BEHAVIOR BASED MALWARE DETECTION SYSTEM FOR CORPORATE E-MAIL TRAFFIC.

(E-Secure)

17-002

Design Document

Supervisor-Amila Nuwan Senarathne

Co-supervisor - Nuwan Kuruwitaarachchi

Author: S.Ashok

Bachelor of Science (Honors) in Information Technology Specialized in Computer System and Network Engineering

Department of Information System Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

May 2017

BEHAVIOR BASED MALWARE DETECTION SYSTEM FOR CORPORATE E-MAIL TRAFFIC.

(E-Secure)

17-002

Design Document

(Design document submitted in partial fulfilment of the requirement for the Degree of Bachelor of Science Special (honors) in information Technology)

Bachelor of Science (Honors) in Information Technology Specialized in Computer System and Network Engineering

Department of Information System Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

May 2017

Declaration

I Ashok Selvakumar (IT14502484) hereby declare that this Design Document entitled E-Secure submitted by me, under the supervision of Mr. Amila Nuwan Senarathne and co-supervised by Mr. Nuwan Kuruwitaarachchi of Sri Lanka Institute of Information Technology is my own work and has not been submitted to any other University or Institute or published earlier.

Student ID	Name	Signature	Date
IT14502484	S.Ashok		

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Purpose

The Design Document (DD) is a document to provide documentation which will be used to aid in system development by providing the details for how the system should be built, tools and technologies, research study and development, functional and nonfunctional requirements of the system. Within the DD are narrative and graphical documentation of the system design for the project including use case models, collaboration models, object behavior models, and other supporting information.

This document will include most of the requirements in developing the proposed system, also a variety of interfaces such as user interfaces, software interfaces, hardware interfaces, communication interfaces and system interfaces, constraints and limitations under which the proposed system must be operated, functionality flow, performance related requirements and other system attributes involved in developing the system.

Interested set of audiences to this document are, supervisor, co-supervisor, project coordinators, developers, testers and end users.

1.2 Scope

This document contains high-level detailed description of the requirements gathered for implementing E-Secure system, which will enable detection of malwares. The scope of this DD document is bounded only to describe the functionalities embedded under detection procedure, tools and techniques to be used and relevant technologies referred for the implementation of the system. The overall system is developed that has a widespread of audience such as cyber security analysts, network engineers and etc. The system is developed as a solution for malware attacks that are existing in current cyber world. The main functionalities of the proposed system would contain are extraction of attachments from the e-mail traffic. This should be done for the e-mail traffic that comes through the corporate network. Attachments should be separated such a way that it can be send to the virtual environment for the analyzation. The other main functionality is identification of virus using ML. Identification is done by feeding the behaviors of virus into the system and then comparing the results from the report with already fed behavior.

1.3 Definitions, Acronyms and Abbreviations

Terms	Definitions
DD	Design Document
ML	Machine Learning
PHP	PHP Hypertext Processor
OS	Operating System

Table 1

1.4 References

[1] Ivan Firdausi, Charles Lim, Alva Erwin, Anto Satriyo Nugroho, Analysis of machine learning techniques used in behavior based malware detection,2010

[2] Cesar Augusto Borges de Andrade, Claudio Gomes De Mello, Julio Cesar Duarte, Malware Automatic Analysis,2013

[3] Radu S. Pirscoveanu, Steven S. Hansen, Thor M. T. Larsen, Matija Stevanovic, Jens Myrup Pedersen, Alexandre Czech, Analysis of Malware Behavior: Type Classification using Machine Learning, Feb 2015

[4] Arshi Dhammi, Maninder Singh, Behavior Analysis of Malware Using Machine Learning, 2015

[5] Harry Kurniawan, Yusep Rosmansyah, Budiman Dabarsyah, Android Anomaly Detection System Using Machine Learning Classification, Aug 2015

[6] Misha Mehra, Dhawal Pandey, Event Triggered Malware: A New Challenge to Sandboxing, 2015

[7] Chia Tien Dan Lo, Ordonez Pablo, Cepeda Mora Carlos, Towards an effective and efficient malware detection system, 2016

[8] LIU Wu, REN Ping, LIU Ke, LI Xing, WU Jian-ping , LIU Ke, Analysis and Forensics for Behavior Characteristics of Malware in Internet, 2016

[9] Nidal Zeidat, Christoph F. Eick, and Zhenghong Zhao, Supervised Clustering: Algorithms and applications, June 28, 2006

[10] Anthony McGregor, Mark Hall, Perry Lorier, and James Brunskill, Flow Clustering Using Machine Learning Techniques

[11] Konrad Rieck, Philipp Trinius, Carsten Willems, and Thorsten Holz, Automatic Analysis of Malware Behavior using Machine Learning. This is a preprint of an 19 article published in the Journal of Computer Security, 2010.

