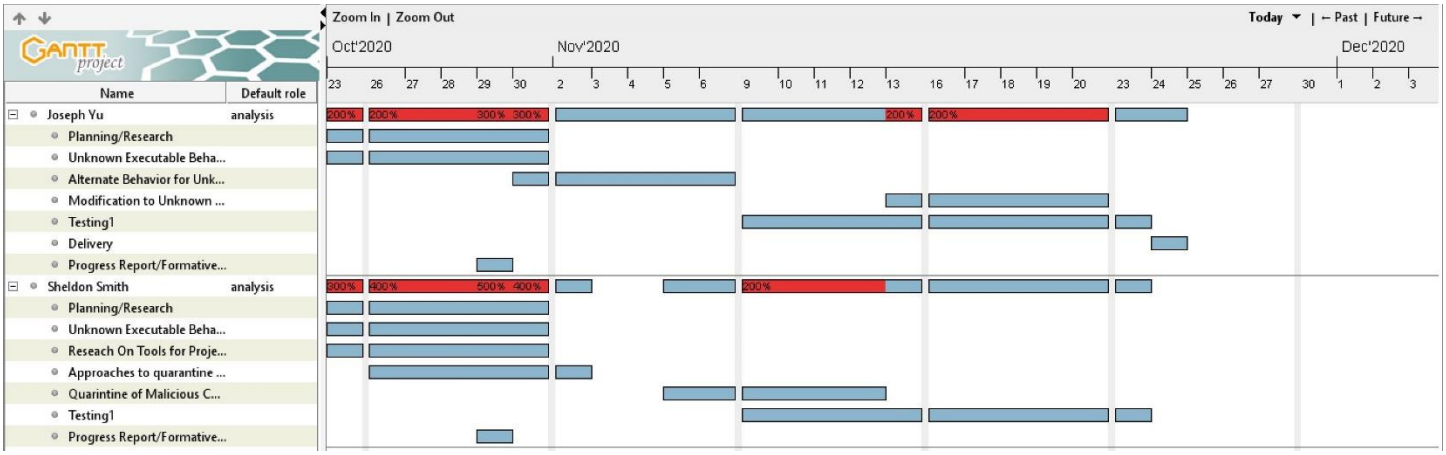
Progress Report

10-28-2020

Team Name: Windows cannot find "Paint"
Members: Sheldon Smith, Joseph Yu

Roles / Responsibilities



Statement of Work

The purpose of this project is to identify what the provided unknown executable ECS.exe does when opened. To aid in our investigation, we will find and use online reverse engineering tools, including disassemblers and decompilers. After using them, we will compile a list with the tools used, in which we describe where to find them, their ease of use, advantages, disadvantages, and lessons learned using the program. Our initial investigation has led us to believe that this executable is either a simple calculator, malicious code, or a mix of both. If the code is a calculator, we will modify is so that is does a different function. If the code is malicious, we will isolate the dangerous code. We will do both modifications if the code is a calculator than has malicious code within it. To complete this project, we will meet both online as well as in person. To safety analyze the executable, we will use a Windows 10 virtual machine so that we do not risk infecting our own systems. We will start the project on October 21, 2020 and will finish by December 1, 2020.

We will submit a Progress Report to our client on October 29, 2020, documenting our initial findings and project plan. We will submit our final executable and final report, and present our findings, on December 1, 2020.

List of Reverse Engineering Tools Used

Ghidra –

This reverse engineering tool is found at https://ghidra-sre.org/. This program was designed by the NSA for public use. Ghidra is intended to analyze compiled code on a variety of platforms, such as Windows, Mac OS, and Linux. Some of the features that are included in this program are disassembly, assembly, decompilation, among others. This tool supports a wide array of processor instruction sets.

