Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering

VIRUS INTRUSION AND DETECTION SYSTEM USING ANTIVIRUS

CSE3501
Information Security Analysis and Audit

JOSEPH T JOHN	20BDS0140
KARTHIK KISHORE	20BCT0172
RINOSH K	20BCI0203

Under the guidance of

Prof. Saritha Murali

21st October 2022

Table of Contents

1. PROJECT DESCRIPTION

1.1. Abstract

People use computers for all kinds of activities: online gaming, shopping, entertainment, emails, social media, study, research, etc. At the same time, the risk of infection by malicious programs in these computers is also on the rise. The main issue here is that the general users usually don't understand what a virus is and how easily computers can get infected. Although these days, there are many vendors that produce antivirus software with different features to prevent or remove these viruses, general users end up not understanding the concept of the features provided in these programs.Additionally there is no tool to advise users about what the features mean and help them select the right software for personal or business needs. The purpose of this project is to create an antivirus system with various tasks and features that would provide better information to the users on how to tackle these situations in the current digitally enhanced world. A virus program would also be built in order to showcase all the important aspects of the antivirus program.

1.2. Motivation

Infection proliferation on the Internet has brought about huge misfortune and security breaks. Although huge examination exertion has been spent on creating antivirus programming devices, the engendering elements of infection and antivirus aren't completely researched.

Most antivirus software is difficult to understand how it works because of the absence of GUI. But our proposed project aims to make use of a simple GUI so that the working of the antivirus can be easily understood.

The principal issue is that overall clients don't comprehend what an infection is and how PCs get contaminated while they perform simple tasks like gaming, shopping, study, research, etc.

2. INTRODUCTION

2.1. Scope

In the world of computers and the internet, it is important to verify what you do unless you want to get infected with a malicious piece of code that may hinder performance or just turn your computer or device into a spying system. To understand more let us get formal definitions for viruses and their nemeses' anti-viruses. A virus is a malicious code that is loaded onto your device with the intent to cause damage and steal information. Computer viruses replicate themselves and occupy all the available memory and result in system damage. Some viruses can replicate and pass on their copies across various networks and bypass security systems as well. To protect your computer or network– an antivirus program is needed. A generic antivirus software scans identify and remove viruses, computer worms, Trojans, etc. Most antivirus programs are capable of an auto-update feature to stay up-to-date with new virus definitions that are released into the world. They offer on-demand and on-access scanning options and choice varies from user to user. Here we aim at simulating a virus attack, its prevention, and its cure.Some goals that we intend to carry out are as follows:

- To understand intrusion detection systems and their functioning.
- To build a virus that replicates itself in all the .py files present in the same directory.
- To build a system to scan for the presence of a virus in any file using various methods.
- To build an antivirus program that can be used to stop the intrusion caused by the replicating virus.

2.2. Purpose

Antivirus softwares are generally designed to find known viruses and oftentimes other malware such as Ransomware, Trojan Horses, worms, spyware, adware, etc., that can havea detrimental impact on the user or device.

Antivirus programs provide a way to protect one's device against known threats. The effectiveness of an antivirus program is heavily dependent on how often it is updated. Therefore, it is important to have the antivirus program scheduled to update daily. Most antivirus programs rely on a library or database of known viruses that they use to compare with programs on a user's device. If a match is found, the malicious program will either be deleted or placed into a quarantine area from which a user can decide to restore or delete the program manually. With an antivirus program configured with regular updates and scans, users should feel safe from known threats. Antivirus programs are a key part of a user's total cybersecurity hygiene practice.

3. LITERATURE SURVEY

Sr. no.	Paper Title	Name of the Conference/Journal with year	Methodology proposed	Pros	cons
1	**An Analysis of Various Anti VirusSoftware Tools Based On Different Effective Parameters**	**International Journal of Computer Science Trends and Technology (IJCST) – Volume 4 Issue 4, Jul - Aug 2016**	For this research work user feedback form is given to fifty users in various categories like students, employees and hardware service engineers in Thanjavur District. This questionnaire was given to ten groups of people, each group using the same antivirus software for their machines. They gave maximum 10 points for performance, features and help and support for their used software tools.	This paper gives an answer to which antivirus software has the best features overall. In this comparison, Kaspersky is the best antivirus tool in terms of performance, Norton is the best in terms of features, and Kaspersky is the best in terms of help and support.This paper hence provides details on what a good antivirus software should contain.	However this paper only has a comparative study about these preexisting softwares.

2	**Introduction to Malware and Malware Analysis**	**November 2016 International Journal of Advance Research in Computer Science and Management Volume 4**	Norman SandBox: The Norman SandBox Analyzer is a utility meant to automate, simplify, and speed up the information gathering process when analyzing malware Anubis: Anubis is developed by the International Secure Systems Lab and is capable of analyzing both files and URLs. CWSandbox: CWSandbox is a tool for malware analysis that satisfies the three design conditions of automation, effectiveness and correctness. Dynamic analysis of malware is done to achieve automation.	The concepts of malware, the many varieties of malware, and malware analysis have all been thoroughly explored. Dynamic analysis is a superior way of malware analysis than static analysis, according to the data gathered. Although dynamic analysis has the obvious issue of studying only one virus operation, static analysis is more difficult to accomplish well because the source code is usually not exposed.	Recent trends in malware attacks show highly advanced techniques being applied to secure sensitive information. Hence updation is required in the preexisting antivirus solutions

3.	**The rise of machine learning for detection and classification of malware: Research developments, trends and challenges**	**Journal of Network and Computer Applications** **Volume 153, 1 March 2020,**	It presents a systematic review of M.L. approaches for malware detection. Traditional approaches are classified into static, dynamic and hybrid approaches. It provides a detailed description of the features in a traditional M.L. worflkow.It introduces new research directions such as deep learning and multimodal approaches. It discusses the research issues and challenges faced by security researchers.	This paper provides a comprehensive review of machine learning-based malware detection and classification techniques. In total, 67 research articles addressing the topic of malware detection and classification on the Windows platform were examined. The studies are compared and examined based on a variety of important aspects such as input features, classification algorithms, dataset characteristics, and the objective task.It provides new research directions and classifiers that detect malware using more than one sort of feature or data	Most of the new methodologies proposed are quite expensive to implement.

				modality.It also addresses the most pressing research issues and challenges that researchers face.	
4.	**Classification Of Malware Detection Using Machine Learning Algorithms: A Survey**	**INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 9, ISSUE 02, FEBRUARY 2020**	This paper presents a survey about existing literature on malware analysis using different machine learning algorithms. The different literature of existing works is listed in table form, with what tools were used in their work, what machine learning algorithms were used in their work, from what sources dataset was collected, what parameters they considered to reach their goal and the corresponding experimental results, and what future works are proposed.	It has a comprehensive review on the various pre existing AV solutions	It does not provide new research directions. It does not address the major issues faced by the researchers currently.

