I. J. Computer Network and Information Security, **2017, 10, 24-32**
Published Online October 2017 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2017.10.03

Design and Implementation of a Security Scheme for Detecting System Vulnerabilities

Sonali Sharma
Department of CSE & IT
The NorthCap University, Gurugram, India
E-mail: sonalisharma.261993@gmail.com

Shilpa Mahajan
Department of CSE & IT
The NorthCap University, Gurugram, India
E-mail: shilpa@ncuindia.edu

Received: 26 May 2017; Accepted: 01 August 2017; Published: 08 October 2017

***Abstract*—**With evolution of internet, security becomes a major concern. Number of malicious programs called malware, travels through network into systems. They have many advanced properties like self-hiding, self-healing and stealth mode execution, which are hard to detect. Therefore, the major challenge for researchers today is to detect and mitigate such programs. Since there is a new virus implemented every minute no detection mechanism can be designed which gives 100% protection but by keeping the anti-virus database up to date we can escape many attacks. In this paper, an effort has been made to explain the design of a system program which can scan the vulnerable files on the system, generate logs and this can later be used to design antivirus software and stop virus execution. This program aims to scan system files and target the files which are vulnerable present on the system based on their file extensions. It generates logs after the system scan is complete which can be studied and used for anti-virus creation.

***Index Terms*—**File extensions, Code access Security, Registry files, Anti-virus, System Scan.

I. Introduction

Virus is a program that changes behavior of the program by modifying existing user's program [1]. These viruses can enter in to a system, when a user unknowingly installs malicious programs, clicks on links in mails send by untrusted accounts, connects infected USB to the system, connect to another systems through blue tooth, downloads files containing malware etc. These programs are executable files which get downloaded and installed on the system unknowingly and can cause serious damage to user's information and reputation.

It is an undetectable feature of parallel attacks in which attackers/hacker compromise a computer system or a network for their unethical actions. Once a group of computers of any organization at different locations is compromised and controlled by attacker. It is difficult to track and find the actual cause of attack due to complexity of internet. Thus malicious activities have become serious threats nowadays [6]. There are many tools to detect such attacks. These are well known framework that detect security threats, collect malware data, uses signature based detection and monitors network traffic on the basis of a defined database. However, the presence of all such methods are not enough to detect novel attacks as attackers constantly work to exploit security breaches and are also well equipped to carry out any hazardous attacks on a computer system or organization.

As detecting such attacks in a network is a major concern. Number of research opportunities in designing novel softwares like antivirus to detect these unknown attacks is there. It is important to be cautious of malicious links or files which contain viruses. Therefore, there is a need to aware which files are safe to open and which links can be trusted and clicked on.

In this paper, a program is designed that will not only scan system for vulnerable files but will also generate logs so that in future the anti-virus designing becomes easy and the efficiency increases. The log files can later be used for analyzing and detecting harmful files and this also speeds up the process since the designers can focus only on the vulnerable files. It can also detect vulnerable files downloaded from internet. For example, IRC BOT attack is basically an executable file activated by a specific command from IRC server [6].For complete system coverage and so that no file is left unattended the concepts of CAS and registry are used which give better efficiency and detect all the vulnerable files residing on the system. Various parameters are defined on the basis of which the scanning of system files occurs and two types of log files are generated.

Top companies like Microsoft, Apple and adobe have vulnerabilities. Apple is known for its strongest encryption and Windows is used in almost every

computer system. Hence, a good number of populations can be targeted by exploiting these vulnerabilities [10]. Therefore, the cyber experts have to work at a faster pace than the attackers so as to keep everyone secure of cyber attacks.

II. Related Work

Ankush R Kakad et al. [1] proposed that signature based methods are easiest to detect viruses but this method fails to detect all type of viruses (novel attacks). Three types of detection methods i.e. Signature based, Anomaly based and Code emulation along with their advantages and disadvantages have been discussed. The signature based methods contain a database having some specific signatures and are accurate, can detect the simple viruses and are easy to implement but they do not detect the novel viruses. Apart from this as signature based anti-virus has a database of some defined signatures it also occupies more storage space. In their work, they have classified viruses as transient and resident. Transient means depending on the life of host that is these viruses terminates when the life of host ends whereas Resident attaches itself to the memory, work actively as a standalone program even when program terminates.

