

Project Specs and Project Plan

**Project supervisor:** Joseph KEHOE

Fionn Finane | Final Year Project | 20/11/2024

**Table of Contents**

[**Project Specification** 1](#_Toc184050795)

[**Metrics** 2](#_Toc184050796)

[**Project Planning** 3](#_Toc184050797)

[**Requirements** 3](#_Toc184050798)

[**Major Milestones** 3](#_Toc184050799)

[**Timeline Planning** 4](#_Toc184050800)

[**Exceptional requirements** 4](#_Toc184050801)

[**Conclusion** 4](#_Toc184050802)

# **Project Specification**

**Defining the project**

**What is the project supposed to be?**

Here, we plan to develop a Malware Analysis Report Generation tool to detect if any possible malware sample is indeed malicious and extract all its file contents.

**What will the project delivered?**

To provide a practical, comprehensive and intuitive software tool that can effectively deconstruct any malware sample from different operating system architectures and binary file formats and identify key features we would expect from a malware analysis tool, with the additional element of generating a realistic malware report that can accurately mimic professional malware analysis report.

The project will require analysing strings that include the standard string types of URL, Base64, ASCII and Unicode.

Generate a hash of any given file from a possible malware sample.

Calculate the File Entropy of all files to determine if a file is packed or unpacked by how high or low its generated entropic value is.

The project can produce a PDF report exported by generating our report in markdown format, which can then be transformed into a PDF report.

The project can request VirusTotal for further analysis utilising VirusTotal’s latest version of its application programming interface or API.

**Who is going to be using this?**

The target audience for this project is Malware Analysts, as this tool is a dual-functional malware analysis tool capable of producing malware analysis reports.

## **Metrics**

**How will we gauge if the project is successful or not?**

If the project can reproduce near-professional malware analysis reports to operate additionally as a malware analysis tool.

Suppose the project fulfils all the mandatory requirements based on the overall project requirements.

It can also successfully fulfil some of the exceptional requirements, which will all be mentioned in the project planning.

**Is there a precedent for this application?**

Well, some malware analysis tools are GitHub projects.

One of them is Pyew, an exclusive binary analysis tool that analyses Portable executable formats, Executable and Linkable Format, Executables and Dynamic Link Libraries but not compatible with multiple architectures, just Linux Distributions and Windows Operating but does not support the Mac-O or Macintosh OS.

Another that is utilising as the backbone for this project and the product itself is a binary analysis toolkit that is multi-architecture and can analyse all binary formats mentioned above in the Pyev GitHub tool, except it supports multiple architectures, including MAC-O format, a variety of multiple binary analysis techniques that is unique, this tool is an open-source project called Angr or Angr.io.

**How does it differ?**

It will have the features of angr, but we will have the additional elements of calculating file entropy, which is a feature that allows you to export your report into any format you choose and the fact this tool will be analysing all information from given binary formats derived from the different possible malicious samples to gather enough information and some additional technical information by submitting our malware sample to VirusTotal but all so it can at the end create a professional malware analysis report.

# **Project Planning**

In this section, project planning, we will outline the essential compounds that will determine the success of our project.

We are including and defining the requirements for both software and hardware, establishing major milestones in chronological order to identify any exceptional requirements that may arise during the project lifecycle.

## **Requirements**

Hardware: 4GB RAM and a minimum of 2-5GB storage.

Software: Integrated Development Environment such as Visual Studio Code or IDLE Python.

Upgraded to the latest version of Python or installed with the pip command or installed through the official Python website.

Install and import Angr through pip and download all its essential libraries.

Ensure you have imported the important Python libraries you will need for the functionalities of this tool to work correctly.

These are Hashlib, SciPy, NumPy, markdown-pdf, VirusTotal-Python, Urllib, String, Markdown, Matplotlib, and finally, OS, System, Maths, and Regex Module if you want to use this instead of the String module and Base64 module can be utilised if you do not want to utilise the Strings module.

## **Major Milestones**

We will work backwards from the final presentation date to effectively plan our project.

Here is a rough outline of the major milestones:

* + **Final Presentation Round 1**
	+ **Project Research Document**
	+ **Project Specifications and Project Planning**
	+ **Presentation Deck**
	+ **Project Report**
	+ **Final Presentation Round 2**
	+ **Final Installation and Testing:**
	+ **Development Completion:**

## **Timeline Planning**

Using the final presentation date as a reference, we will establish a timeline that includes all major milestones. The following table outlines the milestones and their respective timelines:

|  |  |  |
| --- | --- | --- |
| Milestone | Date | Duration Before Final Year Project Showcase |
| Final Presentation Round 1 | 03/12/24 | 1 weeks |
| Project Research Document | 11/11/2024 | Completed – 2 weeks ago |
| Project Specifications and Project Planning | 02/12/2024 | 3 days |
| Presentation Deck | 13/01/2024 | 6 weeks |
| Project Report | 26/04/2024 | 3 months |
| Final Presentation Round 2  | 01/01/2024 | 4 weeks |
| Final Installation and Testing | 01/02/2024 | 2 Months |
| Development Completion | 01/03/2024 | 3 Months |

## **Exceptional requirements**

The project is compatible with multiple platforms, mainly Windows and Linux.

Implement Flask Web Framework to transform our Python tool with a user-friendly interactive GUI without needing a library like Tkinter to create our GUI.

To determine the most likely malware type based on its unique malware signatures or traits with a malware classification system integrated within the Python malware analysis tool and generated malware analysis report.

We can integrate Wireshark to inspect the traffic and examine the malware’s effect on the network traffic, or we can integrate the Python project known as Scapy for web security auditing purposes.

## **Conclusion**

This project planning document serves as a foundational guide for our project.

By clearly defining requirements, establishing milestones, and identifying exceptional needs, we can ensure a structured approach to project management that leads to successful outcomes.