Application Security
CA 3 – Virus Total Tool

By Dean
B0009

Department of Informatics
School of Informatics and Engineering
XXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXX

Digital Forensics & Cyber Security
Application Security
Stephen

09/01/2020

Plagiarism Declaration

TECHNOLOGICAL UNIVERISTY XXXXXXXXXXXXXXXXXXXX
DEPARTMENT OF INFORMATICS
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
LECTURER: Stephen

DECLARATION ON PLAGIARISM

I declare that the work I/We am (are) submitting for assessment by the Institute examiner(s) is entirely my (our) own work, except where the author or source has been duly referenced and attributed.

I confirm that this material has not been previously submitted for a degree or any other qualification at XXXXX or any other institution. I further confirm that I have read and understood the Institute policy on plagiarism in assignments and examinations (3AS08.doc) and that I am not, so far as I am aware, in breach of any of these regulations.

Signed: Dean
Date: 09/01/2020

Table of Contents

Plagiarism Declaration 2

Given Brief 4

Coding Process 5

How to use the Application 7

- The Menu 7
- Menu Option 1 – Scan a File 7
- Menu Option 2 – Scan a URL 8
- Menu Option 3 – Hash Search 9
- Menu Option 4 – Scan IP Address 9
- Menu Option 5 – Report status 10
- Option Failure 10

Code Use Cases 12

Code 14

- The imports 14
- Global Variables 14
- Functions 14
  - Clear 14
  - sha256sum 15
  - Main Menu 15
  - Scan File 16
  - Scanurl 17
  - Hashsearch 17
  - Searchip 18
  - Reportstatus 18

The Code 19

Reference 22

Given Brief

The bellow images are of the given details from the brief, the brief contained 4 options which all where coding options, and one paper written report. The options available, a VirusTotal report parser, Windows User Activity Tracker, Vulnerability Scanner and a Tool of your choice.

Option 1

VirusTotal Report Parser

VirusTotal (VT) produce reports on malware samples uploaded to their repository. The reports are in JSON format and contain information such as the output from the AV engines, hash values, exif data, imports, timestamps etc.

The aim of this assignment is to build a tool that can perform several actions on files, URLs or hashes. The application must present the user with the following menu:

1. Scan file
2. Get file report
3. Upload URL
4. Get URL report
5. Report stats

- You must have a separate function for each choice
- Use the API code to help you https://developers.virustotal.com/reference

Suggested functions:

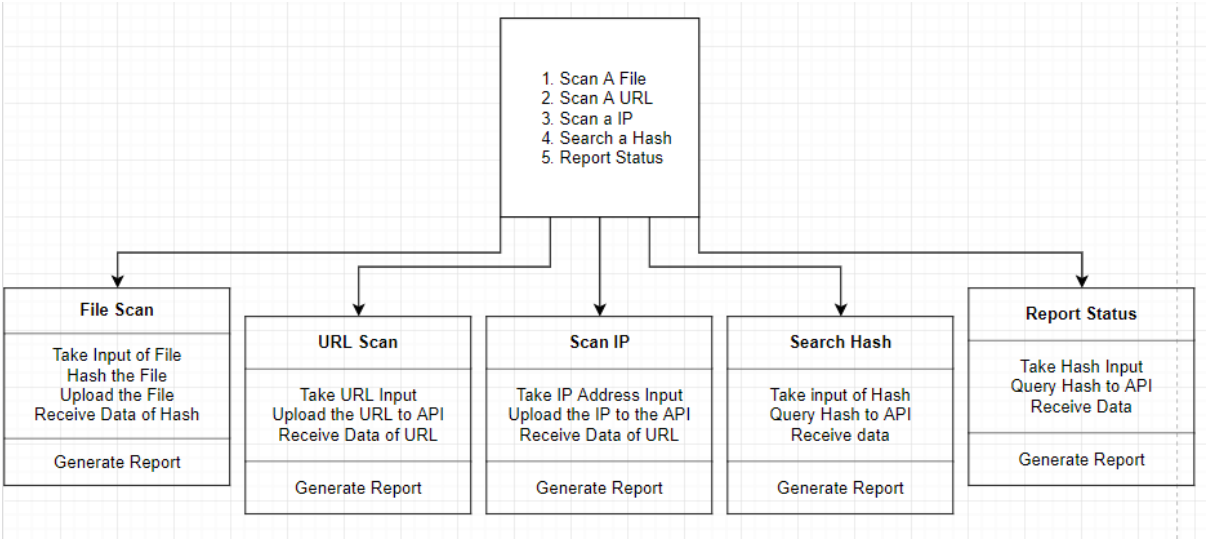
- **menu():** prints the menu to the user and collects their input.
- **scan_file(path to file)**: Uploads a file to VT for scanning - the response should be given back to the user
- **get_file_report(hash of file):** Gets the file report based on the hash. Write the file report to a JSON file
- **upload_url(url):** upload a URL for scanning - the response should be given back to the user
- **get_url_report(url):** Gets the URL report based on the hash. Write the file report to a JSON file
- **report_stats(path to JSON file):** print each malware name, how many times it was predicted and also the highest single family prediction.

