Analysis Report for: Install.cmd

Overall Functionality

This code is a Windows batch script (despite the `.c` extension, it's not C code; it's a `.bat` script disguised as C). It appears to be malicious, designed to replace core system DLLs and potentially compromise the system. The script operates by first checking for the existence of `sppsvc.exe` in either `%Windir%\System32` or `%Windir%\Sysnative`, indicating a 32-bit or 64-bit system, respectively. Then, it proceeds to take ownership and grant full control to "everyone" for several critical system DLLs and executables (`slwga.dll`, `sppwmi.dll`, `systemcpl.dll`, `user32.dll`, `winlogon.exe`, `winver.exe`), but restricts access to `sppcomapi.dll`, `sppuinotify.dll`, and `sppsvc.exe` for specific users ("LOCAL SERVICE" and "users"). It then renames these original files, copies replacement files (from `data\x86T` or `data\x64T` subdirectories depending on the system architecture) into their place, and finally moves the renamed original files to the temporary directory. The script also targets `wlms.exe`, denying everyone access to this file. The script interacts with `mcbuilder.exe` and uses Visual Basic Script files (`Pleasewait.vbs`, `Successfuly.vbs`) to display messages to the user. The use of `ping` commands create artificial delays. The overall effect is a stealthy DLL replacement, strongly indicative of malware activity.

Function Summaries (Note: There are no functions in a batch script; it's a sequence of commands.)

The code consists of a series of commands, not functions in the C or any other programming language sense. Each command performs a specific operation within the operating system. Examples include:

* `takeown`: Takes ownership of a file or folder.
* `icacls`: Modifies access control lists (ACLs) for files and folders.
* `ren`: Renames files.
* `copy`: Copies files.
* `move`: Moves files.
* `rd`: Removes directories.
* `md`: Creates directories.
* `taskkill`: Kills processes.
* `start`: Starts a process.
* `call`: Calls another batch file or executable.
* `ping`: Sends ICMP echo requests.
* `if exist`: Conditional statement checking for file existence.
* `goto`: Jump to a different label in the batch script.

Control Flow

The control flow is primarily determined by conditional statements (`if exist`) and `goto` statements, creating a branching structure.

1. **System Detection:** The script first checks if `sppsvc.exe` exists in `System32` or `Sysnative` to determine the system architecture (32-bit or 64-bit). This determines which branch of the code is executed.

2. **File Manipulation:** Regardless of the system architecture, the script performs the following steps in the appropriate system directory (`System32` or `Sysnative`):
- Takes ownership of and modifies permissions for target files.
- Renames the original system files, appending ".vvv".
- Copies replacement files from `data\x86T` (32-bit) or `data\x64T` (64-bit) directories.
- Moves the renamed original files to a temporary directory.

3. **`wlms.exe` Handling:** The script further modifies the permissions for `wlms.exe` if it exists, denying access to everyone.

4. **Process Termination and Message Display:** The script kills the `wscript.exe` process (likely used for the initial "Please Wait" message) and then displays a "Successfuly" message using a VBScript.

5. **Cleanup:** Finally, it cleans up by removing the temporary directory.

Data Structures

There are no explicit data structures used in the sense of C arrays or structs. The data is managed implicitly through the file system and environment variables (`%Windir%`, `%Temp%`, `%Random%`). The `data\x86T` and `data\x64T` directories act as implicit structures, holding the replacement DLLs.

Malware Family Suggestion

Given the functionality, this script strongly suggests a **rootkit** or **DLL hijacker** type of malware. The stealthy replacement of core system files, coupled with permission modifications to restrict access to the original files, is a hallmark of these types of malware. The modification of permissions and use of `mcbuilder.exe` suggests an attempt at code signing manipulation, to further increase the malware's likelihood of avoiding detection. The temporary removal and replacement of critical files also indicates that this malware aims for persistence, to reinfect the system upon reboot. The use of VBScripts for simple user interaction and display of messages is characteristic of many less sophisticated malware families.