SPL-1 Project Report, 2020

eSpy

Mailing Keylogger with Screenshot, Parsing Functionalities and Control Panel

SE 305 : Software Project Lab I

Submitted by

Md Siam

BSSE Roll No. : 1104

Exam Roll No. : 1122

BSSE Session: 2018-19

Supervised by

Dr. B M Mainul Hossain

Designation: Associate Professor

Institute of Information Technology

Institute of Information Technology
University of Dhaka
26-08-2021

Project:	eSpy
Author:	Md Siam
Submitted:	24.08.21
Supervised By:	Dr. B M Mainul Hossain Associate Professor, Institute of Information Technology University of Dhaka

Supervisor's Approval: ______________________________

ACKNOWLEDGEMENT

I would like to express my deep and sincere gratitude to my Software Project Lab -1 Supervisor, Dr. B M Mainul Hossain Sir, Associate Professor, Institute of Information Technology, University of Dhaka, for giving me the opportunity to do this project and providing invaluable guidance throughout the entire course. His dynamism, vision, sincerity and motivation have deeply inspired me. It was a great privilege and honor to work under his guidance.

Table of Contents

1. Introduction

eSpy can seamlessly track the keystrokes of a user and store them in a file. It's truly a SURVEILLANCE since the program runs completely anonymous in the background and the console is hidden. It takes screenshots on a regular basis. The keystrokes of the users are parsed and their browsing history is extracted from there. The recent tabs and windows the user has been through are also tracked. And all that information is passed through email to the admins. Sending mail is done with SMTP protocol. So I had to implement the SMTP protocol for the project too.

There is an extensive Control Panel, that the authorized users can use to have an overall summary and look through the track records. An authorized user can add members too.

2. Objectives

Surveillance is a thing of great concern nowadays. In order to make sure the resources are used in a proper way where they should have been, the organizations should have a surveillance tool. This is at the same time, a matter of security concern for the organizations since data confidentiality is one of the most prioritized things.

The objective of **eSpy** is to make this ethical spying easier and more compact. Giving a summary of the spied data, an overview of the tracks in a more convenient and simpler way is the main goal.

3. Scope

The scope of **eSpy** is as follows-

- ➔ Log the keystrokes of a user, run completely anonymous in the background.
- ➔ Take screenshot of the screen within a regular interval

- ➔ Parse the file and get the compact result of keystrokes (get the browsed websites)
- ➔ Send the details over email on a regular interval
- ➔ Emails can be manually sent via admin panel too
- ➔ Get the windows the user has clicked
- ➔ Authentication and Admin surveillance support (CTRL + SHIFT + TAB to launch the admin panel)
- ➔ View the screenshots, add new user, view the browsing history, list of clicked windows/tabs from admin panel

4. Background Study

4.1. Keyboard Buffer

A keyboard buffer is a section of computer memory used to hold keystrokes before they are processed. The keyboard buffer is LOCAL. Since tracking the keystrokes of a user is a vital part of my project, I had to study the keyboard buffer mechanism, and get a copy of the buffer saved in the storage.

4.2. Socket Programming

Since it's a mailing keylogger with some advanced additional functionalities, mailing the current logs and the visited windows to the admins is a major part of the project. And in order to get hands-on network programming, I had to learn the basic socket, server-client programs, and learn the overall structure of how it works, eg. establishing a connection, sending and receiving data, etc.

4.3. Simple Mail Transfer Protocol

The Simple Mail Transfer Protocol (SMTP) is an internet standard communication protocol for electronic mail transmission. Mail servers and other message transfer agents use SMTP to send and receive mail messages. User-level email clients typically use SMTP only for sending messages to a mail server for relaying, and typically submit outgoing emails to the mail server. I had to learn the SMTP protocol, sending multiple files (multipart content) on it, and learned how to implement it raw with C++.

4.4. SMTP Response Codes

SMTP is an Application Layer Protocol. Since I was implementing the SMTP protocol in raw to send the emails, I had to study the responses received from a server with respect to my requests, just to have a better idea of what is going on and what to expect. And the best way to understand the server response is to understand the response codes. A better idea of the response codes enabled me to figure out where things were going unexpectedly and gave me ideas about how to fix them.

4.5. Screen Capturing Mechanism

My project features a regular interval screenshot service, it takes a snap of the screen on a regular basis. I had to learn the screenshot mechanisms to implement it raw with C++. Getting the current Handle to Device Context and copying and storing it to a BMP file was the task to be done. I had to learn how it works and how to implement it raw.

4.6. Top-Level Domain

A top-level domain (TLD) is one of the domains at the highest level in the hierarchical Domain Name System of the Internet after the root domain. In my project, I parsed the user keystrokes and extracted the browsed websites from them. I distinguished the keystrokes with the top-level domain as a website and thus had to study the Top Level Domain.

4.7. Pattern Matching

I learned **Knuth–Morris–Pratt** (KMP) string matching algorithm, in order to match the top-level domains with a lesser complexity with the stored buffers. Since the buffers can get too long overtime, it was important to get the pattern matching done with lower time complexity, thus I implemented the KMP algorithm.

4.8. Base64 Encoding

While sending attachments to SMTP, it is good to send the data encoded in Base64. Since my program features attachments of log text files and images, I had to learn Base64 encoding and used it while sending the file contents.

5. Challenges

There are a number of challenges I had to face while implementing the software. A lot of terms and the majority of the tasks were completely new to me that led me towards much confusion and complexity in implementation. Some of the challenges I faced during this implementation is enlisted below-

- Working with Raw windows programming for the first time
- Working with Network Programming for the first time
    - Learning the basic structures
    - Understanding the protocols
    - Implementing the protocols in raw code
    - Getting multiple attachments successfully done in SMTP
- Working with multiple source files
- Getting introduced with various new Jargons of windows programming
- Relatively structured and protocol-oriented programming rather than rigorous plain coding
- Management of a large source code
- Finding a spam-friendly mailing server
    - Firstly, I made my program able to send email to Gmail's SMTP, it seems they blocked my ISP and IP within 3 days

(it's because I couldn’t warm up the port and make myself trusted).

- Then I tried to create my own mailing server, things didn’t work out
- Finally came up with an idea of using a spam-friendly (that doesn’t check for spam) temporary mailing server and got the job done.

6. Project Description

6.1. Log Keystrokes

The keystrokes of a user are saved in a log file. eSpy also keeps track of the possible formatting of the keystrokes (ie. capital letters (with SHIFT or CAPS LOCK key being pressed), key pressed while shift button is down, num lock)

6.2. Run Unnoticed

The console is completely hidden with the Stealth function, making it a true surveillance being completely anonymous in the background.

6.3. Capture the Screen

A snap of the screenshot is taken every 15 minutes and saved in the local directory. It's done using completely raw C++ code. The time interval for the screenshot capture can be modified too. Screenshots can be viewed from the control panel by the authorized users.

6.4. Track the Windows/Tabs the User Visits

Each time the user changes the tab or window, it's tracked and logged into a file. The track file of the windows is sent over mail, and can also be viewed from the control panel.

6.5. Parse the Browsed Websites from Logged Keystrokes

This project features a parser that extracts the browsed websites from the logged keystrokes.