Based on my knowledge of coding, I chose option one based on the fact it was interacting with pre-built API and could be built with the only language in which I have an understanding, python. I had an interest in the windows user activity tracker as it was leaning towards what my thesis idea but I found the issues would be I would need to have an understanding of C or windows native programming languages. The idea of this project is to develop a tool that integrates into with the API of Virus Total to allow the user to be able to complete multiple functions.

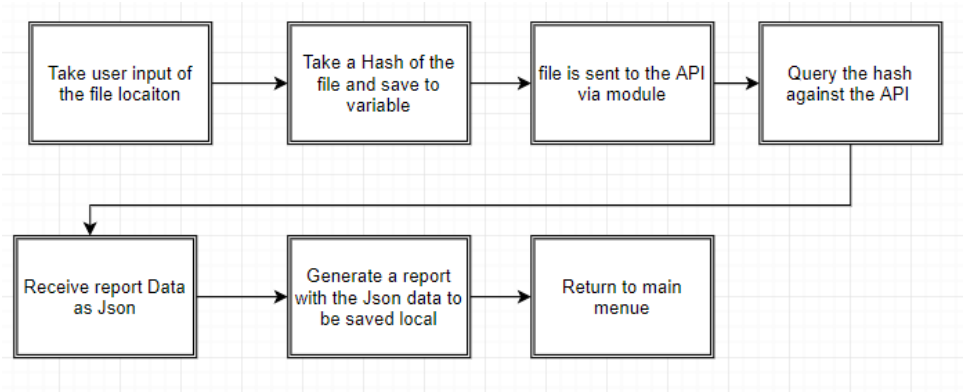
Coding Process

In this section you will see the idea flowcharts of the application such as how the program will work in stages of use such as receiving data and forwarding data to the appropriate place.

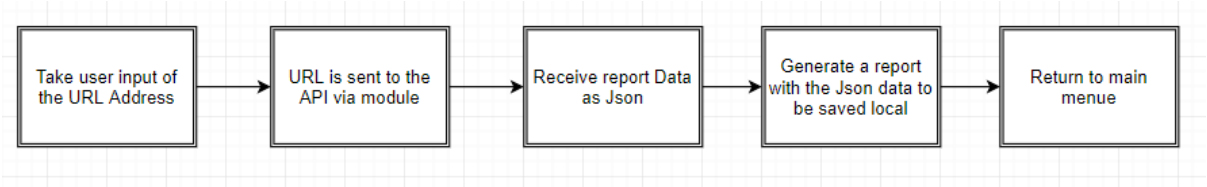
Looking at how to build the application based of the menu options you can see the image bellow would be the offshoot of the menu to each function



The File Scan Function



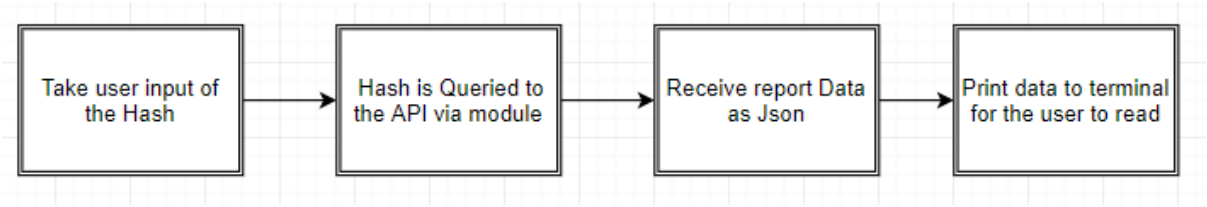
The URL Scan Function



The Scan IP Function

The Search hash Function

The Report Status Function



How to use the Application

In this section we will look at how to use the application.

