Analysis Report for: E5DEBA5E5F7DB54992F12CDF34985BE4.exe

Overall Functionality

This C code is obfuscated and likely malicious. It appears designed to interact with a specific website (HDFC Netbanking) to automate login and potentially financial transactions. The code uses a self-modifying function and string encoding to hide its true actions. Its primary goal seems to be to steal credentials or perform unauthorized actions on the victim's HDFC Netbanking account.

Function Summaries

* **`_0x5123()`**: This function acts as a string encoding/decoding mechanism. It returns an array of strings which are used later in the code. The strings themselves are highly suggestive of actions performed in an HDFC netbanking environment like login, password filling, button clicking, etc.

* **`_0x3685()`**: This is a self-modifying function that acts as a decoder for the strings returned by `_0x5123()`. It takes an integer index as input and returns the corresponding decoded string from the array. The self-modification makes reverse engineering more difficult.

* **`_0x52f494()`**: This function handles the storage retrieval and checks for specific data within chrome storage and uses it to conditionally execute operations on the HDFC netbanking site.

* **`_0x27ba7e()`**: This function uses `setInterval` to poll for the appearance of a specific HTML element (related to remarks input in HDFC), and upon finding it, triggers a series of events in the context of the website.

* **`_0x1a2081()`**: This function waits for specific elements on the HDFC page and performs actions. It appears to be involved in filling out forms and triggering events.

* **`_0x56ce47()`**: This function seems to be performing checks on the current HTML, potentially for anti-malware systems or specific identifiers indicating that it is inside the scope of HDFC Netbanking. It seems it will continue to attempt to execute itself until it successfully manages to obfuscate itself.

Control Flow

* **`_0x3685()`**: The control flow is simple. It checks if the function has already been modified (by checking if it is the first call). If it is, it initializes the decoder and then executes the decoder logic. If it's not the first call, it directly uses the previously modified function.

* **`main (anonymous function)`**: The main logic is contained within an immediately invoked function expression (IIFE). This IIFE performs these steps:
1. Sets up an event listener for a specific event in the `document` (`#TranRequestManagerFG\\.CP_REMARKS_SINGLE`).
2. Uses `setInterval` to check for the presence of certain elements in the HTML page. If found, it interacts with HDFC login elements, performs filling of login credentials, and attempts to proceed with a payment process.
3. It checks for and handles cases where the HDFC Netbanking flow isn't detected or crucial elements are missing.
4. Includes a `chrome.storage` interaction to check for and use previously stored data potentially related to preferences or credentials.

* **`_0x27ba7e()`**: This function uses a `setInterval` loop to repeatedly check for the presence of a remarks input field. If found, it fills the field with "joker" and dispatches an event. If not found after a timeout (100 times 64ms), it logs an error message.

Data Structures

The primary data structure is the array of strings (_0x31f185) within the _0x5123 function. This array holds the obfuscated strings that represent the actions the malware performs. The strings are decoded using the self-modifying _0x3685 function.

Malware Family Suggestion

Based on its functionality, this code strongly resembles a **banking trojan**. Its obfuscation techniques, focus on interacting with an online banking website (HDFC), and actions involving credential filling and transaction initiation are all hallmarks of this malware type. The use of `chrome.storage` suggests it may also be capable of persistence. Additionally, it shows characteristics of other information stealers with the persistent monitoring and checking of HTML elements.