CLOUD-BASED RANSOMWARE DETECTION THROUGH FILE ENTROPY AND INACCESSIBILITY

By

Fazal Saadat Ali Khan

A thesis submitted to the faculty of Information Security Department, Military College of Signals, National University of Sciences and Technology, Islamabad, Pakistan, in partial fulfillment of the requirements for the degree of MS in Information Security

January 2022

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS Thesis written by **Mr. Fazal Saadat Ali Khan** Registration No. **00000330685**, of **Military College of Signals** has been vetted by undersigned, found complete in all respects as per NUST Statutes/ Regulations/ MS Policy, is free of plagiarism, errors, and mistakes, and is accepted as partial fulfillment for the award of MS degree. It is further certified that necessary amendments as pointed out by GEC members and foreign/ local evaluators of the scholar have also been incorporated in the said thesis.

Signature: _____

Name of Supervisor: **Brig Imran Rashid, Phd**

Date: _____

Signature (HOD): _____

Date: _____

Signature (Dean/ Principal) _____

Date: _____

DECLARATION

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and acknowledgments.

Fazal Saadat Ali Khan

January 2022

DEDICATION

This thesis is dedicated to

MY TEACHERS, PARENTS, SIBLINGS, WIFE AND DAUGHTER

for their love, endless support and encouragement

ACKNOWLEDGEMENTS

I am grateful to Allah Almighty for giving me strength to keep going on with this thesis, irrespective of many challenges and troubles. All praises for HIM and HIM alone.

I am very grateful to my Project Supervisor Brig Imran Rashid, PhD, Co-Supervisor Asst Prof Dr Shahzaib Tahir and GEC members who supervised the thesis / research in a very encouraging and helpful manner. They always guided me with their profound and valuable support that have helped me in achieving my research aims.

I would like to extend my feelings of gratitude towards my father Fazal Din Khan, mother Shereen Gul, wife Aiman Saadat and daughter Amal Saadat for their endless support and towards Captain Hammad Nazir, Hassaan Bin Mohsin and Asjad Abdul Rehman for their continuous guidance and motivation.

Finally, I would like to express my appreciation to all the people who have provided valuable support to my study and whose names I couldn't bring to memory.

.

ABSTRACT

In the current era, data is said to be more expensive than any of the materialistic things in the world. Data is so fragile that it can easily be tampered with to either decrypt it or it can be made permanently inaccessible without any 1st personal intervention. The term Cloud Computing has gained widespread over the last couple of years. For approximately two decades Cloud computing has had a lead role in the field of IT and a bulk portion of the business community is depending on cloud storage. With the coming of big data and cloud services, client data has turned into a significant issue. Although a variety of detection and anticipation advancements are utilized to ensure client data, ransomware that requests money in return for one's data has arisen. There have been many incidents in the past where data of many users including even high-end companies became compromised which was followed by a ransom note to pay for the cost of decryption of their data mostly through bitcoins. Ransomware has various effects on data characteristics e.g. change in entropy, signatures, extensions, encryption, etc. In our research, we have focused on two main attributes of a file that is entropy and file inaccessibility. These two attributes will be used in our detection algorithm which will actively monitor the data saved in a secure vault with a flexible time interval. Initially, the surveillance code was employed in the sandbag environment of a virtual machine. The same surveillance code was then deployed on Amazon Web Server EC2 virtual server to carry out surveillance of shared storage on the cloud for pre-emptive detection of ransomware. In the end, the resource intensity in terms of processing power and memory of surveillance code will be analyzed on a cloud server.

Keywords: Ransomware detection, File Entropy, File Accessibility

TABLE OF CONTENTS

Contents

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 Overview

In today's modern times, data has replaced money as the most valuable asset for an organization or an individual. Financial assets can be redeemed, re-earned, and restored with time, whereas, valuable data once compromised will rarely come back to the owner in its original form with its confidentiality, integrity, and availability intact to its original version.

Ransomware is malware that encrypts the user data or locks the target device and demands money in exchange to provide access. Ransomware is motivation and gaining popularity among attackers to have monetary gains. In this scenario, even after removable malware the damage is irreversible and requires a key from the originator/ creator. Furthermore, after paying the required ransom there are other damages in terms of time, service unavailability, and reputation.

The first ransomware showed up in 1989 [1]. Starting around 2012, the number of ransomware attacks has expanded altogether. Diverse ransomware exists [2] as Reveton, CryptoLocker, CryptoWall, and WannaCry. The ransomware empowers any person to dispatch his/her own created attack without past significant information on the targeted organization. Ransomware payload not just targets PCs but also cellphones and cloud storage. For instance, a medical clinic has been hit by ransomware, its servers have been scrambled uncovering more than 73k patients [3]. According to black fog business organizations are hit by a ransomware attack every 11 seconds [4].

Cloud storage is a goldmine of information, and not just for companies attempting to transform their information into significant experiences or sell data. Attackers have observed the measures of information stored in the cloud and cloud users as an important target. Service

providers of cloud offer various types of assistance for organizations and people all through the globe, and not simply basic Software as a Service (SaaS) contributions. Many organizations shift their whole data sets to the cloud, frequently utilizing Databases as a Service (DBaaS). Some organizations shift the whole setup to the cloud, utilizing Infrastructure as a Service (IaaS). These administrations have important information needed for business progression, drawing in the attention of malicious actors.

To safeguard the valuable data of cloud users and cloud service providers against ransomware attacks, a detection technique is required to be proposed for timely detection to avoid the spreading of ransomware.

In the present environment, data has surpassed money and become the most valuable asset/ property for the individual organization. Cloud services provide one of the most suitable options to store data as it requires no manpower, less expenditure, and more flexibility. Retrieving Compromised data raises questions on its integrity, confidentially, and availability. The antecedent is better to detect ransomware rather than focusing on the recovery of data.

1.2 Motivation

The inspiration to join the contest against cloud ransomware is the expanded number of attacks in the last couple of years. In the present environment, data has surpassed money and become one of the most valuable assets/properties for an individual as well as for an organization. Once the data has been compromised it is very difficult to regain it in its previous state and raise questions on its integrity, confidentially, or availability. However, considering the case of ransomware it is more often that data is not able to be restored even after paying the required ransom.

Cloud services provide one of the most suitable options to store data. Previously Organizations were used to depending on their in-house servers to store data. Today storing

data online is one of the prime ways as it provides unlimited storage. It is a great platform for all types of business as it does not involve huge investment and provides global access. Furthermore, it provides excessive benefits in terms of manpower, less supervision, and more opportunity to achieve organizational goals.

Keeping in view the threat of ransomware it is better to detect it rather than focusing on the recovery of data after the attack to avoid the loss of the organization's reputation, time, and previous data. The proposed detection technique will be able to detect the ransomware attack in real-time

1.3 Research Objectives

The main objectives of this thesis are:

- Detailed To study the entropy of different types of files e.g PDF, Word documents.
- To study various types of ransomware and their effects on the entropy of files.
- Analyzing file accessibility to detect locker ransomware and data corruption.
- Finally implementing detection techniques on the cloud against ransomware and analyzing its efficiency.

1.4 Contribution

Ransomware detection techniques will contribute in the following ways:

- The research will analyze and monitor the accessed data and compare the entropy before and after access. The system will generate an alarm in case there is a correlation of entropy of the file with the infected file.
- The best defense against ransomware is to detect it before it encrypts/locks the data. The proposed detection technique will monitor the data in real-time and will act as proactive rather than reactive to avoid malware spreading.

- Today cloud is the most useful platform being used to store data. Organization saves their manpower, investment, space, etc. to avoid data servers. Cloud service providers provide various services. However, ransomware as a service is not being provided till now. The detection technique will help service providers to provide ransomware detection as a service.
- Mostly the end-user/ human is considered to be the weakest link in the security chain. The same detection technique can also be deployed on the end-user device as a defensive measure against ransomware.

1.5 Thesis Outline

The research work has been organized and distributed in the following chapters:

- Chapter 1: A brief introduction is given, a problem statement is highlighted, followed by the motivation behind the research, and research objectives are enumerated. Furthermore, the contributions made through this research are highlighted.
- Chapter 2: An overview of existing detection techniques of ransomware and followed by pros and cons of each detection
- Chapter 3: An introduction of the Cloud Computing environment and its importance especially in a pandemic situation. Flow diagram of proposed detection is discussed. Each step of the detection technique is explained in detail.
- Chapter 4: This chapter includes the implementation of surveillance code for actively monitoring the secure vault for the detection of ransomware. Results of detection techniques on different ransomware infected and encrypted files are shown.
- Chapter 5: Finally, the deployment of our detection algorithm on the Amazon Web Server is shown in this chapter. Power consumption in terms of processing and ram is also taken into consideration.
- Chapter 6: The recommendation, conclusion and future work is covered by this chapter.

Chapter 2

LITERATURE REVIEW

2.1 Introduction

There have been many incidents in the past where data of many users including even high-end companies became compromised which was followed by a ransom note to pay for the cost of decryption of their data mostly through bitcoins. This is a part of cybercriminal activity for hackers to get popularity and monetary gains. It has become quite easy for a person to just remain anonymous and ask for ransom for a specified key that will decrypt the data of the victim.

In the current era, data is said to be more expensive than any of the materialistic things in the world. Using data, the value of a company rises or falls but the security measures are taken nowadays to ensure the safety of this data are still in the development phases with new ideas and technologies coming out every day. But the importance of a substance is very different than its fragility. Data is so fragile that it can easily be tampered with to either decrypt it or it can be made permanently inaccessible without any 1st personal intervention.

The cost of protecting data is very high and the simple antivirus is now becoming obsolete for its protection. So, if a regular user wants to save data, they cannot do so themselves. Although they can keep it in an offline hard drive when accessed from 3rd party public computers, it becomes vulnerable to malware. To cater to this drawback, Cloud technology has taken a trend, and to store large amounts of data it is stored in the cloud. Companies like Google, Microsoft, and Amazon are using millions of dollars each day to make these clouds secure from external attacks or malware. Users can pay economically and store data in clouds without the risk of having it stolen or destroyed. But the issue remains for government, military or private companies who host their private clouds. They still require malware detection and eradication capabilities.

In 2018, a Netskope security report identified that 43.7% of malicious software found in the cloud is carrying ransomware which is malware that encrypts or locks the targeted data and can affect all files on a system, and as of 2021, the following number has risen to a staggering 67% making it all-time high and suspected to increase in the coming years. After files are infected, there is very little to no chance an entity can retrieve their data without the proper key which is provided in 90% of the cases by hackers if their demands are met. A lot of work and research can be found for the retrieval of data when it has been infected but detection of malware and its eradication is still anew. If a file is detected to be malware accurately and removed from the system before other files are infected hence saves a lot of time and money. It would hence be recommended to pursue this solution.

The purpose of this literature review is to provide an in-depth view of different techniques and new technologies which are in trend for ransomware detection and entropy measurement. We will review many of the studies which have already implemented or theorized the solution of protecting data on the private cloud from ransomware and some which have also worked on entropy measurement.

2.2 Ransomware Detection

Ransomware can be detected through the deployment of decoy files in the system. These files are also known as honey files [5]. It is a kind of deception mechanism which detects unapproved access to the system. The initial study on the topic has been done by Moore in which he discusses numerous ways to track decoy files by using a hierarchical multi-tier response model. The goal of these files is to:

- Reveal the location of an attacker by having them access these files
- To infect the files which own an attacker to later reveal his position.

The following method only works if the following files are not accessed by the everyday user, and this creates a pattern that can then be identified by the attacker to reveal the location of these files from $STANDARD INFORMATION which can be found under NTFS. Using different methods, we can determine the robustness of the following system and use the results to remove drawbacks from the system.

Another technique to detect ransomware is to monitor the metadata of a filesystem. It is designed to manage and store all information related to file events such as modification, deletion, and encryption, and renaming of files. This data is categorized into Meta properties. In case an attack occurs, which encrypts a file on the victim's system, the Metadata will be modified and when compared to the earlier data, it will help to raise the alarm for the attack. [6] FACE, a ransomware detection framework, monitors the $J flag in $UsnJrnl entry. The following framework helps to detect ransomware more effectively in a cloud computing system and blocks detection bypass by decreasing the overhead of analysis and monitoring by 0.1% CPU overhead and 0.3% read performance overhead than the usual behavior-based ransomware detection architectures which are used to measure the entropy of the system.

When ransomware is run on a system, it is known to run in a set pattern in which it encrypts files via their search order. [7] Implements a hooking technique to first create a large number of decoy files then monitors all the processes of the system by injecting all processes of the system. Even if new processes are created, they will be injected. Moreover, it hooks the FindFirstFile and FindNextFile APIs so that NewFindFirstFile and newFindextFile are called first, which if called will prioritize the search of decoy files of a system. The decoy file monitor is always checking the Shannon entropy of files. If a file is modified, the process will then identify the process and user which modified the file and hence shut them down. The cost of disk, memory, and CPU utilization is low concerning the benefits of the technique which

identifies and stops encryption processes, so no harm is done to data. The flow chart below describes the process:

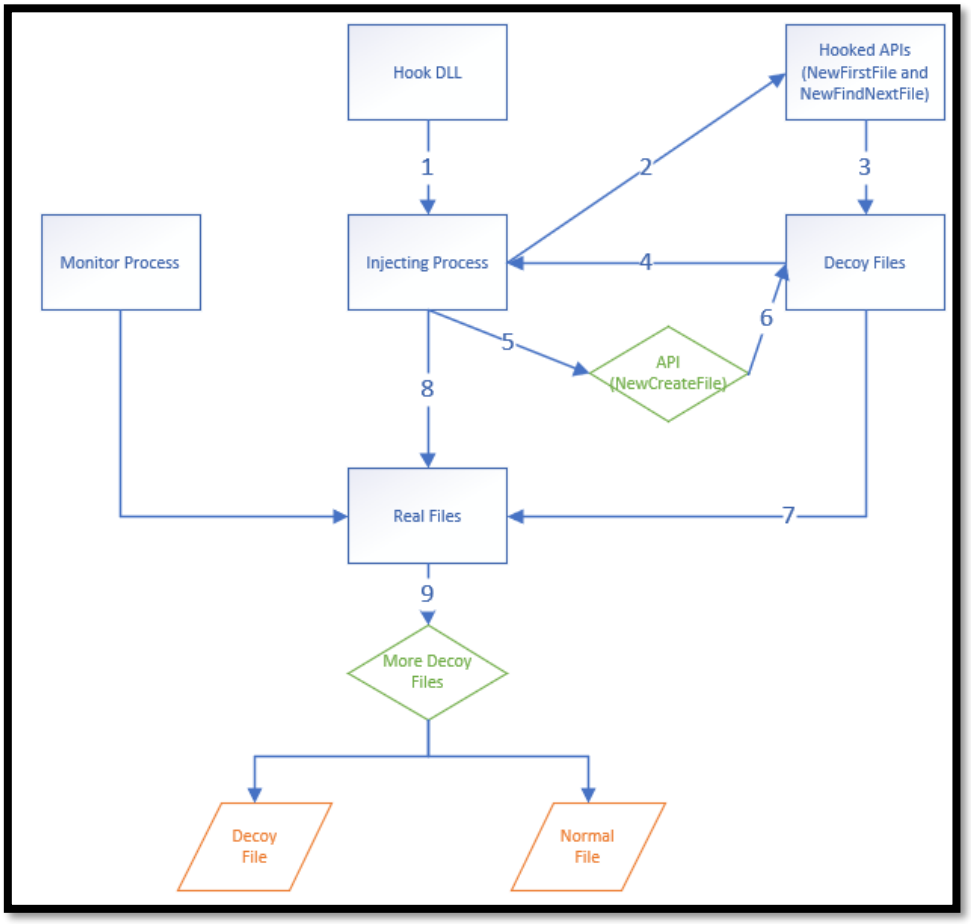

Figure 2.1 Decoy File System

Honeypots can also be implemented to detect ransomware. The following study [8] uses the Microsoft Files Server Resource Manager feature and Event Sentry to influence the security logs of a system which acts as an alarm system to alert in case of any unauthorized access or any malware attack to block any further damage. The step of triggers and responses can be seen in Figure 2.2.

