Analysis Report for: 48B1B314B23036AC3156369BAEB9CD75.exe

Overall Functionality

This code is a VBScript (not C as initially stated) that checks the registry for the presence of a specific Adobe Acrobat Reader installation. It then uses the result to set a return code and potentially trigger a system reboot. The script interacts with the Windows registry to read and write values related to Adobe Acrobat Reader and a "Lumension Patch Player." The primary goal appears to be determining if a compatible version of Adobe Acrobat Reader is installed and signaling success or failure to another process (likely a patcher or installer). The inclusion of the `PLCCAgent_` functions strongly suggests integration with the Lumension Patch Player, a vulnerability management software.

Function Summaries

* **`ValidateRegExpression(str, expression)`:** This function checks if a given string (`str`) matches a regular expression (`expression`). It returns `True` if there's a match, `False` otherwise.
* **`GetRegValue(strKeyPath)`:** This function reads a registry value from the specified key path (`strKeyPath`). It returns the value if found; otherwise, it returns an empty string.
* **`Debugger(message)`:** This function displays a message box if the `isDebugMode` flag is set to `True`. Primarily for debugging.
* **`PLCCAgent_InitiateSystemShutdown()`:** This function writes a value to the registry that triggers a reboot by the Lumension Patch Player.
* **`PLCCAgent_SetReturnCode(RetCode, RetDesc)`:** Writes a return code (`RetCode`) and description (`RetDesc`) to the registry for the Lumension Patch Player.
* **`PLCCAgent_PollHost()`:** An empty function; its purpose is unclear without further context.
* **`PLCCAgent_Sleep(Delay)`:** An empty function; presumably intended for pausing execution for a specified `Delay`, though it currently does nothing.
* **`WScript_Quit(RetCode)`:** Writes a return code to the registry and then terminates the script.

Control Flow

* **Main Script Block:** The main part of the script reads configuration settings (Adobe Acrobat version, language code, registry key path, regex pattern). It then calls `GetRegValue` to retrieve the product GUID from the registry, followed by `ValidateRegExpression` to check if it matches the expected pattern. Based on the validation result, it either calls `Debugger("Applicable")` and exits with code 1 (success), or it calls `Debugger("Not Applicable")` and exits with code 0 (failure).
* **`ValidateRegExpression`:** Creates a regular expression object, sets its pattern and case-insensitive flag, uses `re.Test()` to perform the match, and returns `True` or `False` accordingly.
* **`GetRegValue`:** Reads a registry value using `WinShell.RegRead()`. Error handling is included to return an empty string if the key is not found.
* **`Debugger`:** A simple conditional statement; it only executes if `isDebugMode` is `True`.
* **`PLCCAgent_InitiateSystemShutdown`:** Directly writes to the registry to trigger a reboot.
* **`PLCCAgent_SetReturnCode`:** Writes two values to the registry.
* **`WScript_Quit`:** Writes a return code to the registry before terminating the script.

Data Structures

The code primarily uses simple variables (strings, booleans). No complex data structures are employed. The regular expression object is a temporary object used within the `ValidateRegExpression` function.

Malware Family Suggestion

While this code isn't inherently malicious, its functionality raises concerns. The actions of writing registry keys to trigger a reboot (`PLCCAgent_InitiateSystemShutdown`) and setting return codes (`PLCCAgent_SetReturnCode`, `WScript_Quit`) are often found in installers or legitimate software update mechanisms. However, the stealthy registry manipulation and potential for unintended system restarts (if integrated into malicious software) could classify it as a component of a **Dropper** or **Installer** type of malware. A malicious actor could easily modify this script to target different registry keys and potentially install additional malware after the Adobe Acrobat check. The reliance on the Lumension Patch Player for reporting and reboot functionality suggests a potential vector for exploitation of that system's trust. The script itself is not self-replicating or actively harmful, but its components could readily be incorporated into something malicious.