Digital Security Landscape

Analyzing Cyber Threats and Credential Dynamics in Online Marketplaces Operating inside Anonymous Tor Network

Group: Daim Limited Edition

Members

Mohammad Asif Ibtehaz (asif.ibtehaz@tuni.fi) – 50366228

Table of Contents

Abstract

This study delves into the digital security landscape, particularly focusing on the dynamics of cyber threats and compromised credentials within marketplaces operating inside the Tor network. The study systematically categorizes and analyzes extensive datasets from the different marketplaces and malware log dump, revealing critical insights into personal, financial, and online account information trends. It highlights United States as a primary target for online attacks and identity theft, offering a comprehensive view of regional and currency-based price variation in the illicit trading. Through data parsing, cleanup and ratio analysis of unique usernames, passwords and email, the research tried to uncover prevalent credential reuse patterns and diverse cyber threat scenarios across different countries. This study tried to showcase the complexities and evolving nature of digital security challenges in the shadowy realms of the internet.

Background Information

The Tor Network

Tor, short for The Onion Router is a free open-source software, initially developed by United States Naval Research Laboratory's employees in 2002. The project has been open sourced in 2004. The project was initially designed to protect American intelligence communication online. In 2006, The Tor Project, a non-profit organization has been established to maintain Tor. [9] The mission statement for The Tor Project is -

> *To advance human rights and freedoms by creating and deploying free and open source anonymity and privacy technologies, supporting their unrestricted availability and use, and furthering their scientific and popular understanding. [10]*

Tor works by sending the network traffic through three servers, also known as relays in the Tor network. The first node or the Entry/Guard node is randomly selected by the Tor browser. It's publicly known. Entry node introduces the data to the Tor circuit. After the request enters through the entry node, the data is fully encrypted, it's sent through series of nodes and to ensure anonymity each middle node only knows the preceding and subsequent middle nodes. Each node takes off one layer of encryptions, thus the data is fully safe from eavesdroppers on transit. Once the last layer of encryption is peeled off, the request leaves the Tor network via an exit node and reaches the public internet. [11][12] Even though when we hear Tor, we somehow automatically connect it with illicit, dark sites, Tor has lots of other uses [13] –

1. Tor provides privacy from identity thefts.
2. Tor enables users to circumvent censorship.
3. Tor enables Citizen journalists in countries like China to report secretly where the public internet is subject to scrutiny.
4. Human rights activists use Tor to anonymously report abuses from danger zones.
5. Tor allows law enforcement officials to view questionable websites without leaving a trail, so system administrators cannot identify anything from the log files.

Onion Websites

Servers that receive request through the Tor network only are called Onion Services (formerly known as Hidden Services). Onion Service is accessed through Onion Address, via the Tor browser. These services aren't attached to any IP addresses, unlike how the public internet works. Onion addresses are usually quite long, and not memoizable directly. For example, Onion Service URL for BBC is https://www.bbcnewsd73hkzno2ini43t4gblxvycyac5aw4gnv7t2rccijh7745uqd.onion/. Onion services are not indexed, and Tor is decentralized by design. There's no direct list of all readable list of onion services. [9] Some popular Onion Services are -

1. Hidden Wiki: http://6nhmgdpnyoljh5uzr5kwlatx2u3diou4ldeommfxjz3wkhalzgjqxzqd.onion/
2. Proton Mail: https://protonmailrmez3lotccipshtkleegetolb73fuirgj7r4o4vfu7ozyd.onion/
3. The Silk Road: An online black market and first modern darknet market, operated from 2011-14 [14]
4. Genesis Marketplace: Genesis Marketplace, an English language website that facilitates identity fraud. It operates in 218 countries. [3]

Malware Log Dumps

Malware log dumps provide real-time insights into stolen information by malware like RedLine Stealer. These logs capture live data directly from the users. Unlike data dumps found on the dark web, which often contains outdated or recycled credentials, malware log dumps usually contain unaltered, fresh credentials, personal information. [15]

Cryptocurrencies

Cryptocurrencies are digital currencies that are not reliant on any central authority, such as government or bank. Cryptocurrencies typically use decentralized control. They don't have any physical form. Each cryptocurrency works through a distributed ledger, typically blockchain that serves as a public financial transaction database. The first cryptocurrency was Bitcoin, first released in 2009. As of 2023, there are over 25000 other cryptocurrencies, out of which 40 had a market cap exceeding $1 billion.

Validity of cryptocurrencies are provided through blockchain and it's a continuously growing list of records, called blocks. Each block contains a hash pointer that links itself to the previous block, except the initial block.

Properties of cryptocurrencies gave them the popularity to use them in controversial settings like online black markets. [16]

Marketplaces on the Dark Web

Darknet markets are commercial websites that operates via Tor or other anonymous networks. Darknet sells items involving drugs, cyber-arms, weapons, forged documents, stolen credit card details etc. The first marketplace that uses both Tor and Bitcoin escrow was Silk Road, operated from 2011 till 2014. Some of the popular dark web marketplaces are -

1. InTheBox: Largest marketplace for mobile malware
2. GenesisMarketplace: Largest marketplace for stolen credentials, cookies, and digital fingerprints. This is an invite-only market. Each item listed for sale is known as bots. Each bot on Genesis contains stolen information, that is kept up to date through the use of malware that silently lurks on the victims system.
3. Invictus Market: Offers a wide range of recreational and prescription drugs.
4. Cartel Market: All-purpose darknet market that hosts wide variety of services like drugs, self-defense products, electronics even hosting.

There were several other darknet markets that have become defunct now such Silk Road, Silk Road 2.0, Silk Road 3.0 Reloaded, AlphaBay, White House Market, Atlantis. After Silk Road, Atlantis was the first marketplace to accept Litecoin as well. All the darknet markets operate using some form cryptocurrency and escrow system as a form of transaction. [17][18][19]

OpenAI

OpenAI is an artificial intelligence research organization. They have developed several large language models such GPT-4, as well as advance image generation models like DALL-E 3 and in the past published many open-source models. GPT 3.5 is a set of models that can understand as well as generate natural language or code. GPT-4 is OpenAI's most advanced system, improved on GPT-3.5. GPT-4 also understands natural language or code and produces safer and useful responses. [3][20][21][22]

Research Questions

The dataset that I have used in this research has been collected, organized, and previously used in the study conducted by Nurmi, Niemelä, and Brumley, as detailed in 'Malware Finances and Operations: a Data-Driven Study of the Value Chain for Infections and Compromised Access', in the proceedings of the 18th International Conference on Availability, Reliability and Security (ARES '23), August 2023, Article No.: 108, pp. 1–12. [1] They have been divided into three sections:

1. Account Access Set
2. Victim Access Set
3. Malware Infection Set

In each of these sections, in broad term, I primarily tried to find the relationship among the variables present. Research methodologies and how to recreate similar outputs have been described in the later sections. Account Access Set contains information from various online database marketplaces from the Tor network. In this category, I have tried to answer:

1. categorize the data.
2. how much prices vary on location for different product categories in various currencies.
3. data distribution in different locations, top 10 countries in different categories.

While working with this data set, I had to rely a lot on OpenAI [2] Apis, details are described in later sections. Victim Access Set has been used to show users personal authorization information from 2019 till 2022. These data were up for sale at Genesis Marketplace [3]. With this data set, I tried to find:

1. Top 20 Services
2. Top countries from where a certain domains data are from
3. Price range for top domains credentials
4. Price for a specific domain in top 10 countries
5. Compromised domain and OS in use.
6. Most popular OS systems in usage
7. Different Windows Systems and their percentage in the overall data

In Account Access Set, I have used about 98% of the entire data set (33896 files). In Victim Access Set, I have managed to utilize 100% (half a million victim files) of the given data set. However, in Malware Infection Set, I have only managed to use 25% of 1.8 million victim data. Due to some architectural decision and my lack of foresight, I failed to use the entire data set. Malware Infection Set uses malware dump log from 2019 and 2020, which originally originated from 14 malware networks.

However, with the data that I already have, I tried to find –

1. Unique password to Unique domains ratio
2. Unique Username/Email to Unique Password ratio for selected countries
3. Top 10 Services in the data set in use
4. Top Domains and their corresponding countries (from where the victim might be)
5. Number of Top-Level Domains in the files
6. Different services and how many accounts might be compromised.

Methodology

As previously stated, the data set I have used in this study has been acquired previously by J. Nurmi, M. Niemela and B.B. Brumley, "Malware Finances and Operations: A Data-Driven Study of the Value Chain for Infections and Compromised Access," [1]. This section contains how to replicate the study using this dataset. The codes to replicate this study is available in this GitHub Repository: https://github.com/shepherd-06/Tor-Marketplace-Analysis. This section assumes the datasets has been downloaded from the website: https://zenodo.org/records/8047205. All three datasets have used Python, Panda, Matplotlib, Seaborn and Sqlite3 to store, process and visualize the data. However, this comes with some limitations, which will be explained at the end of this section.

Account Access Set

This data set has been acquired by parsing the Database marketplace operating inside the Tor Network. Various kinds of information are being sold here from Drivers License, Passport, Social Security Number, Bank Account, Credit/Debit Card information, Email Address, Account in different websites. Previous authors crawled the website between November 2021 to June 2022 and the original dataset contains 33,896 victim files. Using this dataset, I have wanted to answer the price range of different category data that are being sold in the onion marketplace and the locations.

Replication

In "AccounctAccessSet" folder, the files can be broken down into three categories:

a. **Data Parsing**: Here I have parsed the original text file and tried to find relevant information from them and add them in the sqlite3 table. By manual trial and error, I tried to understand how the JSON files may look, tried to understand the similarities among them to write the regex for this parsing. It managed to parse 98% of the total dataset. There are two files, 'data_parser_1.py' and 'data_parser_2.py'. After using data_parser_1.py, I have realized that the regex I have used in this file, failed to parse some files. As such I have written a new file, 'data_parser_2', together I have managed to fetch most (98%) of the information from the JSON files.

b. **Data Cleanup**: A lot of the data contains different location names or location names are in suburb or city area in some country. Prices are also in different currencies in different formats. For example, 0.1BTC, 0.1 BTC, BTC 0.1 etc. Moreover, prices also come with variation or ranges, for example USD 10 to USD 30 depending on locations. I have used OpenAI API to pass the location and price data in two separate script file, ("gpt_location_cleanup.py" and "gpt_price_cleanup.py") and get the overall location. For location cleanup, I have used the pycountry package, in the beginning to get the location and then if I don't have any concrete result, I send the data to OpenAI. This reduces the reliance on OpenAI API and furthermore, reduces the overall cost. "gpt_location_cleanup" script has been developed in a way that it will use pycountry first, then a local cache which I have managed from already parsed location data and then finally, if the previous two method fails it will use OpenAI. OpenAI has been instructed to mark the location "Unknown" if it's undecipherable. Similarly, in price cleanup, I have asked OpenAI to fix the prices in "X Currency" format i.e. "1 USD, 2 BTC". If there's a price range, OpenAI sends the average between them. Sometimes I failed to parse price without the proper currency, in that case, it has been marked as "X aa". Each script uses model GPT-3.5-Turbo and GPT-4-1106-Preview, depending on the need. I had 500K total token limit in 24H, so even though it's possible to finish these scripts within half an hour, different users might experience different time frame depending on what tier their organization is at. In this study, I have used two paid accounts (two

different organizations) and 9 different API Keys to increase my overall token limit to 1M/24H. API Keys are designed to be chosen randomly from the list using "random.choice" method.

c. **Data Visualization**: There are two files "data_analysis" and "visualizer" where "data_analysis" in the main function to fetch data from the database and if needed, do some calculation, and then finally send the data to visualizer to plot using "panda, seaborn and matplotlib". When the first file is executed, it will show a simple menu in the terminal to choose which category and which data the user is interested in seeing. Depending on the situation all graphs can be visualized in less than 2 minutes. When fetching the data from the database, I have used "LIKE %parameter%" query, so instead of looking for exact string match, the query checks if "parameter" exists in that specific column.

Other than these three categories, I also have a SQL Manager file to manage the SQL Queries in some situations. The code in this folder is a jumbled mess, as I had to go through lots of trial and error to create a good result. All the scripts in this section will create a lot of log files, they are marked in ".log" extension.

Limitations

There are some limitations to this approach. For one, OpenAI API might send unknown/hallucinate and start sending random data. When the script runs, the user must keep an eye on the data log. It's also a good instinct to keep a backup of the database. Another major bottleneck is, I have parsed the file assuming what 99% of the data set will look like depending on 10 random files I have manually checked. Even though I have managed to parse 98% of them, and usable data point from them, there's no way knowing whether my data set is correct or not in this study.

Victim Access Set

This dataset contains details dataset of necessary information regarding user's online account like username, password, emails and the services and domains they are acquired from. All this data has been collected from Genesis Marketplace between 2019 to 2022. The size of this data is 500,000 and I have managed to use 100% of this data set in my study.

