Problem Definition:

What is a Rat :

Why:

Purpose:

Objectives:

Clear Scope

What did I plan to do

How did I go about doing it

Results:

What did I achieve:

Features and functionality

Originality

Conclusions:

Result Correctness towards intended purpose

Appropriate AI

Visualisation:

Algorithms

Reliability

Completeness

“Investigating the parallels between using a RAT Style Software for Malicious Purposes and Virtuous intent”.

To understand the scope of this project first you must be able to understand what a RAT is and what it implies.

RAT is an acronym, with two meanings distinguished by the context and purpose in which it was designed. The two meanings are: Remote Administration Tool, and Remote Access Trojan, while at heart the functionality of these two RATs are virtually identical, the different expansions represent the intent for how the software should be used. With further development of both these ideas, comes features exclusive to them – and features designed to enhance the experience.

Generally, a Remote Administration tool has moral and true intentions whereas a Remote Access Trojan is designed with baneful objectives.

RAT software is used for many reasons, the most common principled reason is as a support utility tool, this is quite commonly used by tech-support team in most major companies and establishments to provide remote support. Another common use is for file transfer from a remote location, or even remote access to a system, a common necessity induced by the COVID-19 Pandemic. RAT software usually makes up a smaller part of a larger system. This is also true for the less noble variants – which usually are installed with the intent of delivering a payload with even more malicious intent.

A RAT used for malicious purposes is also synonymous with the term “Back Door”, and is quintessential in most compromising attacks. here is a definition given by Virus Radar.

Here are some of the more famous RAT-style Software ranging from some of the most famous malicious trojans, to some of the industry standard support utility software including remote desktop style software. These can be written in nearly any programming language, with some of the ones shown here being Delphi, Python, java and even VBS. There are many factors which affect the choice of language used.