5.	**Internet of Things Cyber Attacks Detection using Machine Learning**	**Jadel Alsamiri, Khalid Alsubhi 2021 International Journal of Advanced Computer Science and Application**	Multiple algorithms are tested on the dataset for maximum accuracy such as K nearest neighbors, Random Forest, Multi layer Perceptron and Naïve Bayes on the Bot-IoT metadata is best dataset is the best dataset for the experiments because of wide attack diversity, regular updates, the capability to make distinct points from the fresh dataset, and the addition of IoT-generated network traffic. Bot-IoT metadata contains triply types of cyber attacks just like DoS, Probing, and Data Theft	Machine Learning algorithms are proposed to secure the data from cyber security risks. Machine-learning algorithms can apply in different ways to limit and identify the outbreaks and security gaps in networks. The major objective of this article is to explore the efficacy of machine learning (ML) algorithms in combating network-related cyber security assaults, with an emphasis on DoS attacks.	1. Using signature based approaches requires frequent manual updates of attack traffic signatures and that these approaches cannot detect previously unknown Attacks 2. The main disadvantage of utilising unsupervised machine learning methods for detection problems is that most network traffic flows are regular, and anomalies such as assaults and outliers are uncommon, lowering success rates and making anomaly detection more difficult.
6.	**Modeling virus and antivirus spreading over hybrid wireless ad hoc and wired networks.**	**Zhang, X. and Tadi, K.C., 2007, IEEE GLOBECOM 2007-IEEE Global Telecommunications Conference (pp. 951-955). IEEE.**	Infection proliferation in the Internet has brought about huge misfortune and security breaks. Albeit huge examination exertion has been spent on creating	In this paper, we display and break down the spreading qualities of infections as existing together with the counter infection spreading	The spreading elements of most infections are network geography subordinate making the examination a difficult issue.

			antivirus programming devices, the engendering elements of infection and antivirus isn't completely researched. Both infection and antivirus have comparative spread attributes somewhat balancing one another.	measure in a changed two-layer little world geography for half and half remote specially appointed and wired organizations. We reenact our proposed infection and antivirus model over the crossover remote specially appointed and wired organizations and come to end result which can be utilized by for creating practical antivirus arrangements	
7.	**Computer virus and antivirus software a brief review**	**Patil, B.V. and Jadhav, R.J., 2014. International Journal of Advances in Management and Economics, 4(2), pp.1-4.**	A PC infection is programming deliberately written to duplicate itself without the PC proprietor's authorization and afterward play out some other activity on any framework where it dwells. Presently a days, infections are being composed for pretty much every figuring stage Antiinfection assurance is, or ought to be, a basic piece of any Information Systems activity, be it individual or expert. There are number of PC infection are made	The enormous number of Anti-infection programming accessible on the lookout and some are being dispatched, each one of them offers new highlights for recognizing and destroying infections and malware. Individuals regularly change they're Against infection programming as indicated by their enjoying and needs without assessing the presentation and capacities of the different Anti-	it does not applied for advance virus infections.

			and these PC infections are influenced in day today life.	infection programming accessible. This exploration paper features the basic ideas of PC infections and antivirus programming. And furthermore, portray the subtleties sorts of PC malware or malicious code and working of antivirus programming	
8.	**State-based cache for antivirus software**	**Nachenberg, C.S., Symantec Corp, 1998. State-based cache for antivirus software. U.S. Patent 5,854,916.**	A PC actualized technique for executing a PC document in a CPU emulator to identify a PC infection.	The strategy incorporates mimicking the execution of a foreordained number of directions of the PC document in the CPU emulator, suspending the execution, building a state record, incidentally putting away the state record in memory, contrasting the built state record to state records put away in a state reserve, and demonstrating that the record is sans infection when the developed state record matches one of the put away state records.	took longer time in detection.

9.	**Anti-virus method, computer, and recording medium**	**Kuwamura, S.Y., Fujitsu Ltd, 2012. Anti-virus method, computer, and recording medium. U.S. Patent 8,176,558.**	In one PC framework, causing the second virtual machine, which executes antivirus programming for distinguishing and eliminating the infection, to screen in any event one first virtual machine that is made on the PC and execute at least one application program, occasionally putting away a condition of the primary virtual machine as preview, suspending the main virtual machine from which the infection is recognized if the antivirus programming executed on the second virtual machine identifies the infection, and reestablishing the principal virtual machine at a condition of a point in time when the depiction is put away by utilizing the preview of the suspended first virtual machine	The object and advantages of the invention will be realized and attained by means of the elements and combinations particularly pointed out in the claims. It is to be understood that both the forgoing general description and the following detailed description are exemplary and explanatory and are not restrictive of the invention, as claimed.	This embodiment relates to a technique for taking measures against the invasion of a virus into a computer having virtual machines. This infected virtual machine can infect the original computer.

10.	**The design and implementation of an antivirus software advising system.**	**Chamorro, E., Han, J. and Beheshti, M., 2012, April, In 2012 Ninth International Conference on Information Technology-New Generations (pp. 612-617). IEEE.**	General clients don't comprehend the idea of each component in these projects, nor is there an instrument to prompt clients about what the highlights mean and assist them with choosing the correct programming for individual or business needs.	help general clients to learn about PC infection and antivirus, comprehend the different highlights of antivirus programming items, and select right antivirus programming to secure their PCs	General clients don't comprehend the idea of each component in these projects, nor is there an instrument to prompt clients about what the highlights mean and assist them with choosing the correct programming for individual or business needs.
11.	**A Hybrid Intrusion Detection System Based on Decision Tree and Support Vector Machine**	**Anku Kumari, Ashok Kumar Mehta, Dec 2020 Department of Computer Applications National Institute of Technology Jamshedpur, India**	To examine the network by collecting an adequate amount of data and detecting sensor nodes' abnormal behavior. A hybrid system is suggested in this paper. The hybrid system is a combination of two approaches, so the combination of two approaches covers up for the imperfections in each. A hybrid system is a way of integrating more than two different classification algorithms to estimate the best accuracy result. In this paper, the Voting method with the combiner rule	Intrusion Detection System is a security software that continuously analyses the network traffic and generates an alert signal when any suspicious event occurs. It examine the network by collecting an adequate amount of data and detecting sensor nodes' abnormal behavior.	1. Handling the complexity of the newly generated model. 2. Proper integration with already existing datasets.

			of a product of probability is used to integrate the J48 Decision Tree and Support Vector Machine and to estimate the result. The proposed model showed 99.6% highest accuracy and least false alarm rate 0.9% based on different ratios		
12.	**A study of cyber security challenges and its emerging Trends on Latest Technologies**	**G.NIKHI TA REDDY , G.J.UGA NDER REDDY 2018 B.E, CSE second year at Chaitanya Bharathi Institute of Technology, Osmania** **University, Hyderabad., India B.E, M.B.A. and Founder Director - Peridot Technologies, Hyderabad, India**	1. Web servers 2. Cloud computing and its services 3. APT's and targeted attacks 4. Mobile Networks	data id being transmitted or sent to the other person safely without any leakage of information. government s are now enacting strong cyber security legislation in order to avoid the loss of sensitive data. Every individual must be educated on cyber security in order to protect oneself from the growing number of cyber crimes.	Computer security is a vast topic that is becoming more important because the world is becoming highly interconnected, with networks being used to carry out critical transactions. Cyber crime continues to diverge down different paths with each New Year that passes and so does the security of the information.

13.	**Testing and evaluating virus detectors for handheld devices**	**Morales, J.A., Clarke, P.J., Deng, Y. and Kibria, B.G., 2006. Testing and evaluating virus detectors for handheld devices. Journal in Computer Virology, 2(2), pp.135-147.**	This examination assesses four at present accessible antivirus answers for handheld gadgets. A conventional model of infection change that gives change discernibility is introduced. Two arrangements of ten tests each were managed; nine tests from each set included the change of source code of two known infections for handheld gadgets.	The testing methods utilized are settled in PC testing; along these lines the focal point of this examination is exclusively on handheld gadgets.	Factual examination of the test outcomes shows high bogus negative creation rates for the antivirus programming and a general bogus negative creation pace of 47.5% with a 95% certainty span somewhere in the range of 36.6% and 58.4%. This high rate shows that current arrangements ineffectively distinguish altered forms of an infection. The infection is left undetected and fit for spreading, tainting and causing harm

