BlockChain Life

Mohamed Legheraba Dinesh Bisesser Philip Blankendal
Michail Vrachasotakis Ka-Wing Man

February 2, 2018

1 Introduction

This work is an extension of the Cloudomate[1][1] project for which the main function was to automatically buy Virtual Private Servers (VPS). Its aim is to create a network of self-autonomous replicating entities that buys servers using cryptocurrencies and use their bandwidth to earn trust in order to trade it in the Tribler[2] decentralized market. The current version of the system updates the Cloudomate utility with Virtual Private Network support for a variety of providers, CAPTCHA solving functionality and a wallet for the Ethereum cryptocurrency.

Our goal was to not only reinforce existing functionalities, but also to add (extra) functionalities in order to realize the Blockchain Life's main purpose. We have added several new functionalities to this project based on certain perceptions on what was lacking in the previous Blockchain Life project in order to strive and make this concept truly successful. These new implemented features ensure that the agent's P2P traffic and VPS buying functionalities are equipped to not only handle but also to surpass the modern obstacles as self-sustaining bots in todays web-based traffic, making a large and exiting step into the direction of becoming truly autonomous.

2 Feature selection and Design Overview

Features implemented

- CAPTCHA Solver
- reCAPTCHA Solver
- Ethereum Wallet
- TorGuard VPN
- vpn.ac VPN
- Mullvad VPN

[1] https://github.com/Tribler/cloudomate
[2] https://www.tribler.org/

3 CAPTCHA Solver

For various VPN's and VPSs, an autonomous way of CAPTCHA solving is needed for instance to create an account or bypass certain anti-bot prevention mechanisms. In order to overcome this obstacle, it was decided to implement a CAPTCHA solver.

3.1 Solver Choice

Initially, various command line OCR tools were tested for simple CAPTCHAs that would enable us to avoid using a system like Amazon Mechanical Turk[3] or Anti-Captcha[4]. Some of the tools used were Tesseract, Gocr, Ocrad and Tesseract in Python with CAPTCHA image preprocessing by OpenCV. The results were disappointing and we had to resort to a system that employs other humans to solve it for our bot. This approach has a margin for human error and takes at least five seconds but it seems to be enough for most systems. Amazon Mechanical Turk was rejected because it involves a more lengthy account generation process and it is not CAPTCHA-centered unlike Anti-Captcha. The Anti-Captcha API was not only chosen for their specialization in all main CAPTCHA types but also because of the fact that they have been operational for over 10 years and have thereby proven their services to be resilient against the evolution of anti-bot Google reCAPTCHAs[5] during the past years. Next to the aforementioned advantages they are also cheap (1/1000 per CAPTCHA); and not to mention the fact that they provide means of buying their services through cryptocurrencies.

3.2 CAPTCHA Reload

To ensure that bots always have sufficient balance on their Anti-Captcha account, we have provided a script to automate the adding of funds to the account. An amount in dollars is given as parameter and the payment can be paid in either Bitcoin, Litecoin or DASH. The script will return a dictionary with the amount of the desired cryptocurrency and the adddress to send the cryptocurrency to.

4 Ethereum Wallet

Bitcoin transaction fees have risen to a point where they cost a significant amount of money. Therefore, other cryptocurrencies had to be chosen in order to avoid spending too much money on fees. As a popular alternative to Bitcoin, Ethereum is chosen for the payments. An Ethereum wallet has been implemented to handle these payments.

4.1 Infura scraping

To access the Ethereum network, we need a node. For Bitcoin, we use a light node included in the Electrum wallet. We tested the light node of Ethereum (both Geth and Parity) but they are "experimental" version and they don't work at all. We can install a full node but we need tens of gigabytes so it is not suitable for a small automate script.

[3] https://www.mturk.com/

[4] https://anti-captcha.com/

[5] https://www.google.com/recaptcha/intro/android.html

The other solution is to use a remote node. We tried different API's (like MyEtherAPI or Etherscan API) but they ask for a lot of personal data to use them and they don't allow all Ethereum network requests. The most simple way to get a node is to use infura[6].

Infura is a service that provides full access to an Ethereum node, free of charge and without limits. We have scraped the infura website and our script can get access to an Ethereum node automatically.

4.2 Wallet creation

To create an Ethereum wallet, we use the web3 and Ethereum libraries. We need access to an Ethereum node and a private key. The user can give its own private key and/or Ethereum node access, or we can generate an access to the infura service and generate a new private key.

4.3 Send of transactions

With access to a node and processing a private key, we can send transactions. You need to provide the address of the receiver and the amount of Ether to be sent. If you want you can choose the fee (in GWEI) and the number of gas. By default the script calculates the fee (using http://gasprice.dopedapp.com/) and uses 21000 as default gas number.

5 VPN

The basic component of the system is a process that uses the Cloudomate Python2 and Python3 package to buy a VPS among a list of certain providers. These providers do not allow the use of BitTorrent-like services in their servers, which in turn would not be a useful hosting provider for a bot that uses Tribler to upload P2P traffic and gain value-coins.
For this reason we decided to implement multiple VPN providers for the use of our agents.
The solution lies in that the VPN services provide a means for masking the torrent traffic. Buying a VPN subscription is similar to the original functionality of purchasing a VPS subscription. Therefore, the rationale is that a process buys not only a VPS and logs in to it, but it also buys a VPN using the same cloudomate package. In order to be versatile enough, we implemented numerous VPN providers. All of them accept several cryptocurrencies as payment for their services.

