VIRUSPARTIAL

Administrator Manual

By Team J

Contents

Introduction **2**

System Overview **3**

Background 3

Hardware and Software Requirements 3

Administrative Procedures **4**

Installation 4

Routine Tasks 5

Periodic Administration 5

User Support 5

Troubleshooting **6**

Dealing with Error Messages and Failures 6

Known Bugs and Limitations 6

Introduction

MD5 Hashes

MD5 message-digest algorithm is a widely used hash function that produces a 128-bit hash value. Made by Ronald Rivest from MIT, MD5 was designed to replace its predecessor, MD4, as it was flawed at the time. As with any hash function, any file can be encrypted by MD5, thus producing a hash unique to that file. This concept is also applicable to malware files as well.

Virus Partial

Virus Partial takes an MD5 hash, or a file, cross-validating it with current, renowned malicious hashes, and produces a prompt that informs the user if the input is a danger to the system or not. The verification of the file also lies with third-party software such as VirusTotal, AVG, Kingsoft, and many more.

System Overview

Background

The web application takes in a hash or file and sends a link of the input to a secondary virtual server through SSH. The server performs a request to get the input, thus storing it. The secondary server runs a Python script that analyzes the input as the DigitalOcean restores it from a backup. Thus, the safety of the main server is ensured as the process of reading the hash is contained in a sandbox server. Furthermore, the sandbox server's IP address is blacklisted to the main server, safeguarding the one-way communication between them.

Hardware and Software Requirements

There should be at least two servers required in order for the web application to run properly. One server running Python that is connected to the web, and another one running the Flask Framework, with the required dependencies already installed. Moreover, Port 22 should be reserved in the lesser server as it contains the sandbox server utilized by the main application. Additionally, it can be configured such that the sandbox could also operate through a VPN connection or a virtual machine.

Administrative Procedures

Installation

1. To set up the web application, simply clone the repository in the main server

```
Linux

git clone git@github.com:xMBlades/CMSC447-SemesterProject.git
```

2. Install sandbox.py in the secondary server

```
Python

import math
import hashlib
import os

if __name__ == "__main__":
    path = "hashme/"
    dir_list = os.listdir(path)
    f = open(path + dir_list[0], "rb")
    data = f.read()
    sha256 = hashlib.sha256()
    sha256.update(data)
    f.close()
    print(sha256.hexdigest())
    os.remove(path + dir_list[0])
```

3. Acquire the SSH information of the secondary server and code it to driver.py in the main server
4. Run the flask application

```
Linux

flask run
```

Routine Tasks

Any user can register on our website to fully experience the features embedded in the application. To do this:

1. Simply click on ‘Register’ in menu
2. Enter your desired username, email address, and password
3. Enter your profile picture (optional)
4. Enter your first and last name.

The session cookies, as well as the request queue, should be cleaned every couple of hours to mitigate any database errors and/or overloading.

Periodic Administration

It is important to regularly test the API calls within the main application to ensure the standard performance of the web application. This can be done with Postman.

For further information, the following link will lead to the documentation of Postman:

https://learning.postman.com/docs/getting-started/introduction/

The sandbox server should also be checked periodically to ensure that it has not failed to reboot.

User Support

For our user support, you can contact Team J through Discord by the tag Bridget.G#7902.

Troubleshooting

Dealing with Error Messages and Failures

For debugging, Flask's built-in debugger offers a full detailed report of an error. To utilize this, set the flask environment to "Development".

```
Linux

export FLASK_ENV=development
flask run
```

For more information on Flask's built-in debugger, the following link will lead to the documentation for it:

https://flask.palletsprojects.com/en/2.1.x/debugging/

Known Bugs and Limitations

Currently known bugs and limitations are:

- Using the Enter key to traverse across the menu on the home page
- Unable to drag and drop files