14.	**Neural networks for computer virus recognition**	**Tesauro, G.J., Kephart, J.O. and Sorkin, G.B., 1996. IEEE expert, 11(4), pp.5-6.**	they have built up a neural organization for nonexclusive discovery of a specific class of PC infections the alleged boot area infections that taint the boot area of a floppy circle or a hard drive. This is a significant and moderately manageable subproblem of nonexclusive infection location.	Just about 5% of all known infections are boot area infections, yet they represent almost 90% of all infection episodes. they have effectively conveyed their neural organization as a business item, appropriating it to a huge number of PC clients worldwide as a feature of the IBM Antivirus programming bundle	confronted a few difficulties in taking their neural organization from an exploration thought to a business item, shortage of accessible preparing information;
15.	**Modeling and analysis of the effects of antivirus software on an infected computer network.**	**Shukla, J.B., Singh, G., Shukla, P. and Tripathi, A., 2014. Applied Mathematics and Computation, 227, pp.11-18.**	In this paper, a nonlinear numerical model for cleaning a tainted PC network by utilizing antivirus programming is proposed and broke down. In the demonstrating cycle, the allout number of hubs in the organization are isolated in three subclasses, specifically, the quantity of powerless hubs, number of contaminated hubs and the quantity of secured hubs. A variable speaking to the quantity of antivirus programming projects, thought to be	The model is examined by utilizing dependability hypothesis of differential conditions and PC recreation. The investigation shows that it is conceivable to clean the PC network under certain condition which rely on the inflow pace of tainted hubs in the PC organization, the pace of connection of contaminated hubs with vulnerable hubs and their associations with antivirus programming, and so forth It is	the whole organization can be cleaned in the long run if the antivirus programming is applied on the organization

			corresponding to number of contaminated hubs, is likewise considered in the model which interfaces with different hubs bilinearly to direct the cleaning cycle.	discovered that the whole organization can be cleaned in the long run if the antivirus programming is applied on the organization, where a different class of ensured hubs is framed. The PC reenactment affirms the diagnostic outcomes	
16.	**Computer virus strategies and detection methods**	**Al Daoud, E., Jebril, I.H. and Zaqaibeh, B., 2008. Int. J. Open Problems Compt. Math, 1(2), pp.12-20.**	The run of the mill antivirus approach comprises of hanging tight for a number of PCs to be tainted, identifying the infection, planning an answer, furthermore, conveying and sending the arrangement, in such circumstance.	This paper shows that to grow new dependable antivirus programming a few issues must be tackled, for example, another technique to identify all transformative infection duplicates, new dependable observing strategies to find the new infections or joining a computerized signature and an authentication to each new programming	it is very hard to keep each machine from being undermined by infection.

17.	**Aggregation of the knowledge base of antivirus software.**	**Costea, M., Goebel, D.A., Marinescu, A.M. and Thomas, A.F., Microsoft Corp, 2010. U.S. Patent 7,765,400**	Client mode applications, for example, antivirus programming applications, access record framework tasks through a typical data model, which deters the requirement for antivirus programming merchants to make bit mode channels	At the point when record framework tasks are accessible to antivirus programming applications, the current innovation may cause each antivirus programming application introduced on a processing gadget to play out a sweep to decide whether the information is malware	---
18.	**Android Security: A Survey of Issues, Malware Penetration and Defenses**	**2018, Parvez Faruki, Ammar Bharmal, Vijay Laxmi, Vijay Manoj Singh Gaur and Mauro Conti**	Exponential rise in Android malware apps between 2015-2018 Objective defined in the paper: In this review, we discuss Android security enforcement and its issues, Android malware growth timeline between 2015-2018, malware penetration and anti analysis techniques used by malware authors to bypass analysis methods	Application Sandboxing, Permissions at Framework-level, Secure System Partition, Secure Google Play Store, Mandatory ap its Access Control (MAC) policies over the traditional Discretionary fro Access Control (DAC) on the device	Preventing malware entry via applications downloaded ary from Google Play Store, a common market for Mobile Applications

19.	**Malware Security Issues and Solutions**	**FEB 2020, Karthick S, Christ University, and Dr. Sumitra Binu, Christ University**	Types of and Android Attacks: Binu, Permission rsity Escalation Escalation Attack, Permission Attack, Time of Check and Time of Use Attack, Spyware. Attack, Permission Attack, Time of Check and Time of Use Attack, Spyware	Procedure of the proposed model: List all the applications based on its app ID that is its package name. List all the applications for which shared User ID is set. Compare all the applications with every shared User ID set app. List the finalized apps. Provides explicit notification to the user when the shared User ID app tries to access the. permissions with other apps. Display the resources used by shared user ID apps by the security tool app.	---
20.	**Android Security**	**February, 2019,** **Omar M. Ahmed, Duhok Polytechnic University and Amira B. Sallow, Nawroz University**	Information Leakage, Privilege Escalation, Repackaging Apps,Denial of Service Attack, Colluding. Expanding the coverage of malicious application growth, and Android security threats	They have used various static and dynamic methods like: Crowdroid, AndroSimilar, Kirin, Recdroid, Aurasium, Fire Droid, DroidScope, RiskMon, RiskRanker. DroidRanger	→Sometimes, Aurasium is not robust, and has to depend on other software for functioning. →Sometimes, Aurasium is not robust, and has to depend on other software for functioning.

4. PROPOSED WORK

After conducting the research work we have decided to make a project which is able to detect a virus intrusion. We are also planning to make our own replicating virus. In this project we make use of the concept of intrusion detection and intrusion management.

Intrusion is basically some cyber attack, for instance trojans, or a variety of types of phishing attacks,viruses and so on. In this particular project, we will implement the intrusion of the system by a **replicating virus** and the intrusion management done by using an antivirus program.

The intrusion detection part will be done by one part of the antivirus program that scans all the files on the system and checks for the possible presence of any virus, and the management of intrusion, that includes both handling the infected files and also preventing other files from such an attack will be handled by the other part which majorly uses the concept of Quarantine.

In our project, we are going to implement a two stage scanning process followed by quarantining the detected virus file.

We will be doing **the signature scan** first. The signature scan is a type of scan that will search for a particular line character called signature in the whole file. It will read line by line and try to find a signature. If it is found, it will show that the file is infected. Or else it is safe.

The second type of scan that we will be doing is a **heuristic scan**. A heuristic scan is a type of scan which checks for changes in file size. When a virus code replicates itself, the file will get modified. Which also results in a change in the size of the file. The system already has stored the original file size data in a list. After running this scan, it will check for the current file size of all files. If the change is found, it will print the file is infected and also the original and current file data of filename, file size, and time stamp of modification.

Quarantine is a new method used nowadays by antivirus softwares. It is basically a ‘room’ for infected or malicious files. When a file is found dangerous to the system, that file is moved to the quarantine folder. Here the antivirus software will troubleshoot the problem and try to solve it and make the file safe again. Now there are two conditions, if the file is repaired, it can be restored from the quarantine list. And if it is not repaired, for the safety of our system we will have to remove it from our system.

Given below is the pseudocode for the virus and the scanner modules:

VIRUS PSEUDOCODE

```
inVirus=  False
for line in lines:
if("#starting virus code" in line):
inVirus=True;

if(inVirus==True):
virusCode.append(line)

if("^#end of virus code"in line)):
break

for p in programs:
open  file;
read  file;
close file;

#check if the file is already infected
infected= False
for line in programCode:
if(#starting virus code in line);
infected= True
break
#no need to infect it again.
```

```
if not infected:
#newVersion = current version + virus code
newCode.extend(virusCode)

#new version of file. overwrite the original
open file;
write file;
close file;
```

SCANNER PSEUDOCODE

```
signature scan
thisFileInfected=False
for p in programs:
open file;
readlines in file;
close file;

for line in lines:
if("#starting virus code" in line):
found virus;
thisFileInfected = True
if (thisFileInfected == False):
virus not found;

heuristic scanning
#get file data
for p in programs:
get  file  size;
get modified time;
get file name;

programlist=[filename, file size, modified time];
create file filedata.txt;
write prgramlist;


#check for changes
get current programlist;
get original programlist;
```

```
if(filename unchanged):
if(filesize changed or file modified time changed):
print("\nalert!!! File mismatch")
print original programlist;
print modified programlist;
else:
print("file appears to be unchanged")
```

4.1. Information security concept used in this project

In this project, we have made use of the concept of intrusion detection and intrusion management.An Intrusion Detection System (IDS) is a system that monitors network traffic for suspicious activity and issues alerts when such activity is discovered. It is a software application that scans a network or a system for harmful activity or policy breaching. By intrusion, it may mean any attack like an attack of trojans, or a variety of types of phishing attacks and so on.In this project, we have implemented the intrusion of the system by a virus and the intrusion management by using an antivirus program. The intrusion detection part is done by a part of the antivirus program that scans all the files on the system and checks for the possible presence of any virus, and the management of intrusion includes both handling the infected files and also preventing other files from such an attack. Both of these are implemented with the help of the concept of quarantine, which is also further explained below.

