MALWARE DETECTION TOOL

Here's a simple Python script that performs basic static analysis to detect potential malware. This script uses the pefile library to inspect the Portable Executable (PE) format of Windows executable files and yara-python to apply YARA rules for detecting malware signatures.

INSTALLATION

- **Install pefile**:

  $ pip install pefile

- **Install yara-python**:

  $ pip install yara-python

YARA Rule:

Create a file named rules.yar with the following content. This rule is very simple and checks for common strings found in many malware samples:

```
rule MalwareExample
{
    strings:
        $a = "malicious_string1"
        $b = "malicious_string2"
        $c = "malicious_string3"
    condition:
        $a or $b or $c
}
```

MALWARE DETECTION TOOL

PYTHON SCRIPT

```
import pefile
import yara
import os

def check_pe_format(file_path):
    try:
        pe = pefile.PE(file_path)
        print(f"{file_path} is a valid PE file.")
        return True
    except pefile.PEFormatError:
        print(f"{file_path} is not a valid PE file.")
        return False

def check_yara_rules(file_path, rules):
    matches = rules.match(file_path)
    if matches:
        print(f"YARA detected malware signatures in {file_path}:")
        for match in matches:
            print(f" - {match.rule}")
        return True
    else:
        print(f"No YARA signatures matched in {file_path}.")
        return False

def main():

# Path to the file to be analyzed
```

MALWARE DETECTION TOOL

```
file_path = "test_file.exe"


# Check if the file exists
 if not os.path.exists(file_path):

    print(f"File {file_path} does not exist.")
    return

 # Perform basic static analysis
 if check_pe_format(file_path):
    # Compile YARA rules
    rules = yara.compile(filepath="rules.yar")

    # Check YARA rules
    if check_yara_rules(file_path, rules):
       print(f"File {file_path} is potentially malicious.")
    else:
       print(f"File {file_path} appears to be clean.")
 else:
    print("Skipping YARA check due to invalid PE format.")

if __name__ == "__main__":
   main()
```

MALWARE DETECTION TOOL

EXPLANATION:

- **Check PE Format**: The check_pe_format function uses the pefile library to verify if the file is a valid PE file.

- **Check YARA Rules**: The check_yara_rules function uses the yara-python library to match the file against defined YARA rules.
- **Main Function**: The main function handles the file path input, performs the static analysis, and prints the results.

USAGE:

1. Place the YARA rule file (rules.yar) and the Python script in the same directory.
2. Replace "test_file.exe" in the main function with the path to the file you want to analyze.
3. Run the Python script:

```
$ python malware_detection.py
```

This script provides a basic framework for malware detection using static analysis and YARA rules. For more comprehensive analysis, you can expand the YARA rules and add more sophisticated checks.