Figure 2.2 Steps of Triggers and Responses

The random forest classifier which is one of the most prominent and robust machine learning techniques to detect ransomware is discussed in [9]. It provides the following advantages:

- Requirement of fewer input parameters.
- Resistance to overfitting.
- Variance decreases without resulting in bias and an increase in the number of trees.

The experiment performed shows a high performance of a forest classifier for ransomware detection for static analysis.

By monitoring API call sequences of malware binaries and translating them to a set of features, Chen et al. [10] suggested a method for ransomware detection based on a dynamic API calls flow graph. Random forest, SVM, Naive byes, and logistic regression were among

the data mining algorithms employed. The logistic regression model has the highest accuracy (98.2%) and the lowest false positive rate (1.2%). However, the evaluation was based on a dataset of only 168 ransomware samples, and the focus was solely on a single feature for detecting ransomware.

Schultz and colleagues experimented on the usage of machine learning to detect malware. They used strings information, Portable Execution (PE), and binary-byte sequences to categorize the ransomware by using the Naïve Bayes classification algorithm. The different phases which the malware takes to spread to the cloud have been discussed in [11]. The overview of the steps is as per below:

1) Exploitation and Infection
2) Delivery and Execution of malware
3) Backup Spoilation
4) Normal Files Encryption
5) Disabling User Notification
6) Cleanup

Routa and colleagues in [12] discuss steps to detect ransomware and their early mitigation. They achieved an accurate detection by using per-thread file system transversal over the standard old metrics such as the use of system calls or Shannon entropy. They use a sample set of 700 active ransomware to perform testing and no work has been done in this area. The solution suggests the marking of folders in which thread is passed and afterward decoy folder counter is incremented. The final score is calculated by dividing the number of decoy counters by the total number of decoy folders. In case the value obtained passes a specific set threshold, the files in the folder are all marked as malicious. There is a very low chance that a normal thread will bypass at the very least three Decoy Folders. For the following experiments, the threshold has been set to 0.6. The drawback of the following method is, If

the value obtained is less than 0.6, the ransomware will not be highlighted to the algorithm, hence it will not be detected.

Secondly, their algorithm plots a trace of the malware which is then compared to the pre-existing data and helps decide to classify the type of malware. Using the following information, necessary steps are taken accordingly. Plotting of trace is done in the form of a dendro-gram. Machine Learning is used as a means for comparison of current data obtained to the dendro-gram of saved malware. Hence the following method was successfully and precisely able to identify many different families of malware.

In [13] the author has proposed a taxonomy of comprehensive ransomware. Using the results of said taxonomy and a newly discovered attribute of high survivable ransomware (HSR) present in the key protocol step, the author suggests a method of detecting Hight Survivable Ransomware and prevents them from taking hold of or encrypting user's data. Following Ransomware's have been used in the experiments listed below:

a) Non-Cryptographic Ransomware (NCR)
b) Cryptographic Ransomware (CR)
c) Private-key Cryptosystem Ransomware (PrCR)
d) Public-key Cryptographic Ransomware (PuCR)
e) Hybrid Cryptosystem Ransomware (HCR)

The results of the experiment can be seen below in Table 2.1.

Table 2.1 Results of HSR Detection

Ransomware Name	Ransomware Type				HSR	Detection
	HCR	*PuCR*	*PrCR*	*NCR*		
Cryptolocker	√	×	×	×	√	√
Cryptolocker 2	√	×	×	×	√	√
Cryptolocker 3	√	×	×	×	√	√
Cryptowall	√	×	×	×	×	√
Cryptowall 2	√	×	×	×	√	√
Cryptowall 3	√	×	×	×	√	√
CoinVault	√	×	×	×	√	√
CryptoGraphic Locker	√	×	×	×	×	√
CryptoDefense	√	×	×	×	×	×
CryptoDefense 2	√	×	×	×	√	√
CryptorBit	×	×	√	×	×	×
TorrentLocker (original)	×	×	√	×	×	×
TorrentLocker	√	×	×	×	√	√
ACCDFISA	×	×	√	×	×	×
BuyUnlockCode	√	×	×	×	×	×
CryptoFortress	√	×	×	×	×	×
PClock2	×	×	√	×	×	×
Critroni(CTB Locker)	√	×	×	×	×	×
ComputerCrime&IntellectualProperty Section	×	×	×	√	×	×
Harasom	×	×	√	×	×	×

The following system is the first of a kind framework designed to monitor connections made to the network which are suspicious, and it also prevents them to attack the user's data. It is also useful in the detection of malware such as botnets, Bitcoin-mining malware, drive-by downloads, etc.

A framework named 2entFOX is proposed to detect High Survivable ransomware (HSR) by monitoring Windows ransomware behavior and finding unique features having very low false-positive rates and high accuracy is discussed in [14]. After performing the experiments, the team was able to extract 20 such parameters including 2 highly accurate ones which helps to achieve HSR detection. The framework of 2endFOX can be seen below in Figure 2.3:

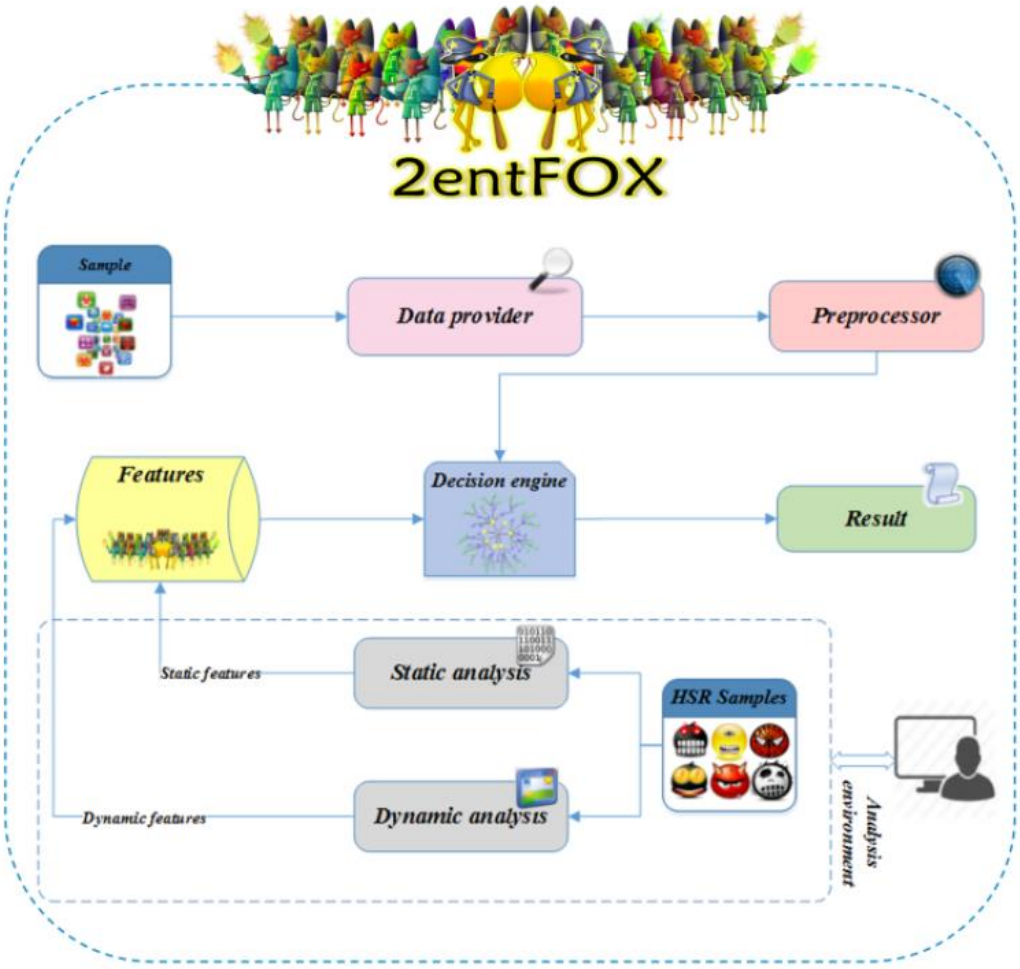

Figure 2.3 Framework of 2entFOX

2entFOX is meant to point out the most destructive and dangerous malware known as HSRs. Low Survivable Ransomware (LSR) detection is not prioritized in the following experiments. In case we want to design a framework to detect all ransomware, the following challenges will be faced.

- Increase in False Positive and Negative rates during detection of HSRs.
- Increase in False Positive Rate for detection of all types of malware.
- Increase in run-time and static overhead for collecting data and making decisions.

The major difference between 2endFOX and any other specification-based detection system is it can detect and classify HSRs unlike any other tools currently available using 20 specific features found in windows as listed below in Table 2.2.

Table 2.2 Specific list of features in windows

#	Feature name	Analysis type	Detection stage
1	Access to cryptographic libraries	Dynamic	Triggering
2	Decryption help file key words	Static	On disk
3	Decryption help file key words	Dynamic	Dormant
4	Decryption help file key words	Dynamic	Execution
5	Targeted files search key words	Static	On disk
6	Targeted files search key words	Dynamic	Dormant
7	VSS	Static	On disk
8	VSS	Dynamic	Dormant
9	VSS	Dynamic	Triggering
10	Specific registry paths key words	Static	On disk
11	Specific registry paths key words	Dynamic	Dormant
12	Access to specific registry paths	Dynamic	Triggering
13	Access to specific registry paths	Dynamic	Execution
14	Specific directories access key words	Static	On disk
15	Specific directories access key words	Dynamic	Dormant
16	Access to specific directories	Dynamic	Dormant
17	decryption help file content key words	Static	On disk
18	decryption help file content key words	Dynamic	Dormant
19	Abnormal access to the paths and files	Dynamic	Execution
20	Key-exchange step feature	Dynamic	Triggering

In [15] the author describes a method to detect the malware in the system at the time of encryption of the files. This method works regardless of any specific strain of ransomware. The experiment done contained over 130,000 files which included Microsoft Office files as well as higher entropy files such as archives and encrypted files. The goal of the experiment was to detect ransomware by calculating the entropy of files. The following model is accurate enough to yield a 99.96% success rate in over 80,000 files. The basic issue usually faces is the differentiation of a simple compressed file and one that has been encrypted using malware. The following method successfully addresses this issue as when we are differentiating the entropy

plot of a file under inspection with a file generated that contains purely random numbers, the amount of correlation helps to determine whether the file under analysis comprises encrypted data.

The major aspect of this paper is to detect and classify malware even when the overall entropy level is too high. This was done by firstly to find the entropy of different formats of files, the following procedure inspects the initial bytes of files which are under investigation. Afterward, it determines the correct number of encrypted files present in a data set. Analyzing a small portion of the file and giving results in a specified time is one of the reasons if using this method.

Aviad and colleagues in [16] presented a procedure for detecting malware present on virtual servers' part of cloud being used by a private company. They analyzed volatile memory dumps taken from a VM using volatility framework and created meta-features. The help of machine learning on these meta-features led to the detection of ransomware in these Virtual servers present on the private cloud.

Using a random forest classifier, they were able to obtain the following results:

Table 2.3 Results using Forest Classifier

TPR	1
FPR	0.052
F-measure	0.976
AUC	0.966

These results were obtained using the following formulas.

$$\text{Precision} = \frac{TP}{TP+FP} \tag{2.1}$$

$$\text{Recall} = \frac{TP}{TP+FN} = TPR \tag{2.2}$$

$$\text{F - Measure} = \frac{2.Precision.Recall}{Presicion+Recall} = \frac{2.TP}{2.TP+FP+FN} \tag{2.3}$$

The process works in a way as to take continuous snapshots of a Virtual Machine present on a cloud. These snaps are taken at a specified time interval. Afterward, they are analyzed immediately, and specified features are to be extracted, analyzed, and classified as if they are infected. In case they are infected, the machine is disabled and shut down.