Savan Ghaiya and kaushal Bhavsar [4], in 2013 proposed that there are two ways of malware detection-Static and dynamic. Malware or malicious software can be of many types for example- Worm, virus, Trojan horse etc. and there are different detection mechanisms for each one of them. Static malware analysis is analyzing the inspected file without executing it and only examining the source code and this process is usually done manually. The different techniques discussed for static detection are file fingerprinting, extraction of hard coded strings, file format, AV scanning, packer detection and disassembly. This technique is easy to implement and detects most of the viruses except the new ones. Other disadvantage of this technique is that sometimes the source code of the malware sample is not readily available and thus the inspection cannot be done properly. Dynamic malware detection is the best way of malware analysis since it executes the malware code in controlled environment and then analyzes it. It further has two approaches the first one emphasizes on comparison of the current system state with the state of system after execution of malware, and the other approach is analyzing the runtime behavior of malware using specialized tools. Many tools have been discussed in this consent for the detection of malware but the basic emphasis is given on the sandboxing method. Sandbox environment makes a virtual environment in order to isolate malicious program from rest of the system for its proper analysis. This mechanism is used widely to implement this technique.

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. For IRC based botnets, the thorny problem is that the source code cannot be obtained of most of the bots. Hence, in-depth analysis at networking level and system level for bots' behaviors are hardly carried out. For P2P-based botnets, the following practical challenges should be further considered: (1) maintaining the rest of bots after some have been taken down by defenders; (2) hiding the botnet topology while some bots are captured by defenders; (3) managing the botnet more easily; (4) changing the traffic patterns more often and making it harder for detection. 10 EURASIP Journal on Wireless Communications and Networking Detecting and tracking compromised hosts in a botnet will continue to be a challenging task. Traffic fingerprinting is useful for identifying botnets.

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. The key insight of the analysis is that in order to mount a successful attack, ransomware must tamper with a user's files or desktop. UNVEIL automatically generates an artificial user environment, and detects when ransomware interacts with user data. In parallel, the approach tracks changes to the system's desktop that indicate ransomware-like behavior.

III. Virus And Anti-Virus Functioning

Before developing a program for detection of viruses, it become important to know exactly how a virus spreads in a device. The working of virus is explained through a pseudo code as defined below.

Pseudo code showing how a virus works:

BEGIN V1

if spread-condition **then**

for some set of target files **do**

if target is not infected **then** determine where to place virus instructions

Copy instructions from **BEGIN V1** to **END V1** into target

Alter target to execute added instructions

END if

END for

END if

Perform some action(s)

GOTO beginning of infected program

END V1

A. Antivirus Functioning

It is important to understand how an antivirus works in a device [3]. This knowledge is used for creating antivirus program. The various steps are described in Fig.1.

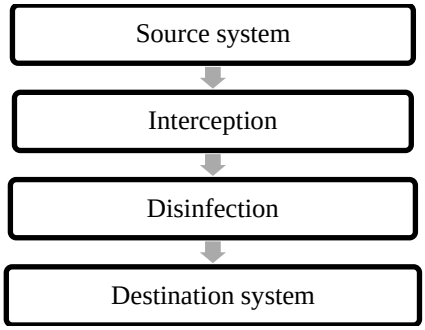

Fig.1. Virus Detection Framework

Source system: A source can be any device like floppy disk or hard disk.

Interception: Mechanism of interception of the information from the source and then scanning it for any virus variant. This interpretation mechanism is specific and different for each operating system and depends on the component in which anti-virus program is executed. For example, a virtual driver is used in windows 8 which monitor the activity of disks. Therefore first the information is accessed through floppy disk or hard disk and then antivirus program will intercept the read and write call to the disk. Scanning of the system is done afterwards.

