PROJECT REPORT ON

Malware Detection And Analysis

GRAPHIC ERA

DEEMED TO BE UNIVERSITY

Submitted To : Department of Computer Science & Engineering

Partial fulfillment for the award of the degree of

Bachelor of Technology (CSE)

Semester - 3

Session: (2021-2022)

Submitted by

Name of the student: Mr. Y.V.VISHAL REDDY

Enrollment Number: GE-200217141

Roll no:2017141

Section : D

Under the Guidance of

Mr. RAMESH RAWAT

GraphicEra
UNIVERSITY
University under section 3 of UGC Act, 1956
Accredited by NAAC with Grade A

ACKNOWLEDGEMENT

I would like to take this opportunity to express my gratitude to entire faculty at Department of Computer Science and Information Technology, Graphic Era Deemed to be University; Dehradun who evaluated the project from time to time and gave me valuable suggestions as to how to improve the project.

I am grateful to **Mr. Ramesh Rawat,** Graphic Era Deemed to be University, Dehradun, for his supervision, encouragement, inspiration and guidance. Working under him is being an enriched experience.

In all, I found congenial work environment in Graphic Era University, Dehradun and this project completion will mark a new beginning for me in the coming days.

I am highly indebted to Graphic Era University for providing me the required infrastructure and facilities to accomplish the given task.

Y. V. Vishal Reddy

Bachelor of technology - CSE

Semester 3

Session (2021- 2022)

Graphic Era (Deemed To Be University)

Problem Statement

Malware detection and analysis using malware hashes (MD-5, SHA-1, SHA-256).

My project includes usage of a simple antivirus coded in python capable of scanning selected files and deleting files that it detects as infected. This antivirus uses a large list of MD5, SHA1 and SHA256 malware hashes to determine infections. However as this project progresses I would like to implement machine learning detection with the long term goal of becoming a fully functioning antivirus.

Motivation

The motivation for this project stems from my fascination with Deep Learning in Machine Learning and topic that involves the data analysis, hence this topic helped me to explore this field and work on it. At first I had to understand the concept of hashing and various algorithms associated with it, so as to proceed in my project involved.

Goals

The goal of this project is to scan various number of files and detect the infected files and later delete them accordingly. As we know that there are large number of increasing malware worldwide and we need special anti-virus/anti-malware to handle them. So, my antivirus uses large list of MD5, SHA1, SHA256 malware hashes to determine the infections**.**

Development Environment

- Visual Studio Code
- Pycharm

Language used

- Python (3.10 - 64bit)
- Libraries included are:

1. **Haslib** - This Library consists of many different functions which perform transforms a string to other string in different ways

2. **Tkinter** - Python provides the standard library Tkinter for creating the graphical user interface for desktop based applications.

3. **Json** - **JSON** is an open standard file format and data interchange format that uses human-readable text to store and transmit data objects consisting of attribute–value pairs and arrays .

4. **Functools** **-** Functools module is for higher-order functions that work on other functions. It provides functions for working with other functions and callable objects to use or extend them without completely rewriting them.

5. **Filedialog** - The Python filedialog module offers you a set of unique dialogs to be used when dealing with files. Tkinter has a wide variety of different dialogs, but the ones in filedialog are specifically designed for file selection.

Algorithms Used:

1) MD-5 Hash
2) SHA-1 Hash
3) SHA-256 Hash
….. Etc

Overview:

- **What Is MALWARE :**

Malware (a portmanteau for **malicious software**) is any software intentionally designed to cause disruption to a computer, server, client, or computer network, leak private information, gain unauthorized access to information or systems, deprive users access to information or which unknowingly interferes with the user's computer security and privacy. By contrast, software that causes harm due to some deficiency is typically described as a software bug. Malware poses serious problems to individuals and businesses. According to Symantec's 2018 Internet Security Threat Report (ISTR), malware variants number has increased to 669,947,865 in 2017, which is twice as many malware variants as in 2016.

