“INVESTIGATING THE PARALLELS BETWEEN USING A RAT-STYLE SOFTWARE FOR MALICIOUS PURPOSES AND VIRTUOUS INTENT”

By Daniel Broomhead

27016005

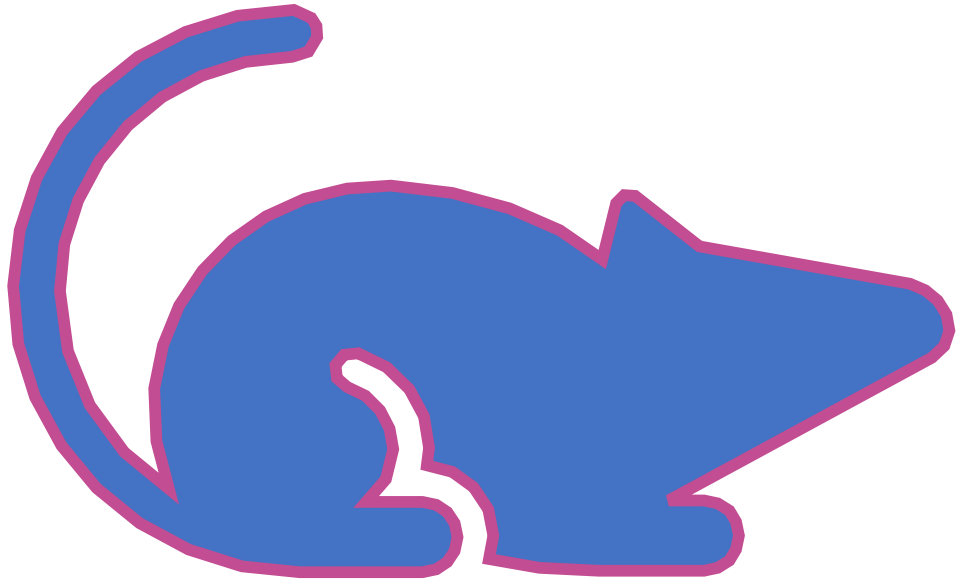
WHAT IS A RAT?

Remote Administration Tool:

- Virtuous
- Productive
- Benevolent
- Benignant

Remote Access Trojan:

- Promiscuous
- Malicious
- Deceitful
- Surreptitious



At The Heart of it, they are the same ideology and technology.

HOW ARE RATS USED?

- Remote Desktops
- Administration
- Computers
- Remote access software
- Usually part of a bigger system
- Delivering a payload

Backdoor, Remote Access Tool/Remote Access Trojan (RAT)

A backdoor is an application allowing remote access to a computer. The difference between this type of malware and a legitimate application with similar functionality is that the installation is done without the user's knowledge.

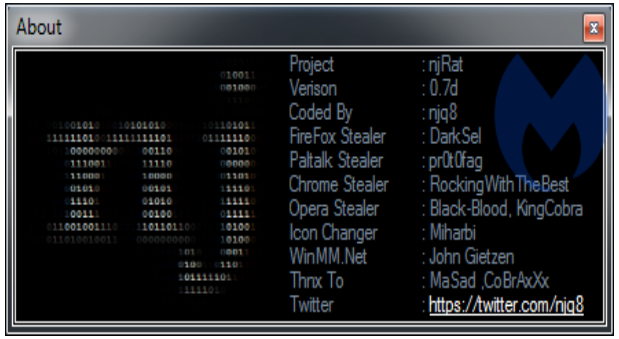
Typical backdoor functionality includes the capability to send files to the host computer and execute files and commands on it, and to exfiltrate (send) files and documents back to the attacker. Often this is coupled with key-logging and screen-grabbing functionality for purposes of spying and data theft.

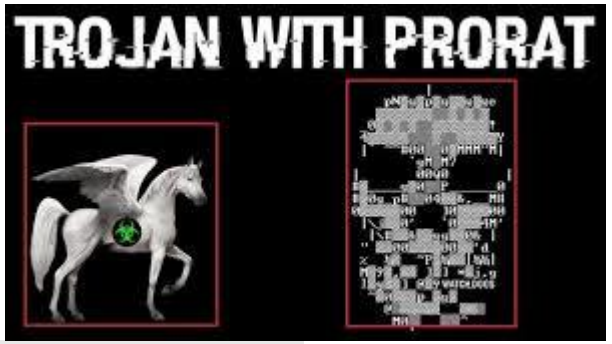
The term "RAT" (Remote Access Tool) can be considered a synonym to "backdoor", but it usually signifies a full bundle including a client application meant for installation on the target system, and a server component that allows administration and control of the individual 'bots' or compromised systems.

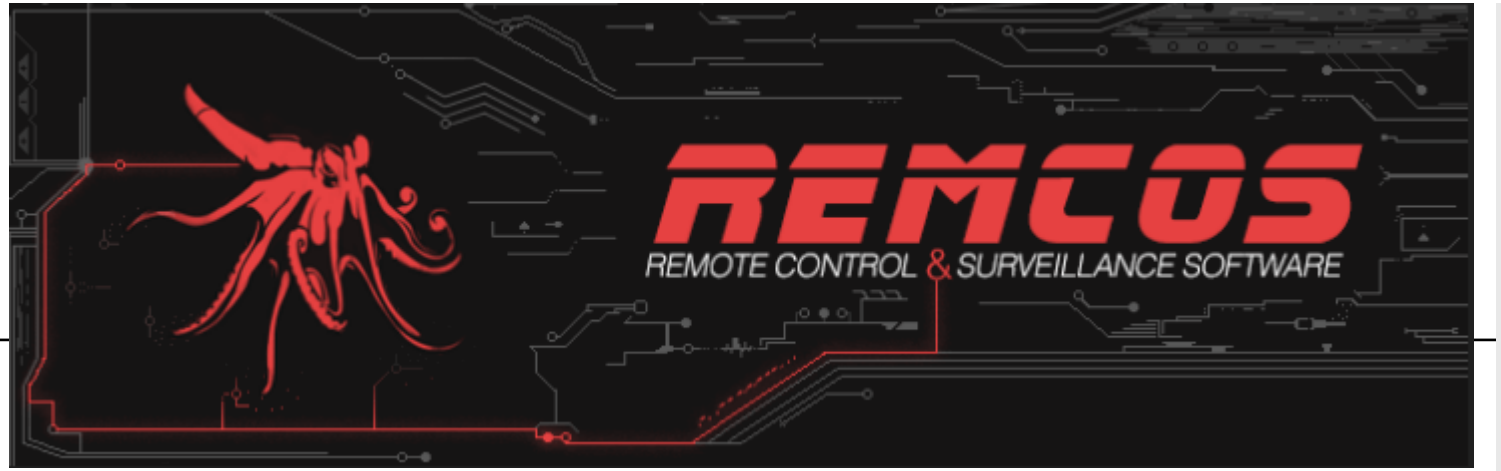
https://www.virusradar.com/en/glossary/backdoor

EXAMPLES:

LogMeIn Hamachi



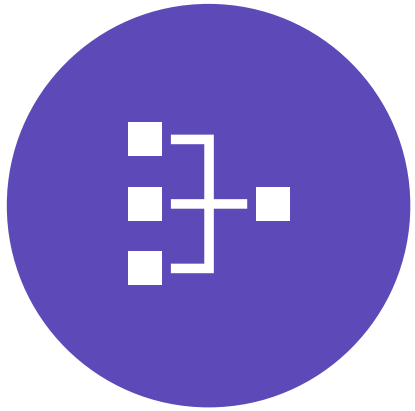




GENERAL CONSIDERATIONS FOR RATS:

LANGUAGE:

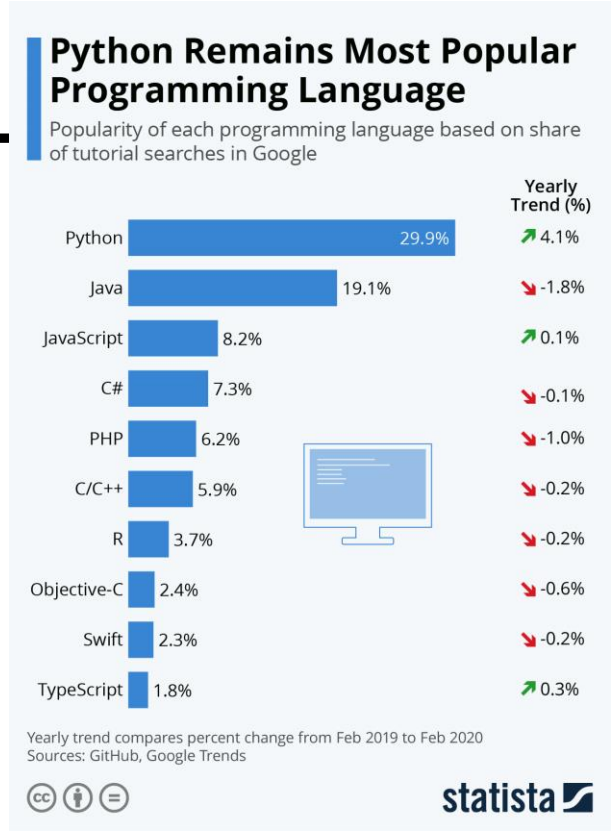
METHODOLOGY:

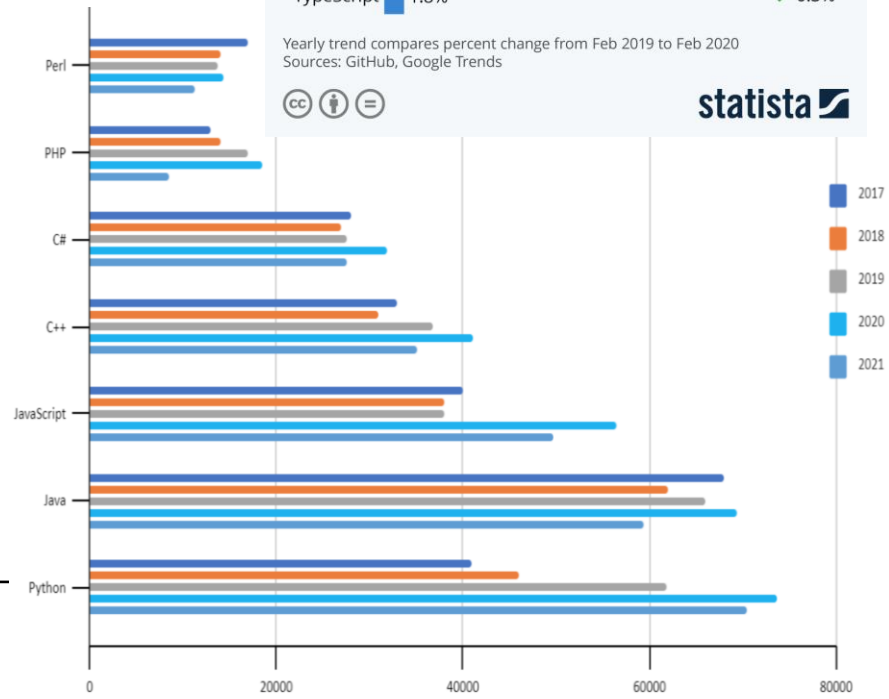


SYSTEM:

PYTHON 3:

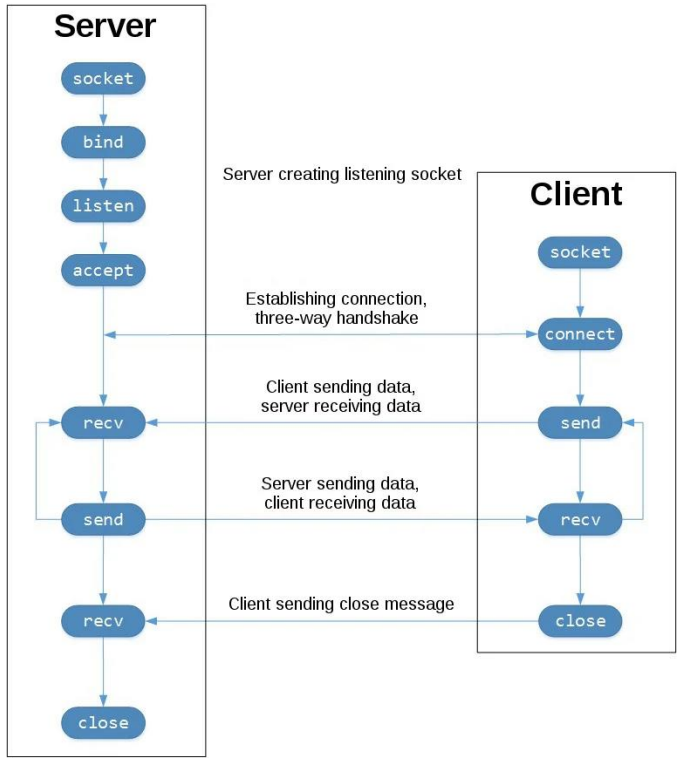
- Python 2 is now deprecated.
- Cross-Compatibility :Compatible with all major Platforms and Systems
- Robust Standard Library
- Depth of open-source frameworks, Extended libraries, community support
- Effective in the IoT sphere
- Tried & Tested Scalability
- Python is a desired Language in the workplace – so this will be a good project to demo
- Gaining popularity faster than ever
- Fairly confident in this language but looking to develop skills further





https://www.statista.com/chart/21017/most-popular-programming-languages/
https://www.codingdojo.com/blog/top-7-programming-languages

MY RAT: THE CRUCIAL DECISIONS



- Using the Python Socket module as an interface to the Berkeley Sockets API
- TCP Sockets
  - Reliable: Dropped Packets retransmitted – no data lost
  - In-order: Data is read in the order it was written

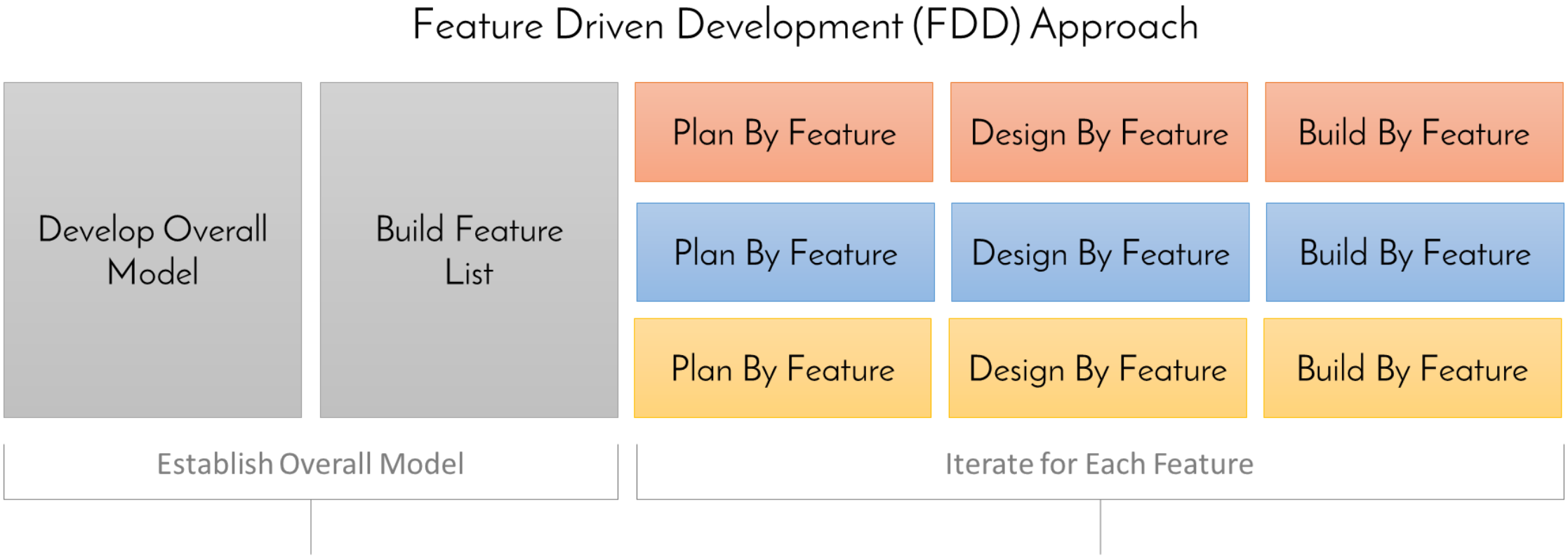
  UDP Does not share there properties and thus would not be appropriate.
- A reverse connection is also used meaning the client initiates the connection with the server.
- This prevents the firewall from blocking the connection as it sees the connection as being instantiated from within.