5.1 Mullvad VPN

Mullvad[7] is a provider that offers real anonymity as it generates an account number for a customer without asking them any personal information. In order to get this account number, the contents of a simple CAPTCHA image must be filled in by the client. For solving this CAPTCHA obstacle, we use our own implemented CAPTCHA solver explain in section 3. This number is then used for any interaction with the provider, like buying or updating a subscription and downloading the necessary files for the service. There are several ways of installing the VPN. We have chosen to make use of OpenVPN, as it is supported on almost every type of machine. An end-to-end implementation is added to Cloudomate, meaning that you can create an account, buy a subscription and install this VPN completely automatically.

[6] https://infura.io/

[7] https://www.mullvad.net/en

5.2 TorGuard & VPN.AC

Coinpayments.net is one of the fastest growing payment providers platform for paying with different cryptocurrency (supports over 125 Coins), including Bitcoin, Ethereum, Litecoin, DASH, Dogecoin, Mintcoin and many others. We decided to implement TorGuard[8] (due to its popularity) as well as vpn.ac[9] (due to its stability) in order to have VPN support that can bypass Bitcoin's transaction fees, by using the Ethereum as explained before. Both VPN providers uses Coinpayments.net as the payment platform for accepting cryptocurrency other than Bitcoin. Therefore, it provides future support for payments with other coins than the coins supported now by Cloudomate.

For scraping these VPN's, Selenium with headless Chrome is used. This is proven to be much better than libraries such as MechanicalSoup and RoboBrowser, as these libraries cannot run JavaScript and thus websites, where JavaScript is necessary, could not run properly and therefore they could not be scraped properly. With Selenium, a real browser is opened and it navigates through websites with much more ease. These VPN's have a normal account (email/username and password) and make use of the Coinpayments.net gateway for their transaction.

After payment a different username (given by VPN) and password (given by VPN, but can be changed) are used for setting up and installing the VPN. Both these VPNs have an end-to-end implementation as well.

5.3 Code structure Design and Design choice

Multiple VPN's are added to Cloudomate, so that the bot can host an exit node anonymously. We divided the code of each VPN provider into two parts: For buying the service (contained within the cloudomate folder) and for installing and running the VPN service obtained trough purchase , with each time randomly choosing a country to route traffic to.

We structured it this way because it is typical for VPN providers to provide services that can be used simultaneously on different machines. Therefore by choosing this design, we note the fact that once an agent has bought a VPN, it can create at least four children that can use the VPN service bought by its parent. Therefore, the child only needs to utilize the script to run a certain VPN as opposed to first needing to buy one. Perhaps evenly sharing this with agents that are not necessarily related.

6 System overview

As mentioned before, agents are supplied with a either a Bitcoin or an Ethereum wallet which can be used for acquiring a VPN service for both themselves as well as for their children. This part of the system needs to become more flexible and that is why other well established cryptocurrencies have to be included, which means that a different wallet needs to be implemented for each of them. To increase the range of providers, additional features like CAPTCHA solving and phone authentication need to be utilized. Figure 1 shows this whole process.

[8] https://torguard.net/

[9] https://vpn.ac/

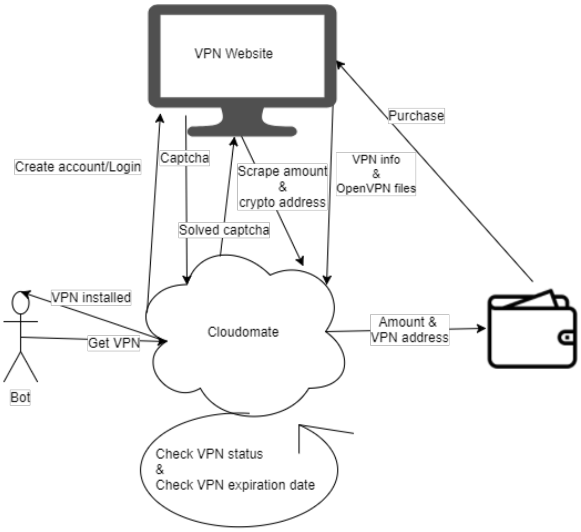

Figure 1: BlockChain Life process

7 Testing

Most of this work was tested on a server bought by Cloudomate. Additionally, a test of Plebnet's[10][1] installation and functionality was performed on this VPS. After a successful setup, it was run after enabling the VPN and a check of its Tribler plugin was also successful. An issue occurs though: using the VPN changes the routing table and some services like SSH. As a result the VPS is not reachable by other machines even in its new IP, although it is connected to the Internet. Despite trying various approaches discussed online like firewall and routing rules, the issue was not resolved and may potentially lead to a problem with the communication of the Plebnet community.

8 Conclusion

Cloudomate has been enriched with quite some stunning new features. It can now buy and use VPN's with multiple types of cryptocurrencies and solve CAPTCHAs. Now, it has a means of bypassing Bitcoin transactional fees, due to Ethereum support. By using scraping, crowdsourcing and blockchain techniques, Tribler is step closer to dominating the world of privacy and peer-to-peer.

9 Future work

There are several features that should be included by future blockchain engineers. Due to time shortage, the Litecoin wallet had to be dropped. This wallet could have been used for reloading the Anti-Captcha, for they also accept litecoin payments. Another aspect that should be done is looking at the distributing part of the bots and how they would share all this information (wallet, VPN, VPS). More unit tests could have also been included to better test the performance.

[10] https://github.com/Tribler/PlebNet

References

[1] Jaap Heijligers, René van den Berg, and Mitchell Hoppenbrouwer. Plebnet: Botnet for good. 2017.