DISSERTATION
ON

PATTERN BASED MALWARE DETECTION TECHNIQUE IN CLOUD ARCHITECTURE

Thesis submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology
In
Computer Science & Engineering

By
Under the supervision of

Manish Kumar Gupta
Roll Number
10400113141
Registration Number
131040120006 of 2013 - 2016

Sagar Shaw
Roll Number
10400113151
Registration Number
131040120016 of 2013 - 2016

Prof. Sanjay Chakraborty

Assistant Professor
Computer Science & Engineering
Institute of Engineering & Management

Department of Computer Science & Engineering
Institute of Engineering & Management
Gurukul, Y-12, Block –EP, Sector-V, Salt Lake Electronics Complex
Kolkata-700091, West Bengal, India

CERTIFICATE

This is to certify that the project report entitled "**PATTERN BASED MALWARE DETECTION TECHNIQUE IN CLOUD ARCHITECTURE**", submitted by **Sagar Shaw (Registration No. 131040120016 *OF 2013 – 2014*) Manish Kumar Gupta (*Registration No. 131040120006 OF 2013 – 2014)***, students of **INSTITUTE OF ENGINEERING & MANAGEMENT,** in fulfillment of requirement for the degree of **Bachelor of Technology in Computer Science & Engineering** is a bona fide work carried out by him under the supervision and guidance of **Prof. Sanjay Chakraborty** during the academic session of 2015-2016. The content of this report has not been submitted to any other University or Institute for the award of any other degree.

We are glad to inform that the work is entirely original and the performance is found to be satisfactory.

_____ _____

Prof. Sanjay Chakraborty
Project Guide
Dept. of Computer Science Engineering
Institute of Engineering & Management

Prof. Dr. Debika Bhattacharya
H.O.D
Dept. of Computer Science Engineering
Institute of Engineering &Management

_____

Prof. Dr. Amlan Kusum Nayak
Principal
Institute of Engineering & Management
Sector- V, Saltlake Electronics Complex, Kolkata-700091

ABSTRACT

Security is one of the major concerns in cloud computing. Security is obtained to prevent threats that affect both the cloud user and cloud provider. Malicious code deployment is the main cause of threat. Many antivirus software unable to detect many modern malware threats and its enlargement in its complication has resulted in indebtedness that are being explored by malware. Apart from this Cloud computing is becoming an increasingly popular paradigm due to new services and increased media attention. Thus, we propose a new model for malware detection on cloud architecture. This model enables identification of malicious and unwanted software by multiple detection engines as a result this approach provides several important benefits including better detection of malware. In this proposal we use combined detection techniques, DNA Sequence Detection Process, Symbolic Detection Process and Behavioural Detection Process. The Proposed approach (PMDM) can be deployed on a VMM which remains fully transparent to guest VM and to cloud users. PMDM prevents the malicious code running in one VM (infected VM) to spread into another non-infected VM with help of hosted VMM. The main aim of this proposal is to detect the malicious code by some advanced technique and warn the other guest VMs about it. A prototype of PMDM is partially implemented on one popular open source cloud architecture – Eucalyptus.

ACKNOWLEDGEMENTS

We would like to take this opportunity to thank everyone whose cooperation and encouragement throughout the ongoing course of this project remains invaluable to us.

We are sincerely grateful to our guide and mentor **Prof. Sanjay Chakraborty** of the Department of Computer Science & Engineering, IEM Kolkata, for his wisdom, guidance and inspiration that helped us go through with this project and take it to where it stands now.

We would also like to express our sincere gratitude to **Prof. Satyajit Chakraborty**, Director**, Prof. Dr. Amlan Kusum Nayak**, Principal and **Prof. Dr. Debika Bhattacharyya**, HOD of Computer Science & Engineering and other faculties of Institute of Engineering & Management, for their assistance and encouragement.

Last but not the least, we would like to extend our warm regards to our families and peers who have kept supporting us and always had faith in our work.

Sagar Shaw

Reg No. 131040120016 of 2013-2014.

Manish Kumar Gupta

Reg No. 131040120006 of 2013 - 2014.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1. INTRODUCTION

1.1 Motivation

Due to the recent developments in technologies, A cloud computing is one of the advance technology. In past few years the concept of cloud computing is the technology which is evolved rapidly in computer science. Cloud computing can be define as getting the work complete by sharing and using resources and application of a network environment. A large amount of files are used daily so in this environment so it is difficult to track that which file is malware free. So detecting malicious is a complex problem. The vast, ever-increasing ecosystem of malicious software and tools presents a daunting challenge for network operators and IT administrators. Antivirus software is one of the most widely used tools for detecting and stopping malicious and unwanted software. However, the elevating sophistication of modern malicious software means that it is increased challenging for any single vendor to develop signatures for every new threat. Indeed, a recent Microsoft receives over 150 thousand new unknown files each day to be analyzed. Till now many malwares are identified some of the known malwares are shown in [Table I] with their effects [8]. However, there were many techniques are define till now but no one can detect all malwares at the same time. Most existing techniques for malware detection are dependent on traditional malware signatures, i.e. the signature based detection techniques. The main motivation of this research work is to detect the malicious code by some advanced technique and warn the other guest VMs about it.

Malware	Effect
GrooveMonitor.exe	Files fails to respond or starts missing
indexervolumeguid	Once entered, tries to conquer every file
rundll32.exe	CPU is significantly consumed.
iloveyou.VBS	Overwriting random types of files and sent a copy of itself to all addresses in the Windows
Smishing.C	It is spam that lures user into phishing website.