The Menu

Upon running the python script you will be presented with the following menu to let the user chose what options they want.

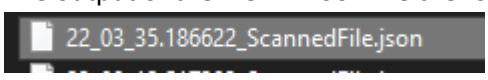
```
Code Currently Only Works on Windows with python 3.7
Wrote By Dean O'Neil B00091839 for Application Security 2019/2020

The Current Time is:
23:09:12.517303
Each Section Will Generate its own Report

        1.Scan a File
        2.Scan a URL
        3.Search a Hash
        4.Scan an IP
        5.Report Status

What would you like to do? use numbers to select:
```

Menu Option 1 – Scan a File

To scan a file the user will be prompted with the bellow prompt and asked to input a file to scan, the format can be see below. The application will upload the file to VirusTotal via the API, this will also generate a sha256 to be queried against VirusTotal for the report. This tool will automatically generate a report in .json with the current time and details of the scan type to the local folder. After the output is complete the user will be redirected back to the main menu.

```
What would you like to do? use numbers to select: 1

This Will scan files for possible viruses

Please set as Filename.blah this can be done by dragging the file into the console
Please give file location: C:\Users\Dean\PycharmProjects\VirusTotal\message.txt

The file is set to: C:\Users\Dean\PycharmProjects\VirusTotal\message.txt
Files Sha256 hash: 6445633f4eb06b531c13ea2b2c5e4acbd1c0036268c07e8c794d16afe3e57140
Scanning the File Now, this may take a few second

Generating Report Now
This Will Creat a HTML File with the report data
Returning to Main Menu after File Created
```

The output of the file will look like the following



And will contain the results data in the following output, to make it easier for the user to read


```
        },
        "Panda": {
            "detected": false,
            "version": "4.6.4.2",
            "result": null,
            "update": "20200109"
        },
        "Qihoo-360": {
            "detected": false,
            "version": "1.0.0.1120",
            "result": null,
            "update": "20200109"
        }
    },
    "scan_id": "6445633f4eb06b531c13ea2b2c5e4acbd1c0036268c07e8c794d16afe3e57140-1578607088",
    "sha1": "c0256e016c2a7e6f90f111f2d4aef019e95b8b6d",
    "resource": "6445633f4eb06b531c13ea2b2c5e4acbd1c0036268c07e8c794d16afe3e57140",
    "response_code": 1,
    "scan_date": "2020-01-09 21:58:08",
    "permalink": "https://www.virustotal.com/file/6445633f4eb06b531c13ea2b2c5e4acbd1c0036268c07e8c794d16afe3e57140/analysis/1578607088/",
    "verbose_msg": "Scan finished, information embedded",
    "total": 60,
    "positives": 0,
    "sha256": "6445633f4eb06b531c13ea2b2c5e4acbd1c0036268c07e8c794d16afe3e57140",
    "md5": "2bad578ccae8613d5e852d6b96138dab"
}
```


Menu Option 2 – Scan a URL

This option allows a user to scan a URL in the following format that can be seen bellow; this can be as simple as google.com as a URL to scan, this will send the URL through the API to VirusTotal. The tool will automatically generate a report in .json with the current time and details of the scan type to the local folder. After the output is complete the user will be redirected back to the main menu.

```
What would you like to do? use numbers to select: 2

This will scan a URL for issues
in format sample.com
What URL do you want to scan: google.com
Scanning the File Now, this may take a second

Generating Report Now
This Will Creat a HTML File with the report data
Returning to Main Menu after File Created
```

The output of the file will look like the following

And will contain the results data in the following output, to make it easier for the user to read


```
{
    "status_code": 200,
    "json_resp": {
        "https_certificate_date": 1578421821,
        "detected_downloaded_samples": [
            {
                "date": "2019-10-11 15:53:58",
                "positives": 2,
                "total": 69,
                "sha256": "45bf8866af1dfdbfc4177d3807df43221e87bd3037d6681b4bd3d8b402376f9e"
            },
            {
                "date": "2019-10-08 16:30:36",
                "positives": 2,
                "total": 70,
                "sha256": "ec7ee9eb45069d5716658a384b9c6738543ae20c60004dfbf7c6abcffa38a134"
            },
            {
                "date": "2019-03-04 21:06:36",
                "positives": 1,
                "total": 71,
                "sha256": "10a01be40c332fffbbad2df799c98d6a45af24cd78a10c264f37c51b7fc50869"
            },
            {
                "date": "2019-03-01 00:08:36",
                "positives": 1,
                "total": 71,
                "sha256": "61bb6b68bc02d5d445bae6d1a19214fc9899acf8b0a327e0c38288a48ed3cc47"
            },
```