This has been developed on Windows 10
Mainly for use on Windows 10
Some of the features may work on Linux,
No testing has been conducted on MacOS

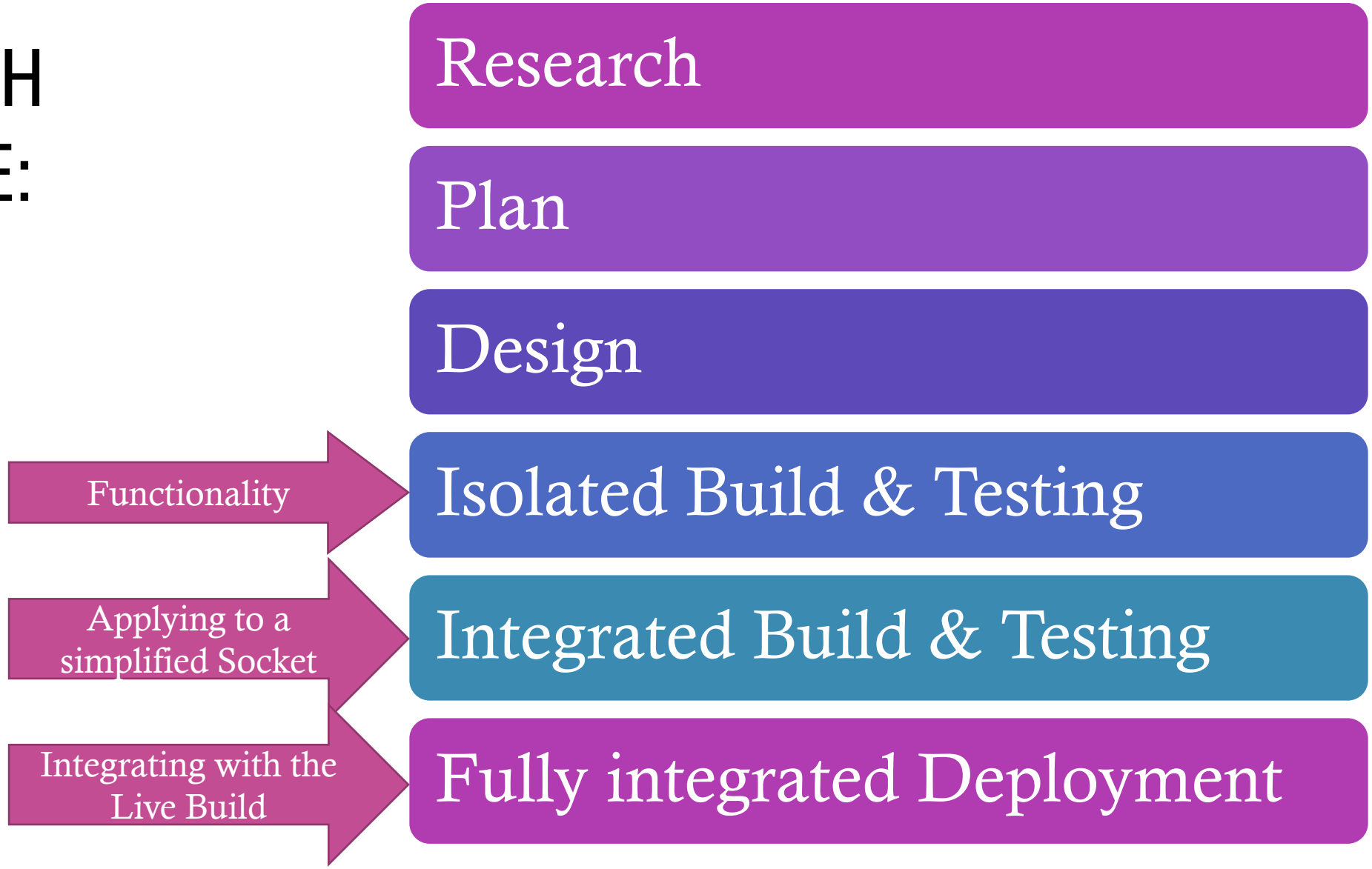
https://en.wikipedia.org/wiki/Berkeley_sockets

DEVELOPMENT IDEOLOGY

- The development of the functionality for the system followed a similar ideology to that of Feature driven development

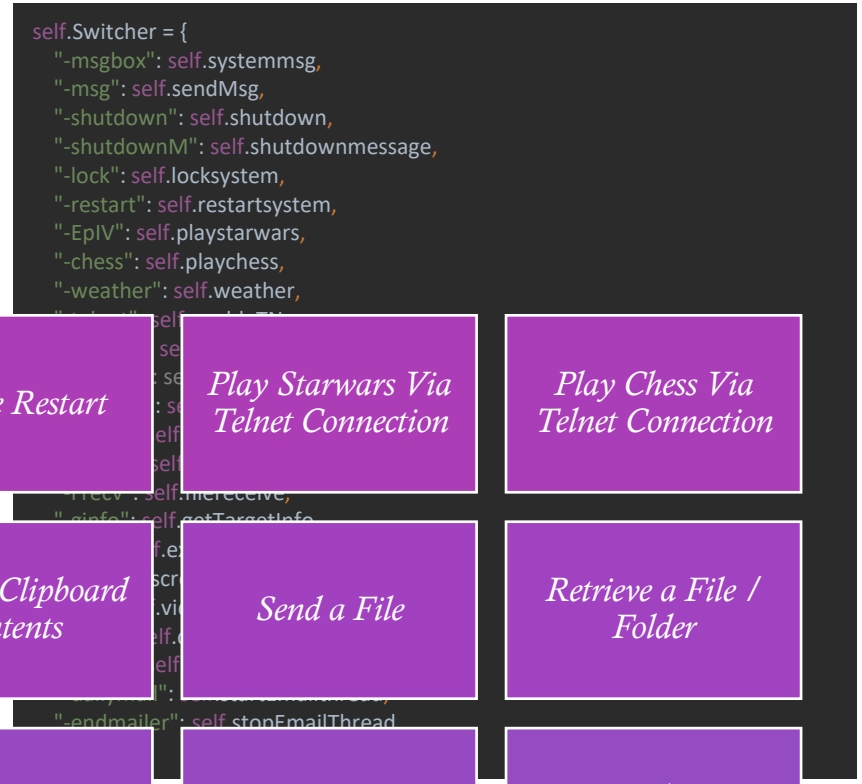


FOR EACH FEATURE:



Design Methodology offers easy expansion and advancements.

CURRENT FUNCTIONALITY:



Send a console message

Send a custom messagebox Alert

Remote Shutdown (+ Remote Shutdown with custom message)

Remote Lock Device

Remote Restart

Play Starwars Via Telnet Connection

Play Chess Via Telnet Connection

Check the Weather Via Telnet Connection

Enable Telnet Client

Start Keylogger → Stop Keylogger

Retrieve Keylogs

Retrieve Clipboard Contents

Send a File

Retrieve a File / Folder

Obtain information about the target system

Execute a file

Screenshot

Series of screenshots → Video

Reverse shell

Send an email

Auto emailer – start /stop

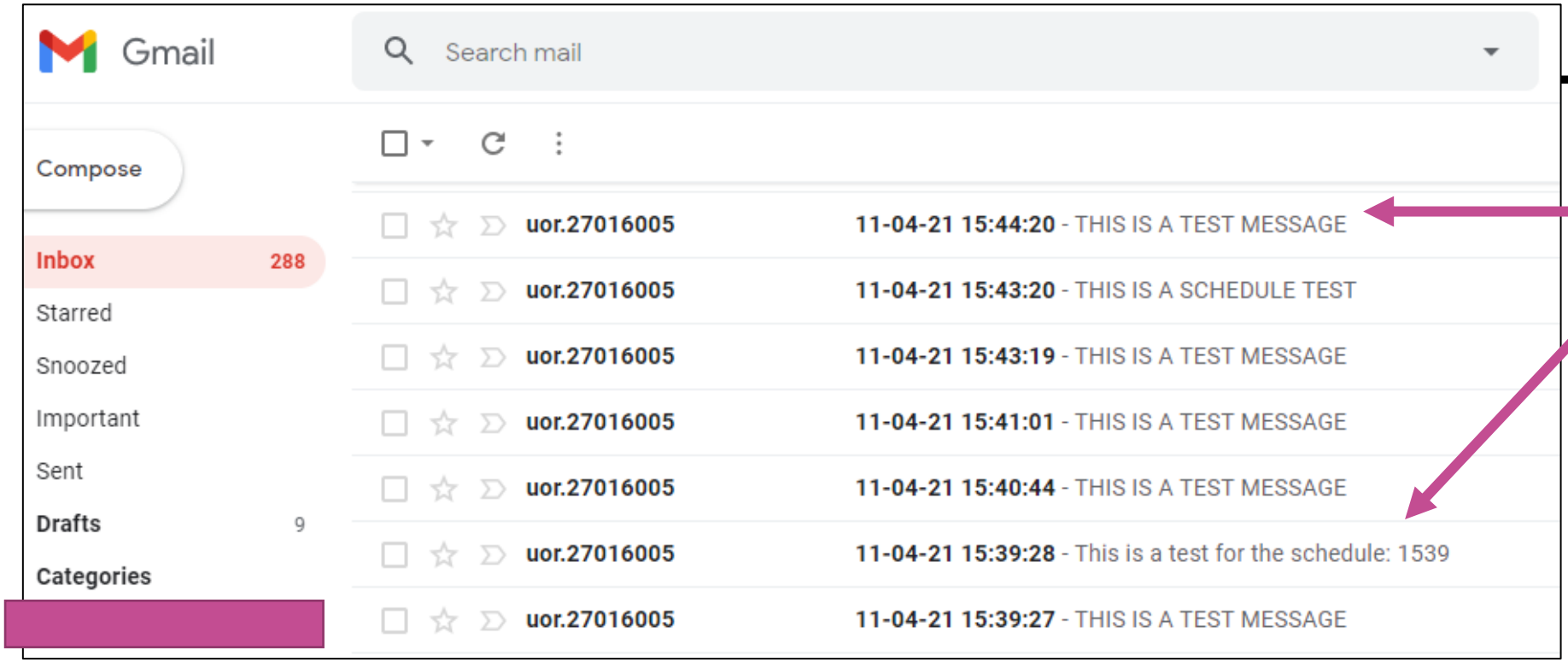
Record Webcam

Playback recording

Many of these will be demonstrated in the demo video.