[12] Taiwo Oladipupo Ayodele, Types of Machine Learning Algorithms, 2010

[13] Nikolaos Papachristou, Christine Miaskowski, Payam Barnaghi, Roma Maguire, Nazli Farajidavar, Bruce Cooper and Xiao Hu, Comparing Machine Learning Clustering with Latent Class Analysis on Cancer Symptoms' Data, 2011 Jan

[14] Kateryna Chumachenko, Machine Learning methods for malware detection and classification, Bachelor's Thesis Information Technology, 2017

1.5 Overview

Global communication has become an important trend in the current society and usage of Internet has grown exponentially in the modern society. At the same time, criminal activities have been increased with the help of Internet where a large black market has been created by hackers in the intent of producing money or to destroy others wealth and name. There are many security threats around the world and malwares are one type among them. Malwares are software which is typically developed to disrupt, damage and gain authorized access to a computer system. There are many types of malwares currently prevailing in the world which are considered as dangerous to the cyber world. Virus, Worms, Trojan, Rootkits, Spyware, Adware, Backdoors, Sniffers and Reverse Code are some examples of malwares.

The goal of E-secure is to facilitate in securing email communication for small business originations. E-mail service are very common way to spread malware into the organization. So nowadays the intruders use e-mail and a very advanced technique to create and hide the malware. Mostly many malwares are attached with email as scripts even sometimes bound with useful documents and pdf files. With the features provide by the E secure, through every business organization email attachment will be scanned and analyzed as well as notified to the security administrator if any malware found.

This DD contains two main functions, The first section contains an overall description about the system which specifies an overview of the entire system while providing sample background information of the entire system, the design and functioning environments, the users of the system, constraints that affect the design and implementation of the system, assumptions and dependencies for execution of the system. Hence this chapter provides the reader with a comprehensive viewpoint of what the functionality and behavior of the system. The second section contains System interfaces, User interfaces, Hardware interfaces, Software interfaces, Communication interfaces, Memory constraints, Operations, Site adaptation requirements, Product functions, User characteristics, Constraints, Assumptions and dependencies, Apportioning of requirements

Finally References and Appendices will be in this document. References can be used to find more details about relevant sections and for maintenance purposes of the system these

details can be used to get more knowledge about the technologies and algorithms that are going to use in the development of system.

1.6 Document Organization

Introduction	Provides information related to this document (e.g. purpose, term definitions etc.)
Overall description	Describes the approach, architectural goals and constraints, Guiding principles, software interfaces, hardware interfaces and communication interfaces, constraints and operations.
Specific requirements	Describes the various system requirements of the system.
Supporting Information	Table of contents, appendix, list of figures, list of tables.

Table 2

2. Overall Description

Malwares are software which is typically developed to disrupt, damage and gain authorized access to a computer system. There are many types of malwares currently prevailing in the world which are considered as dangerous to the cyber world. Virus, Worms, Trojan, Rootkits, Spyware, Adware, Backdoors, Sniffers and Reverse Code are some examples of malwares.

One of the main functionality of this system is to detect the Virus malware. Virus is a program written for enter user's computers/system to damage or alter the files/data. Virus can replicate them self. Virus can come to the endpoint with the attachment of images, files, audios, videos. Virus can't spread without human help. After virus came to the system/computer it can change the Hard Disk size, boot up the system memory, damage the system files, can encrypt user's data. There some different type of virus.

1. File virus: - This type of virus infects the .exe and .bat files

2. Macro Virus: - This type of virus infects word, excel, PowerPoint, access and other data files

3. Master boot record files: - This type of virus infects particular area of storage device instead of normal files

4. Multipartite Virus: - This type of virus infects the program files. When program file start from the system/computer virus boot from the system

5. Polymorphic Virus: - This type of virus creates encrypt it code in different ways, it is difficult to identify and delete it.

So, the system contains main functions. Two functions are covered in this DD.

Removal of Attachment in an E-mail

For the identification of malwares, first the same type of malwares should be clustered. These clustering should be done in sandboxing environment. These sandboxing environment is implemented in a virtual environment. Since the system is fully associated with the malwares, prevention of host machine is important when developing the system. Due to this reason sandboxing environment is used, this environment separates the host OS and virtual machine OS. So, no harm will happen to the host machine, if anything happens in the virtual machine. But the problem is, sandboxing environment will analyze the files one by one. So, the files should be send one by one. So, the attachments that come from e-mail should be separated and stored in a database to send into the sandboxing environment one by one. These removals of attachments will be done by configuring mail server or writing script.

Identification of Virus

The system will be already fed with the behavior of Virus malware attacks. If the cluster of files with same behavior is matched with the behavior of malwares which is already fed into the system, then the type can be easily identified. Cuckoo sandbox will give report of every attachment that come through the e-mail. These reports will be clustered based on the same

behaviors or characters. So, main task of this functionality is to identify the cluster of reports which is affected by the virus. This is done using an algorithm in ML.

2.1 Product Perspective

Typically, all malware security items depend on the signatures and hash esteem. Current malware security items are with the capacities of recognizing the malwares, yet we can't guarantee alternate records which are not distinguished by the malware security item as an uninfected document.

If a malicious record which contains the malignant signature which does not put away in the unified signature database, that is going to be a huge problem. This turned into the real issue in the malware examination. Fundamentally, the current malware security items can't recognize the new kind of malware assault, since its signature is not contained in the centralized database.

