Project:

SAM Toolkit

Cyber Security

A Project Report
Submitted in partial fulfillment of the
Requirements for the award of the Degree of

BACHELOR OF SCIENCE (COMPUTER SCIENCE)
BY

ABDUL MANNAN MAQBOOL NADKAR
Seat Number ()

Under the esteemed guidance of
Mrs. SAJELI M.BUTALA

COLLEGE LOGO

DEPARTMENT OF COMPUTER SCIENCE
ICS COLLEGE OF ARTS COMERCE AND SCIENCE KHED
(Affiliated to University of Mumbai)
KHED, PIN CODE: 415709
MAHARASHTRA
YEAR (2021-2022)

CERTIFICATE

This is to certify that the project entitled “**SAM Toolkit for Cyber Security”** is bonafide work **ABDUL MANNAN MAQBOOL NADKAR** Exam Seat No.___submitted in partial of the requirement for the award of degree **Bachelor of Science (COMPUTER SCIENCE) f**rom university of **Mumbai.**

Internal Guide **Coordinator**

External Examiner

Date: **College Seal**

ABSTRACT

Security and Monitoring toolkits made for cyber security are written mostly in python, because python is more powerful when working on backend tasks. Cybersecurity is security as it is applied to information technology. This includes all technology that stores, manipulates, or moves data, such as computers, data networks, and all devices connected to or included in networks, such as routers and switches. All information technology devices and facilities need to be secured against intrusion, unauthorized use, and vandalism. Additionally, the users of information technology should be protected from theft of assets, extortion, identity theft, loss of privacy and confidentiality of personal information, malicious mischief, damage to equipment, business process compromise, and the general activity of cybercriminals. The public should be protected against acts of cyberterrorism, such as the compromise or loss of the electric power grid.

Acknowledgement

It's my great pleasure to take this opportunity and sincerely thanks all those, who have showed me the way to successful project and helped me a lot during the completion of my project.

I greatly thank my Project Guide **PROF.SAJELI M.BUTALA** without whom the completion of this project couldn't have been possible.

I take this opportunity to express my deep gratitude towards all the members of the Computer Science. Department, for helping me in the completion of the project.

My sincere thanks to respect **Principal Dr. G .B Sarang** and Head of Computer Science Department **Prof. Sachin Bhosle** for providing all the facilities including availability of Computer Lab.

I am greatly thanks teaching & Nonteaching staff of Computer Science dept. of ICS College (Science & Commerce & Arts) Khonde Tal. khed, Dist. Ratnagiri.

Finally, I am thank to my all my friends for their encouragement & support throughout the period of completion.

Mr. A.MANNAN M.NADKAR

TY.BSC(CS)

DECLARATION

I hereby declare that the project entitled, “**SAN Toolkit for Cyber Security**” done at **ICS College Khed**, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree of **BACHELOR OF SCIENCE (COMPUTER SCIENCE)** to be submitted project as part of our curriculum.

Name and Signature of the student

INDEX

Chapter 1:

Introduction

SAM (Security And Monitoring) Toolkit is a Cyber Security tool which enables a user (student, researcher, auditor, etc.) to work on various hacking methodologies like security auditing, monitoring network traffic and system files, cryptography, etc.

The framework is used to perform various attacks on self network/devices to scan and exploit vulnerabilities and to monitor malware analysis and other processes with the monitoring features available in the framework. The project is completely made in Python.

In this project many powerful python libraries are imported to achieve the goal such as urllib2, smtplib, pandas, numpy, etc.

Computer security software or cybersecurity software is any computer program designed to influence information security. This is often taken in the context of defending computer systems or data, yet can incorporate programs designed specifically for subverting computer systems due to their significant overlap, and the adage that the best defense is a good offense.

The defence of computers against intrusion and unauthorized use of resources is called computer security. Similarly, the defence of computer networks is called network security.

BACKGROUND OF THE PROJECT

Our project will provide a command-line interface. To use this tool a user must know how to deal with commands in any OS. The cli interface is designed based on the features of the tool. This tool is designed for easy to use users. And it's designed to support any version of operating system. Most of the features of the tool will be working at the background. The tool is completely designed and written in python. The user interface will also have the option to choose any feature.

OBJECTIVES

- The main purpose of the project is to provide best security testing services in private enterprises.
- The goals is to keep it simple and perfect for working in companies and enterprises.
- Performing a strategic literature review and investigation of ongoing security efforts.
- The goal is to keep the framework open source and powerful, which makes the user modify the scripts as per their requirements.
- Review state of the art technologies across multiple disciplines.

PURPOSE

The main purpose of the project is to secure and monitor private/public enterprises, networks and systems. This Cyber Security project covers the understanding and knowledge of programming, networking and system administration processes.

SCOPE

Good cyber security toolkit is all about managing risks. The process for improving and governing cyber security will be similar to the process you use for other organisational risks. It is a continuous, iterative process and comprises three overlapping components, summarised below:

1. Get the information you need to make well informed decisions on the risks you face.
2. Use this information to understand and prioritize your risks.
3. Take steps to manage those risks.

LITERATUR SURVEY

A. About Security Testing Methodology:

One of the basic tenets of software development is that you can’t control the thing which you can’t measure. There is nothing different in Security testing. Unfortunately, measuring security is a quiet difficult process. First of all what do we mean by testing? During the development phase of a web application there are many things that are need to be tested.

The Merriam-Webster Dictionary describes testing as:

- to put to test or proof.
- to be assigned a standing or evaluation based on tests.

In simple words testing is a process of comparing a system’s state against a set of internationally accepted standards or an ideal system. In the security industry such tests are done against a set of criteria that are neither well defined nor complete. For this reason security testing is sometimes known as black art.

B. Automatic Testing:

When security testing is done with the help of click and scan type tools then such kind of Scanning is called Automatic Testing. There are many tools available in market for this purpose; some are open source while others are commercial tools. The role of automated tools. As we know Automated security analysis and testing tools are sold by a number of companies in day to day market. Limitations of these tools should be kept in mind so that you can use them for what they’re good at.

Chapter 2: System Analysis

Existing System

This chapter deals with description of the work conducted on features and processes.
Applications related to security.

Following are existing applications related to conversion:

In the earlier python based security tool, there was limited number of facility and was limited to features. So whenever we are in need of extra features, we have to use other security services and sometimes the use of manual operations. Many of security softwares, doesn't allow us to work on monitoring processes within the firewall, so that we can use it whenever we are in need of it.
This frameworks aims at covering every part of system to monitor and run many scripts.

Problem in existing system:

1) Less security service.
2) All processes were done manually by executing scripts..
3) The ***steganography*** feature was not implemented in the framework.
4) The manual system was time consuming.
5) Lack of better facilities.

Limitations of present system:

There are certain limitations in system, which are as follows:-

- The tool doesn’t have any GUI layout.
- Python should be installed in system.
- User must have a good understanding of CLI.
- No machine learning algorithms are used for better results.

Fact-Finding Techniques

Fact-finding technique is one of the parts of the system analysis. At the time of analysis of the system or before starting actual work, system analysts collect the information. For gathering information system analyst prefers any fact-finding technique such as:

1] **Interview.**

2] **Questionnaire.**

3] **Record review.**

4] **Observation.**

While developing this system we have done this part by using interview & record-review techniques.

1] **Interview:-**

The whole system investigation part has done by taking interview of the concerned people, user & staff. By asking them manual process of each work, by pointing hints regarding the work, by discussing their problems deeply, by asking them their requirements, by taking their valuable suggestion guidance regarding system study.

