

Domain Blocker

User GUIDE

ICTPRG434\_CSP | 14/06/2024

# Purpose

This Python script allows you to update your system's `hosts` file with a list of domains to block certain types of content, such as advertisements, malware, trackers, and malicious sites.

The `hosts` file is a local file used by the operating system to map hostnames to IP addresses. By adding domains to the `hosts` file and mapping them to `0.0.0.0` (an invalid IP address), you can effectively block access to those domains.

# Configuration

The script reads the URLs of the domain block lists from a `config.json` file in the same directory. The configuration file should have the following structure:

```json

{

"ad\_block\_lists": [

"https://www.github.developerdan.com/hosts/lists/ads-and-tracking-extended.txt"

],

"malware\_block\_lists": [

"https://raw.githubusercontent.com/StevenBlack/hosts/master/hosts"

],

"tracking\_block\_lists": [

"https://www.github.developerdan.com/hosts/lists/tracking-aggressive-extended.txt"

],

"malicious\_block\_lists": [

"https://malware-filter.gitlab.io/malware-filter/phishing-filter-hosts.txt"

]

} ```

You can modify this file to include the desired URLs for the respective block lists.

# Usage

When you run the script, a graphical user interface (GUI) will appear. In the GUI, you can select the types of content you want to block by checking the corresponding checkboxes (Advertisements, Malware, Tracking, Malicious). After making your selections, click the "Update Hosts File" button to update the `hosts` file with the combined list of domains from the selected block lists.

A screenshot of a computer screen

Description automatically generated

# Backup and Restore

Before updating the `hosts` file, the script creates a backup of the original file with a timestamp (e.g., `hosts\_backup\_20230614123456`). Additionally, it saves a copy of the last used `hosts` file as `hosts\_last\_used`. If you need to restore the `hosts` file to its previous state, you can click the "Restore Hosts File" button in the GUI.

# Permissions

The script requires administrative privileges (root on Unix-like systems or Administrator on Windows) to modify the `hosts` file. If the script encounters a permission error, it will display an error message and prompt you to run the script with elevated privileges.

# Dependencies

The script requires the following Python libraries:

- `os` and `platform` (built-in modules)

- `shutil` (built-in module)

- `requests` (for downloading the block lists)

- `json` (for parsing the configuration file)

- `tkinter` (for creating the GUI)

- `datetime` (for generating timestamps)

Make sure these libraries are installed before running the script.

# Limitations

The script assumes that the block lists follow a specific format, where each line starts with either `0.0.0.0` or `127.0.0.1` followed by the domain to be blocked. If the block lists have a different format, the script may need modifications to parse them correctly.

# Maintenance

As new domain block lists become available or existing ones change, you may need to update the `config.json` file with the appropriate URLs. Additionally, it's recommended to periodically update the `hosts` file to ensure it includes the latest domains to be blocked.

Please note that modifying the `hosts` file can have unintended consequences if not done correctly, so it's important to understand the implications and use the script with caution. Additionally, some applications or services may bypass the `hosts` file, in which case other security measures may be necessary.