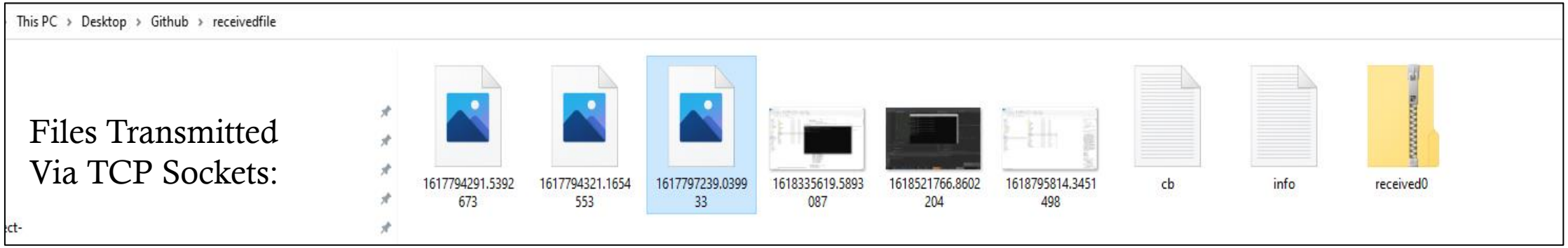
KEY LOGGER:

- Runs on separate thread
- Capture Ctrl+C, Ctrl+V shortcuts for copy and paste
- Creates a hidden text file to store the contents and be retrieved at another time
- Captures Clipboard contents
- Runs in the background.
- Invisible

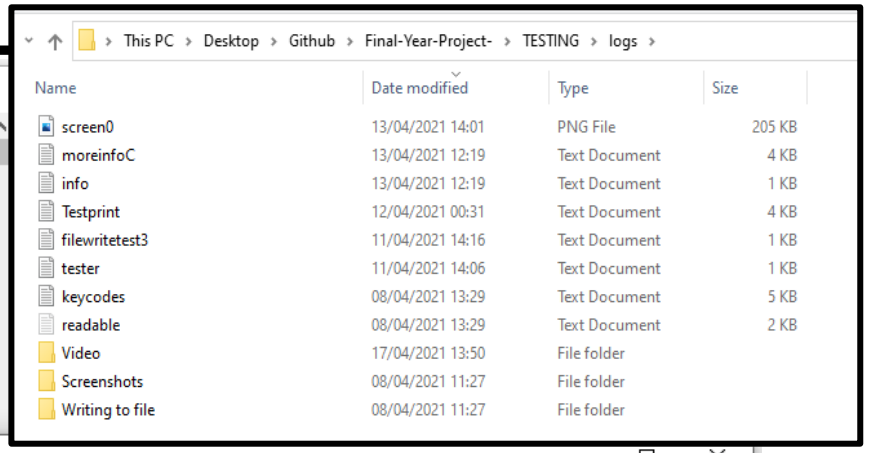


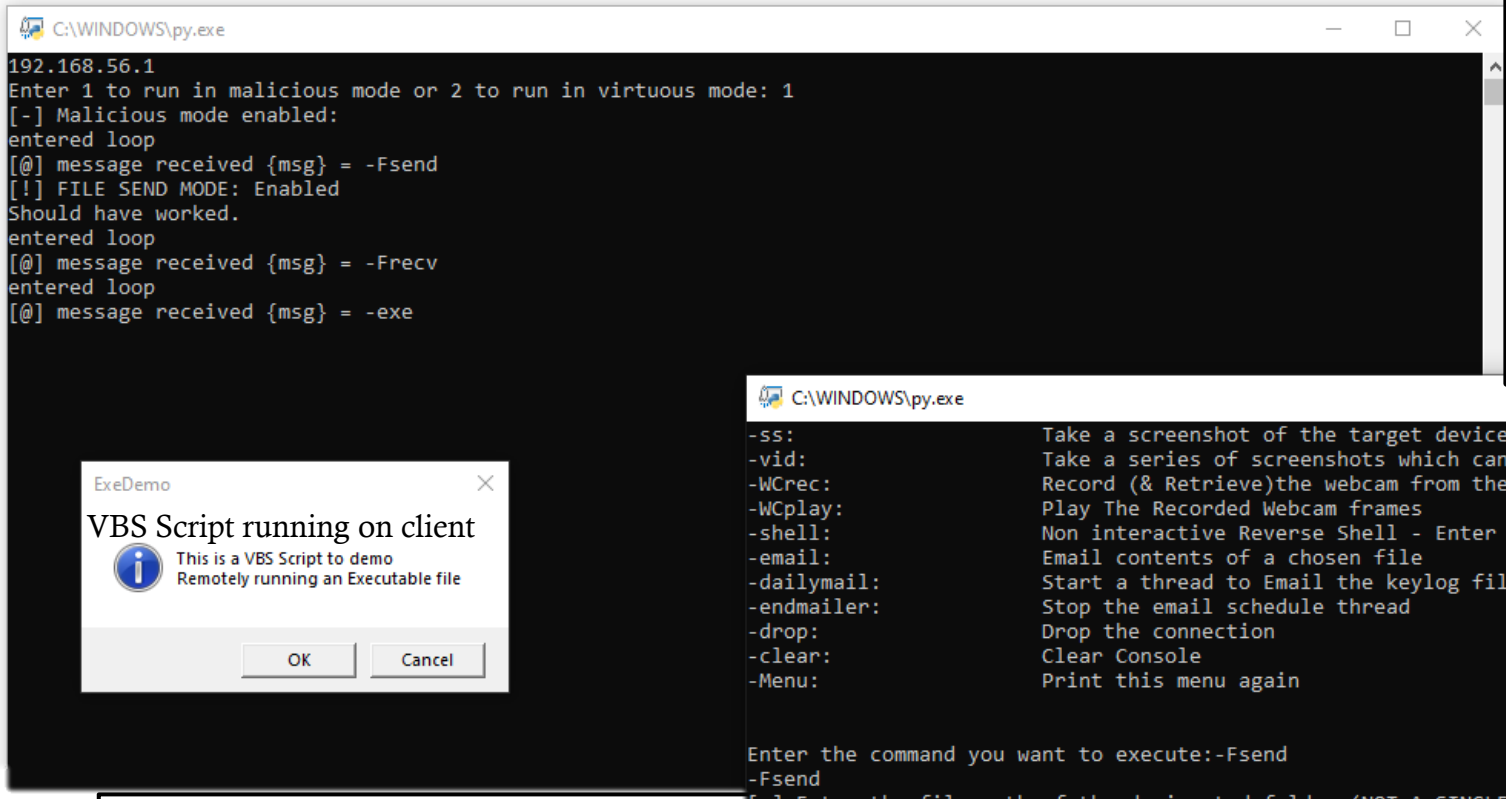
File contents Sent via email. Including for a Scheduler automailer