An antivirus program may scan data on your computer in two ways. The first method is passive scanning and the second method is active scanning.

Passive scanning

Allowing your antivirus to run in the background is known as passive scanning. If you've ever tried to download a file from the Internet and received a warning about a possible hazard, that's your antivirus working in the background to safeguard your computer. Because your antivirus is open even when you're not using it, it takes more battery power, but it's an apt method to secure your devices without having to do anything yourself**.**

Active scanning

It is different and can be more powerful than passive scanning. Active scanning occurs when you tell your antivirus software to scan your files. Depending on the software, you can choose between a basic scan or a full scan. The difference usually has to do with the depth and breadth of the scan. A basic scan may only scan major files to save time, while a deeper scan will usually take longer as it scans every file on your computer. If an infected file is found, it may be sent to quarantine automatically.

Quarantine is the process of isolating a file that’s suspected of being infected with a virus in order to prevent it from contaminating other parts of your computer.When an antivirus places an infected file in quarantine, it deletes the file from its original location and makes changes to it so that it cannot run as a program.
It then transfers it to a hidden folder that other programs (or yourself as the user) cannot access where it stays until you choose to deal with it. A suspicious file can also be quarantined manually in the rare case that it’s not picked up by your antivirus scan.

4.2. Proposed model

Virus Program:

The virus program will be a program that would infect the victim file and copy its code into the file. This would create a self-replicating virus since the code keeps getting copied over and over.

Antivirus Program:

The program has two ways of detecting the infected file - Simple signature detection & Variation in size difference. The hash signatures are downloaded and updated in the program and further the program has a list of functionalities such as - Scan, Quarantine, Full Scan, etc. that the user can use in order to keep their device secure.

Modules or Tasks:

1. Replicating virus: A virus that infects and copies itself into every file of the folder it is run in.
2. Simple Signature Detection: This is the module where we find out whether a file is infected by any known virus or not.
3. Change in Size Detection: When the virus code copies itself into any file the size of the victim file increases. This is observed in this task.
4. Scan: Any file can be scanned to identify threats using this function.
5. Adding to Quarantine: Any suspected file can be added to the quarantine folder.
6. Deleting files in Quarantine: We can delete the files in quarantine if deemed necessary.
7. Restoring files from Quarantine: We can restore any quarantined file if deemed not harmful or necessary.

Flow diagram:

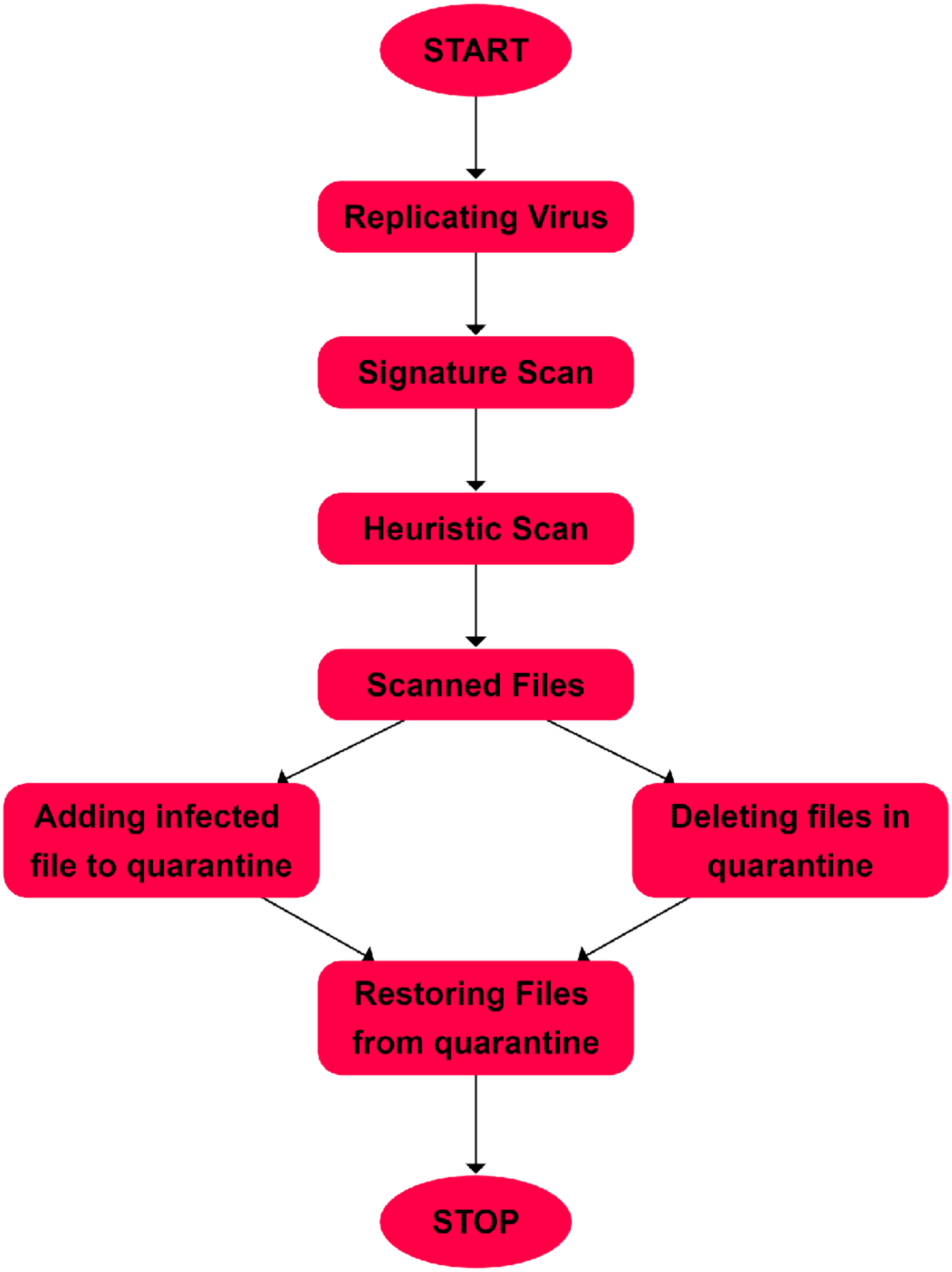

START
Replicating Virus
Signature Scan
Heuristic Scan
Scanned Files
Adding infected file to quarantine
Deleting files in quarantine
Restoring Files from quarantine
STOP

Fig1: Flow chart for the project

In this project we have mainly implemented three modules:

Module 1: Virus Creation

In this module, we have created a sample virus file in python which replicates itself when we run it in other files present in that same directory. It will be done using the append function in python. This virus is basically able to copy its content into other files.

Module 2: Virus scanning

In this module, we will be scanning the virus using python. We are doing the signature scan first. The signature scan is a type of scan that will search for a particular line character called signature in the whole file. It will read line by line and try to find a signature. If it is found, it will show that the file is infected. Or else it is safe. In our project “#starting virus code” is used as a signature.