6.6. Send Email with Attachments

An email with so far logged keys and attachments of the windows a user has visited and a file containing the list of the websites browsed are sent to a recipient. Currently, it is only limited to send email to the temporary spam-friendly mailing server (guerillamail.com). If email is sent via timeout, it is sent to the default recipient email. An admin can also set the recipient email after authentication (that's also limited to guerillamail's SMTP)

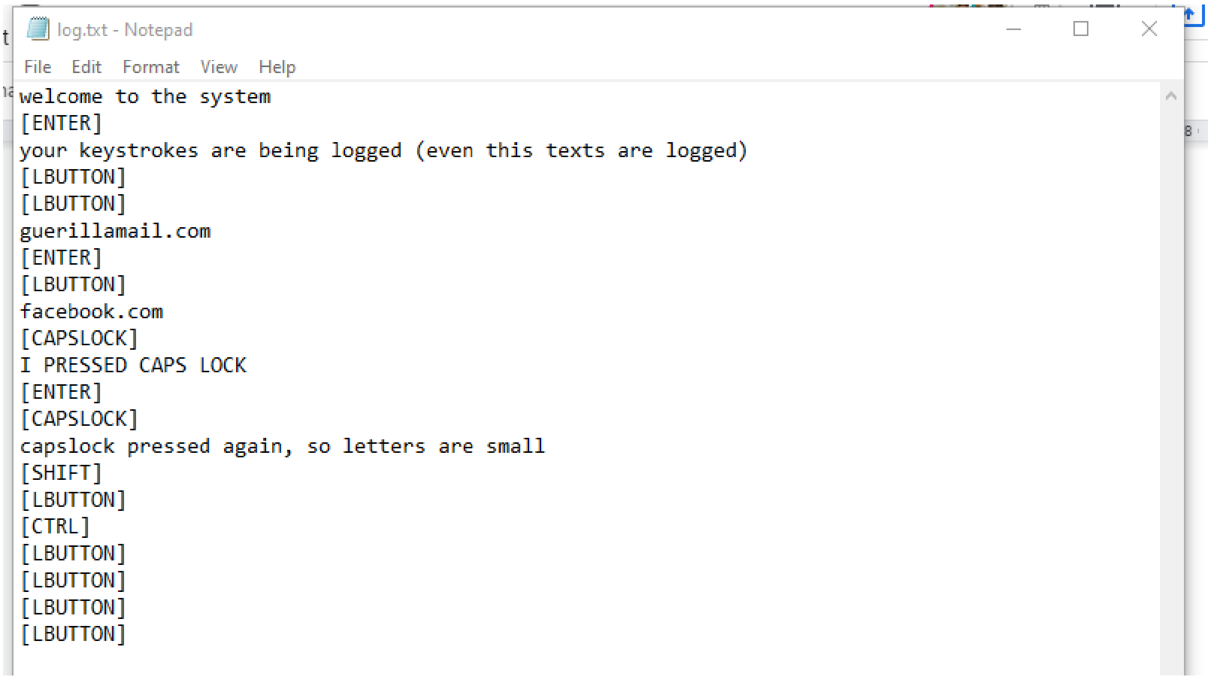
6.7. Control Panel

An user can prompt the control panel by pressing (CTRL+SHIFT+TAB) and authenticate himself with UserID and Password. An authorized user can-

□ View the screenshots taken so far
□ Add new users
□ View the browsing history (that's extracted from the logged keys)
□ Send an email to the server
□ View the windows the user had been through

7. User Manual

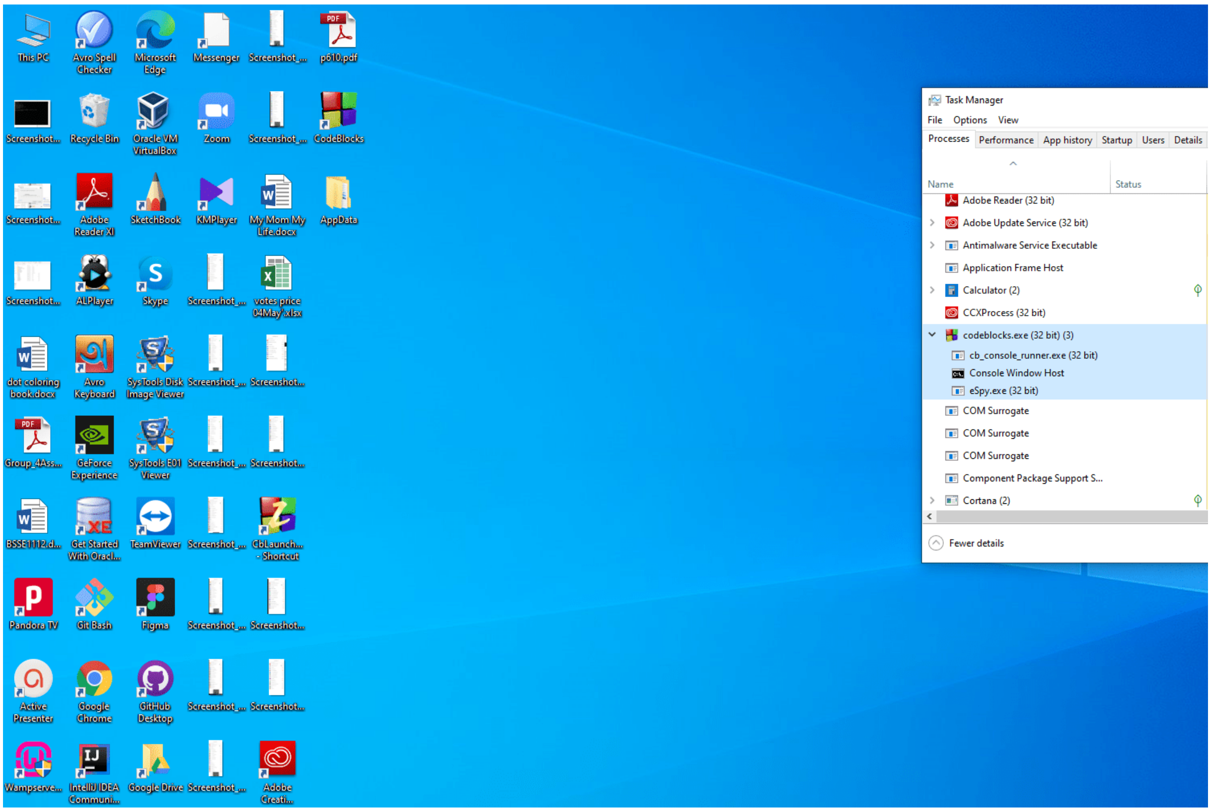
★ First Click at the executable file (.exe). A console will be prompt and will disappear within a few seconds. The keystrokes are being logged from now on. In the same directory where the .exe is located, a file named “log.txt” will be there, containing all the keystrokes logged so far.



```
welcome to the system
[ENTER]
your keystrokes are being logged (even this texts are logged)
[LBUTTON]
[LBUTTON]
guerillamail.com
[ENTER]
[LBUTTON]
facebook.com
[CAPSLOCK]
I PRESSED CAPS LOCK
[ENTER]
[CAPSLOCK]
capslock pressed again, so letters are small
[SHIFT]
[LBUTTON]
[CTRL]
[LBUTTON]
[LBUTTON]
[LBUTTON]
[LBUTTON]
```

