implemented by the fact that the extension of the file is changed and is locked with a password which cannot be known to the victim unless a

ransom is paid to the attacker. This code will first check if there is no password set on the file or folder. Then it will select the path of the file or folders for which it need to set the password, and then lock that folder. This folder will unlock only when the correct password is provided otherwise it will display an error message. The pseudo code is defined below.

Set password:

```
If password status =False;
  Initialize component();
    {
If (radiobutton.checked());
DirectoryInfo d= New
            DirectoryInfo(folderBrowserDilog.S
            electedPath);
String SelectedPath= d.Parent.FullName+
            d.name;
if (checkBox.Checked)

            setpassword(folderBrowserDialog.S
            electedPath);
}
Else
{
Status = GetStatus(status);
}
```

Check Password:

```
If (Text.equals(Pass))
{
Status =True;
This.close()
}
Else
MessageBox.show("incorrectPassword",
"error".);

End.
```

Outcomes:

- This code locks files and folders and make them inaccessible.
- Password is only provided to the victim when the ransom is paid.

File Encryption

In this using AES encryption the files are encrypted and stored as a different file name which cannot be opened. When this virus runs the attacker will encrypt the file and rename it so the victim will not even know the location of the encrypted file nor the name by which that file has been saved on his computer. AES encryption in Visual Studio using .NET framework is implemented using "RijndaelManaged" which is a built in function. The different destination save is implemented using the ReadByte and WriteByte operations.

```
Initialize Component();
EncryptFile
{
String Password = @"Mykey123";
String CryptFile= Outputfile;

        FileStream fsCrypt = new
FileStream(cryptFile, FileMode.Create);
        RijndaelManaged RMCrypto = new
RijndaelManaged();

        CryptoStream cs = new
CryptoStream(fsCrypt,
RMCrypto.CreateEncryptor(key, key),
            CryptoStreamMode.Write);
data = fileStream.ReadByte()) != -1)
            cs.WriteByte((byte)data);

End
```

Impact:

- The files get encrypted.
- The location and file name of the encrypted files remains hidden from the victim.
- Decryption is key is provided to the victim only after he agrees on attackers terms and conditions.

B. Services disabled

This is a harmful virus as this virus code will first get and list all the services running on the system and then it will stop the services. The code is discussed and possible impact is also referred below.

```
Initialize component;
GetAllServices();
For each (ServiceController service in
ServiceController.GetServices())
{
Service.Stop();
Service.WaitForStatus
(ServiceControllerStatus.Stopped, Timeout);
{
Service.Start();
Service.WaitForStatus
(ServiceControllerStatus.Running, Timeout);
}}

End
```

Outcomes:

- This will disable many services like Antivirus and firewall.
- The detection of this virus is difficult.
- Many other services stop running that leads to abnormal functioning of the system.

C. Delete File or Directory

This code can delete entire directory or the folders which cannot be retrieved afterwards. The victim will not know about the command which deleted the files or folders. The pseudo code with its possible impact is discussed below.

```
Initialize component;
For each (String file in Directory.Getfiles(strpath))
{
File.Delete(file);
}
Foreach (Subfolder in
Directory.GetDirectories(strpath))
{
RemoveDirectories(subfolder);
}
Directory.Delete(strpath);
}
String path = "E:\\New folder (2)";
RemoveDirectories(path);
MessageBox.Show("deleted");
}
End
```

Outcome:

The files or folders from the directory will be deleted and the victim will not know the cause for this deletion of files or folders neither will he be able to recover the deleted files.

D. WIFI Disabled

This code first lists the entire Wi-Fi networks available in the LAN and then disables the one with which the system is connected to.

```
Initialize Component();
PctNetwork Adapter. Image =
(NetworkAdapter.NetEnabled > 0)
? Resources.ImgEnabledNetworkAdapter
: Resources.ImgDisabledNetworkAdapter;
btnEnableDisable.Text = (networkAdapter.
NetEnabled > 0)
? Resources. BtnText_Disable: Resources.
BtnText_Enable;
End
```

Impact:

This code in running mode will disable the internet connectivity of the device by deactivating the network adapter.