After taking their interviews regarding the systems study noting down the points regarding the system. This work of taking interview-asking difficulties to concern person till all the points of the system understood. This interview technique proved beneficial outcome for system analysis.

2] Questionnaire:-

In this method we actually provide a list of questions to the user. According to the list the user answers the questions, taking his own time without stress to answer quickly. In this method the results Obtained are rather accurate and thoughtfully given. Questionnaires can be an effective method for gathering facts.
E.g. how your system actually works?

3] Recordreview:-

Review of record is good but a tedious way to retrieve information from an organization. This refers to a personal viewing of records.

4] Observation: -

This technique proves useful in finding the facts of the system.
Thus by doing their techniques of the fact finding method different facts of the system gathered.

Proposed system

Due to network traffic is forwarded through the router to the Internet, it is possible to gather the data of the wireless devices connected to the network. A MITM attack eavesdrops and manipulates sensitive data of the users, using network traffic analysis or packet sniffing tools that intercept the incoming and outgoing network traffic. A passive MITM attack sniffs information of the users connected to the Access Point (AP) in a second plane. To overcome from these attacks, SAM toolkit is made handy to perform these attacks before the attacker harm your system and patch the vulnerability found in the system.

Features of this project

1. This python tool can able to run on any type of O.S.
2. This tool can be run on any version of distribution.
3. Steganography module developed for advance image encryption.
4. Unique and User friendly Interface.
5. No signup or login required.
6. Can run online as well as offline attacks.
7. Sharable.

Feasibility study

The feasibility study proposes one or more conceptual solutions to the problem set for project. The conceptual solution gives an idea of how the system looks like. The following three things should be done to establishfeasibility

1] **Technical feasibility:-**

Technical feasibility study determines whether the organization has technology and the skills necessary to carry out the project. If not then it determines how they should be obtained. In this proposed system, the existing technology satisfies the need for the system that is available. Therefore the system is feasible.

Technical feasibility study determines whether the organization has the technology & the skills necessary to carry out the project. If not then it determines how they should be.

2] **Operational Feasibility: -**

Operational feasibility study determines if the proposed system satisfies the user's objectives & can be fitted into the current system operation. Our proposed system will certainly satisfy the user's objective & it will enhance their capability. The proposed system can be best fitted instead of the current system operation. Therefore, the system is operationallyfeasible.

3] **Economic Feasibility:-**

Economic feasibility study determines whether the project goals can be achieved within the limits allocated to it. It must also determine that the benefits obtained from the new system are not worth the cost.

System Requirements:

➢ Hardware Requirements:

- **Operating System :**
  - ✓ Any OS
- **Ram :**
  - ✓ Minimum 1 Gb ram.
- **Disk Space :**
  - ✓ Minimum 100 mb for changes being applied.

Software Requirements:

- **Software installations :**
  - ✓ Python 2.7 or higher
  - ✓ Any Code Editor for making changes in code.
  - ✓ Installed python libraries for working with scripts.

Justification of selection of technology

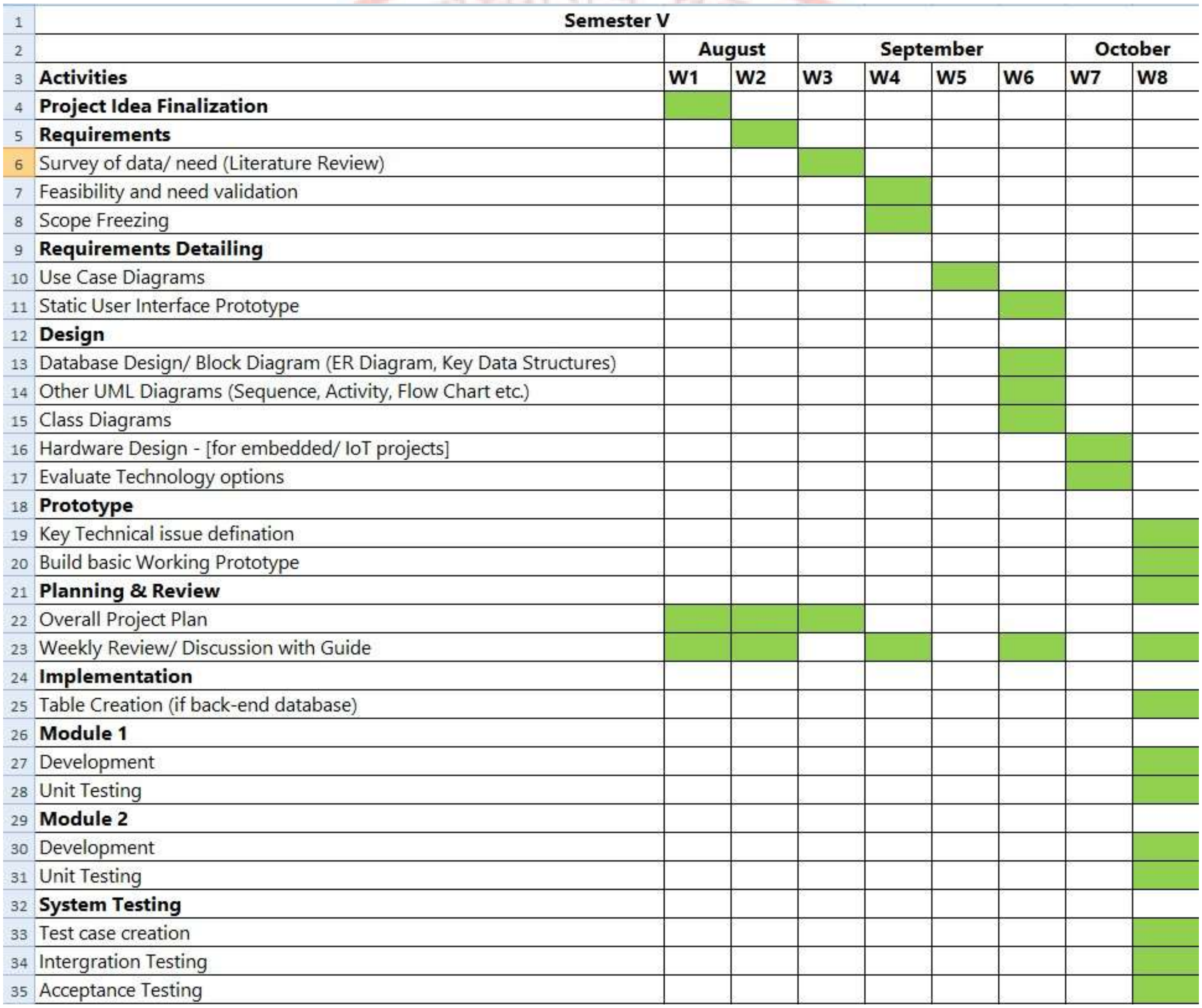
In the world of cyber space, there are many framework and languages that developers can choose from. Each framework has features and benefits that make them different. But I have selected desktop with Python for the development of my project. There are plenty of good reasons to use Python and developing an system security application:- high speed low cost and vast language support are among the most significant benefits. It gives you full control of your development and can easily be used on any project, big or small and Abdation can be done using this language.

Python : Python is a general-purpose, interpreted and high-level OOPs based dynamic programming language that focuses on rapid application development and don't repeat yourself. Due to the ease of syntax in Python, the programmers can complete coding in fewer steps as compared to Java or C++. Python is considerably one of the fastest-growing languages. Python's ever-evolving libraries and support make it a viable choice for any project, be it Mobile App, Web App, IoT, Data Science or AI.

