

SECURE ADVANCED SHELL DOCUMENTATION

Written by Pau Santana, 30th July 2023

Table of Contents

1. Introduction
2. Class Definition
3. Class Initialization
4. Command Execution
5. Built-in Commands
6. Command Preprocessing
7. Main Execution
8. Differences between Windows and Linux Versions

1. Introduction

The Secure Advanced Shell is a command-line interface (CLI) that provides a secure environment for executing commands. It is designed to work on both Windows and Linux operating systems. The shell provides a set of built-in commands and also allows execution of system commands in a secure manner.

2. Class Definition

The SecureAdvancedShell class is a subclass of the Cmd class from the cmd module. This module provides a simple framework for writing line-oriented command interpreters. These command interpreters are often useful for test harnesses, administrative tools, and prototypes that will later be wrapped in a more sophisticated interface.

3. Class Initialization

The `__init__` method initializes the SecureAdvancedShell class. It calls the superclass's `__init__` method and initializes an empty list `command_history` to keep track of the command history. This list will store all the commands that the user has entered during the session.

4. Command Execution

The default method is overridden to execute commands in a safer manner. It splits the input line by '&&', and then tries to execute each command separately. If an exception occurs during the execution of a command, it prints an error message and stops executing the remaining commands.

The `run_command` method is used to execute a given command securely. It uses the `subprocess` module to run the command in a separate process and captures the output and errors. If the command execution fails, it raises a `subprocess.CalledProcessError` exception.

5. Built-in Commands

The Secure Advanced Shell provides several built-in commands:

- `exit` or `quit`: Exits the console.
- `history`: Shows the command history.
- `mkdir`: Creates a directory.
- `rmdir`: Removes a directory.
- `touch`: Creates a file.
- `rm`: Removes a file.
- `ls`: Lists files in a directory along with their permissions.
- `clear`: Clears the console.

Each command is implemented as a method named `do_command`, and the help for each command is provided by a method named `help_command`.

6. Command Preprocessing

The `precmd` method is overridden to preprocess the command line before executing it. It converts the command line to lowercase and changes 'quit' or 'exit' to 'exit' to ensure that the exit command works regardless of case.

7. Main Execution

If the script is run as the main program, it creates an instance of `SecureAdvancedShell` and enters the command loop, which will run until the `exit` or `quit` command is issued.

8. Differences between Windows and Linux Versions

The main difference between the Windows and Linux versions of the Secure Advanced Shell is in the way commands are executed. The Windows version uses the `subprocess.Popen` method to run commands in a separate process and handles CTRL+C to kill the process. The Linux version uses the `subprocess.run` method to run commands and automatically captures the output and errors.

The Windows version also includes additional imports for `getpass`, `signal`, and `time` which are not present in the Linux version. This is because these modules are used to handle specific Windows functionalities.

The `do_clear` method in the Windows version uses the command `cls` to clear the console, while the Linux version uses the command `clear`.

License: This document is released under the MIT License.