ISSN:2348-2079

Volume-8 Issue-2

International Journal of Intellectual Advancements and Research in Engineering Computations

Deep scan based potential virus detection using legitimate signature for securing systems

Imran Khan M[1], Ramesh S[2]

[1]PG Scholar, Dept. of Computer Science and Engineering, Krishnaswamy College of Engineering and Technology, Nellikuppam Main Road, S.Kumarapuram, Cuddalore 607109.

[2]Asst Professor, Dept. of Computer Science and Engineering, Krishnaswamy College of Engineering and Technology, Nellikuppam Main Road, S.Kumarapuram, Cuddalore 607109.

ABSTRACT

Scandalous venomous files like Trojan, Worm, and Malware has the potential to destroy any line businesses by luring those organization vital data's. This is also a huge headache for cloud computing architecture where many virtual machine and cloud service provider are involved for data uploading and downloading as no company can afford its valuable data loss. Security is the main issue for many large organization and cloud service provider. Amongst many security issues zombie attack is the most notorious type of attack. This attack has the capability to reduce network performance either by delaying the service or by consuming network bandwidth. This attack can be an insider attack or it can be an outsider attack, here the malevolent users will gain the data of any legitimate user by luring them with venomous zombie file which will compromise the victims system and this node will start communicate with virtual machine on the behalf of legitimate user. Despite many techniques like Honeypot, antivirus software are there to prevent such activities it cannot completely protect the system from malicious files as most of the harmful files which can compromise the system goes undetectable by the existing systems. The proposed method through deep content mining technique along with the existing techniques shows a promising result by detecting all possible vulnerable files along with those undetectable venomous file by existing systems.

Keywords: Antivirus, Insider attack, Malware, Outsider attack, Trojan, Worm, Zombie

INTRODUCTION

In this era detecting Malware and other potential files which are harmful to security related issues has become one of the most significant features of security software in the Computer Science area. With the mounting figures of personal computer users the need for noble malware detection algorithms has been increasing with considerable rate.

It is also estimated that in early 90's, the quantity of computer infections was projected from 1,000 to 2,300 viruses, whereas in 20K there were 60,000 known viruses, Trojans, worms, and disparities. Today there are well over 100,000 known malicious computer programs [1]. Studies and researches show that a computer system connected to the Internet may experience an attack every 39 seconds [2]. Fresh susceptibilities in the system are revealed every few days. These susceptibilities are fixed by the software vendors like Antivirus companies who provide patches and updates for the system. However, in mean time the computer system will be compromised by hackers using malevolent programs that are installed on user machines to steal secret data for financial gains. The compromised system can also be made a part of huge Internet-connected devices that can be used to blastoff Denial of Service attacks on

Author for correspondence:
Dept. of Computer Science and Engineering, Krishnaswamy College of Engineering and Technology

servers, or be used in an attempt to intervene the computers of any organization [3].

Computer virus creator uses many tactics to elude detection such as space filling, compressing and encryption, in another hand; the antivirus software trying to detect the viruses by using variant static and dynamic methods. However; all the existing methods are not adequate. In order to develop new steadfast antivirus software some problems must be fixed.

One such approach is Virus Total. Virus Total [5] is a crowd-sourced virus scanning project backed by Google. Virus Total implores uncertain files and URLs from users, subscribers and other site visitors and scans them with solutions from over 70 anti-virus (AV) tools suppliers. Basic outcomes are shared with submitters and among contributing commercial partners who, in theory, use results to enhance their anti-virus software, collectively contributing to the advancement of global Information Technology security.

Many of the antivirus companies had piggybacked on Virus Total and other shared Anti-Virus services as a means to advance their virus signature libraries by effectively re-using their competitors' detection engines for free. Even then many harmful files are getting undetected by these global solutions. In order to detect and prevent such attack the proposed system uses a novel technique which involves in deep content mining technique and calculating MD5 hash code for detecting the malware contents in file like .bat extensions, exe file respectively which are undetected by the existing software systems.

The proposed method uses Suffix tree algorithm for identifying harmful pattern inside the file and (TF-IDF) Term Frequency Inverse Document Frequency for identifying the needed pattern by ignoring unwanted terms in a file. The aim of this project is to build an application that could combine the existing system along with the extraction of malicious patterns in the suspected files which are undetectable by existing system. The techniques considered in the proposed system could be split in two – First is to identify the malicious file using Virus-Total database and add those signature in virus database and the second is to analyze the file content whether it has the potential to harm the system using content mining with Suffix tree and TF-IDF algorithm. Here the Suffix tree algorithm used to identify the virus pattern and the TF-IDF algorithm to check the malicious command pattern count for calculating weighted score to determine whether the file is malicious or not.