Disinfection: In this step the actual deletion of malware prone or affected files is done. An alert message is raised for the files not disinfected or can still harm the system so that the user does not execute such files.

Destination: A destination can be a hard disk or an ISP where a client can send the message when required. Alerts are generated after the disinfection is done and forwarded to the destination system.

Broadly there are two techniques for malware detection [6].

1. **Static malware analysis**: this malware analysis analyzes the malware without executing the program. In this approach, the source code of malware is not available. Thus it is difficult to analyze the malware by static method.
2. **Dynamic malware analysis**: In dynamic malware analysis, the infected code is executed and monitored. Its actions are recorded that can be used for analyzing its malicious behavior But this method suffers from incomplete code coverage as only one execution path is monitored.

Further the detection methods can be classified as:-

- Signature based detection: Detecting viruses on the basis of database containing all known viruses.
- Anomaly based detection: Detecting viruses on the basis of abnormal behavior of the system. It has the capability of detecting novel attacks.
- Pattern matching: Data pattern of the file under observation is checked and compared with the anti-virus database to find any similar pattern, if found the file is termed as malicious [3].
- Code Emulation: Creates a virtual environment which simulates CPU and memory activities to mimic the code activity [1].

B. Advantages of the Proposed Work

Some of the major drawbacks of the existing anti-virus programs are:

- Firstly, they can detect the potentially known attacks but not the unknown ones.
- Secondly, they take much more time to scan the system.
- Thirdly, the database of known attacks or viruses is not updated regularly for the anti- virus to function correctly [1].

These limitations can be mitigated by using the mechanism of the proposed work which is as follows:

1. To get hold of all files on the system and for complete coverage concept of registry and CAS is used.
2. Logs about vulnerable files on the system are generated so that time is not wasted to check the entire system instead only the vulnerable files can be targeted to speed up the process.
3. The logs can be used later to build the anti-virus software which will be capable to detect the unknown viruses also.

C. The Proposed Program Works in Three Phases

Detection Phase

In this phase the system scan takes place. Using code access security the registry files and all the system files are scanned.

Identification Phase

After the scan is complete those files which have vulnerable file extensions (as listed in Table no.3) are recorded and a Log file is generated.

Log generation Phase

There are 2 types of log files that are generated. The first log file, Consists of the file names and locations of those files which have vulnerable file extensions and the Second log file, which is a notepad file which consists of the date and time of scans that the system has ever undergone.

These steps ensure that all the files in the system that can ever have any malicious content are listed and later on these can be used so as to detect novel viruses and

train the antivirus so that it can detect the viruses and malicious content.

After these steps an Antivirus program can be formulated:-

Once the infection has been detected, determine which file has been infected and locate the virus location. Once Identification has been achieved, identify the specific virus that has infected a program.

Removal phase: Once identified, remove all traces of the virus from the infected program and restore it to its original state. Remove the virus from all infected systems so that it cannot spread further.

IV. Detection Phase

For designing an antivirus program, it becomes essential to know about various vulnerabilities existing in the system. To study this, A NIST (National Institute of Standards and Technology) report 2015 on the National Vulnerability Database (NVD) has been reviewed. The database gives statistics on number of vulnerabilities in operating systems [10]. Some of the vulnerabilities in some of the known operating systems are shown in Table 1.

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

Similarly, database also contains information on number of vulnerabilities in web browsers. The report can be seen in Table 2.

Table 2. Browsers Vulnerability Report

S. No.	Browser	Number of vulnerabilities
1.	Microsoft internet explorer	231
2.	Google chrome	187
3.	Mozilla Firefox	178
4.	Apple safari	135
5.	Mozilla Firefox extended support release	94

From the above information, it can be inferred that even the best known web browsers and widely used operating systems are not secure and can be compromised easily. Therefore, it becomes very important for the user to take certain preemptive measures to act as a safe guard while downloading and opening some unknown file extensions.