Replication

In "VictimAccessSet" folder, the files are broken in to two categories –

a. **Data Parsing**: The "data_parsing.py" contains the necessary code to parse through the JSON files and insert the relevant information in the SQLite database. Unlike Account Access Set, in this section, the data set is properly organized and as such very few data cleanup has been required. Almost all the data cleanup occurred in the Data Analysis and Visualization phase. There's no separate cleanup phase in this section. The "data_parsing.py" assumes that the datasets are in "genesisvictims/genesisvictims" in this path inside the current directory. However, since there are too many files, unlike previously where I have used to list directories of the Pythons OS class, I have used scandir here. I don't have a specific timeline for how long this script may take, however, it should take less than six hours.

b. **Data Analysis & Visualization:** I have two files here. "data_analysis.py" fetches the data from the sqlite3 table, and "data_viewer.py" does the visualization. Each graph takes ~1 minute to generate. There's a menu in the terminal which tells the user what to do. There are options to generate graphs by different domains as well. When fetching the data from the database, I have used "LIKE %parameter%" query, so instead of looking for exact string match, the query checks if "parameter" exists in that specific column.

Other than these two categories, I also have a utility file to manage the SQL queries in one place. Unlike "AccountAccessSet", codes here are quite clean and organized. There are necessary commands and comments to replicate this portion easily.

Limitations

I thought of doing a time series, when each domain's information (username, password, emails etc.) has been possibly compromised (based on the installed & updated time). However, due to time limitations, I couldn't complete this work.

Malware Infection Set

This data set contains 245 malware log dumps from 2019 – 2020 from 14 malware networks. The dataset contains 1.8 M victim files. As I have mentioned before, due to my lack of foresight and bad architectural decision early on, I couldn't manage to complete the work. The queries and the code here are a little complicated and I am going to try to give a description of what I was trying to do, to the best of my ability. In the limitations section, I will describe the workaround and where I came short.

Here the log files contain username, password, email addresses in hash, instead of plain-text, associated domains, services, and countries they are compromised from. I have 4 tables, services, domains, summary, and counts. In services and domains, I have inserted only the unique URLs, how times they have been compromised, their corresponding countries. The summary table contains data by a specific country, and the count table contains the overall for a specific file, i.e. how many unique username, password and email address, domains and services has been compromised.

Using the method described above, I have managed to parse the following data (a stat generated using the Counts table):

TOTAL FILE PARSED: 446,235
TOTAL PERCENTAGE: 24.65% OF 1,809,988 DATA
0
USERNAME 5,338,894
PASSWORD 4,305,776
EMAIL 1,768,702
DOMAINS 8,756,779
SERVICES 9,939,847

Using the content of the "Summary Table", I have generated country specific visualizations like

a. Ratio between unique username, email address and password for some specific countries
b. Ratio between unique password vs domains for some specific countries.
c. Top 10 countries and their corresponding compromised data

It was easier to work Counts and Summary table, as the relationship between data points are straightforward and less complex compared to other two tables. With the "Service" data, I have generated the following relationships –

a. Top 10 Compromised Services (and their base URLs)

b. Top domains and their associated countries
c. A Count of common Top-Level Domains that has been compromised.
d. Relationship between compromised Services and their associated compromised credentials from the Count table.
e. Top services and the domains they are associated with.

I have used the data from Domains table on the visualization above, I haven't done anything separately with those data. I have underestimated the volumes of unique URLs for both domains and services table as well as I haven't considered taking the base URL. As such, each table has over 1 million rows. Moreover, I have used a standard loop to parse through the file in the beginning, which was quite time consuming. Later, I wrote a new script to use parallel processing to go through multiple files at once. While the second script is slightly faster than the first script, there was an issue of database being locked by another process as such several files data wasn't inserted/updated by a specific process. Since there are two scripts, each script saved data on a separate database. In addition, I have indexed the databases, as such insertion took more time, especially for Service and Domains table. Besides, to keep the index data, one of the database sizes became 1 GB.

Replication

Keeping the above information in mind, I am again going to break down this codebase into two sections:

a. **Data Parsing**: I have two files "analyer_1.py" and "analyzer_2.py". Each of them parses through the datasets. Dataset file path is hard coded in the code and as such they must modified before the code runs. "analyzer_1.py" uses a loop to go over files and "analyzer_2.py" uses parallel processing. Besides these differences, both scripts remain identical. In the Domain and Service table, the scripts update the entries if the URLs already exist. If it's a new entry, then the scripts insert a new entry. In the Summary table, the scripts insert/update the entry by a country name and in the Count table, the scripts insert a new row for each file.
b. **Data Analysis and Visualization**: I have five files in this category:
    i. data_summary_visualization.py
    ii. domain_data_analyzer.py
    iii. service_data_analyzer.py
    iv. summary_visualizer.py

    "data_summary_visualization" is a statistic that prints out how many files both scripts has processed. "summary_visualizer" file works with the Summary tables data and answers the questions I wanted to answer using those data. "service_data_analyzer" analyzes the Services, Counts data and answers the questions from this sections. "domain_data_analyzer" has some functions to use in the "service_data_analyzer" class, but it doesn't generate any graph.

Limitations

a. For such large amounts of data, SQLite is a bad choice. It was preferred use PostgreSQL/MySQL
b. Instead of using Parallel processing to handle the files, I should have used a Queue system like Celery. It will reduce the overhead from file-parsing, making it a lot faster and smoother and if Celery is coupled with PostgreSQL/MySQL database system, it would have been lot faster.

c. Service/Domain's table contains a lot of datapoints that are hard to visualize in a single graph. As such tabular format has been used.
d. Service and Count relationships is not 100% correct. As I can't confirm which email/username and password combination has been used for which domains/services.
e. If the current code is in use, file parsing will probably take ~3-5 days, maybe more. Data visualization take ~<1-4 minutes for each of the graph.

Limitations of this Study

1. Reliance on OpenAI API for data cleanup might introduce inaccuracies due to hallucinated responses or random data generation.
2. The assumption made during file parsing, based on a small subset of data, might not accurately represent the entire dataset, leading to potential errors in interpretation.
3. The use of SQLite for large volumes of data and lack of time series analysis for compromised account information further constrained the depth of the research.
4. Study finds it difficult to a conclusive decision particularly using data set from the malware dump.
5. In Price Point Analysis and Domain-Service Relationships in Malware Infection Set section are limited due to partial dataset being used (only 24.65% of the malware data was analyzed)

Results

Account Access Set

In this section, I have managed to categorize the data into 3 categories mainly. Personal, Financial and Online Account (anything that relates to any online credentials). I have 2% unparsed file. You can see the details from Figure 1. From the dataset, I have categorized any information where SSN (Social Security Number), DL (Drivers License), DOB (Date of Birth), Full Info (Full Information), Passport are mentioned as Personal Identification Information that are being sold. Any information where Bank, Cheque, Credit, Debit, Card, Cash, Tax, Payment is mentioned, I have categorized them as Financial. They are mostly used for money laundering, cash/cheque fraud or any other kind of financial fraud. Finally, on the third and final category, I have selected any information where Email, Password, Username, @, any domain name (.com), Phone, Online, Social is mentioned. With this categorization, I have found that 82.2% data belongs to Personal category, while 9.3% belongs to Online Account related and 6.5% belongs to the financial category.

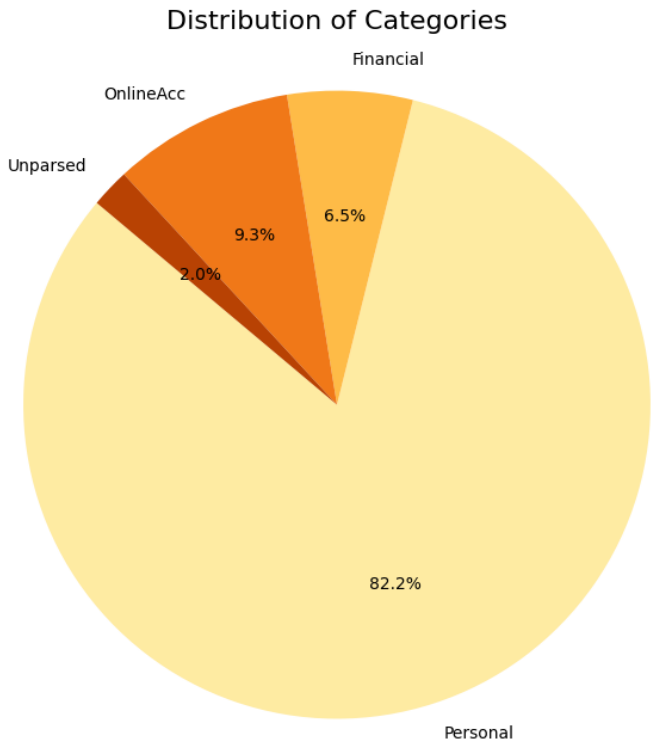

Figure 1: Account Access Set, Distribution of Categories

After that I figured out the top 10 countries in each category. I have selected how many times each country has appeared in the overall dataset. In figure 2, I have displayed 3 bar plots showing the top 10

countries. In all 3 categories, information from United States is always on high demand, as it's visible from the figure 2. There are some common countries on this list, such as US, Great Britain, Canada but there are some uncommon countries as well such Republic of Moldova, Kenya, Panama, Brazil. From the graph, its clearly visible that compared to other countries, users from United States are subjected to a high number of Online attack and identity theft. Especially, in the Online Account section, compared to the number of accounts from US, Moldova, Great Britain, and Estonia's stolen online account number is less than half in the dataset. Some common names are in the financial category like Monaco, Cayman Island as they are popular for money laundering and other financial fraud. [4][5]