Before analysis, the features extracted from the snapshot can be saved in SQL for future works and the rest can be deleted. The fooling prototypic mode before implementation should be tested and made ready by running a large amount of static and continuous instances of the VMs as possible. The following drawbacks were faced by the team such as the snapshotting causing the VM to hang up and freeze for small instances of time. The recovery from this state depends upon the computing power and specifications of the machine itself. To overcome this issue, the system has to be designed in such a way that multiple servers should be connected through a load balancer and when snapshots are taken on one server, they are not taking on the other servers rather they are taken serially as can be seen in Figure 2.4:

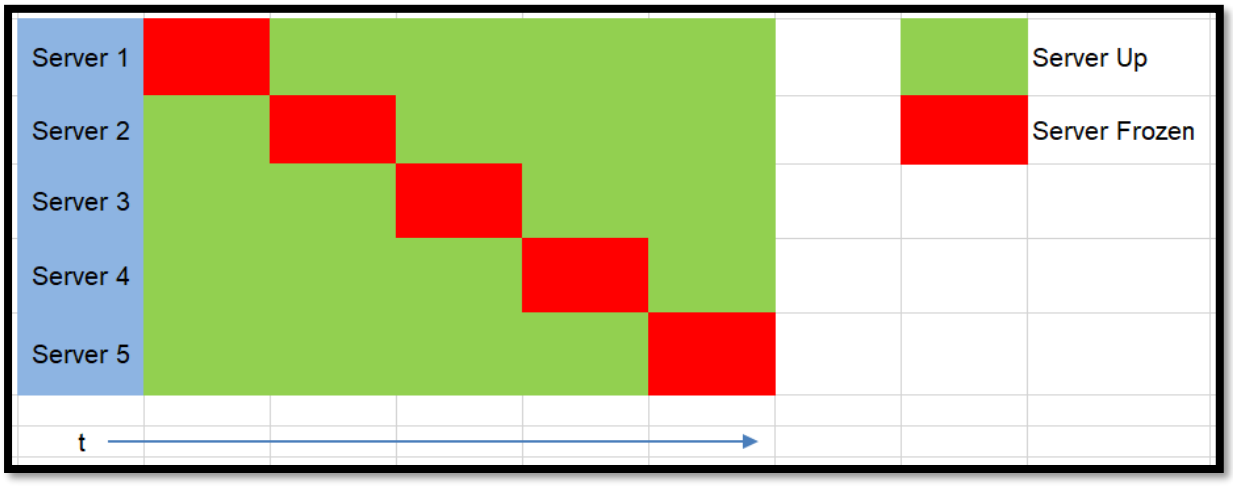

Figure 2.4 Working of VM through Multiple Servers on Load Balancer

Some of the limitations of the following model as discussed by the author are as below:

a) Single Virtual Server Freeze Up

b) A large amount of Snapshot Data

c) Snapshot Transfer Time

d) Increase in Computing Overhead due to large size of the snapshot

e) Separate Framework for Physical host machine

In [17], the author uses Identify-Collect-Examine-Analyze-Present (ICEAP) procedure for identifying LooCipher ransomware-based activities which he prefers over the traditional complex algorithms. The following method monitors online behavior and source/destination entities. By using minimum efforts, analyst can combine security mechanisms into eigen values to analyze the full payload and discover hidden network threats. The step-by-step approach can be seen as per below:

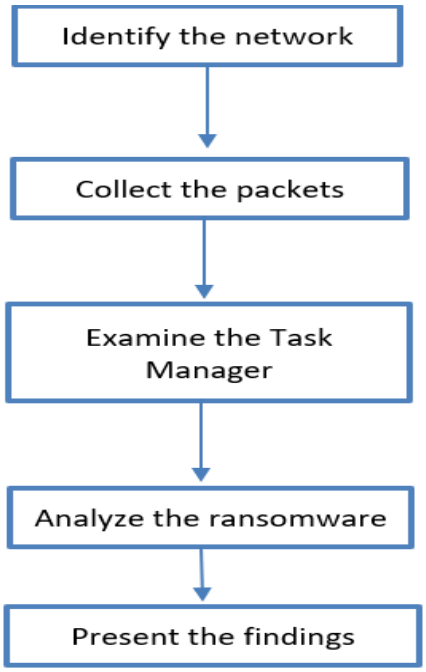

Figure 2.5 Steps of finding threat in network using packets

For LooCipher ransomware detection, to identify the networks, we need to use several tools to monitor the packages without any filters such as Wireshark, NPCAP, etc., and establish a normal baseline for the features or parameters we are looking for and provide a filter for it. These filters help us to reduce the amount of data under observation and discard the rest. Afterward, we need to create a task to monitor the I/O, Network usage, CPU utilization in the

tasks manager and connect to a network drive for detection of ransomware I/O pointers. Apply a watchdog to continuously check for ping and connectivity to internet services. Now as all the services are enabled, the teams infect the system with ransomware and check its behavior. The whole process is then repeated to check the behavior of ransomware and results are compiled which then help to add security measures to stop the service to the system in case any similar behavior is detected.

The following ICEAP procedure can help to lessen the efforts and protect the data of the user efficiently. The efficacy and feasibility of the lightweight approach in Wireshark packet extraction are verified as compared to the everyday conventional methods that are being used. The only drawback is the limited dataset of ransomware the following technique has been currently implemented but will increase in the future.

In [18] the authors present a framework known as CryptoDrop which can act as an early-warning detection and alerion system in case of any suspicious file activity. CryptoDrop can help to stop any process which has been interfering with the huge amount of user's data by using a set of behavior filters/indicators. Moreover, by using common ransomware pointers, the system can be made intelligent for rapidly detecting ransomware having very low false-positive rates. By performing a detailed analysis of the behavior of ransomware, cloud users can create a foolproof system for its early detection which helps in less loss of data for the user. Some indicators for helping to detect malware at runtime are as follow:

a) File rights Changes
b) Similarity Measurement
c) Shannon Entropy
d) Deletion / File type Funneling
e) Union Indicator (Combination of first 3 indicators)
f) Indicator Evasion

CryptoDrop detects malware by the change in real-time data of users. Firstly, it monitors all the read/write permission of files present on the system. If there is any questionable change that triggers any CryptoDrop indicator, it increments a counter which if breached pauses the system. To resume the system, the permission of the owner is required. 492 real-life samples which included 14 different classifications of malware families were detected using CryptoDrop with a hundred percent detection rate and less to zero files lost before detection.

Jinsoo and colleagues [19] use machine learning and dynamic analysis to detect ransomware using a two-stage process. Initially, they build a Markov Model to obtain the characteristics of the ransomware by monitoring Windows API call sequence Patterns. Afterward, they construct a Random Forest Machine Learning Model for analysis of the rest of the data. This helps to control the False Negative Rate (FNR) and False Positive Rates (FPR) Error Rates which results in a combined accuracy of 97.3% with 1.5% False Negative Rate and 4.8% False Positive Rate.

When a computer application or program enters any system, it calls Windows API's which reveal the properties of the program. In the experiment, they create 2 different Markov Chains, the first for malware and the other for a normal sleeper program. Comparison of analysis from results of both scenarios shows whether the concerned program is harmful or benign. 1909 malware and 1139 plain samples were used for the experiment which they ran in the Cuckoo sandbox to get API sequences that are called-upon on runtime. 303 unique APIs were identified.

In the 1st stage, the Markov model analyzes and helps to detect malware as it gets a low False Positive Rate. The random forest method is applied to the remaining data during the 2nd phase. Below Table 2.4 shows the result of the experiment using different threshold values rather than the usual value of 0.5.

Table 2.4 Results of Different Thresholds

Threshold	ACC	FPR	FNR	F1 Measure
0.1	0.9665 ± 0.0156	0.0693 ± 0.0312	0.0120 ± 0.0074	0.9737 ± 0.0120
0.2	0.9728 ± 0.0140	0.0483 ± 0.0275	0.0147 ± 0.0073	0.9785 ± 0.0109
0.5	0.9702 ± 0.0125	0.0219 ± 0.0148	0.0346 ± 0.0143	0.9759 ± 0.0101
0.9	0.9560 ± 0.0094	0.0097 ± 0.0061	0.0644 ± 0.0119	0.9638 ± 0.0079

If they only used the Windows API calls to determine the presence of malware, it gave a good False Positive Rate, but False Negative Rate was quite high as much as 20% which is not good. To decrease this value, they introduced a second stage to the experiment which was to run the results through a random forest classifier. As False Negative Rate and False Positive Rate are inversely related, they create a rule to find a threshold value that provides the least value for False Negative Rate and False Positive Rate less than five percent. Hence the threshold value is decided to be 0.2 which gives 97.28% accuracy with 1.47% FNR and 4.83 False Positive Rate to the two-staged mixed detection dynamic analysis model. In malware detection, higher values of False-negative rates are catastrophic to the system, so it is good if this value is decreased as much as possible. The high value of False Positive Rate can be recouped by external signature-based methods.

In most cases of ransomware personal pictures of the victim are mostly encrypted and hence [20] aims its efforts towards developing a system that helps to detect any online doubtful

processes which try to access a large number of files and encrypt them. The focus of their research is JPEG format files and to detect TorrentLocker ransomware using the Kullback-Liebler divergence method. Initially, the team tried to first narrow down the method which they were going to use, they optioned it down to either Shannon Entropy or Kullback-Liebler divergence method. To evaluate the performance of Shannon's entropy, a set of total 150 files were encrypted using AES 256-bit encryption algorithm in Cipher Block Chaining (CBC) mode. Using the results, the team was able to conclude that it would be near impossible to differentiate between normal JPEG and encrypted files as the values are too close to each other. If they had moved forward with Shannon Entropy, it may have led to a higher False Positive or False Negative Rate which may have had the worst impact on the experimentation. Any ideal encryption-based malware should encrypt files that seem like random files. In the 2nd part of the experiment, the Author and team used a set of 2000 JPEG files which were split into 12 categories i.e., space, sports, animal and residential, etc. These files were also encrypted in CBC mode using the same AES 256-bit encryption algorithm as in the 1st part of the experiment. Results can be seen in below Table 2.5.

TorrentLocker is known to encrypt the 1st two megabytes of data of each file. They have calculated the first 128, 256, and 512 KBs of each file to provide a more efficient value. When considering a threshold of 0.002, Kullback-Liebler divergence can distinguish JPEG from encrypted files with a detection rate of 99.95% which is one of the most prominent results in the field.

The following experiment has been performed offline. In Future Work, the following method is to implement the solution on an online system. A process will be running periodically which as soon as any form of file encryption, will auto-shutdown the server/system which will help to minimize the damage to the user's property and minimize any loss of data.

Table 2.5 CBC Encryption of JPEG Files

File Types	Block Size	Minimum	Average	Maximum	Variance
JPEG	N=1	0.007	0.0189	0.1456	0.00010044
	N=2	0.0428	0.1737	0.2535	Too Low
Encrypted	N=1	0.00034939	0.0013	0.0149	0.0000005471
	N=2	0.0078	0.1105	0.1497	0.0018

Chapter 3

PROPOSED DETECTION METHODOLOGY TO DETECT RANSOMWARE

3.1 Introduction

In this chapter, a methodology adopted to conduct detailed research on the topic is discussed. We proposed a detection technique that is compromising of five steps to efficiently detect the attack of ransomware. The flow of the technique will be explained in detail. Various popular ransomware attacks will be covered in this chapter. The impression of ransomware on data will be studied and entropy will be taken into consideration. Furthermore, the changes in the extensions by ransomware will be analyzed. Moreover, the emerging technology of cloud computing with threats will be covered. The proposed technique will finally be implemented on cloud storage and results will be examined after testing on different files.

3.2 Emerging Cloud Computing Technology

The term Cloud Computing has gained widespread over the last couple of years. For approximately two decades Cloud computing has had the lead role in the field of IT and a bulk portion of the business community is depending on cloud storage. According to the latest statistics [21] 69% of world businesses, are relying on cloud technology. With the exponential increasing trend in digital data, it is becoming more and more difficult for individuals and companies to hold all of their vital information, storage devices, processing infrastructure, and systems to be running on in their in-house servers. Moreover, the pandemic situation of Covid -19 has further elevated the Work from Home concept which results in more reliance on cloud technology.

3.2.1 Types of Cloud

Mainly the Cloud has 4 broad categories in which cloud technology has been divided. These are briefly described below: -

3.2.1.1 **Private Cloud**

Private cloud is usually kept behind a firewall to have a strict check on network traffic. Only authorized clients can access the cloud. Organizations having sensitive data usually prefer this type of a cloud facility

3.2.1.2 **Public Cloud**

Amazon and Google are the two famous organizations/ companies that are well-known public clouds. These clouds are usually for large storage spaces and are publicly accessible to all. Platforms can provide more storage spaces on payment to their clients.

3.2.1.3 **Hybrid Cloud**

This type of cloud has the facility of both public and private cloud. Banks/ Universities put their non-sensitive information on the public cloud for readily available to users. Sensitive data is stored on a private cloud that has only authorized access.

3.2.1.4 **Community Cloud**

A community cloud is a private cloud that works similarly to a public cloud. Community Cloud is a private cloud but works on similar lines to a public cloud. It is shared among the various authorized organization to work on the same application. For example, Government departments that work on a similar application can have a community cloud.

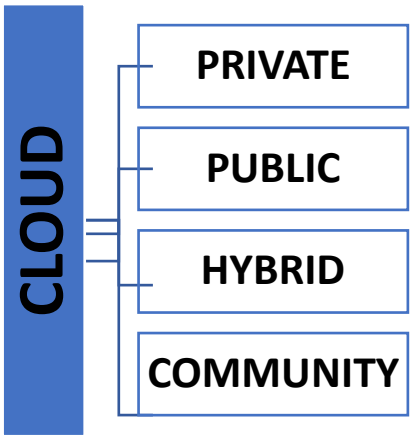

Figure 3.1 Types of Cloud

3.2.2 Threats to Cloud

With increasing dependence on cloud technology, the threats also raise incredibly. A few common threats are:-

- Database breaches
- Ransomware attack
- Misconfiguration of network
- Denial of Services
- Hijacking of accounts and gaining auto access

The threat entrusted to us against the cloud is ransomware which will be discussed in detail here.

3.3 The threat of Ransomware to Cloud

With the coming of big data and cloud services, client data has turned into a significant issue. Albeit an assortment of detection and anticipation advancements are utilized to ensure client data, ransomware that requests money in return for one's data has arisen. Mainly the Ransomware is of 2 broad types as shown in figure 4.2. Crypto Ransomware encrypts the targeted data and Locker Ransomware locks the targeted device/ file and demands a ransom in

return to allow access to it. There have been many incidents in the past where data of many users including even high-end companies became compromised which was followed by a ransom note to pay for the cost of decryption of their data mostly through bitcoins. This is a part of cybercriminal activity for hackers to get popularity and monetary gains. It has become quite easy for a person to just remain anonymous and ask for ransom for a specified key that will decrypt the data of the victim.