E. Sound Disabled

A Sound adapter in the computer system has an associated code with it. Thus, in order to trigger them, that specific value is called by the function. This virus disables the sound by calling the code for the mute operation. The code is defined as

```
AppCommand_Volume_up = 0x80000
Appcommand_Volume_down =0xA0000
AppCommand_Volume_Mute =0x90000
Initialize Component();
{
SendMessage (this.handle,
Appcommand_volume_Mute);
}
End
```

Impact:

When this code runs, it will disable the sound driver of the system.

These viruses are created to test the functionalities and capabilities of the antivirus designed in the subsequent sections. These viruses can be triggered once infected files are open. These file can either be sent by an attacker over the internet to the victim or can be attached in the system itself.

Table 3. Vulnerable Extensions of Program Files

.EXE	An executable program file.
.PIF	Program information file for MS-DOS programs. These files do not contain code but when they do windows treats them as .EXE files.
.APPLICATION	Application Installer File
.GADGET	A gadget files for windows desktop gadget technology.
.MSI	A Microsoft installer file
.MSP	Windows installer patch file.
.COM	Original type of program used by MS-DOS
.SCR	Windows screen saver file.
.HTA	Html application
.CPL	Control panel file
.MSC	Microsoft management console file
.JAR	Contain executable java code.
Script Files	
.BAT	Batch file. Contains list of commands that run on the computer when you open it.
.VS,.VBS	A VBScript file
.VBE	An encrypted VBScript file.
.JS	Java script file
.JSE	Encrypted Java script file
.JSE	Encrypted Java script file
.WS, .WSF	Windows script file
.WSC,.WSH	Windows Script Component and Windows Script Host control files.
.PS1, .PS1XM	Windows power shell script
Shortcuts	
.LNK	A link to program on your computer
.INF	Text file used to auto run
.SCF	Windows explorer command file
Others	
.REG	Windows registry file
.DOC,.XLS,.PPT	Microsoft word, excel and power point documents. Can contain malicious macro code.
.DOCM, .DOTM, .LSM, .XLT	New file extensions added in office 2007. M in the end indicates that the document contains macros.

VI. System Scan

A system detection framework is developed to detect viruses designed in previous phase. This phase is further divided into 3 steps.

A. Identification of Vulnerable Files

First step is to scan the system and search for the prevailing vulnerable files and documents based on the extensions. All the phishing attacks and spyware travel through the network in form of legitimate file extensions embedded with viruses. In order to nullify this effect it is essential to track these files and remove them [12]. These files have to be handled with extra care so that the system remains secure. Table 3 shows different file extensions that are vulnerable in nature.

B. Implementation of System Scanning

The framework of system scan has been designed in. NET framework and the various parameters taken into consideration have been discussed in Table 4.

Table 4. System Scanning Parameters

Scan type	Description
Control scan	Time, date and battery status etc.
User scan	USB scan
System software	Notepad and other default software.
System fonts	System installed and downloaded fonts scan.
System help files	Help registration files
Shared libraries	Shared .dll files
Startup entries	Files that run during startup.
Installation strings	Installed file paths, or other .exe downloaded files.
Virtual devices	Connected VMware, mobile devices.
History and start menu	Start menu and system history.
Deep system scan	All the leftover files in the registry.
MRU lists	Services are scanned.

CAS (Code Access Security)

To gain access of the files in the system it is required to gain permissions using this functionality in .NET framework. CAS is a mechanism that limits the access of code to protect resources and operations. It specifies permissions which the code has and the one's that the code will never have [13].

The implemented code uses this function for the required results:

<assembly: PermissionSetAtrribute(SecurityAction.Requestminimum, Name:= "full trust")>.

The detailed implementation is as follows:

Begin scan: By clicking on the start button the front end interface will load and the scanning will begin based on the parameters listed in Table 4.

Fig.2. Scanning Initializes

System Scan in Process

Parameters listed in Table 4 form the basis of scan. Each of the listed parameter consists of a registry path which leads to the set of files it accesses and scans.

Registry - Hierarchical database maintained centrally by Microsoft windows to store system configuration information of the hardware devices as well as of the applications. Basically all the files present in the system have their instances in the registry.

Registry has 5 main folders or root keys. These may further contain sub keys. The root keys are [2]:

- HKEY_CLASSES_ROOT (HKCR)
- HKEY_CURRENT_USER (HKU)
- HKEY_LOCAL_MACHINE (HKLM)
- HKEY_USERS (HKU)
- HKEY_CURRENT_CONFIG (HKCC)

The system scan once completed will generate logs of the files scanned as shown in Figure 4.