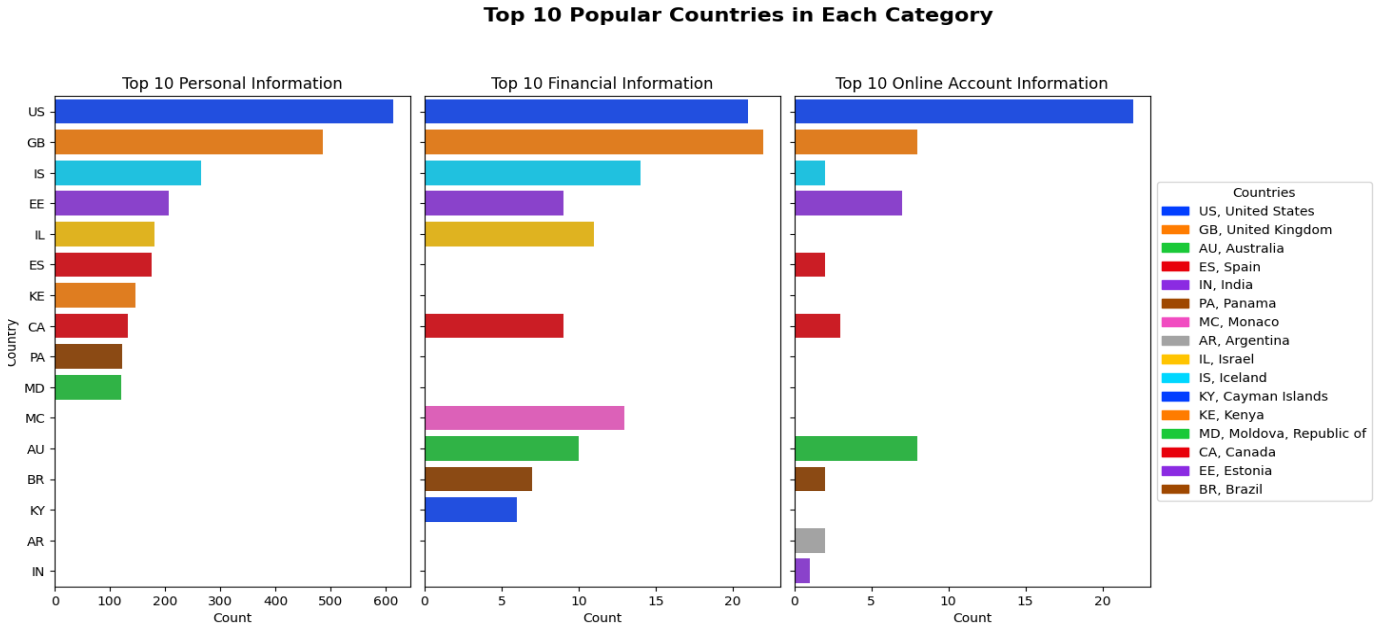

Figure 2: Top 10 countries in Each Category

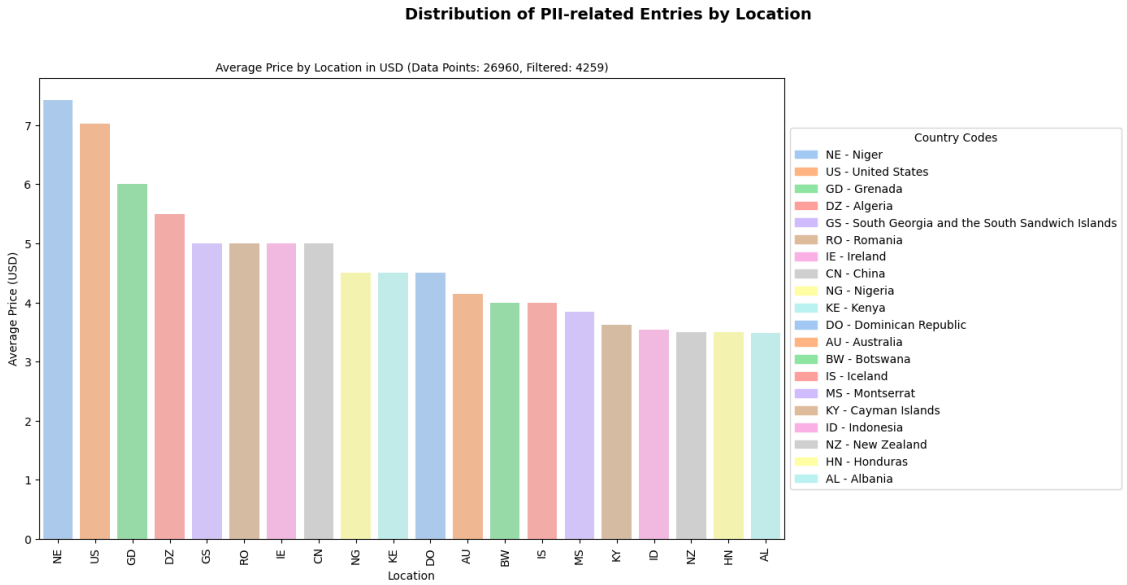

Figure 3: PII Average Price by Countries in USD

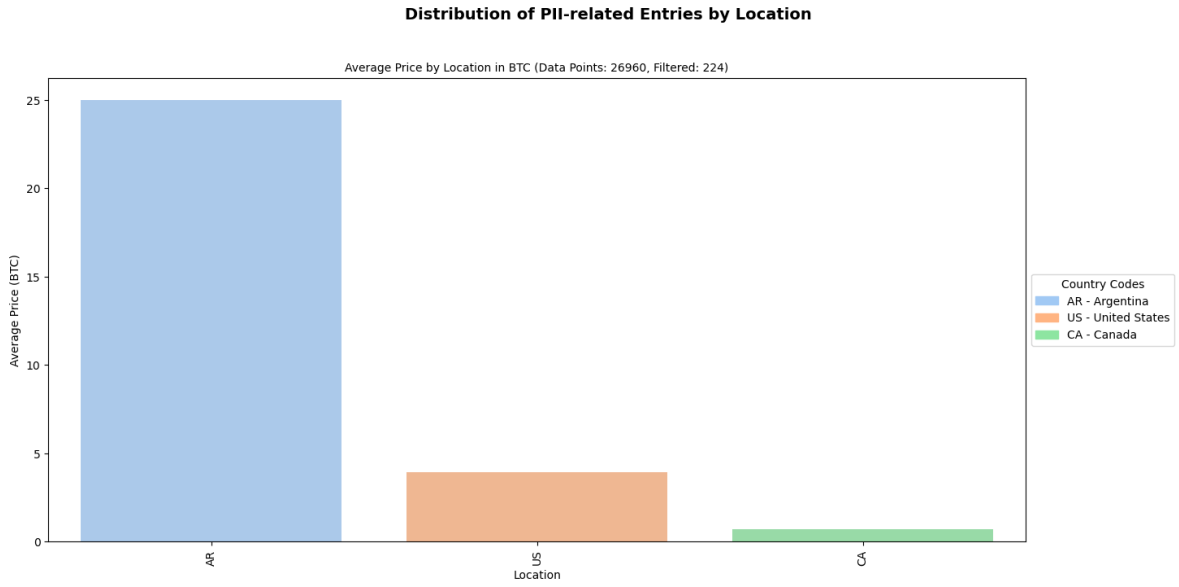

Figure 4:PII Average Price by Countries in BTC

Figure 3 and Figure 4 describe the average price for Personal Identification Product in both USD and BTC. Even though figure 2 shows US has lot more products from Personal Identification category, when we are averaging the country from African continent Niger has an average product price just above 7 USD, where products originating/PII information from US are sold at around 7 USD per product. To get to the Top 20 countries, we have used 4259 data points. During this study, I couldn't verify the currency for a lot of products. This is a limitation. Products that sold/where prices are mentioned in BTC, Argentina has average product of 25 BTC! In the dataset, there are lot less product mentioned in BTC than USD.

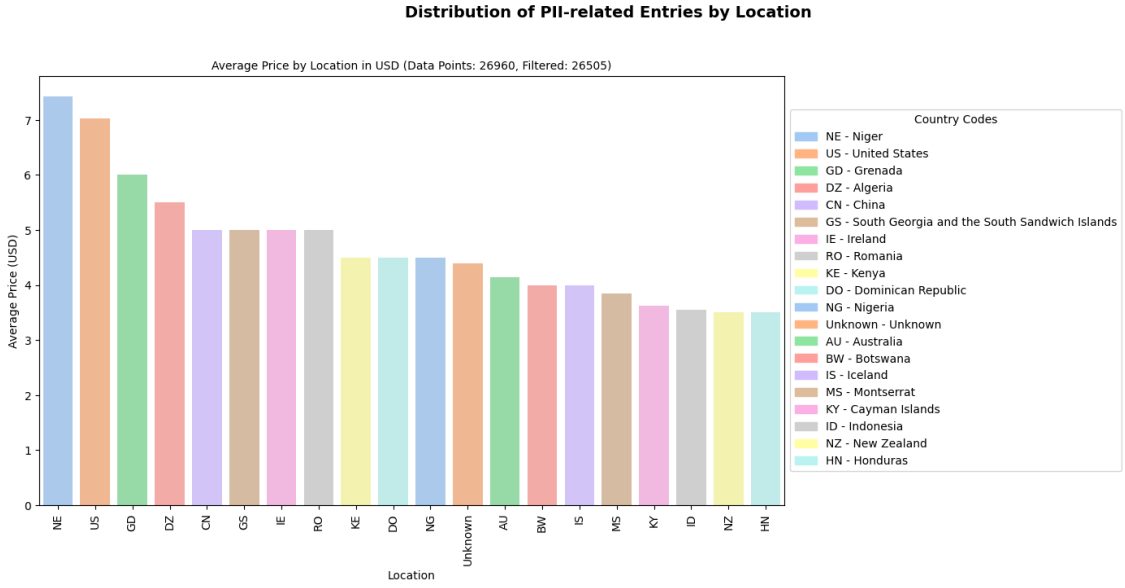

Figure 5: Average Price of Products, with Unknown Location

Figure 5 is the same graph describing Figure 3. However, I have added datasets where locations couldn't be identified properly, as such I have marked them unknown. Even though the graph is nearly identical, now we have considered nearly 26505 data points. On the other hand, there's no difference if we consider Unknown locations for products sold in BTC currency. Dataset is identical in that scenario.

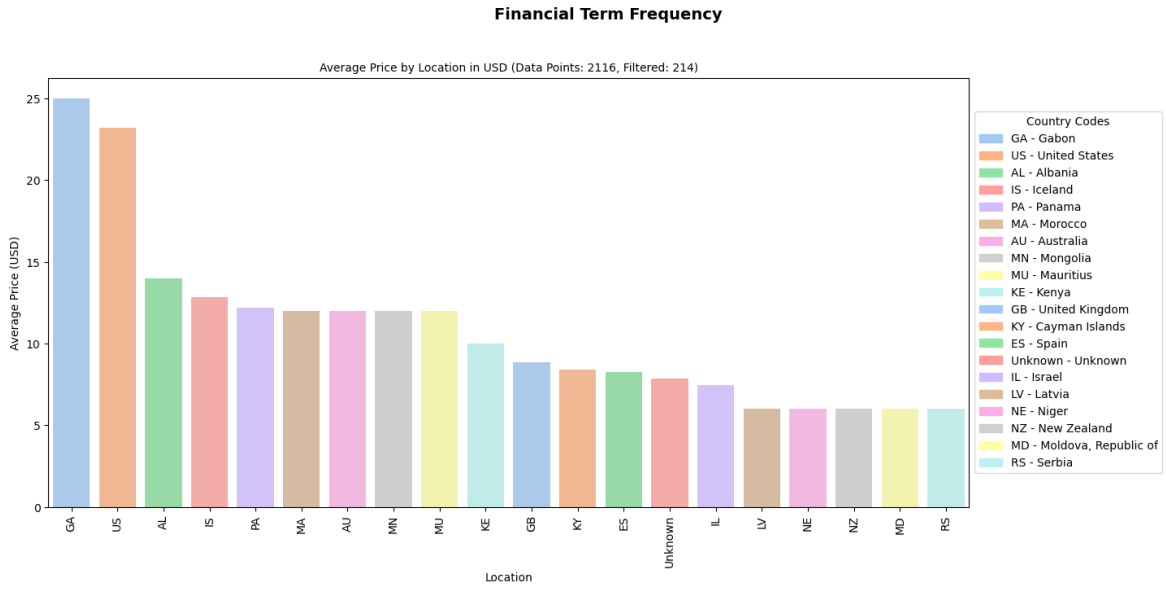

Figure 6: Financial Items Sold in USD

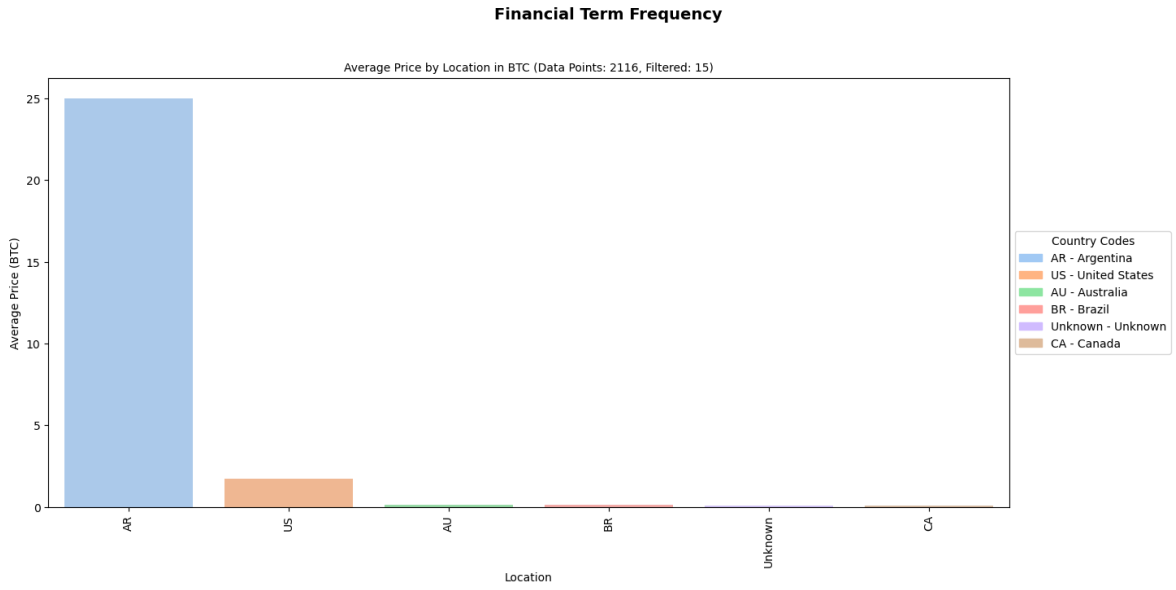

Figure 7: Financial Items Sold in BTC

Figures 6 and 7 show the top 20 countries from where financial category items are sold either in USD or BTC. We have considered Unknown locations here. If look at Figure 6, we can see that in Gabon and US, items are sold as high as 25 USD on average. From Figure 7, we can see that Argentina is selling items at an average price of 25 BTC. However, the rest of the countries (including the Unknown) have selling prices close to 0. Due to the high conversion rate between BTC and USD, most of the items are selling in less than 0.5 BTC.