STATE OF ART

A zombie is a computer that a remote attacker has accessed and set up to forward transmissions which includes spam and viruses to other computers on the Internet. The persistence is usually either for economic gain or meanness. Attackers usually exploit multiple computers to create a single connected botnet. Normally, a zombie is a user's personnel computer whose possessor is naive that their computer is being exploited by an external third party. The increasing pervasiveness of high speed connections makes user's personnel computer computers become an attractive targets for attack. Insufficient security measures make access comparatively easy for an attacker. For example, if an Internet port has been left open, a small Trojan horse program can be left there for future instigation.

Zombie attack is usually happens in large organization that have huge customers data, cloud services, web server in order to gain financially. Rakshitha C M; Ashwini B P in their paper surveyed the techniques which can detect and mitigate the zombie attacks in cloud environment [7]. Sujatha Sivabalan, P J Radcliffe in their paper explained an adaptive, real time scoring system for detecting zombie attack in web server [8]. P.K. Agrawal, B.B. Gupta, Satbir Jain, proposed a machine learning approach based on support vector machine for regression to predict the number of zombies in a Distributed Denial of Service (DDoS) attack using Network Simulator [9]. It will count the number of zombies present in DDoS attack.

In Zombie attack the compromising of a system is done using malwares and viruses which evade the detection of security software installed in the user system. Here such computer virus and techniques to evade the security systems as given in the following. A computer virus is a computer program that has the capability to copy itself and infect a computer without authorization or

awareness of the user. In order to avoid detection [5], many virus programs will use different kinds of trick such as the virus program overwrites files with their own copy. Though, this is a very primeval procedure, but it is certainly the easiest approach of all.

The another approach is to becoming a companion that is to give the virus the same base name as the targeted program, but use different extension than the original file. Here when the victim attempts to launch program, the virus will be created in such a way that user system will give priority to a virus file over a file with the same base name.

In some technique, a jump instruction is inserted at the front of the host to point to the end of the original host. This technique can be implemented for any type of executable file. Usually files like that will have a header section that stores the address of the core entry point, which, in most cases, will be replaced with a new entry point to the start of the virus code affixed to the end of the file.

In order to prevent the user system from vulnerability, the security software in the system must be update its security features to detect the harmful files. The antiviruses scan the computer using some specific patterns of bytes indicative of known viruses. To stay up-to-date, the Antivirus organization must be updating their databases periodically whenever new viral strains arise. Computer virus scanners use pattern matching algorithms to scan for many different signatures at the same time the best checking up to 10,000 signatures in 10,000 programs in less than 10 minutes [6].

Computer virus authors and antivirus vendors have constantly fought in an evasion of detection game through creation of new virus signatures. Computer malwares have become more and more sophisticated, using advanced code obfuscation techniques to resist antivirus detection. The computer viruses like Polymorphic and metamorphic are presently the toughest kinds of viruses to identify. Both types of viruses are able to mutate into an infinite number of functionally equivalent copies of themselves [10]. This sophistication comes with the creation of new virus patters that are not easily detectable by the antiviruses available in the market today. Heuristic detection is a scanning mechanism that anti-virus software employs in detecting for virus signatures. The heuristic detection methods encompass more than 250,000 new virus signatures and are most effective for locating new virus signatures. Virus Total [11] is one such mechanism which is backed by Google along with other Antivirus companies. Multiple solutions fail to detect the very same viruses which can be observed using the Virus Total Scanner.

The key objectives in this project is to club the existing signature-based detection techniques like hash signature detection and byte signature detection using Virus Total database and content mining of the suspected file to analyze whether it contains any malware pattern, so that the virus which evades detection even by the global Virus Total database can be detected with ease, so that the system can be safeguard from any harmful files which cause zombie or any other attack.

SYSTEM OVERVIEW

The proposed system uses Internet of Things (IOT) information analysis from Virus Total database for identifying the potential viruses which are the cause for severe attack. It uses Suffix tree algorithm for virus pattern identification and TF-IDF algorithm for finding the harmful virus file which has the capability to bypass the security system present in the system and also which evades the detection even by Virus Total database. The flow the proposed system is shown in Fig -1 below.

