

Mini Project Report
On
INTRUDER IDENTIFICATION SYSTEM
VI SEMESTER
INFORMATION TECHNOLOGY

Submitted by

NIKHIL PAONIKAR
PRATIK KHANKE
ROHIT GHUNGRUDKAR
SAMUEL KUMAR

Under the guidance of
Prof. P. C. Golar
Assistant professor

Academic Year 2014-15

Department of Information Technology



ST. VINCENT PALLOTTI COLLEGE OF ENGINEERING
AND TECHNOLOGY
Wardha Road, Gavsi Manapur, Nagpur

ST. VINCENT PALLOTTI COLLEGE OF ENGINEERING AND TECHNOLOGY-NAGPUR

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

Certified that this project report “**INTRUDER IDENTIFICATION SYSTEM**” is the bona fide work of “**NIKHIL PAONIKAR, PRATIK KHANKE, ROHIT GHUNGRUDKAR and SAMUEL KUMAR**” who carried out the mini project work under my supervision in partial fulfillment of VI Semester, Bachelor of Engineering in **INFORMATION TECHNOLOGY** of RASHTRASANT TUKADOJI MAHARAJ NAGPUR UNIVERSITY, NAGPUR.

Prof. M. V. Bramhe

Associate Professor

HEAD OF THE DEPARTMENT

Prof. P. C. Golar

Assistant Professor

GUIDE

**PRINCIPAL
ST. VINCENT PALLOTTI COLLEGE OF ENGINEERING AND
TECHNOLOGY**

ACKNOWLEDGEMENT

Our mini project is titled, “**INTRUDER IDENTIFICATION SYSTEM**”. Any project requires a lot of hard work, sincerity and systematic work methodologies. We express our deepest gratitude to our Project Guide, **Prof. P. C. GOLAR**, for giving us an opportunity to be a part of this project and guiding us in every step of the project.

We would also like to thank **Prof. M. V. Bramhe, Head of the Department of Information Technology** and all our faculty members who regularly evaluated our project and pointed out the shortcomings in the projects. They also gave us important feedback for the further improvement of our project. We are highly indebted to them.

We are also grateful to the **Management of the College, Dr. V. V. Sohoni, Principal** and **Prof. R. B. Gowardhan, Vice-Principal** for the overwhelming support in providing us the facilities of computer lab and other required infrastructure. We would like to thank our Library Department for providing us useful books related to our project.

Project group members' names

NIKHIL PAONIKAR

PRATIK KHANKE

ROHIT GHUNGRUDKAR

SAMUEL KUMAR

ABSTRACT

As the digital age takes firm root with the passage of time, our smartphones and tablets grow increasingly important to us. What started out as a simple method of communication, has turned into a warehouse of our digital selves, holding everything from photographs, sensitive text messages, important passwords, crucial financial information and more. As such, the need for protection increases too and security has become one of the top priorities in the digital world.

The old adage “Prevention is better than cure.” still holds true in today’s digital world. But, sometimes things tend to get out of hand and there’s little we can do about it. Like when someone’s computer is broken into illegally, there’s no definite way to recognize the identity of the suspect.

In information security, intruder detection is the art of detecting intruders behind attacks as unique persons. This technique tries to identify the person behind an attack by analyzing their computational behavior. This concept is sometimes confused with Intrusion detection (also known as IDS) techniques; which are the art of detecting an intruder’s actions.

Though conventional methods use computational behavior of intruders, other unconventional approaches might employ getting biometric feedback, implementing decoy interfaces and other discreet traps.

Our main objective is to create a solution for these problems by making a set of modular programs designed to monitor keystrokes, create an activity log, automate the webcam to photograph the intruder and then send all these files to the owner’s cloud storage. These files then can be viewed from any place the owner wishes to just by logging into their cloud account. In addition to the aforementioned files will also be stored locally on the hard disk.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ACKNOWLEDGEMENT	i
	ABSTRACT	ii
	LIST OF TABLES	iii
	LIST OF FIGURES	iv
1.	INTRODUCTION	1
	1.1 Intrusion detection	1
	1.2 Motivation	1
	1.3 Problem statement	2
2.	LITERATURE SURVEY	3
	2.1 Existing systems	3
	2.2 Hardware keyloggers	3
	2.3 Software keyloggers	4
	2.4 Screen recording software	4
3.	FEASIBILITY STUDY	5
	3.1 Technical feasibility	5
	3.2 Economic feasibility	6
	3.3 Operational feasibility	6
	3.4 Legal feasibility	6

4.	REQUIREMENTS	7
	4.1 Hardware requirements	7
	4.2 Software requirements	7
5.	IMPLEMENTATION	8
	5.1 Automated webcam contrivance	8
	5.2 Keystroke logging	8
	5.3 Activity logging	8
	5.4 Automated cloud transfer	8
	5.5 Enabler and disabler keys	9
	5.6 Background processing	9
6.	FLOWCHARTS	9
	6.1 Keylogger	9
	6.2 Activity logger	10
	6.3 Automated webcam contrivance	11
	6.4 Enabler key program	13
	6.5 Disabler key program	15
7.	SCREENSHOTS	17
	7.1 Program location	17
	7.2 Program enabler in startup	18
	7.3 Log files in cloud storage	19
	7.4 Log file showing logged activities and keystrokes.	21
	7.5 Logs and intruder's photographs accessed on the owner's cellphone via cloud storage.	22
	7.6 Disabler key program to disable Intruder Identification System	23

8.	PROGRAM CODES AND FEATURES	24
8.1	Automated webcam contrivance	24
8.2	Source code for Activity logger and keylogger	26
8.3	Source code for Enabler key program	33
8.4	Source code for Disabler key program	34
8.5	Source code for stealth VBScript	34
9.	CONCLUSION AND FUTURE SCOPE	35
	REFERENCES	36

LIST OF TABLES

Serial number	Table	Page number
4.1	Hardware requirements	7
4.2	Software requirements	7

LIST OF FIGURES

Serial number	Figure	Page number
6.1	Keylogger flowchart	9
6.2	Activity logger flowchart	10
6.3	Automated webcam contrivance	11
6.4	Enabler key program	13
6.5	Disabler key program	15
7.1	Program location	17
7.2	Program enabler in startup	18
7.3.1	Log files in cloud storage - 1	19
7.3.2	Log files in cloud storage - 2	20
7.4	Log file showing logged activities and keystrokes.	21
7.5	5 Logs and intruder's photographs accessed on the owner's cellphone via cloud storage.	22
7.6	Disabler key program to disable Intruder Identification System	23

Chapter No. 1

Introduction

INTRODUCTION

1.1 Intrusion detection

Intrusion detection (ID) is a type of security management system for computers and networks. An ID system gathers and analyzes information from various areas within a computer or a network to identify possible security breaches, which include both intrusions (attacks from outside the organization) and misuse (attacks from within the organization).

ID systems are being developed in response to the increasing number of attacks on major sites and networks, including those of the Pentagon, the White House, NATO, and the U.S. Defense Department. The safeguarding of security is becoming increasingly difficult, because the possible technologies of attack are becoming ever more sophisticated; at the same time, less technical ability is required for the novice attacker, because proven past methods are easily accessed through the Web.

1.2 Motivation

Intrusion identification is needed in today's computing environment because it is impossible to keep pace with the current and potential threats and vulnerabilities in our computing systems. The environment is constantly evolving and changing fueled by new technology and the Internet. To make matters worse, threats and vulnerabilities in this environment are also constantly evolving. Intrusion identification tools can assist in managing threats and vulnerabilities in this changing environment.

Threats are people or groups who have the potential to compromise your computer system. These may be a curious teenager, a disgruntled employee, or espionage from a rival company or a foreign government. The hacker has become a nemesis to many companies.

Vulnerabilities are weaknesses in the systems. Vulnerabilities can be exploited and used to compromise your system. New vulnerabilities are discovered all of the time. Every new technology, product, or system brings with it a new generation of bugs and unintended conflicts or flaws.

Intrusion identification tools can assist in protecting a company from intrusion by expanding the options available to manage the risk from threats and vulnerabilities. Intrusion identification capabilities can help a company secure its information. The tool could be used to detect an intruder, identify and stop the intruder, support investigations to find out how the intruder got in, and stop the exploit from use by future intruders.

1.3 Problem statement

The problem with the current methods to classify/identify attackers is, that they are not effectively implemented in the real world, in a standardized manner.

Detecting the identity of the attacker merely on the basis of one's computational behavior is not enough. There need to be systemized programs and/or protocols put into use to identify a perpetrator's identity.