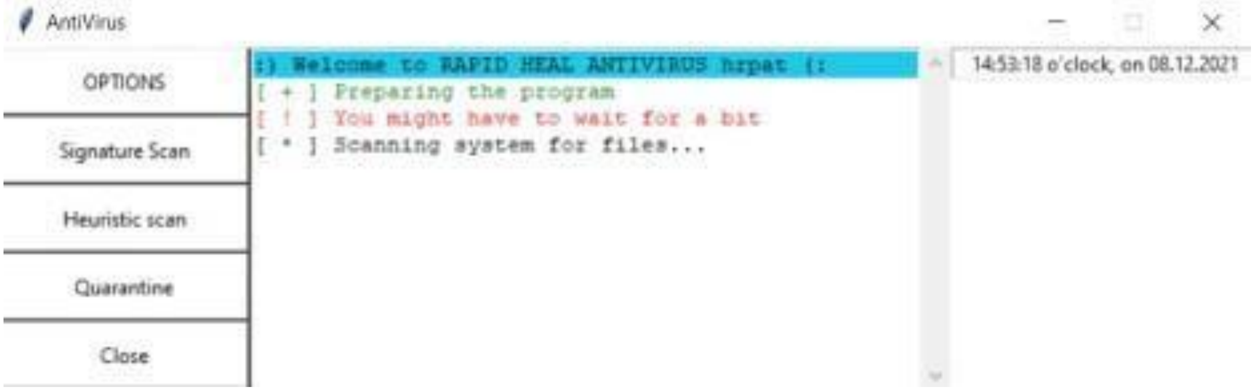
The second type of scan that we are doing is a heuristic scan. A heuristic scan is a type of scan which checks for changes in file size. When a virus code replicates itself, the file will get modified. Which also results in a change in the size of the file. The system already has stored the original file size data in a list. After running this scan, it will check for the current file size of all files. If the change is found, it will print the file is infected and also the original and current file data of filename, file size, and time stamp of modification.

Module 3: Quarantine

Quarantine is a new method used nowadays by antivirus software. It is basically a room for infected or malicious files. When a file is found dangerous to the system, that file will be moved to the quarantine folder. Here the antivirus software will research the problem and try to solve it and make a file safe again. Now there are two conditions, if the file is repaired, it can be restored from the quarantine list. And if it is not repaired, for the safety of our system we will have to remove it from our system.

5. RESULTS AND DISCUSSION

This is the output screen of our proposed Antivirus System:



```
:) Welcome to RAPID HEAL ANTIVIRUS hrpat (:
[ + ] Preparing the program
[ ! ] You might have to wait for a bit
[ * ] Scanning system for files...
```

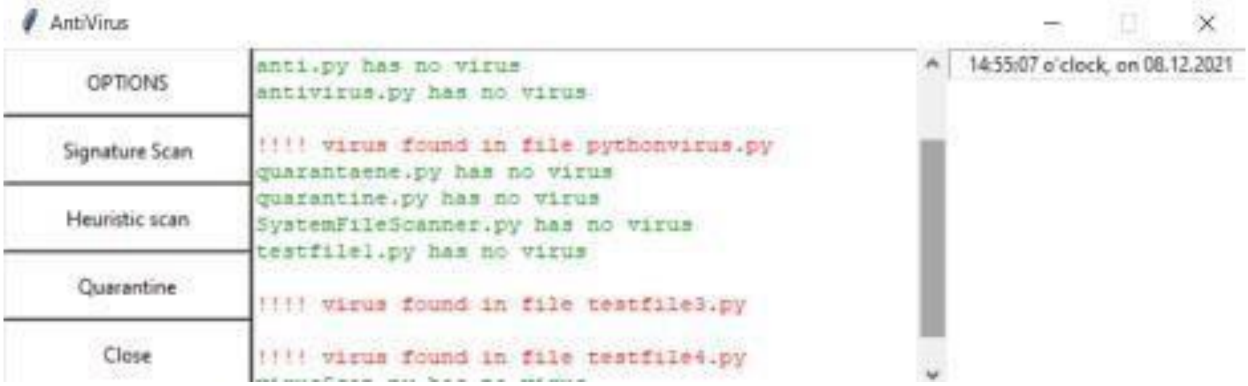
14:53:18 o'clock, on 08.12.2021

Fig 2: GUI interface

On left side we have options for different kind of operations that can be performed. We will see each one by one:

For signature scan operation we will select that option

If the file is safe, it will show no threat was found in green color or else show virus found threat in red color.



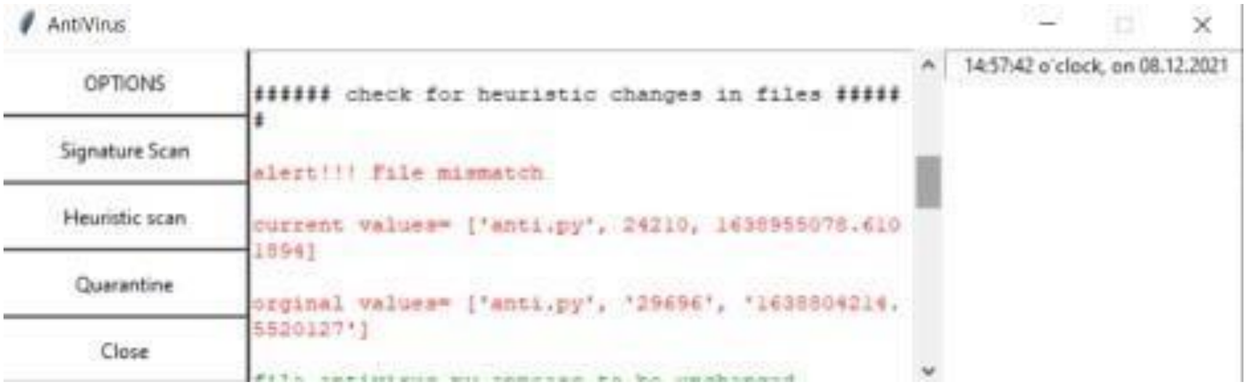
```
anti.py has no virus
antivirus.py has no virus

!!!! virus found in file pythonvirus.py
quarantaene.py has no virus
quarantine.py has no virus
SystemFileScanner.py has no virus
testfile1.py has no virus

!!!! virus found in file testfile3.py

!!!! virus found in file testfile4.py
```

14:55:07 o'clock, on 08.12.2021

Fig 3: Results of the signature scan

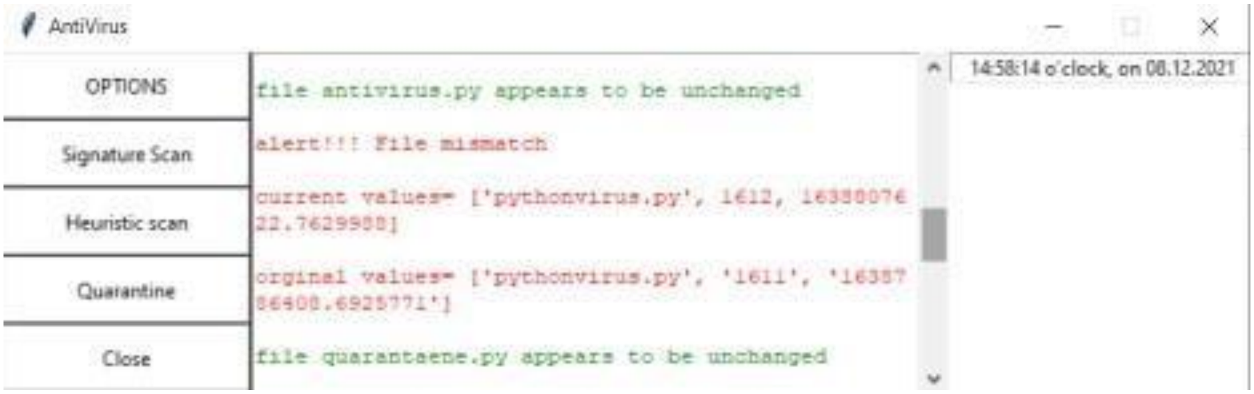
For heuristic scan operation we will select that option

If the file is safe, it will show that the file looks unchanged in green color or else show a virus found threat with the original size of file and current file size in red color.