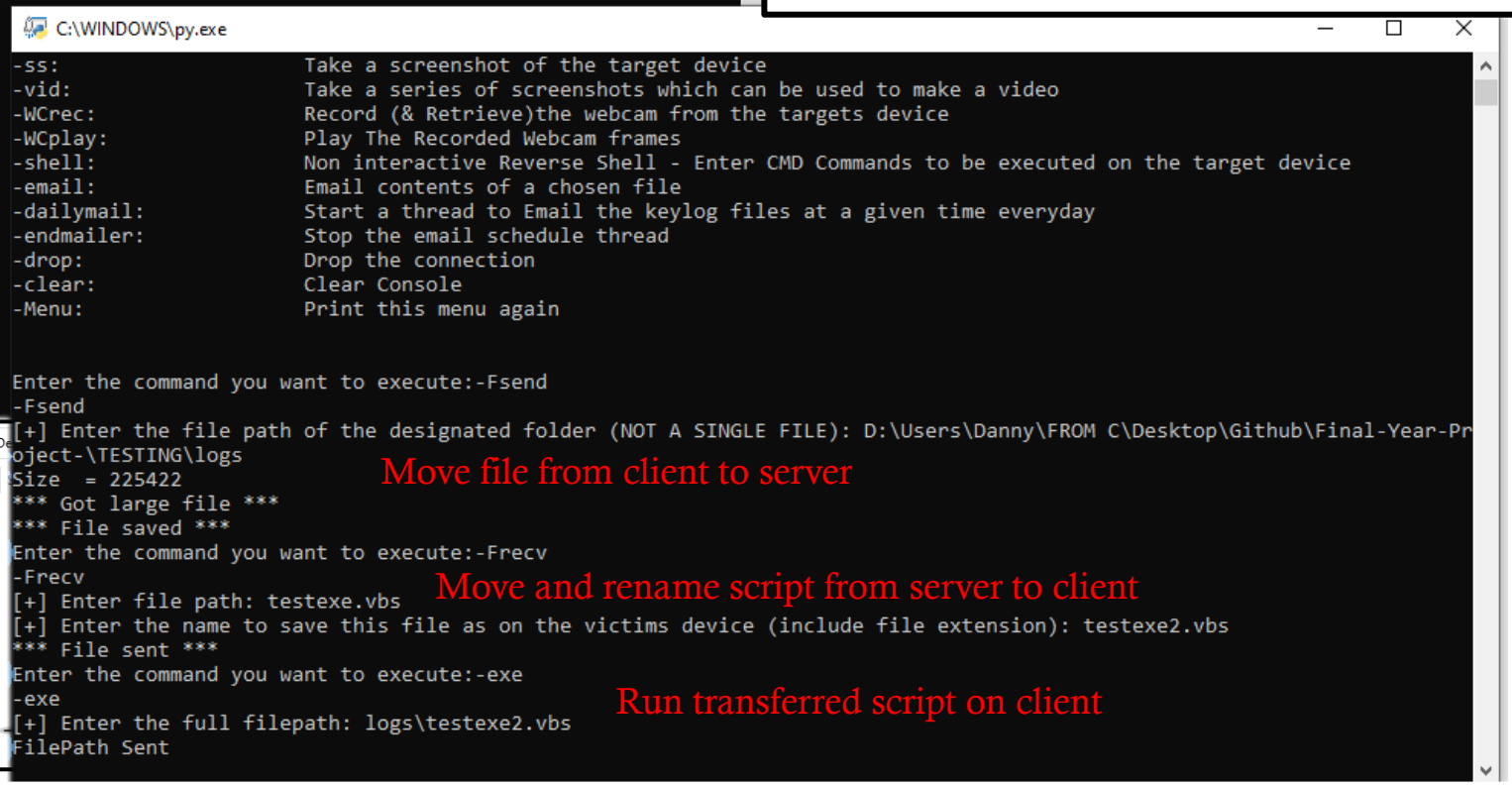
Files Transmitted Via TCP Sockets:



Directory on Client:



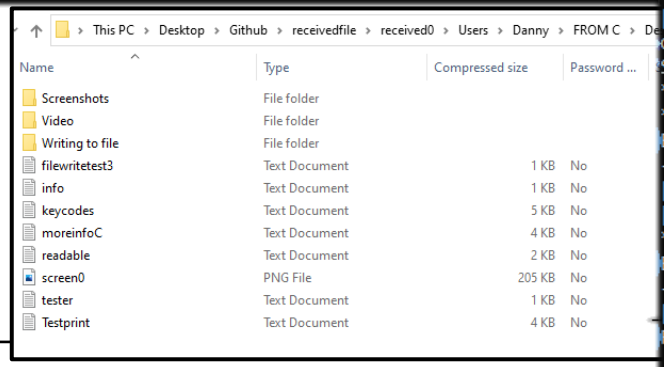




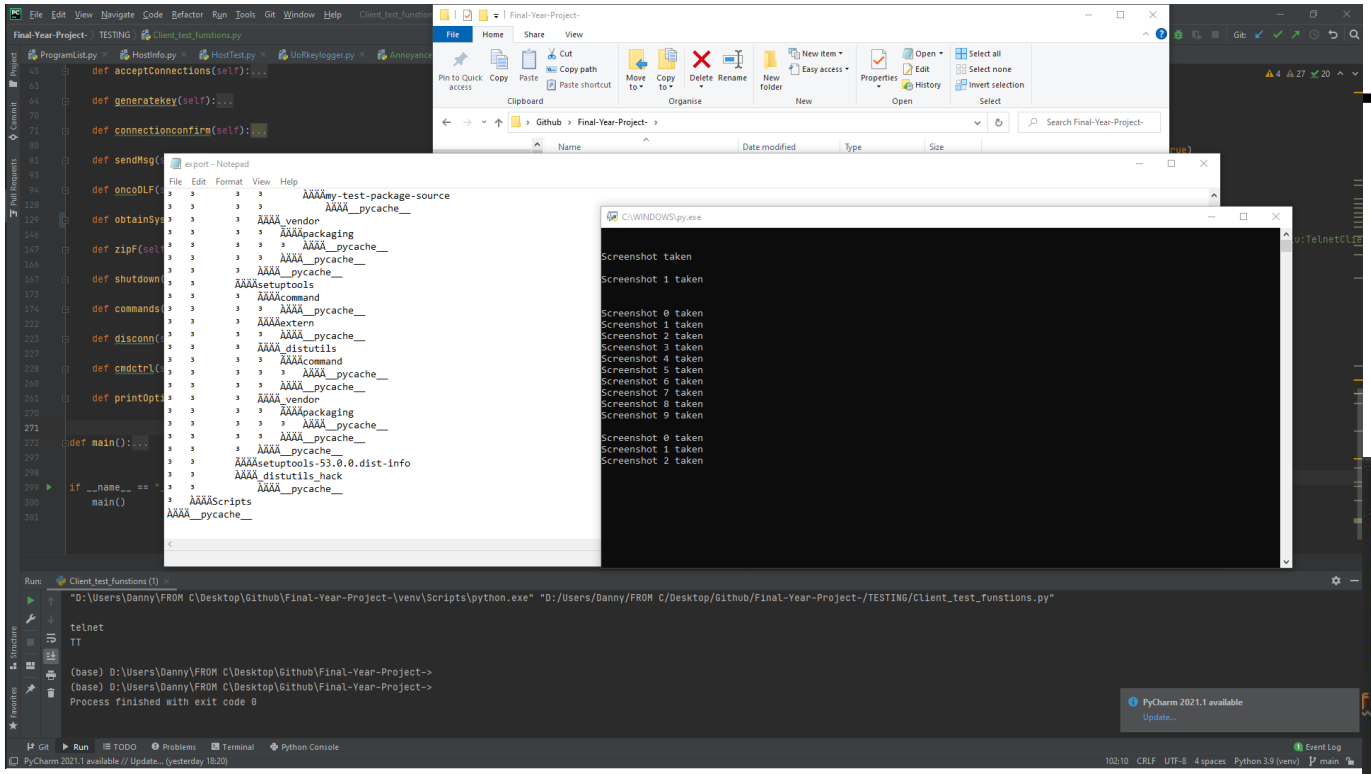
Move file from client to server

Move and rename script from server to client

Run transferred script on client

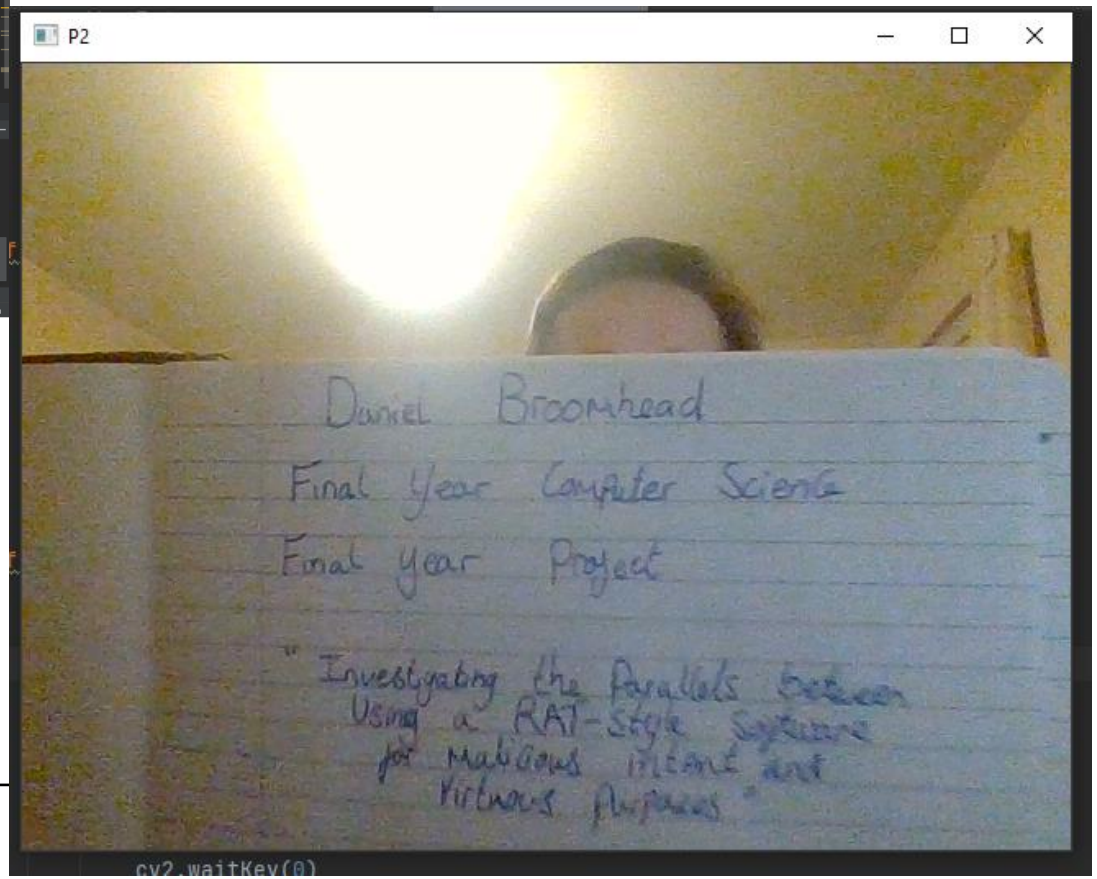


Directory on server:



Screenshot taken from victims device

Capture taken from target's webcam



C:\WINDOWS\py.exe

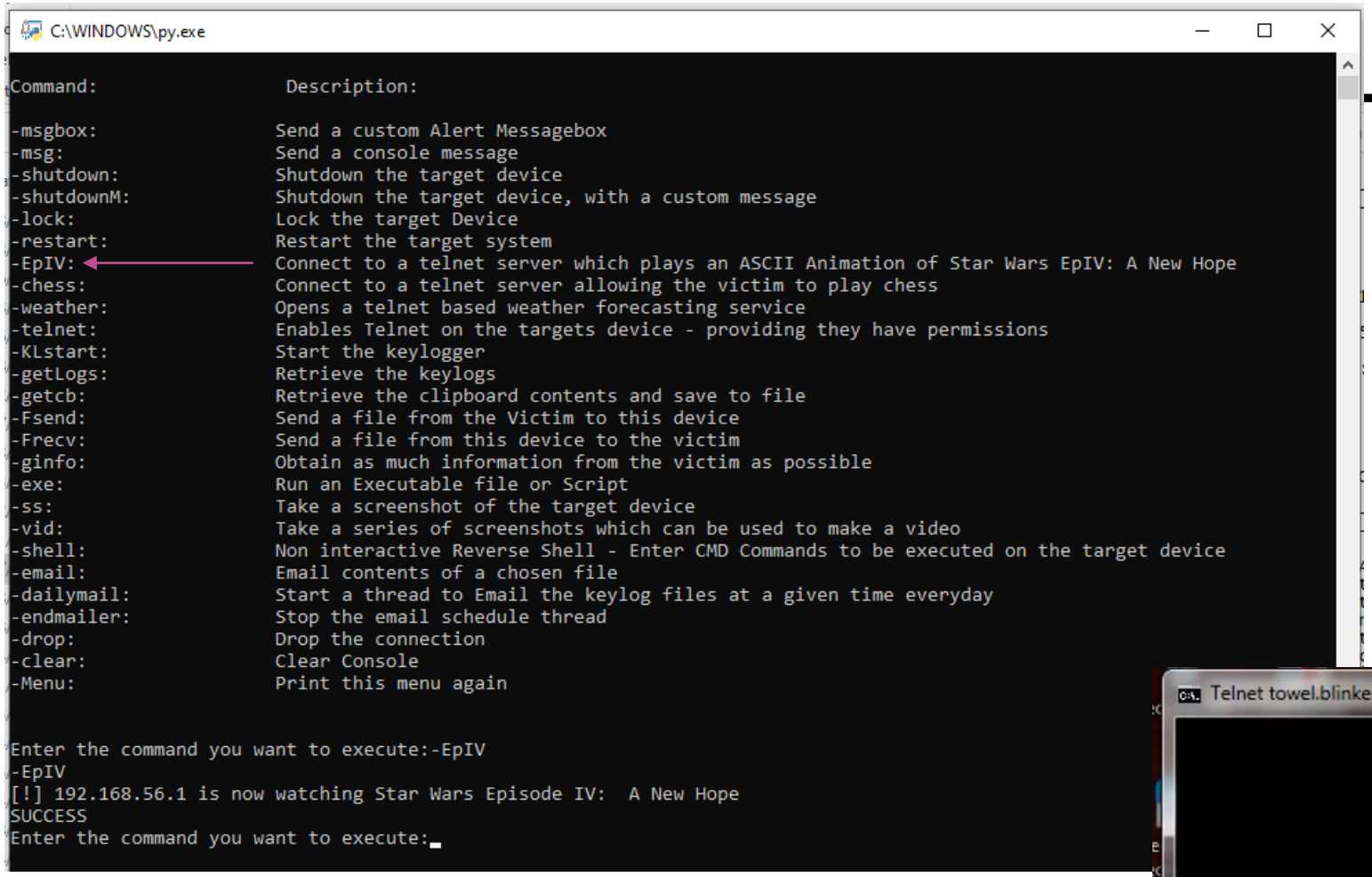
SERVER:

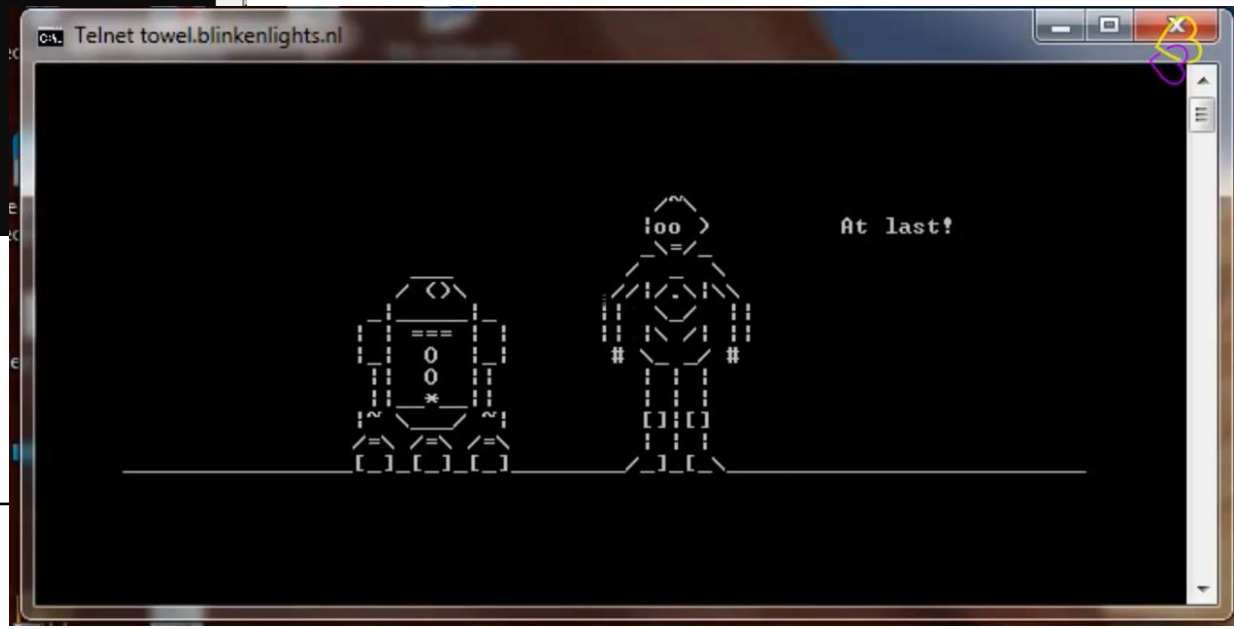
```
192.168.56.1
*** Listening for incoming connections ***
*** Connection from ('192.168.56.1', 57831) has been established! ***
2
[-] Virtuous mode
[##] Key =  XfXi}qK}j+
```

C:\WINDOWS\py.exe

CLIENT:

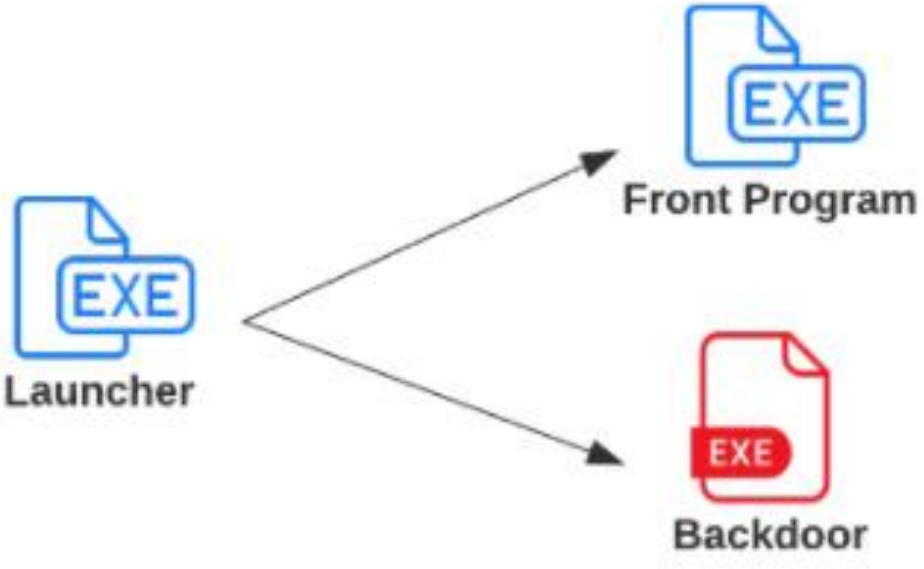
```
192.168.56.1
Enter 1 to run in malicious mode or 2 to run in virtuous mode: 2
[-] Virtuous mode enabled:
Enter the Given Key: XfXi}qK}j+
KEYS MATCHED - PAIRING SUCCESSFUL
entered loop
```





Example Command: “-EpIV” to run StarWars Asciimation on Target device

Backdoor on the victim's machine



THE GENERATED EXECUTABLES

- PyInstaller used to generate a packaged executable
- Executable then taken from maze game
- Create a script which launches the front program(Maze) and a threaded daemon for the client side of the RAT
- Create an executable from this script.
- All 3 executables needed on client device.
- Need to be in designated file paths specified in launcher.py

DEPENDENCIES

D0_Server_31.py

socket
sys
os
time
random
string
cv2
Zipfile

D0_Wclient_31.py

socket
Sys
subprocess
Os
Platform
threading
Time
Datetime
Cv2
mss
Zipfile
Pprint
Schedule
Smptlib
Pynput
pyperclip

D0_Wclient_31.exe

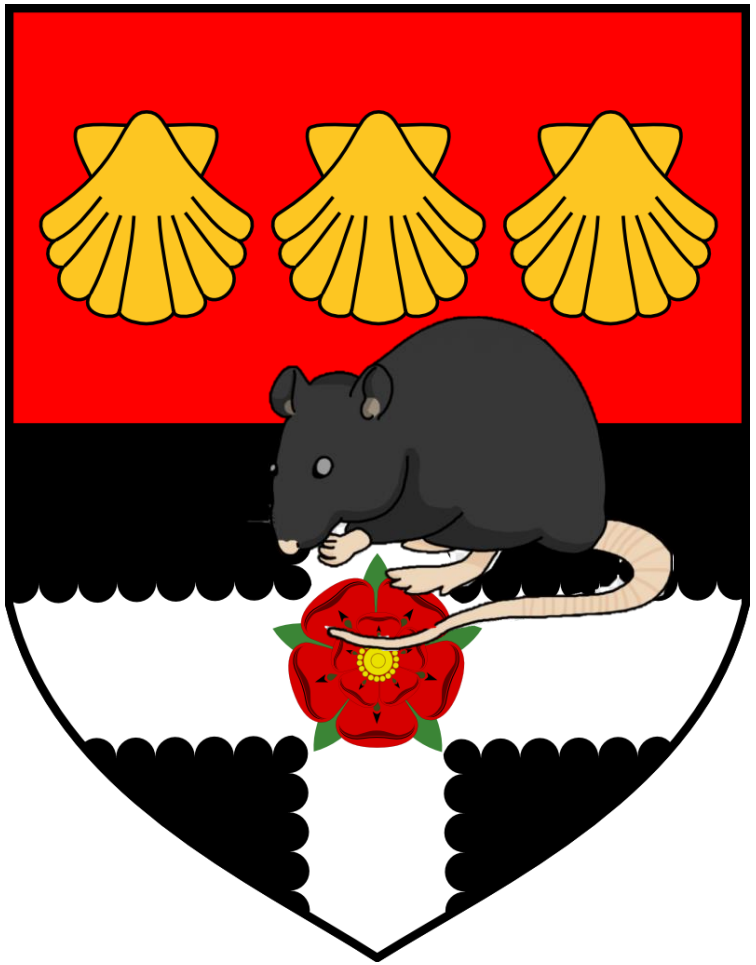
NONE – Not even python needed

When Converting to an .exe using Pyinstaller, all dependencies can be included meaning the program will run standalone.

Launcher.py

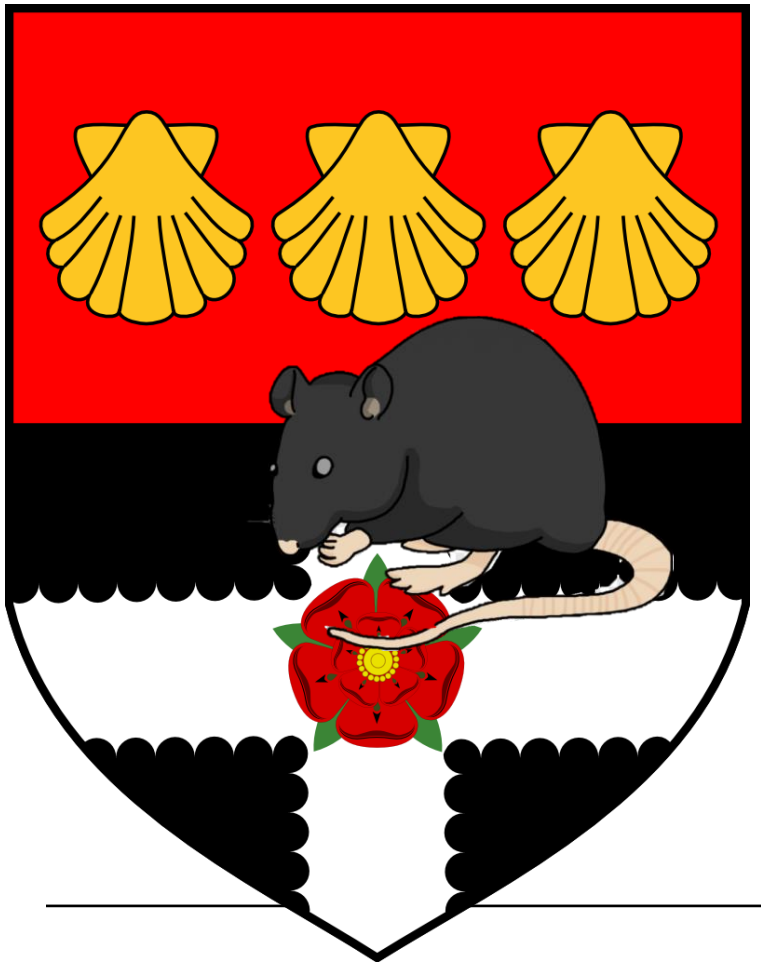
Sys
Os

WHAT DOES UORAT OFFER?



- Lightweight,
- Fast,
- OpenSource,
- Free,
- Easy to use,
- Easy to modify and expand
- Simple / effective

WHAT DOES UORAT OFFER?



- Lightweight,
- Fast,
- OpenSource,
- Free,
- Easy to use,
- Easy to modify and expand
- Simple / effective

SUPPLEMENTARY SLIDES:

WHO AM I?