The solution we are providing involves implementation of the following modules:

- Automated webcam contrivance
- Keystroke logging
- Activity logging
- Automated cloud transfer
- Enabler/disabler keys
- Background processing

Chapter No. 2

Literature survey

LITERATURE SURVEY

Intrusion detection is just one facet of a larger discipline known as endpoint security, which includes everything from malware protection to policy enforcement and asset tracking. Large enterprise computing environments demand comprehensive endpoint-security systems, consisting of server software coupled with client software on each user's machine; that can handle many of these functions at once. These systems tend to be complex enough to require the expertise of a trained IT professional. But in the solution we recommend, we will be looking primarily at simpler tools designed for individuals and smaller organizations.

For an individual or a small business, there are several good ways to achieve endpoint security. One can install a Web-hosted system that combines software on the PC with remote monitoring services to protect computers and enforce compliance with company policies. A few complementary tools, such as a desktop security suites and professional tracking software can be combined.

2.1 Existing systems

- Hardware keyloggers
- Software keyloggers
- Screen recording software

2.2 Hardware keyloggers

A hardware keylogger is mainly a small electronic device used for capturing the data transmitted between a keyboard device and an I/O port [1]. When they are mounted in a computer system they start capturing the keystrokes in their inbuilt memory. At present, a myriad of hardware keyloggers are available in market. These keyloggers can be plugged into the keyboard port, or directly inside the keyboard or at the end of the keyboard cable. The main advantage of hardware keylogger is that it does not use any computer resource so it becomes quite infeasible for the anti-viral software or scanners to detect. The keystroke logs are stored in encrypted form in its own memory instead of the computer's hard disk. The major disadvantage of hardware keylogger is that they necessitate physical installation in the keyboard or computer case.

2.3 Software keyloggers

Software keyloggers logs and monitors the keystrokes and data within the target operating system, store them on hard disk or in remote locations, and send them to the attacker. Software keylogger [2], [3] monitoring is mainly based inside the operating system. The major problem of data theft due to use of key loggers were minimized by the use of various anti-key logging mechanisms. Virtual keyboard is the most popular countermeasure of software based keylogging. Since virtual keyboard only operates through mouse clicks so the key strokes are not captured [4]. The virtual keyboard uses the concept of random shuffling of keys; hence it does not have a definite structure. Therefore the keystrokes, if captured cannot be used because of the random placement of key locations.

2.4 Screen recording software

Screen recording software programs are prevalent because they are used to capture whatever is happening on the screen for monitoring [5] purposes or for the purpose of educational demonstrations. This could be used to conjure a negative effect because the software could be used to capture the screen and mouse movements, so whosoever is using the virtual keyboard to avoid keyloggers are no longer safe. This software records the screen activities which includes key presses through virtual keyboards. Whatever activities are done on the screen are recorded and hence the passwords can be easily picked off. Several models have been proposed to deceive keyloggers and screen recording software programs, and thereby bypassing the malicious techniques and help the access control techniques to work properly.

Chapter No. 3

Feasibility study

FEASIBILITY STUDY

3.1 Technical feasibility

The necessary information for procession to come to firm conclusions regarding the project's viability is provided in the sections that follow.

3.1.1 Prototype modules

The prototype modules include:

- Automated webcam contrivance
- Keystroke logging
- Activity logging
- Automated cloud transfer
- Enabler and disabler keys
- Background processing

3.1.2 Design

The design of the final software is intended to be of a minimalistic demeanor. The user should feel at ease when using the software. The software will run in the background and have no front-end except for the key programs to enable/disable the software.

3.1.3 Target niche

The group of people for whom the software is intended to be developed for include small businesses, firms and individuals.

3.1.4 Integration

The modules will be created individually and then consolidated to run as a single program. There will be distinct key programs to enable and disable the software.

3.2 Economic feasibility

The analysis of the project's costs and revenues in an effort to determine whether or not it is logical and possible to complete has been carried out. A modern computer, being the most important prerequisite will be used to run the software. The system is quite affordable and if commercialized in the current economic scenario, will be priced for a value less than ₹ 2,000 for individual users and around ₹ 3,500 for an enterprise.

3.3 Operational feasibility

As mentioned before, the software will have two distinct key programs to enable and disable the Intruder Identification System. And in addition to this, the way the software will operate will be unbeknownst to the intruder or any other users, except the owner of the machine.

3.4 Legal feasibility

The user of the software can use this Intruder Identification System on the computer that they personally own. It is advised that this software is not to be used on computers not belonging to the person who buys this software. If someone does so, this won't honor the ethical use of the Intruder Identification System and be inadvertently fall under malicious activity.

Chapter No. 4

Requirements

REQUIREMENTS

4.1 Hardware requirements

Component	Minimum requirements
Processor	1 gigahertz (GHz) or faster, 86-bit or 64-bit processor
Memory (RAM)	1 gigabyte (GB) RAM (32-bit); 2 gigabytes (GB) RAM (64-bit)
Hard disk	200 Megabytes (MB) available
Display	1024 x 768 or higher resolution monitor

4.2 Software requirements

Component	Version
Operating system	Windows 7 SP1 or later
Java Runtime	Java SE 6 Update 17
Command Prompt	Microsoft Windows [Version 6.1.7600]
A cloud-storage account and synchronization manager	Google Drive, Dropbox, MediaFire, or Microsoft OneDrive

Chapter No. 5

Implementation

IMPLEMENTATION

5.1 Automated webcam contrivance

Its basic function is to capture a picture of the intruder via webcam, using a scheduling system. The webcam will click pictures of the intruder at a default time interval of 10 seconds and save those pictures at a location specified by the user. These files are later sent to the owner's cloud based storage.

5.2 Keystroke logging

The keylogger saves each key pressed and every character that is typed, and makes periodic captures of the computer screen at a default time interval of 10 seconds to create a log file which can be later used to better illustrate what was going on at that moment.

5.3 Activity logging

The activity logger keeps a list of all activities performed on the computer. It records all visited Web sites, keeps logs of chats and instant messenger conversations as well as other things typed or received by the intruder in chats, messengers, blogs, forums and other online and offline applications.

5.4 Automated cloud transfer

The log files, pictures and the collected data on the intruder is stored on the owner's computer locally. These files are then uploaded to the user's cloud-based storage. These files can be accessed both remotely and locally.

5.5 Enabler and disabler keys

The owner can disable the Intruder identification system just by opening a key program, and may re-enable it by opening a separate key program. This may come in handy when instead of an intruder, the owner logs into the system.

5.6 Background processing

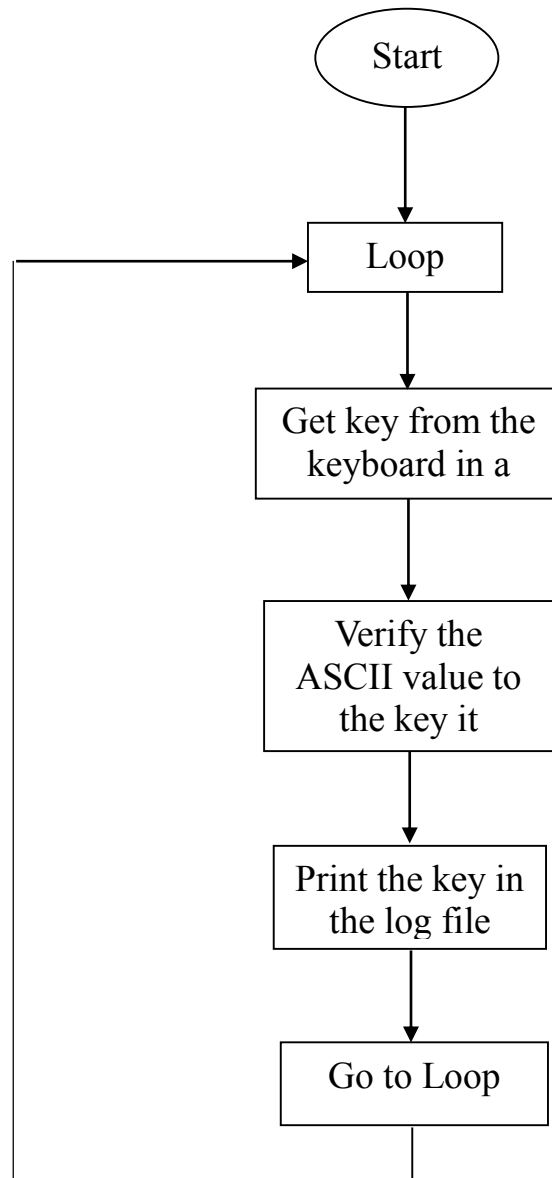
For even better protection, the intruder identification system software-set is completely hidden from the intruder. It runs silently and unobtrusively on the owner's PC while taking screen shots and recording every key that they press, logging applications that they use and Web sites that they visit. These files and logs can be accessed either locally or remotely at any time.