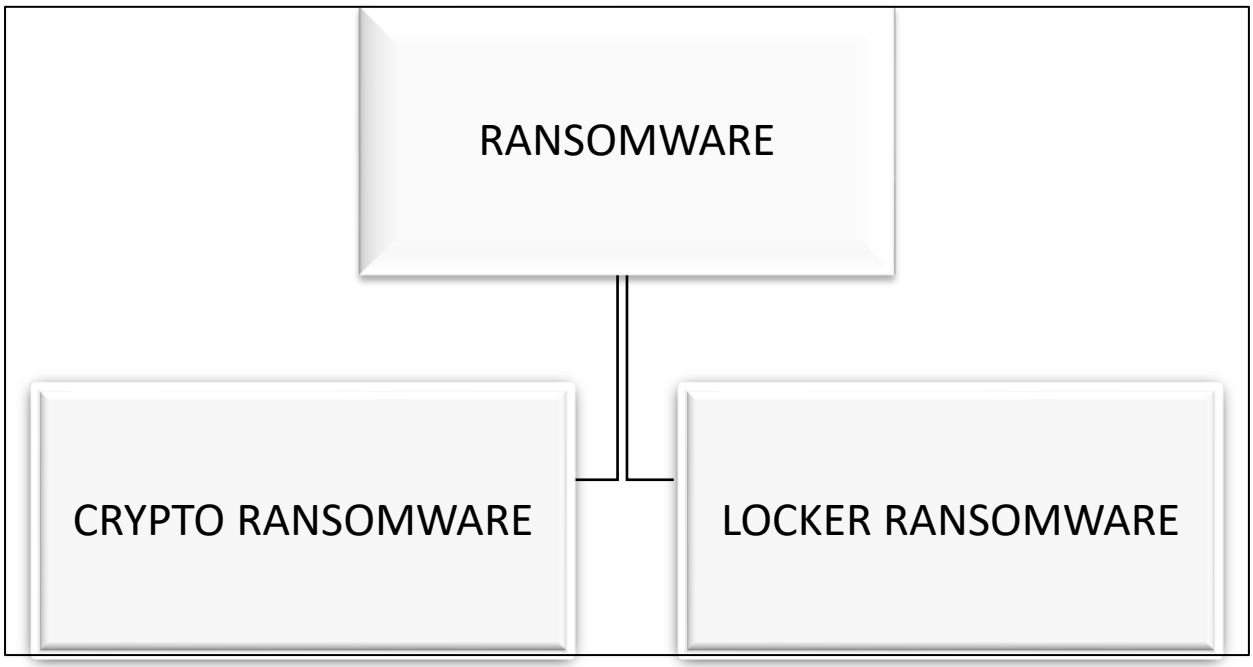

Figure 3.2 Types of Ransomware

3.3.1 Popular Variants of Ransomware

Ransomware has dozen of variants, each of them having its distinctive features and characteristics. However, some of them were successful to achieve their goal which makes them stand out from the crowd which are highlighted below:-

- **Ryuk**

It is a much-targeted ransomware variant that is delivered via phishing emails. It is well-known which demands an average ransom of over one million US $. The ransomware was discovered in 2018 and is currently still an active threat

- **Maze**

Maze is a famous ransomware that has a dual damaging nature which is data theft and encryption [22]. It encrypts the data and demands a ransom in return. If a victim is not able to meet the demands, the attacker publicly exposes the victim's data or sold to the highest purchaser. The variant was initially discovered in 2019 [23].

- **Reyil**

This Ransomware targets large organizations [24]. This malware was responsible for Kaseya and JBS. It was discovered in 2019. This ransomware is using a double extortion technique in which attackers after getting the first payment also sometimes threaten victims that their sensitive data will be exposed if a second ransom is not paid.

- **Lock bit**

Lock bit was discovered in 2019 and it is a data encryption malware. This malware is designed to encrypt the data of large organizations rapidly to prevent its detection quickly [25].

- **Dear Cry**

Dear Cry is a variant of ransomware that took advantage of security patches against vulnerabilities released by the Microsoft office in 2021 [26]. It further instructs the victims how to decrypt the data using and email.

3.3.2 Encryption Scheme of Various Ransomware

The ransomware mostly used RSA, AES-128, AES 256 in different modes like CBC, etc. Table 3.1 shows various popular ransomware encryption schemes [27].

Table 3.1 Encryption Mode Used by Ransomware

RANSOMWARE	DISCOVERY	ENCRYPTION MODE
AES_NI	2016	AES-256

Alcatraz Locker	2016	AES-256
Hidden Tear	2015	AES
Ryuk	2018	AES & RSA
Maze	2019	RSA
Lock bit	2019	AES

3.3.3 Effects of Ransomware on Data

The ransomware attack encrypts the data or locks access to the file. There are various effects on data properties e.g change in entropy of data, the transformation of extensions of a file according to a type of ransomware, and change in magic bytes/ signature in header bytes. We will discuss entropy and extensions of data after a ransomware attack.

3.3.3.1 **The entropy of Ransomware Infected file**

Entropy is the randomness measured of a file in digital systems. The Entropy concept was first originated in the study of thermodynamics. Claude E. Shannon applied the concept of entropy in digital communication in his work “A Mathematical Theory of Communication” [28]. The compressed file has shortened pattern of bits which replaces the large bit pattern. Thus the encrypted and Compressed have a high degree of randomness meaning by next bit pattern of a character is less predictable than on the previous character. Shannon introduces a formula to calculate the entropy of a file system. Entropy is maximum when the bit pattern of all characters is equally distributed in a file system. The Entropy H formula provided by Shannon is under:

$$H = -\sum pi \log 2\ pi\ i \qquad (3.1)$$

Where pi is the probability of character I appearing in the alphabet stream. After the attack of ransomware on the file, the randomness is bit pattern is increased and entropy of the file is increased.

3.3.3.2 Extensions of Ransomware Infected Files

As previously discussed, that ransomware changes the original extensions of files. Every ransomware has a different extension. Table 3.2 shows various ransomware extensions of infected files [29].

Table 3.2 Extensions of Ransomware Infected Files

RANSOMWARE	DISCOVERY	EXTENSIONS
AES_NI	2016	.aes_ni, .aes256
Alcatraz Locker	2016	Alcatraz
Hidden Tear	2015	locked, .34xxx, .bloccato,
Ryuk	2018	.ryk
Maze	2019	ecc, . ezz, . Exx
Lock bit	2019	.abcd"

3.4 Proposed Detection Methodology to Detect Ransomware

Subsequently discussing the importance of Cloud Computing technology, the affiliated threat of ransomware against it, we finally propose a detection technique to detect ransomware to thwart the damage caused by it. The flow diagram 3.3 of the detection technique is shown below. There are a total of five steps of our technique which comprises two checks for the detection of ransomware.

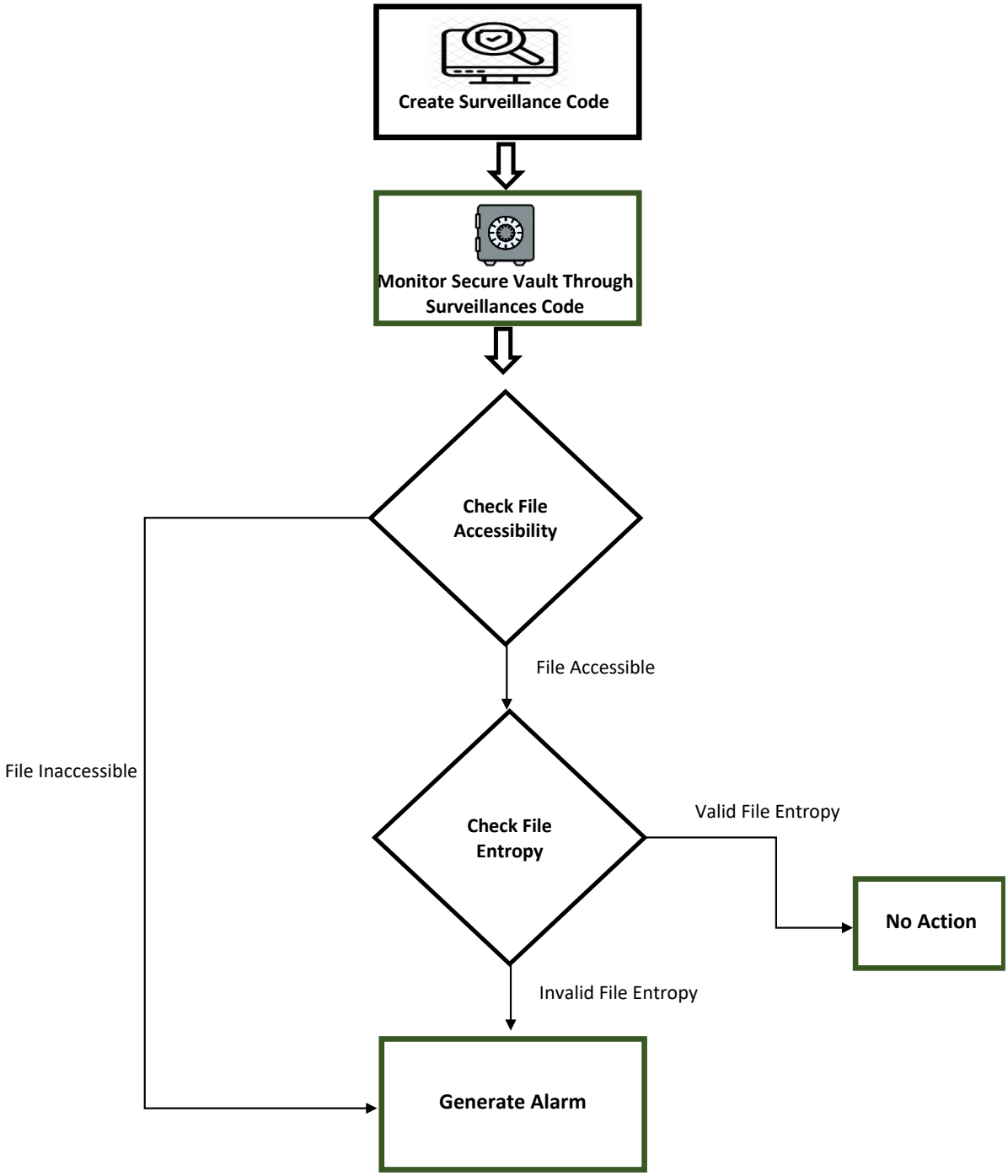

Figure 3.3 Flow Diagram of Detection Model

3.4.1 Explanation of Flow Diagram

Each step of a flow diagram is discussed below:

- **Step 1 – Create Surveillance Code**

The first step of our detection technique is a creation of a surveillance code. This surveillance code will continuously monitor all the files/ data. The period after which the code will repeat the monitoring process will be decided by us e.g after 10 seconds the surveillance code will check complete data again and the cycle will repeat itself.

- **Step 2 – Monitor Secure Vault through Surveillance Code**

The Surveillance code will be applied on a secure vault which is a storage on the cloud on which code is actively monitoring the data. User will store their data in this vault to have a guard for early detection of ransomware.

- **Step 3 – File Save, edited/ modified**

In the third step of the detection scheme, data/files saved in the secure vault will be monitored which includes saving of new file or modification/ editing of the already saved file.

- **Step 4 – Check Accessibility to Detect Ransomware**

Ransomware can corrupt the file and make it inaccessible. In this step, the surveillance code will check the accessibility of the newly added file in a secure vault. In a similar pattern, the surveillance code will also check the accessibility of saved data which have some modification e.g editing, etc. If the code is not able to open the file, an alarm will be generated against ransomware.

- **Step 5 – Check File Entropy**

Entropy is measured by randomness. Ransomware infected files have more randomness which results in a high degree of entropy. In the fifth step of the detection scheme, the code will calculate the entropy of each file and will generate an alarm if the entropy of the file is higher than the threshold entropy. The threshold value of entropy will be calculated by taking the

mean of entropy of ransomware free files e.g the average of 100 files entropy is 3.5, so the threshold value will be selected around it.

Chapter 4

IMPLEMENTATION AND TESTING OF PROPOSED DETECTION TECHNIQUE

4.1 Introduction

In this chapter, a proposed methodology of detection of ransomware files using file inaccessibility and entropy will be implemented. The implementation will be carried on using python 3. The libraries of python 3 required in code will be explained. Each step of the detection technique will be discussed in detail. Different formats of files will be passed through each step and the result will be shown and analyzed.

4.2 Dataset

Data is a significant entity on which detection algorithm will be tested and results will be deduced. As previously discussed, ransomware uses different encryption algorithms e.g AES, RSA, etc. The data set will be comprised of Synthetic data and samples of different ransomware taken from GitHub. A synthetic dataset will be generated using different encryption schemes on the different formats of files. The algorithm will be tested on this synthetic data set for measuring its accuracy. For the samples of ransomware virtual environment will be created on which ransomware attack will have occurred and infected samples of ransomware will be collected. Finally, the algorithm will be tested on these infected ransomware samples.

4.2.1 Data Collection – Ransomware Infected Samples

Various ransomware exe is available on Github. This ransomware will be executed in a controlled virtual environment to collect the different ransomware samples. Table 4.1 shows the ransomware families and the samples collected from them.

All these samples were executed to further check their effect on sample files kept in a sandbag environment.

Table 4.1 Number of Ransomware Samples per Family

Family	No. of Samples
Cerber	2
Badrabbit	1
7even	1
Derialock	1
Infinity Crypt	1
WannaCry	2
Total	8

4.2.2 Experiments Conducted in Sandbag Environment

An Oracle VM Virtual box is used for creating a virtual environment of Windows 7 as shown in figure 4.1. The normal working files of different formats were saved before the execution of the ransomware attack. Each ransomware is executed to collect its sample for further analysis. The ransomware attack of Cerber ransomware and wana cry ransomware is shown in Figures 4.2 and 4.3 respectively.