- Daniel Broomhead
- 27016005
- R. U. Hacking? Society President
- BCS Berkshire Student Chapter President
- BCS Berkshire Early Careers Advocate
- BCS Open Source Specialist Group – Young Representative
- On track for BSc(Hons) Computer Science – First Class

<R. U. Hacking?/>

ABOUT THE DEMO:

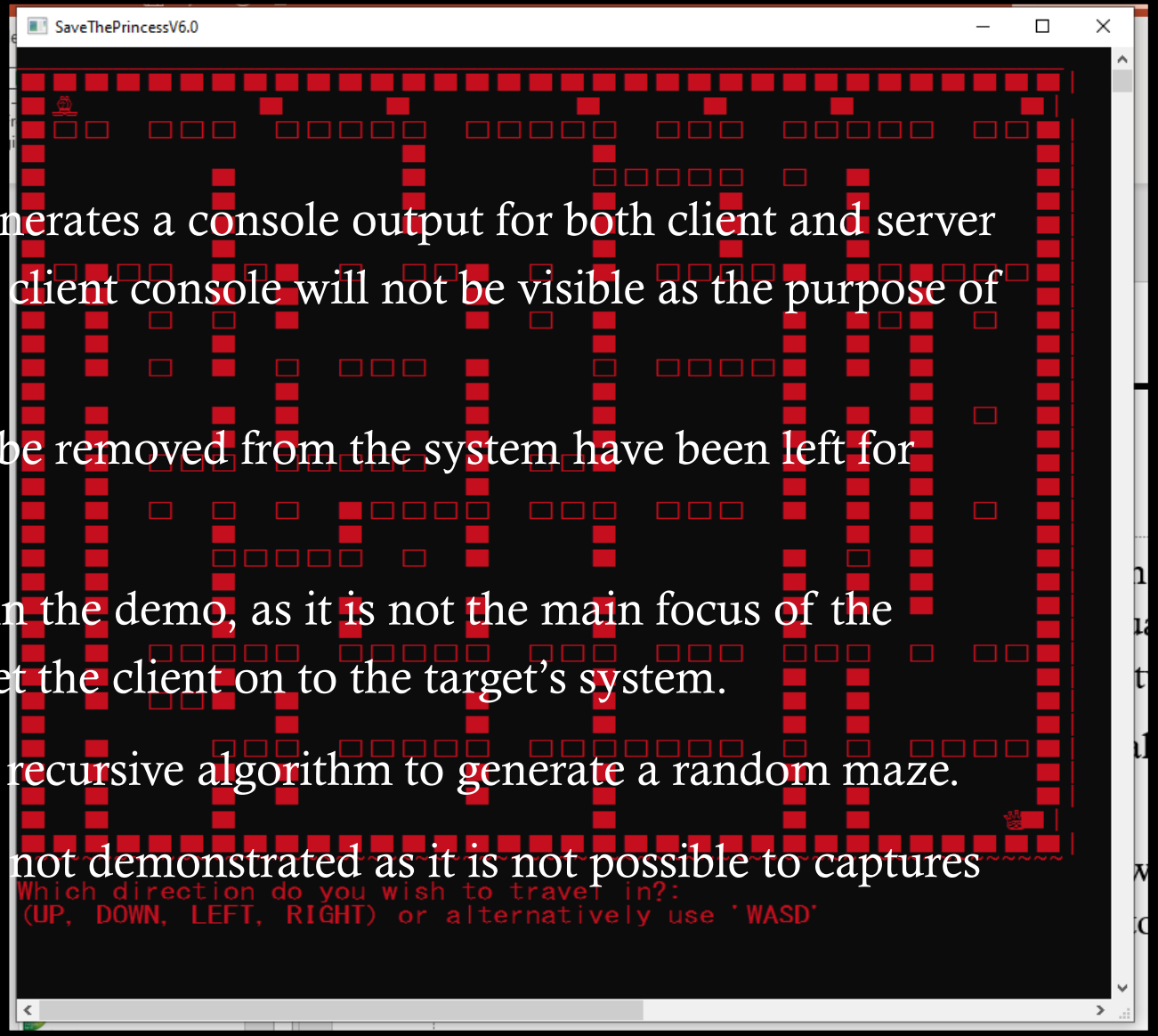
In the demo – the python script is used which generates a console output for both client and server is used. When used in a real world situation, the client console will not be visible as the purpose of this program is designed to be virtually invisible.

For similar reasons, files which would normally be removed from the system have been left for viewing purposes.

The maze front end program will not be shown in the demo, as it is not the main focus of the project, and more of a disguise and method to get the client on to the target's system.

The maze program is created in C++ and uses a recursive algorithm to generate a random maze.

The Shutdown, lock and restart features are also not demonstrated as it is not possible to captures these with screen recording software.



GLOSSARY & TERMS:

RAT : Remote Access Trojan / Remote Administration Tool
TCP : Transmission Control Protocol
UDP : User Datagram Protocol
OOP : Object Oriented Programming
OOD : Object Oriented Design
FTP : File Transfer Protocol
UI : User Interface
GUI : Graphical User Interface
RDP : Remote Desktop Protocol

CONCLUSION:

FUTURE FUNCTIONALITY & IDEAS:

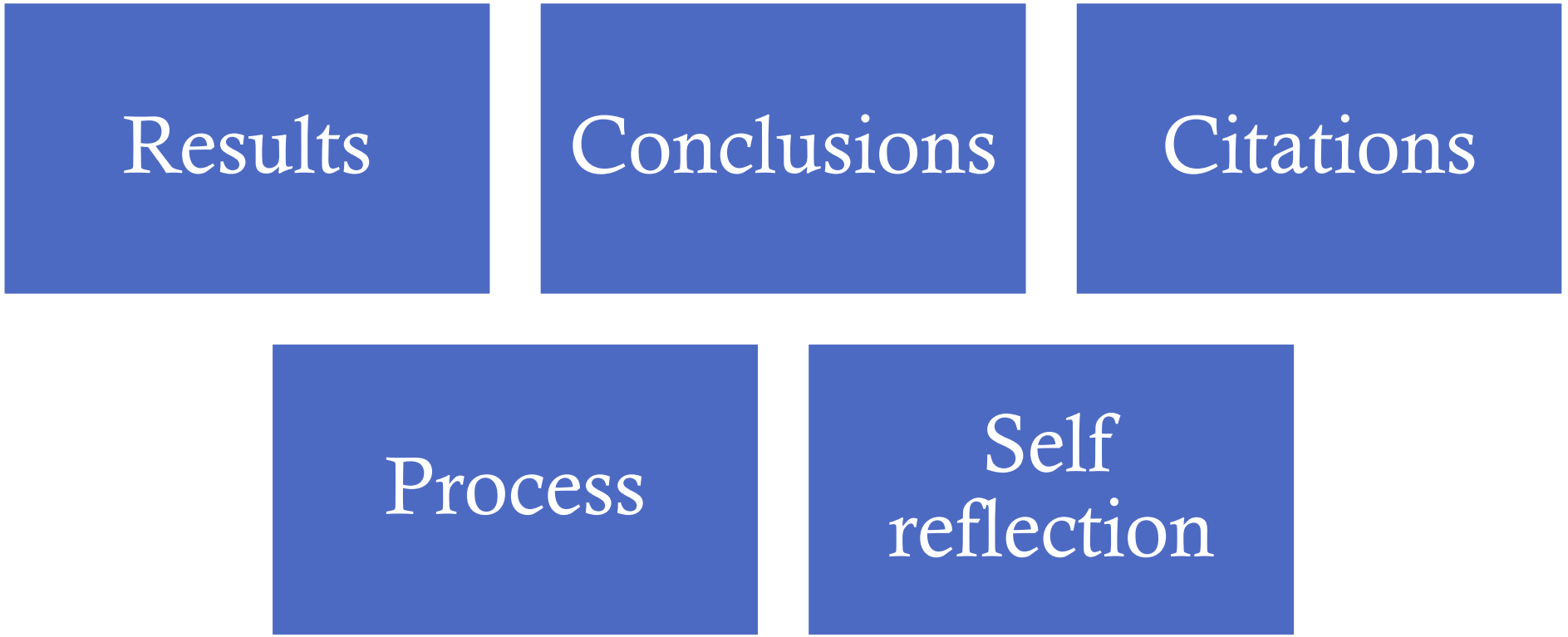
- Graphical user interface : either fully interactive or a graphical design made in console using ASCII art.
- More effort into aesthetics
- Time out feature: Session terminates after extended period.
  - Only required on virtuous version
- More emphasis placed on the Virtuous style software
- More research into RDP
  - Other Desktop Protocols
- More support for other operating systems
- EVEN MORE FUNCTIONALITY

I would have also liked to have done more research into the delivery of the system; however, this would require much hardware and tools which were not accessible due to the COVID-19 Pandemic.

WHAT NEEDS TO BE DISCUSSED:

- Introduction, methodology, implementation, results
- Background, Objectives, Algorithm, Scalability, applications

STRUCTURE & CONTENT QUALITY



TECHNICAL CONTENT

- Clear scope
- Purpose of the application / program
- Logical navigation
- Clarity of explanation
- Innovative functions and features

FUNCTIONALITY & PERFORMANCE

- Results correctness toward the intended purpose
- Appropriate UI
- Visualisation
- Algorithms
- Reliability
- Completeness