Chapter No. 6

Flowcharts

FLOWCHARTS

6.1 Keylogger



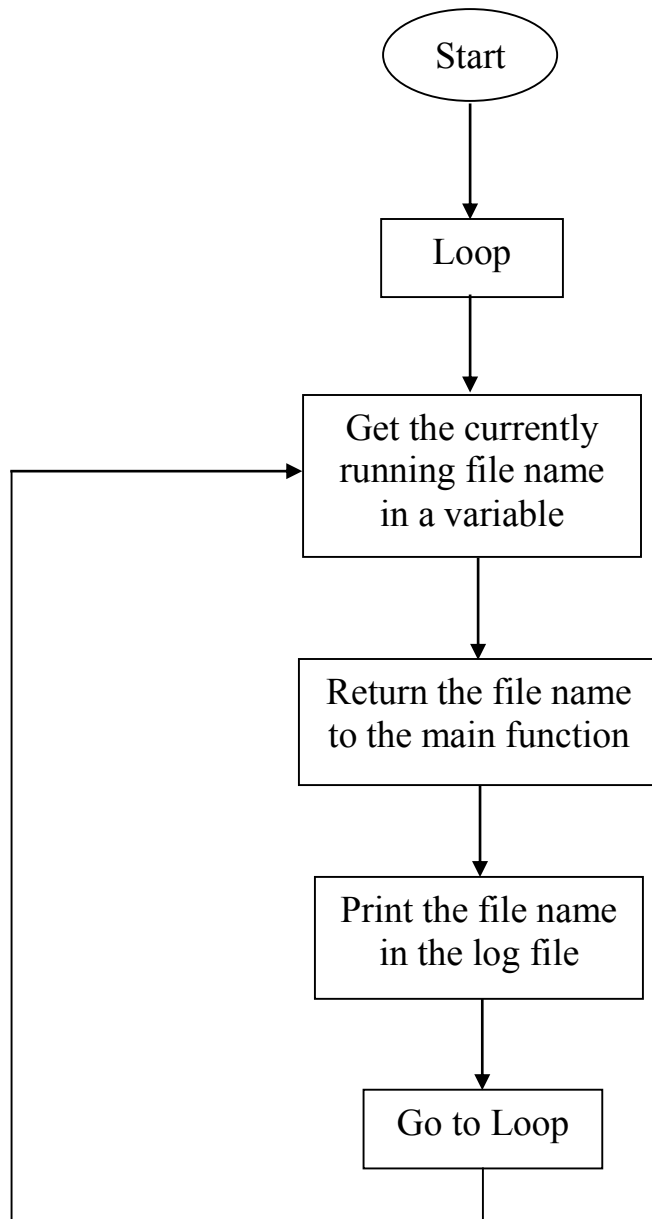
Explanation:

The complete keylogging process executes in an infinite loop, because the logs need to be created right from the time the system is turned on. When the intruder presses a key or multiple keys on the keyboard, it is stored as a variable. Out of the 104 keys in a computer, the intruder can press any key.

Each key has an ASCII value by which the keys are identified by the system. The ASCII value of the key pressed needs to be verified to the key it belongs to. The keys that are pressed by the intruder are then stored in a log file. This log file is created to keep a record of the keystrokes struck by the intruder.

The program also records the movements of the cursor. If the cursor is moved or the buttons of the mouse are clicked the logger will log the mouse movements and/or button-presses and store them in the log file.

6.2 Activity logger



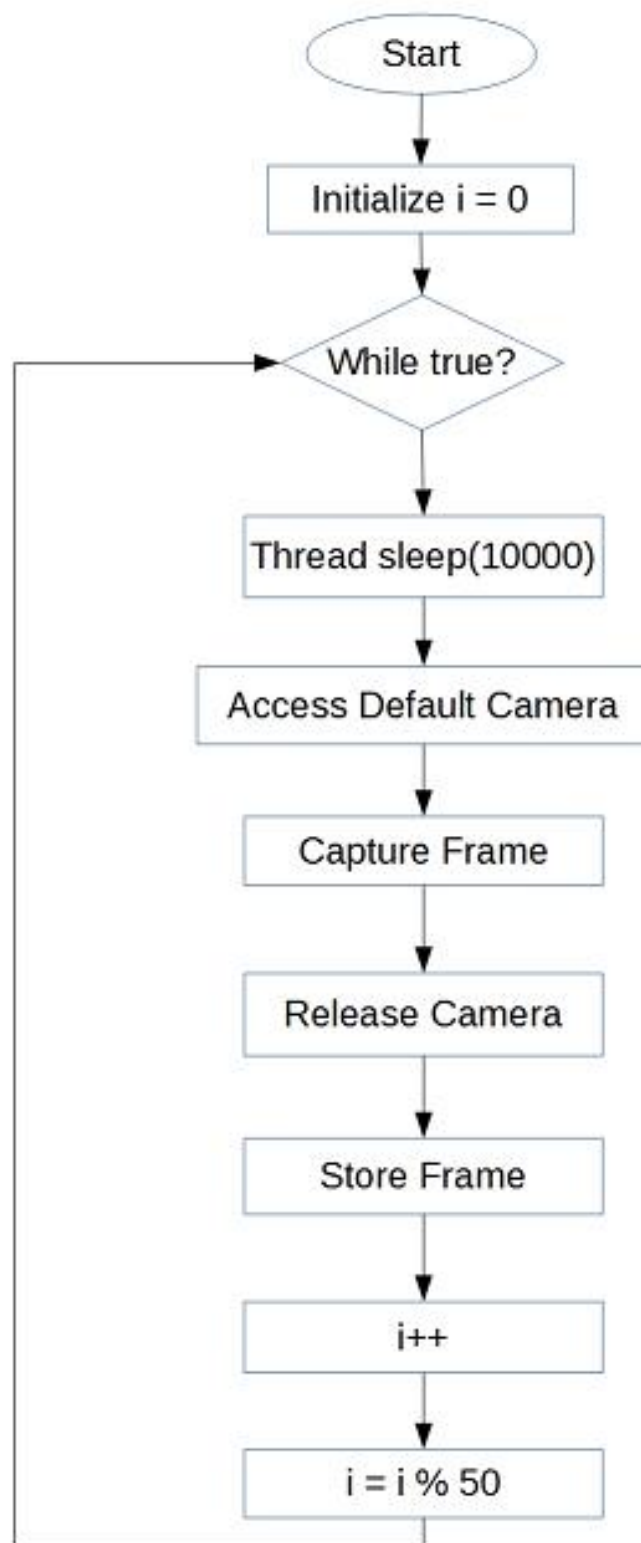
Explanation:

The activity logger runs in an infinite loop as well, because activity logs need to be created as soon as the time the system is turned on. Activity logs are made when the intruder executes or accesses a file.

In the process of logging activities, when the intruder accesses a file, the activity logger function is triggered automatically from the main function of the program. The name of the file and the document type of the file that the intruder is accessing is taken in a variable.

The file name and the document type is returned to the main function. The file name is then printed in the log file. The file name that the intruder is accessing is printed in the same log file that is used to store the keystroke logs. This is because every 10 seconds, the keystrokes and the activities are logged together. This process runs in an infinite loop, the activity logs are created simultaneously alongside the keystroke logs.

6.3 Automated webcam contrivance



Explanation:

This flowchart shows the working of web cam module. It is important that the program runs continuously.

Initialize i:

The variable “i” is used for limiting the photos captured by the web cam, to reduce the amount of space consumed.

While:

For program to continuously run in the background, this loop is set to “True”.

Thread sleep():

This gives time for the camera to initialize and calculate light to capture a frame. This also specifies the time interval required for program.

Access default camera:

Using OpenCV API program takes control of the default camera. The default camera may be a built-in web cam in notebooks or an external web cam for desktops.

Capture frame:

Photo is captured and filtration is done in the background by OpenCV API.

Release Camera:

Camera is released for the specified time interval.