TABLE I: Some Existing Malwares

[9]

1.2 Objective

The main objective of this work is to develop, implement and evaluate a prototype system for malware detection in cloud architecture. We try to propose a new model for malware detection on cloud architecture. This model enables identification of malicious and unwanted software by multiple detection engines. In this model a file mainly undergoes these process to detect malicious behavior. Firstly DNA sequencing is the process of determining the precise order of nucleotides within a DNA molecule to identify regions of local or global similarity. Then Symbolic detection process we cluster the files and use symbol to detect malware. At last to detect malware it is one of the best way to identify malware accurately is to analyses behavior of the file. In Behavioral detection process we determine whether it is a malicious program or not.

This Proposed Malware Detection Model is deploy into cloud architecture which gives the resultant as Cloud Deployment Model (CDM) with the help of free open source computer software Eucalyptus.

1.3 Organization

CHAPTER 1. INTRODUCTION: Gives a brief introduction on what is being done and why this topic has been chosen.

CHAPTER 2. BACKGROUND: In this section we shall discuss a literature survey of projects done on similar topics and how the authors tried to achieve their objective.

CHAPTER 3. PROPOSED FRAMEWORK: In this section, the methods of how to go about the entire technique is described.

CHAPTER 4. EXPERIMENTAL RESULTS AND ANALYSIS: This section produces a report of how our framework is performing.

CHAPTER 5. CONCLUSION: The limitations of our framework and future scope of the work i.e. where and how we can improve this technique to make it more suitable.

CHAPTER 2. BACKGROUND

Cloud computing, also known as on-the-line computing, is a kind of Internet-based computing that provides shared processing resources and data to computers and other devices on demand. It is a model for enabling ubiquitous, on-demand access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services), which can be rapidly provisioned and released with minimal management effort. Cloud computing and storage solutions provide users and enterprises with various capabilities to store and process their data in third-party data centers. It relies on sharing of resources to achieve coherence and economy of scale, similar to a utility (like the electricity grid) over a network.

Advocates claim that cloud computing allows companies to avoid upfront infrastructure costs, and focus on projects that differentiate their businesses instead of on infrastructure. Proponents also claim that cloud computing allows enterprises to get their applications up and running faster, with improved manageability and less maintenance, and enables IT to more rapidly adjust resources to meet fluctuating and unpredictable business demand. Cloud providers typically use a "pay as you go" model. This can lead to unexpectedly high charges if administrators do not adapt to the cloud pricing model.

The present availability of high-capacity networks, low-cost computers and storage devices as well as the widespread adoption of hardware virtualization, service-oriented architecture, and autonomic and utility computing have led to a growth in cloud computing. Companies can scale up as computing needs increase and then scale down again as demands decrease.

Cloud computing has become a highly demanded service or utility due to the advantages of high computing power, cheap cost of services, high performance, scalability, accessibility as well as availability. Some cloud vendors are experiencing growth rates of 50% per year, but being still in a stage of infancy, it has pitfalls that need to be addressed to make cloud computing services more reliable and user friendly.

Though service-oriented architecture advocates "everything as a service" (with the acronyms EaaS or XaaS or simply aas), cloud-computing providers offer their "services" according to different models, [need quotation to verify] which happen to form a stack: infrastructure-, platform- and software-as-a-service.

- **Software-as-a-service (SaaS)**: In the software as a service (SaaS) model, users gain access to application software and databases. Cloud providers manage the infrastructure and platforms that run the applications. SaaS is sometimes referred to as "on-demand software" and is usually priced on a pay-per-use basis or using a subscription fee.

  In the SaaS model, cloud providers install and operate application software in the cloud and cloud users access the software from cloud clients. Cloud users do not manage the cloud infrastructure and platform where the application runs. This eliminates the need to install and run the application on the cloud user's own computers, which simplifies maintenance and support. Cloud applications differ from other applications in their scalability—which can be achieved by cloning tasks onto multiple virtual machines at run-time to meet changing work demand. Load balancers distribute the work over the set of virtual machines. This process is transparent to the cloud user, who sees only a single access-point. To accommodate a large number of cloud users, cloud applications can be multitenant, meaning that any machine may serve more than one cloud-user organization.

  The pricing model for SaaS applications is typically a monthly or yearly flat fee per user, so prices become scalable and adjustable if users are added or removed at any point.

  Proponents claim that SaaS gives a business the potential to reduce IT operational costs by outsourcing hardware and software maintenance and support to the cloud provider. This enables the business to reallocate IT operations costs away from hardware/software spending and from personnel expenses, towards meeting other goals. In addition, with applications hosted centrally, updates can be released without the need for users to install new software. One drawback of SaaS comes with storing the users' data on the cloud provider's server. As a result, there could be unauthorized access to the data. For this reason, users are increasingly adopting intelligent third-party key-management systems to help secure their data.

- **Platform-as-a-service (PaaS)**: PaaS vendors offer a development environment to application developers. The provider typically develops toolkit and standards for

development and channels for distribution and payment. In the PaaS models, cloud providers deliver a computing platform, typically including operating system, programming-language execution environment, database, and web server. Application developers can develop and run their software solutions on a cloud platform without the cost and complexity of buying and managing the underlying hardware and software layers. With some PaaS offers like Microsoft Azure and Google App Engine, the underlying computer and storage resources scale automatically to match application demand so that the cloud user does not have to allocate resources manually. The latter has also been proposed by an architecture aiming to facilitate real-time in cloud environments. Even more specific application types can be provided via PaaS, such as media encoding as provided by services like bitcodin.com or media.io.

Some integration and data management providers have also embraced specialized applications of PaaS as delivery models for data solutions. Examples include iPaaS and dPaaS. iPaaS (Integration Platform as a Service) enables customers to develop, execute and govern integration flows. Under the iPaaS integration model, customers drive the development and deployment of integrations without installing or managing any hardware or middleware. dPaaS (Data Platform as a Service) delivers integration—and data-management—products as a fully managed service. Under the dPaaS model, the PaaS provider, not the customer, manages the development and execution of data solutions by building tailored data applications for the customer. dPaaS users retain transparency and control over data through data-visualization tools.