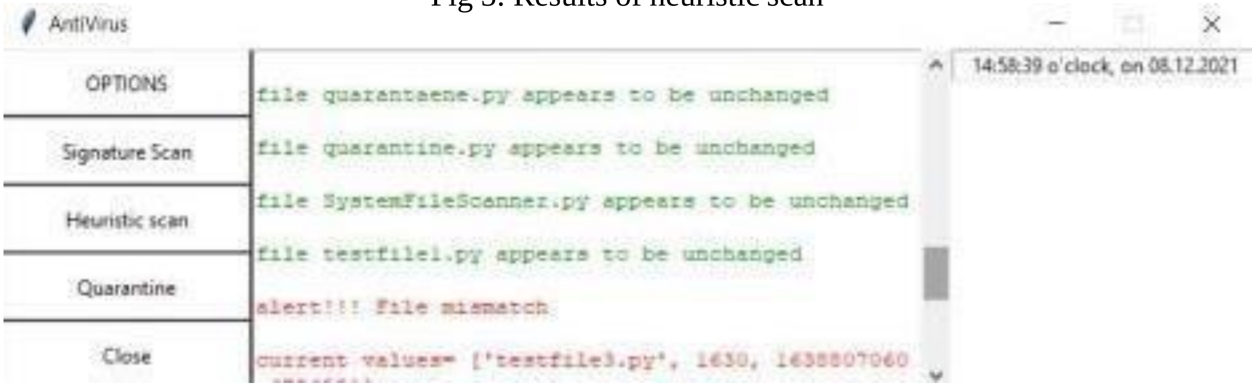
```
###### check for heuristic changes in files #####
#

alert!!! File mismatch

current values= ['anti.py', 24210, 1638955078.610
1894]

orginal values= ['anti.py', '29696', '1638804214.
5520127']
```

Fig 4: Results of heuristic scan



```
file antivirus.py appears to be unchanged

alert!!! File mismatch

current values= ['pythonvirus.py', 1612, 16388076
22.7629988]

orginal values= ['pythonvirus.py', '1611', '16387
86408.6925771']

file quarantaene.py appears to be unchanged
```

Fig 5: Results of heuristic scan

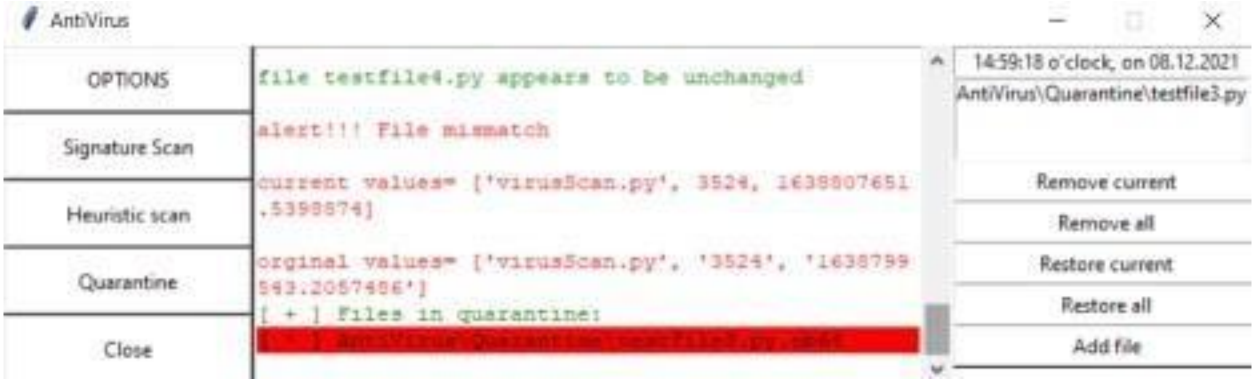


```
file quarantaene.py appears to be unchanged

file quarantine.py appears to be unchanged

file SystemFileScanner.py appears to be unchanged

file testfile1.py appears to be unchanged

alert!!! File mismatch

current values= ['testfile3.py', 1630, 1638807060
```

Fig 6: Results of heuristic scan

Now we will click on Quarantine.

It is used to quarantine the harmful files scanned by the scanner. We can select the file and put it in the quarantine list.

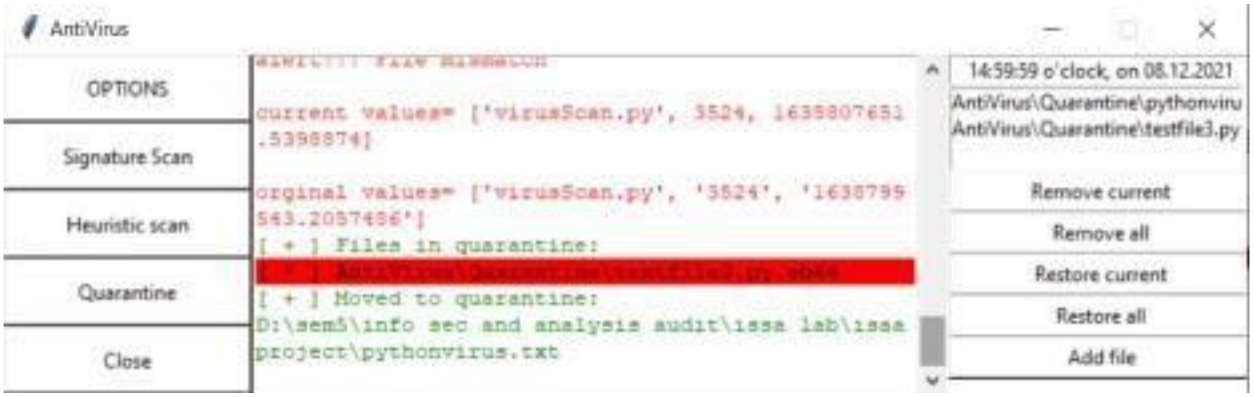


```
file testfile4.py appears to be unchanged

alert!!! File mismatch

current values= ['virusScan.py', 3524, 1638807651
.5398874]

orginal values= ['virusScan.py', '3524', '1638799
543.2057486']
[ + ] Files in quarantine:
[ + ] AntiVirus\Quarantine\testfile3.py.dd46
```

```
14:59:18 o'clock, on 08.12.2021
AntiVirus\Quarantine\testfile3.py
```

Fig 7: Adding file to quarantine

After adding a python virus file into quarantine:



```
alert!!! File mismatch

current values= ['virusScan.py', 3524, 1638807651
.5398874]

orginal values= ['virusScan.py', '3524', '1638799
543.2057486']
[ + ] Files in quarantine:
[ + ] AntiVirus\Quarantine\testfile3.py.dd46
[ + ] Moved to quarantine:
D:\sem5\info sec and analysis audit\issa lab\issa
project\pythonvirus.txt
```