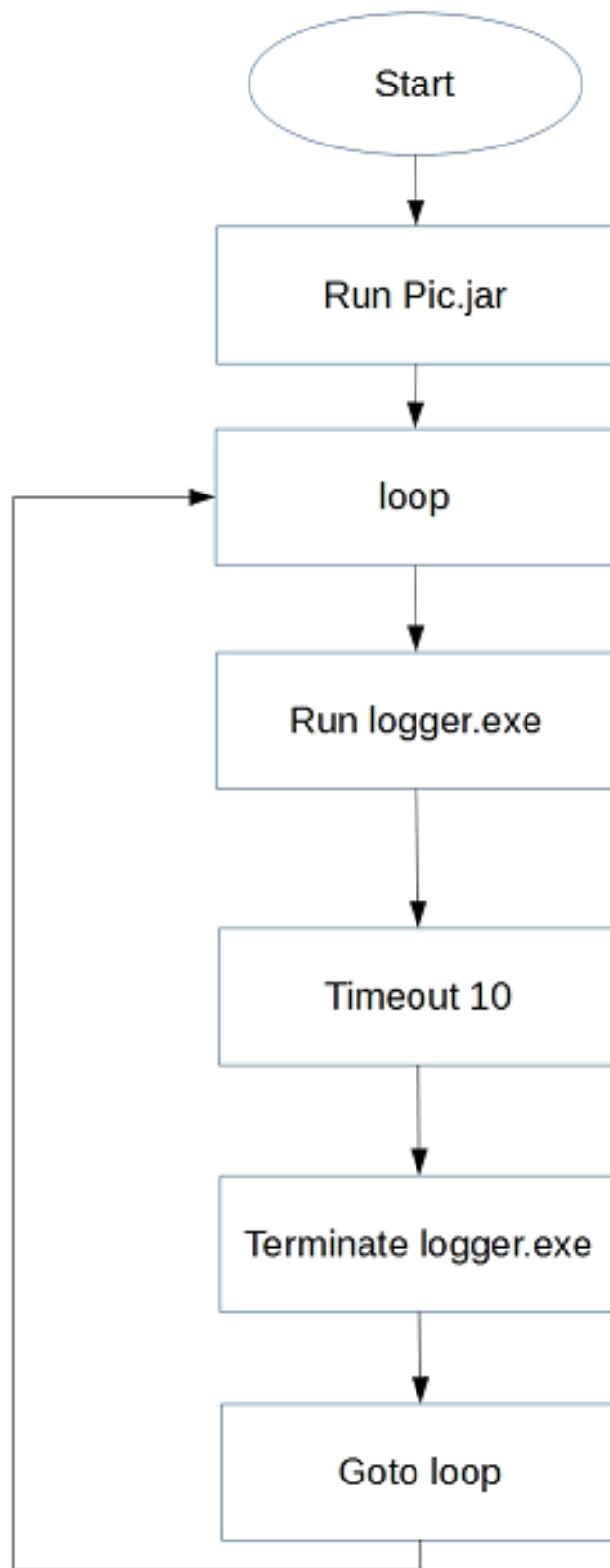
Store Frame:

Captured image is named as specified and stored in the secondary drive.

Buffer:

A limited buffer is maintained at the final stage of program. The buffer is necessary for two reasons; first is the uploading of files to cloud storage and second is for conserving disk space.

6.4 Enabler key program



Explanation:

This flowchart shows the initialization of our main software-set. This script starts each module and loops itself to monitor those modules. This is a non-terminating script which runs till the computer is shut down or safely disabled using the disabler key program.

Run pic.jar:

This runs web cam module. This initializes the web cam for photo-capturing process.

Loop:

This is the looping point for the program.

Run logger.exe:

This starts activity logger and key logger. For synchronization purposes the activity logger and key logger are combined into a standard executable called logger.exe.

Timeout 10:

This gives time for logger.exe to initialize and work.

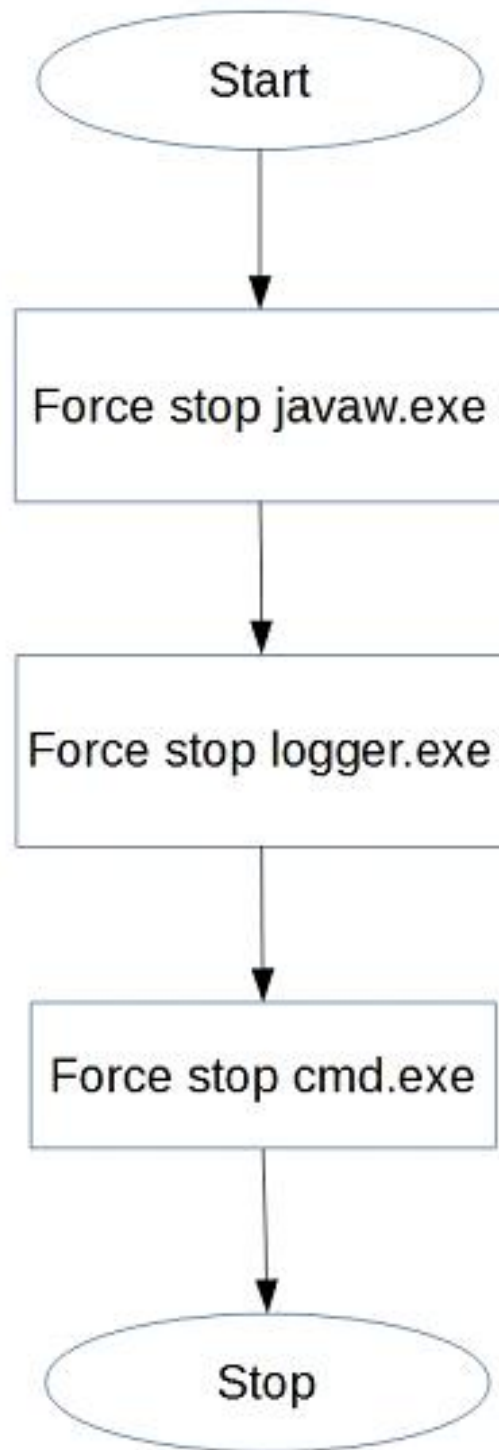
Terminate logger.exe:

This terminates working of logger.exe to generate log associated with activity logger and key logger.

Goto loop:

This makes the script run continuously by looping unconditionally.

6.5 Disabler key program



Explanation:

This flowchart shows the termination of the entire software-set. Each component of the program cannot be stopped manually by the owner as they run in the background. So a batch script is used for stopping all running components safely. This script can be run from a USB flash drive.

Force stop javaw.exe:

The web cam module is developed using Java programming language. This program runs on Java virtual machine. To stop the web cam module, the Java virtual machine needs to be terminated.

Force stop logger.exe:

The activity logger and key logger are combined into a single program called logger.exe to stop any balks during cloud-synchronization. Logger.exe is a standard windows executable file which is directly stopped using this script.

Force stop cmd.exe:

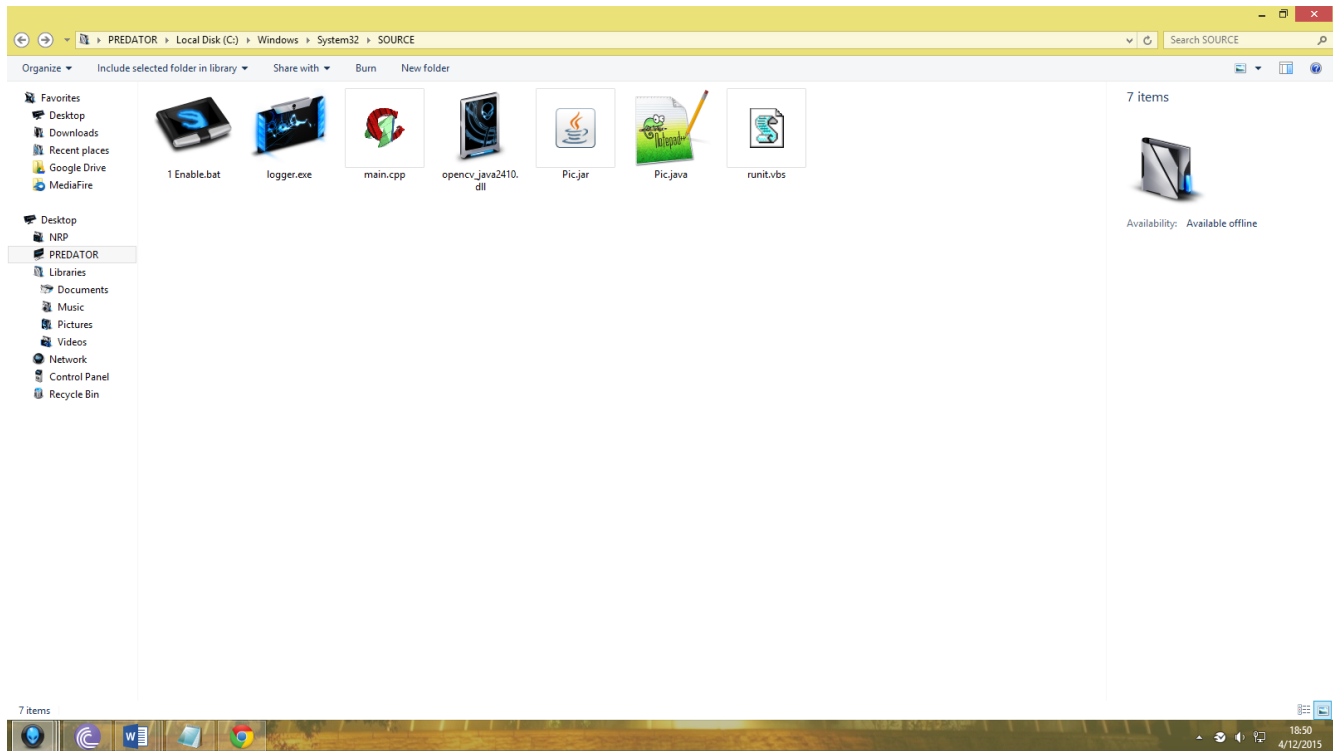
Multiple batch scripts are run in the background to monitor the working of each module. This script runs on Windows Command Prompt. To safely terminate this program the monitoring of scripts needs to be put at an end. Windows cannot distinguish between different batch scripts as it compiles all scripts using Command Prompt. So command prompt is terminated by stopping cmd.exe.