Platform as a Service (PaaS) consumers do not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but have control over the deployed applications and possibly configuration settings for the application-hosting environment.

- **Infrastructure-as-a-service (IaaS)**: In the most basic cloud-service model—and according to the IETF (Internet Engineering Task Force)—providers of IaaS offer computers—physical or (more often) virtual machines—and other resources. IaaS

refers to online services that abstract the user from the details of infrastructure like physical computing resources, location, data partitioning, scaling, security, backup etc. A hypervisor, such as Xen, Oracle VirtualBox, Oracle VM, KVM, VMware ESX/ESXi, or Hyper-V runs the virtual machines as guests. Pools of hypervisors within the cloud operational system can support large numbers of virtual machines and the ability to scale services up and down according to customers' varying requirements. IaaS clouds often offer additional resources such as a virtual-machine disk-image library, raw block storage, file or object storage, firewalls, load balancers, IP addresses, virtual local area networks (VLANs), and software bundles. IaaS-cloud providers supply these resources on-demand from their large pools of equipment installed in data centers. For wide-area connectivity, customers can use either the Internet or carrier clouds (dedicated virtual private networks).

To deploy their applications, cloud users install operating-system images and their application software on the cloud infrastructure. In this model, the cloud user patches and maintains the operating systems and the application software. Cloud providers typically bill IaaS services on a utility computing basis: cost reflects the amount of resources allocated and consumed.

2.1 Security in the Cloud computing

The Security in the Cloud is provided by many companies to detect malware with industry-leading detection rates. High-performance scanning engines detect the latest malwares. Cipher Cloud is company which provides service like this. The basic infrastructure of all these services are same. We are providing a basic infrastructure of Cloud Malware Detection see [Figure 1] [3] [9].

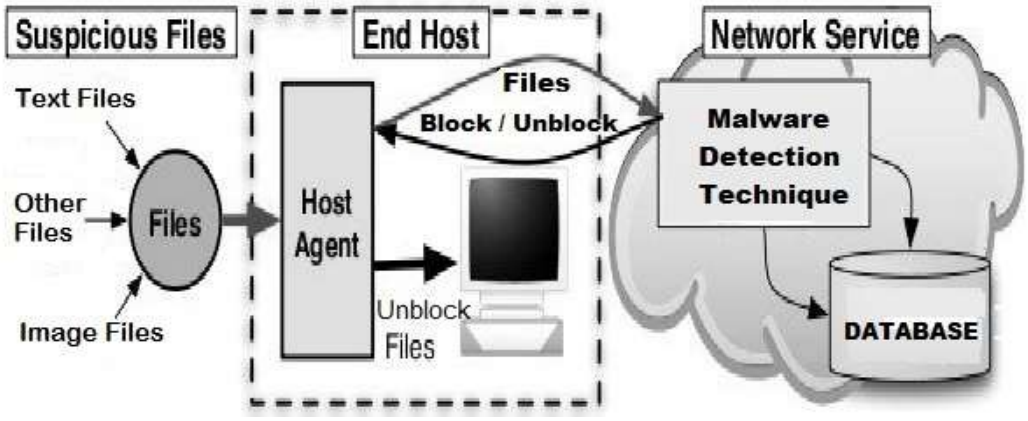

Figure 1. Cloud Malware Detection Technique

2.2 Related Literature Review

The traditional anti- virus software's can detect only those malware whose signatures are already present in the databases. These software's are not able to detect the new incoming malware because their signature is not present in the databases. This approach is based on the anomaly approach in which we try to analysis whether the incoming patterns is consists of malware or not see [Figure 2] [3].

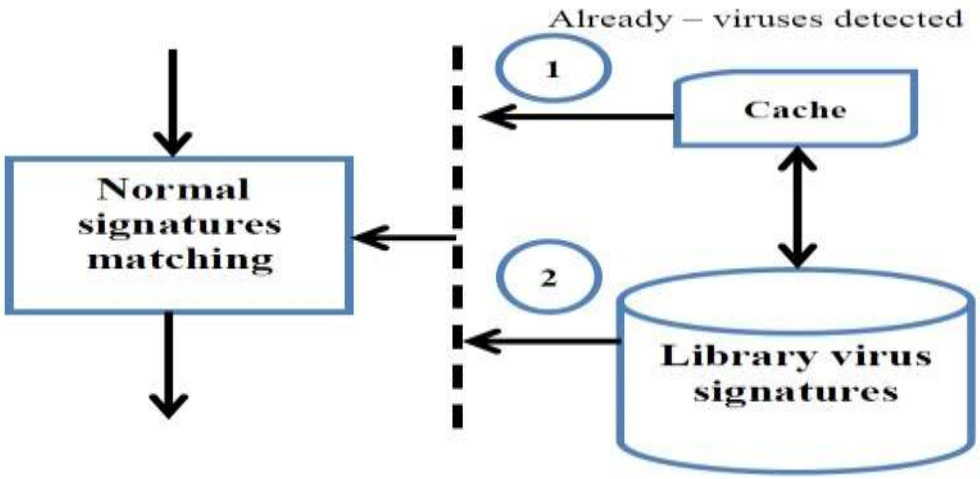

Figure 2. Traditional Anti-Virus Detection Method.

CHAPTER 3. PROPOSED FRAMEWORK

This work proposes a malware detection system to be built on cloud environment, by using a series of following steps:
Initially, we will divide the system architecture into two main sections according to the mechanism of action of each part. First part, explains the Proposed Malware Detection Model (PMDM) and the second part, explains Cloud Deployment Model (CDM).