As such, ordinarily anti-virus software utilizes signatures to recognize the malware. It has some arrangement of signatures which are characterized as the malware in the concentrated database. Amid the scan of a document if a signature is coordinated with the signature in the database that specific record is blocked. For instance, most intensely utilized antivirus programming, vigorously relies on upon the infection signatures to identify the malware. Once a suspicious code is compasses an antivirus merchant, those codes are dissected by the digital security analysts and once the code is delegated infection, an exceptional signature of the document is removed and included to the product database of antivirus programming recognition. These malwares additionally scramble or change their code parts without anyone else's input as a system of mask, so they don't coordinate with any signatures.

Signature based email security arrangement items keep up a centralized signature database. This database is fabricated in view of the past assaults. By that way, this arrangement work in a cycle, for instance if new malware is made and sent through an email, the email server will permit this email and that malware will harm the host, then the operator running on the end point will illuminate the security arrangement that this document comprise of malware. And afterward the signature of the malware is made and refreshed to all world servers that are running this security arrangement. Along these lines, zero-day malware assault

identification is outlandish in this situation. What's more, the progressed malwares constantly change their signature, so it can't recognize from common malware location.

Malware can shroud its identity yet not its behavior. In this way, our security arrangement essentially concentrates on the behavior examination of malware. Our item get each email attachment and run it on the virtual condition and analyze the behavior. From that report utilizing machine learning we decide if the document is influenced by malware or not. Along these lines, no approach to zero-day assault even most progress malware assault.

Current market products are not affordable by smaller corporate networks. Security solutions must be less than the value of damage caused by the malware. So simply security solutions must be open sourced and should be affordable by every corporate companies. In a way that our solution will be more helpful.

2.1.1 System Interfaces

The function mainly cover the back end of the system. This section mentions the system interfaces that system going to be used.

Linux Environment with CentOS 7.

The above system interfaces are used to feed the behavior of virus into the system

Zimbra mail server in the Linux environment is used to separate the attachments from the e-mail. Configurations of the mail server are done in Linux environment.

2.1.2 User Interfaces

The function mentioned in the DD is an internal processing, so this function will not be going to have any user interfaces.

2.1.3 Hardware Interfaces

Personal Computers – Personal computers with Linux environment will be the main hardware Interface. CentOS is used as the operating System

2.1.4 Software Interfaces

PHP Storm – Python language is used in PHP Storm to feed the behaviors of the Virus.

CentOS: Entire project development will be done in CentOS based PCs. Server can be configured in CentOS based PCs.

Documentation related work: Microsoft Office 2013 will be used for creating of project documents. Diagramming tool Draw.io software for creating diagrams.

PHP Storm to run Python (Machine Learning)

IDLE to test the Machine Learning algorithms

2.1.5 Communication Interfaces

Wi-Fi or access to internet connection: Provides internet facility to the application. With this switches, hubs and Ethernet cables are used for the network connectivity.

2.1.6 Memory Constraints

The main memory constraints in the system is the memory consumption that takes to process all the behaviors of the virus. The input of the system is going to be the overall behaviors of virus. So, the memory should be capable of processing every behavior when classifying the clusters. And since the system is working with malwares the system is ran in a virtual environment. So normally virtual environment will be consuming some memory.

2.1.7 Operations

Security analyst will be the main user. If any further modification is required, all those modifications will be done by him.

2.1.8 Site Adaptation Requirements

- A computer with CentOS
- Recommended RAM 4GB
- Processor i3 2GHZ
- Zimbra Mail Server

2.2 Product Functions

In recent times attackers have found new ways to bypass the technologies that are used in the cyber world. Malware possess a serious threat to the cyber security world these days. And virus is one of the serious type malware which is a large threat to the cyber world. These viruses can enter the computer as an attachment and then can delete or corrupt the data in the computer. Machine learning is used to analyze and identify the virus based on its behavior. In this component in addition to virus analysis, files which come as an e-mail attachment is taken and send into the sandboxing environment for the report production. The main outcome of this component is to analyze and identify Virus based on its behavior analysis and to get the files from e-mail attachment that is to be send into the sandboxing environment.

Initial function of the system is to extract the attachment from the e-mail.

2.2.1 Use Case Scenario

Use case name	Separate attachment
Actor	Mail server
Description	Separate the attachment from the mail to check whether that file contain malware or not
Main Flow	1. System will check income mail 2. Separate the attachment 3. Send that files to sandboxing virtual averment
Alternate flow	2.a Using mirroring technique separate the attachment

Table 3

Use case name	Identify the Virus malware
Actor	System
Description	Help the user, whether file contain which type of malware
Main Flow	4. Get the analyzed or analyzed clustered report from sandboxing 5. Identify the Worms malware type 6. Inform the user/admin
Alternate flow	2.b If the user upload file through website user will get the details in the website 2.c If it is mail attachment, the details will be store in the database and later admin will notify through the dashboard