Menu Option 3 – Hash Search

This option lets a user search a hash against the virus total API, this hash will be taken via the command line after a user is prompted. The data will then be sent to the API and the response will be sent back to the system. The tool will automatically generate a report in .json with the current time and details of the scan type to the local folder. After the output is complete the user will be redirected back to the main menu.

```
What would you like to do? use numbers to select: 3

This will Allow you to search a Hash
Please insert the MD5 / SHA1 / SHA256 Hash: 0d8c2bcb575378f6a88d17b5f6ce70e794a264cdc8556c8e812f0b5f9c709198

Generating Report Now
This Will Creat a HTML File with the report data
Returning to Main Menu after File Created
```

The output of the file will look like the following

And will contain the results data in the following output, to make it easier for the user to read

Menu Option 4 – Scan IP Address

This option will allow a user to submit and scan an IP Address via the command line when a user is prompted, the application will capture the response. The tool will automatically generate a report in .json with the current time and details of the scan type to the local folder. After the output is complete the user will be redirected back to the main menu.

```
What would you like to do? use numbers to select: 4

This will Allow you to search a IP Address
Please Input IP to Search: 8.8.8.8
Scanning the File Now, this may take a second

Generating Report Now
This Will Creat a HTML File with the report data
Returning to Main Menu after File Created
```

The output of the file will look like the following

And will contain the results data in the following output, to make it easier for the user to read


```
"status_code": 200,
"json_resp": {
    "undetected_downloaded_samples": [
        {
            "date": "2017-07-20 16:26:59",
            "positives": 0,
            "total": 68,
            "sha256": "2d4b2832357ae115fa2558f3f349c96e190110c810ef40fd29c008d59fe04bd0"
        },
        {
            "date": "2017-07-20 16:12:24",
            "positives": 0,
            "total": 68,
            "sha256": "fc56c63a514546f8ce213425237516ad7db7e77a360765832162da7caeb23fb8"
        },
        {
            "date": "2018-05-14 20:58:56",
            "positives": 0,
            "total": 69,
            "sha256": "b52a9b9daab35cc52f960125ce0e8170b2064b4b154feffd8cc87695c52e83c7"
```


Menu Option 5 – Report status


```
What would you like to do? use numbers to select: 5

This section will show current report status
Hash Format Accepted Sha256
Please Enter your Has to Search: 0d8c2bcb575378f6a88d17b5f6ce70e794a264cdc8556c8e812f0b5f9c709198

{
    "status_code": 200,
    "json_resp": {
        "scans": {
            "Bkav": {
                "detected": true,
                "version": "1.3.0.9899",
                "result": "W32.GenTrojan.Trojan",
```


All output is viewable in the console


```
            "update": "20200103"
        }
    },
    "scan_id": "0d8c2bcb575378f6a88d17b5f6ce70e794a264cdc8556c8e812f0b5f9c709198-1578049168",
    "sha1": "758240613c362bb1fd13e07d3d19f357b7f8a6da",
    "resource": "0d8c2bcb575378f6a88d17b5f6ce70e794a264cdc8556c8e812f0b5f9c709198",
    "response_code": 1,
    "scan_date": "2020-01-03 10:59:28",
    "permalink": "https://www.virustotal.com/file/0d8c2bcb575378f6a88d17b5f6ce70e794a264cdc8556c8e812f0b5f9c709198/analysis/1578049168/",
    "verbose_msg": "Scan finished, information embedded",
    "total": 72,
    "positives": 57,
    "sha256": "0d8c2bcb575378f6a88d17b5f6ce70e794a264cdc8556c8e812f0b5f9c709198",
    "md5": "cc1db5360109de3b857654297d262ca1"
}
```


Option Failure

If the wrong option is entered the program will say that it is not a Valid option and return the user to the main menu, to be able to reselect an option.

```
What would you like to do? use numbers to select: 943

 Not Valid Choice Try again
Code Currently Only Works on Windows with python 3.7
Wrote By Dean O'Neil B00091839 for Application Security 2019/2020

The Current Time is:
01:10:30.316679
Each Section Will Generate its own Report

        1.Scan a File
        2.Scan a URL
        3.Search a Hash
        4.Scan an IP
        5.Report Status

What would you like to do? use numbers to select:
```

