



RE PREPROCESSOR

Jacob Greenberg

Gary S. *industry advisor*

Dr. Justin Zhan *faculty advisor*



OVERVIEW

Reverse engineering is a technique used to identify the purpose of a compiled or obfuscated program. It is frequently used in vulnerability research and malware analysis.

This project seeks to simplify common RE workflows, such as file identification and extraction. The goal is to allow reverse engineers to spend more time investigating a program, and less time getting it into a workable state.



GOALS

- Streamline the process of preparing a file for reverse engineering
 - Users should be able to quickly identify and extract compressed files
- Employ modularity to make sharing and collaboration easier
 - Users can make their own addons to the tool to make it more effective for them
- Enable users to build their own tools when existing ones don't work for them
 - If the tool cannot identify a file users should be able to make their own addon to identify it



INTELLECTUAL MERITS

This program seeks to solve a number of nuanced issues:

- File formats are often widely varied
 - Some formats use consistent headers while others are far more dynamic
- Not all file formats are documented
 - Many manufacturers intentionally use unusual formats to make reverse engineering more difficult
- Vulnerability researchers may face difficulties sharing work
 - Vulnerabilities in commercial software are sensitive issues and companies are often reluctant to allow researchers to share their work

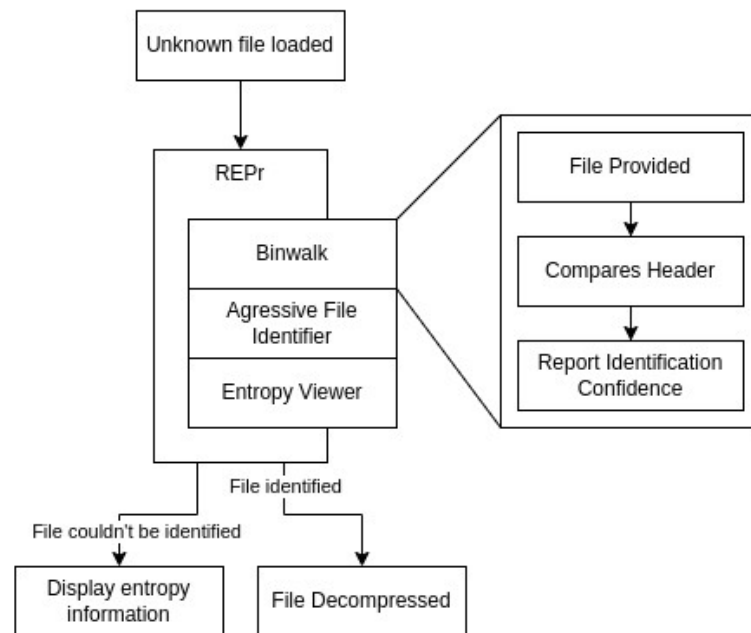


BROADER IMPACT

Vulnerability researcher is a critical part of a good cybersecurity posture. By making VR faster and easier this project stands to assist in enhancing the security of everyday applications.

DESIGN SPECS

D1: Plugin Operations



The tool supports three types of plugins: identifiers, extractors, and informational. These plugins work in conjunction to provide the operator with information about the file they are dealing with and suggested next steps. The general usage path is:

1. User loads an unknown file
2. The program loops over all identifier plugins
3. After identifying a file an extractor is chosen to decompress the file
4. Information about the file is displayed if it cannot be identified



TECHNOLOGIES

- This project makes use of Python, an industry standard interpreted programming language. This language was selected because it is user friendly and makes it quick to write new modules.
- A plugin architecture makes development of new features quick and independent
- Integration with popular disassemblers makes this tool easy to use for reverse engineers



MILESTONES

- Project begins (September 2025)
- Plugin functionality (October 2025)
- Basic identifiers and extractors (December 2025)
- Informational plugins (January 2026)
- Integration with disassemblers (February 2026)
- Additional plugins (February/March 2026)
- Expo (April 2026)



RESULTS

- As of writing, the project has achieved the basic goals it sought out to accomplish, users can identify files in a convenient and modular plugin system
- I plan to continue adding more plugins along with implementing integrations with common disassemblers
- I also plan to develop a demo for the expo which clearly demonstrates the advantages of this tool.



CHALLENGES

This project required me to use aspects of Python I didn't have a lot of experience with

- I had never created a packaged Python project before and need to learn how to build and export packages
- Many file formats are poorly documented which made making identifiers difficult
- The nature of the work this tool is targeted for made it difficult to get feedback