Table 4

2.2.2 Use Case Diagram

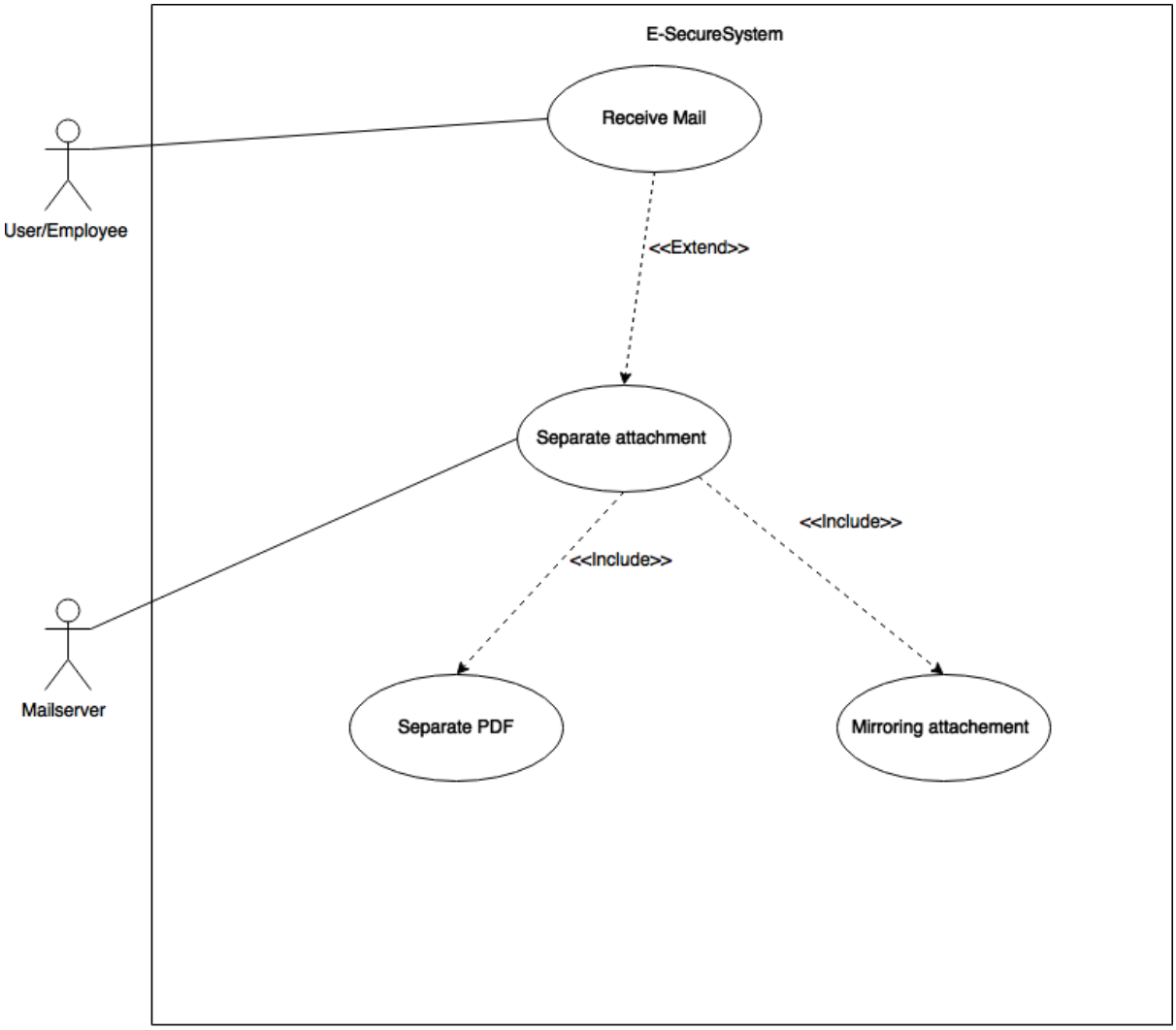

Figure 1

The above use case diagram depicts how an attachment is removed from the e-mail.