Chapter No. 7

Screenshots

SCREENSHOTS

7.1 Program location



In this screenshot we can see, main files that constitute the software-set of the Intruder Identification System. These files will be enabled by the enabler key program. The files are:

- runit.vbs

It is enabled via Windows Startup. Its main function is to hide and run its sibling programs in the background.

- 1 enable.bat

This file enables the files “logger.exe” and “Pic.jar”. It is enabled by “runit.vbs”.

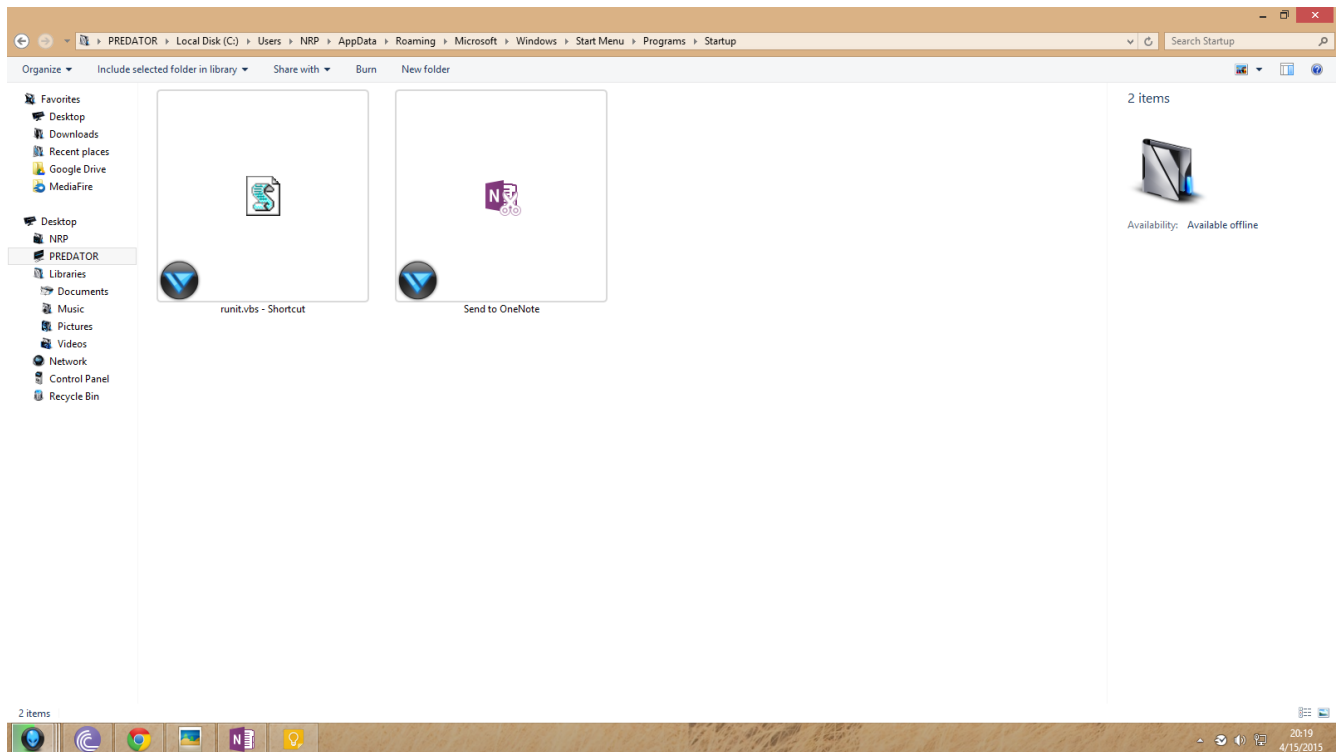
- logger.exe

This file logs the activity and keystrokes.

- Pic.jar

This file automates photo-capture via webcam.

7.2 Program enabler in startup



The file “runit.vbs - Shortcut” is kept in the Windows Startup folder. When the computer is turned on and is logged into, the VBScript is run automatically. It enables the Intruder identification system’s files shown in the previous screenshot.

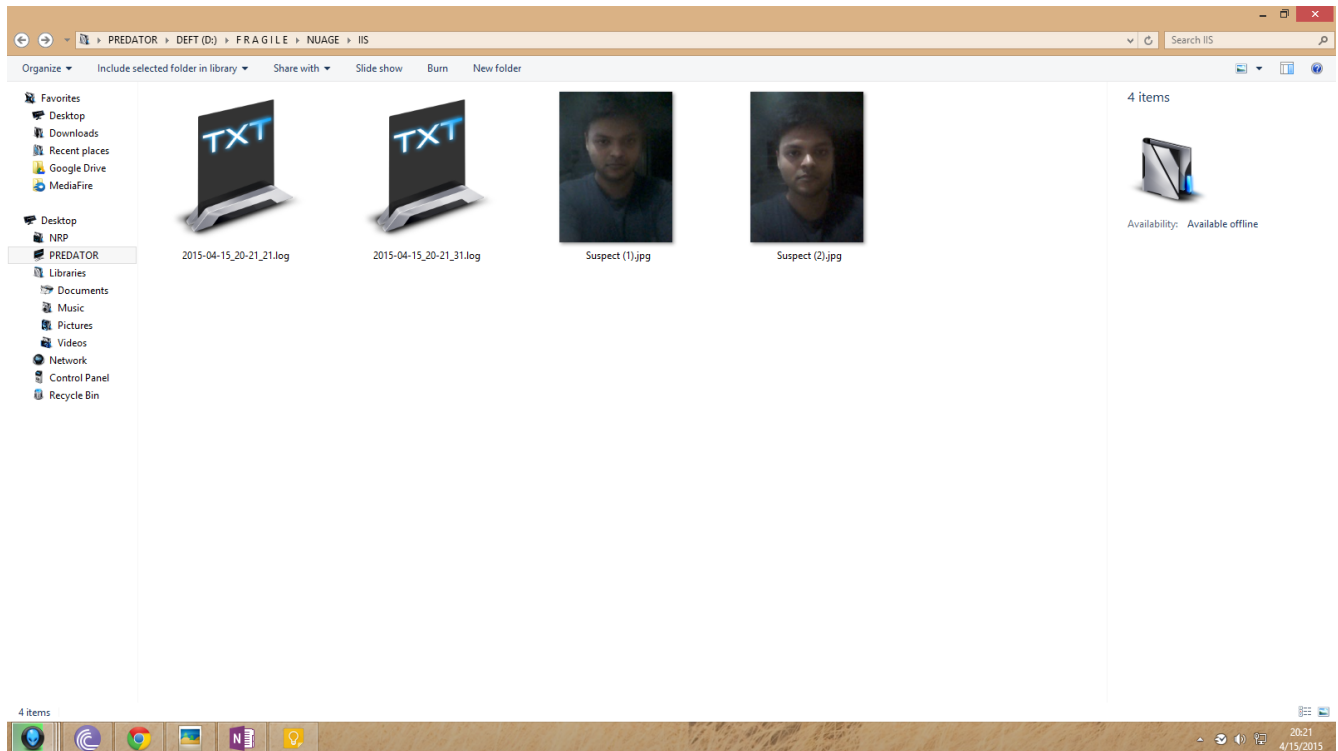
The VBScript is scripted with the location of the enabler key program and works by following the written script.

The VBScript runs “1 Enable.bat” which in turn executes its sibling programs. In addition to initiating the execution of the enabler key program, runit.vbs is also responsible for making the specified batch file run in stealth mode. The enabler key program, in turn, runs the files “logger.exe” and “Pic.jar” which are responsible to run the activity/key logger and the automated webcam contrivance respectively.

This way, we are able to keep the execution of the entire software-set away from the intruder’s attention.