Why choose Python? Most of the software development companies are choosing Python because of its fewer programming codes and versatile features. Approximately, 14% of the programmers use operating systems like Linux, UNIX, Windows and Mac OS. However, the programmers of big companies make use of Python as it has created a trademark for itself with characteristic features like :

- Object-oriented
- Interpreted
- Modular
- Dynamic
- Portable
- High level
- Extensible in C++ & C

Gantt chart

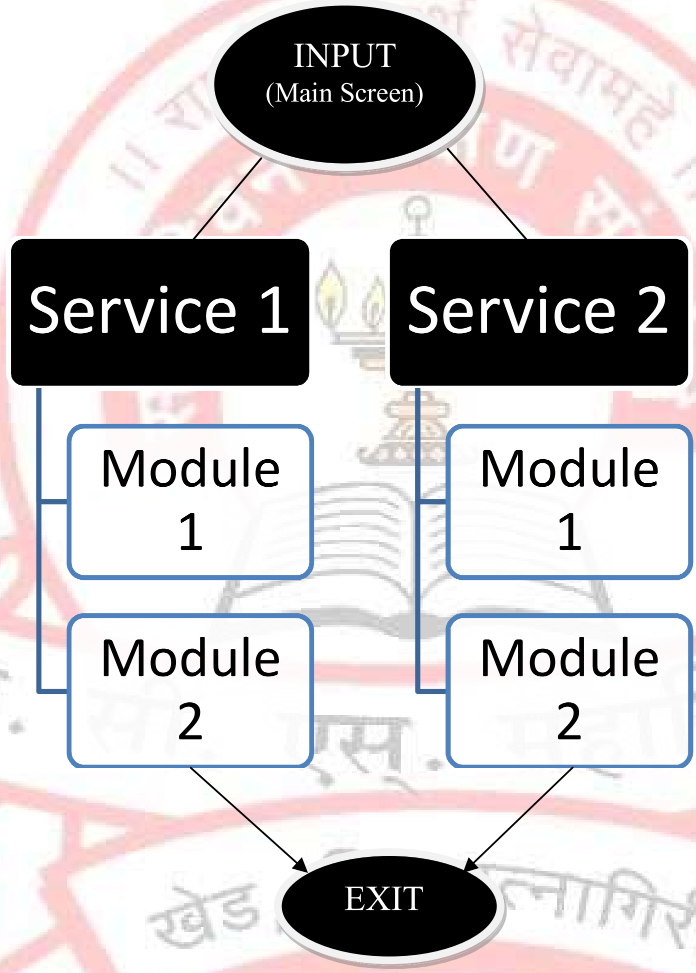


1	Semester V								
2		August		September				October	
3	Activities	W1	W2	W3	W4	W5	W6	W7	W8
4	**Project Idea Finalization**	🟩							
5	**Requirements**		🟩						
6	Survey of data/ need (Literature Review)			🟩					
7	Feasibility and need validation				🟩				
8	Scope Freezing				🟩				
9	**Requirements Detailing**								
10	Use Case Diagrams					🟩			
11	Static User Interface Prototype						🟩		
12	**Design**								
13	Database Design/ Block Diagram (ER Diagram, Key Data Structures)						🟩		
14	Other UML Diagrams (Sequence, Activity, Flow Chart etc.)						🟩		
15	Class Diagrams						🟩		
16	Hardware Design - [for embedded/ IoT projects]							🟩	
17	Evaluate Technology options							🟩	
18	**Prototype**								
19	Key Technical issue defination								🟩
20	Build basic Working Prototype								🟩
21	**Planning & Review**								🟩
22	Overall Project Plan	🟩	🟩	🟩					
23	Weekly Review/ Discussion with Guide	🟩	🟩		🟩		🟩		🟩
24	**Implementation**								
25	Table Creation (if back-end database)								🟩
26	**Module 1**								
27	Development								🟩
28	Unit Testing								🟩
29	**Module 2**								
30	Development								🟩
31	Unit Testing								🟩
32	**System Testing**								
33	Test case creation								🟩
34	Intergration Testing								🟩
35	Acceptance Testing								🟩

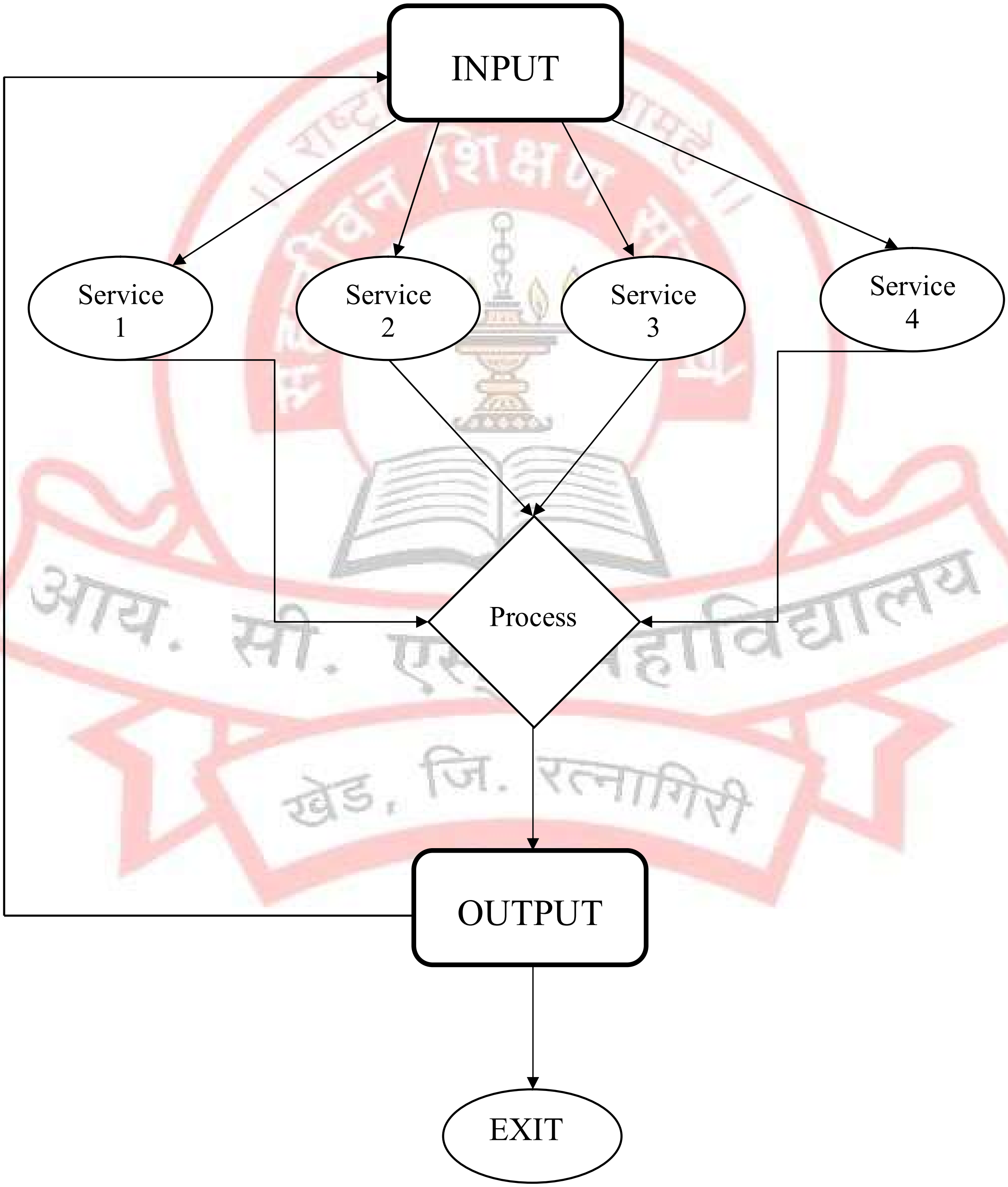
Chapter 3: System Design

Firstly, we will design a MAIN screen where the users would interact with the framework. While making the CLI simple, the backend work is also formatted in systematic manner.