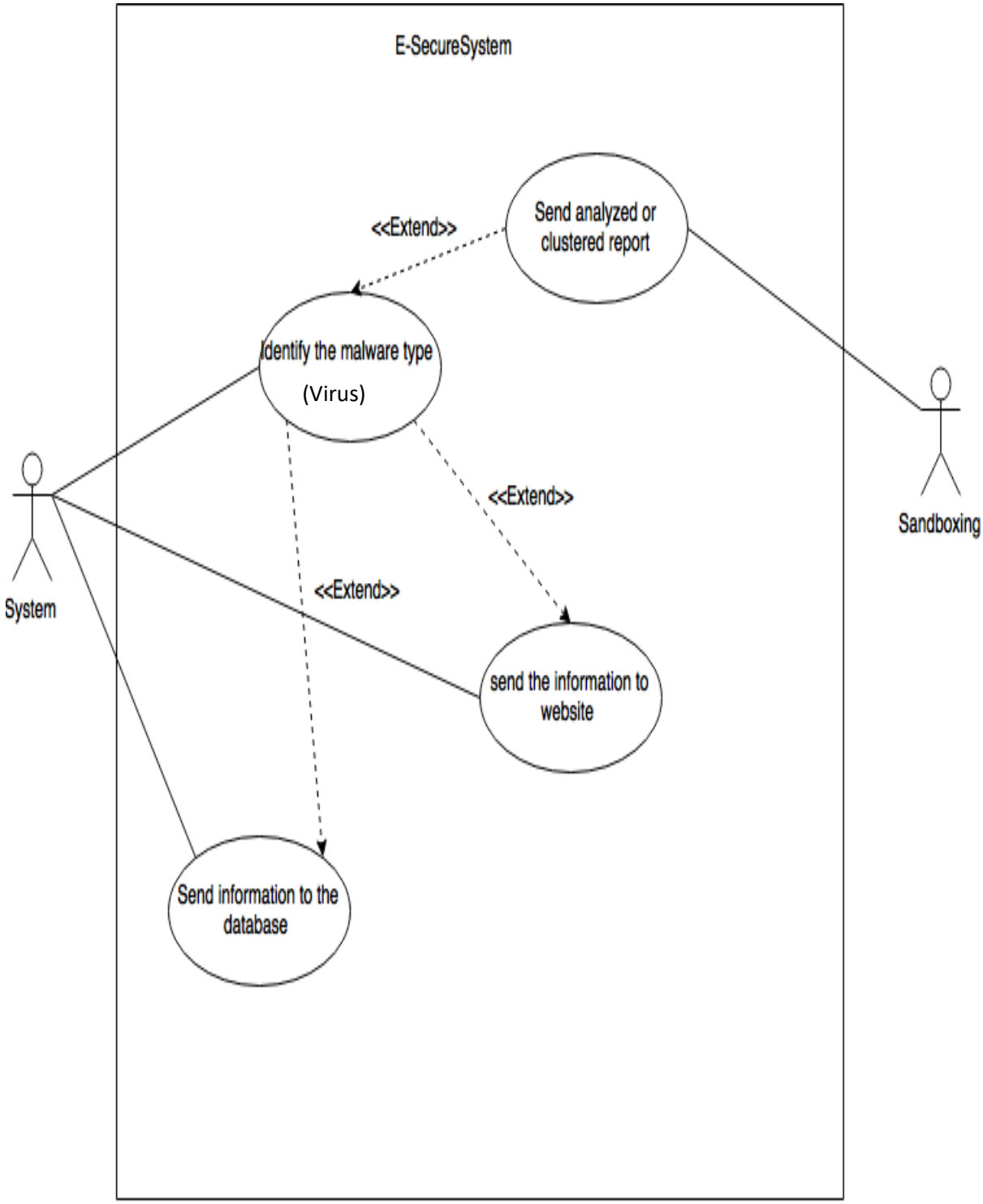
E-SecureSystem
Send analyzed or clustered report
<<Extend>>
Identify the malware type
(Virus)
Sandboxing
<<Extend>>
System
<<Extend>>
send the information to website
Send information to the database

Figure 2

2.3 User Characteristics

User of the system will be security engineers or security analysts who have the full knowledge about the cyber security.

2.4 Constraints

- Testing the system
  - Files with Virus must be created to check the system
- Getting the unique behaviors of the Virus and feeding into the system
  - All behaviors cannot be fed into the system considering the amount of time and memory. So only major behaviors will be fed into the system

2.5 Assumptions and dependencies

- If a malware is to be identified minimum 60% should be matched with its behaviors that are fed into the system.
- There is no delay in getting the detailed reports after the malware analysis.
- Only certain common behaviors are fed into the system.

2.6 Apportioning of requirements

E-Secure system is a well-planned product which is developed by analyzing the current problem faced in the cyber world. There is no specific client. The product is not customized based on any user requirements. The methodology of implementing the system may slightly different from the content described in this document. During system designing, requirements specified will not be changed and the system released will totally contains its purposes and objectives.

3 Specific requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The function mentioned in the DD is an internal processing, so this function will not be going to have any user interfaces.

3.1.2 Hardware Interfaces

Personal Computers – Personal computers with Linux environment will be the main hardware Interface. CentOS 7 is used as the operating System

- Recommended RAM 4GB
- Processor i5 2.7GHZ
- Zimbra Mail Server in Linux Environment

3.1.3 Software Interfaces

PHP Storm – Python language is used in PHP Storm to feed the behaviors of the Virus.

CentOS 7: Entire project development will be done in CentOS 7 based PCs. Server can be configured in CentOS 7 based PCs.

Documentation related work: Microsoft Office 2013 will be used for creating of project documents. Diagramming tool Draw.io software for creating diagrams.

3.1.4 Communication Interfaces

Wi-Fi or access to internet connection: Provides internet facility to the application. With this switches, hubs and Ethernet cables are used for the network connectivity.