7.3 Log files in cloud storage

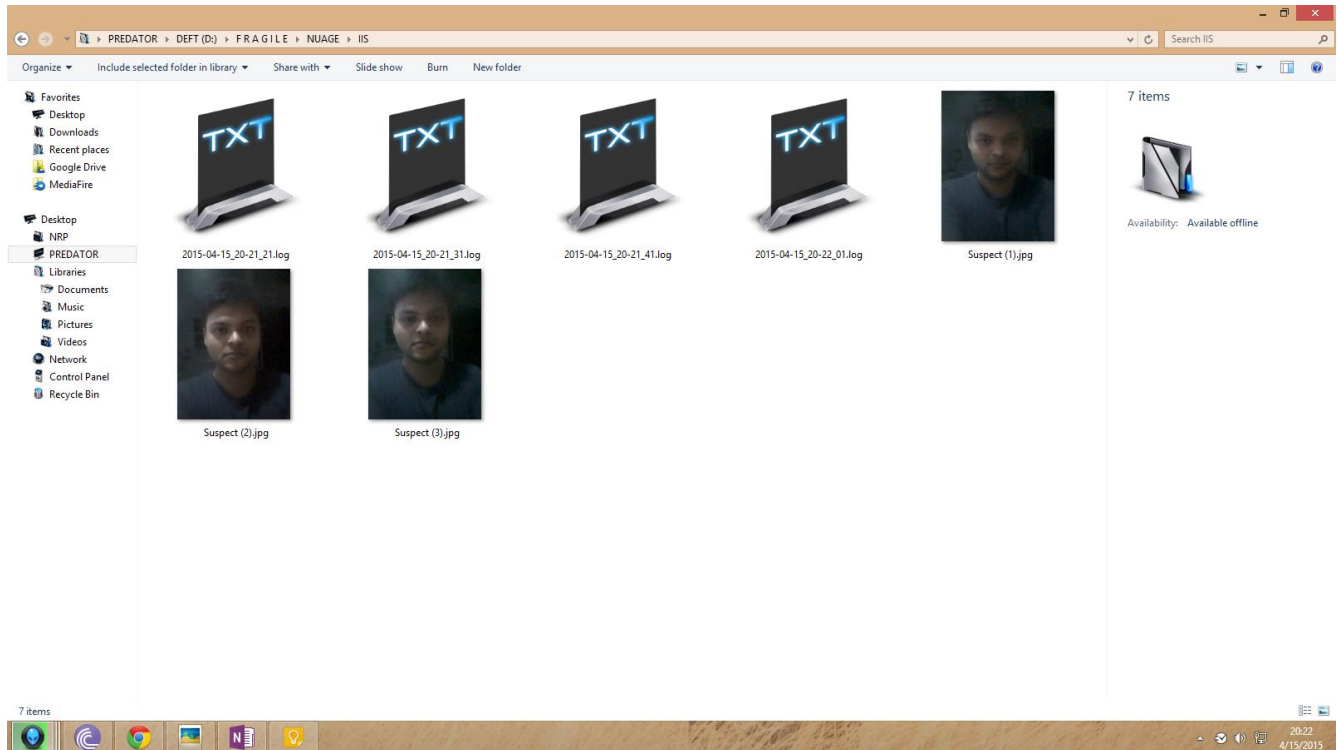
7.3.1



In the above screenshot, we can see that the intruder was caught on camera and file logs were created. The first two files shown in the screenshot are the log files. They contain the intruder's activity as well as the keystroke-log. The first file is named "2015-04-15_20-21_21.log". This means that this log file was created on 15th of April, 2015 at 20:21:21, i.e. at 21 seconds after 21 past 8:00 p.m. Similarly, the second log file is named "2015-04-15_20-21_31.log". This means that this log file was created on 15th of April, 2015 at 20:21:31, i.e. at 21 seconds after 31 past 8:00 p.m. This file is named so, because logger.exe is programmed to create log files at intervals of 10 seconds; so the first file was generated at 20:21:21 and the next one 10 seconds later, at 20:21:31.

In addition to the log files, two .jpg files can be seen. The intruder was photographed because Pic.jar, i.e. the automated webcam contrivance was also being run. The files are named serially with the common name of "Suspect". The first image is named "Suspect (1).jpg" and the second "Suspect (1).jpg".

7.3.2

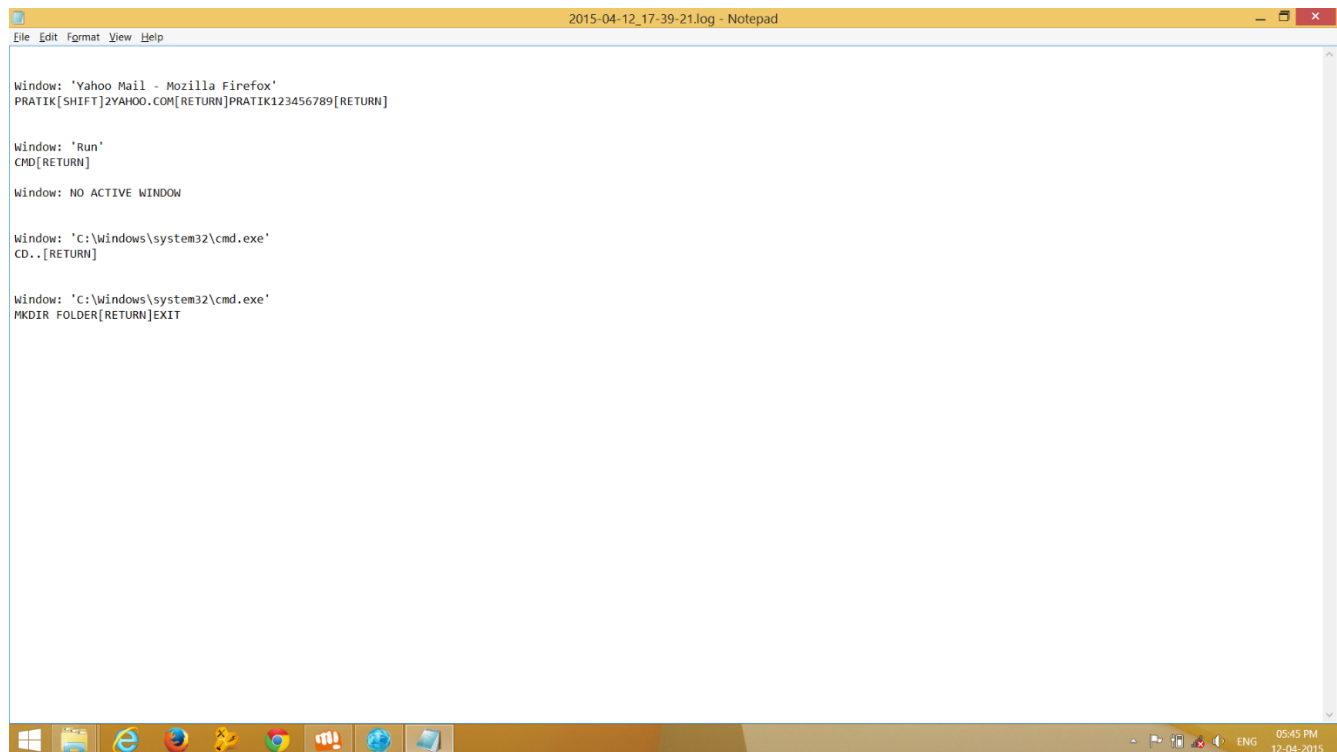


In this screenshot, it can be seen that more log files of the intruder’s activity and photographs of the intruder were sent to the owner’s cloud storage. These files can be accessed by the owner by logging into the cloud-storage account. Its major advantage is, the owner can access the data on the intruder, which is directly sent to the said owner’s cloud, using any device logged into the cloud account.

It can be observed that the Windows timestamp marks the time at 20:22. In accord to this, logger.exe created appropriately timed log files. The third log file is named “2015-04-15_20-21_41.log” and the fourth is named “2015-04-15_20-22_01.log”. There is no file named “2015-04-15_20-21_51.log” because there was no activity during the time duration 20:21:42 to 20:21:51. Since there was no user activity, no log file was created.

In addition to this, another photograph of the intruder was captured by the automated webcam contrivance. This file was named “Suspect (3).jpg”. These files were synchronized with the user’s cloud-based storage simultaneously as they were created.

7.4 Log file showing logged activities and keystrokes.



```
2015-04-12_17-39-21.log - Notepad
File Edit Format View Help

Window: 'Yahoo Mail - Mozilla Firefox'
PRATIK[SHIFT]2YAHOO.COM[RETURN]PRATIK123456789[RETURN]

Window: 'Run'
CMD[RETURN]

Window: NO ACTIVE WINDOW

Window: 'C:\Windows\system32\cmd.exe'
CD..[RETURN]

Window: 'C:\Windows\system32\cmd.exe'
MKDIR FOLDER[RETURN]EXIT
```

In this screenshot, we can see that the intruder opened a new tab titled “Yahool Mail” in the browser “Mozilla Firefox”.