- **Work Flow Design:**
  - Diagram



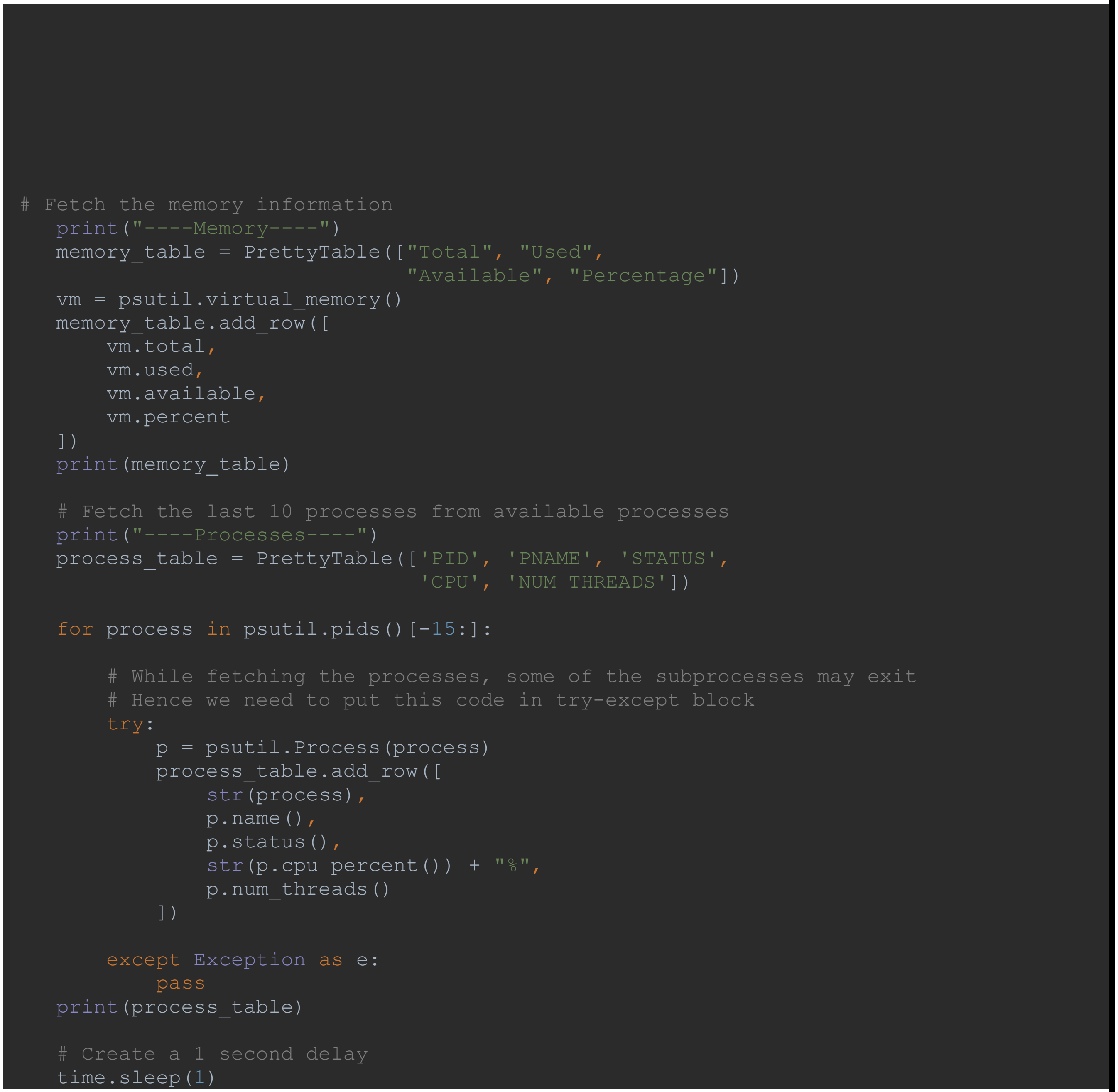
- **Data Flow Design:**



➢ Code Analysis:

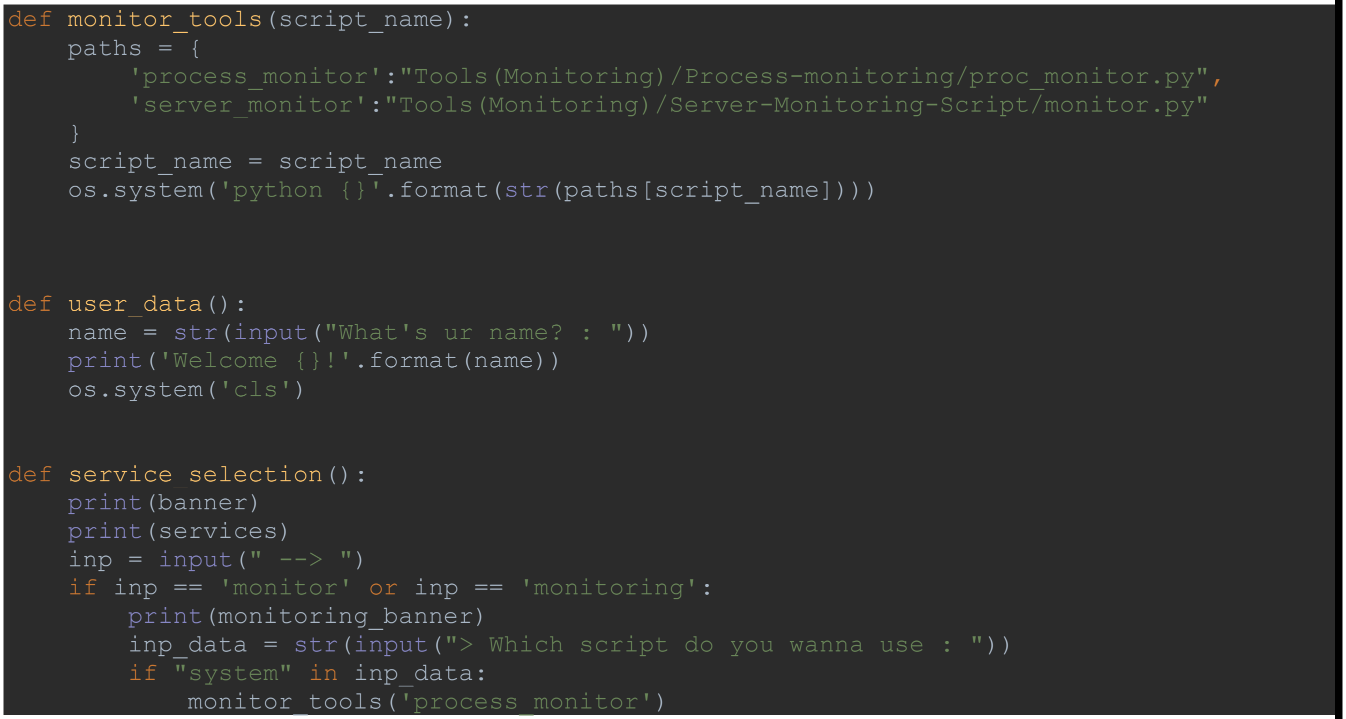
- Some tools code :

```
# Import the required libraries
import psutil
import time
import os
from subprocess import call
from prettytable import PrettyTable

# Run an infinite loop to constantly monitor the system
while True:

    # Clear the screen using a bash command
    os.system('cls')
    print("==============================Process
Monitor\======================================")

    # Fetch the battery information
    battery = psutil.sensors_battery().percent
    print("----Battery Available: %d " % (battery,) + "%")

    # We have used PrettyTable to print the data on console.
    # t = PrettyTable(<list of headings>)
    # t.add_row(<list of cells in row>)

    # Fetch the Network information
    print("----Networks----")
    table = PrettyTable(['Network', 'Status', 'Speed'])
    for key in psutil.net_if_stats().keys():
        name = key
        up = "Up" if psutil.net_if_stats()[key].isup else "Down"
        speed = psutil.net_if_stats()[key].speed
        table.add_row([name, up, speed])
    print(table)
```



```
# Fetch the memory information
   print("----Memory----")
   memory_table = PrettyTable(["Total", "Used",
                               "Available", "Percentage"])
   vm = psutil.virtual_memory()
   memory_table.add_row([
       vm.total,
       vm.used,
       vm.available,
       vm.percent
   ])
   print(memory_table)

   # Fetch the last 10 processes from available processes
   print("----Processes----")
   process_table = PrettyTable(['PID', 'PNAME', 'STATUS',
                                'CPU', 'NUM THREADS'])

   for process in psutil.pids()[-15:]:

       # While fetching the processes, some of the subprocesses may exit
       # Hence we need to put this code in try-except block
       try:
           p = psutil.Process(process)
           process_table.add_row([
               str(process),
               p.name(),
               p.status(),
               str(p.cpu_percent()) + "%",
               p.num_threads()
           ])

       except Exception as e:
           pass
   print(process_table)

   # Create a 1 second delay
   time.sleep(1)
```