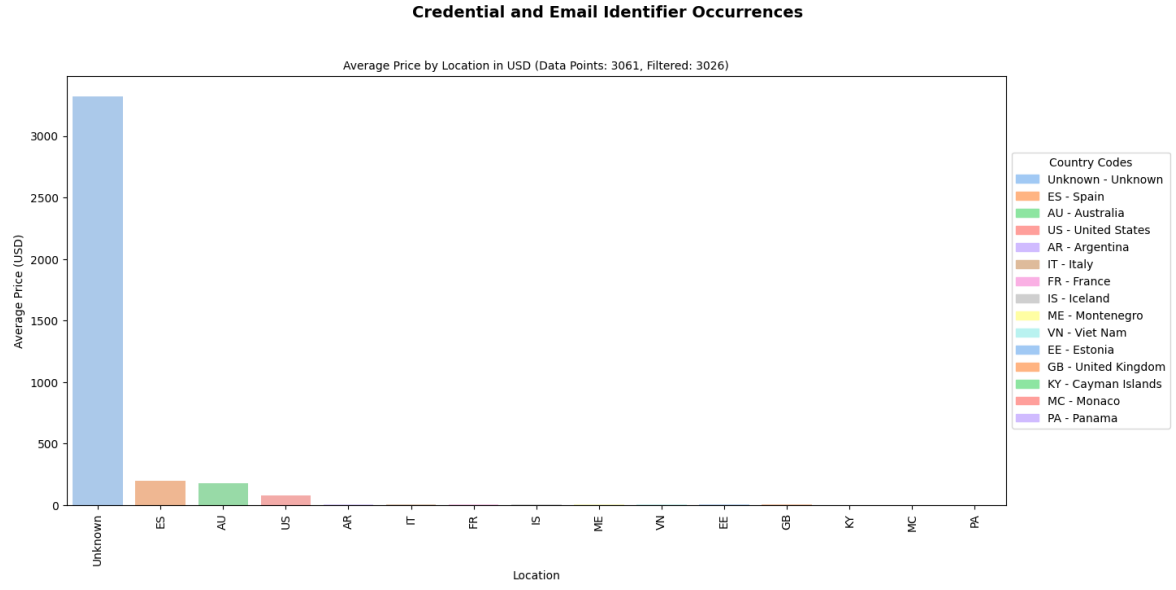

Figure 8: Average Price of Online Items Sold in USD (Including Unknown)

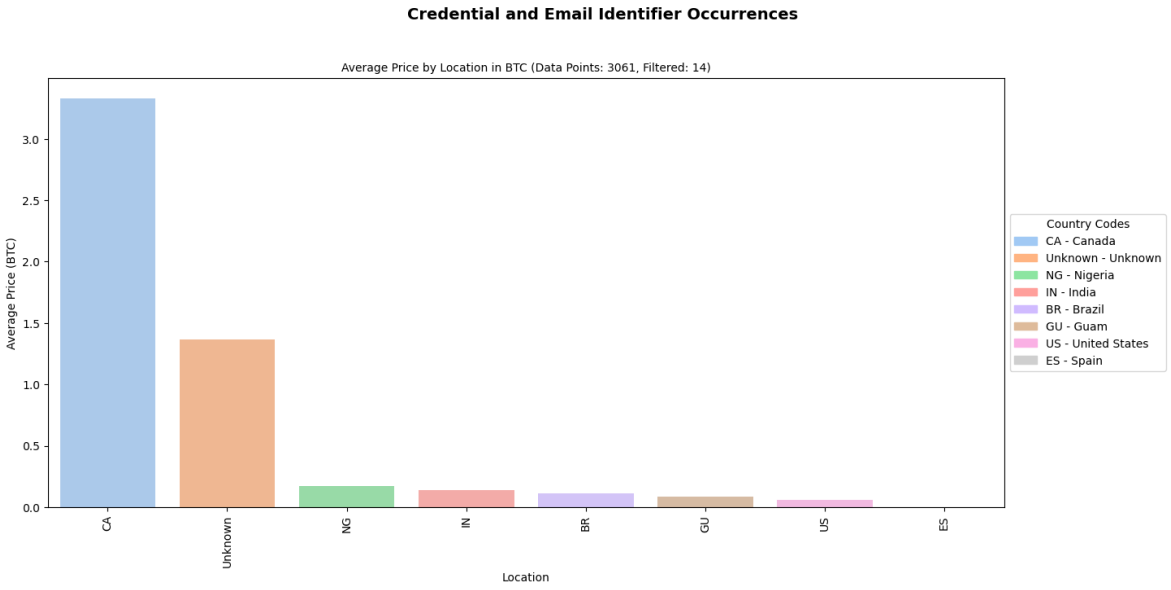

Figure 9: Average Price of Online Items Sold in BTC (Including Unknown)

In this category, a lot of the items are sold where I couldn't parse the location from the dataset. In total, there are 3061 data points and I have utilized 3026 + 14 = 3040 data points to generate figures 8 and 9. In the remaining 21 data points, I couldn't identify the currency correctly. Now, if we look at figure 8, we can see online credentials (emails, username, or different online account to different websites) are sold where location is Unknown on average more than 3000 USD. I hypothesize that online accounts rarely need to be from any specific locations and can be accessed from anywhere. As such locations are not marked in these product descriptions. However, I didn't do any complete study, nor did I do any count on how many data points I exactly have in this section, where location is Unknown. As such, I can only hypothesize at this stage. Similarly, in BTC (from Figure 9) we can products are sold up to 3 BTC on average in Canada, however a lot of the products are being sold between 0 – 0.5 BTC.

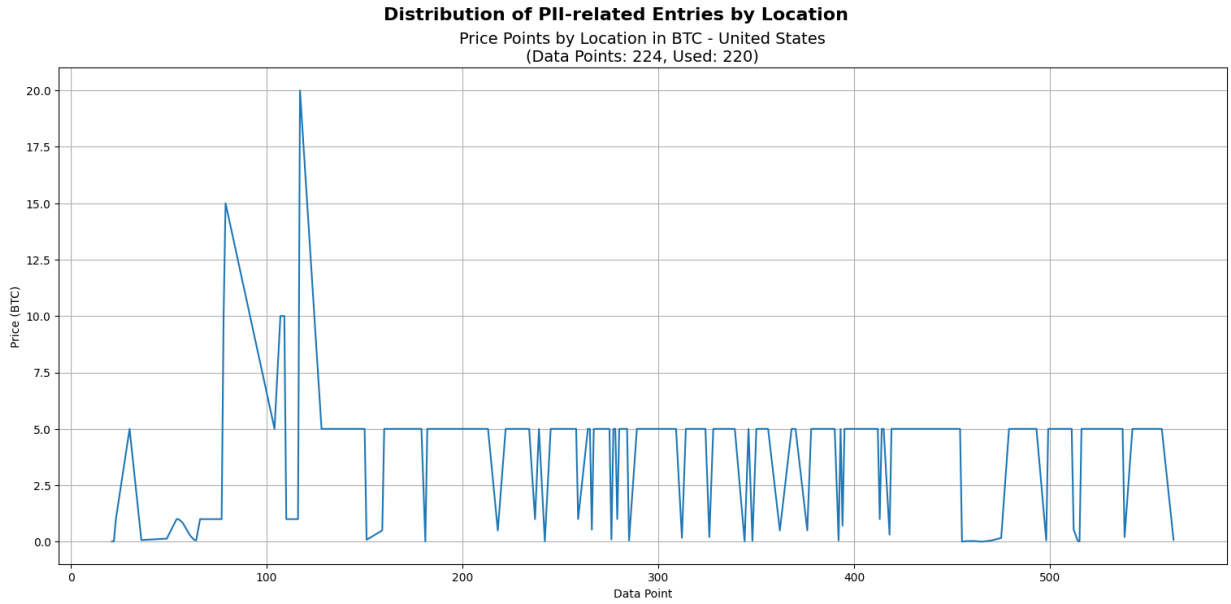

Figure 10: Price Point of Different Items in PII Category in BTC

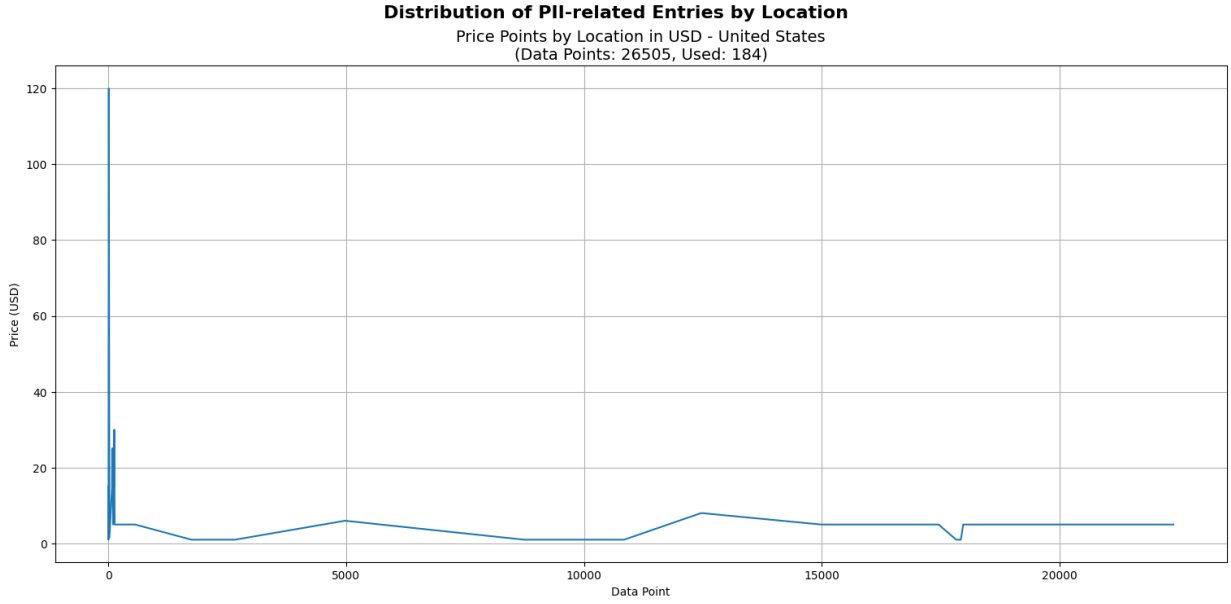

Figure 11: Price Point of Different Items in PII Category in USD

Figure 10 and Figure 11 is a line graph, where price point of PII items is shown from United States. There was no way to properly confirm when a specific product came into the market, as such X-axis of the graph is marked as data-point, when that specific product has been parsed from the file. This is a limitation of this study; this is visible for all three categories. Items sold in BTC have varied price range, visible from figure 10. If we look at figure 4, we will see Argentina has an average product price of 25 BTC. However, from figure 10, we can see that 220 products are sold in US locations, prices ranging from 20 BTC to ~>0 BTC. This is the shortcoming of taking mean/average into consideration. Similarly, if we look at figure 3, we can see NE (Niger) has average price of 7 USD for products sold in PII category.

However, if we compare Figure 11 and 12, we can see US (United States) has been marked as location in lot more category than NE (Niger), as such average has skewed the graph.