3.2 Architectural Design

3.2.1 High level Architectural Design

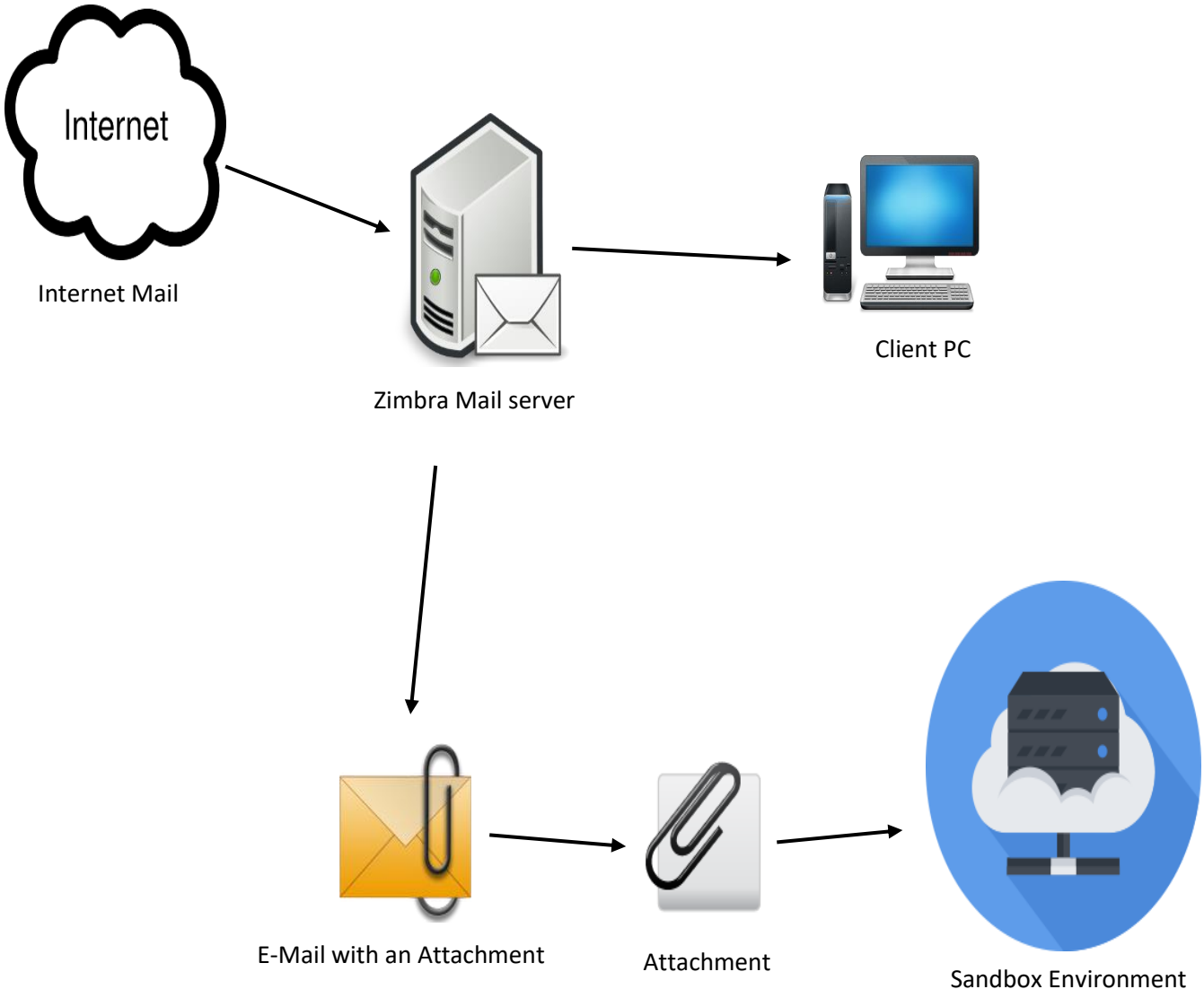

Figure 3

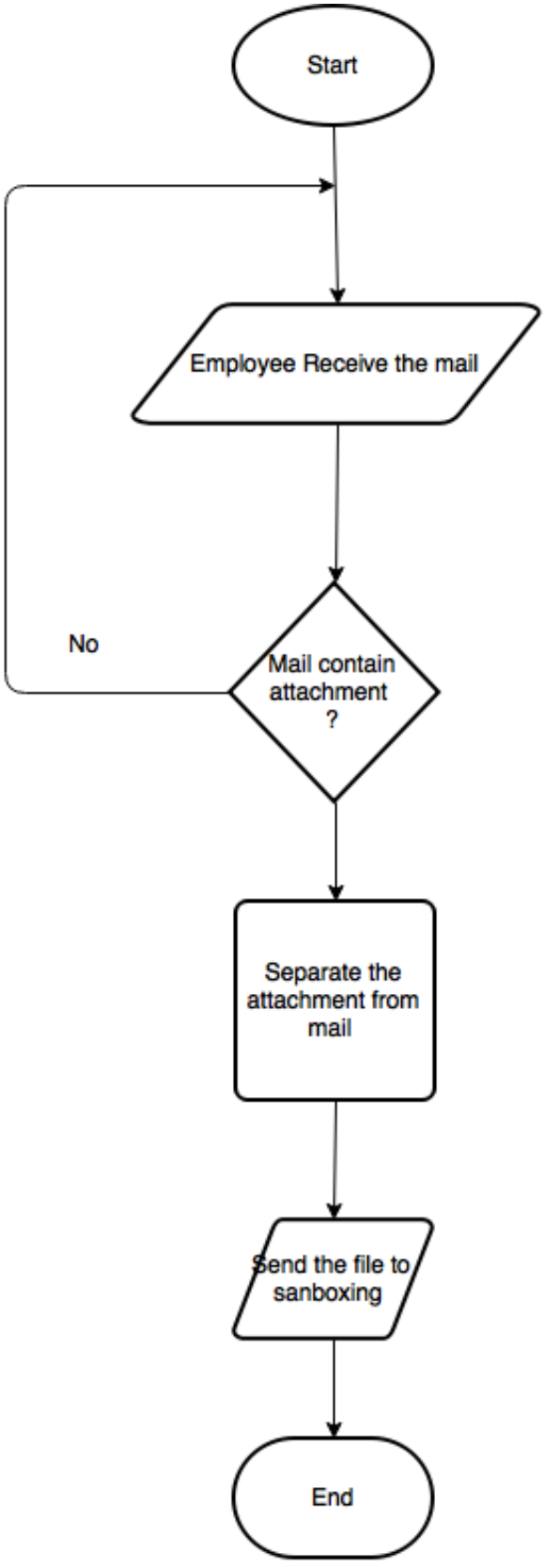
Start
Employee Receive the mail
No
Mail contain attachment ?
Separate the attachment from mail
Send the file to sanboxing
End