- Functions :



```
def monitor_tools(script_name):
    paths = {
        'process_monitor':"Tools(Monitoring)/Process-monitoring/proc_monitor.py",
        'server_monitor':"Tools(Monitoring)/Server-Monitoring-Script/monitor.py"
    }
    script_name = script_name
    os.system('python {}'.format(str(paths[script_name])))



def user_data():
    name = str(input("What's ur name? : "))
    print('Welcome {}!'.format(name))
    os.system('cls')


def service_selection():
    print(banner)
    print(services)
    inp = input(" --> ")
    if inp == 'monitor' or inp == 'monitoring':
        print(monitoring_banner)
        inp_data = str(input("> Which script do you wanna use : "))
        if "system" in inp_data:
            monitor_tools('process_monitor')
```

➢ Module Classification :

The Tool is basically having a Two modules.

- SECURITY
- MONITORING

➢ The Security module contains various tools like :

- Cryptographic tools :
  - Text data encryption
  - Image Steganography
- Port scanner
- Admin scanner

➢ The Monitoring module contains tools like :

- System Process Monitor
- Server Process Monitor
- Keylogger
- Network Traffic Analyzer

In many of the systems the firewall or sysadmins doesn’t allows a user to run softwares or scripts without permission or it blocks access through the network. By using this tool a user has full access to use any service without blockage of any firewalls.

➢ Security Tools :

- **Cryptographic tools**:

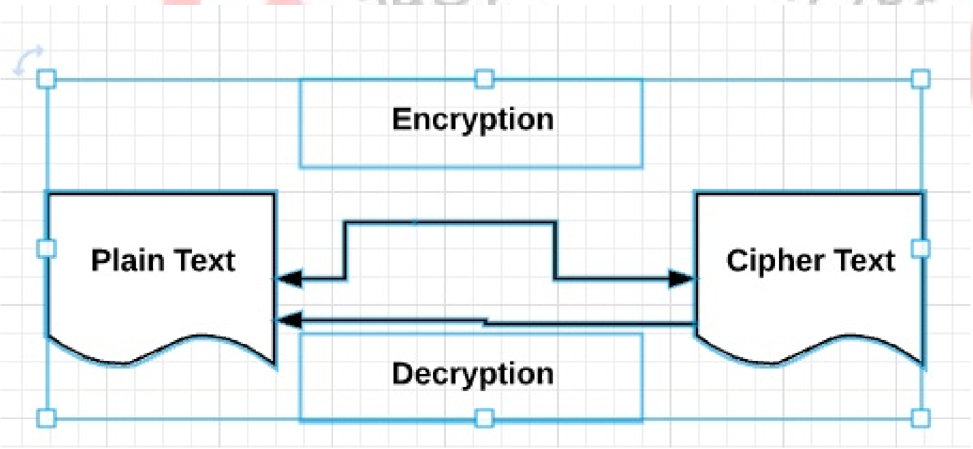
- Cryptography is the art of communication between two users via coded messages. The science of cryptography emerged with the basic motive of providing security to the confidential messages transferred from one party to another.
- Cryptography is defined as the art and science of concealing the message to introduce privacy and secrecy as recognized in information security.

- Terminologies of Cryptography
  The frequently used terms in cryptography are explained here –

- Plain Text
- The plain text message is the text which is readable and can be understood by all users. The plain text is the message which undergoes cryptography.
- Cipher Text
- Cipher text is the message obtained after applying cryptography on plain text.

- Encryption
- The process of converting plain text to cipher text is called encryption. It is also called as encoding.
- Decryption
- The process of converting cipher text to plain text is called decryption. It is also termed as decoding.

- The diagram given below shows an illustration of the complete process of cryptography –



- Image Steganography :

  Steganography is the practice of concealing a message within another message or a physical object. In computing/electronic contexts, a computer file, message, image, or video is concealed within another file, message, image, or video.

- **Port Scanner**:

  This Port Scanner will work for both the Web Applications as well as remote Host. This tool has been created to provide the basic functionality of a Port Scanner. The general concept of Sockets had been used to provide the functionality. Port Scanner is built on Python 3 and uses some extra libraries such as socket and pyfiglet (for a fancy banner).

- **Admin Scanner** :

  Admin-Scanner is an automated python language script that can detect the admin pages on the target domain. If you are an attacker and trying to penetrate the domain, these admin pages can help you. If you find any misconfigured admin page, you can easily be the domain administrator and make changes as per your choice. Admin-Scanner tool provides you with inbuilt wordlists, which are brute-forced on the target domain to detect admin pages.

Features of Admin-Scanner Tool

1. Admin-Scanner is open-source and free to use.
2. Admin-Scanner is Python language-based tool.
3. Admin-Scanner is an automated tool for finding admin pages.
4. Admin-Scanner provides the feature to customize and use wordlists.
5. Admin-Scanner provides the feature to set the value of the thread for efficient usage.
6. Admin-Scanner is easy to use.

