Hacking the Great Firewall (HtGFW) Product Implementation

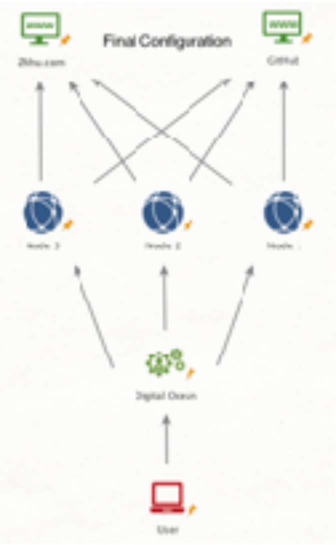
Mitchell Edwards

Introduction

The purpose of building the Hacking the Great Firewall platform was to allow for extendability and future implementation by other users. The Zhihu API was built specifically for scraping, but can be extended to allow the user to archive specific posts or scrape different subjects. The filesystem, naming convention, and text file layout can be changed according to different users' data processing standards to allow for a far more extensible platform that is customizable for each user.

The true value of the project lies in the data collected. I open-sourced the data along with the source code to allow for other users to download and view the data just as I can, so that the data can be used by different parties with similar goals, whether it be understanding Chinese culture in general or specifically understanding the Great Firewall on a deeper level than before. So while this document details the particular technical implementation that the author chose, it is designed to be used in any environment in any way the user sees fit, as is the accompanied data.

Infrastructure



The final product was implemented on three different DigitalOcean virtual private servers. This implementation allows for decreased home network overhead, increased network connectivity, guaranteed uptime, and high-performance networking and data processing, at a relatively cheap cost. The three servers divide the workload evenly by processing different text files with different TopicIDs. This also escaped many of the anti-scraping methods employed by Zhihu, including CAPTCHAs and IP/MAC bans.

The three VPS's run bash scripts that run on a continuous loop, calling the Dump and Check modules at differing, optimized time intervals. These scripts gather and check the posts and upload them to GitHub, the central data repository.

File Structure

The file structure is relatively simplistic, with the main directory holding all of the source code and subdirectories for each topic. New posts are dumped to their respective subfolders, named by TopicID, and Censored posts are moved into the Censored directory, also organized by TopicID.

I chose this flat directory structure for the sake of convenience, and plan to organize it in a more efficient manner in the future. The directory structure will remain simplistic, though, as the project is focused on Censorship-by-topic, so it would make logical sense for the File Structure to be organized as such.