Code Use Cases

The code can be used in different based on the information needed; the menu system displays what is allowed by the program but. The use cases including speeding up the searching process via the command line, this also allowing people who do not have a GUI to also be able to use the tool, There are other benefits that the output of the file can be used and in other utilities that can accept the .Json data

The user can use functions like the following, such as getting virus total to scan a file and generate a report.

```
What would you like to do? use numbers to select: 1

This Will scan files for possible viruses

Please set as Filename.blah this can be done by dragging the file into the console
Please give file location: C:\Users\Dean\PycharmProjects\VirusTotal\message.txt

The file is set to: C:\Users\Dean\PycharmProjects\VirusTotal\message.txt
Files Sha256 hash: 6445633f4eb06b531c13ea2b2c5e4acbd1c0036268c07e8c794d16afe3e57140
Scanning the File Now, this may take a few second

Generating Report Now
This Will Creat a HTML File with the report data
Returning to Main Menu after File Created
```

It allows a user to scan a domain such as google and generate a report for a user to read over

```
What would you like to do? use numbers to select: 2

This will scan a URL for issues
in format sample.com
What URL do you want to scan: google.com
Scanning the File Now, this may take a second

Generating Report Now
This Will Creat a HTML File with the report data
Returning to Main Menu after File Created
```

It allows a user to check for a hash on VirusTotal and pull down a report

```
What would you like to do? use numbers to select: 3

This will Allow you to search a Hash
Please insert the MD5 / SHA1 / SHA256 Hash: 0d8c2bcb575378f6a88d17b5f6ce70e794a264cdc8556c8e812f0b5f9c709198

Generating Report Now
This Will Creat a HTML File with the report data
Returning to Main Menu after File Created
```

It can allow a user to scan an IP address and generate a report based on that data

```
What would you like to do? use numbers to select: 4

This will Allow you to search a IP Address
Please Input IP to Search: 8.8.8.8
Scanning the File Now, this may take a second

Generating Report Now
This Will Creat a HTML File with the report data
Returning to Main Menu after File Created
```

The report status function can allow the process to be faster by dumping the information from a queried has into the terminal for a user to read.

```
What would you like to do? use numbers to select: 5

This section will show current report status
Hash Format Accepted Sha256
Please Enter your Has to Search: 0d8c2bcb575378f6a88d17b5f6ce70e794a264cdc8556c8e812f0b5f9c709198

{
    "status_code": 200,
    "json_resp": {
        "scans": {
            "Bkav": {
                "detected": true,
                "version": "1.3.0.9899",
                "result": "W32.GenTrojan.Trojan",
```

Code

The application was completely wrote in python as it is the only language I know at current, there were code segments taken from past projects and notes from the official Virus Total API guide, these will be referenced in the reference section.

The imports

```
from virustotal_python import Virustotal
import datetime
import time
import hashlib
from os import system, name
import json
```

Virus Totals Python Module for interacting with the API
Datetime is imported for choosing and working with current date and time
Time is imported to assist with the Datetime function
System and Name from the OS module are imported for working with the console
json module was imported for working with the Reponses from the VirusTotal API
hashlib for generating the the sha256 of the file.

Global Variables

```
vtotal = Virustotal("1889b1ae9287d8d3f55ab4e4ab0baa501849cfa09d55dd30021a317c35a64f6d")
utc = datetime.datetime.now().time()
main()
```

I decided to use global variables for the application as it seemed to work better, I had original wrote the option in for the user to add their own API key after a prompt when the user started the program first, but due to speed up the process I decided to hardcode my own API key, just to test the system.

Functions

Here is a list of all the functions in the application

Clear

```
def clear():
    if name == 'nt':
        _ = system('cls')
    else:
        _ = system('clear')
```

The purpose of this function is to clear the console so that the console is not filled with cluttered data.

sha256sum

```
# This code section was built with help from https://stackoverflow.com/questions/22058048/hashing-a-file-in-python
def sha256sum(filename):
    h = hashlib.sha256()
    b = bytearray(128 * 1024)
    mv = memoryview(b)
    with open(filename, 'rb', buffering=0) as f:
        for n in iter(lambda: f.readinto(mv), 0):
            h.update(mv[:n])
    return h.hexdigest()
```

This code section is for the generation of a Sha256 hash of the file for the file upload function, this code section was built with help from users on Stackoverflow, the link to the thread is available in the references of this document.