➢ Monitoring Tools :

- **System Process Monitor:**

The tool gives the programmer the ability to monitor the system for certain events, and then receive callbacks when those events occur. We're going to leverage this interface to receive a callback every time a process is created. When a process gets created, we're going to trap some valuable information for our purposes: the time the process was created, the user that spawned the process, the executable that was launched and its command-line arguments, the process ID, and the parent process ID. This will show us any processes that are created by higher-privilege accounts, and in particular, any processes that are calling external files such as VBScript or batch scripts. When we have all of this information, we'll also determine what privileges are enabled on the process tokens. In certain rare cases, you'll find processes that are created as a regular user but which have been granted additional Windows privileges that you can leverage.

There are two scripts in the subfolder name monitor.vbs and monitor .bat. Adding these files in Task Scheduler gives the ability to run the process in background.

- **Server Process Monitor:**

Python Server Monitoring Script enables you to monitor if your server or computer is active and running. It can show you how much downtime your computer or server had. We will be using server sockets to check if the specific port on a server is open or not, the Ping command to monitor a simple regular computer and the SSL to check if the particular server requires an SSL connection or not.

- **File Monitor :**

In order to exploit file monitoring processes, we have to effectively win a race against the executing code. When the software or scheduled task creates the file, we need to be able to inject our own code into the file before the process executes it and then ultimately deletes it. The trick to this is the handy Windows API called *ReadDirectoryChanges*, which enables us to monitor a directory for any changes to files or subdirectories. We can also filter these events so that we're able to determine when the file has been "saved" so we can quickly inject our code before it's executed. It can be incredibly useful to simply keep an eye on all temporary directories for a period of 24 hours or longer, because sometimes you'll find interesting bugs or information disclosures on top of potential privilege escalations.

- **Keylogger** :

Keystroke logging is the process of recording (logging) the keys pressed on a keyboard (usually when the user is unaware). It is also known as keylogging or keyboard capturing.
These programs are used for troubleshooting technical problems with computers and business networks. It can also be used to monitor network usages but more often than not it is used for malicious intents like stealing passwords.

- **Network Packet Sniffer :**

Sniffing or network packet sniffing is the process of monitoring and capturing all the packets passing through a given network using sniffing tools. It is a form wherein, we can “tap phone wires” and get to know the conversation. It is also called wiretapping and can be applied to the computer networks.

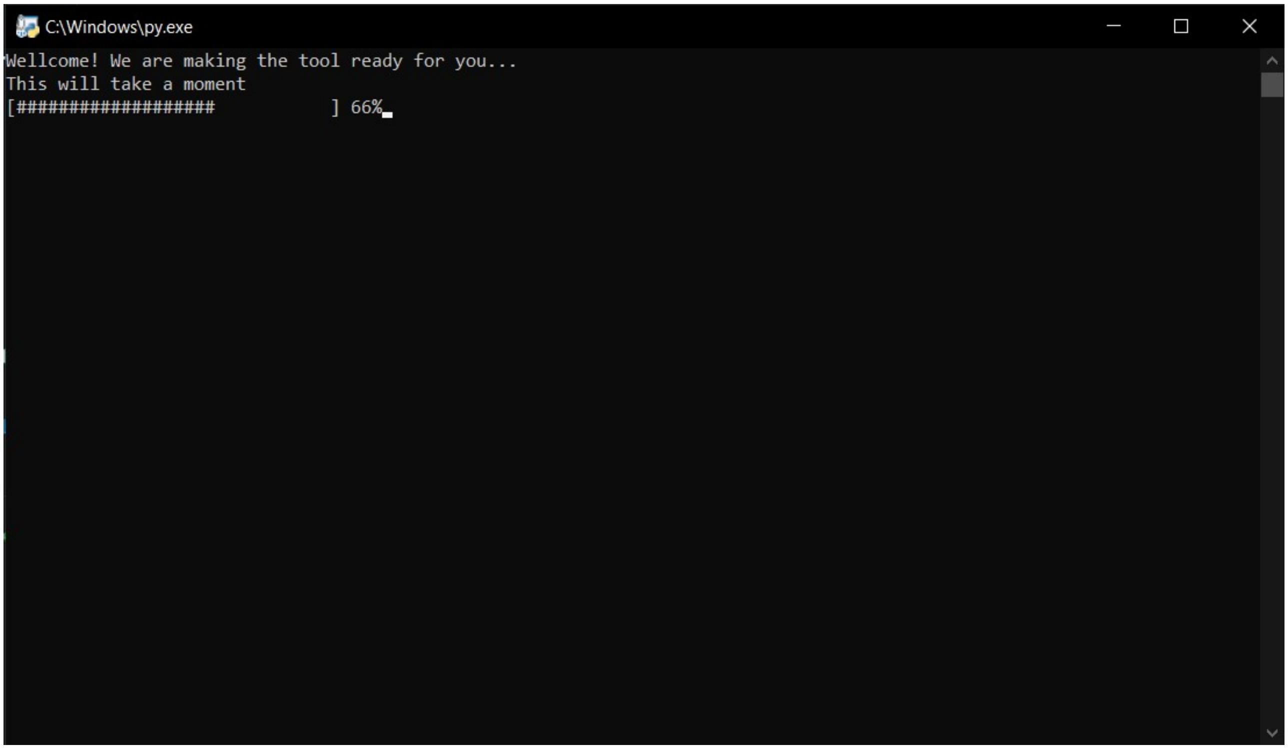
There is so much possibility that if a set of enterprise switch ports is open, then one of their employees can sniff the whole traffic of the network. Anyone in the same physical location can plug into the network using Ethernet cable or connect wirelessly to that network and sniff the total traffic.

In other words, Sniffing allows you to see all sorts of traffic, both protected and unprotected. In the right conditions and with the right protocols in place, an attacking party may be able to gather information that can be used for further attacks or to cause other issues for the network or system owner.

A simple packet sniffer in Python can be created with the help socket module. We can use the raw socket type to get the packets. A raw socket provides access to the underlying protocols, which support socket abstractions. Since raw sockets are part of the internet socket API, they can only be used to generate and receive IP packets.