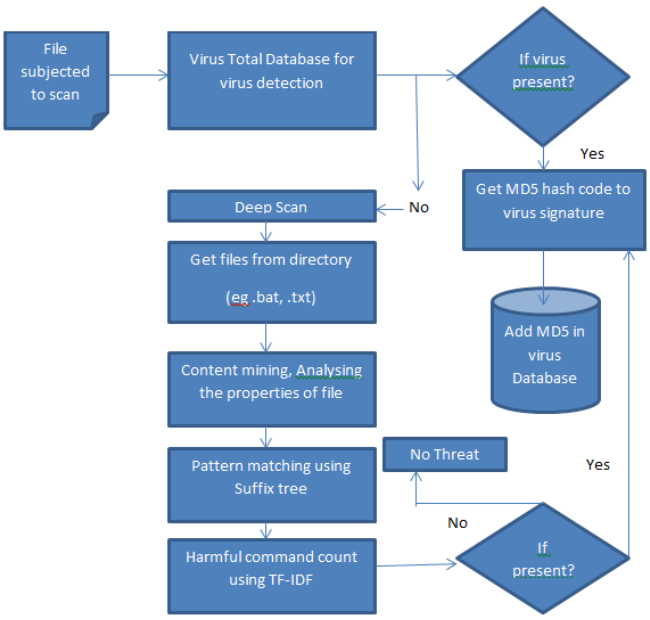

Fig -1: **Virus Detection Mechanism**

Problem Definition

For antiviruses, a signature is an algorithm or hash that uniquely identifies a specific virus. Reliant on the nature of scanner being used, it may be a static hash which, in its meekest form, is a calculated numerical value of a piece of code that is distinctive to the virus [12]. Javier [13] stated that a virus signature should be understood how a reliable way to detect a host infected by concrete malware. It encapsulates the essence of a virus. Signature detection is complex and challenging but we will keep the focus on the need of gathering a simple signature together with related context information [14]. As mentioned earlier one such encapsulated virus signature database is there for Virus Total database. It will detect the viruses which could not be identifies by the security software present in a user system as it has global database which may not be present in the security software present in user system. Table -1 shows the prediction of virus by Virus Total service. It lists some Antivirus company which take active participation in Virus Total service for virus detection.

Table -1: Virus detection using Virus Total

Virus detection using Virus Total	
Anti-Virus	**Status**
TotalDefence	Clean
CMC	Clean
MicrWorld-e	Trojan.Joke.PXJ
ESET-NOD32	Clean
K7GW	Clean
K7 Antivirus	Clean
Baidu	Clean
Nano-Antivirus	Clean
Symantec	Clean
McAfee	Clean
Zilya	Clean
TheHacker	Clean

Bikav	Clean
AegisLab	Clean
Avast	Clean

Proposed System

The virus detection by Virus Total service is given in the above Table -1, it shows that most of the antivirus could not predict the virus file precisely only one antivirus could able to detect it, though we could find the virus by correlating all the results provided by the Virus Total, it shows false positive result, as it predict clean even for some virus file. In order to predict these undetected virus we are using suffix tree to identify the virus pattern whether it exist in the scanned file or not.

A suffix tree is built of the text. After preprocessing text, we can search any pattern in O(m) time where m is length of the pattern. Suffix tree algorithm is good for fixed text or less frequently changing text in less time compare to other technique like Rabin Karp Algorithm, Finite Automata based Algorithm, Boyer Moore Algorithm. The following Fig -2 explains the pattern mapping of suffix tree.

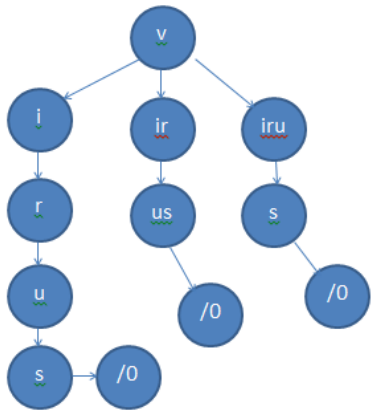

Fig -2: **Suffix tree pattern matching for "virus" string**

Output: vi, vir, viru, virus

Once the pattern has been identified using TF-IDF algorithm the proposed identify the frequency of pattern in order to identify the vulnerability of the file, the Term frequency (TF) count will be calculated using the given formula below.

$$TF = \frac{Tw}{Tn},$$

Where T_w is the number of times the pattern appears in file and T_n is the total number of words in that file.

Thus the virus which escapes from security software present in the system and Virus Total service could be easily identified.