A. Registry

Registry is a hierarchical database maintained centrally by Microsoft windows to store system configuration information of hardware devices as well as of applications. To scan the entire system, it is required to provide the essential paths in the code of various registry files present in the system. Basically all the files present in the system have their instances in the registry. It contains system wide information and user specific information.

On opening registry editor in windows, it lists five main folders or root keys. These may further contain sub keys [2]. The root keys are:-

1. HKEY_CLASSES_ROOT (HKCR)

Provide all the necessary information about specific user profile.

2. HKEY_CURRENT_USER (HKCU)

Provides information about the current user who is logged in. Control panel, display and built-in program settings are stored here.

3. HKEY_LOCAL_MACHINE (HKLM)

Contains system wide information specific to local machine.

4. HKEY_USERS(HKU)

Contains information about all the user profiles that exist on the system.

5. HKEY_CURRENT_CONFIG (HKCC)

Provides information about the hardware profile used by computer during start up.

The registry information is mainly used by these files:-

- **SAM files (Security Account Manager)**-stores login information and password hashes during account creation.
- **SECURITY file-** Used to identify current and archived system passwords if present.
- **SOFTWARE file–** It contains information regarding windows like install date, recycle bin settings, user profiles and other.
- **SYSTEM registry file –** It contains Hardware configuration settings required to boot and manage the system.
- **NTUSER.DAT**– it stores Information regarding most recently used files.

Once the system scan is completed and logs are generated for various files present on our system, we can look for those files which have dangerous file extensions as they can contain malicious code. These files should be handled with care and scan them for viruses.

B. Code Access Security

Today's computer system are frequently exposed to code originating from various possibly unknown resources To access the files in the system it is required to gain file permissions. For this it is necessary to deploy CAS in .NET.

- Defines permissions and permission sets that define the rights to access various system resources.
- Code access security is a mechanism that is implemented to protect the access to the files for the security of file and its operations.
- It also outlines various positions and right to outline various resources.
- This mechanism also allows various administrators to configure security settings by associating sets of permissions with groups of code.
- It also specifies the permissions which the code has and the ones which the code should never have.
- Enforces restrictions on code at run time by comparing the granted permissions of every caller on the call stack to the permissions that callers must have.
- Enables code to demand that its callers have specific permissions.
- Enables code to demand that its callers possess a digital signature, thus allowing only callers from a particular organization or site to call the protected code.

Full trust is a function in CAS which specifies requests on built in permission sets.Permissions are accessed by CAS using<*assembly: Permission SetAtrribute (SecurityAction.Requestminimum, Name: = "full trust")*>.

V. Identification Phase

After getting hold of each and every file on the system using the concepts of registry and code access security, in order to detect which files are vulnerable we need to look into the file extensions first of the files scanned. This is done so that the files which can have some code embedded in them are critically analyzed and no virus goes undetected.

A. Dangerous File Extensions

All phishing attacks and spyware travel through a network in the form of legitimate file extensions embedded with viruses. In order to nullify their effect, it is required to track these files and remove it. Looking at the file extensions, it can be found out whether a file attached to an email or file downloaded from web is safe or not [8].

These files are required to be handled with extra care to ensure that the system and the data remain protected. These extensions contain embedded code, scripts and other potentially dangerous arbitrary commands whereas media files like JPEG and MP3 are however, not vulnerable as these files do not contain code. [8]. Table 3 shows different file extensions that are vulnerable in nature.

Table 3. Vulnerable Extensions of Programs 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, .PS1XML, .PS2, .PS2XML, .PSC 1, .PSC2	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,.PP T	Microsoft word, excel and power point documents. Can contain malicious macro code.
.DOCM, .DOT M, .XLSM, .XLTM, .XLAM , .PPTM, .POTM, .PPAM, .PPSM, .SLDM	New file extensions added in office 2007. M in the end indicates that the document contains macros.

VI. Designing

The framework for system scan is designed in .net

framework. The various parameters taken for scanning the system are defined in Table 4.