Figure 1: The log.txt (the keystrokes logged)

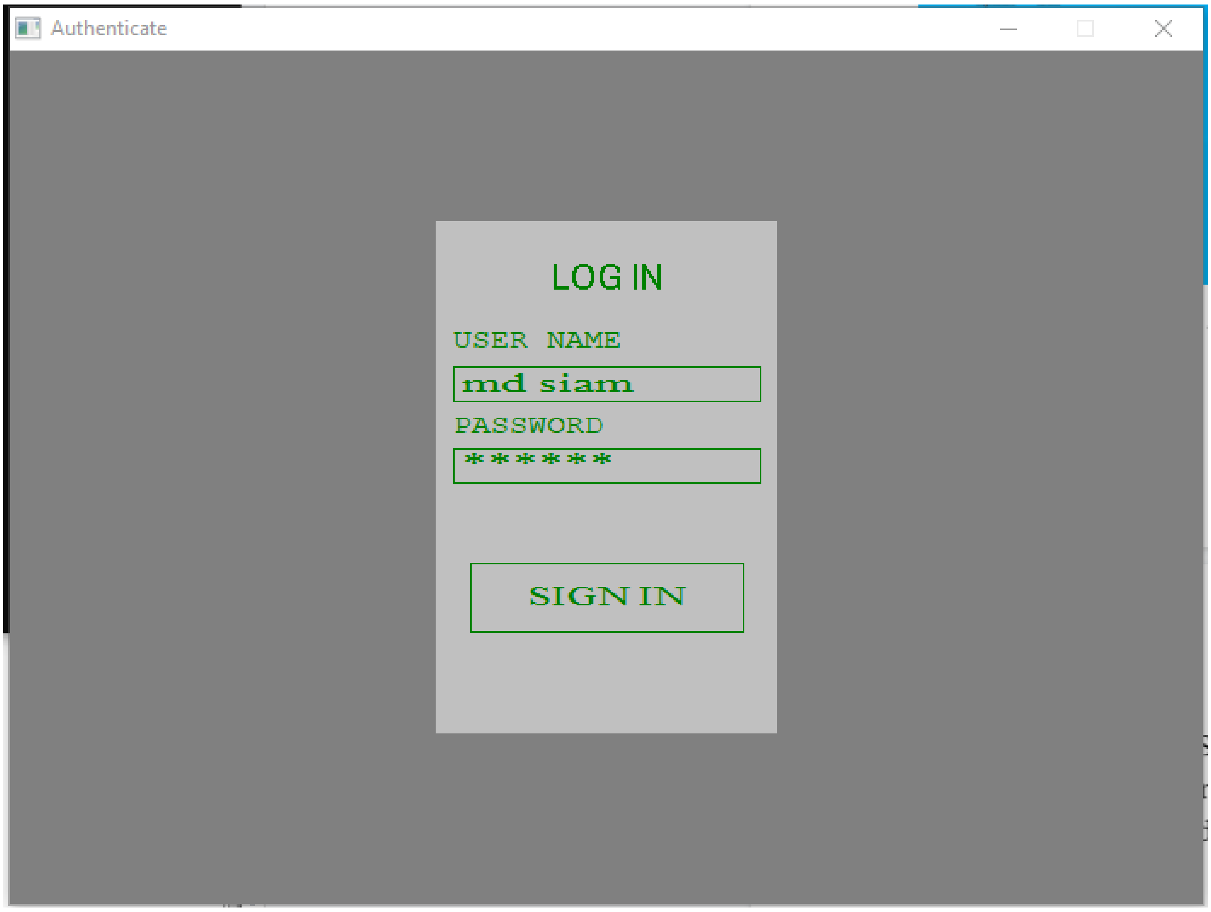
★ Screenshots will be taken automatically every 15 minutes. All the screenshots can also be found in the same folder with .bmp file extension. Image names would be in the format imageX.bmp ...



Name	Status
Adobe Reader (32 bit)	
Adobe Update Service (32 bit)	
Antimalware Service Executable	
Application Frame Host	
Calculator (2)	
CCXProcess (32 bit)	
codeblocks.exe (32 bit) (3)	
cb_console_runner.exe (32 bit)	
Console Window Host	
eSpy.exe (32 bit)	
COM Surrogate	
COM Surrogate	
COM Surrogate	
Component Package Support S...	
Cortana (2)	

Figure 2: image2.bmp (the screenshot captured)

★ Press CTRL + SHIFT + TAB to launch the login form. User needs to authenticate himself to get into the control panel. Default UserID and password is “md siam” and “siam01” (case sensitive and without quotes)