Part I

3.1. Proposed Malware Detection Model

The proposal is to find the optimal solutions to the problems of anti-viruses and improve performance and find possible alternatives for a better working environment without problems with high efficiency and flexibility.
In this malware detection model, there are total three process are used to explain the mechanism,

3.1. 1. Process 1: DNA Sequence Detection Process

- DNA Sequence malware detection.

3.1. 2. Process 2: Symbolic Detection Process

- Cluster using file properties.
- Symbolic malware detection.

3.1. 3. Process 3: Behavioral Detection Process

- Behavioral malware detection using sandbox testing.

All of them are explained below in detail see [Figure 3].

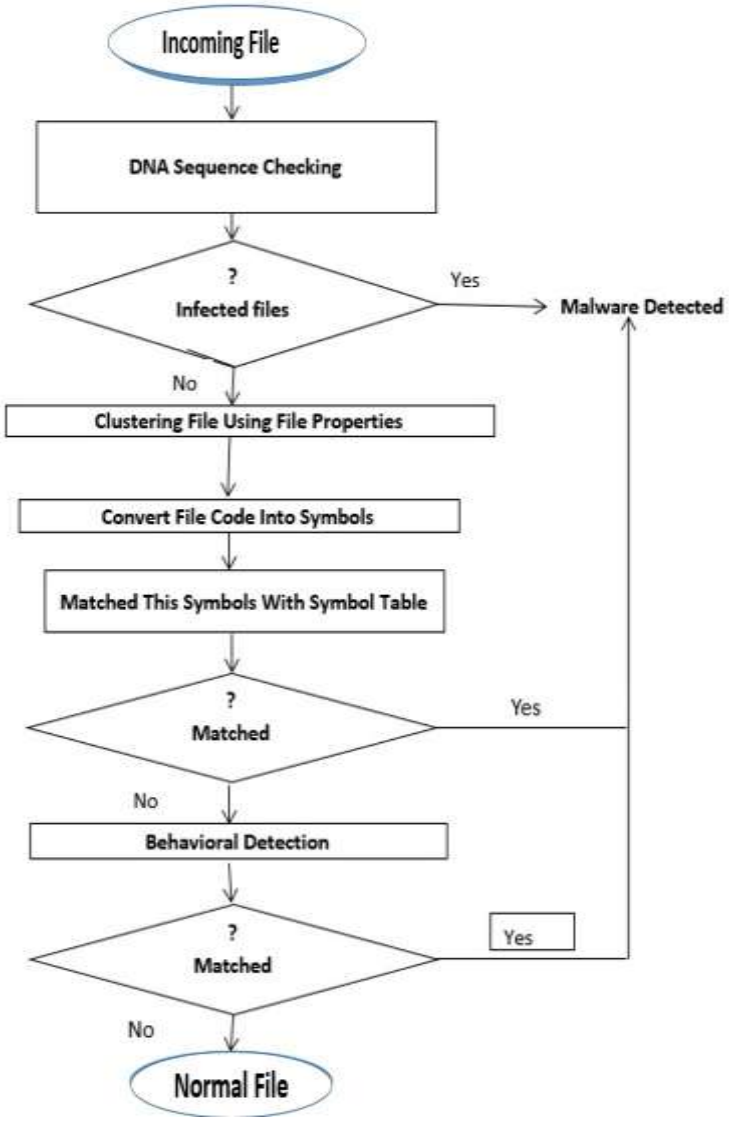

Figure 3. Flow Chart for Detection Method

Our proposed flowchart we basically create 2 step protections where first step is registry based checking and second step is behavior based checking. Number of files enter in our system to check first come to DNA sequence checking step where files DNA sequence are created and check with the system database if matched the file is malicious file and it is blocked by system and the rest of files move to the next step keeping on mind that the file is a new malicious file. Next step is Symbol checking before that file are clustered and then converted to the symbol and then that symbol check with symbol database if matched the file is malicious if not files are moved to the next step where the

behavior of the files are checked. If matched then it is an malicious file and files are blocked if not file are normal file.

In Process 1 we go for DNA Sequence checking, the initial step is the extraction of DNA sequence from a file is done by converting the file into its binary form and change each two corresponding bits into a DNA sequence character by using [Table II] [2]. The conversion is completely reversible.

Binary Bits	DNA Character
00	T
01	G
10	C
11	A

TABLE II: DNA Sequence Mapping Table

After that Malware_Sequence_Database is created using these steps shown in Figure 4.

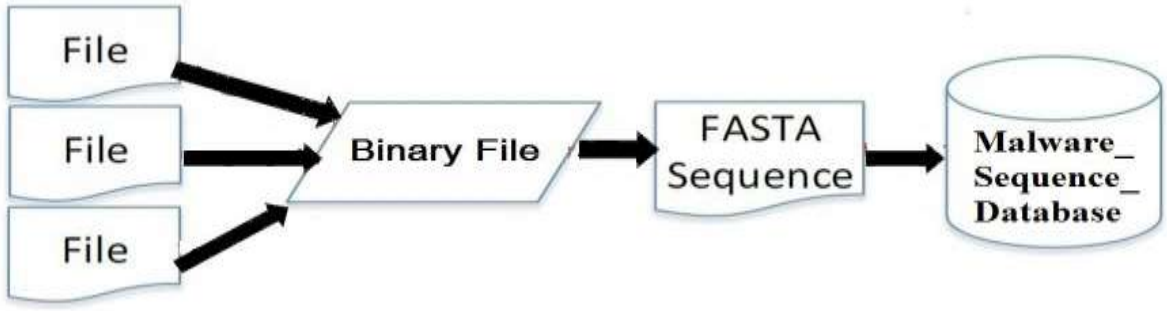

Figure 4: Creating BLAST Database.

