# User Manual for Sandbox Execution Tool

## Overview

The **Sandbox Execution Tool** is a PowerShell-based application designed to execute external executables in a controlled Windows Sandbox environment. This tool allows users to run potentially unsafe executables without affecting the host system, capturing their output for analysis. The tool includes a graphical user interface (GUI) for ease of use.

**Features**

* **Execute External Programs**: Run any executable file in a Windows Sandbox.
* **Output Capture**: Redirects the output of the executable to a specified file.
* **Network Control**: Option to disable network access for the sandboxed application.
* **Read-Only Mode**: Restrict file access to the mapped folder.
* **Timeout Setting**: Specify a timeout for the execution to prevent hanging processes.
* **User-Friendly GUI**: Simple interface for selecting files and configuring options.

**Requirements**

* **Windows 10/11**: The tool requires a Windows operating system with Windows Sandbox enabled.
* **PowerShell**: The tool is built using PowerShell and requires PowerShell to be installed.
* **Windows Sandbox**: Ensure that Windows Sandbox is enabled in Windows Features.

**Installation**

1. **Download the Tool**: Obtain the Execute.ps1 and gui.ps1 files.
2. **Place Files**: Store both files in the same directory for proper execution.
3. **Enable Windows Sandbox**: Ensure that Windows Sandbox is enabled in your Windows Features.

## Usage Instructions

### Running the Tool

1. **Open PowerShell**:

* Press Win + X and select Windows PowerShell.

1. **Navigate to the Tool Directory**:

* Use the cd command to change to the directory where the scripts are located:

cd C:\path\to\your\tool

1. **Launch the Script**:

* Run the script to open the Sandbox Execution Tool. There are two possible ways:

.\Execute.ps1 -file "C:\{{pathToYourFolder}}\{{your\_executable.exe}}" -execParams “—param1 -param2” -output "{{output\_file}}.txt" -hostFolder "C:\{{path\to\your\executableFolder}}" -NoNetwork -timeout 20

Or

 .\ Execute.ps1

* If you execute the script with full parameters, it will automatically perform its tasks. However, if your run it with no parameters at all, a GUI will be triggered to offer user with manual input selection for each required parameter.

### Executing the script manually

A computer screen with blue and white text

AI-generated content may be incorrect.

1. Following the sample syntax to fully execute the script, with required parameters:
   * **-file**: path to the executable
   * **-output**: the output file name
   * **-hostFolder**: the folder to contain the output file
   * **-execParams**: the parameter string that will be used by the executable
   * **-NoNetWork**: flag to disable network use of the script
   * **-ReadOnly**: flag to restrict file access
   * **-timeout**: flag to set the timeout for the script execution

### Using the GUI

1. **Executable Selection**:

* Click the **Browse** button to select the executable file you want to run. The file must be an .exe file.
* Input the corresponding parameter for the executable, left empty if none is required

1. **Output Settings**:

* Specify the name of the output file where the results will be saved (e.g., output.txt).
* Specify the host folder where the output file will be saved. The default is C:\Users

1. **Execution Options**:

* **Disable Network**: Check this box if you want to disable network access for the executable.
* **Read Only**: Check this box to restrict file access to the mapped folder.
* **Timeout**: Set the timeout duration in seconds for the execution (default is 120 seconds).

1. **Execute**:

* Click the **Execute in Sandbox** button to run the selected executable in the Windows Sandbox.

### Example Usage

1. **Select an Executable**:

* Choose C:\path\to\your\executable.exe by click the **Browse** button
* A screenshot of a computer

  AI-generated content may be incorrect.

1. **Set Output File**:

* Enter {{output\_file}}.txt as the output file name.

1. **Set Host Folder**:

* Ensure the host folder is set to an existing folder, in which the output file will be created.
* A screenshot of a computer

  AI-generated content may be incorrect.

1. **Execution Options**:

* Optionally check **Disable Network** and **Read Only**.
* If necessary, add the timeout for the program to execute, the default value is 20 seconds
* A screenshot of a computer

  AI-generated content may be incorrect.

1. **Run the Tool**:

* Click **Execute in Sandbox**. The tool will launch the Windows Sandbox, run the executable, and capture the output.
* The window Sandbox will be open for a brief moment to run the executable in a secured environment
* A screenshot of a computer

  AI-generated content may be incorrect.
* After that it will automatically closed, with a notification window popup
* A screenshot of a computer error

  AI-generated content may be incorrect.
* Finally, the notification about success execution will appear
* A screen shot of a computer

  AI-generated content may be incorrect.

### Output

* After execution, the output will be saved in the specified output file (e.g., C:\Users\920322\Workspace\SandBox\output.txt).
* If the execution completes successfully, a message box will confirm the completion.

### Error Handling

* If any required fields are not filled, or if the specified files or folders do not exist, the tool will display an error message.
* Ensure that the executable and host folder paths are valid before executing.

### Conclusion

The Sandbox Execution Tool provides a safe and efficient way to run executables in a controlled environment, capturing their output for further analysis. By following the instructions in this manual, users can easily utilize the tool for various applications in computer science education and research.