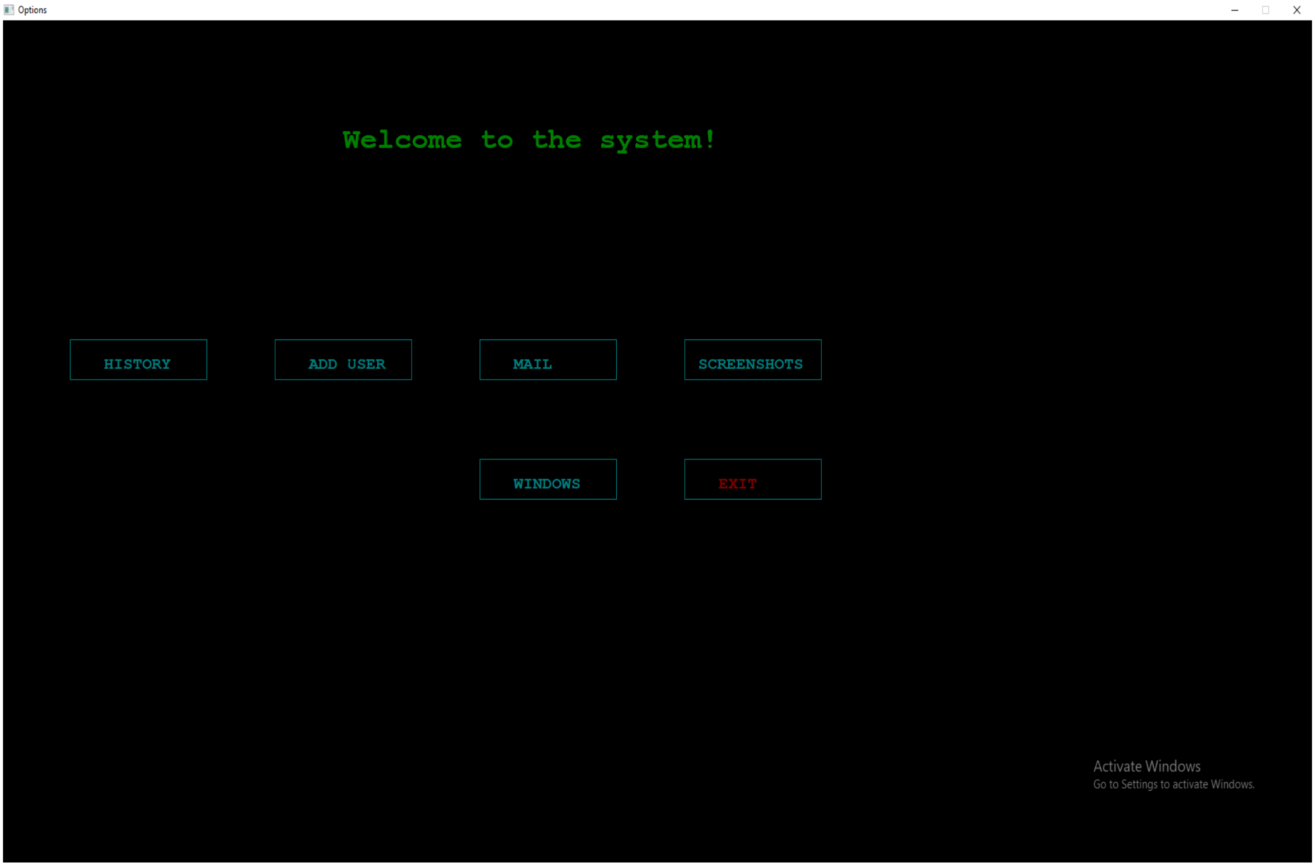
md siam

******

Figure 3: Login form

★ Upon successful authentication, a control panel will launch. User will find several options there

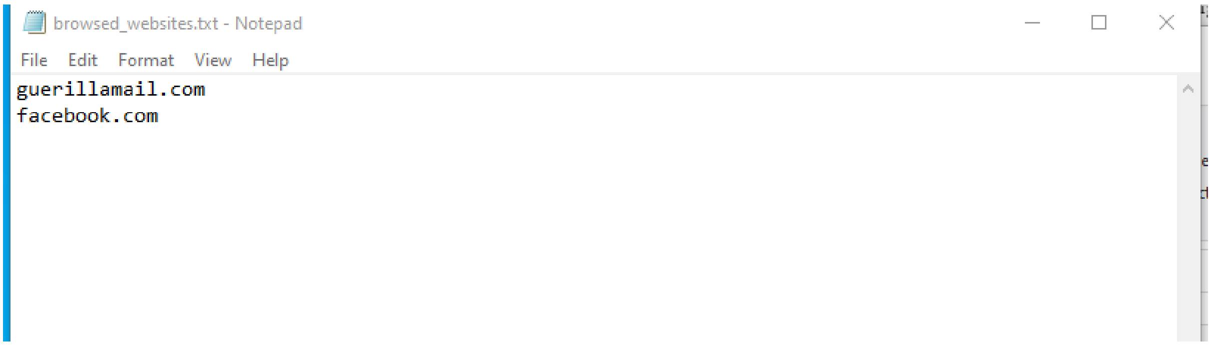
- History
    - To view the browsing history
- Add User
- Mail
- Screenshots
- Windows
- Exit



Welcome to the system!

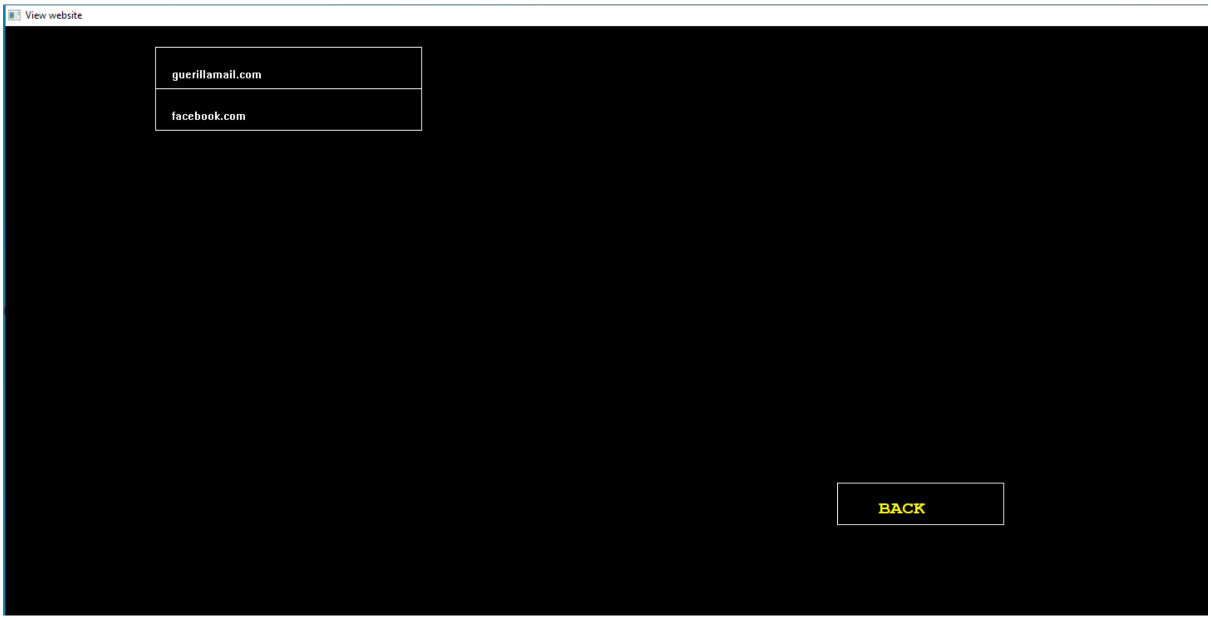
Figure 4: Menu

❖ Upon clicking on HISTORY, “browsed_websites.txt” file will get updated by parsing the logged keystrokes and will also be displayed in the GUI.



```
guerillamail.com
facebook.com
```

Figure 5: browsed_websites.txt

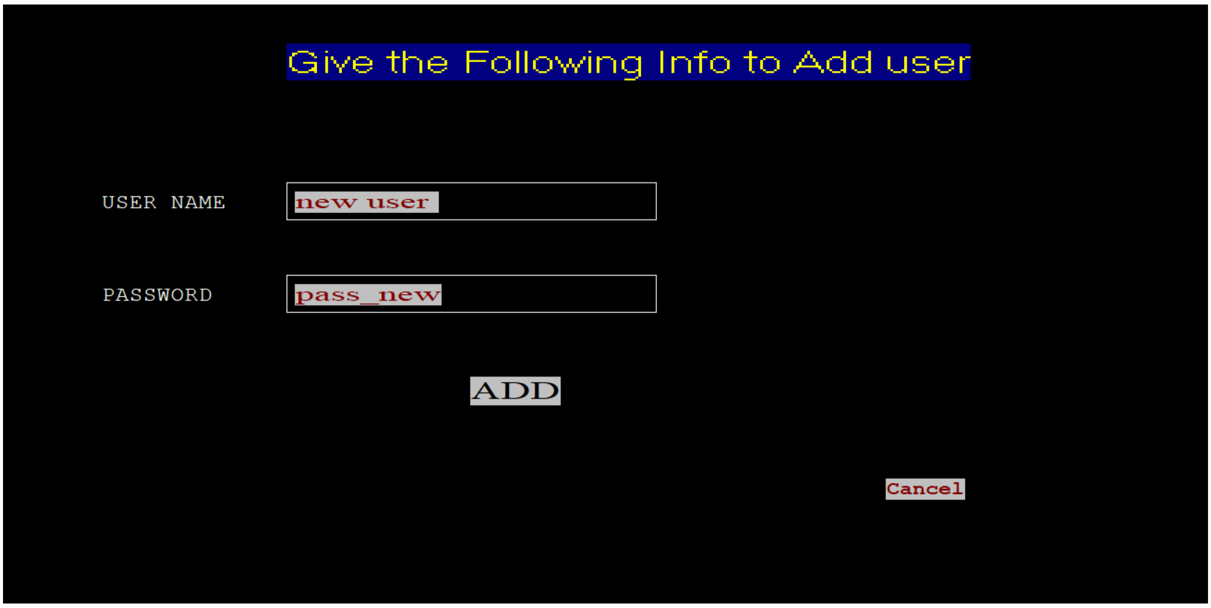


```
guerillamail.com
facebook.com
```

Figure 6: list of browsed website from GUI

★ Clicking back will bring us back to the menu.

