Analysis Report for: AF6CF4D9B4F0CC6C4FC871D860CD6091.vbs

Overall Functionality

This C code snippet is a malicious script injector, likely part of a browser extension or other malware. It injects a JavaScript file (`./js/injectors/after_inject.js`) into a webpage by creating a new script element and appending it to the document's head or body. The code heavily obfuscates its functionality through:

* **Variable renaming:** Using nonsensical names like `a14_0x236d89`, `a14_0x3fbb`, etc.
* **String encoding:** The strings used are likely encoded, using an array (`a14_0x37be`) for look up of string values.
* **Function obfuscation:** The code utilizes self-executing anonymous functions and closures to hinder readability.

Function Summaries

* **`a14_0x37be()`:** This function acts as a string encoding/decoding mechanism. It returns an array of strings which are likely used to avoid detection. It uses a self-executing function to return the array.

* **`a14_0x40542d()`:** This is a closure that creates a function that can only be called once. This is likely used to prevent multiple injections of the same script.

* **`a14_0x1e9ffa()`:** This function modifies the `console` object of the webpage. It overwrites the standard logging functions (`log`, `warn`, `error`, etc.) with custom functions, likely to suppress or alter logging attempts during the injection or execution of `after_inject.js`.

* **`a14_0x3fbb()`:** This is a self-modifying function serving as a decoder for the string array returned by `a14_0x37be()`. It takes an index and returns the corresponding decoded string from the array.

* **Unnamed self-executing function:** This function contains the main logic of the injection process. It uses `a14_0x3fbb` to decode strings, and performs the actual script injection.

Control Flow

* **Unnamed self-executing function:** This function starts by initializing a variable `_0xf63126` through a series of arithmetic operations on decoded string values from `a14_0x3fbb`. The result is then compared to a hardcoded value `0x39759`. This seemingly meaningless calculation is probably a way to obfuscate a simple conditional check. If the condition is false, the array `_0x3111b7` (initially populated by `a14_0x37be()`) undergoes a circular shift using `push` and `shift`, which acts as a simple obfuscation technique, likely not having any practical effect on the code's behavior. The loop continues until `_0xf63126` matches `0x39759`.

* **`a14_0x1e9ffa()`:** This function iterates through an array of console methods (`log`, `warn`, etc.). For each method, it creates a new function and overwrites the original console method with this new function. The purpose is to intercept and potentially modify console outputs.

Data Structures

* **`a14_0x37be()`'s return value:** An array of strings used for obfuscation, representing various JavaScript commands and API calls (e.g., 'console', 'log', 'createElement', etc.).

* **`_0x3111b7`:** An array used in the obfuscated loop in the main self-executing function.

* **`_0x1d24d4`:** An array holding the names of console methods that are overwritten.

Malware Family Suggestion

Based on its functionality, this code is strongly indicative of a **JavaScript injector**, commonly used as part of a larger malware campaign. Its use of obfuscation techniques suggests an attempt to evade detection by antivirus software and security tools. The injection of an external script (`after_inject.js`) implies that further malicious actions will be performed by that secondary script. It's difficult to pinpoint a specific malware family without analyzing `after_inject.js`, but it could be associated with various types of malware, such as browser hijackers, information stealers, or click fraud tools.

Security Implications: This code is highly dangerous. The injection of arbitrary JavaScript code into web pages can lead to various attacks, including:

* **Data theft:** Stealing cookies, passwords, or other sensitive information.
* **Session hijacking:** Taking over user accounts.
* **Phishing:** Redirecting users to fake websites.
* **Malicious code execution:** Running arbitrary code on the victim's computer.

It is crucial to avoid encountering and executing this type of code.