EXPERIMENTAL RESULT

The experimental results were done several times using various viruses with the free commercial Avast Antivirus and Virus Total service, Though Virus Total systems could detect virus which is not detected by the Avast, it also could not detect many viruses even with such huge database. The proposed system with the combined feature of Virus Total and its detection mechanism which explained above could detect those viruses efficiently. The Error Rate (ER) is estimated using the correctly identified page with the samples of 100 fake sites. The correctness value (CV) is identified using the formula.

$$CV = \frac{CI}{n}$$

The Correctness Rate is calculated using the formula,

$$CR = \frac{CV * 100}{100}$$

The Error Rate is given by the formula,

$$ER = 100 - \frac{\sum_{i=l}^{l} CR}{l}$$

Here l is 10 since we took samples in the terms of 10's up to 100 viruses. From the ER, the system accuracy can be identified as 97%. The tabulation for virus identification is given below in Table -2.

Table -1: Identification of virus

Identification of virus with Existing and Proposed system			
	Free Avast Antivirus	**Total Virus Software**	**Proposed system**
No of viruses (n)	Correctly identified (CI)	Correctly identified (CI)	Correctly identified (CI)
10	6	8	9
100	87	95	98
150	139	143	148
200	187	192	198
250	240	246	197

The Chart -1 shows the performance evaluation of existing and the proposed system

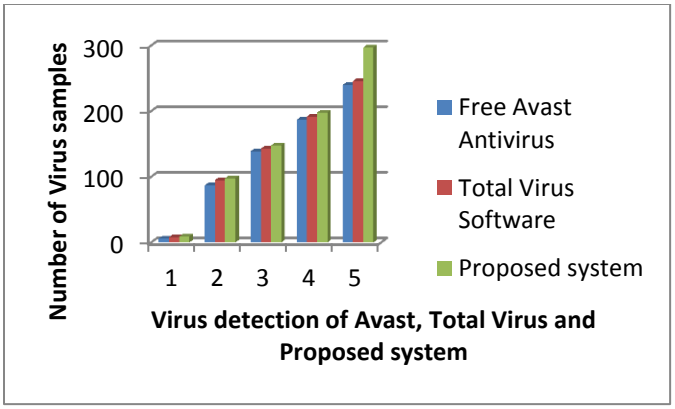

Chart -1: **Virus detection of existing and proposed system**

CONCLUSIONS

Mostly the security software's are works better against known viruses' signature and will not stand against any new viruses, as its signature will not be present in its database. Even though the virus detection methods have some major issues for newbie virus, global virus signature database has the capability to overcome these issues to some extent, but could not provide complete solutions. Thus the proposed system combined with these existing method and with its pattern extraction of suspicious command from given scanned file could be effective and can provide a satisfied result.

REFERENZES

[1]. http://www.cknow.com/vtutor/NumberofViruses.html,2008.
[2]. http://csdl2.computer.org/comp/mags/it/2007/02/f2004.pdf, 2008.
[3]. R. Srinivasan, "Protecting Anti-Virus Software under Viral Attacks, Master Degree of Science", Arizona State University, 2007.
[4]. M. Bailey, J. Oberheide, J. Andersen, Z. M. Mao, F. Jahanian, and J. Nazario, "Automated classification and analysis of internet malware", In Proceedings of the 10th Symposium on Recent Advances in Intrusion Detection (RAID'07), 2007, 178–197.
[5]. S.I. Shoutkov, A.V. Spesivtsev, "Computer viruses: ways of reproduction in MS-DOS", 2002.
[6]. Jeffrey Kephart, Gregory Sorkin, David Chess, Steve White, "Fighting Computer Viruses. USA: Scientific American", 1997.
[7]. Rakshitha C M, Ashwini B P, "A survey on detection and mitigation of zombie attacks in cloud environment", 2016.
[8]. Sujatha Sivabalan, P J Radcliffe, "Detecting IoT zombie attacks on web servers", 2017.
[9]. P.K. Agrawal, B.B. Gupta, Satbir Jain, "SVM Based Scheme for Predicting Number of Zombies in a DDoS Attack", 2011.
[10]. Serge Chaumette, O. L, "Automated Extraction of Polymorphic Virus Signatures using Abstract Interpretation". France: University of Bordeaux, 2012.
[11]. https://mma.prnewswire.com/media/727967/VirusTotal_and_the_Comodo_Zeroday_Challenge_Black_Hat_2018.pdf.
[12]. Landesman, M. What is a Virus Signature? Retrieved from Lifewire Tech: 2016. https://www.lifewire.com/what-is-a-virus-signature-153629.
[13]. Mellid, J. M. Detecting and removing computer virus with OCaml. Retrieved from 2014. http://javiermunhoz.com/blog/2014/04/19/detecting-and-removing-computer-virus-withocaml.html
[14]. Mellid, J. M. Detecting and removing computer virus with OCaml. Retrieved from 2014. http://javiermunhoz.com/blog/2014/04/19/detectingand-removing-computer-virus-with-ocaml.html