Table 4. System Scan Takes Place on the Basis of These 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.

In this paper, a framework is designed in Visual Basic using .NET and the process how this scan will be take place is defined in Fig.2. The first step is to design a front end interface in visual .NET framework and then to access various permissions assigned in CAS. Certain parameters are to define on the basis of which system scan is to be done. A detail log will be generated showing registry paths and all file extensions. The collected data can be further analyzed for detecting vulnerable files.

Pseudo code of the proposed system:-

Begin SCAN

Load the front end interface

If the permissions are accessed by CAS parameter using*<assembly: Permission SetAtrribute (SecurityAction.Requestminimum, Name: = "full trust")>*

Scan the system (com, API scan) on the basis of listed parameters on defined registry paths,

If the scan is complete,

Generate log files

List details on scanned files on extensions

When the last scan was done,

If logs generated,

Study logs

Analyze files as harmful or safe on their file extensions,

End SCAN

This process defines how the scans will initiate and executed.

Here is the detailed discussion of the flow in which the entire process functions:

1. **Begin Scan:** By clicking on the start button, the process will start and will scan the front end interface.
2. **CAS:** As discussed earlier this parameter will be used by the code to access the registry files. It will basically define the extent to which the code can penetrate and access the files. The implemented code uses this function for the required results.

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

3. **System scan:** System scan parameters are listed in Table 4. It form the basis for the scan. Each of the listed parameter consists of a registry path which leads to the set of files it access and scans. The synchronization between each of them one after other is implemented using the concept of threads.
4. **Log generation**: Two kinds of log files are generated. The first one gives details about the scanned files having any vulnerable file extension and how harmful it is, will be decided according to the extension it holds for example .exe and .dll files are the most vulnerable. The second log file is a notepad file which will give details about the date and time when the last scan was done.
5. **Analyzing**: After the logs are generated they are further analyzed to check if they contain any malware. This can be done by pattern matching algorithms using signature based detection techniques.

Following is a flow chart describing the entire flow in a diagrammatic manner:-

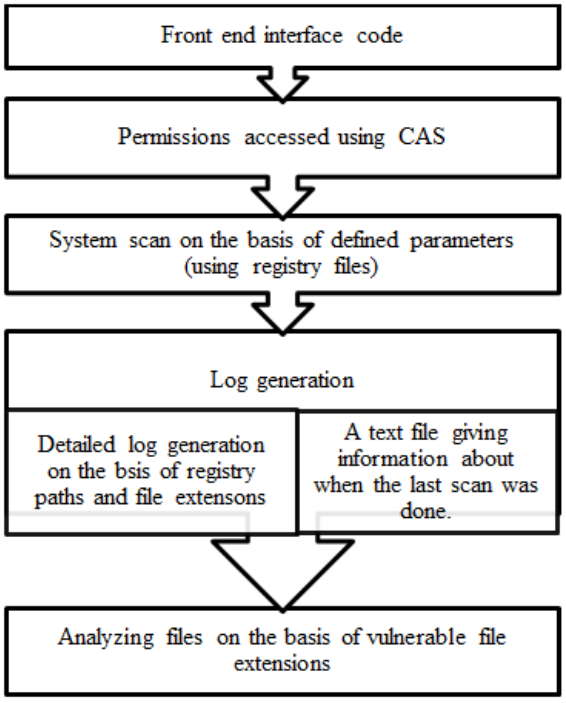

Fig.2. Steps of System Scan

VII. Implementation

To implement a design to scan the entire system and

generate logs, A Visual Basic framework is used which takes into account all the mentioned parameters in Table.4 and detects vulnerable files based on the their file extensions which are present in the system. Visual Basic 2013 tool is used to create the interface and different windows are programmed using the concept of threads so that they load one after the other as the process completes. The screenshots of the multiple windows which occur have been attached and explained further. Programming for this scan is done using .NET.