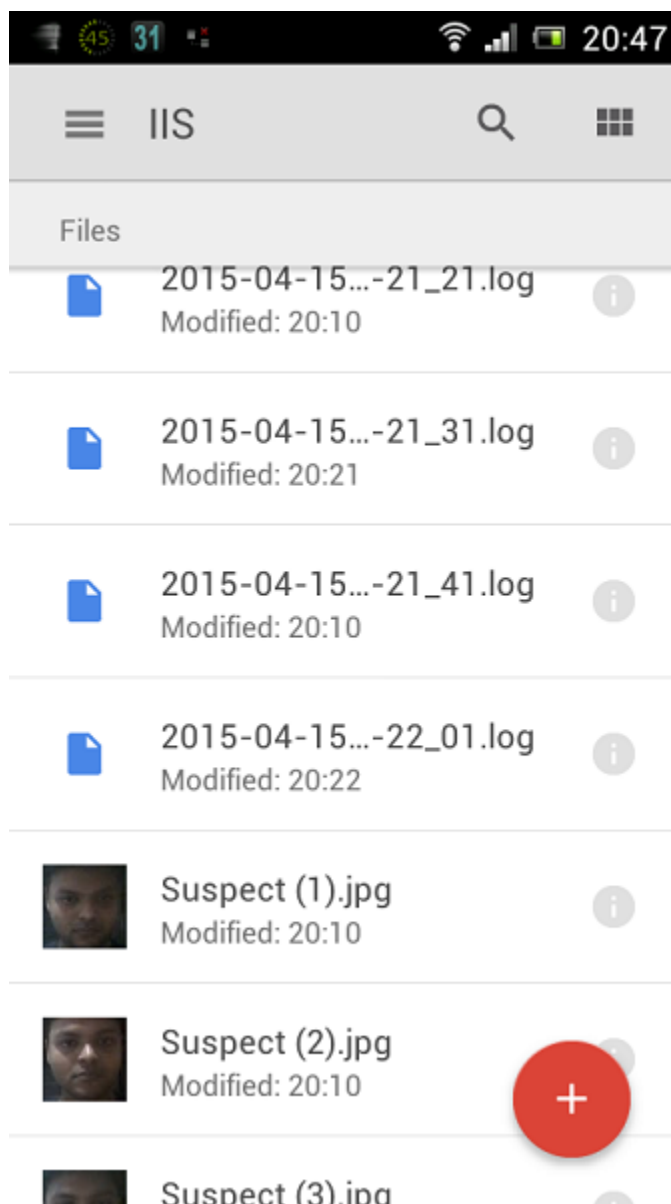
Then the intruder proceeds to type the word “PRATIK”, presses the “Shift key” and then types “2yahoo.com”. This means that the intruder typed PRATIK@YAHOO.COM and then pressed the “Return key”. Then he proceeds to type “PRATIK123456789” and presses the “Return key” again.

Then he uses the Windows’ “Run” utility to run cmd, i.e. the Windows Command Prompt.

It can be seen that cmd.exe is running and its location is also visible.

Then the intruder types “CD..” and presses the “Return key”, proceeds to type in “MKDIR FOLDER” and again presses the “Return key” and exits.

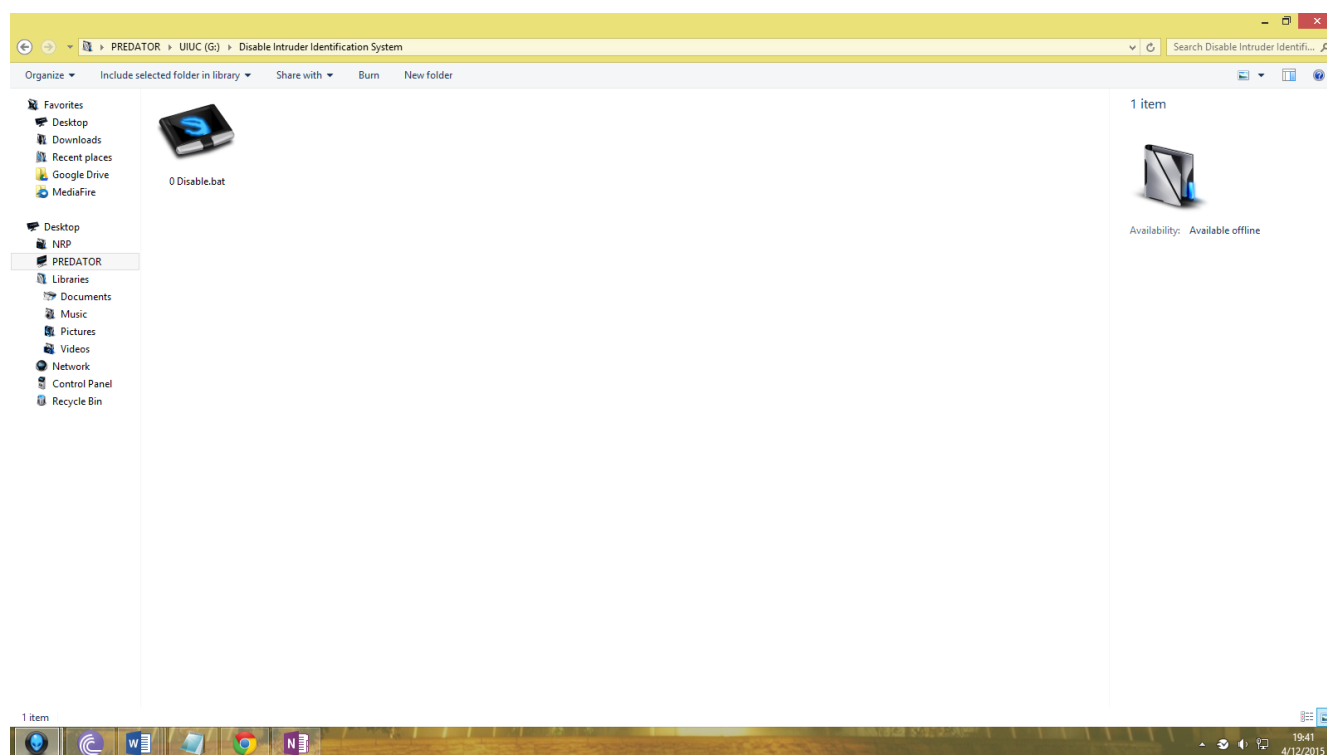
7.5 Logs and intruder's photographs accessed on the owner's cellphone via cloud storage.



In the above screenshot, we see that the owner of the infiltrated system is accessing the cloud-based storage. Here all log files created by logger.exe and photographs of the intruder created by Pic.jar, i.e. the automated webcam contrivance, have been synced successfully to the owner's cloud.

The owner has been notified via the cloud and is now viewing the data collected on the intruder. Here, the owner is viewing these files on a smartphone. Most cloud-storage services release a smartphone-optimized client application for users since in the past few years, the usage of cloud-storage has been consistently prevalent.

7.6 Disabler key program to disable Intruder Identification System



In the above screenshot, we can see the file “0 Disable .bat”. When the real owner of the computer logs into the system and may want to disable the Intruder Identification System, this file is used to disable the programs.

The disabler key program is placed in an external thumb-drive to keep it away from the intruder’s grasp. The disabler specifically terminates the files whose executions were initiated en masse by “1 Enable.bat”. The files disabled by “0Disable.bat” are:

- logger.exe
- javaw.exe
- cmd.exe

Once these files stop running, the Intruder Identification is put to a stop. This means, that all pertinent modules are terminated. Since no log files or pictures are being generated, these files are no longer synchronized to the owner’s cloud-based storage.