Main Menu

```
def main():
    print("Code Currently Only Works on Windows with python 3.7")
    print("Wrote By Dean O'Neil B00091839 for Application Security 2019/2020")
    print("")
    print("The Current Time is:")
    print(utc)
    print("Each Section Will Generate its own Report")
    print("""
        1.Scan a File
        2.Scan a URL
        3.Search a Hash
        4.Scan an IP
        5.Report Status
        """)
    answer = input("What would you like to do? use numbers to select: ")
    if answer == "1":
        scanfile()
    elif answer == "2":
        scanurl()
    elif answer == "3":
        hashsearch()
    elif answer == "4":
        searchip()
    elif answer == "5":
        reportstatus()
    elif answer != "":
        print("\n Not Valid Choice Try again")
        main()
```

The main function is the function for the menu system, this allows after processing of data that the program and revert the user back to the main menu. The Menu options are as follows

1- Scan a file (This will scan a given file against the API, and generate a report)
2- Scan a URL (This will scan a URL against the API and generate a report)
3- Search a Hash (This will search a hash file against the API and generate a report)
4- Scan an IP Address (This will scan an IP address and generate a report)
5- Report Status (A current it will take a hash and display details of that hash as it is unknown to me what the report status function is required to have

There is a redundancy set in place that if the user choses the wrong option it will alert the person and reload the Main Menu.

Scan File

```
def scanfile():
    print("")
    # this will take a file check its hash, if its hash isnt found it will upload the sample
    print("This Will scan files for possible viruses")
    print("")
    print("Please set as Filename.blah this can be done by dragging the file into the console")
    filename = input("Please give file location: ")
    print("\nThe file is set to: " + filename)
    responsefromvt = vtotal.file_scan(filename)
    md5h = sha256sum(filename)
    print("Files Sha256 hash: " + md5h)
    time.sleep(20)
    print("Scanning the File Now, this may take a few second")
    responsefromvt = vtotal.file_report([md5h])
    print("")
    print("Generating Report Now")
    print("This Will Creat a HTML File with the report data")
    utcl = str(utc).replace(':', '_')
    f = open(str(utcl) + "_" + 'ScannedFile' + ".json", "w+")
    f.write(str(json.dumps(responsefromvt, sort_keys=False, indent=4)))
    f.close()
    print("Returning to Main Menu after File Created")
    print("")
    time.sleep(5)
    clear()
    main()
```

This section of the code is the function in which a file can be scanned by VirusTotal, to do this it takes the file name from the local folder as input then and supplies that to the vtotal.file_scan api function, there is a wait to assume that the file is uploaded and scanned, then the app will request the report from VirusTotal.

The response from virus total is then forwarded to a .json file created with the current time and saved to the local folder; this file contains all the data of the application output.

Scanurl

```
def scanurl():
    print("")
    print("This will scan a URL for issues")
    print("in format sample.com")
    urltoscan = input("What URL do you want to scan: ")
    responsefromvt = vtotal.url_scan(urltoscan)
    time.sleep(10)
    print("Scanning the File Now, this may take a second")
    responsefromvt = vtotal.url_report(urltoscan)
    responsefromvt = vtotal.domain_report(urltoscan)
    print("")
    print("Generating Report Now")
    print("This Will Creat a HTML File with the report data")
    utcl = str(utc).replace(':', '_')
    f = open(str(utcl) + "_" + urltoscan + ".json", "w+")
    f.write(str(json.dumps(responsefromvt, sort_keys=False, indent=4)))
    f.close()
    print("Returning to Main Menu after File Created")
    print("")
    time.sleep(5)
    clear()
    main()
```

This function will take in a user's input of a url in the simplest form of "google.com" it will then query this domain against virustotal and wait for it to be scanned. The application will then prepare to write the report. The response from virus total is then forwarded to a .json file created with the current time and saved to the local folder; this file contains all the data of the application output.

Hashsearch

```
def hashsearch():
    print("")
    print("This will Allow you to search a Hash")
    hashtosearch = input("Please insert the MD5 / SHA1 / SHA256 Hash: ")
    responsefromvt = vtotal.file_rescan
    time.sleep(10)
    responsefromvt = vtotal.file_report([hashtosearch])
    print("")
    print("Generating Report Now")
    print("This Will Creat a HTML File with the report data")
    utcl = str(utc).replace(':', '_')
    f = open(str(utcl) + "_" + 'HashSearch' + ".json", "w+")
    f.write(str(json.dumps(responsefromvt, sort_keys=False, indent=4)))
    f.close()
    print("Returning to Main Menu after File Created")
    print("")
    time.sleep(5)
    clear()
    main()
```

This function allows the user to supply a hash to be searched against Virus Total for past items that have been scanned or generate a report based on the information that virus total currently has of the data. The response from virus total is then forwarded to a .json file created with the current time and saved to the local folder; this file contains all the data of the application output.