1. *Front end interface*

Scanning of the system is done on the basis of parameters as discussed earlier, Figure 3 Shows front end interface of system scan.

Fig.3. Parameters for system scan

2. *Scan in progress*

This option gives a key number assigned to each file being scanned and gives the previous match found of the similar file. The result is displayed in percentage scan completion as shown in Fig.4.

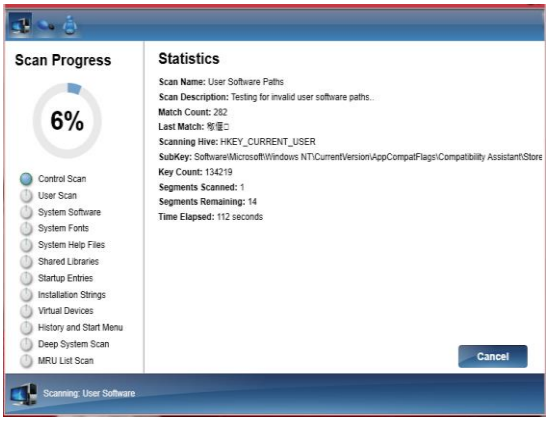

Fig.4. Scan in Progress

3. *Scan complete*

Overall Scan of the system is complete. Generated result shows in the form of log as shown in Fig.5.

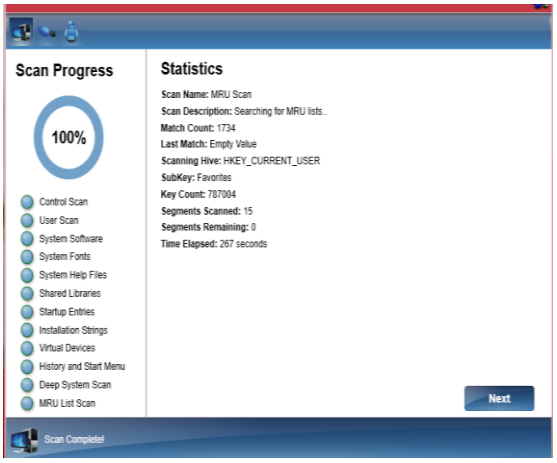

Fig.5. System Scan Complete

4. *Log Details*

On completion of scan, logs will be generated. This file gives a complete overview of files scanned as shown in Fig. 6. All files with vulnerable extensions and others can be clearly seen. Other details that can be seen are the date and time of last scans till now.

These details can be viewed as a notepad file also as shown Fig.7. The study of these logs can help in determining vulnerable file that are required to be handled with care or scan them through an antivirus so as to keep system secure.

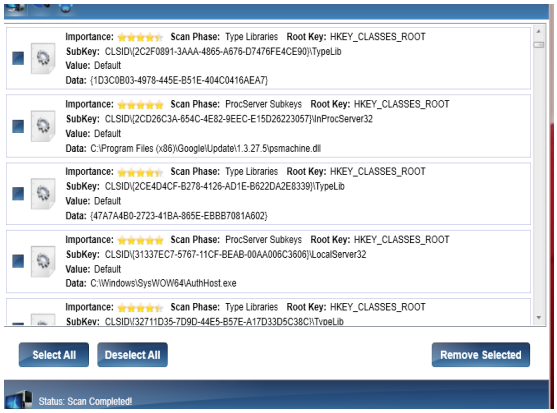

Fig.6. Detailed Log Generation

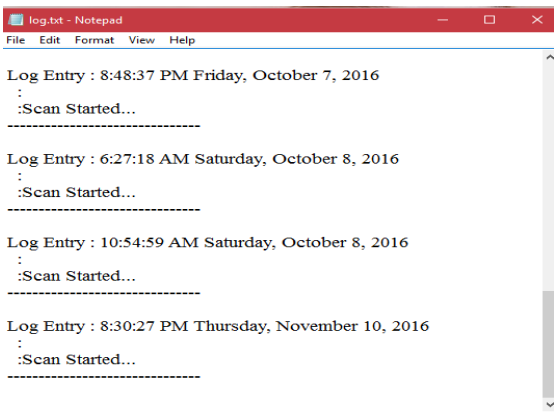

Fig.7. Notepad File

Once the complete system scan is done and logs are generated, study of these logs can help in determining vulnerable files and which are the files that are required to be handled with care or scan them through an antivirus so as to keep system secure.