Chapter No. 8

Program codes and features

PROGRAM CODES AND FEATURES

8.1 Automated webcam contrivance

```
import org.opencv.core.Core;
import org.opencv.core.Mat;
import org.opencv.highgui.Highgui;
import org.opencv.highgui.VideoCapture;

public class Pic
{
    public static void main (String args[])
    {
        System.loadLibrary(Core.NATIVE_LIBRARY_NAME);
        Mat frame = new Mat();
        VideoCapture camera = new VideoCapture(0);
        int i=0;
        //i=Integer.parseInt(args[0]);
        while(true)
        {
            try {
                Thread.sleep(10000);           //1000 milliseconds is one second.
            }
            catch(InterruptedException ex)
            {
                Thread.currentThread().interrupt();
            }
        }
    }
}
```

```
camera.open(0);
camera.read(frame);
camera.release();
Highgui.imwrite("C:\\Users\\User_4A\\SkyDrive\\Pic_log\\Pic"+i+".jpg", frame);
i++;
i=i%50;
}

}

}
```

8.2 Source code for Activity logger and keylogger

```
#define FILEEXT ".log"

#include <cstdlib>
#include <iostream>
#include <sstream>
#include <fstream>
#include <windows.h>
#include <ctime>
#include <cstring>
using namespace std;

LRESULT CALLBACK WndProc(HWND hwnd, UINT Message, WPARAM wParam,
LPARAM lParam)
{
    switch(Message)
    {

        case WM_DESTROY:
        {
            PostQuitMessage(0);
            break;
        }

        default:
            return DefWindowProc(hwnd, Message, wParam, lParam);
    }
    return 0;
}
```

```

}

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR
lpCmdLine, int nCmdShow)
{
WNDCLASSEX wc;
HWND hwnd;
MSG msg;

memset(&wc,0,sizeof(wc));
wc.cbSize = sizeof(WNDCLASSEX);
wc.lpfnWndProc = WndProc;
wc.hInstance = hInstance;
wc.hCursor = LoadCursor(NULL, IDC_ARROW);

wc.hbrBackground = (HBRUSH)(COLOR_WINDOW+1);
wc.lpszClassName = "WindowClass";
wc.hIcon = LoadIcon(NULL, IDI_APPLICATION);
wc.hIconSm = LoadIcon(NULL, IDI_APPLICATION);

if(!RegisterClassEx(&wc))
{
MessageBox(NULL,"Window Registration Failed!", Error!",
MB_ICONEXCLAMATION|MB_OK);

return 0;
}

```

```

//start
char filepath[]="C:\\Users\\User_4A\\SkyDrive\\";
time_t rawtime;
struct tm *timeinfo;
time(&rawtime);
timeinfo = localtime(&rawtime);
char filename[MAX_PATH];
strftime(filename, 100, "%Y-%m-%d_%H-%M-%S", timeinfo);
//sprintf(filepath, "%s\\%s%s", basepath.c_str(), filename, FILEEXT);

//cout << filepath << endl; exit(0);
strcat(filename, FILEEXT);
strcat(filepath, filename);

string lastTitle = "";
ofstream klogout(filepath);

SHORT lastc = 0;
while(1)
{
Sleep(2);
// get the active windowtitle
char title[1024];
HWND hwndHandle = GetForegroundWindow();
GetWindowText(hwndHandle, title, 1023);
if(lastTitle != title){
klogout << endl << endl << "Window: ";
if(strlen(title) == 0)

```

```

klogout << "NO ACTIVE WINDOW";
else
klogout << "" << title << "";
klogout << endl;
lastTitle = title;
}

```

```

// keylogger
for(unsigned char c = 1; c < 255; c++){
SHORT rv = GetAsyncKeyState(c);
if(rv & 1){ // on press button down
string out = "";
if(c == 1)
out = "[LMOUSE]"; // mouse left
else if(c == 2)
out = "[RMOUSE]"; // mouse right
else if(c == 4)
out = "[MMOUSE]"; // mouse middle
else if(c == 13)
out = "[RETURN]";
else if(c == 16 || c == 17 || c == 18)
out = "";
else if(c == 160 || c == 161) // lastc == 16
out = "[SHIFT]";
else if(c == 162 || c == 163) // lastc == 17
out = "[STRG]";
else if(c == 164) // lastc == 18
out = "[ALT]";

```

```
else if(c == 165)
out = "[ALT GR]";
else if(c == 8)
out = "[BACKSPACE]";
else if(c == 9)
out = "[TAB]";
else if(c == 27)
out = "[ESC]";
else if(c == 33)
out = "[PAGE UP]";
else if(c == 34)
out = "[PAGE DOWN]";
else if(c == 35)
out = "[HOME]";
else if(c == 36)
out = "[POS1]";
else if(c == 37)
out = "[ARROW LEFT]";
else if(c == 38)
out = "[ARROW UP]";
else if(c == 39)
out = "[ARROW RIGHT]";
else if(c == 40)
out = "[ARROW DOWN]";
else if(c == 45)
out = "[INS]";
else if(c == 46)
out = "[DEL]";
```



```

else if(c >= 65 && c <= 90 || c >= 48 && c <= 57 || c == 32)
out = c;

else if(c == 91 || c == 92)
out = "[WIN]";
else if(c >= 96 && c <= 105)
out = "[NUM " + intToString(c - 96) + "]";
else if(c == 106)
out = "[NUM /]";
else if(c == 107)
out = "[NUM +]";
else if(c == 109)
out = "[NUM -]";
else if(c == 109)
out = "[NUM ,]";
else if(c >= 112 && c <= 123)
out = "[F" + intToString(c - 111) + "]";
else if(c == 144)
out = "[NUM]";
else if(c == 192)
out = "[OE]";
else if(c == 222)
out = "[AE]";
else if(c == 186)
out = "[UE]";
else if(c == 186)
out = "+";
else if(c == 188)

```

```

out = ",";
else if(c == 189)
out = "-";
else if(c == 190)
out = ".";
else if(c == 191)
out = "#";
else if(c == 226)
out = "<";

else
out = "[KEY \\" + intToString(c) + "]";
klogout << out;
klogout.flush();
lastc = c;
}
}
}

klogout.close();
while(GetMessage(&msg, NULL, 0, 0) > 0) { /* If no error is received... */
TranslateMessage(&msg); /* Translate key codes to chars if present */
DispatchMessage(&msg); /* Send it to WndProc */
}
return msg.wParam;
}

```

8.3 Source code for Enabler key program

```
@echo off
```

```
cd C:\Windows\System32\SOURCE\
```

```
start pic.jar
```

```
:loop
```

```
start logger.exe
```

```
@echo off
```

```
timeout /t 10 > NUL
```

```
@echo off
```

```
taskkill /F /IM logger.exe
```

```
goto loop
```

8.4 Source code for Disabler key program

```
taskkill /F /IM logger.exe
```

```
taskkill /F /IM javaw.exe
```

```
taskkill /F /IM cmd.exe
```

8.5 Source code for stealth VBScript

```
CreateObject("WScript.Shell").RUN "1 Enable.bat", 0, True
```

Chapter No. 9
Conclusion and
Future scope

CONCLUSION AND FUTURE SCOPE

The issue of intrusion and breaking into computer systems are addressed and Intruder Identification System was created. This system not only provides methods for detection of an intrusion but also identifies the intruder and invariably photographs the intruder while logging his activity and keystrokes on the owner's computer. All log files and photographs are sent to the owner's cloud-based storage. This data can then be accessed remotely on other devices as well. The software-set developed will help in identifying any act of intrusion along with the intruder.

The software-set constitutes of 5 major files, viz. a logger (activity and keystrokes), webcam automation module, an enabler, a disabler, and a stealth module. This software-set can be used for individual purposes as well as in small enterprises. Security is always enforced in most computer systems, but sometimes systems encounter a security breach and the owners lament that they could not know who the intruder was. But putting the Intruder Identification System into full effect, an intruder can be detected with ease, unbeknownst to the intruder. This increases the reliability of the overall endpoint security of a computer system to a great extent, making it difficult for intruders to walk away without any repercussions for their actions.

As the endpoint security of a system in one platform is bolstered, by implementing Intruder Identification System in Microsoft's Windows operating system, it can further be used in other operating systems like Apple's OS X and other Unix-based operating systems. It can also be optimized in the future for mobile devices like tablets and smartphones.

Furthermore, all modules can be made to encrypt data before transmission to the owner's cloud storage. This would add extra layers of security to the current system. It has the potential to replace conventional technologies like standalone keyloggers and poorly integrated activity monitors used in many small business enterprises. Further research could be carried out to design better ways to integrate the major modules into a single software-based solution. This software-set can then be commercialized and monetary gains be made.

REFERENCES

- [1] S. Seref and C. Gurol, “Keyloggers increasing threats to computer security and privacy”, IEEE Technology and society magazine, 2009, pp.10-17.
- [2] G. Canbek, “Analysis, design and implementation of keyloggers and anti-keyloggers”, Gazi University, Institute Of Science And Technology, M.Sc. thesis (in Turkish), Sept. 2005, pp. 103
- [3] F.S. Lane, “The naked employee: How technology is compromising workplace privacy”, AMACOM Div American Mgmt. Assn., 2003, pp.128-130.
- [4] S. Gong, “Design and Implementation of Anti-Screenshot Virtual Keyboard Applied in Online Banking E-Business and E-Government”, (ICEE), 2010 International Conference, 7-9 May 2010, pp. - 13201322
- [5] L. Valeri, “Screen Recording System for Windows Desktop”, Russian-Korean International Symposium Science and Technology conf., 2004, pp.107-109