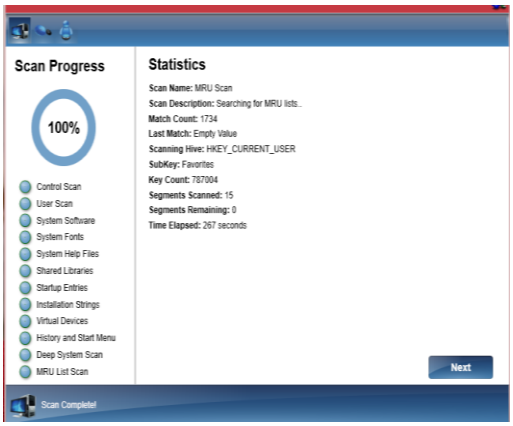

Fig.3. System Scanning Complete

Log generation

A detailed log file is generated which shows the number of files scanned as shown in Fig. 4 those vulnerable files can be seen here.

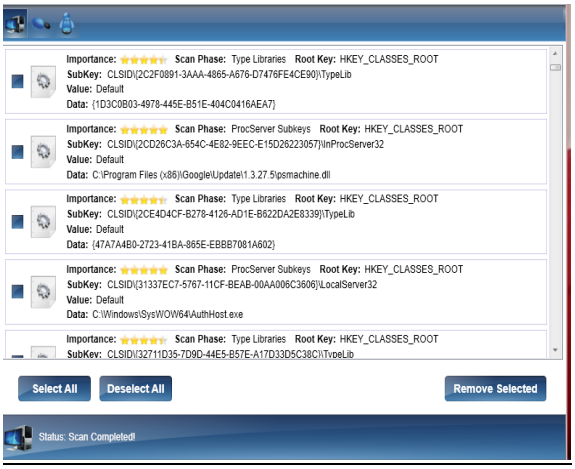

Fig.4. Detailed Log Generation

VII. Detection

The last phase is the phase where the actual anti-virus works and detects the malicious activity. The antivirus program is a software utility designed to protect the computer or network against computer viruses. It is a program or set of programs that are designed to prevent, search, detect and remove other programs which show any abnormal behavior or malicious activity [14]. Usually different viruses have different detection mechanisms therefore only one technique cannot eradicate the malware present in the system [3].

- Size: some viruses append their malicious code at the end of the file. An antivirus software can compare it's before and after sizes and if there is no modification then it can be concluded that the file is free of virus.
- Pattern matching: every virus has a unique feature that they use to infect the files or computer systems. This signature could be some lines in assembly language that overwrite the stack pointer and then jump to new line of code. An antivirus compares the information with a virus database; if the information matches any of the virus signatures then an alert is raised.
- Heuristic: this approach is based on behavioral analysis and the antivirus software which works on this mechanism raise alarm when any of the activity exhibited by the file under observation behaves in some abnormal manner.

The proposed anti-virus is implemented in Visual studio 2013 framework. For detection purpose our approach uses 2 mechanisms:

A. Machine Learning based on Behavior Analysis:

To detect the malware different training sets and samples from malware repositories and security forums like malwr.com and virusshare.com have been used to train this proposed system which is build on visual studio 2013[15]. This will help the antivirus to detect all the already known viruses and with periodic updating it also helps to detect the new viruses.

B. Using Dynamic Approach:

Now, since we have trained the anti-virus software of all the known vulnerabilities and attacks the virus embedded in the file will not execute unless the anti-virus is deactivated.

So as to study and analyze the working and function of the viruses and the implemented system we deactivate the system antivirus and then execute the malware file for some time and then compare the results with the initial system state. This comparison will help in generating results about the changes the malware has caused to the system. Fig. 4 shows the log file generated which specifies the files which the malware has deleted or encrypted.

- In this case the malware sample is executed for a certain period of time and after this the modifications made by the virus are checked and its behavior is studied.
- Similarly, the .EXE file which has the virus embedded in it is executed and then the modifications made by the malware on the system are studied by comparing them with the initial state of the system. The fig. 5 is the notepad log file which shows the following results:

1. Time of last scan.
2. Date of last system scan
3. The files which are deleted due to the virus execution
4. Encrypted files on the system.