Figure 4

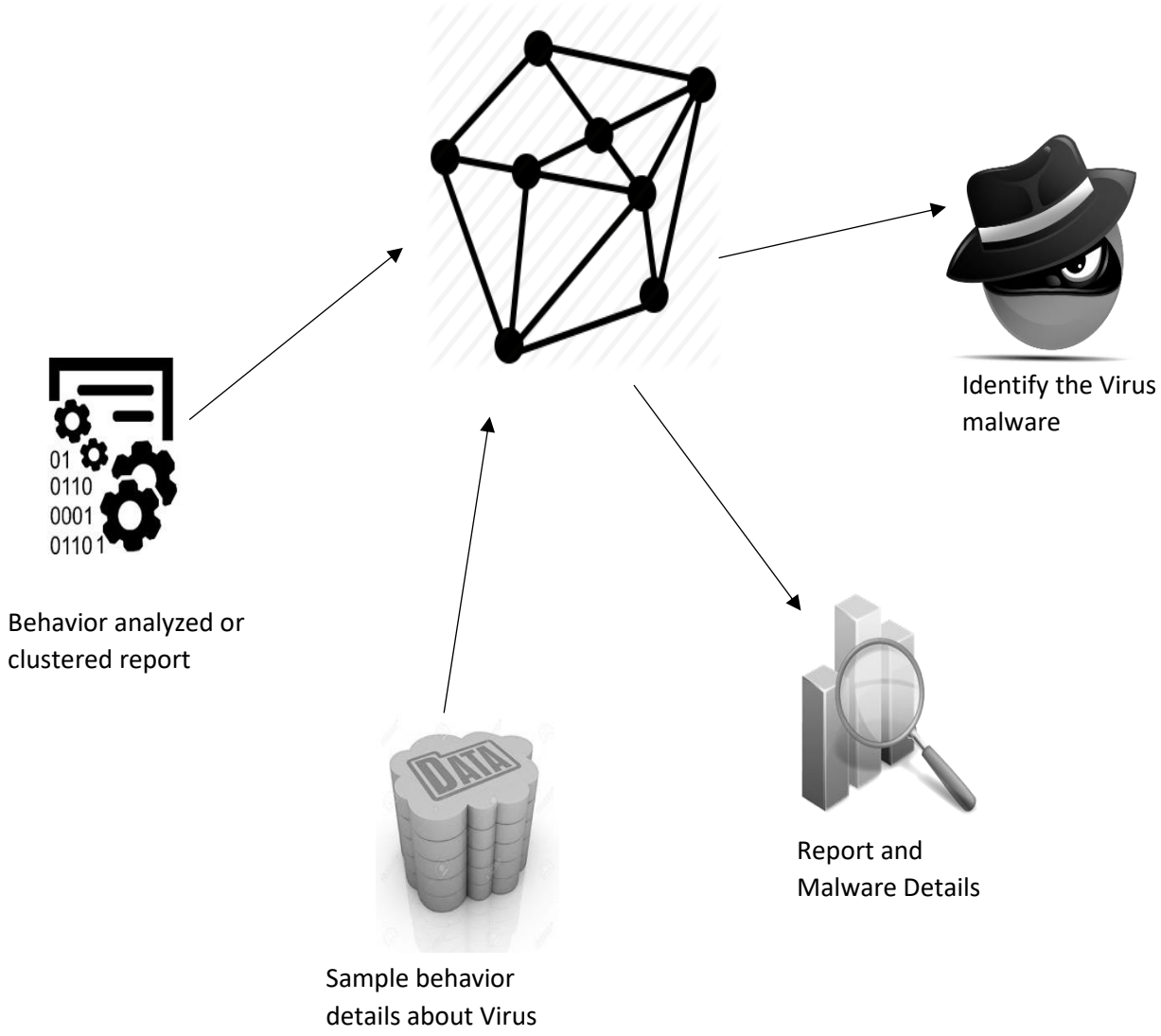
01
0110
0001
0110 1
Behavior analyzed or clustered report
Identify the Virus malware
DATA
Sample behavior details about Virus
Report and Malware Details