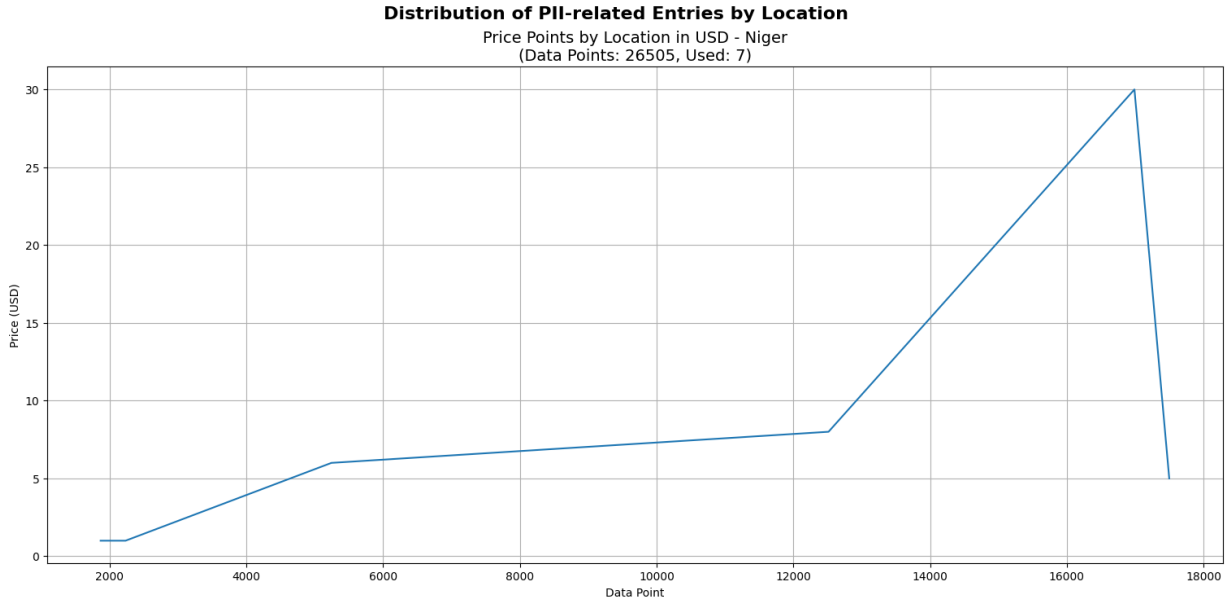

Figure 12: Price Point of Different Items in PII Category in USD

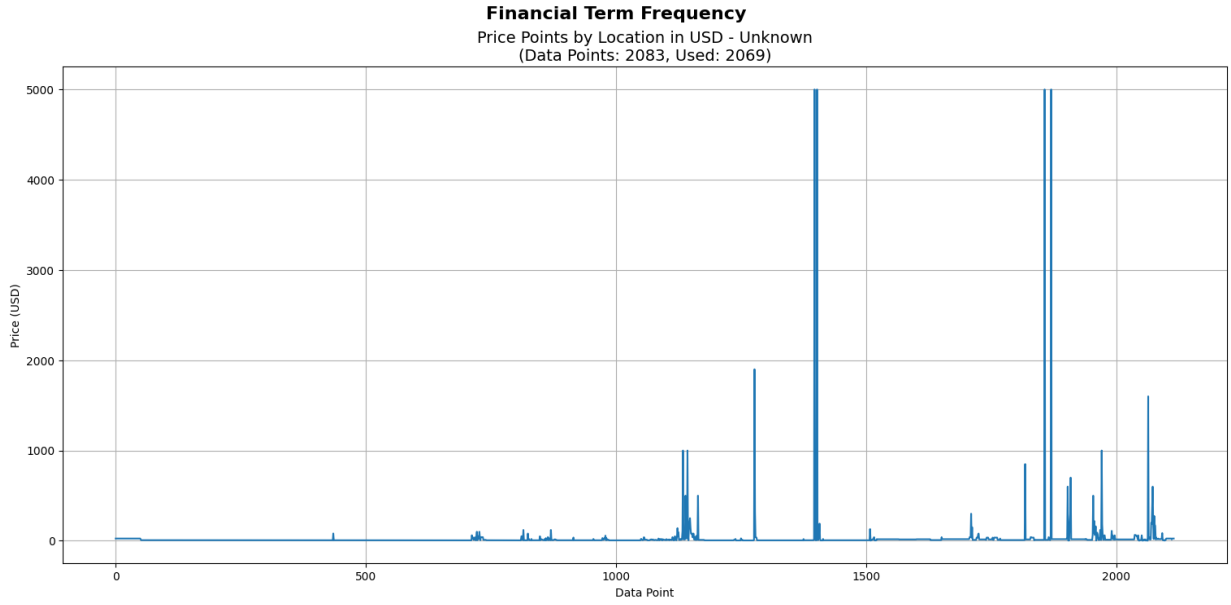

Figure 13: Financial Category Price range in USD, location Unknown

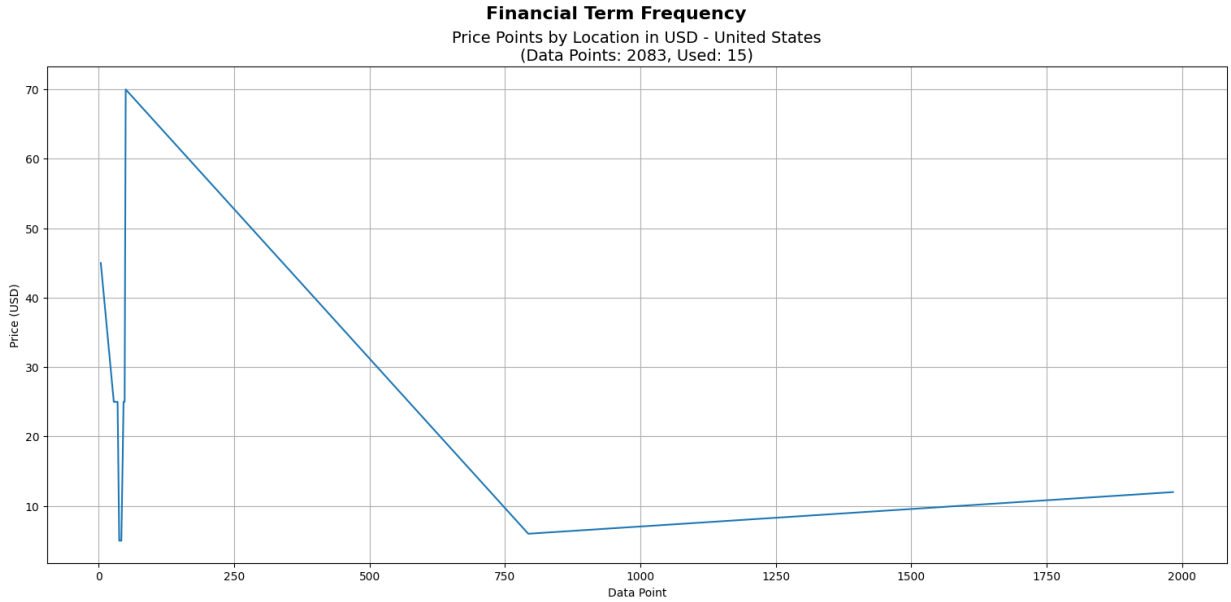

Figure 14: Financial Category Price range in USD, location US

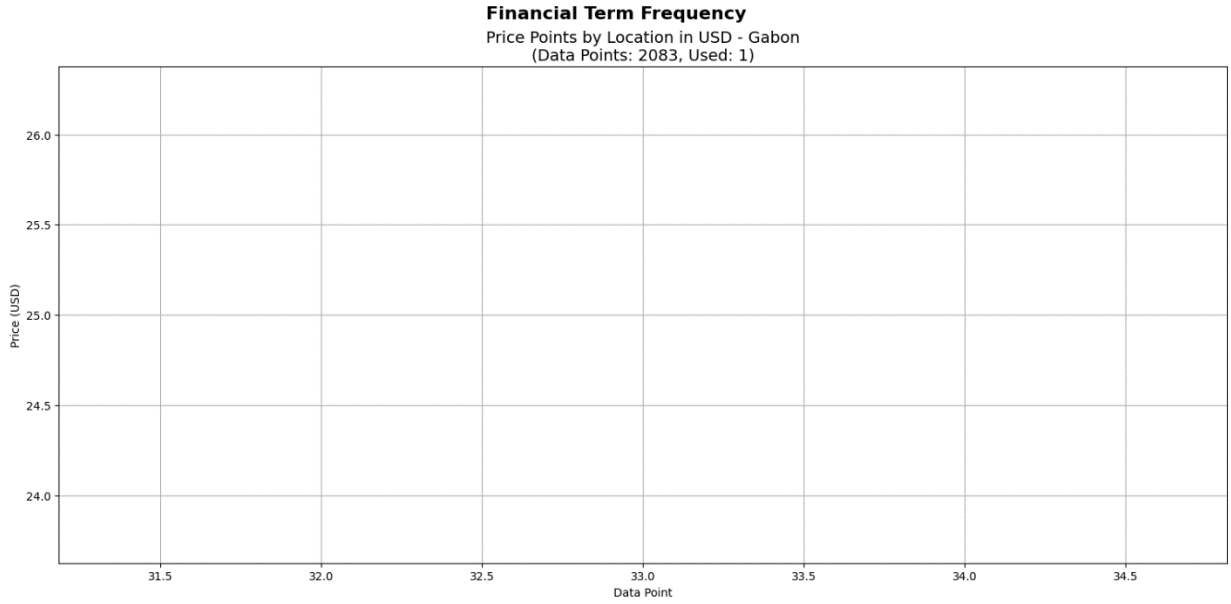

Figure 15: Financial Category Price range in USD, location Gabon

From figure 13, we can see, out of 2083 data points, we couldn't identify the location in 2069 of them where financial items are sold in USD. However, I would like to point out one thing categorization by location Unknown isn't working properly for the financial category, as such there are two different numbers of data points in figure 6 and figure 13, 14 and 15. However, if we look at fig 14, we can see US has 15 data points, and Gabon has 1. As such average price for products sold in USD from Gabon is 25 USD.

Victim Access Set

In this section, I found the original dataset a lot structural. As such there's not really any data cleaning involved. We identified the top 20 domains whose accounts have been compromised (Fig 16). Compared to the rest of the domains, accounts from Google, Facebook and Live (Microsoft) have been compromised more than 2x times.

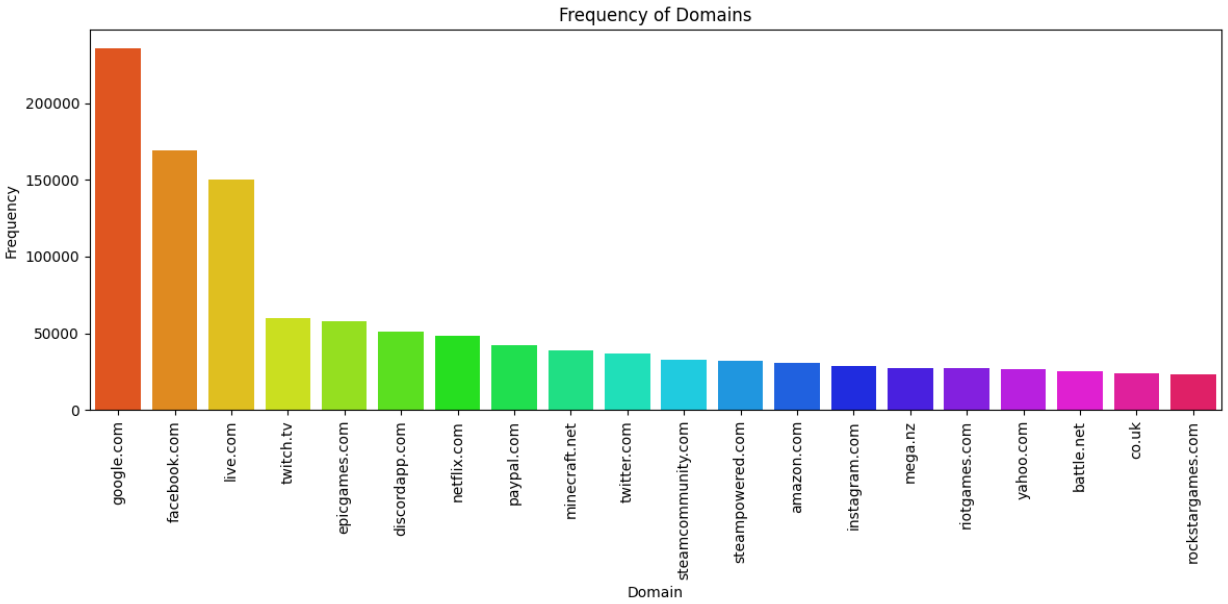

Figure 16: Top 20 Compromised Domains from Genesis Marketplace (2019 - 2022)

If we investigate the top 10 locations, from where Google.com accounts (Fig 17) have been compromised the most, we can see Italy is in top with about 25000 compromised accounts. Similarly, the Fig 18 shows France, Italy and Spain are the top 3 locations from where popular video game company, Riot Games account has been compromised the most. For the top 20s, in a lot of them accounts from Italy have either been compromised the most, or they are in Top-3.

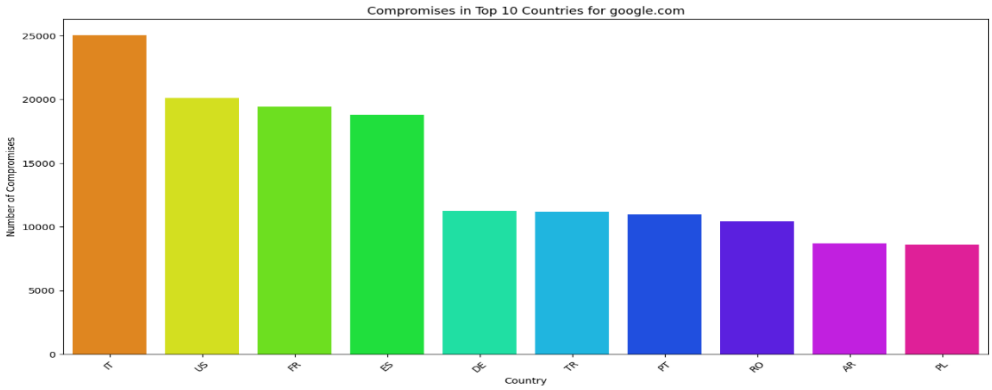

Figure 17: Top 10 Countries - Google.com

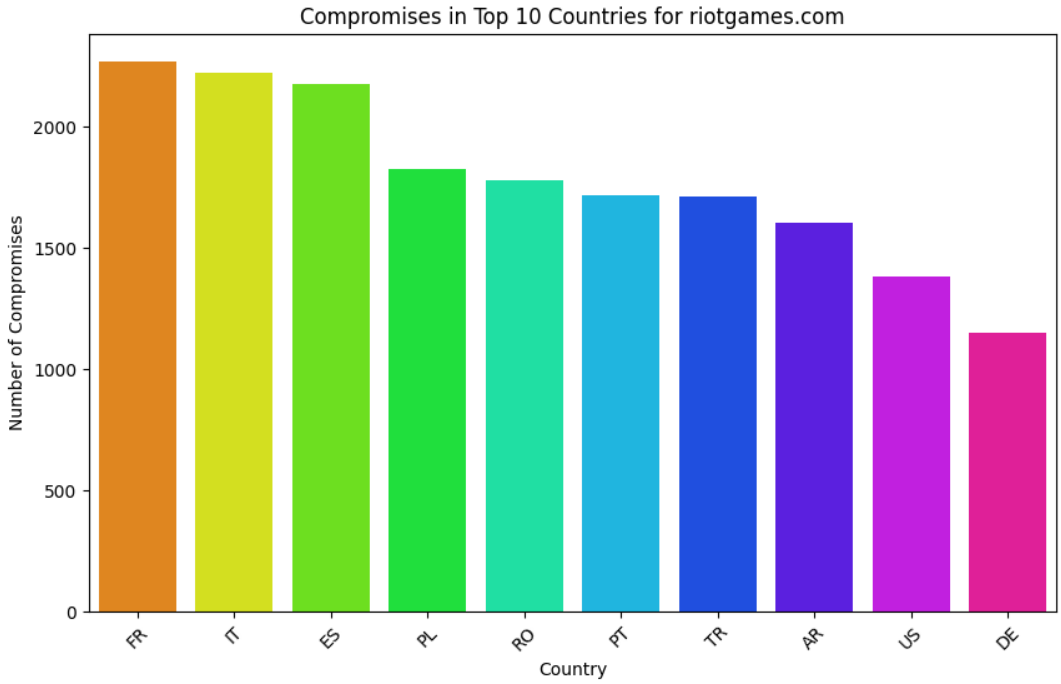

Figure 18: Top 10 Countries (Riotgames.com)