The various input files are converted into binary files which are converted into FASTA sequence and are merged into a single FASTA sequence file called Malware_Sequence_Database. These Sequence are shown in Figure 5.

```
>c: /user/name.txt
GTAGGGCCCGTTTGGCCAAAAATTTTTTTT.
```

Figure 5: DNA sequence.

The example of DNA sequence is shown in Figure 5.That starting with ">" sign then the location of file followed by the sequence.

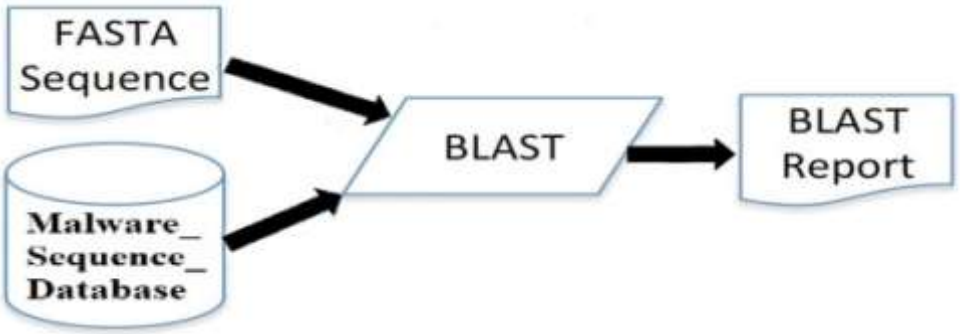

Figure 6: Comparing FASTA Sequence with Malware_Sequence_Database.

Once the Malware_Sequence_Database has been created, we are comparing for the similarities among the FASTA sequence file and Malware_Sequence_Database using Blast online software. The result of this comparison is a BLAST report which will determine that the file is malicious or not see [Figure 6]. If in the BLAST report percentage of identities is less than 80% then the file is not malicious and is pass to the second process for further detection, if the percentage of identities is greater 80% and less than 90% then the file may or may not contain malicious code so it is pass to the second process for further detection otherwise the percentage is greater than 90% which is malicious file and blocked for the second process.

In Process 2 the files which are pass through the first process are only go for the second process. In this process first we cluster the files according to their file format, by only checking the file format see [Figure 7]

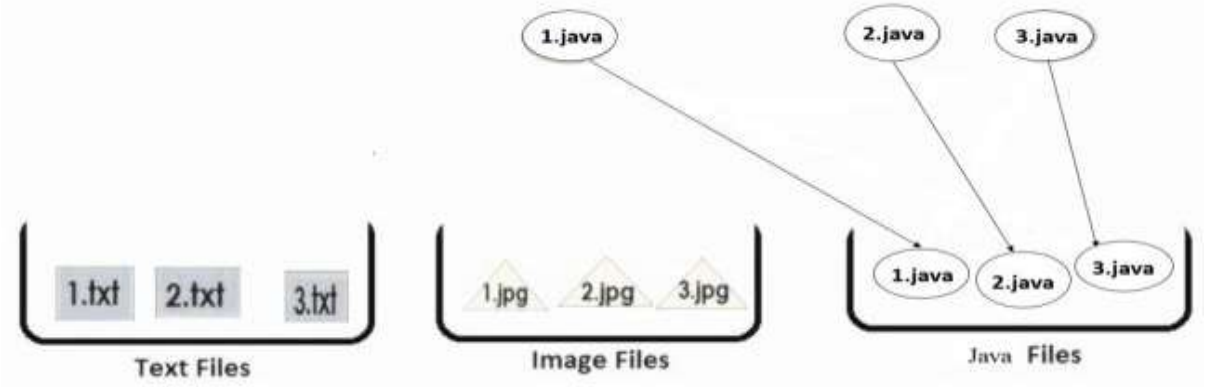

Figure 7: Clustering File Formats.

After that we use symbolic detection technique in which the files are convert to symbol called symbol file. Then after we match symbol file with the existing symbol database table which contain all the symbols of conventional malware signatures, see [Table III].

SL No.	String	Symbol
1.	:A	◄
2.	Start	↕
3.	goto	§
4.	explorer	‼
5.	shutdown	¿
6.	taskkill	×
7.	md	≡
8.	>nul	Ñ
9.	off	Ö
10.	-m	…

TABLE III: Symbol Database Table

If the file symbols are not matched with the existing database table then the file may be malicious or may not be. Otherwise the file symbols are matched with the existing database table and we can say that the file is malicious and blocked for the third process.

In Process 3 the files which are pass through the second process are only go for the third process, which is behavioural malware detection. In this process we detect malicious files using a virtual machine that extensively used for this type of analysis by testing and running the file into a sandbox gives an optimal result to detect malware. For this purpose we use Anubis sandbox [7] which is free available. Anubis interact with file using API call and check the behaviour of the file to identify whether it contain malware or not.

Part II

3.2. Cloud Deployment Model

The Proposed Malware Detection Model (PMDM) discuss in Part-I is deploy into cloud architecture i.e. Cloud Deployment Model (CDM) by using a free open source computer software Eucalyptus. The purpose of this CDM is to implement PMDM in real cloud environment. In this experiment we determine the fitness of the proposal into the Eucalyptus architecture. PMDM is valuate against known and unknown malicious attacks. Here the PMDM system is partially implemented in the Eucalyptus architecture due to the architectural (infrastructure) limitations. The PMDM is mainly used to detect the malicious code based on the above discussed processes. And keep them as a set of warnings in a dedicated thread storage pool and block the malicious file to enter into a Guest VM or Guest Operation System see [Figure 8] [4] [5] [6].