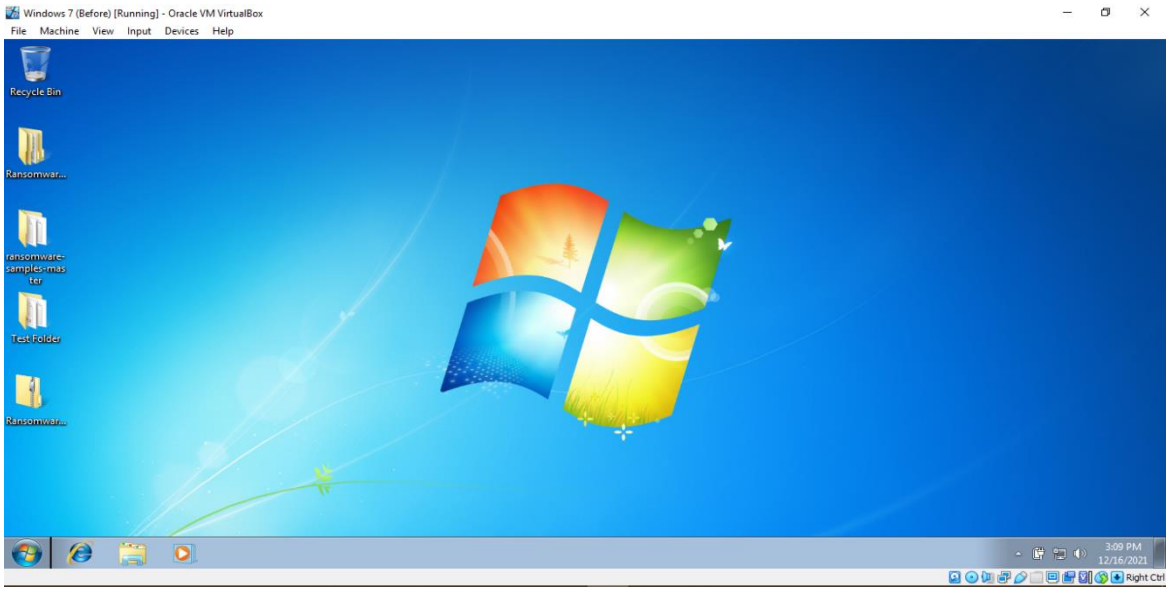

Figure 4.1 Normal Working of Windows 7 on Virtual Machine

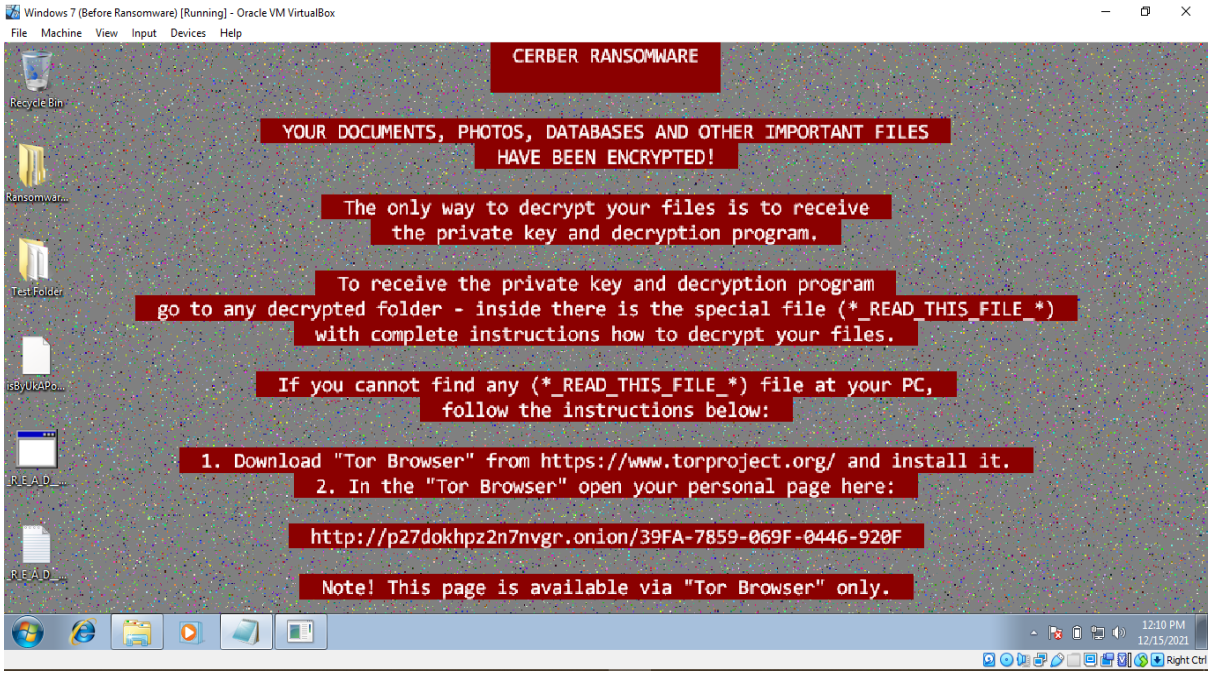

Figure 4.2 Windows 7 after Cerber Ransomware Attack

Figure 4.3 Windows 7 After Wanna Cry Ransomware Attack.

After the execution of ransomware, its behavior on sample files already placed in windows 7 will be monitored. The infected sample files will then be dragged into the dataset folder for further experimentation on surveillance code.

This process was repeated for all the above-mentioned ransomware families and their samples through the data collection process. This process is described in flow diagram 4.4 below.

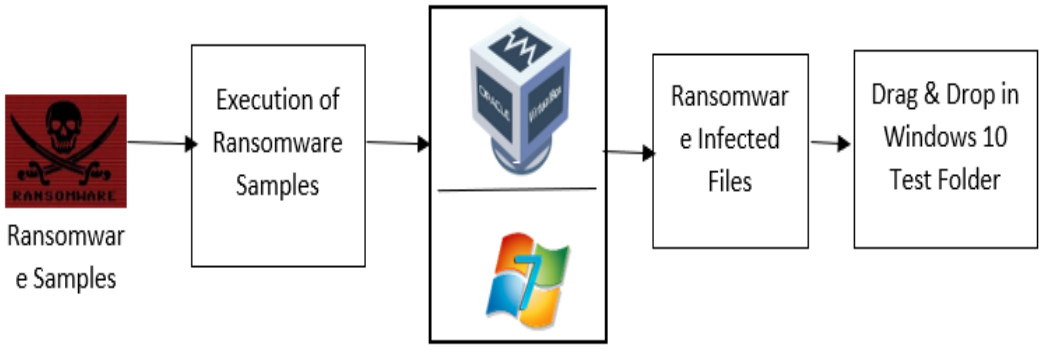

Figure 4.4 Steps of data collection from Ransomware samples

4.2.3 Synthetic Data Collection

A collection of synthetic data files of different formats will be encrypted using different encryption schemes. As previously explained that ransomware mostly uses RSA/ AES encryption. These two main encryption schemes will be used on sample files and the result in infected samples will be analyzed through surveillance code. The process is shown in following diagram 4.5 below: -

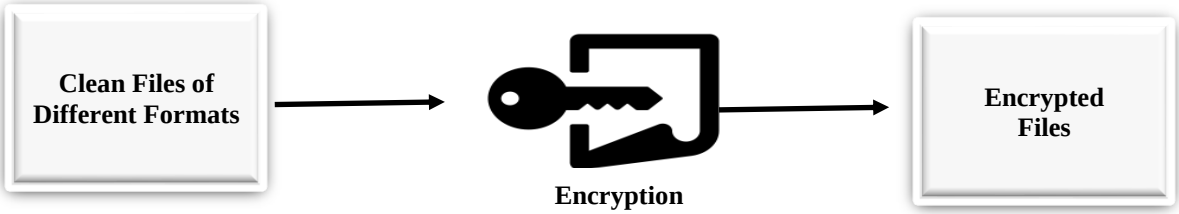

Figure 4.5 Synthetic data collection

4.3 Work Flow of Detection Technique

As previously discussed, our proposed detection technique consists of 5 steps which provide two checks against the intrusion/ ransomware threat. The methodology adopted in our research can be easily represented/ understood with the help of figure 4.1 below.

The first step is the creation of a surveillance code. Python 3 is used upon which surveillance code is designed. This code will monitor the secure vault, step 2 of our detection technique. The code will monitor any modification in existing files or the addition of a new file in a secure vault.

The ransomware has various effects on files as previously discussed in chapter three. The effects which we will consider are file accessibility and entropy. Ransomware infected files are inaccessible and result in high entropy of a file. Encrypted files have a high measure of randomness. The formats which we consider are PDF, Docx, text and PNG, JPEG, and JPG.

4.3.1 Testing of Detection Technique on PDF File

We have taken a sample PDF file with the name 1.pdf. Initially, the secure vault is empty. We pasted the 1.pdf file in a secure vault and our surveillance code did not give any alarm as shown in figure 4.7. However, the code has detected the addition of 1 pdf file in a secure vault. The surveillance code is monitoring all files with a span of 5 seconds. Any addition or modification is detected by code in 5 seconds.

Now we will infect 1.pdf file with ransomware. The file name with 1_encrypted.pdf is an encrypted file. The result is shown in figure 4.8. The surveillance code successfully detected the 1_encrypted file. The code uses the file open checks of python 3 to check whether the files are accessible or not. As shown in the figure the infected version of 1.pdf is not accessible which results in the generation of alarm and message “Issue in file 1_encrypted”.

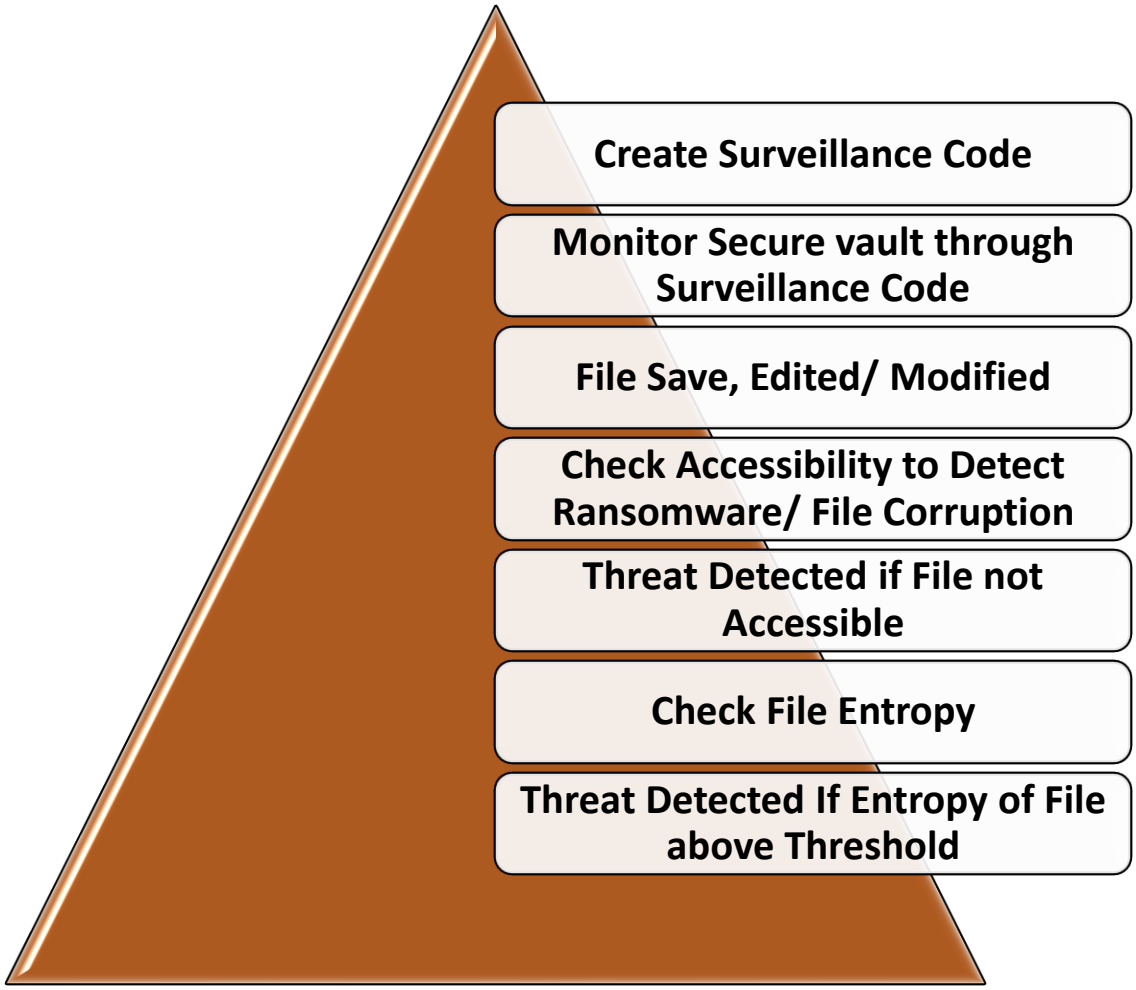

Figure 4.6 Flow Diagram of Detection Technique

```
********** Current status **********
Vault path: /home/fazal/ransomware_vault/secure_vault
Existing files: ['1.pdf']
Check at: Sat, 11 Dec 2021 15:10:46 PKT



********** Current status **********
Vault path: /home/fazal/ransomware_vault/secure_vault
Existing files: ['1.pdf']
Check at: Sat, 11 Dec 2021 15:10:51 PKT
```

Figure 4.7 Result of 1.pdf file

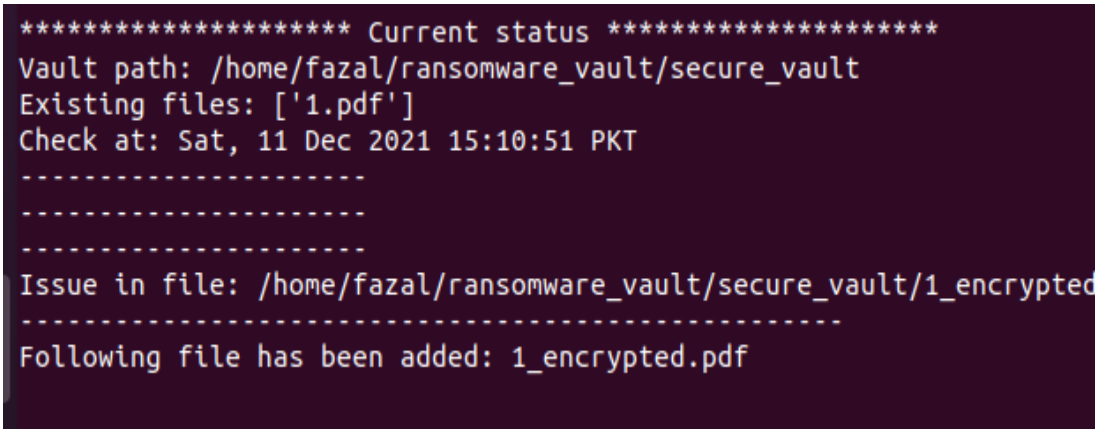

Figure 4.8 Result of 1_encrypted.pdf file

Now we will detect the change in entropy of a file. We will paste a simple pdf file "1.pdf". The result is shown in figure 4.9. The entropy of clean pdf is shown in figure 4.4 that is 4.28284. Now we will infect the same file and measure its entropy of it.

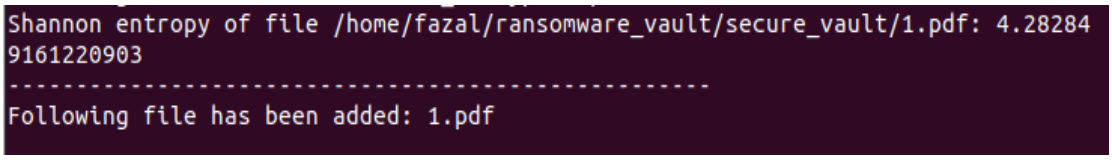

Figure 4.9 Shannon Entropy of 1.pdf file

The entropy of the infected file is increased from 4.28284 to 5.99999. As previously explained that ransomware results in an increase in randomness which ultimately results in higher entropy. As shown in figure 4.10, the surveillance code successfully detected the issue in a file.