★ Upon clicking on ADD USER, a new window will be prompted. User needs to input the USERNAME and PASSWORD and click ADD to add a new user.



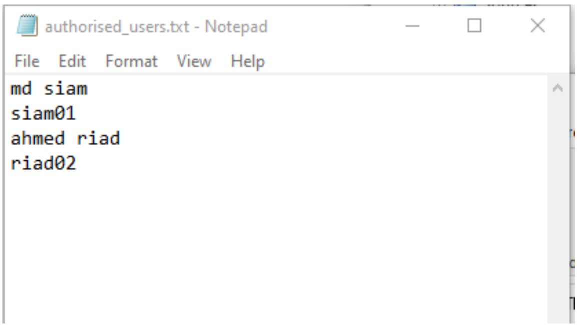
Give the Following Info to Add user

USER NAME: new user

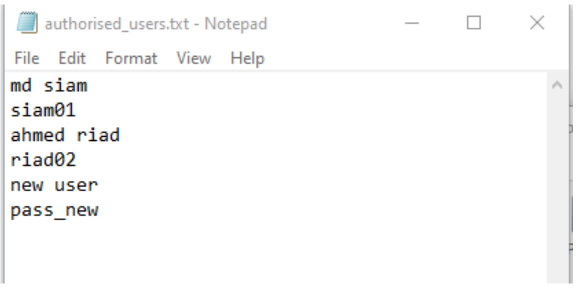
PASSWORD: pass_new

Figure 7: adding user (input values were "new user" "pass_new")

❖ After clicking ADD, the new user will be added, and we can view the user list on the "authorised_users.txt" file.



```
md siam
siam01
ahmed riad
riad02
```



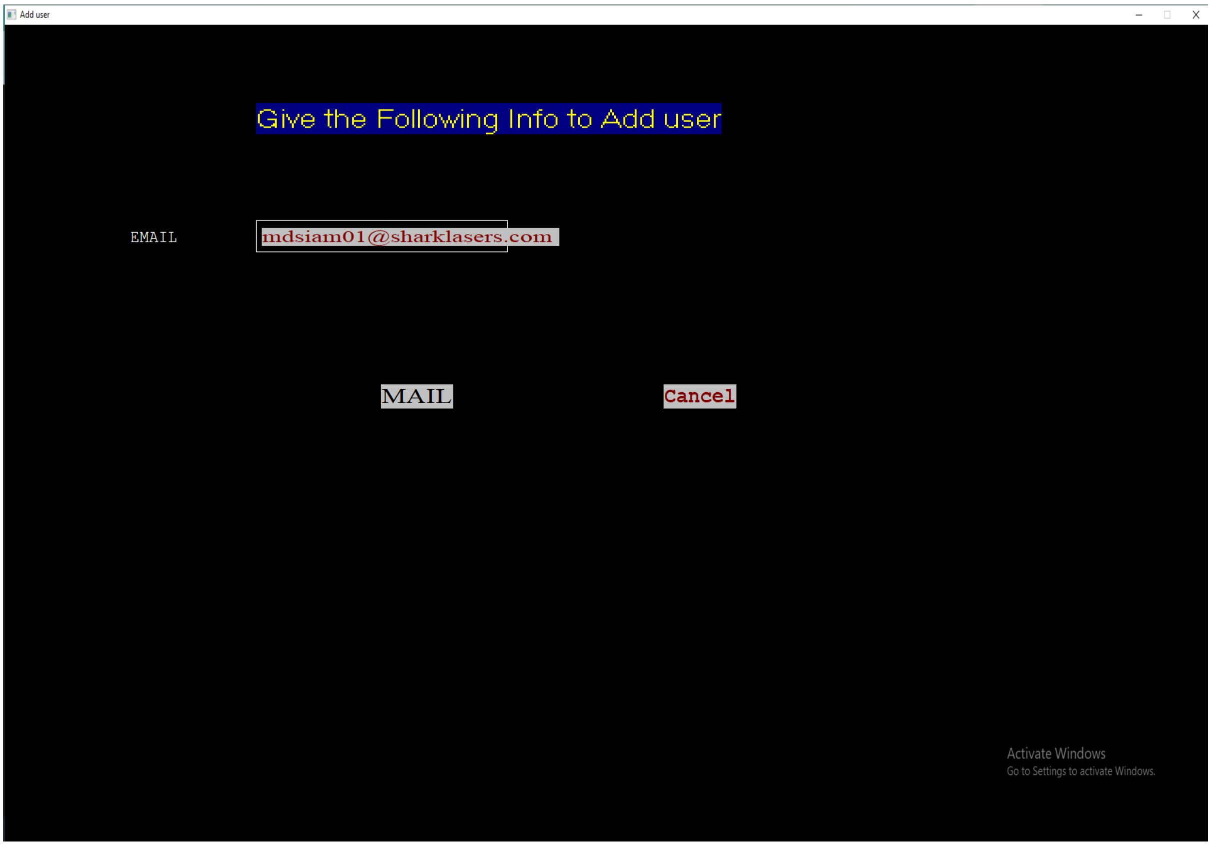
```
md siam
siam01
ahmed riad
riad02
new user
pass_new
```

Figure 8: before adding new user

Figure 9: after adding new user

★ Clicking CANCEL or ADD (prompts a SUCCESS window as well) will bring us back to the MENU again.