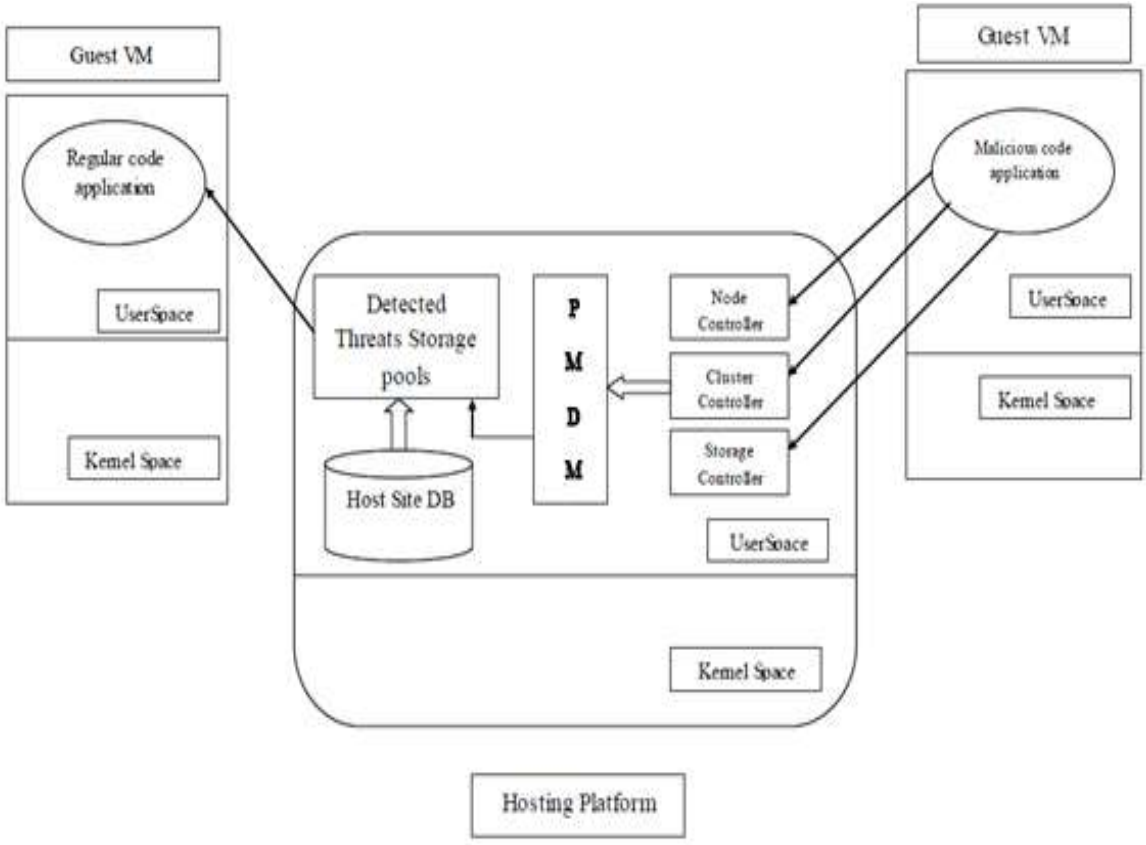

Figure 8: Proposed Malware Detection Method (PMDM) Embedded In Cloud Deployment Model (CDM).

CHAPTER 4. EXPERIMENTAL RESULTS AND ANALYSIS

4.1 Experimental Setup

Our approach of moving the detection of malicious software into the cloud is aligned with a strong trend toward moving services from end host servers into the cloud. Fast forward to today a malware is widespread, sophisticated and increasingly evasive while security technologies are struggling to keep up. At the same time, more and more people and things are connected to the Internet every day. In addition, there have been several attempts to provide cloud services for malware detection so it is similar kind of contributions which is introduced by our proposed approach.

4.2 Performance Evaluation Metric

4.2.1. Comparison Between Traditional Malware Detection Vs Proposed Malware Detection.

Features	Traditional Malware Detection Method	Proposed Malware Detection Method
Effected by security attacks	More	Less
Process time	More	Less
Implementation	Expensive	Cheap
For large number of files	Decrease Performance	Increase Performance
Attackers Intrusion	Easy	Difficult
Cloud solution	May or may not be available	Available

Table IV: Comparison between Traditional Malware Detection Vs Proposed Malware Detection.

4.2.2. Advantages

4.2.2. I. Advantage of DNA Sequencing

The benefit of using DNA Sequencing detection method is that we can detect malware without opening the file i.e. by matching only the DNA sequence of a file form a database, we can say that a file is malicious or not. It will save time because we not need to see the whole file content to detect a malware.

4.2.2. II. Advantage of Symbolic Detection Process.

In symbolic detection first we cluster the file which give a benefit of Post infection protection i.e. we actually know which portion of file content we have to see exactly for malware detection. After that matching the converted symbol of file with symbol table database, will increase time efficiency as we not required to see for the whole malware signature matching only a small part of traditional malware signature symbol is sufficient to detect malware [10] [11].

4.2.2. III. Advantage of Behavioral Detection Process

In Behavioural malware detection solve the problem of cannot cope with malware variants i.e. malware can change their code and compiler setting to bypass the detection or Zero day protection problem i.e. once a new malware is produced its signature or symbol is unknown, so by testing the malicious file into a sandbox we can say that it is malicious or not. The overall advantage of the system is that it increases the efficiency and effectiveness for detection of malwares [10] [11].

4.3 Results

4.3.1. *Result of DNA Sequence Process.*

Several experiments were designed to evaluate the usefulness of this process including document gathering, modification of DNA sequences, database creation, and software identification.

4.3.1. I. Document Gathering

A set of 1020 files of five different types were collected to test the DNA Sequence process. The files included text, image, binary, java and executable files. The types and counts of file were recorded in [Table V].