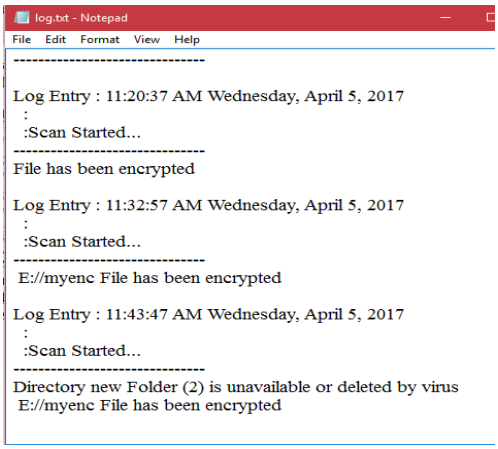

Fig.5. Notepad Log File

- The deleted files are detected by comparing the

last scan state of the system and the current scan status.

- The encrypted files are detected as they cannot be opened and therefore do not function in a normal way.
- The function which helps to read the files in the volume targeted is: System.Text.RegularExpressions.Regex.Is atch(line, @"^[a-zA-Z0-9]+$")).

By this each file will be checked and the files which are encrypted or deleted by the virus can be seen in the logs.

Pseudo code for detection process:

```
Initialize Component()
{
Streamwriter writestream= File.appendText(File path);
WriteStream.write("\r\n log entry: ");
WriteStream.writeline ("{0} {1}",
DateTime.Now.ToLongTimeString();
DateTime.Now.ToLongDateString());
WriteStream.writeline ("  :");
WriteStream.Writeline ("  :{0}", msg);
If (Directory.exists ("E:\\ new folder (2)")
{
}
Else
WriteStream.Writeline(""Directory new Folder (2) is unavailable or deleted by virus");
        }
        StreamReader sr = new StreamReader("E:\\myenc");
        String line = sr.ReadToEnd();
        sr.Close();

if(System.Text.RegularExpressions.Regex.IsMatch(line, @"^[a-zA-Z0-9]+$"))
        {
                }
        else
        {
           writeStream.WriteLine(" E://myenc File has been encrypted");
        }
        writeStream.Flush();
        writeStream.Close();
```

VIII. Advantages of The Implemented Work

Firstly, the anti-viruses available do not tell the modifications made by the malware to the system, But by using the dynamic approach: Comparing the state of system after execution of virus with the initial state of system we can infer many valuable results to study the nature of malware.

Secondly, there are many variants of ransomware but the available anti-viruses are not capable or trained enough to detect them. So in this paper we have implemented some unknown variants of ransomware, studied their functioning and detection mechanism.

The brief step-wise execution of the detection mechanism has been showed in Fig. 6.

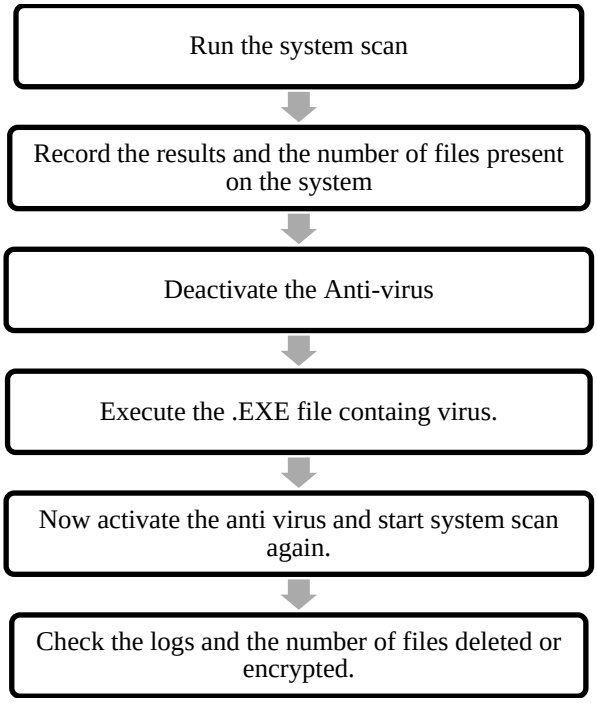

Fig.6. Detection Mechanism

IX. Future Work and Conclusion

Anti-virus software is critical for users to install and should be up-to date to protect the computer system from the various new viruses in execution over the internet. Our major concern was to detect the damages caused by the ransomware as well as other viruses and to stop them from execution since the victim does not get to know if his system gets affected by malware, this is necessary as it will inform the victim about the encrypted files on his system.