- ★ Upon clicking on MAIL, a new window will launch, user needs to provide recipient email address (it must be under sharklaser SMTP server)



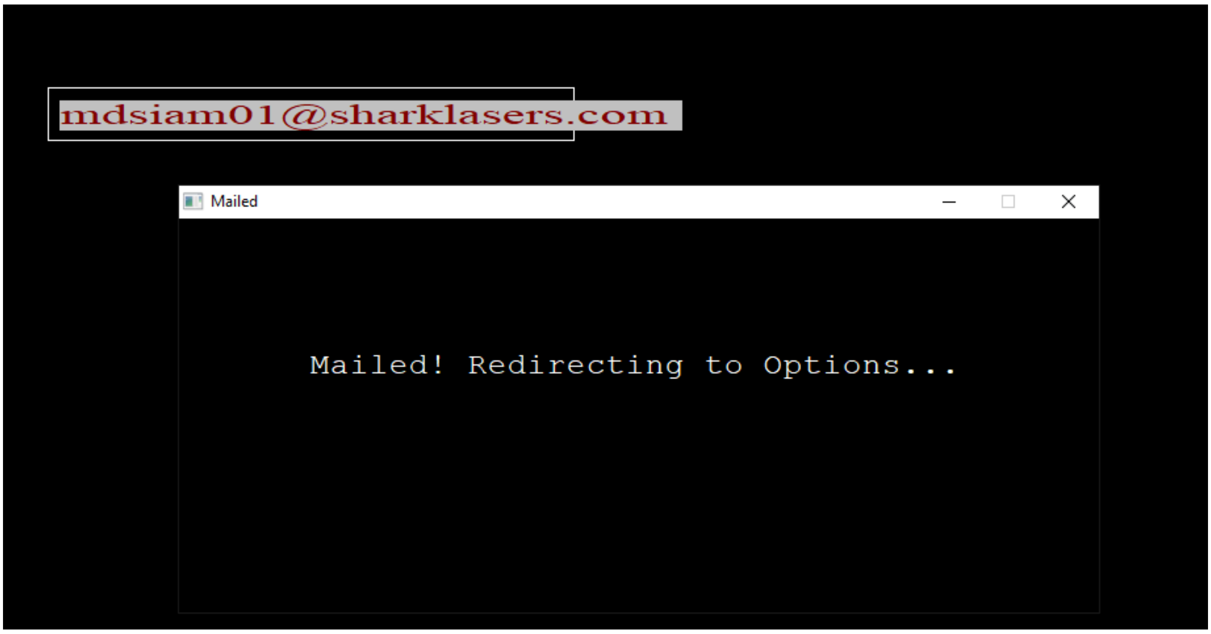
Add user
Give the Following Info to Add user

EMAIL mdsiam01@sharklasers.com

MAIL
Cancel
Activate Windows
Go to Settings to activate Windows.

Figure 10: send email to a non-default recipient

- ★ Clicking on MAIL or CANCEL will bring us back to MENU. Clicking MAIL does send the email to the input recipient and launches a success window. The delivered email can be found on guerillamail.com

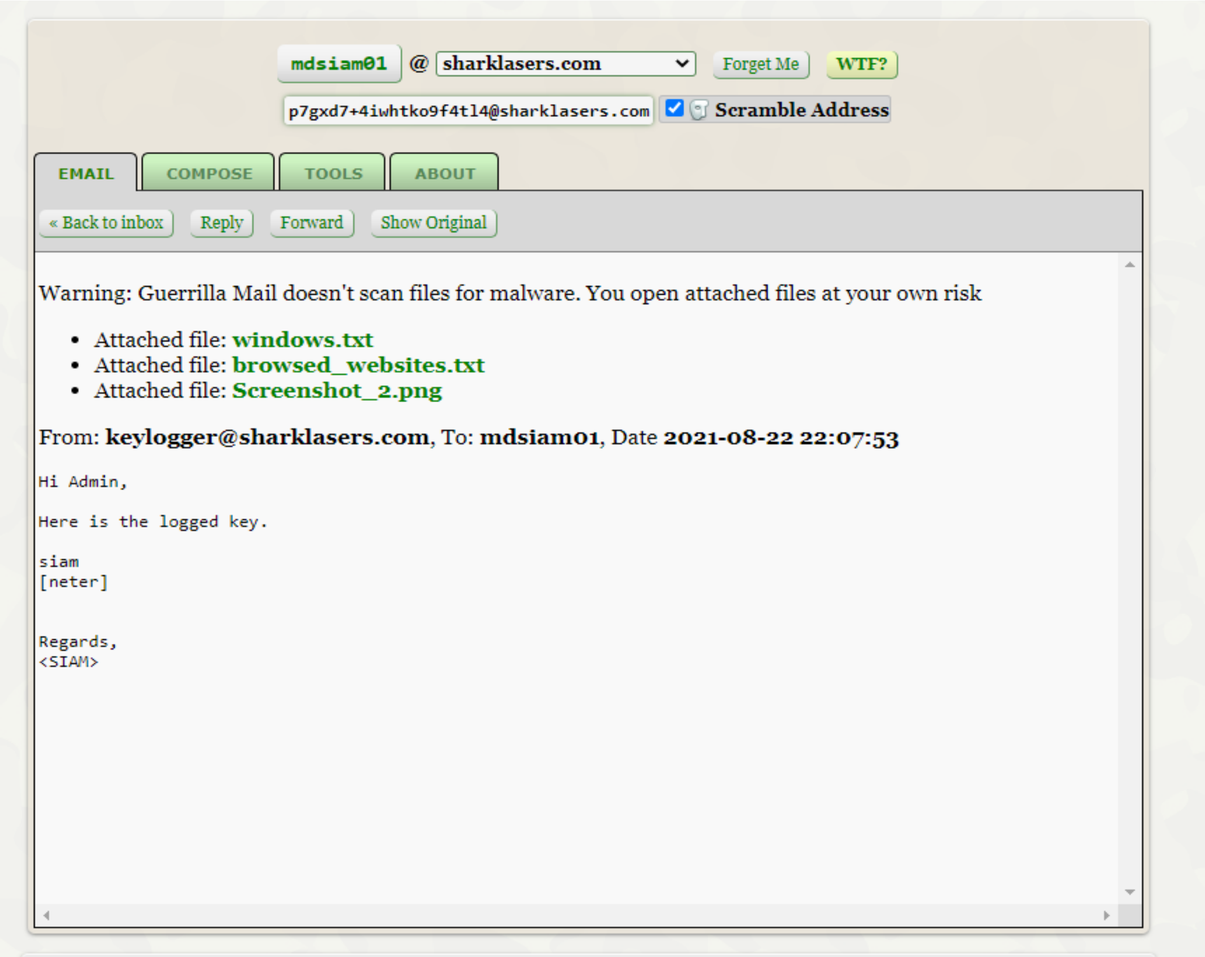


mdsiam01@sharklasers.com

```
Mailed! Redirecting to Options...
```

Figure 11: Mail sending Confirmation

★ The sent email can be found in guerillamail.com



Warning: Guerrilla Mail doesn't scan files for malware. You open attached files at your own risk

- Attached file: **windows.txt**
- Attached file: **browsed_websites.txt**
- Attached file: **Screenshot_2.png**

From: **keylogger@sharklasers.com**, To: **mdsiam01**, Date **2021-08-22 22:07:53**

```
Hi Admin,

Here is the logged key.

siam
[neter]


Regards,
<SIAM>
```

Figure 12: sent email on guerillamail.com

★ Clicking on SCREENSHOTS will show the already captured screenshots one by one, a user needs to press ANY key to view the next screenshots. It terminates once all the screenshots are viewed