186 Text files	220 Executable Files
169 Java Files	99 Binary Files
152 Image Files	194 HTML Files

TABLE V: File Counts

4.3.1. II. Modify DNA Sequence

Converting the file into its binary form and change each two corresponding bits into a DNA sequence character see [Table II].

4.3.1. III. Database and software

Merged FASTA sequence file create database see [Figure 4].

The result which comes out from Process 1 is obtain by using BLAST software. We analyse the above files and observe the result one of the result output is given below:

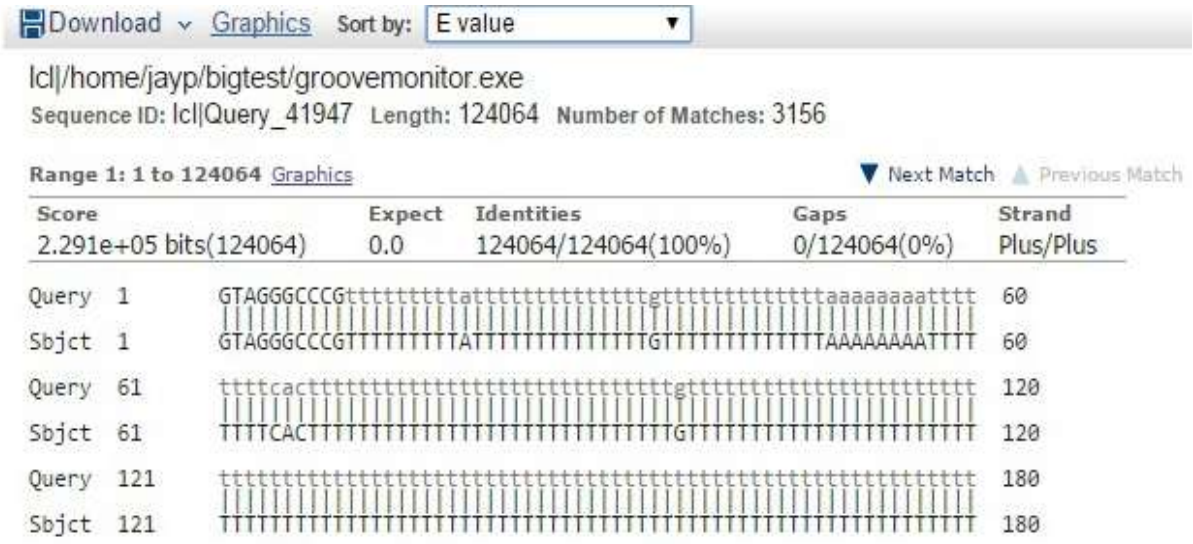

Figure 9 (a). Descriptive Analysis Result of groovemonitor.exe

Groovemonitor.exe malware on BLAST and found a 100% identities matched i.e. it is a malware, see results in [Figure 9(a)] and [Figure 9(b)]. That means the system found file as groovemonitor.exe and block then file as soon as it detected.

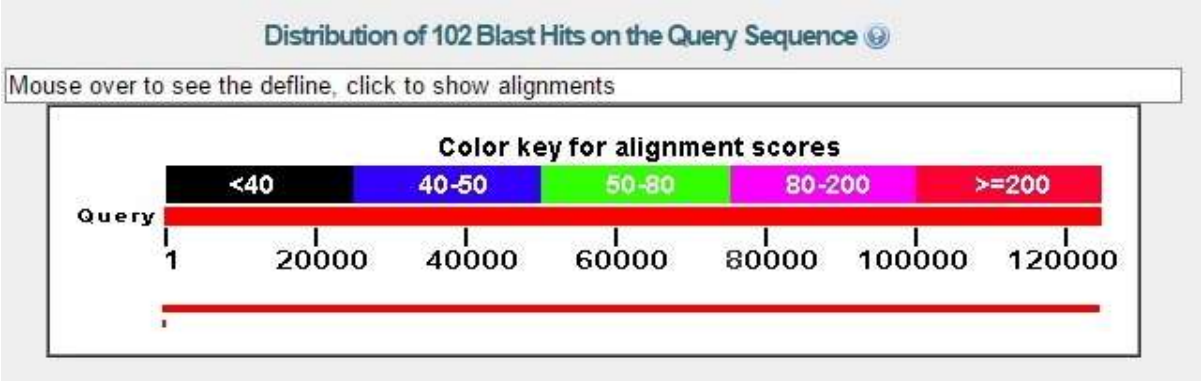

Figure 9 (b): Graphical Analysis Result of groovemonitor.exe

This shown the graphical representation of the result the graphic show how the file dna sequence match with the groovemonitor.exe store sequence.

Same Analysis process is applied on an image file which is malware free and it result is below 70% as expected .From the above various files results the groovemonitor.exe file is blocked and the Image File1.bin and other files are pass for the second process.

4.3.2. Result of Symbolic Detection Process

For symbolic detection there is also several operations were designed to evaluate the usefulness of this process including file clustering, converting characters into symbols and matching symbol with symbol table database.

4.3.2. I. File Clustering.

In file clustering a set of different types were taken and group them into a single group according to their file format see [Figure 7].

4.3.2. II. Converting File Characters into Symbols

The conversion of file characters into symbol is done by using [Table III].

4.3.2. III. Matching Symbol with Symbol Table Database.

The matching of symbols with symbol table is done by using the following pseudocode.

```
Take symbolfile name as Input
Scan symbolfile
While not EOF do
        If symbolfile match with virus symbol Then
                Print "File contain virus"
        Else
                Print "File does contain virus"
        Endif
End While
```

Figure 10: Pseudocode for Matching Symbol with Symbol Table Database.