In Figure 19, we can see the average selling price for each of the Top 20 domains. The lowest price range is close to 1, and for some of the domains, the highest range goes up to 350. However, we can see the median price for all of them hover between 6 – 11. In this data set, currency is not mentioned. According to [7], user data can be sold for as low as 1 USD. By which standard, the median price we have found in this study is within range. It is worth noting that even though a lot of accounts have been compromised from Google, Facebook and Live, their median price is low around 6. On the other hand, PayPal, Riot Games, Battle. Net's accounts are sold off for between 9 – 11. Figure 20, 21, and 22 shows median price by country for Google, PayPal, and Riot Games account. As we can see, some values go over 40, however, for Google.com, the price drops top 20 after that, but for PayPal and Riot Games account, the price stayed up for other countries as well. There are lot of countries where these accounts are compromised from and it's really difficult to put them together in a single graph, as such I have only shown Top 10 countries by Median price.

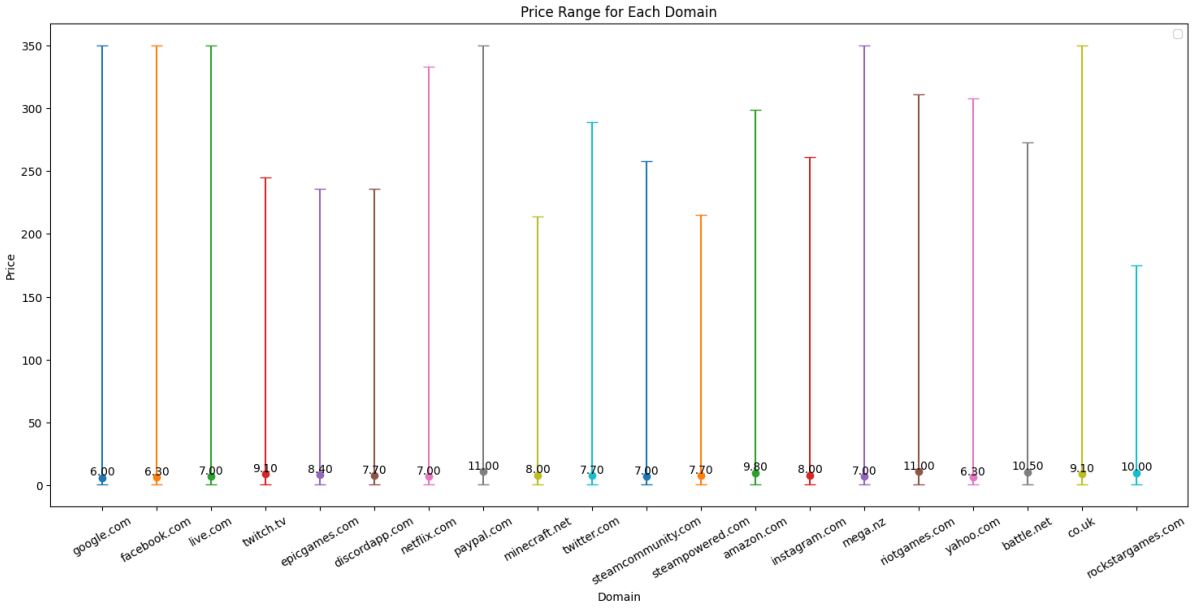

Figure 19: Median Selling Price of Top 20 Domains

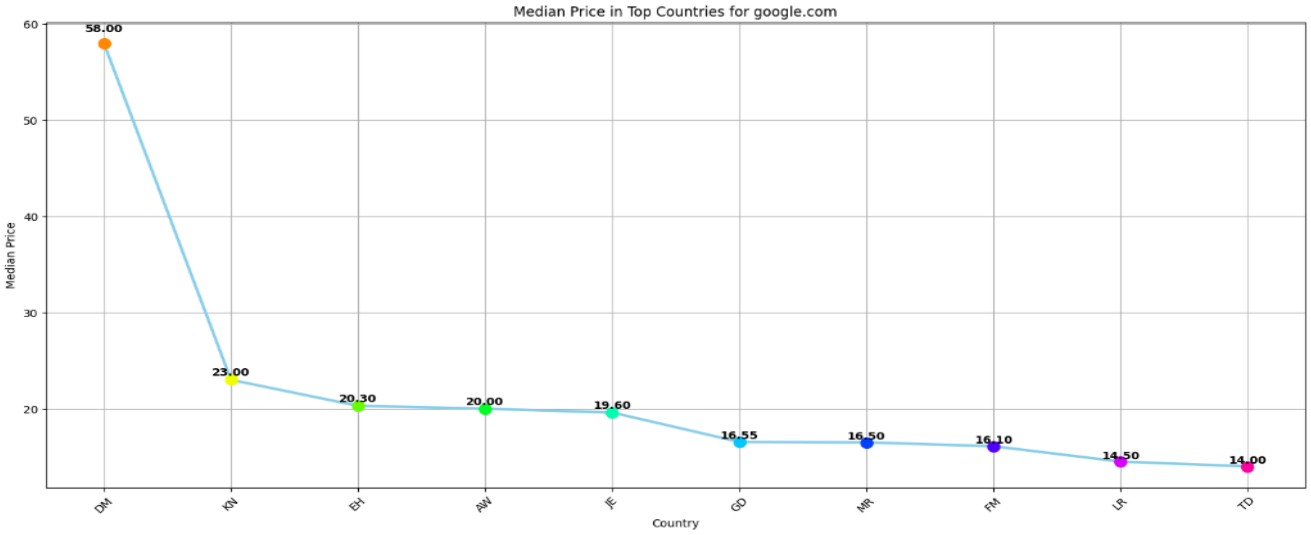

Figure 20: Median Price for Google.com in 10 Countries

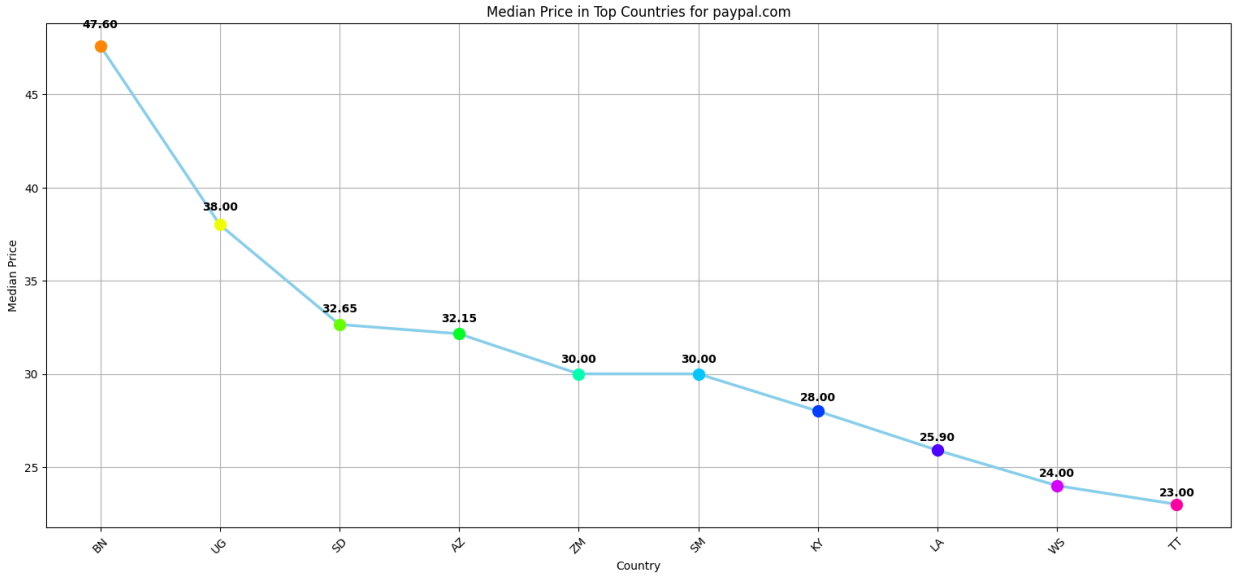

Figure 21:Median Price Top 10 Countries for PayPal.com

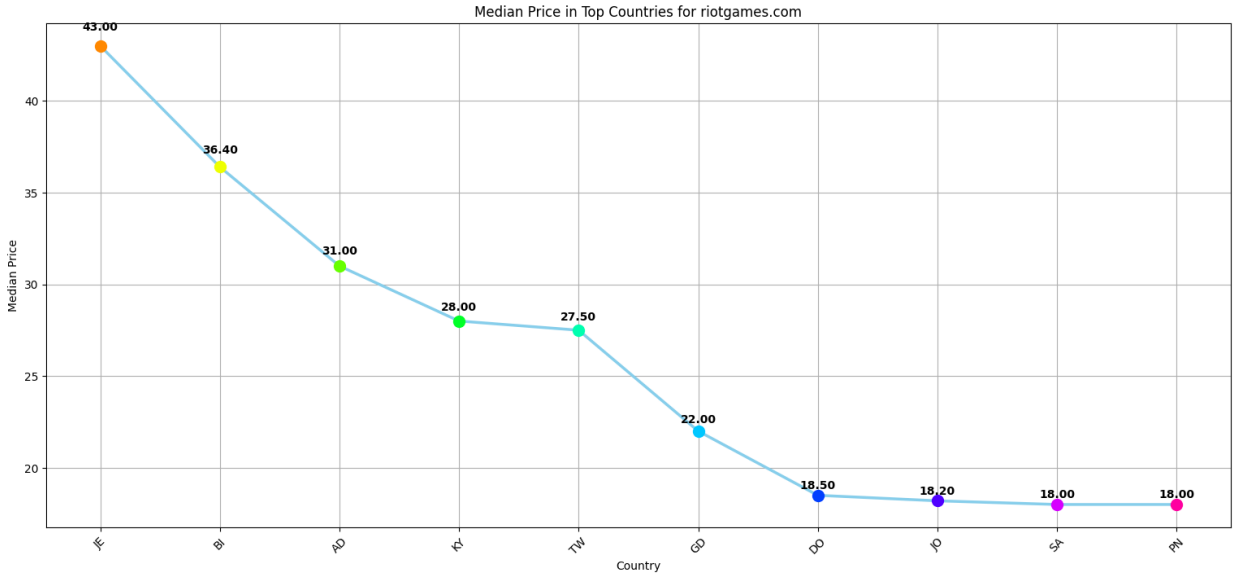

Figure 22: Median Price for Top 10 Countries for RiotGames.com

In fig 23, we can see the operating system distribution. Over 99% of them are some sort Windows Operating System. Figure 24 shows different top 20 different versions of Windows in the dataset. From there, I have broad categorized the OS based on their parent version. For example, Windows 10 Home, Windows 10 Pro, Windows 10 Enterprise all categorized into Windows 10. Figure 25 shows distribution of different Windows variant in the dataset, where 76% compromised user was using Windows 10, 6.9% on Windows 8/8.1 and 16.2% on Windows 7.