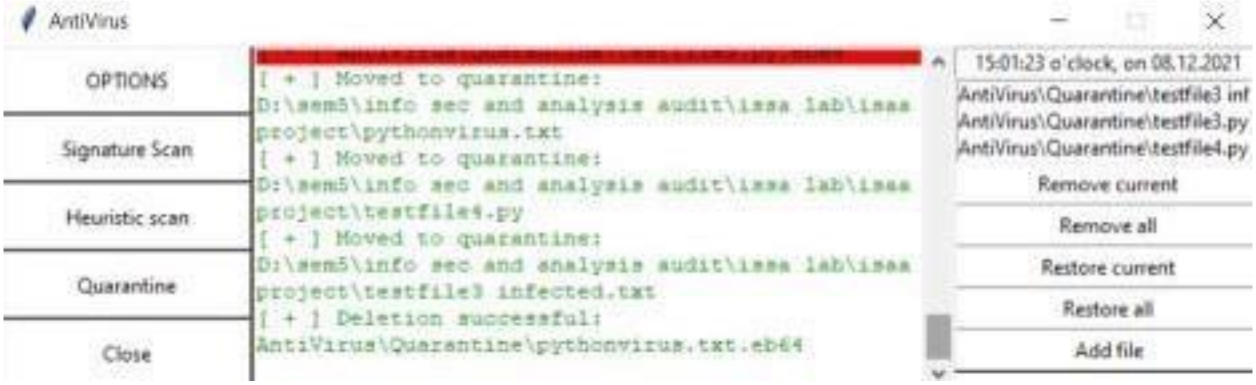
```
14:59:59 o'clock, on 08.12.2021
AntiVirus\Quarantine\pythonviru
AntiVirus\Quarantine\testfile3.py
```

Fig 8: Adding file to quarantine

In the quarantine folder we have options to add file, restore current file, restore all files, remove current file and remove all files.

Basically, when a file is repaired in quarantine, we can restore it back. And if it is not able to be repaired, we will remove it.

After removing current file from quarantine:

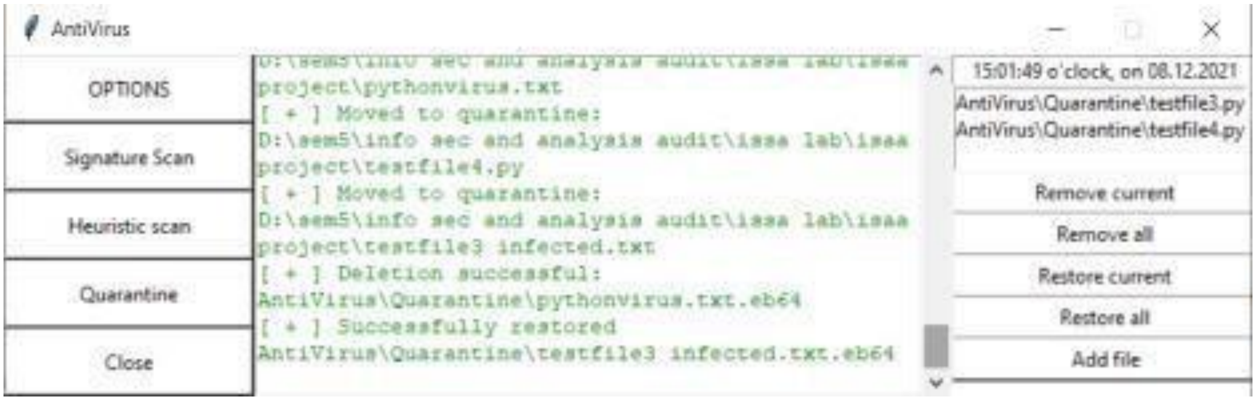


```
[ + ] Moved to quarantine:
D:\sem5\info sec and analysis audit\issa lab\issa
project\pythonvirus.txt
[ + ] Moved to quarantine:
D:\sem5\info sec and analysis audit\issa lab\issa
project\testfile4.py
[ + ] Moved to quarantine:
D:\sem5\info sec and analysis audit\issa lab\issa
project\testfile3 infected.txt
[ + ] Deletion successful:
AntiVirus\Quarantine\pythonvirus.txt.eb64
```

```
15:01:23 o'clock, on 08.12.2021
AntiVirus\Quarantine\testfile3 inf
AntiVirus\Quarantine\testfile3.py
AntiVirus\Quarantine\testfile4.py
```

Fig 9: Deleting a file from quarantine

After restoring current file from quarantine:

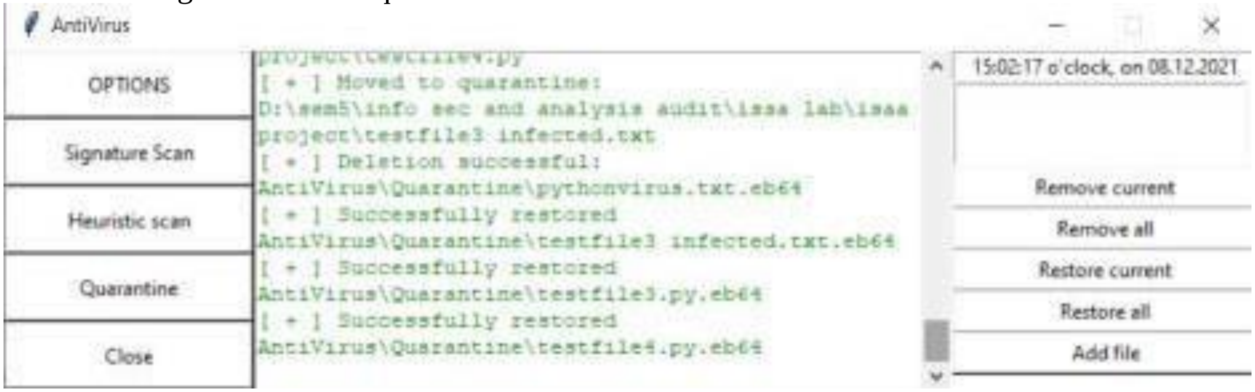


```
D:\sem5\info sec and analysis audit\issa lab\issa
project\pythonvirus.txt
[ + ] Moved to quarantine:
D:\sem5\info sec and analysis audit\issa lab\issa
project\testfile4.py
[ + ] Moved to quarantine:
D:\sem5\info sec and analysis audit\issa lab\issa
project\testfile3 infected.txt
[ + ] Deletion successful:
AntiVirus\Quarantine\pythonvirus.txt.eb64
[ + ] Successfully restored
AntiVirus\Quarantine\testfile3 infected.txt.eb64
```

```
15:01:49 o'clock, on 08.12.2021
AntiVirus\Quarantine\testfile3.py
AntiVirus\Quarantine\testfile4.py
```

Fig 10: Restoring current file from quarantine

After restoring all files from quarantine:

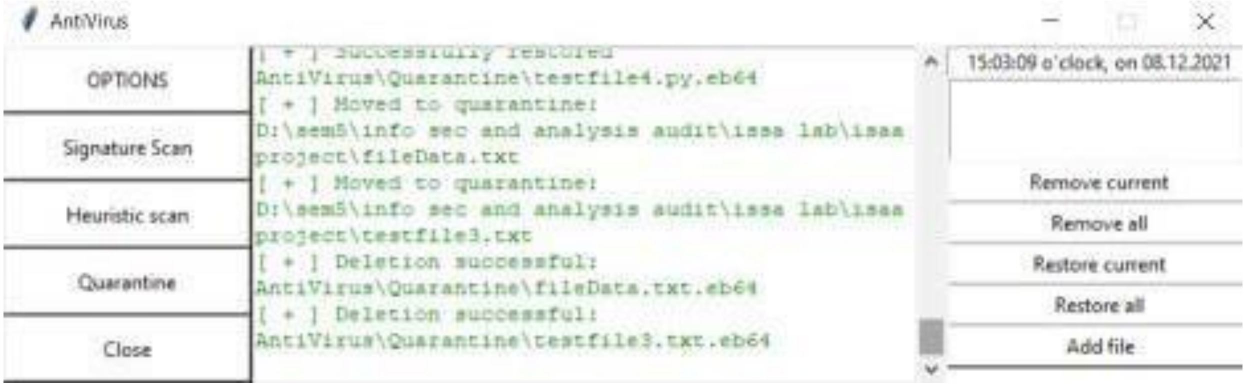


```
project\testfile4.py
[ + ] Moved to quarantine:
D:\sem5\info sec and analysis audit\issa lab\issa
project\testfile3 infected.txt
[ + ] Deletion successful:
AntiVirus\Quarantine\pythonvirus.txt.eb64
[ + ] Successfully restored
AntiVirus\Quarantine\testfile3 infected.txt.eb64
[ + ] Successfully restored
AntiVirus\Quarantine\testfile3.py.eb64
[ + ] Successfully restored
AntiVirus\Quarantine\testfile4.py.eb64
```

```
15:02:17 o'clock, on 08.12.2021
```