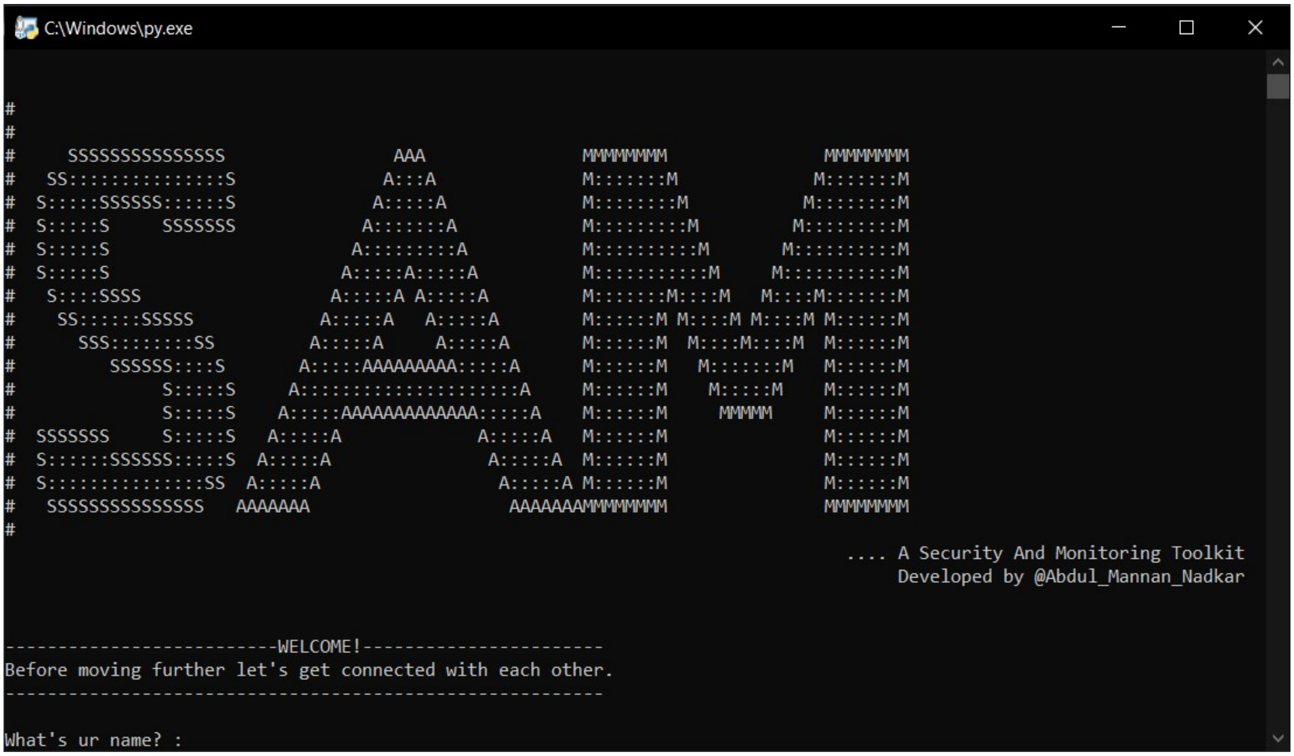
Output screen

- Main screen of the tool at start :



```
Wellcome! We are making the tool ready for you...
This will take a moment
[####################          ] 66%
```

- Main Screen :

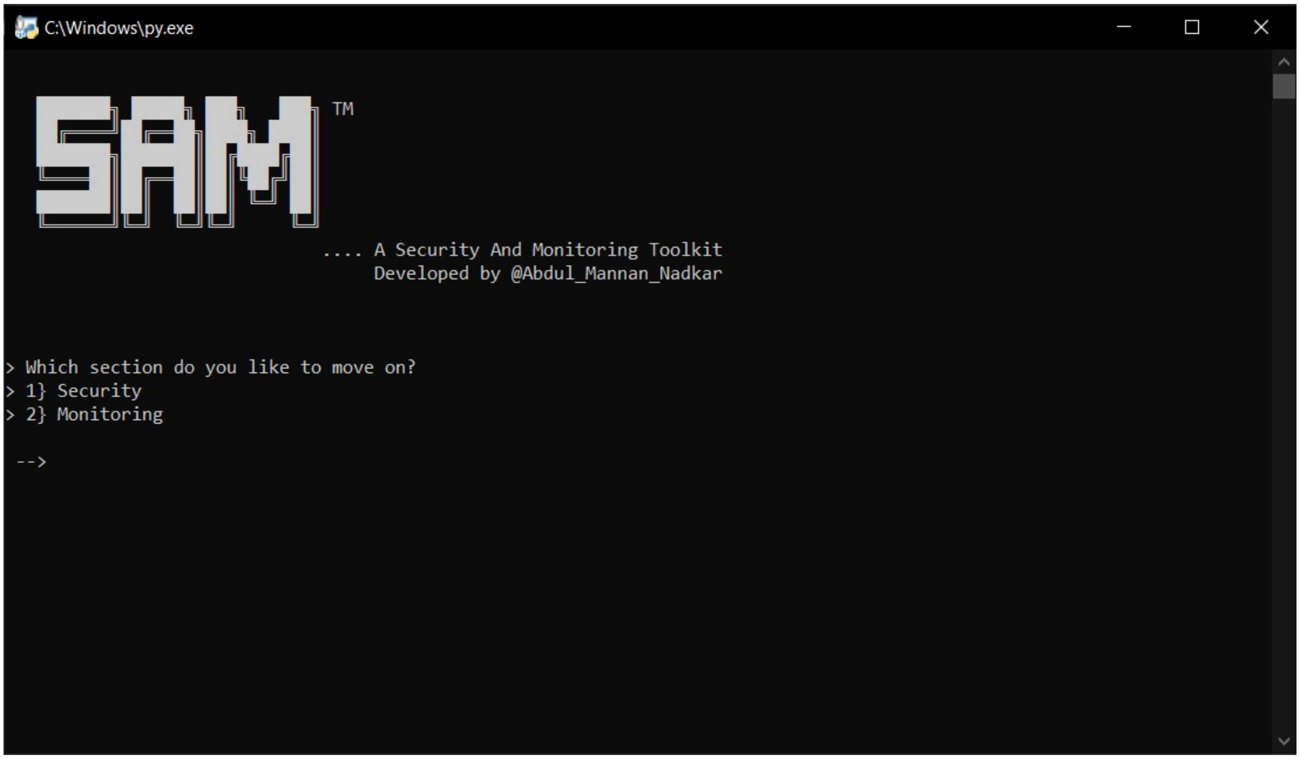


```
#
#
#     SSSSSSSSSSSSSSS                AAA               MMMMMMMM               MMMMMMMM
#   SS:::::::::::::::S              A:::A              M:::::::M             M:::::::M
#  S:::::SSSSSS::::::S             A:::::A             M::::::::M           M::::::::M
#  S:::::S     SSSSSSS            A:::::::A            M:::::::::M         M:::::::::M
#  S:::::S                       A:::::::::A           M::::::::::M       M::::::::::M
#  S:::::S                      A:::::A:::::A          M:::::::::::M     M:::::::::::M
#   S::::SSSS                  A:::::A A:::::A         M:::::::M::::M   M::::M:::::::M
#    SS::::::SSSSS            A:::::A   A:::::A        M::::::M M::::M M::::M M::::::M
#      SSS::::::::SS         A:::::A     A:::::A       M::::::M  M::::M::::M  M::::::M
#         SSSSSS::::S       A:::::AAAAAAAAA:::::A      M::::::M   M:::::::M   M::::::M
#              S:::::S     A:::::::::::::::::::::A     M::::::M    M:::::M    M::::::M
#              S:::::S    A:::::AAAAAAAAAAAAA:::::A    M::::::M     MMMMM     M::::::M
#  SSSSSSS     S:::::S   A:::::A             A:::::A   M::::::M               M::::::M
#  S::::::SSSSSS:::::S  A:::::A               A:::::A  M::::::M               M::::::M
#  S:::::::::::::::SS  A:::::A                 A:::::A M::::::M               M::::::M
#   SSSSSSSSSSSSSSS   AAAAAAA                   AAAAAAAMMMMMMMM               MMMMMMMM
#
                                                                                .... A Security And Monitoring Toolkit
                                                                                     Developed by @Abdul_Mannan_Nadkar


--------------------------WELCOME!-----------------------
Before moving further let's get connected with each other.
---------------------------------------------------------

What's ur name? :
```