This study can further be explored and other new emerging variants of ransomware can also be detected to ensure security. Shadow volume copies are the target of many ransomware creators and their constant focus is on deleting the backups so that the infected files and folders cannot be recovered. So the cyber experts today also need to secure the backups to make it difficult for the virus to act.

References

[1] Ankush R Kakad, Siddharth G Kamble, Shrinivas S Bhuvad and Vinayak N Malavade, "*Study and Comparison of Virus Detection Techniques*", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 3, March 2014.

[2] Khawla Abdulla Alghafli et. al. *,"Forensic analysis of windows 7 registry",* Edith Cowan University Research Online, Australian Digital ForensicsConference,2010.

[3] Sarika chaudhary et. al.," *How Anti-virus Software Works??",* International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 4, April 2013.

[4] Savan Gadhiya and Kaushal Bhavsar, "*Techniques for Malware Analysis* " ,International Journal of Advanced Research in Computer Science and Software Engineering ,Volume 3, Issue 4,India, April 2013.
[5] Sandeep kumar et al., "*Malicious Data Classification Using Structural Information and Behavioral Specifications in Executables*", Proceedings of 2014 RAECS UIET Punjab University Chandigarh, 06 – 08March, 2014.
[6] Jing Liu, Yang Xiao, Kaveh Ghaboosi, Hongmei Deng and Jingyuan Zhang "Botnet: Classification, attacks, Detection, tracing, and preventive measures." Hindawi Publishing Corporation EURASIP Journal on Wireless Communications and Networking, Volume 2009.
[7] Amin kharaz, sajjad Arshad, Collin Muliner, William Robertson and Egin Kirda, "*UNVEIL: A large-scale automated approach to detecting Ransomware*.", USENIX security symposium, Northeastern university, August 2016.
[8] Xin luo and Qinyu Liao, "*Awareness education as the key to ransomware prevention*", Information systems security, USA, 2007.
[9] Tulika Mithal, Kshitij Shah and Dushyant Kumar Singh, "Case Studies on Intelligent Approaches for Static Malware Analysis" ,Emerging Research in Computing, Information, Communication and Applications ,10 may 2016.
[10] http://www.gfi.com/blog/2015s-mvps-the-most-vulnerable-players/
[11] http://research.omicsgroup.org/index.php/Ransomware
[12] http://www.howtogeek.com/137270/50-file-extensions-that-are-potentially-dangerous-on-windows/
[13] https://msdn.microsoft.com/en-us/library/930b76w0(v=vs.90).aspx
[14] https://www.webroot.com/in/en/home/resources/tips/pc-security/security-what-is-anti-virus-software
[15] http://marcoramilli.blogspot.in/2016/12/malware-training-sets-machine-learning.html
[16] https://www.bleepingcomputer.com/tutorials/how-to-recover-files-and-folders-using-shadow-volume-copies/
[17] Sonali Sharma and Shilpa mahajan, *"Design and implementation of security scheme for detecting system vulnerabilities"*, International journal of computer network and information security, Vol.9, October, 2017.

Author's Profile

Sonali Sharma received her Bachelor of engineering in Computer Science from Amity University, Gurgaon, India in 2015. She received her Masters of engineering in Computer Science with specialization in Cyber security from The NorthCap University, Gurgaon, India in 2017. Her research areas include Design and implementation of malware detection using static and dynamic analysis, Creation of antivirus which can detect and remove ransomware, cyber security, network and system security, information security and intrusion detection. Currently, she is working as an Assistant professor of Computer Science and Information Technology Department in Global Institute of Technology and management, Gurugram, Haryana, India.

How to cite this paper: Sonali Sharma,"Design and Implementation of Malware Detection Scheme", International Journal of Computer Network and Information Security(IJCNIS), Vol.10, No.8, pp.58-66, 2018.DOI: 10.5815/ijcnis.2018.08.07