VIII. Conclusion And Future Work

By scanning the system and detecting vulnerable files on the basis of the file extensions we get hold of all files on the system and no file is left undetected by the proposed system. Considering this as the basis an antivirus program can be created which will look for the viruses in these detected files. The developed framework revolves around three domains Detection, Identification and log generation. Further a framework can be developed using the results achieved from this proposed work which may have the following functionalities:-

- To check for malware in the detected files.
- To know the location or service which will be affected by the malware or virus detected.
- Stop the file causing virus to function and generate logs which the user can view and know why the service is affected.

The various system scanning approaches used in the anti viruses currently do not take into account the entire system files and some files are left undetected other than this the they are time consuming but the proposed system not only takes into account the entire system but also generates logs about the files which are vulnerable because of their file extensions. This approach is fast in comparison to the currently used approaches and can be used to build strong antivirus systems which can detect malware without consuming much time.

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 Forensics Conference, 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] http://www.howtogeek.com/137270/50-file-extensions-that-are-potentially-dangerous-on-windows/

[9] https://msdn.microsoft.com/en-us/library/930b76w0(v=vs.90).aspx

[10] http://www.gfi.com/blog/2015s-mvps-the-most-vulnerable-players/

[11] Xin luo and Qinyu Liao, "*Awareness education as the key to ransomware prevention*", Information systems security, USA, 2007.

[12] 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.

[13] Sarat komplli, "*Using Existing Hardware Services for Malware Detection*", IEEE security and privacy workshops, 2014.

[14] Takahiro Kasama, Katsunari Yoshioka, Daisuke Inoue and Tsutomu Matsumoto, "*Malware Detection Method by Catching Their Random Behavior in Multiple Executions*", IEEE/IPSJ 12th International Symposium on Applications and the Internet, 2012.

[15] Parvez Faruki et. al., "*Android Security: A Survey of Issues, Malware Penetration, and Defenses*", IEEE communication surveys & tutorials, vol. 17, no. 2, Second quarter, 2015.

[16] Shirish Singh, Bharavi Mishra and Saket Singh, "*Detecting Intelligent Malware on Dynamic Android Analysis Environments*", the 10th International Conference for Internet Technology and Secured Transactions, 2015.

[17] Konrad Rieck, Philipp Trinius, Carsten Willems and Thorsten Holz, "*Automatic Analysis of Malware Behavior using Machine Learning",* Journal of Computer Security, IOS Press, 2011.

Author's Profiles

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.

Shilpa Mahajan is currently working as a Senior Assistant Professor in the Department of CSE & IT in NorthCap University, Gurgaon, India. She has more than nine years of teaching experience at post graduate and under graduate level. She is a committed researcher in the field of Sensor Network, and has done her PhD in the area of Wireless Sensor Network at Guru Nanak Dev University, Amritsar. She specializes in computer Networks, Data Structures, Operating System and Mobile Computing. She has guided 9 M.Tech Thesis and 12 B.Tech Projects. She has published 27 research papers, 13 in peer reviewed reputed International Journals and 14 in IEEE and Springer indexed Conferences .She is a CCNA certified Instructor and has also done certifications in Data Scientist Tool, Exploratory data analysis and Getting and Cleaning Data from Johns Hopkins University. She is the Lifetime member of ISTE.

How to cite this paper: Sonali Sharma, Shilpa Mahajan,"Design and Implementation of a Security Scheme for Detecting System Vulnerabilities", International Journal of Computer Network and Information Security(IJCNIS), Vol.9, No.10, pp.24-32, 2017.DOI: 10.5815/ijcnis.2017.10.03