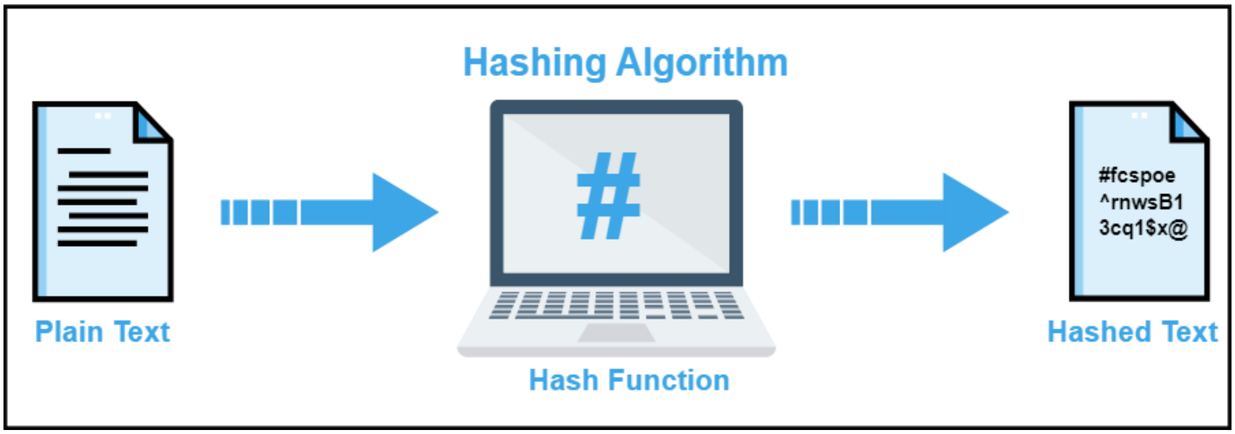
Many types of malware exist, including computer viruses, worms, Trojan horses, ransomware, spyware, adware, rogue software, wiper, and scareware. The defense strategies against malware differs according to the type of malware but most can be thwarted by installing antivirus software, firewalls, applying regular patches to reduce zero-day attacks, securing networks from intrusion, having regular backups and isolating infected systems. Malware is now being designed to evade antivirus software detection algorithms.

- **What is Hash Algorithm**:

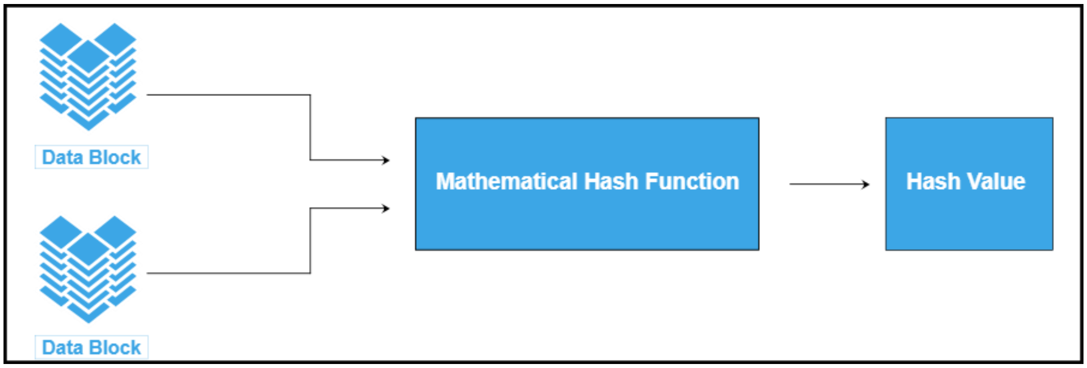
Hashing Algorithm – One of the Methods for Sending Essential Files

Hashing is one of the algorithms which calculates a string value from a file, which is of a fixed size. Basically, it contains blocks of data, which is transformed into a short fixed-length key or value from the original string. Usually, a summary of the information or data within that sent file.

Moreover, it's one of the convenient and secure ways when it comes to identifying or comparing databases and files. It's the process which transforms the dataset into a fixed-length character series, without considering the size of input data. And the received output is known as hash value, digest, code, or simply hash. Additionally, the term "hash" is used for describing both the hash function as well as the hash value.



Hash function is a typical mathematical function which is used for mapping data of different size into the fixed-sized values. It's a mathematical function that is used for converting an input value into a compressed format numerical value called a hash value or simply hashed.



<u>Some of the Popular Hashing Algorithms:</u>

1. MD (Message Digest) Algorithm

It's one of the widely used cryptographic hash function which relies upon the hash functions for generating a unique value, computed using the data and a unique symmetric key. Moreover, message digest algorithms are also known as encryption-only algorithms because of its capability to generate an exceptional value that can never be decrypted.
Ex- **MD5**

2. Whirlpool

In cryptography, the Whirlpool algorithm is one of the hash functions. It's based on an AES-like block cipher. Originally, Whirlpool was called Whirlpool-0, and later on, the revisions were made, and the first revision was known as Whirlpool-T and latest version as the Whirlpool.

3. Secure Hash Algorithm (SHA)

SHA published by the National Institute of Standards and Technology (NIST) is a family of cryptographic functions designed for keeping data secured. It’s intended to be one-way functions, which means once the data are changed into the hash values, it’s not possible to transform it back to the original data. Some of the SHA algorithms are SHA-1, SHA-2, SHA-3 and SHA-256 For example, the SHA algorithm is useful for the encryption of the passwords.