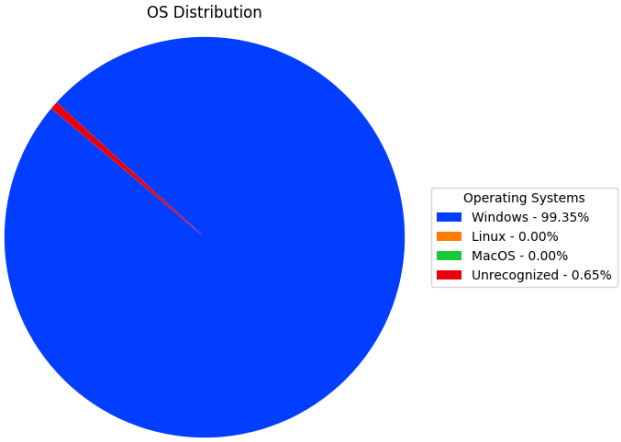

Figure 23: OS Distribution in Dataset

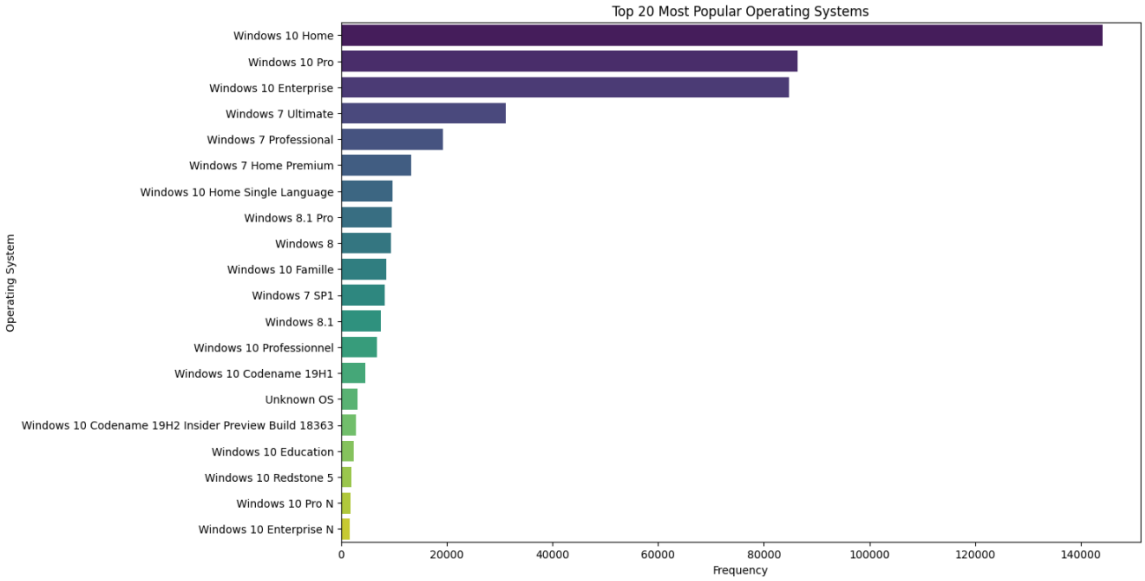

Figure 24: Top 20 Popular OS from the dataset

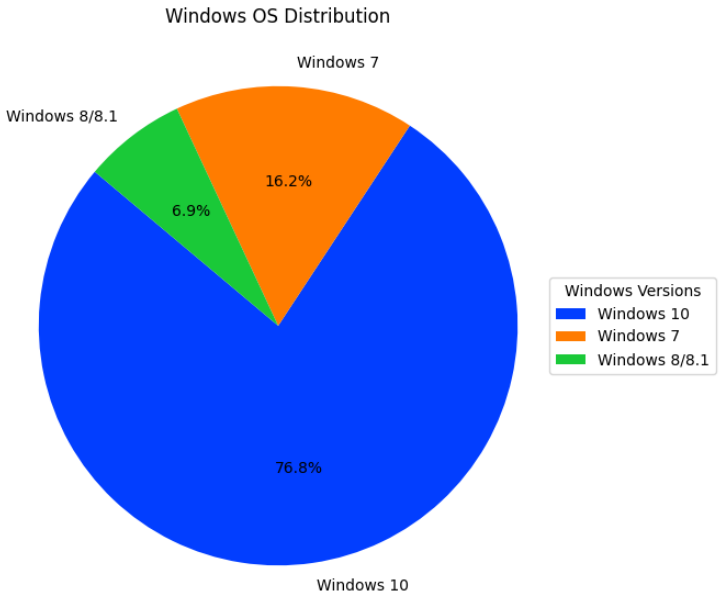

Figure 25: Distribution of Different Windows OS Variant in the dataset

I have also conducted studies on OS distribution for a specific domain. Instead of categorizing the OS, I have used the data (OS name/version) from the dataset directly. It appears that Windows 10 Home's users have been compromised the most (fig 26) for Google.com. From fig 27 and 28, we can see the similar story for both PayPal and Riot Games.

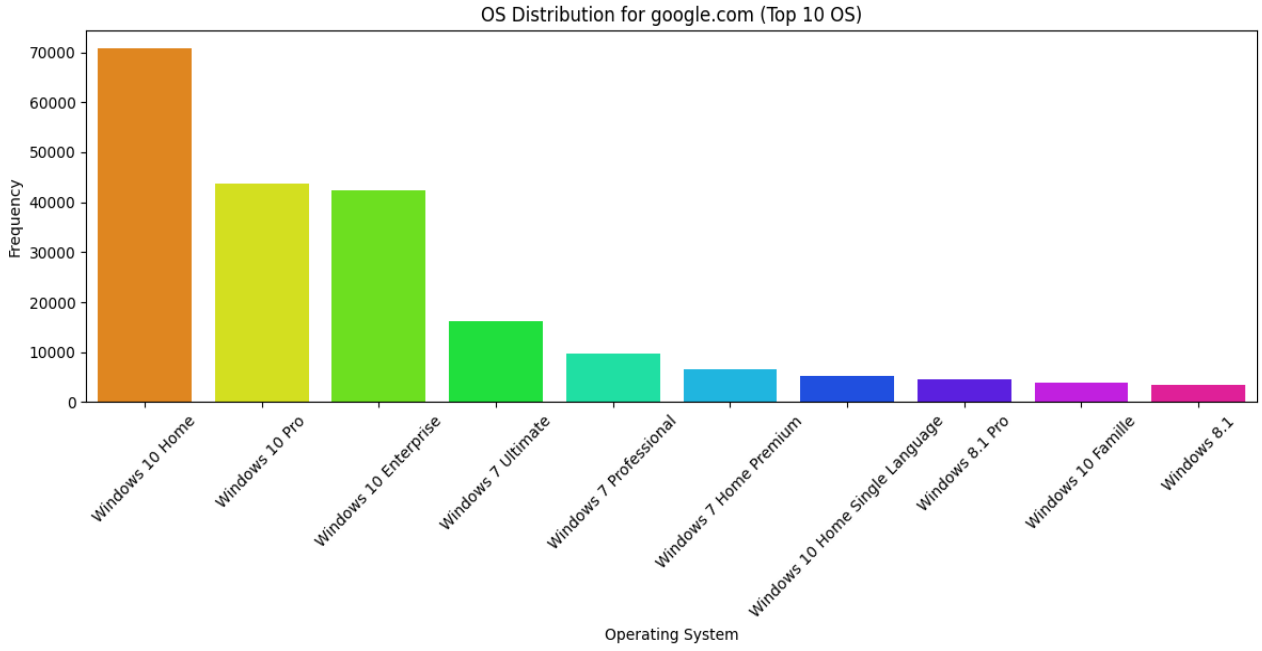

Figure 26: OS Distribution for a Specific Domain

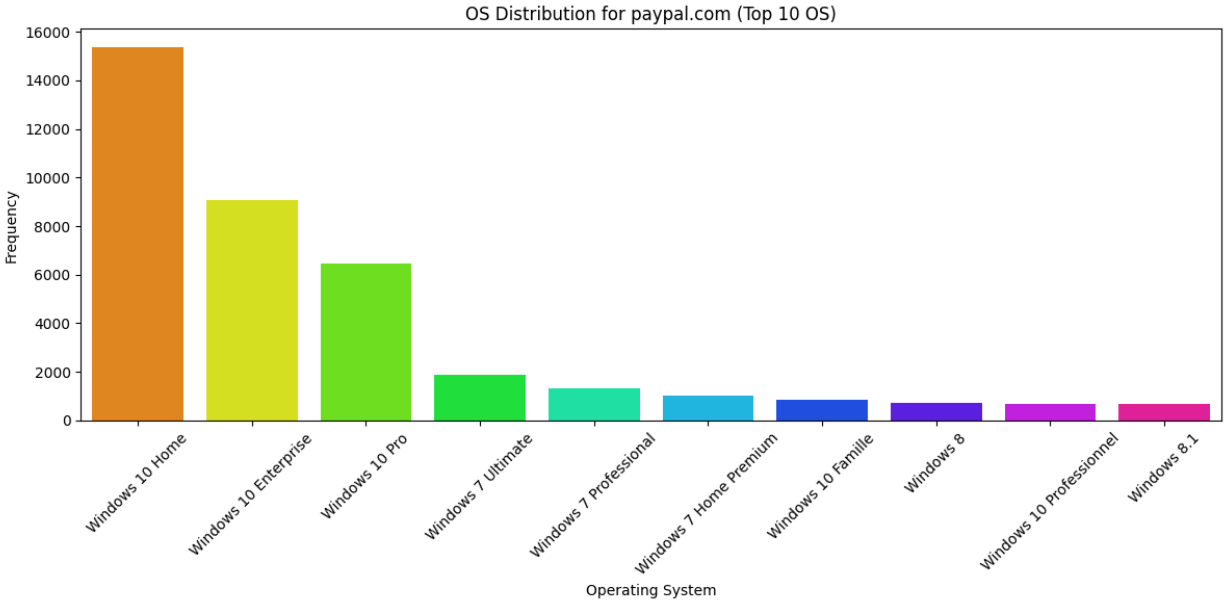

Figure 27: OS Distribution for PayPal

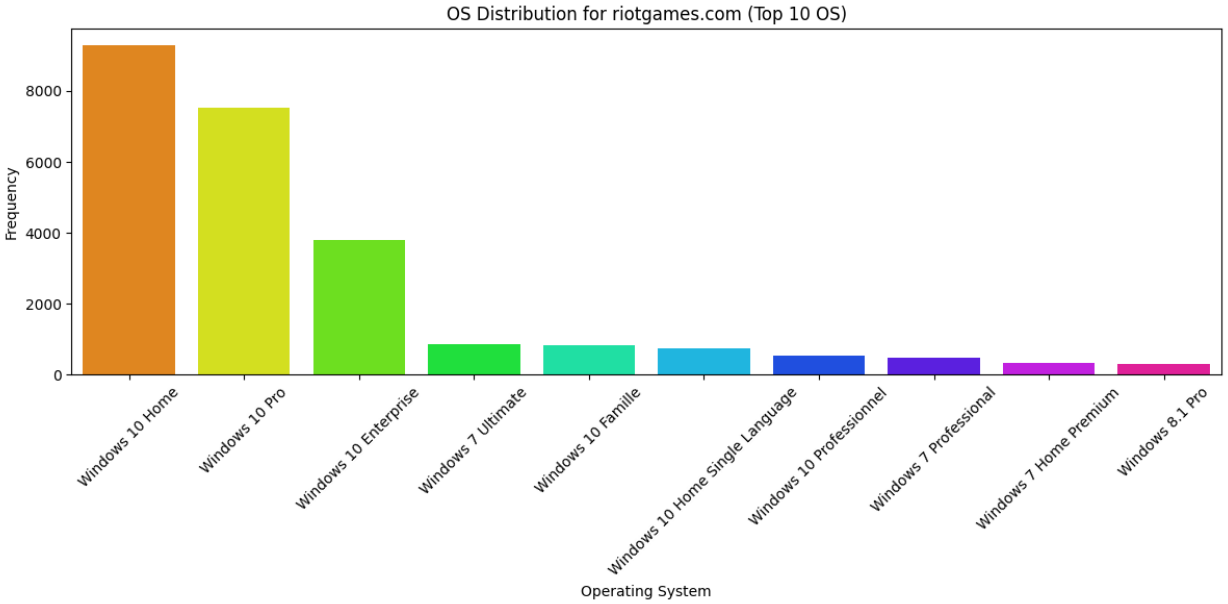

Figure 28: OS Distribution for Riot Games

Malware Infection Set

As I have previously stated, I have parsed through 24.65% of the data set. As such, the result from this section is not conclusive. I have broken down the data sets into multiple sections, and I will go through them one after the other. Unlike the previous data sets, due to the high volume of data, it was difficult to generate proper graphs, as such most of the results are in tabular format.

SUMMARY	
TOTAL FILE PARSED:	446235
TOTAL PERCENTAGE:	24.65% OF 1,809,988 DATA
USERNAME	5,338,894
PASSWORD	4,305,776
EMAIL	1,768,702
DOMAINS	8,756,779
SERVICES	9,939,847

Table 1: Summary of overall parsed Data

From Table 1, we can see the total amount of data. From 446235 files, I had around 5.3 million usernames, 4.3 million passwords, 1.7 million emails, 8.7 million URLs marked as domains and 9.9 million URLs marked as services.