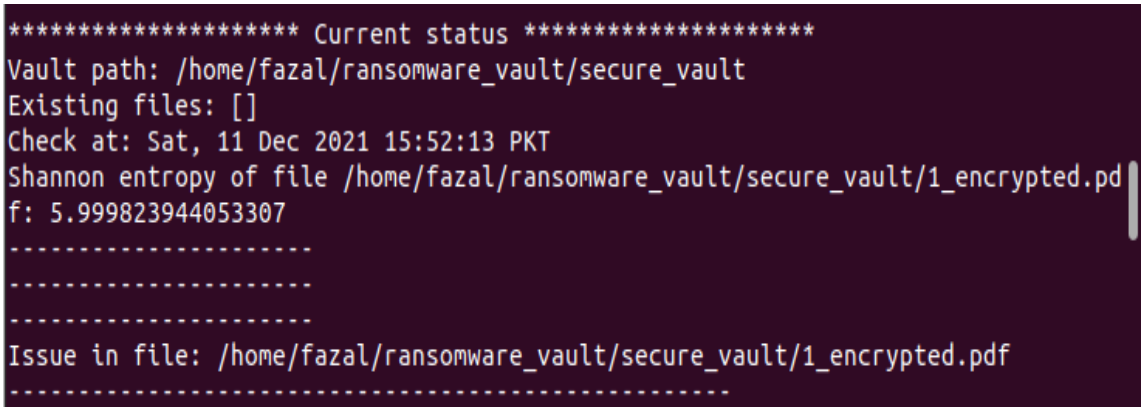

Figure 4.10 Shannon Entropy of 1_encrypted pdf file

4.3.2 Testing of Detection Technique on Docx File

Now after applying our detection algorithm on PDF files we will test this detection technique in docx file format. We will place a docx file name sample.docx in a secure vault. As shown in figure 4.11 the surveillance code detected the addition of a file named simple.docx. The detection algorithm does not generate any alarm on the addition of this file.

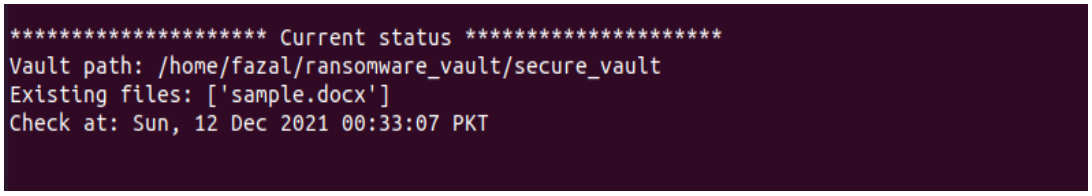

Figure 4.11 Result of sample.docx file

Now adding the infected version of the same file sample.docx in a secure vault. The infected sample. docx-encrypted is an encrypted file. The detection algorithm has detected the intrusion/ ransomware attack using file open checks and generates an alarm and message “issue in file sample-encrypted” as shown in figure 4.12.

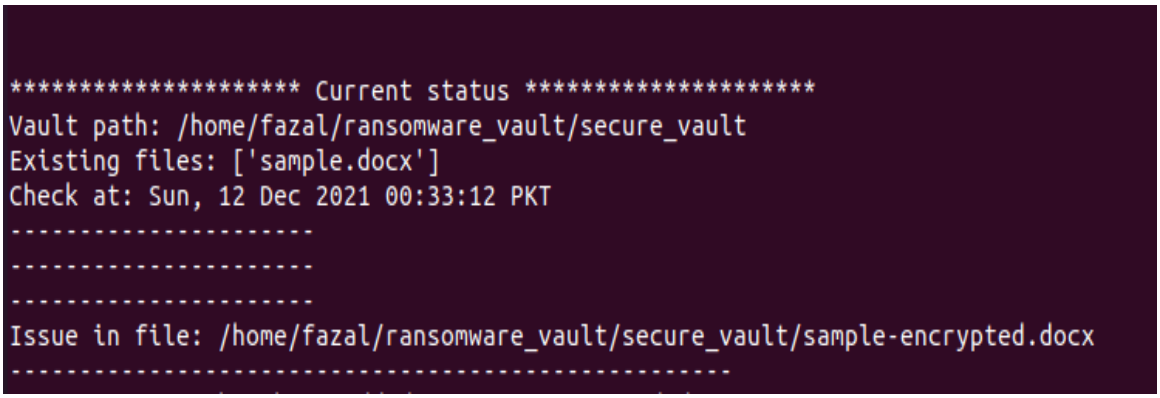

Figure 4.12 Result of sample-encrypted.docx file

Now we will measure the difference between the entropy of clean and infected files. First, the entropy of sample.docx will be measured by inserting the same file in a secure vault. The result is shown in figure 4.13. The Shannon entrput of sample.docx file is 3.9931. Now by infecting the same file by ransomware the entropy is again measured.

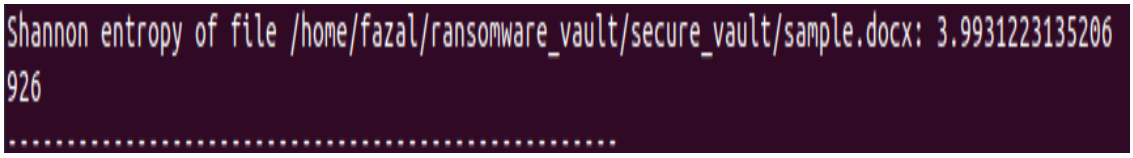

Figure 4.13 Shannon Entropy of sample.docx file

The entropy of the infected file is increased from 3.9931 to 5.99999. As previously explained that ransomware results in an increase in randomness which ultimately results in higher entropy. The infected file has a high entropy value than the original file and this fact is used to detect the ransomware attack. As shown in figure 4.14, the surveillance code successfully detected the issue in a file.

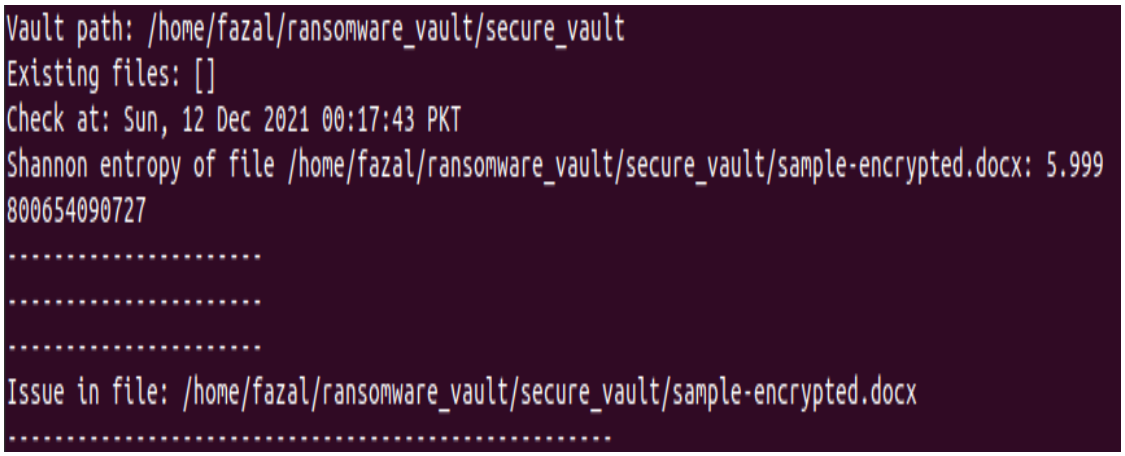

Figure 4.14 Shannon Entropy of sample-encrypted.docx file

4.3.3 Testing of Detection Technique on PNG File

After applying the detection algorithm on PDF and Docx format, now we will check the detection technique on the image having Png format. First as per procedure in the vogue Png file “picture 1” is copied to the secure vault. As same, the surveillance code is monitoring the secure vault and has detected the addition of picture 1 as shown in figure 4.15.

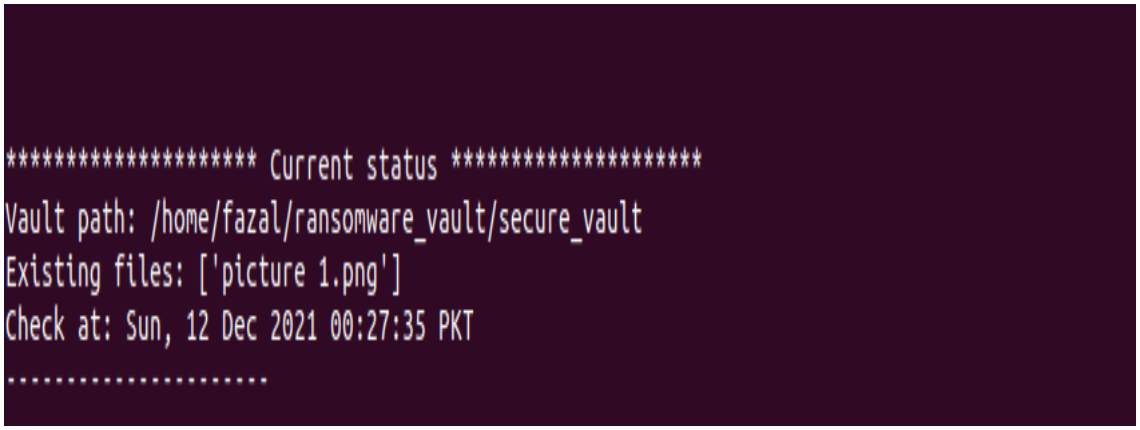

Figure 4.15 Result of picture 1.png File

Now moving towards the infected picture1.png will be placed in a secure vault. The algorithm will use its file open checks of image formats. The file is not accessible which results in the generation of alarm and message as shown in figure 4.16.

Figure 4.16 Result of picture 1_encrypted.png File

Now testing the second detection check of algorithm e.g measuring the difference in entropy of clean and infected files by ransomware. For testing purposes, we will place png file named" picture 1" in a secure vault. The algorithm will detect its addition and measure its entropy using Shannon entropy. As shown in figure 4.17 the file has been detected and the entropy of the clean file is 3.8711.

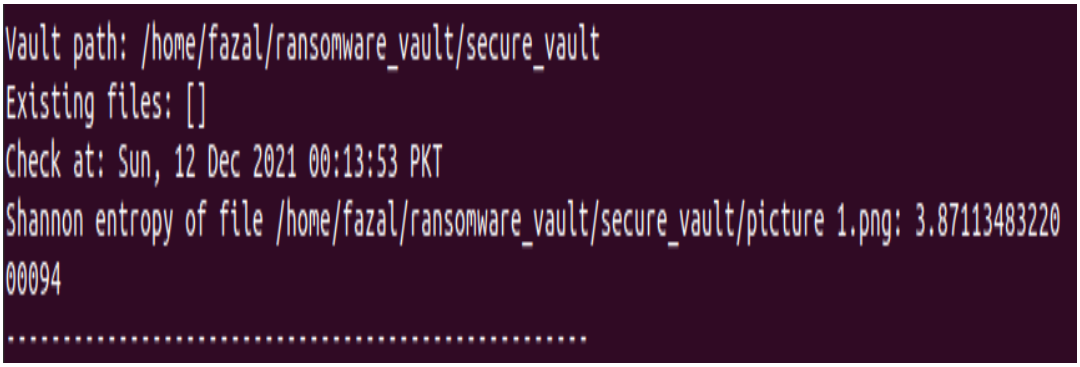

Figure 4.17 Shannon Entropy of picture 1.png

Now by infecting the picture 1.png file with ransomware and measuring its entropy as shown figure 4.18. The Shannon entropy of picture 1_encrypted.png is 5.9981. This increase in entropy is due to more randomness in a file. The algorithm successfully detected the ransomware and generate an alarm.

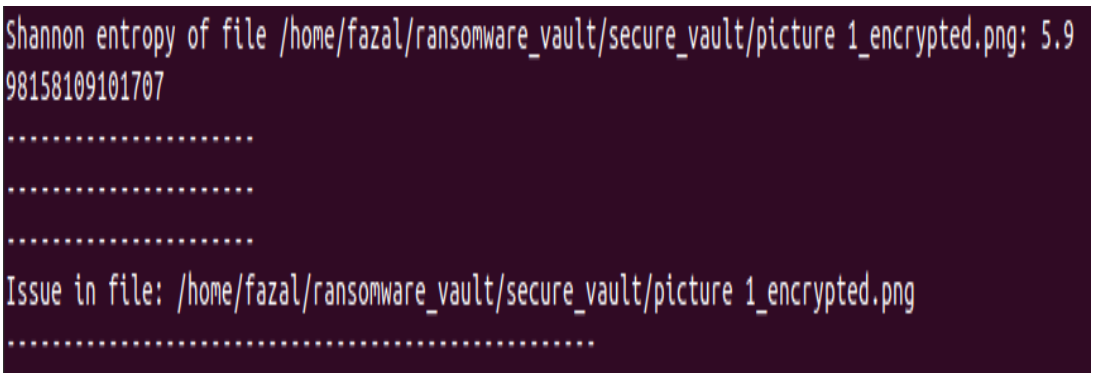

Figure 4.18 Shannon Entropy of picture 1_encrypted.png

Chapter 5

DEPLOYMENT OF DETECTION TECHNIQUE ON CLOUD SERVER AND MEASURING RESOURCE INTENSITY

5.1 Introduction

Keeping in view the importance of cloud technology in the present era there is a significant need for the deployment of proposed detection techniques on a real cloud to thwart the danger of ransomware and its timely detection. The proposed detection technique after being implemented and tested in ubuntu OS will finally be deployed on a cloud environment. Flow chart 5.1 illustrates the deployment of surveillance algorithms in a cloud computing environment.

