Analysis Report for: a.txt

Overall Functionality

This AutoIt script acts as a malicious installer. It reads configuration settings from a `setup.ini` file to perform several actions:

1. **Homepage Hijacking:** If `check1` is set to 1, it attempts to change the homepage of Internet Explorer, targeting both user and machine settings, including Group Policy settings. This demonstrates persistence and privilege escalation attempts.

2. **Shortcut Creation:** If `check2` is set to 1, it creates a shortcut on the desktop, pointing to an executable downloaded from a URL specified in `setup.ini`. This suggests the distribution of further malware.

3. **Executable Download and Execution:** If `check4` or `check5` are set to 1, it downloads and executes an arbitrary executable from specified URLs, adding another layer of potential malicious payloads.

4. **Self-Deletion and Process Termination:** After execution, the script deletes its own files (`setup.ini`, `data.bin`) and attempts to forcefully terminate its own process, hindering analysis.

Function Summaries

* **`_SETUP($dl, $exe)`:** This function downloads a file from a given URL (`$dl`) and saves it to the temporary directory with the specified filename (`$exe`). It then executes the downloaded file. Parameters: `$dl` (download URL), `$exe` (executable filename). Return value: None.

Control Flow

The script's main logic is a series of `If` statements, each controlled by a value read from the `setup.ini` file.

1. **Homepage Hijacking (`If IniRead($ini, "Plugin", "check1", "0") = 0x1 Then`):** If `check1` is 1, the script attempts to modify the Internet Explorer homepage settings in multiple registry locations, both user and machine, including Group Policy settings. It uses `RegDelete` to clear any existing homepage settings before `RegWrite` to enforce its own value from `$url`. `gpupdate /force` and `RunDll32.exe USER32.DLL,UpdatePerUserSystemParameters` are called to enforce the changes.

2. **Shortcut Creation (`If IniRead($ini, "Plugin", "check2", "0") = 0x1 Then`):** If `check2` is 1, it creates a desktop shortcut using `FileCreateShortcut` to an executable specified in `setup.ini`.

3. **Executable Download and Execution (`If IniRead($ini, "Plugin", "check4", "0") = 0x1 Then` and `If IniRead($ini, "Plugin", "check5", "0") = 0x1 Then`):** These blocks conditionally call the `_SETUP` function, downloading and executing additional executables.

4. **_SETUP Function:** This function uses `InetGet` to download a file, waiting until the download is complete using a `Do...Until` loop. It then executes the downloaded file using `Run`.

5. **Self-Deletion:** At the end, the script deletes the configuration file and attempts to kill itself.

Data Structures

The primary data structure used is the `setup.ini` configuration file. It's a simple INI file (likely) holding key-value pairs specifying URLs, filenames, and flags to control the actions of the script.

Malware Family Suggestion

Based on its functionality, this script strongly resembles a **downloader/installer** or a **dropper**. It downloads and executes additional payloads (potentially further malware), modifies system settings without user consent (homepage hijacking), and employs self-deletion techniques to hinder analysis. It also displays signs of trying to achieve persistence through registry manipulation and the use of group policy settings. Its use of hardcoded registry keys suggests a very targeted and less sophisticated approach compared to polymorphic malware. The targeting of Internet Explorer may indicate it is older malware that hasn't adapted to modern browsers.

Further Analysis Notes:

* The use of `#NoTrayIcon`, `#RequireAdmin`, and other preprocessor directives suggests it was likely compiled from an AutoIt script using a compiler.
* The commented-out regions suggest potential configuration options for the compiler.
* The URLs and filenames are not directly visible, indicating that they are loaded from the `setup.ini` file, further hindering static analysis. The hard-coded GUID is suspicious, suggesting it is trying to modify group policy settings.
* The forceful process termination attempts to erase evidence of its execution.
* The script is obfuscated to a certain degree through the use of conditional logic based on the `setup.ini` file. It would need further analysis to fully understand its functionality and potential capabilities. Dynamic analysis would be necessary to fully identify the payloads and their actions.

The analysis highlights serious security concerns as this script performs actions without user consent and likely installs harmful software.