Analysis Report for: new 1.txt

Overall Functionality

This code appears to be a PowerShell script disguised as C code. The null characters () interspersed throughout the "C code" are a strong indicator of obfuscation. The actual script aims to install and configure the Windows Subsystem for Linux (WSL), potentially downloading a Ubuntu distribution. It also attempts to hide certain executable files and then launches two executables, likely malicious, named `Emblink.exe` and `Embfont.exe`. The script prioritizes user profile directories over system directories for executable placement and execution. This behavior is highly suspicious and points toward malicious intent.

Function Summaries

The code doesn't contain C functions in the traditional sense. Instead, it uses PowerShell cmdlets, which are presented within the obfuscated "C code" as if they were function calls. The core logic is structured around these cmdlets. The only thing resembling a function is `Check-WSLCompatibility`, which checks if WSL and the Virtual Machine Platform are enabled.

* **`Check-WSLCompatibility`**: This function (PowerShell cmdlet) checks the system's WSL and Virtual Machine Platform feature statuses. It returns `$true` if both are enabled and the OS build is 2004 or later; otherwise, it returns `$false`. It uses `Get-WindowsOptionalFeature` to retrieve feature information and `Get-ItemProperty` to get the OS build number.

Control Flow

The script's control flow is primarily determined by conditional statements (`if`) that check for the existence of files and the status of WSL and the Virtual Machine Platform.

* **Main Script Flow:**
1. **Enable WSL and Virtual Machine Platform:** It attempts to enable WSL and Virtual Machine Platform using `DISM`.
2. **Download Ubuntu:** If `Ubuntu.tar.gz` isn't found in `%LOCALAPPDATA%`, it downloads it from a suspicious URL.
3. **Check WSL Compatibility:** Calls the `Check-WSLCompatibility` function to verify WSL and Virtual Machine Platform are enabled. This is crucial for the next stage.
4. **Extract and Install Ubuntu:** Extracts the downloaded `Ubuntu.tar.gz` using `gzip` and executes the WSL installation command (`wsl --import`).
5. **Hide Executables:** Sets attributes for `Emblink.exe` and `Embfont.exe` to Hidden and System. This is a clear attempt at stealth.
6. **Run Executables:** Launches `Emblink.exe` or `Embfont.exe` based on their presence in either the user's or system's temporary directory (`%LOCALAPPDATA%\Local\Temp` and `%SystemRoot%\Temp`).
7. **Sleep:** Waits for 30 seconds using `Start-Sleep`.

* **`Check-WSLCompatibility` Flow:**
1. **Check WSL Status:** Checks if WSL is enabled.
2. **Check Build Number:** Checks if the OS build number is greater than or equal to 2004.
3. **Check Virtual Machine Platform:** Checks if the Virtual Machine Platform is enabled (only if the previous checks passed).
4. **Return True/False:** Returns `$true` only if all conditions are met; otherwise, returns `$false`.

Data Structures

The script uses arrays for storing file paths to be hidden (`$filesToHide`) and utilizes strings for file paths, URLs, and other variables. No complex data structures are employed.

Malware Family Suggestion

Given the script's functionality—installing WSL (a potential smokescreen), downloading a file from a suspicious URL, hiding malicious executables, and then launching them—this strongly suggests a **downloader/dropper** malware family. The script downloads and executes additional malicious payload(`Emblink.exe` and `Embfont.exe`), and the download URL itself is highly suspicious. The obfuscation technique further reinforces this classification. It might also be categorized as a **backdoor** if `Emblink.exe` and `Embfont.exe` establish persistence or provide remote access capabilities. More analysis would be required on the downloaded executables to determine the specific malware family.