Analysis Report for: unpacked.au3

Overall Functionality

This AutoIt script creates a simple GUI application that displays information about the running script: its name, process ID, and the MD5 hash of the first chunk of its own executable file. The script optionally takes two command-line arguments: a time delay (in milliseconds) after which the application closes, and a mode flag. If the mode flag is not 2, the GUI is shown; otherwise, it remains hidden, effectively making the script run silently and only provide the MD5 hash. The application closes when the user clicks the "OK" button or when the timer expires (if a time delay is specified).

Function Summaries

* **`_MD5FORFIRSTFILECHUNK($SFILE, $ICHUNKSIZE = 0)`:** This function calculates the MD5 hash of the first chunk of a specified file.
* **Parameters:**
* `$SFILE`: The path to the file.
* `$ICHUNKSIZE`: (Optional) The size of the chunk to hash. Defaults to the entire file size if not provided.
* **Return Value:** A structure containing an error code and, if successful, the MD5 hash as a string.

Control Flow

* **Main Script:**
1. **Initialization:** Sets global constants and initializes variables for time delay (`$TIMEDELAY`) and mode (`$MODE`). Parses command-line arguments to set these variables.
2. **GUI Creation:** Creates a simple GUI window with an edit box (displaying script information) and an OK button. Sets the focus to the OK button.
3. **Conditional GUI Display:** Shows the GUI unless the mode is 2.
4. **Main Loop:** Enters an infinite loop (`WHILE 1`).
5. **Timer Check:** If `$TIMEDELAY` is set, checks if the time limit has been reached using `TIMERDIFF`. If so, exits the loop.
6. **Message Handling:** Uses `GUIGETMSG()` to retrieve GUI messages.
7. **Message Switch:** A `SWITCH` statement handles different messages:
* `$GUI_EVENT_CLOSE`: Exits the loop when the window is closed.
* `$BUTTON1`: Exits the loop when the OK button is clicked.
8. **Loop End:** Continues the loop until an exit condition is met.

* **`_MD5FORFIRSTFILECHUNK` Function:**
1. **Chunk Size Determination:** Determines the chunk size to process.
2. **File Opening:** Opens the specified file using `CreateFileW` (WinAPI call). Handles errors.
3. **File Mapping:** Creates a file mapping object using `CreateFileMappingW`. Handles errors. Closes the file handle.
4. **Mapping View:** Maps a view of the file into memory using `MapViewOfFile`. Handles errors. Closes the file mapping object.
5. **MD5 Calculation:**
* Creates a MD5 context structure using `DLLSTRUCTCREATE`.
* Initializes the MD5 context using `MD5Init`. Handles errors.
* Updates the MD5 context with the file data using `MD5Update`. Handles errors.
* Finalizes the MD5 calculation using `MD5Final`. Handles errors.
6. **Memory Cleanup:** Unmaps the file view using `UnmapViewOfFile` and closes handles.
7. **Return Value:** Returns the calculated MD5 hash or an error code.

Data Structures

* **Global Variables:** The script uses several global variables to store data such as time delay, mode, GUI handles, and command-line arguments.
* **DLLSTRUCT:** The `_MD5FORFIRSTFILECHUNK` function uses `DLLSTRUCTCREATE` to create a structure to hold the MD5 context data, interacting directly with the `advapi32.dll` MD5 functions. This structure is crucial for the MD5 hash calculation process.

Malware Family Suggestion

Given the functionality, this script is not inherently malicious but has characteristics that could be adapted for malicious purposes. The calculation of the MD5 hash of its own file could be used as a form of self-identification or "fingerprint" in a larger malware program. The command-line argument support and ability to run silently (mode 2) suggest it might be used as a component of a more complex malware operation, such as a dropper or information gathering tool. The use of WinAPI calls (especially file operations and MD5) without proper error handling in other contexts would be a strong indicator of malicious intent. On its own, however, it is simply a utility that calculates the MD5 hash of a file. It's important to note that this analysis focuses on the provided code and not hypothetical modifications.