- **Module Section:**

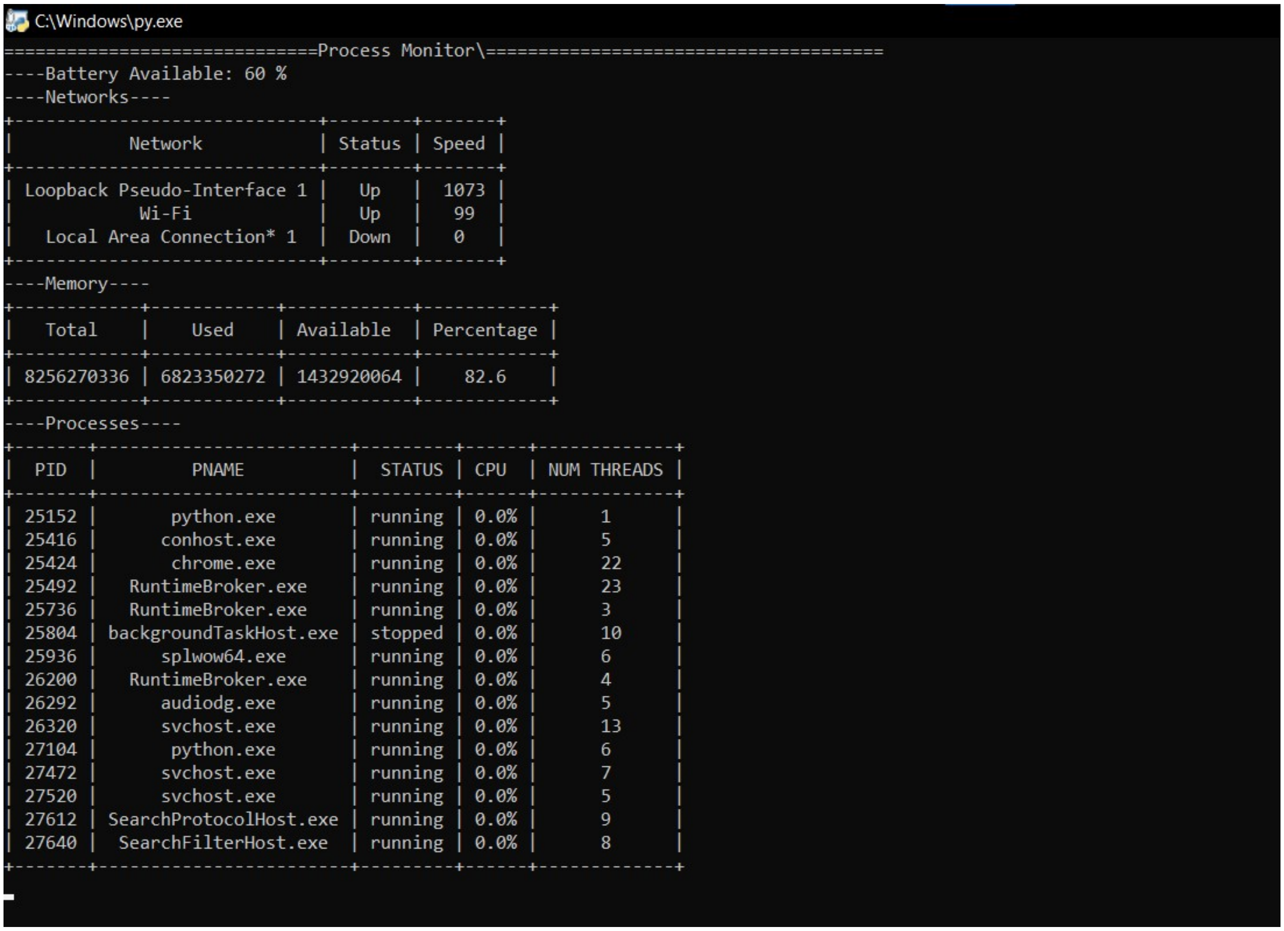
  There are two sections to move in

  1) Security
  2) Monitoring



```
                                        .... A Security And Monitoring Toolkit
                                             Developed by @Abdul_Mannan_Nadkar



> Which section do you like to move on?
> 1} Security
> 2} Monitoring

 -->
```

- Process Monitor :



```
==============================Process Monitor\====================================
----Battery Available: 60 %
----Networks----
+-----------------------------+--------+-------+
|           Network           | Status | Speed |
+-----------------------------+--------+-------+
| Loopback Pseudo-Interface 1 |   Up   |  1073 |
|            Wi-Fi            |   Up   |   99  |
|   Local Area Connection* 1  |  Down  |   0   |
+-----------------------------+--------+-------+
----Memory----
+------------+------------+------------+------------+
|   Total    |    Used    | Available  | Percentage |
+------------+------------+------------+------------+
| 8256270336 | 6823350272 | 1432920064 |    82.6    |
+------------+------------+------------+------------+
----Processes----
+-------+------------------------+---------+------+-------------+
|  PID  |         PNAME          |  STATUS | CPU  | NUM THREADS |
+-------+------------------------+---------+------+-------------+
| 25152 |       python.exe       | running | 0.0% |      1      |
| 25416 |      conhost.exe       | running | 0.0% |      5      |
| 25424 |       chrome.exe       | running | 0.0% |      22     |
| 25492 |   RuntimeBroker.exe    | running | 0.0% |      23     |
| 25736 |   RuntimeBroker.exe    | running | 0.0% |      3      |
| 25804 | backgroundTaskHost.exe | stopped | 0.0% |      10     |
| 25936 |      splwow64.exe      | running | 0.0% |      6      |
| 26200 |   RuntimeBroker.exe    | running | 0.0% |      4      |
| 26292 |      audiodg.exe       | running | 0.0% |      5      |
| 26320 |      svchost.exe       | running | 0.0% |      13     |
| 27104 |       python.exe       | running | 0.0% |      6      |
| 27472 |      svchost.exe       | running | 0.0% |      7      |
| 27520 |      svchost.exe       | running | 0.0% |      5      |
| 27612 | SearchProtocolHost.exe | running | 0.0% |      9      |
| 27640 |  SearchFilterHost.exe  | running | 0.0% |      8      |
+-------+------------------------+---------+------+-------------+
```

Conclusion and

Future Enhancement

Conclusion

This paper compared three network monitoring systems which are popular in their aspect. The suitability of the system, for small businesses and large organizations, has been presented. The tools compared are robust in monitoring systems these days. The differences in these tools have brought out the limitations and advantages of each of them. The importance, of pricing and open source, is also presented in the paper hence leading to the suitability of these systems for particular organizations.

Future Enhancement

Future work will involve improving the library and adding more modules possible future data sets and modules that can be added include web security data, phishing data, steganography data, etc. Additionally, we will provide more use cases of how to use the library in various other applications. There will be more web security frameworks added in future for Web App Peneteration Testing.

This app can be enhanced in very attractive manner in future:

- More features and tools will be added.
- Will be published on Github.
- The tool will be packed in executable (.exe) format.

Reference

[1] International Research Journal of Engineering and Technology (IRJET). Comparative Study on Network Monitoring Tools.
Link - https://www.irjet.net/archives/V7/i4/IRJET-V7I464.pdf

[2] https://www.uv.mx/personal/angelperez/files/2018/10/sniffers_texto.pdf

[3] Black Hat Python Book.

[4] GeeksforGeeks

[5] Clarke, Russell, David Dorwin, and Rob Nash. "Is open source software more secure?." Homeland Security/Cyber Security, 2009.

[6] Nagios. The Industry Standard in IT Infrastructure Monitoring. http://www.nagios.org.Accessed March 24, 2020.

[7] The must-haves of a network monitoring software. https://www.manageengine.com/networkmonitoring/whitepaper-network-monitoringessentials.html. Accessed March 28, 2020.

[8] Isaac Sikubwabo, Mariam Usanase, Dr. Papias Niyigena. “Comparative Study on Network Monitoring Tools of Nagios Versus Hyperic”, 2019.

[9] Comparison of Network Monitoring Systems. https://en.wikipedia.org/wiki/Comparison_of_networkmonitoring_systems.html. Accessed April 3, 2020.