Searchip

```
def searchip():
    print("")
    print("This will Allow you to search a IP Address")
    iptosearch = input("Please Input IP to Search: ")
    responsefromvt = vtotal.ipaddress_report(iptosearch)
    # time.sleep(10)
    print("Scanning the File Now, this may take a second")
    print("")
    print("Generating Report Now")
    print("This Will Creat a HTML File with the report data")
    # For troubleShooting json output
    # print(json.dumps(responsefromvt, sort_keys=False, indent=4))
    utcl = str(utc).replace(':', '_')
    f = open(str(utcl) + "_" + iptosearch + ".json", "w+")
    f.write(str(json.dumps(responsefromvt, sort_keys=False, indent=4)))
    f.close()
    print("Returning to Main Menu after File Created")
    print("")
    time.sleep(5)
    clear()
    main()
```

This function allows a user to query an IP address against VirusTotal, the application will then pull then wait for a response from virustotal, the response from virus total is then forwarded to a .json file created with the current time and saved to the local folder; this file contains all the data of the application output.

Reportstatus

```
def reportstatus():
    print("")
    print("This section will show current report status")
    print("Hash Format Accepted Sha256")
    status = input("Please Enter your Has to Search: ")
    print("")
    responsefromvt = vtotal.file_report([status])
    print(json.dumps(responsefromvt, sort_keys=False, indent=4))
    time.sleep(20)
    main()
```

As I did not have a full understanding of what the report status function was to entail from the brief, I decided to make it in a similar function that that of the querying the hash, the only difference is the report status will make the output appear in the user's console.

The Code

Below is the code from the application itself, this can easily be copy and pasted and should work on most OS's. The requirements are listed below you will need the following modules that can be installed easily in pycharm.