Above is the pseudocode for matching symbol code built in java. First we take file as input that already converted to its corresponding symbol by using the symbol table that we have using another java program and match the file symbol with our database if match the file is a

virus file if not it may be a new virus file and it sent to the last process of our proposed work that is a behavior detection process.

From the above various files match is found in the symbol table, as example here symbolvsample.txt file and it is blocked after that and other files that not matched are pass for the third process.

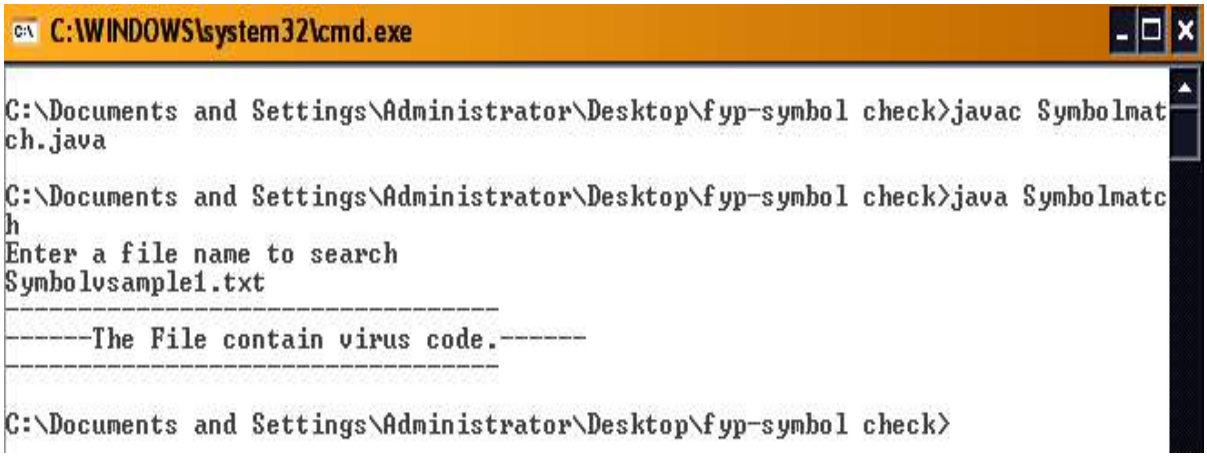

Figure 11: Symbolic Detection Result of symbolvsample.txt

4.3.3. Result of Behavioral Detection Process

The files which pass the second process are only entertain in this process like from the above result all the files are entertained excluding groovemonitor.exe and symbolvsample.txt files. All the files are the input to the Anubis sandbox after that Anubis will check the files using API call and also check the behaviour of the file to identify whether it contain malware or not.

CHAPTER 5. CONCLUSIONS

5.1 Summary

It has proposed an effective and advanced cloud security method that can detect the different malware attacks during cloud communication. It is also partially implemented on a popular architecture of Eucalyptus with modified version i.e. Cloud Deployment Model (CDM). This documentation discusses the basic techniques in brief needed for the development of Proposed Malware Detection Model (PMDM).This technique is actually cheap, requires less processing time and provides good performance for large numbers of files compare to other traditional malware detection systems. It is totally transparent to the user. We used the both optimal traditional detection methods and modern era methods to detect malwares, like signature based detection (traditional) and DNA Sequencing method (modern era).The proposal of this work is to find the best solutions to the problems of anti-malwares and improve performance and find possible alternatives for a better working environment without problems with high efficiency and flexibility.

5.2 Limitations & Future Work

In future, we will see an increase in the dependence of cloud computing. Cloud technologies have become possible because of shearing physical server resources between multiple virtual machines (VMs). The advantages of this approach include an increase in the number of clients that can be served for every physical server.

REFERENCES

[1] G. E. Dahl, J. W. Stokes et al., "Large-scale malware classification using random projections and neural networks", 2013 IEEE International Conference, pp. 3422 - 3426 , 31 May 2013.

[2] Jay Pedersen, Dhundy Bastola, et al., "BLAST Your Way through Malware Malware Analysis Assisted by Bioinformatics Tools", International Conference on Security and Management 2012, 2011.

[3] Safaa Salam Hatem, Dr. Maged H. wafy, et al., "Malware Detection in Cloud Computing", International Journal of Advanced Computer Science and Applications (IJACSA), vol 5, Science and Information, 2014.

[4] Dan C. Marinescu, "Cloud Computing: Theory and Practice", MK Publication, 2013.

[5] Johnson D, Kiran Murari, et al., "Eucalyptus Beginner's Guide- UEC Edition", v1.0, 25 May 2010.

[6]Hiren Parmar, L. D. C. E, GTU, et al., "Comparative Study of Open Nebula, Eucalyptus, Open Stack and Cloud Stack".

[7]Thomas Mandl, Ulrich Bayer, et al., "ANUBIS ANalyzing Unknown BInarieS The automatic Way", Virus Bulletin Conference 2009.

[8] Jignesh Vania, Arvind Meniya, et al., "A Review on Botnet and Detection Technique", International Journal of Computer Trends and Technology, vol-4, Seventh Sense Research Group, 2013.

[9] Jon Oberheide, Evan Cooke, et al., "CloudAV: N-Version Antivirus in the Network Cloud", 17th conference on Security symposium, pp- 91-106.

[10] Mark Graham, “Behaviour of Botnets and Other Malware in Virtual Environments”, The Open Web Application Security Project 2014.

[11]https://www.youtube.com/watch?v=fV5kED7nry

List of Publication

Sagar Shaw, Manish Kumar Gupta, Sanjay Chakraborty, “Cloud Based Malware Detection Technique”, 5th International Conference on Frontiers of Intelligent Computing Theory and Applications (FICTA), Springer 2016. (Communicated).