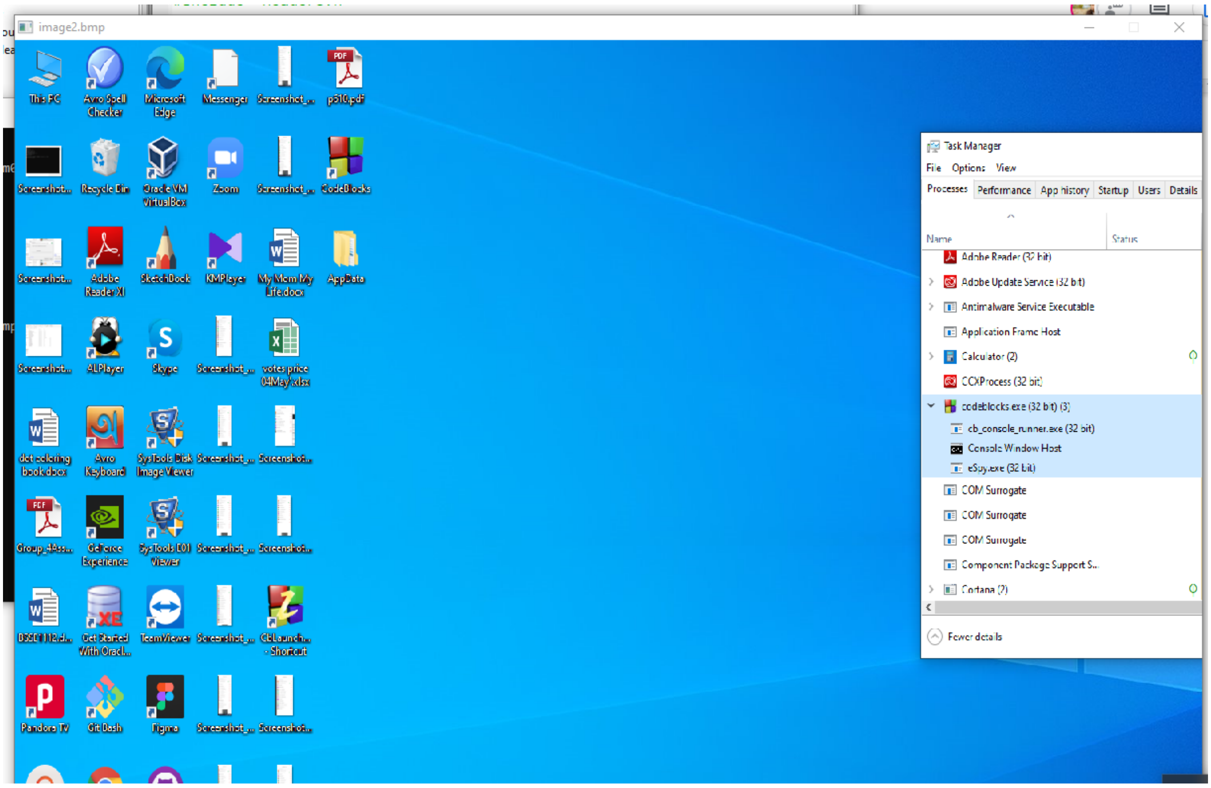
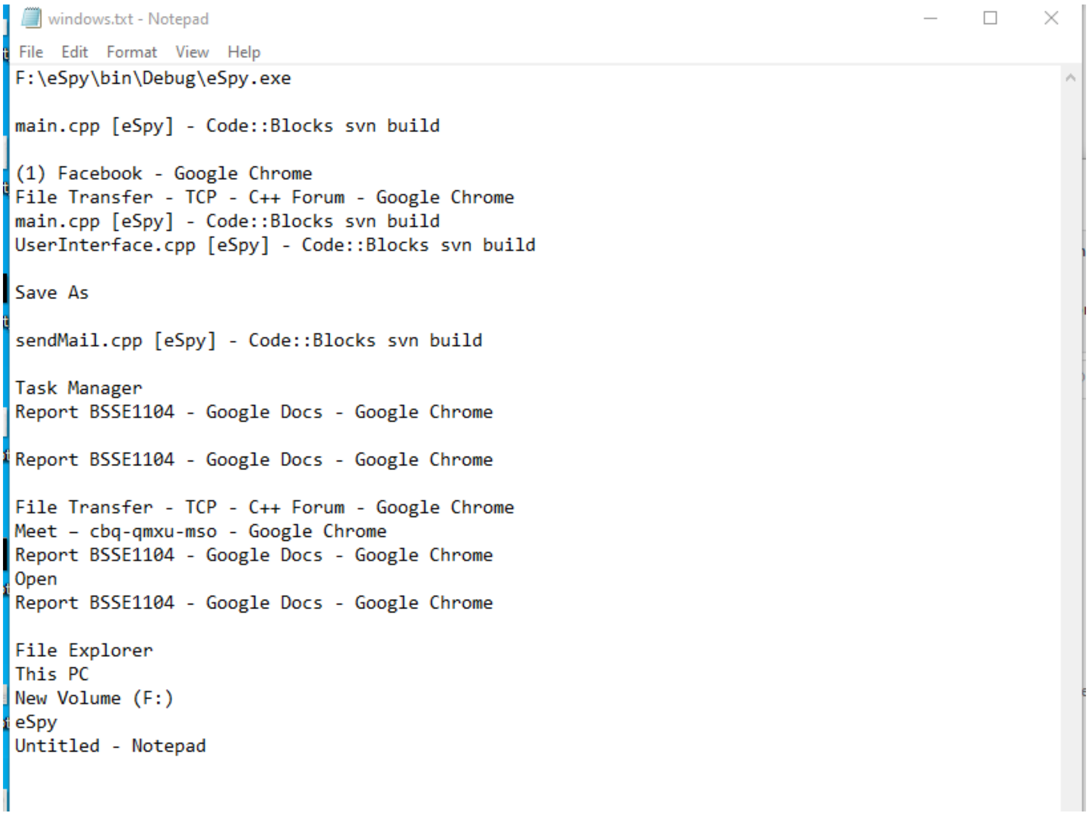