Virustotal_Python
Datetime
Time
Hashlib
Json

```
from virustotal_python import Virustotal
import datetime
import time
import hashlib
from os import system, name
import json

def clear():
    if name == 'nt':
        _ = system('cls')
    else:
        _ = system('clear')

def main():
    print("Code Currently Only Works on Windows with python 3.7")
    print("Wrote By Dean O'Neil B00091839 for Application Security 2019/2020")
    print("")
    print("The Current Time is:")
    print(utc)
    print("Each Section Will Generate its own Report")
    print("""
        1.Scan a File
        2.Scan a URL
        3.Search a Hash
        4.Scan an IP
        5.Report Status
        """)
    answer = input("What would you like to do? use numbers to select: ")
    if answer == "1":
        scanfile()
    elif answer == "2":
        scanurl()
    elif answer == "3":
        hashsearch()
    elif answer == "4":
        searchip()
    elif answer == "5":
        reportstatus()
    elif answer != "":
        print("\n Not Valid Choice Try again")
        main()

def scanfile():
    print("")
    # this will take a file check its hash, if its hash isnt found it will upload
the sample
    print("This Will scan files for possible viruses")
    print("")
    print("Please set as Filename.blah this can be done by dragging the file into
the console")
    filename = input("Please give file location: ")
```

```
    print("\nThe file is set to: " + filename)
    responsefromvt = vtotal.file_scan(filename)
    md5h = sha256sum(filename)
    print("Files Sha256 hash: " + md5h)
    time.sleep(20)
    print("Scanning the File Now, this may take a few second")
    responsefromvt = vtotal.file_report([md5h])
    print("")
    print("Generating Report Now")
    print("This Will Creat a HTML File with the report data")
    utc1 = str(utc).replace(':', '_')
    f = open(str(utc1) + "_" + 'ScannedFile' + ".json", "w+")
    f.write(str(json.dumps(responsefromvt, sort_keys=False, indent=4)))
    f.close()
    print("Returning to Main Menu after File Created")
    print("")
    time.sleep(5)
    clear()
    main()

# This code section was built with help from
https://stackoverflow.com/questions/22058048/hashing-a-file-in-python
def sha256sum(filename):
    h = hashlib.sha256()
    b = bytearray(128 * 1024)
    mv = memoryview(b)
    with open(filename, 'rb', buffering=0) as f:
        for n in iter(lambda: f.readinto(mv), 0):
            h.update(mv[:n])
    return h.hexdigest()

def scanurl():
    print("")
    print("This will scan a URL for issues")
    print("in format sample.com")
    urltoscan = input("What URL do you want to scan: ")
    responsefromvt = vtotal.url_scan(urltoscan)
    time.sleep(10)
    print("Scanning the File Now, this may take a second")
    responsefromvt = vtotal.url_report(urltoscan)
    responsefromvt = vtotal.domain_report(urltoscan)
    print("")
    print("Generating Report Now")
    print("This Will Creat a HTML File with the report data")
    utc1 = str(utc).replace(':', '_')
    f = open(str(utc1) + "_" + urltoscan + ".json", "w+")
    f.write(str(json.dumps(responsefromvt, sort_keys=False, indent=4)))
    f.close()
    print("Returning to Main Menu after File Created")
    print("")
    time.sleep(5)
    clear()
    main()

def hashsearch():
    print("")
    print("This will Allow you to search a Hash")
    hashtosearch = input("Please insert the MD5 / SHA1 / SHA256 Hash: ")
    responsefromvt = vtotal.file_rescan
    time.sleep(10)
    responsefromvt = vtotal.file_report([hashtosearch])
    print("")
    print("Generating Report Now")
    print("This Will Creat a HTML File with the report data")
    utc1 = str(utc).replace(':', '_')
    f = open(str(utc1) + "_" + 'HashSearch' + ".json", "w+")
    f.write(str(json.dumps(responsefromvt, sort_keys=False, indent=4)))
    f.close()
```

```
    print("Returning to Main Menu after File Created")
    print("")
    time.sleep(5)
    clear()
    main()

def searchip():
    print("")
    print("This will Allow you to search a IP Address")
    iptosearch = input("Please Input IP to Search: ")
    responsefromvt = vtotal.ipaddress_report(iptosearch)
    # time.sleep(10)
    print("Scanning the File Now, this may take a second")
    print("")
    print("Generating Report Now")
    print("This Will Creat a HTML File with the report data")
    # For troubleShooting json output
    # print(json.dumps(responsefromvt, sort_keys=False, indent=4))
    utc1 = str(utc).replace(':', '_')
    f = open(str(utc1) + "_" + iptosearch + ".json", "w+")
    f.write(str(json.dumps(responsefromvt, sort_keys=False, indent=4)))
    f.close()
    print("Returning to Main Menu after File Created")
    print("")
    time.sleep(5)
    clear()
    main()

def reportstatus():
    print("")
    print("This section will show current report status")
    print("Hash Format Accepted Sha256")
    status = input("Please Enter your Has to Search: ")
    print("")
    responsefromvt = vtotal.file_report([status])
    print(json.dumps(responsefromvt, sort_keys=False, indent=4))
    time.sleep(20)
    main()

vtotal =
Virustotal("1889b1ae9287d8d3f55ab4e4ab0baa501849cfa09d55dd30021a317c35a64f6d")
utc = datetime.datetime.now().time()
main()
```

Reference

PyPI. (2020). *DateTime*. [online] Available at: https://pypi.org/project/DateTime/ [Accessed 9 Jan. 2020].

PyPI. (2020). *ezhashlib*. [online] Available at: https://pypi.org/project/ezhashlib/ [Accessed 9 Jan. 2020].

PyPI. (2020). *json262*. [online] Available at: https://pypi.org/project/json262/ [Accessed 9 Jan. 2020].

PyPI. (2020). *virustotal-python*. [online] Available at: https://pypi.org/project/virustotal-python/ [Accessed 9 Jan. 2020].

Python, H., Hunt, R. and Mishra, O. (2020). *Hashing a file in Python*. [online] Stack Overflow. Available at: https://stackoverflow.com/questions/22058048/hashing-a-file-in-python [Accessed 10 Jan. 2020].

Python.org. (2020). *Welcome to Python.org*. [online] Available at: https://www.python.org/doc/ [Accessed 9 Jan. 2020].

VirusTotal. (2020). *Getting started*. [online] Available at: https://developers.virustotal.com/reference#file-scan-upload-url [Accessed 9 Jan. 2020].

VirusTotal. (2020). *Getting started*. [online] Available at: https://developers.virustotal.com/reference#file-scan-upload-url [Accessed 9 Jan. 2020].