Fig 11: Restoring all files from quarantine

After removing all files from quarantine:



```
[ + ] Successfully restored
AntiVirus\Quarantine\testfile4.py.eb64
[ + ] Moved to quarantine:
D:\sem5\info sec and analysis audit\issa lab\issa
project\fileData.txt
[ + ] Moved to quarantine:
D:\sem5\info sec and analysis audit\issa lab\issa
project\testfile3.txt
[ + ] Deletion successful:
AntiVirus\Quarantine\fileData.txt.eb64
[ + ] Deletion successful:
AntiVirus\Quarantine\testfile3.txt.eb64
```

Fig 12: Removing all files from quarantine

So, this is how our GUI based virus detection and intrusion system works.

6. CONCLUSION AND FUTURE WORK

Both the virus and antivirus programs have been proved to be working efficiently. The self-replicating feature of the virus has been showcased. All the basic but necessary functionalities of the antivirus program have been implemented and can be used by anyone. With a simple GUI, users can just simply handle the program without any prior knowledge to how an antivirus works. With this project, we were able to learn more about the world of cybersecurity and how to tackle potential threats from malicious websites/hackers. As sure as there will be bad guys trying to steal/ exploit data there will always be a need for white hat hackers to step in and prevent their illegal causes. By researching survey papers, we were able to learn more about the creation as well as detection of viruses and how these are some of the most dangerous threats concerning the cyberworld in the day-to-day lives of many people around the world.

Furthermore, implementing Machine Learning and AI concepts on top of this project would definitely help increase its efficiency and overall performance.

7. REFERENCES

[1] Authors: Devi, K. Durga, and K. Mohan Kumar.
Title: "An Analysis of Various Anti-Virus Software Tools Based On Different Effective Parameters."
Publication: International Journal of Computer Science Trends and Technology (IJCST) 4.4 (2016): 104-110.
Volume Number: 4.4
ISSN Number: 2347-8578
Page Number:pp. 104-110.
Year: 2016

[2] Authors: Ray, Anusmita, and Asoke Nath.
Title: "Introduction to Malware and Malware Analysis: A brief overview."
Publication: International Journal 4.10 (2016).
Volume Number: 4.10
ISSN Number: 2321-7782
Page Number:pp. 22-30
Year: 2016

[3] Author: Gibert, Daniel, Carles Mateu, and Jordi Planes.
Title: "The rise of machine learning for detection and classification of malware: Research developments, trends and challenges."
Publication:Journal of Network and Computer Applications 153 (2020): 102526.
Volume Number:153
ISSN Number:102526
Year: 2020

[4] Author:Harshalatha, P., and R. Mohanasundaram.
Title:"Classification Of Malware Detection Using Machine Learn-ing Algorithms: A Survey."
Publication:International Journal of Scientific & Technology Research 9.02 (2020).
Volume Number:9.02
Year: 2020

[5] Author:Alsamiri, Jadel, and Khalid Alsubhi.
Title:"Internet of things cyber attacks detection using machine learning."
Publication:Int. J. Adv. Comput. Sci. Appl 10.12 (2019)
Volume Number: 10.12
ISSN Number:627-634.
Year 2019

[6]Author: Zhang, Xi, and Krishna Chaitanya Tadi.
Title:"Modeling virus and antivirus spreading over hybrid wireless ad hoc and wired networks."
Publication:IEEE GLOBECOM 2007-IEEE Global Telecommunications Conference. IEEE, 2007.
ISSN Number:1930-529
Page Number: pp.1-5

Year:2007

[7] Authors:Patil, Bhaskar V., and Rahul J. Jadhav.
Title:"Computer virus and antivirus software a brief review."
Publication: International Journal of Advances in Management and Economics 4.2 (2014): 1-4.
ISSN Number: 2278-3369
Page Number:pp.1-4
Year:2014

[8] Authors:Nachenberg, Carey S.
Title:"State-based cache for antivirus software."
Publication:U.S. Patent No. 5,854,916. 29 Dec. 1998.
Year:1998

[9]Authors: Kuwamura, Shin'ya.
Title:"Anti-virus method, computer, and recording medium."
Publication: U.S. Patent No. 8,176,558. 8 May 2012.
Year:2012

[10] Authors:Chamorro, Eugene, Jianchao Han, and Mohsen Beheshti.
Title:"The design and implementation of an antivirus software advising system."
Publication:2012 Ninth International Conference on Information Technology-New Generations. IEEE, 2012.
Year:2012
Page Number:pp. 612-617

[11] Authors:Kumari, Anku, and Ashok Kumar Mehta.
Title:"A hybrid intrusion detection system based on a decision tree and support vector machine."
Publication:2020 IEEE 5th International conference on computing communication and automation (ICCCA). IEEE, 2020.
Year:2020
ISSN Number:2641-8134
Page Number:pp. 396-400

[12] Authors:Reddy, G. Nikhita, and G. J. Reddy.
Title: "A study of cyber security challenges and its emerging trends on latest technologies." arXiv preprint arXiv:1402.1842 (2014).
Year:2014
Publication:International Journal of Engineering and Technology
Volume Number:4
ISSN Number:2049-3444

[13]Authors: Morales, Jose Andre, et al.
Title:"Testing and evaluating virus detectors for handheld devices."
Publication:Journal in Computer Virology2.2 (2006): 135-147.
Page Number:pp.135–147
Year:2006

[14] Authors:Tesauro, Gerald J., Jeffrey O. Kephart, and Gregory B. Sorkin.
Title:"Neural networks for computer virus recognition." IEEE expert11.4 (1996): 5-6.
Year:1996
Page Number:pp.5-6

[15] Authors:Shukla, J. B., et al.
Title: "Modeling and analysis of the effects of antivirus software on an infected computer network." Applied Mathematics and Computation 227 (2014): 11-18.
Page Number:pp.11-18
Year:2014

[16] Authors:Al Daoud, Essam, Iqbal H. Jebril, and Belal Zaqaibeh.
Title: "Computer virus strategies and detection methods."
Publication:Int. J. Open Problems Compt. Math 1.2 (2008):
Page Number:pp.12-20.
Year:2008

[17] Authors:Costea, Mihai, et al.
Title:"Aggregation of the knowledge base of antivirus software."
Publication: U.S. Patent No. 7,765,400. 27 Jul. 2010.
Year:2010

[18] Authors:Faruki, Parvez, et al.
Title:"Android security: a survey of issues, malware penetration, and defenses."
Publication:IEEE communications surveys & tutorials 17.2 (2014):
ISSN Number:998-1022.
Volume Number:17.2
Year:2014

[19] Authors:Karthick, S., and Sumitra Binu.
Title:"Android security issues and solutions." 2017
Publication: International Conference on Innovative Mechanisms for Industry Applications (ICIMIA). IEEE, 2017.

Year:2017

[20] Authors:Ahmed, Omar M., and Amira B. Sallow.
Title:"Android security: a review."
Publication:Academic Journal of Nawroz University 6.3 (2017):
ISSN Number:135-140.
Year:2017
Volume Number:6.3