Analysis Report for: 16C52C02E244FCEFA97AB4CEEFACDBFF.exe

Overall Functionality

The provided code is a heavily obfuscated batch script disguised as C code. It's not valid C code and will not compile. The actual code is a series of `set` commands in a batch script, defining variables containing parts of a command line. These parts are then concatenated and executed, resulting in the execution of a malicious PowerShell command. The goal is to download and execute a malicious payload from a remote server. The obfuscation techniques used include:

* **Encoding of variable names:** Variable names utilize characters from various Unicode ranges (Cyrillic, Armenian) making them hard to read and understand.
* **Comments:** Meaningless comments are scattered throughout to add noise and make analysis more difficult.
* **Unnecessary variable assignments:** The script creates many variables that only contribute to the obfuscation.

The final executed command likely downloads a file (`.exe`), makes it hidden, and then executes it. This strongly suggests malicious intent.

Function Summaries

There are no functions in this code. It's a batch script, not a C program. The structure mimics a C program to confuse static analysis tools.

Control Flow

The code has a linear control flow. There are no loops or function calls in the traditional C sense. The "control flow" is entirely determined by the order of the `set` commands and the final concatenation and execution of the resulting string.

The script proceeds as follows:

1. **Variable definitions:** Multiple variables are defined using `set`, with seemingly random names and values which are fragments of a malicious command.

2. **Command concatenation:** The script uses the defined variables to build a single large string representing a PowerShell command.

3. **Command execution:** The concatenated string, which is the actual malicious payload, is then executed using the `%...%` syntax (batch script command substitution). Note that the `%...%` wraps the entire constructed command.

4. **Additional Attrib command:** A second command, using `attrib`, modifies the attributes of the downloaded file, likely to hide it from the user.

5. **SET command:** A SET command (confusingly still in batch) is used to further confuse the static analysis and does not contribute to the malware functionality itself.

6. **Exit command:** The script ends with `exit`, which terminates the batch process.

Data Structures

There are no data structures used in the traditional C sense (arrays, structs, etc.). The script uses only simple string variables in a batch context. The variables themselves are essentially acting as a rudimentary form of data structure to store and manipulate pieces of the malicious command.

Malware Family Suggestion

Based on the functionality, the code strongly resembles a **downloader** or a **dropper**. It downloads a further payload from a remote location. The downloaded payload might be any type of malware, but the dropper's behavior points to a generic malware infection strategy focused on stealth and execution of an arbitrary payload. Further analysis of the downloaded executable would be necessary to determine the specific malware family. The obfuscation level suggests an attempt to avoid detection by antivirus software.