**GeoIP – RDAP Lookup/Filter Models**

**What can be expected in this project?**

1. Extracted all 5000 IP addresses in a random styled text file.
2. Performed GeoIP and RDAP Lookups for those IP addresses.
3. Can Query on all the Lookups available for both GeoIP and RDAP
4. Returns the desired IP Chunk of IP address pool for given filters as queries.
5. All these models are isolated and decoupled with strict Input/output interfaces with possible test cases available for each.

**Approach:**

1. **IP Address Extraction-**

* Present in basefile.py main function. Used conventional approach of reading a string. Extraction of unique IP address done by using various strategies.

1. **GeoIP Lookup –**

* Present in geip.py
* Downloaded few free public databases (.csv files) available for a GeoIP Lookup.
* Used Math, Pandas and Numpy for data analysis, extraction and strategy building
* From this model 12 various parameters extracted for every GeoIP lookup. They are “Country”, “Region”, “City”, “Postal Code”, “Latitude”, “Longitude”, “Metro Code”, “Area Code”, “Subscriber”, Start IP Number for Subscriber”, “End IP Number for Subscriber”, “Location ID”.
* Merit – It is an independent model which do not rely on 3rd party mechanism for a lookup and data obtaining except using the large databases (.csv files) freely available.

**Lookup can be made using this model WITHOUT an internet connection.**

* Demerit – As those databases are free, the range of IP addresses this model works is 1.0.0.0 – 223.255.255.255

1. **RDAP Lookup –**

* Present in rdap.py
* Used libraries like Requests, JSON and Regular Expressions.
* No free databases found to build this model. So, used public website - <http://rdap.apnic.net/ip/>{address} to obtain the RDAP Lookups for each address.
* Parsing done accordingly to obtain desired contents from output chunk.
* From this model 8 different parameters can be extracted for every lookup. They are “Handle Used”, “IP Version”, “Name of Node”, “Type of Node”, “Country”, “Emails Associated”, “Addresses Associated”, “Phone Numbers Associated”
* Merit – Quick in Data extraction.
* Demerit – Dependent Model. Need internet for model to work.

1. **Querying the lookups –**

* Source file is filtering\_ds.py, also geofilter() and rdapfilter() models in base.py are used.
* Querying is done by a Key-Value approach.
* Filtering parameters are “Keys” and their associated data to search for are “Values.”
* While giving inputs Each key value, should be separated by SINGLE SPACE as per design.
* For GeoIP Query, any number of filters between 1 and 12 can be applied.
* For RDAP Query, any number of filters between 1 and 8 can be applied.
* Each query calls the object-oriented data store to verify each lookup whether all filter parameters are satisfied.
* Each of GeoIP Query model and RDAP Query model returns a list of IP address pool which satisfies the given set of filters for every query.

1. **Parsing/basefile.py –**

* It is the central file to this project.
* Main function, it does IP address extraction as well as requests whether a query (GeoIP or RDAP) can be initiated.
* Extracted IP addresses used for lookup models and then parsed accordingly to save them in a data-store.
* If queries initiated, the necessary filters chosen and applied using this model only.

**Scope to improve:**

* Instead of conventional OOP styled and Dict storage for Lookup’s. I had a plan to use to Redis (Python tool) for in-memory. Though it is also key-value approach. The data extraction while querying would be more simple and flexible.
* I am unsure whether I can use Redis or not, as this is 3rd party package. So, I chose to store data in a dictionary than using a mysql database (another 3rd party package).