Figure 13: view Screenshots on GUI

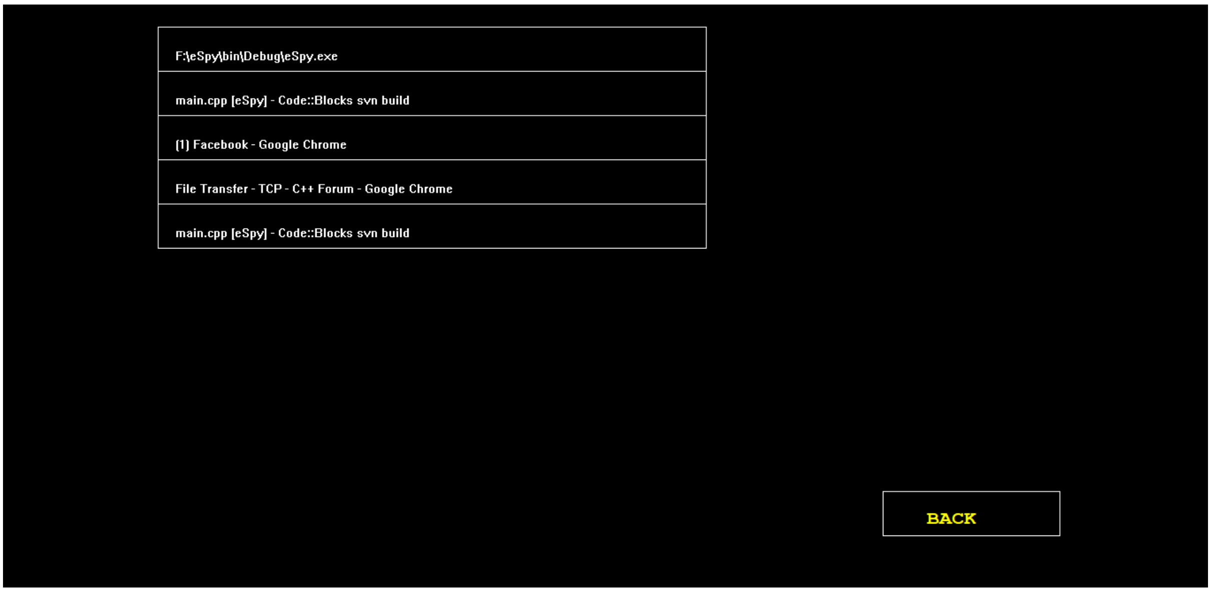
★ The program keeps track on the window/tab changes and saves them on “windows.txt”



```
F:\eSpy\bin\Debug\eSpy.exe

main.cpp [eSpy] - Code::Blocks svn build

(1) Facebook - Google Chrome
File Transfer - TCP - C++ Forum - Google Chrome
main.cpp [eSpy] - Code::Blocks svn build
UserInterface.cpp [eSpy] - Code::Blocks svn build

Save As

sendMail.cpp [eSpy] - Code::Blocks svn build

Task Manager
Report BSSE1104 - Google Docs - Google Chrome

Report BSSE1104 - Google Docs - Google Chrome

File Transfer - TCP - C++ Forum - Google Chrome
Meet – cbq-qmxu-mso - Google Chrome
Report BSSE1104 - Google Docs - Google Chrome
Open
Report BSSE1104 - Google Docs - Google Chrome

File Explorer
This PC
New Volume (F:)
eSpy
Untitled - Notepad
```

Figure 14: windows.txt

★ Upon clicking on windows, the recently changed windows will be shown



F:\eSpy\bin\Debug\eSpy.exe

main.cpp [eSpy] - Code::Blocks svn build

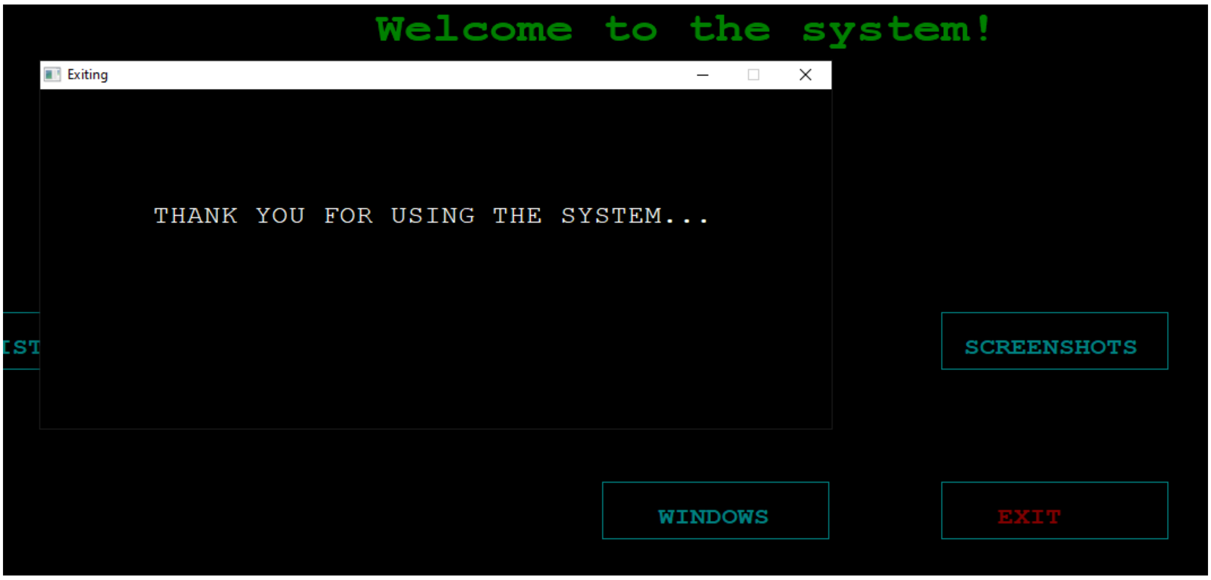
(1) Facebook - Google Chrome

File Transfer - TCP - C++ Forum - Google Chrome

main.cpp [eSpy] - Code::Blocks svn build

Figure 15: the list of the recent windows from GUI

★ Upon clicking on EXIT, a THANK YOU window pops up and the GUI gets terminated (but logging, taking screenshots and sending mail still keeps going)



THANK YOU FOR USING THE SYSTEM...

Figure 16: THANK YOU window (popped for clicking EXIT)

8. Future Scope

I had been working on a feature - when a user notices and terminates the program himself(from the taskbar), sending an alarm to the admins via email. That's half done, I could not accomplish due to time shortage for now. I intend to implement that feature since that makes the project overall usable to any of the organizations who are concerned about their confidentiality, security and proper usage of resources.

9. Conclusion

I tried to learn and implement everything from scratch. Working on this project has helped me have a better idea about windows and network programming. I have fulfilled all the functionalities that I was committed to from the very beginning of the project. Being new to this programming paradigm, this SPL has challenged my limits. I enjoyed learning on-the-go and implementing. I am happy that I could complete the project successfully and have taken on the challenges. I find myself more confident while solving a yet unseen problem, think critically and be more confident for my future projects.

10. Appendix

I have learned so many new programming dimensions such as network programming, windows programming. Applying them raw enabled me to visualize things more clearly and bear a transparent concept on them.

Furthermore, the obtained knowledge and getting used to programming of new dimensions (that I am not already comfortable with) will help me build confidence to deal with the problems more confidently that I face in the future. Also the caliber to work with that knowledge and maintaining a large file will help me do bigger projects in a more efficient and convenient way in the future.

Github URL: https://github.com/Md-Siam07/SPL1

11. References

1. https://securelist.com/keyloggers-implementing-keyloggers-in-windows-part-two/36358/?fbclid=IwAR31e8X2otQGtRZ9QBUpoFIHV_Pd4zTVvXD6rWd5RuFW2SwsKDCPLWHX33w
2. https://www.quora.com/How-to-program-a-keylogger?fbclid=IwAR3DI6TRY9axffPKYnb6w-3L_oMdGQxmqzq1cGeaTdoWTdGAVRwG0DY2ha8
3. https://docs.microsoft.com/en-us/dotnet/api/system.drawing.imaging.bitmapdata?redirectedfrom=MSDN&view=netframework-4.8
4. http://www.exforsys.com/tutorials/c-language/concept-of-pixel-in-c-graphics.html
5. https://stackoverflow.com/questions/4674317/how-to-convert-bitmap-to-byte-faster
6. **Hands-On Network Programming with C**
   Learn socket programming in C and write secure and optimized network code
   by **Lewis-Van-Winkle**
7. https://developer.mozilla.org/en-US/docs/Web/HTTP/Status