Figure 5

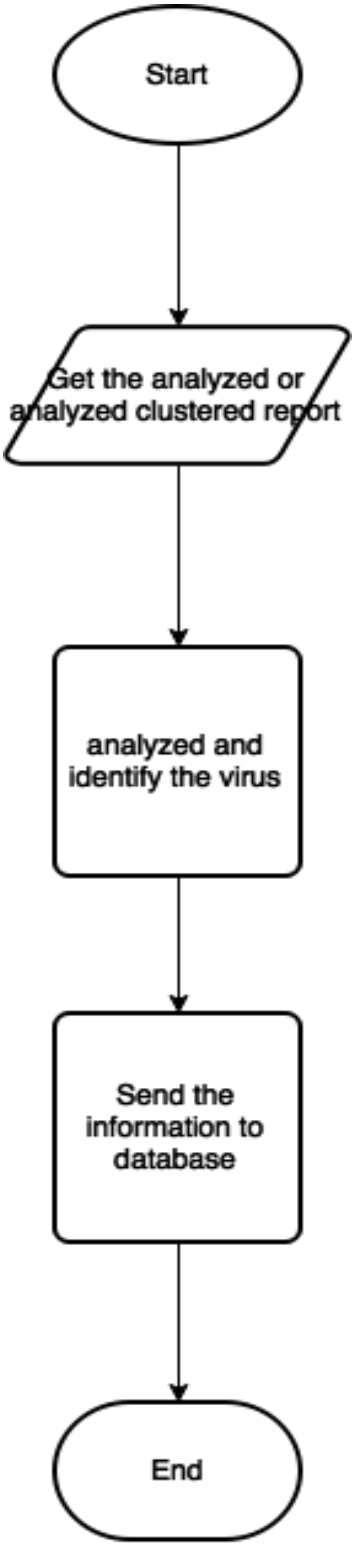
Start
Get the analyzed or analyzed clustered report
analyzed and identify the virus
Send the information to database
End

Figure 6

3.2.2 Hardware and software requirements with justification

Hardware Requirements

Personal Computers – Personal computers with Linux environment CentOS 7 as an operating System

- Recommended RAM 4GB
- Processor i5 2.7GHZ
- 1GB Memory Space

The above requirements are needed since the system is built in a virtual environment, to safeguard the host computer getting affected from the malware attack, since the whole system is associated with the malware.

Software Requirements

- PHP Storm – Python language is used in PHP Storm to feed the behaviors of the Virus.
- CentOS 7: Entire project development will be done in CentOS 7 based PCs. Server can be configured in CentOS 7 based PCs.
- Documentation related work: Microsoft Office 2013 will be used for creating of project documents. Diagramming tool Draw.io software for creating diagrams.

3.2.3 Risk Mitigation Plan with alternative solution identification

Since the system is fully associated with the malwares, prevention of host machine is important when developing the system. Due to this reason sandboxing environment is used, this environment separates the host OS and virtual machine OS. So, no harm will happen to the host machine, if anything happens in the virtual machine.

There is a chance of host computer getting crashed. So, there should be another host computer with same configuration.

3.2.4 Cost Benefit Analysis for the proposed solution

The development of this system was given by the “Cryptogen” company. During the completion of this system there can be some positives for the project group members.

3.3 Performance Requirements

Personal Computers – Personal computers with Linux environment CentOS 7 as an operating System

- Recommended RAM 8GB
- Processor i5 2.7GHZ
- 1GB Memory Space

3.4 Design Constraints

Amid the plan arranges, the real imperative to the advancement group was the constraint in time and accomplishing the breakthroughs. In spite of the fact that the venture is relied upon to be finished inside a year, the time we get is not precisely a year. It's hard to make a framework outline which matches industry guidelines and comprises of appropriate plan designs. The procedure includes heaps of research on the related territories of improvement

3.5 Software System Attributes

3.5.1 Reliability

Reliability plays a major role in this system. To have a system with high performance reliability is a must because it is a real-time application.

3.5.2 Availability

The system is available during rainy or night times. The system would be fault tolerant, all the e-mails should have to pass through the system controller to process and give an output, and in other words system will be able to continue functioning when part of the system fails. Parts of the system are well-designed and will be thoroughly tested before they are used.

3.5.3 Security

In order to support maintainability through possible future expansions to the E-Secure system, proper standards shall be followed throughout the system implementation. System will be implemented to minimize bugs as much as possible. System will have solid security which make system administration much easier.

3.5.4 Maintainability

Maintainability is a difficult task than the implementation, so the support tem have to get full idea to maintain the system, for that each and every task have to be recorded to refer because the maintenance team will be different than the development team.

3.6 Other Requirements

3.6.1 Performance

The system made is made so as to be very accurate in the task and perform in less amount of time. The exact expected outcome is given if the correct details are given to the system as input.