Relyze –

This tool was found at https://www.relyze.com/. Similar to Ghidra, this program is a combination of a disassembler and a decompiler. Relyze can convert compiled code into low-level language as well as into pseudocode for a high-level language to the best of its abilities. One unique feature of this tool is the interactive analysis in which it assembles a flow chart of the program that the user can navigate through. This program can also analyze two sets of binary or pseudocode to discover the similarities and differences.

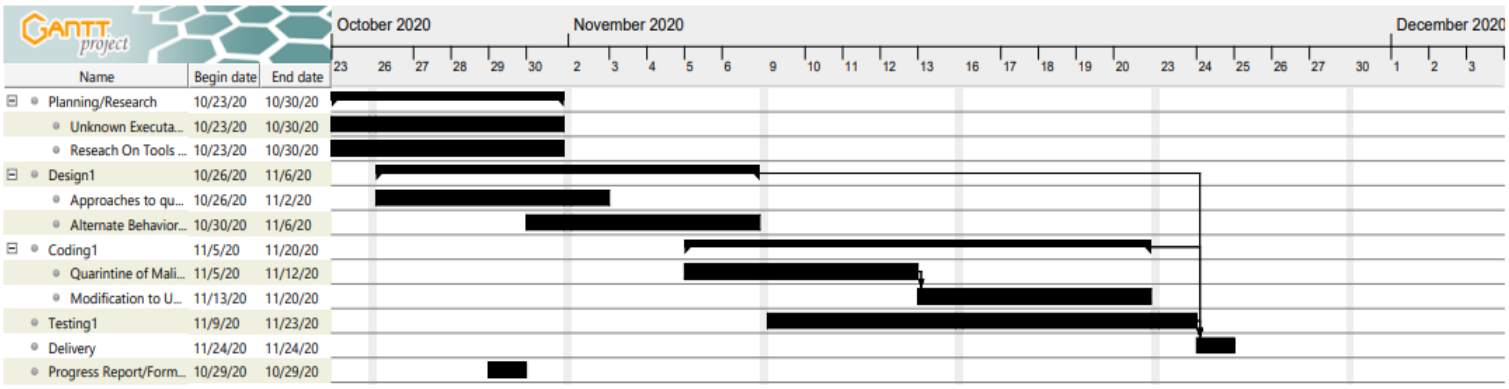
HexEd –

This tool was found at https://hexed.it/. This tool allows the user to look at and modify the hexadecimal of an inputted file. This program shows the offset, hexadecimal code, and the ASCII characters. HexEd allows the user to search for a specific string of hexadecimal codes.

Project Plan

Tasks

Name	Begin date	End date
Planning/Research	10/23/20	10/30/20
Unknown Executable Behavior Research	10/23/20	10/30/20
Reseach On Tools for Project	10/23/20	10/30/20
Design1	10/26/20	11/6/20
Approaches to quarantine malicious segments	10/26/20	11/2/20
Alternate Behavior for Unknown Executable Design	10/30/20	11/6/20
Coding1	11/5/20	11/20/20
Quarintine of Malicious Code Segment	11/5/20	11/12/20
Modification to Unknown Executable Behavior	11/13/20	11/20/20
Testing1	11/9/20	11/23/20
Delivery	11/24/20	11/24/20
Progress Report/Formative Assessments	10/29/20	10/29/20



Preliminary Description of Executable's Behavior

This unknown executable appears to be a calculator with multiple options to select from that also has malicious code intermixed with the main code. There are multiple stages of the malicious code. Upon execution in a Windows 10 VM, the program first starts a repetitive cycle of opening a YouTube video 19 times on both Microsoft Edge and Google Chrome. The program then shows the menu options, prompting the user for input (example: "1.", "2." Etc.) The calculator section contains 10 mathematical functions options and one guessing game mode for the user to select. Upon entering the valid instructions the program will then open the programs Outlook, Paint, Photos, Notepad, along with html page of corndogs floating on Microsoft Edge and Google Chrome on the user's computer that loops.

Github Repository

CSI-2334-Group-Project

https://github.com/legomansps/CSI-2334-Group-Project