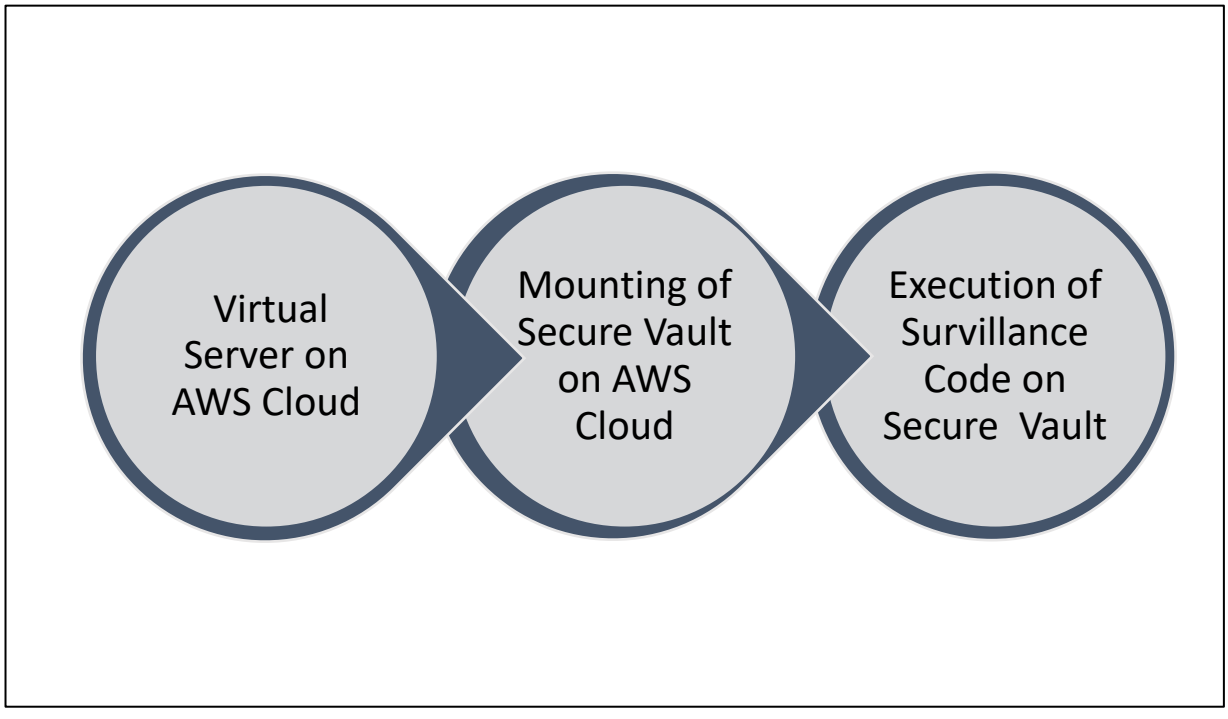

Figure 5.1 Flow Chart Implementation of Surveillance Code on Cloud Computing Environment

5.2 Sequential Phases of Implementation on Cloud

In this section, step for implementation are explain as follows:

5.2.1 Creation of Virtual Server on AWS Cloud

EC2 virtual server on AWS cloud was purchased for implementation of detection algorithm on cloud environment. An instance named “Ransomware instance” was created on which virtual machine was launched for Linux ubuntu OS. Figure 5.2 shows Ransomware Instance on EC2 Virtual Server on AWS Cloud which has a processor of ____ and RAM of ____ GBs. The processing power and RAM of this server were utilized for the execution of the Surveillance Code on the AWS Cloud EC2 virtual server.

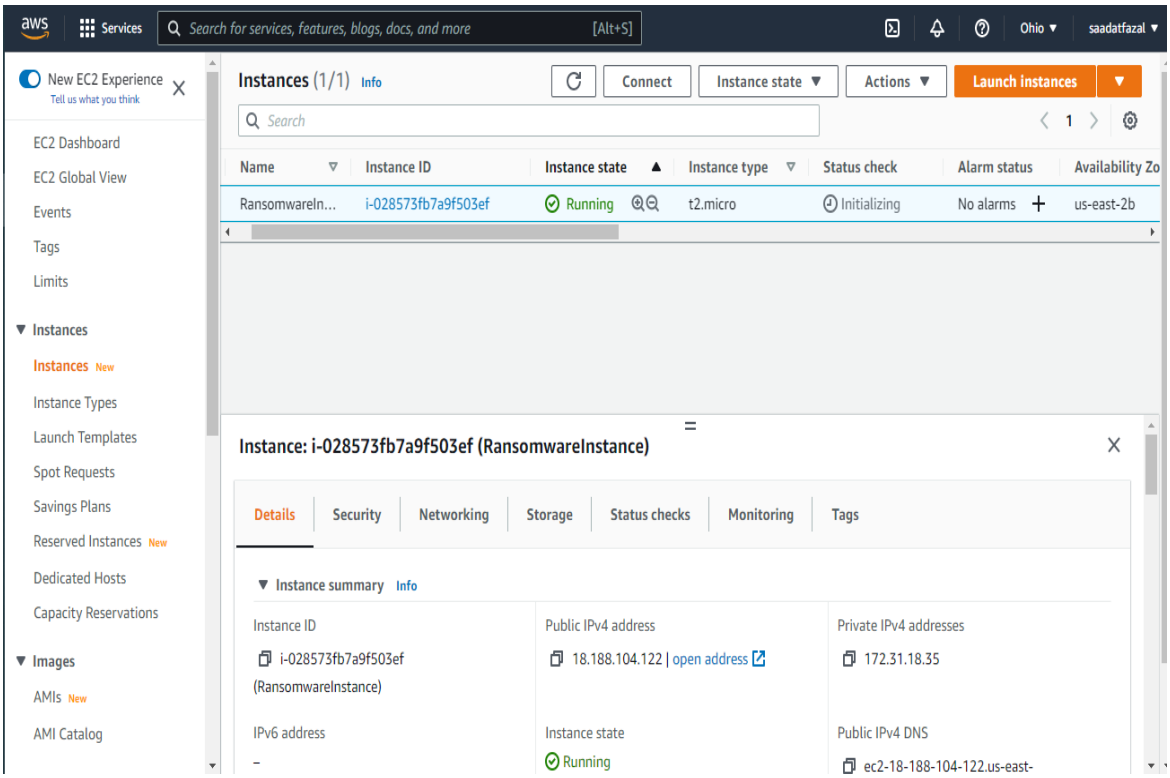

Figure 5.2 Ransomware Instance on EC2 Virtual Server on AWS Cloud

5.2.2 Mounting of Secure Vault on AWS Cloud

For the detection algorithm to work on desired storage, is req to be mounted on AWS cloud. Presently Secure Vault folder locally created on Ubuntu VM is mounted on AWS Cloud. Any

addition of files in a secure vault will automatically be monitored by surveillance code. Figure 5.3 shows that a local folder named secure vault has been mounted on AWS cloud EC2 virtual server. Files added or deleted from the mounted secure vault add or delete on the AWS cloud virtual server. The surveillance Code is then executed on the secure folder to see desired results.

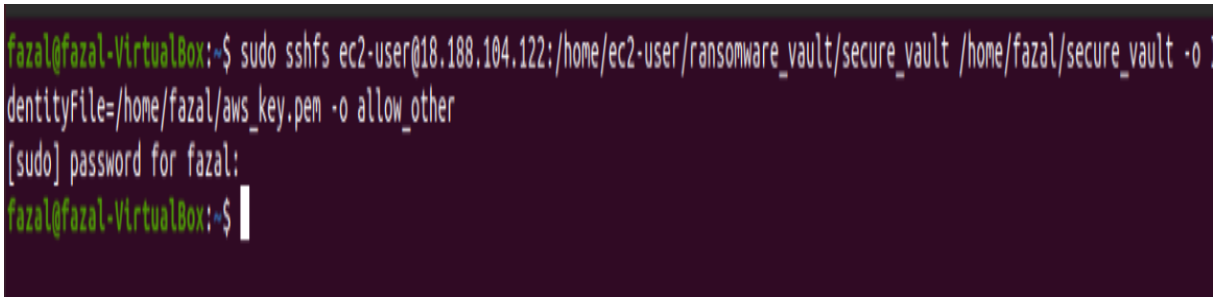

Figure 5.3 Mounting of Secure Vault Folder onto AWS Cloud EC2 Virtual Server

5.2.3 Accessing of AWS Cloud EC2 Virtual Server and Secure Vault

The Cloud server is required to be accessed using the terminal of Ubuntu. Figure 5.4 shows that the AWS Cloud EC2 virtual server is accessed and a secure vault can be opened for further running of surveillance code.

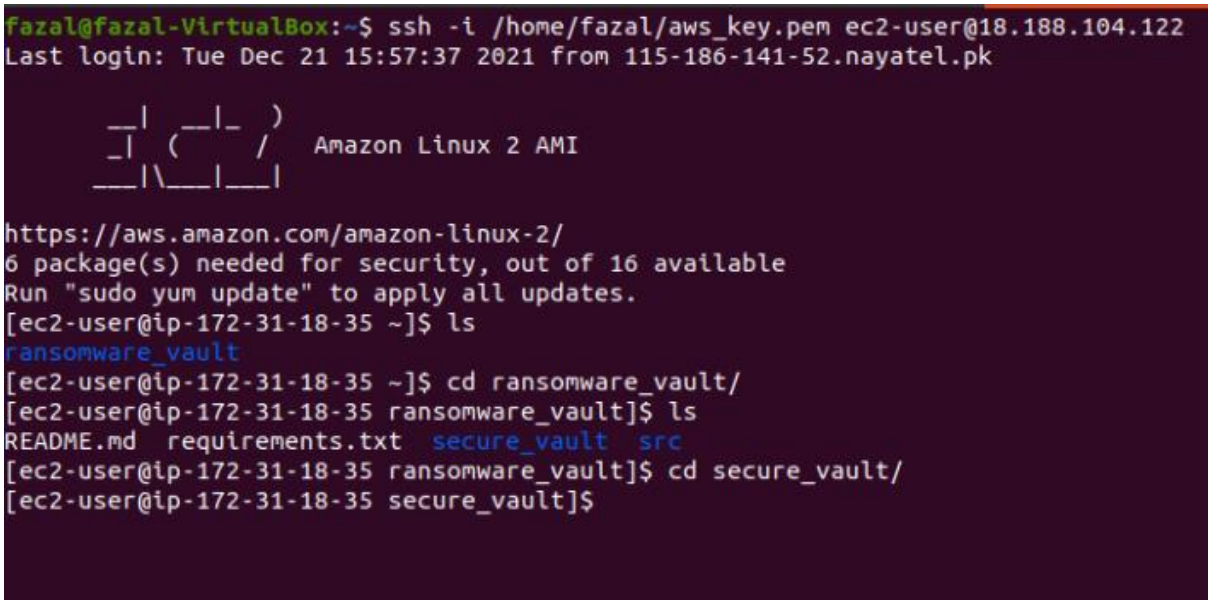

Figure 5.4 Access to Virtual Server AWS Cloud and Secure Vault

5.2.4 Execution of Surveillance Code on AWS Cloud

After mounting the secure vault and accessing the cloud server, surveillance code will be executed now. Figure 5.5 shows that the Surveillance code has been executed on a secure vault. Initially, a secure vault is empty.

Figure 5.5 Execution of Surveillance Code on Secure Vault on AWS Cloud

5.2.5 Effect of Surveillance Code on Original Sample Files

When an original sample file of a pdf file named “2.pdf” is added in the secure vault, the code shows that a 2.pdf file has been added. The surveillance code will, now check the two attributes that use file accessibility using python pdf library and Shannon entropy. Figure 5.6 shows the addition of 2.pdf in secure vault and its subsequent indication by Surveillance Code.

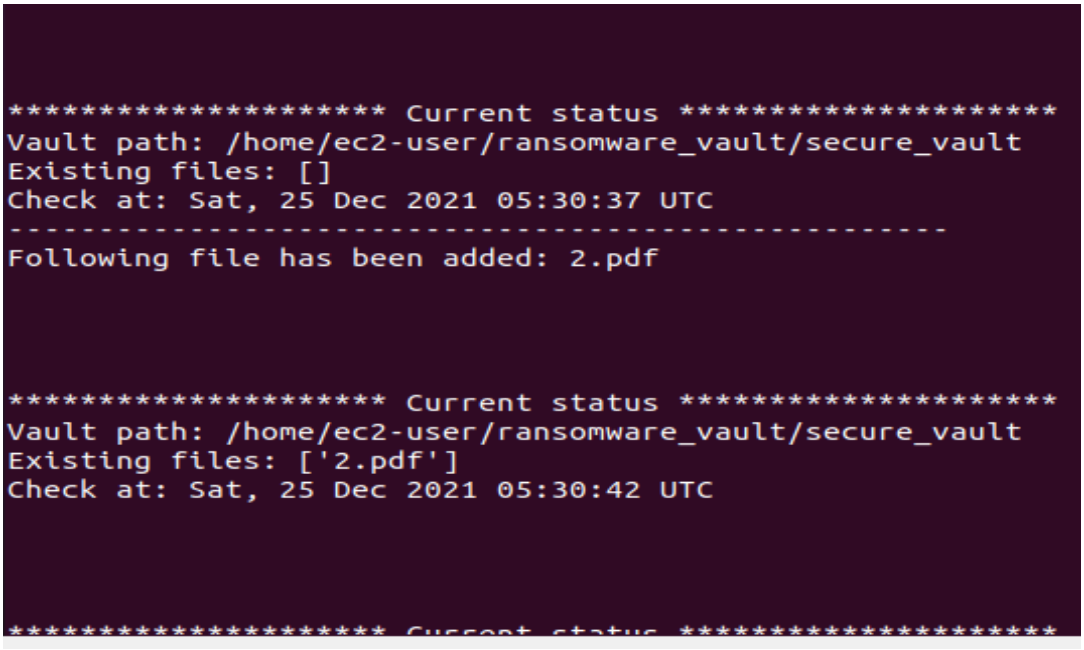

Figure 5.6 Addition of Original Sample File in Secure Vault AWS Cloud

5.2.6 Effect of Surveillance Code on Ransomware Infected Files

For checking the effectiveness of the code we will add Cerber ransomware infected. When this file was added to the secure vault, the surveillance code indicated the addition of a file. The Cerber infected file is checked by the file accessibility libraries of python. As result, these files are not accessible and a message has been displayed that there is an issue in a file. Figure 5.7 shows the addition of ransomware infected files by cerber ransomware and its detection.

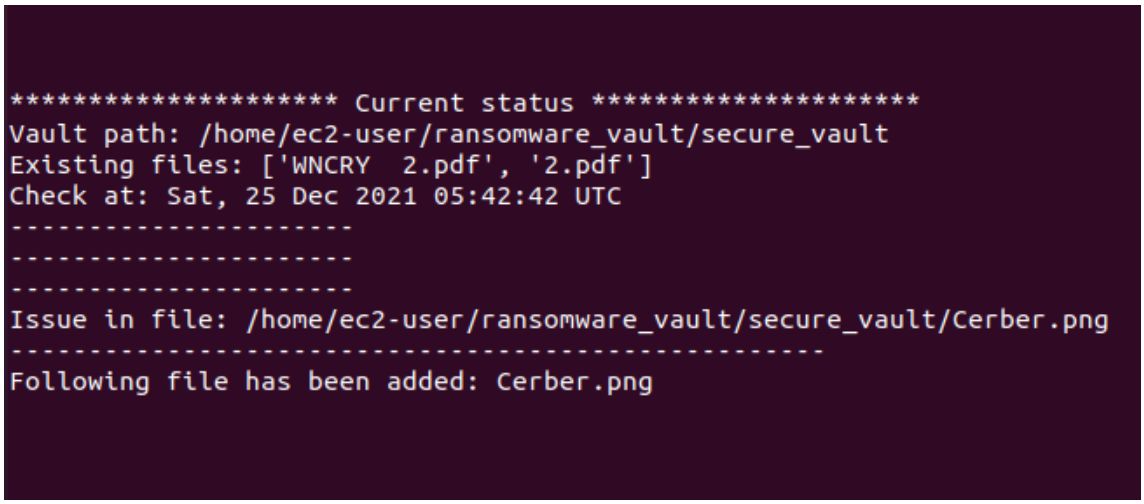

Figure 5.7 Effect of Surveillance Code on Ransomware Infected File on Secure Vault AWS Cloud

5.2.7 Effect of Surveillance Code on Encrypted Files

As previously discussed, ransomware mostly used AES encryption. File name red picture of PNG format has been encrypted using AES encryption. As the file is added to the secure vault the surveillance code gives an alert message of an issue in a file which leads to the detection of ransomware. Figure 5.8 shows the addition of red picture encrypted.png in secure vault and indication of invalid file signature.