Summary

Hashing is quite useful in many ways. But these cryptographic hash functions are remarkably used in IT. Also, it’s used for message authentication codes (MACs), digital signatures, and other different types of authentication.

Moreover, it’s also useful for the identification of the files, indexing of data in hash tables, detecting duplicate data, or as checksums to detect there’s no accidental or intentional corruption of data in the sent file.

Lastly, for security purposes, it’s recommended to use hashing algorithms equipped with the newest technologies.

CODING

```
import hashlib
import os
from functools import partial
import json
from tkinter import *
from tkinter import filedialog

def scan_sha256(file):
    virus_found = False

    with open(file,"rb") as f:
        bytes = f.read()
        readable_hash = hashlib.sha256(bytes).hexdigest();

        print("The SHA256 hash of this file is: " + readable_hash)

        with open("SHA256.txt",'r') as f:
           lines = [line.rstrip() for line in f]
           for line in lines:
               if str(readable_hash) == str(line.split(";")[0]):
                   virus_found = True

           f.close()

    if not virus_found:
        print("File is safe!")
        label_status.configure(text="Status: File is safe!", width = 100, height = 4,
                   fg = "green")
    else:
        print("Virus detected! File quarentined")
        label_status.configure(text="Status: Virus detected! File Deleted!", width =
100, height = 4,
                   fg = "red")
        os.remove(file)
```

```
def scan_md5(file):
    virus_found = False

    with open(file,"rb") as f:
        bytes = f.read()
        readable_hash = hashlib.md5(bytes).hexdigest();

        print("The MD5 hash of this file is: " + readable_hash)

        with open("MD5 Virus Hashes.txt",'r') as f:
          lines = [line.rstrip() for line in f]
          for line in lines:
              if str(readable_hash) == str(line.split(";")[0]):
                  virus_found = True

          f.close()

    if not virus_found:
        print("File is safe!")
        label_status.configure(text="Status: File is safe!", width = 100, height = 4,
                  fg = "green")

        scan_sha256(file)
    else:
        print("Virus detected! File quarentined")
        label_status.configure(text="Status: Virus detected! File Deleted!", width =
100, height = 4,
                  fg = "red")
        os.remove(file)

def scan(file):
    virus_found = False

    with open(file,"rb") as f:
        bytes = f.read()
        readable_hash = hashlib.sha1(bytes).hexdigest();
```

```
            print("The SHA1 hash of this file is: " + readable_hash)

            with open('SHA1 HASHES.json', 'r') as f:
                dataset = json.loads(f.read())

                for index, item in enumerate(dataset["data"]):
                      if str(item['hash']) == str(readable_hash):
                          virus_found = True

                f.close()

      if not virus_found:
            print("File is safe!")
            label_status.configure(text="Status: File is safe!", width = 100, height = 4,
                            fg = "green")

            scan_md5(file)
      else:
            print("Virus detected! File quarentined")
            label_status.configure(text="Status: Virus detected! File Deleted!", width =
100, height = 4,
                            fg = "red")
            os.remove(file)

def browseFiles():
    filename = filedialog.askopenfilename(initialdir = "/",
                                          title = "Select a File",
                                          filetypes = (("Text files",
                                                        "*.*"),
                                                       ("all files",
                                                        "*.*")))

    opened_file.configure(text="File Opened: "+filename)
```

```
scan(filename)

window = Tk()

window.title('Antivirus')

window.geometry("500x500")

window.config(background = "white")

label_file_explorer = Label(window,
                           text = "Antivirus",
                           width = 100, height = 4,
                           fg = "blue"
                           ,bg = "white")

label_file_explorer.config(font=("Courier", 15))

label_status = Label(window,
                           text = "Status: ",
                           width = 100, height = 4,
                           fg = "blue",
                    bg = "white")

label_status.config(font=("Courier", 10))

opened_file = Label(window,
                           text = "File Opened: ",
                           width = 100, height = 4,
                           fg = "blue",
                   bg = "white")

opened_file.config(font=("Courier", 10))

button_explore = Button(window,
                       text = "Browse Files",
                       command = browseFiles)
```

```
label_file_explorer.grid(column = 1, row = 1)
label_file_explorer.place(x=-350, y=0)

opened_file.grid(column = 1, row = 1)
opened_file.place(x=-150, y=250)

label_status.grid(column = 1, row = 1)
label_status.place(x=-150, y=300)

button_explore.grid(column = 1, row = 2)
button_explore.place(x=205, y=400)

window.mainloop()
```

NOTE: The output of this application will be presented as a video presentation as it is entirely based on application.