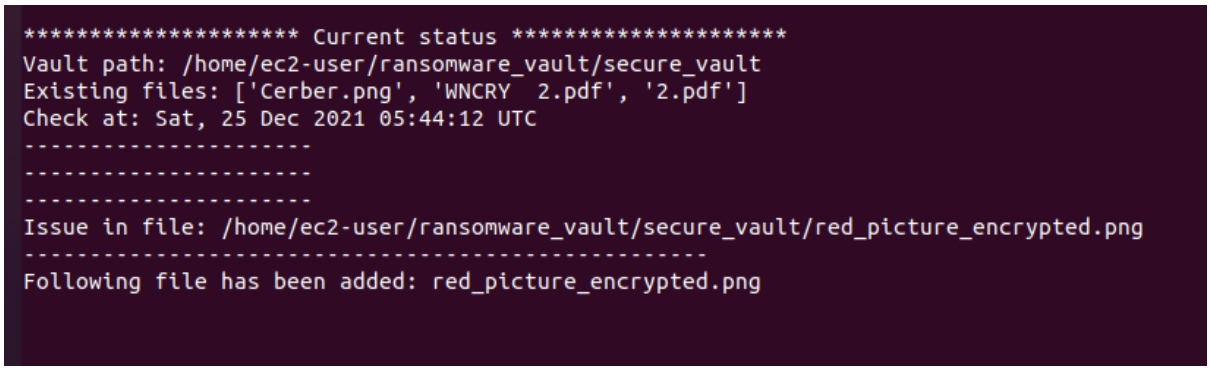

Figure 5.8 Effect of Surveillance Code on Encrypted Files in Secure Vault on AWS Cloud

5.3 Resources intensity of Detection Algorithm

Resource requirement is a significant characteristic of any algorithm/ process. To analyze the processing and memory consumption during execution of Surveillance Code on Secure vault on AWS Cloud, we first checked the total processing power and RAM available for EC2 virtual server to run the surveillance code. The processing power available is 1 thread, 1 core, 2.4 GHz ES-2676 Intel CPU, and RAM 965 MB. Figure 5.9 shows the total processing power available and Figure 5.10 shows the total RAM available on AWS Cloud for the EC2 virtual server.

```
    time.sleep(5) # Add a delay in the loop of 5 seconds. Value can be changed as per need
KeyboardInterrupt
[ec2-user@ip-172-31-18-35 ransomware_vault]$
[ec2-user@ip-172-31-18-35 ransomware_vault]$ ls cpu
ls: cannot access cpu: No such file or directory
[ec2-user@ip-172-31-18-35 ransomware_vault]$ lscpu
Architecture:        x86_64
CPU op-mode(s):      32-bit, 64-bit
Byte Order:          Little Endian
CPU(s):              1
On-line CPU(s) list: 0
Thread(s) per core:  1
Core(s) per socket:  1
Socket(s):           1
NUMA node(s):        1
Vendor ID:           GenuineIntel
CPU family:          6
Model:               79
Model name:          Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz
Stepping:            1
CPU MHz:             2300.166
BogoMIPS:            4599.99
Hypervisor vendor:   Xen
Virtualization type: full
L1d cache:           32K
L1i cache:           32K
L2 cache:            256K
L3 cache:            46080K
NUMA node0 CPU(s):   0
```

Figure 5.9 Details of CPU Used by EC2 Virtual Server AWS Cloud

```
[ec2-user@ip-172-31-18-35 ransomware_vault]$ free -m
              total        used        free      shared  buff/cache   available
Mem:            965          92         411           0         461         743
Swap:             0           0           0
[ec2-user@ip-172-31-18-35 ransomware_vault]$
```

Figure 5.10 Details of Memory Used by EC2 Virtual Server AWS Cloud

For analyzing the power consumption of a surveillance code, files of different formats will be added to the secure vault while actively executing of surveillance code. To find the overall power consumption of Surveillance Code i.e. CPU and RAM consumption, we used the top |grep python3 command to access the overall utilization of power by the Surveillance Code monitoring process. Figure 5.11 shows the power consumption with different time intervals and the percentage of CPU and RAM utilized by the Surveillance Code Monitoring process.

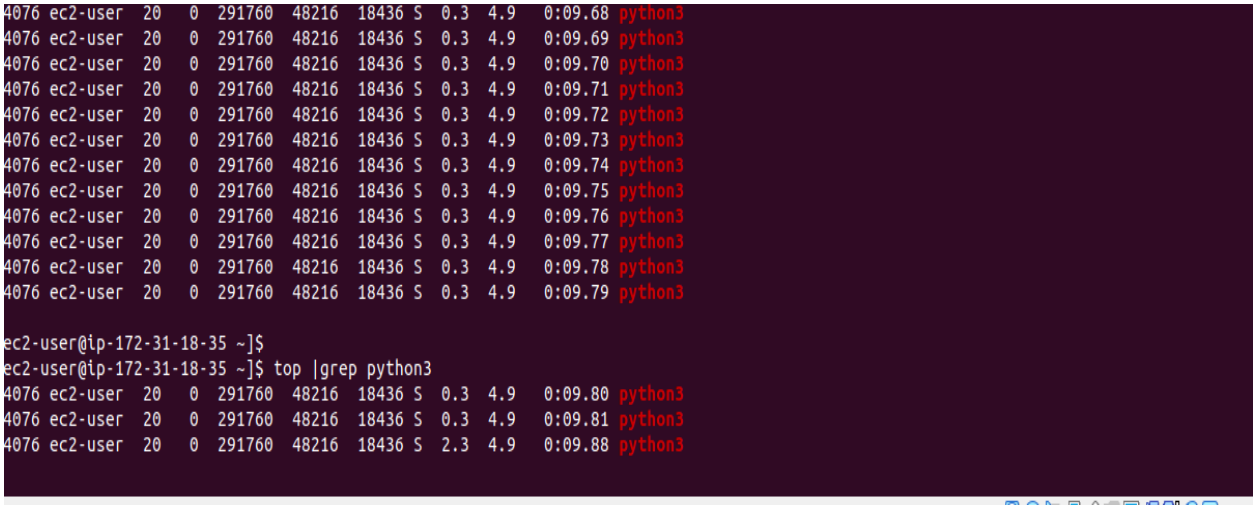

Figure 5.11 Details of Power Consumption by Surveillance Code Monitoring Process

The figure shows that the surveillance code monitoring process is only taking (0.3 - 2.3) of the total processing power and 4.0% of the total RAM while executing the Surveillance code and monitoring files inside the secure vault.

The power consumption is minimal and the surveillance code monitoring process can be easily utilized in a cloud computing environment for effective detection of ransomware.

Chapter 6

RECOMMENDATIONS, CONCLUSION AND FUTURE WORK

6.1 Recommendations

On the Basis of research conducted, following are the proposed recommendations: -

- Ransomware attack makes the file inaccessible for the user. Therefore, it is suggested that more accessibility checks as possible be employed for surveillance code to detect anomaly in file accessibility beforehand.
- Similarly, file entropy of files is altered once ransomware attack is executed. It is suggested that file entropy of as many file formats should be considered as possible for effective detection of ransomware by surveillance code.
- While creating a secure vault folder, it is to be ensured that no other subfolder is created within the secure vault. Our surveillance code detects anomaly only on file structure not on folders. Creation and copying of subfolders in secure vault folder must be disabled for effective detection.
- Copying of files in secure vault require different timings depending on the file size and upload speed of internet service provider. It is suggested that monitoring time of surveillance code should be kept as minimal as possible for effective detection of ransomware keeping in view the processing resources.

6.2 Conclusion

In the current period, information is supposed to be more costly than any of the materialistic things on the planet. The term Cloud Computing has acquired boundless in the course of the most recent few years. For roughly twenty years Cloud processing has had a lead job in its field and a mass part of the business local area is relying upon distributed storage. With the happening to large information and cloud administrations, customer information has

transformed into a huge issue. Although a variety of detection and anticipation advancements are utilized to ensure client data, ransomware that requests money in return for one's data has arisen. Ransomware has various effects on data characteristics e.g. change in entropy, signatures, extensions, encryption, etc. In this research, we have focused on two main attributes of a file that is entropy and file inaccessibility. These two attributes are taken into consideration in our detection algorithm which actively monitor the data saved in a secure vault with a flexible time interval. Initially, the surveillance code was employed in the sandbag environment of a virtual machine. The same surveillance code was then deployed on Amazon Web Server EC2 virtual server to carry out surveillance of shared storage on the cloud for pre-emptive detection of ransomware. The old detection techniques also yield effective detection of ransomware, our detection technique which is a combination of file inaccessibility and file entropy and implemented on cloud computing environment will yield better results. The results of this thesis can help in the formation of future ransomware detection models.

6.3 Future Work

Following are the possible future work: -

- To implement surveillance code on public cloud computing environment.
- To expand the white-listed extensions list with as much file extensions as possible.
- To include almost all file formats and their file signatures for effective detection.
- To execute surveillance code on subfolders created or copied inside secure vault.
- To consider more attributes of file e.g File Signature, Extensions etc.

BIBLIOGRAPHY

[1] https://www.vice.com/en/article/nzpwe7/the-worlds-first-ransomware-came-on-a-floppy-disk-in-1989

[2] https://www.bing.com/search?q=Most+Common+Types+of+Ransomware+%7C+CrowdStrike&cvid=9b8d9c7755ab4dd7ae2c3663fdc50c8e&aqs=edge..69i57j69i60.4638j0j1&pglt=43&FORM=ANNAB1&PC=U531

[3] https://www.healthcareitnews.com/news/asia/singaporean-eye-clinic-serving-over-73000-patients-hit-ransomware

[4] https://www.blackfog.com/the-state-of-ransomware-in-2021/

[5] Genç ZA, Lenzini G, Sgandurra D. On deception-based protection against cryptographic ransomware. InInternational conference on detection of intrusions and malware, and vulnerability assessment 2019 Jun 19 (pp. 219-239). Springer, Cham.

[6] Jeon WJ, Choi SH, Park KW. Detecting Ransomware with File System-Awareness Scheme in Cloud Computing Environment.

[7] Feng Y, Liu C, Liu B. Poster: A new approach to detecting ransomware with deception. In38th IEEE Symposium on Security and Privacy Workshops 2017 May (p. 39).

[8] Moore C. Detecting ransomware with honeypot techniques. In2016 Cybersecurity and Cyberforensics Conference (CCC) 2016 Aug 2 (pp. 77-81). IEEE.

[9] Khammas BM. Ransomware Detection Using Random Forest Technique. ICT Express. 2020 Dec 1;6(4):325-31.

[10] Kim DY, Choi GY, Lee JH. White list-based ransomware real-time detection and prevention for user device protection. In2018 IEEE International Conference on Consumer Electronics (ICCE) 2018 Jan 12 (pp. 1-5). IEEE.

[11] Jeon WJ, Choi SH, Park KW. Detecting Ransomware with File System-Awareness Scheme in Cloud Computing Environment.

[12] Moussaileb R, Bouget B, Palisse A, Le Bouder H, Cuppens N, Lanet JL. Ransomware's early mitigation mechanisms. InProceedings of the 13th International Conference on Availability, Reliability and Security 2018 Aug 27 (pp. 1-10).

[13] Ahmadian MM, Shahriari HR, Ghaffarian SM. Connection-monitor & connection-breaker: A novel approach for prevention and detection of high survivable ransomware. In2015 12th International Iranian Society of Cryptology Conference on Information Security and Cryptology (ISCISC) 2015 Sep 8 (pp. 79-84). IEEE.

[14] Ahmadian MM, Shahriari HR. 2entFOX: A framework for high survivable ransomwares detection. In2016 13th international iranian society of cryptology conference on information security and cryptology (ISCISC) 2016 Sep 7 (pp. 79-84). IEEE.

[15] Davies SR, Macfarlane R, Buchanan WJ. Differential Area Analysis for Ransomware Attack Detection within Mixed File Datasets. Computers & Security. 2021 Jun 19:102377.

[16] Cohen A, Nissim N. Trusted detection of ransomware in a private cloud using machine learning methods leveraging meta-features from volatile memory. Expert Systems with Applications. 2018 Jul 15;102:158-78.

[17] Cohen A, Nissim N. Trusted detection of ransomware in a private cloud using machine learning methods leveraging meta-features from volatile memory. Expert Systems with Applications. 2018 Jul 15;102:158-78.

[18] Scaife N, Carter H, Traynor P, Butler KR. Cryptolock (and drop it): stopping ransomware attacks on user data. In2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS) 2016 Jun 27 (pp. 303-312). IEEE.

[19] Hwang J, Kim J, Lee S, Kim K. Two-stage ransomware detection using dynamic analysis and machine learning techniques. Wireless Personal Communications. 2020 Jun;112(4):2597-609.

[20] Mbol F, Robert JM, Sadighian A. An efficient approach to detect torrentlocker ransomware in computer systems. InInternational Conference on Cryptology and Network Security 2016 Nov 14 (pp. 532-541). Springer, Cham.

[21] https://www.salesforce.com/ap/products/platform/best-practices/benefits-of-cloud-computing/

[22] https://www.checkpoint.com/cyber-hub/threat-prevention/ransomware/

[23] https://www.kaspersky.com/

[24] https://www.checkpoint.com/cyber-hub/threat-prevention/ransomware/

[25] https://www.checkpoint.com/cyber-hub/threat-prevention/ransomware/

[26] https://www.checkpoint.com/cyber-hub/threat-prevention/ransomware/

[27] https://www.mcafee.com/en-us/index.html

[28] Jethva B. A new ransomware detection scheme based on tracking file signature and file entropy (Doctoral dissertation).

[29] https://www.kaspersky.com/resource-center/threats/ransomware-attacks-and-types