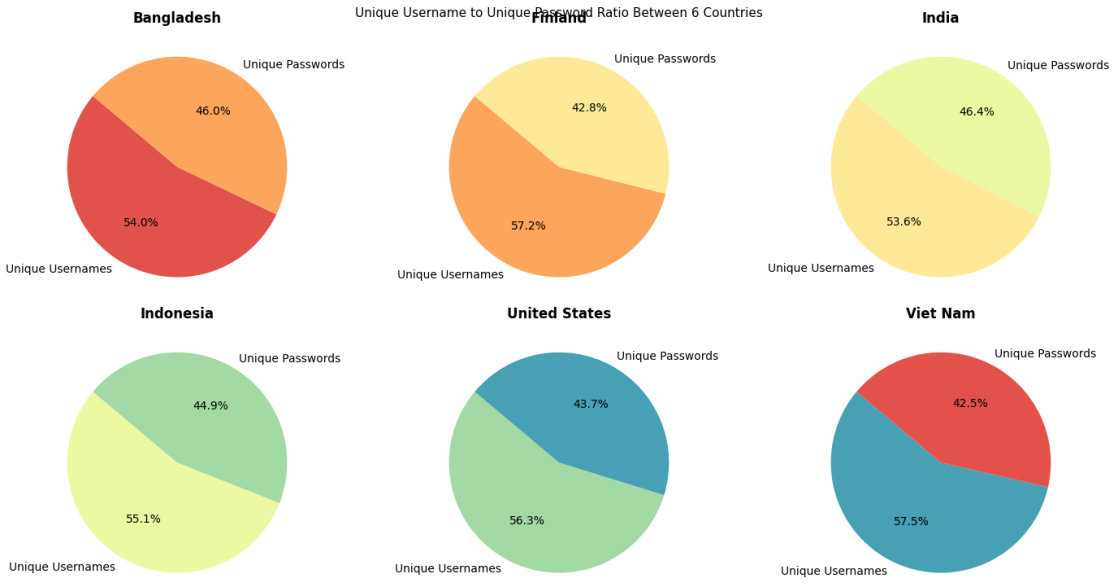

Figure 29: Unique Username, Unique Password Ratio between 6 Pre-Selected Countries

Figure 29 displays unique username and password ratio of 6 pre-selected countries, Bangladesh, Finland, India, Indonesia, United States, Viet Nam. These countries are chosen randomly, there's no significance here, apart from I am currently living in Finland, and I am from Bangladesh. As we can see from the data, in almost all countries the ratio hovers between 53-57%-42-46% between unique usernames and unique passwords. The original data set only contained unique usernames and passwords.

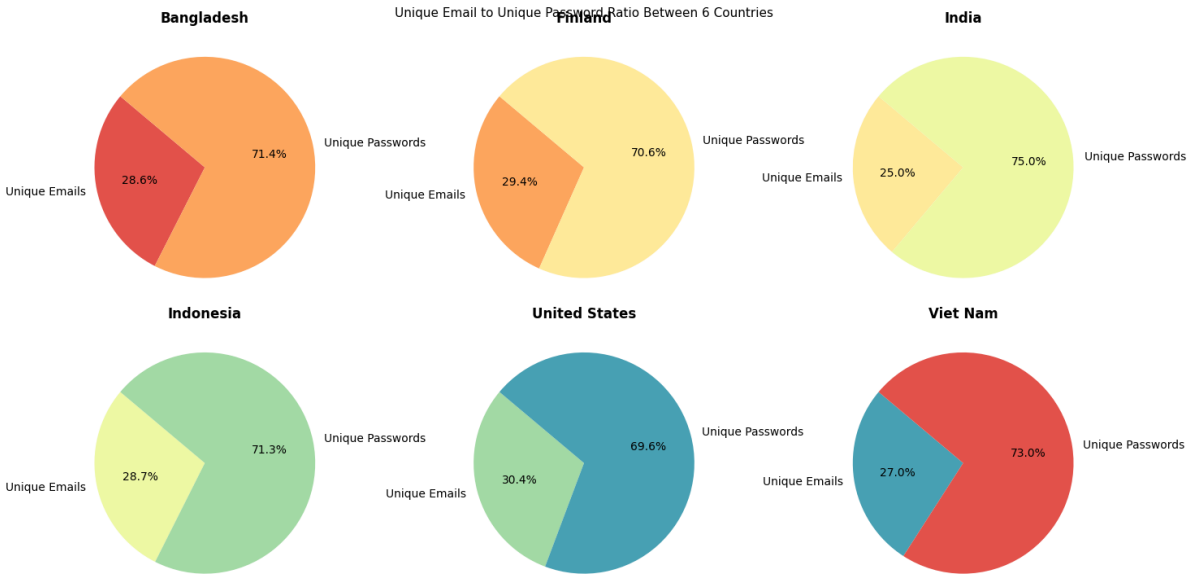

Figure 30: Unique Emails to Unique Password Ratio between 6 Pre-Selected Countries

Figure 30 describes the unique emails to unique password ratio of the same countries as figure 29. While both Fig 29 and 30 indicates higher diversity of unique passwords, especially between unique email to unique password, however, there's no way to determine how many passwords have been reused. Since the original data set only contained unique passwords. Humans tend to reuse passwords [7], but this graph does not show that property. It's only showing the property of this dataset.

Figure 31 shows the data set representation of top 10 countries. I have used "nlargest" function of pandas package, the columns are selected from ('unique_usernames', 'unique_emails', 'unique_passwords', 'unique_domains', 'unique_services'). One issue here is nlargest select the largest value from the first column, and if there's a tie, then it goes to the next column. As we can see from figure 31, all the top 10 countries have higher volume of compromised usernames in the dataset. As such, even though all 5 columns are supposed to be considered, only "unique_username" has been considered to generate this graph.

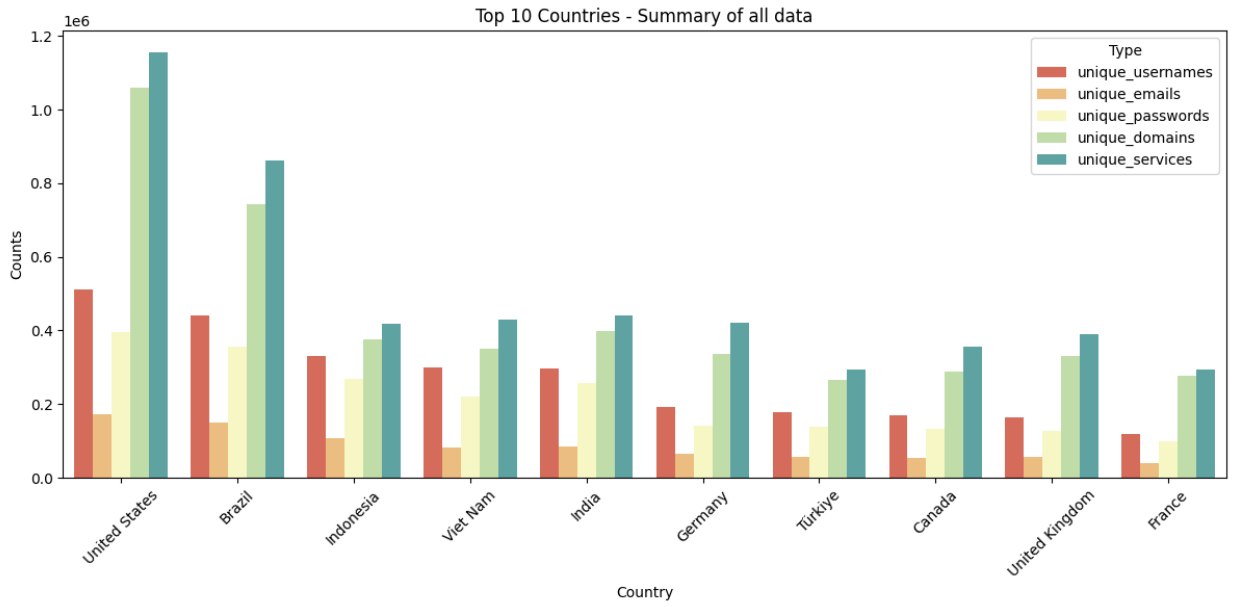

Figure 31: Top 10 Countries

Figure 32 shows the list of top 10 compromised services URL. As we can see, there are similarities between Fig 16 and Fig 32. We can match both fig 16 and 32s data against Top 50 most-visited websites (as of October 2023) [8], both lists are quite similar. Except both figures 16 and 32 contain some account information from video games like Rockstargames, Robolox, Riot Games, Epic Games or financial technological companies like PayPal. Based on the data we have seen in this study, we can say that accounts from these services (Videos Games, Financial Services) tend to sell at higher prices than other services.

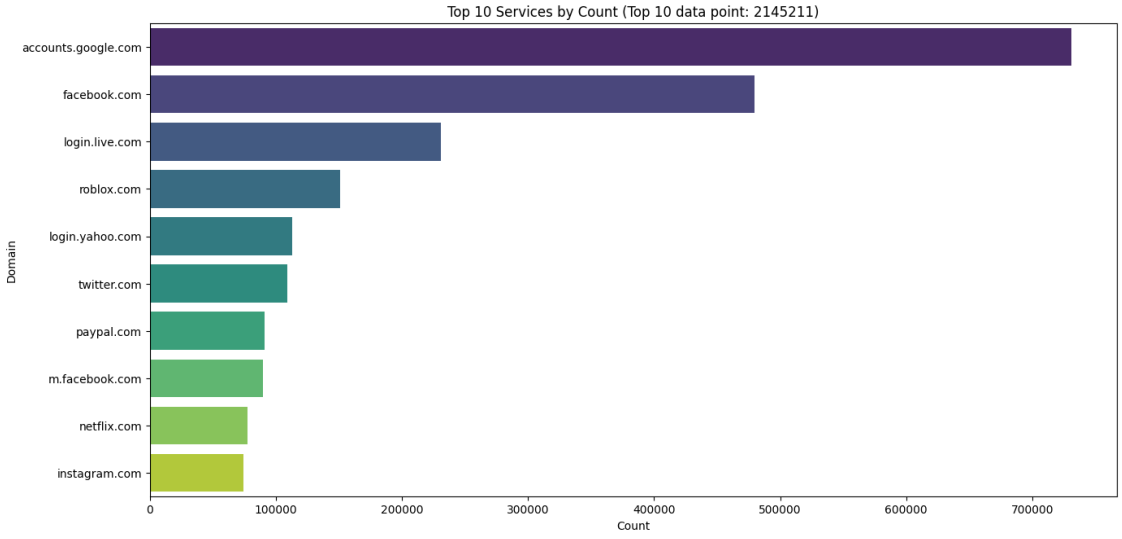

Figure 32: Top 10 Compromised Services

Appendix 1 shows a table of list of countries from where Top 10 domains are compromised from. There are 195 sovereign nations in the world, and some observer states, partially recognized, unrecognized states, dependent territories, and special administrative regions. Popular social networking site facebook.com covers 208 of them, followed by Google which has been compromised from 200 different regions.

Appendix 2 shows the number of TLDs that has been compromised (services) and their count from greatest to least. ".com" is the most common TLDs but there are some country specific TLDs whose services have been compromised a lot. For example, .com.br, .co.uk, .in, .ro, .es, .ir, etc

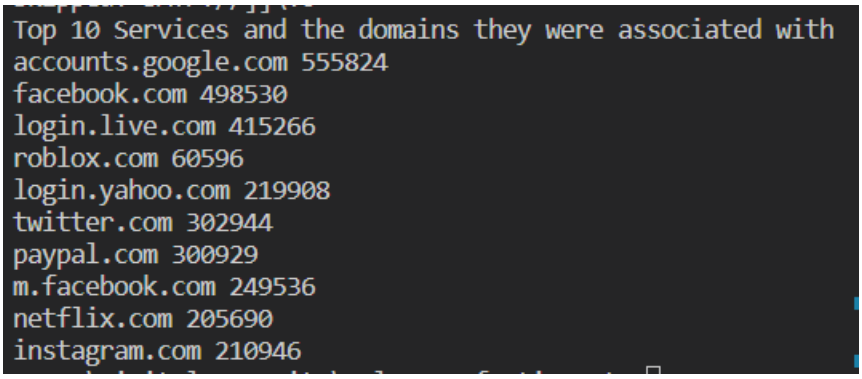

Figure 33: Top 10 Services and Number of Domains they are associated with.

Figure 33 shows how many separate domains the services are compromised for. However, this data does not tell any conclusive details, like how services and domains are connected.

Figure 34 and 35 displays data about how many credentials where service URL pattern matches *.fi or *.gov.bd. .fi is the country code top-level domain for Finland and .gov.bd is the country code top-level domain for Bangladesh. .gov indicates that it is intended for governmental entities. However, data shown in figures 34 and 35 is not conclusive, as we cannot identify which credentials are connected to which services.

Domain/Domain Pattern	Email Count	Username Count	Password Count
*.fi	5828	18364	15273

Figure 34: Service URL pattern and associated credential count (.gov.bd)*

Domain/Domain Pattern	Email Count	Username Count	Password Count
*.gov.bd	8291	31525	26461

Figure 35: Service URL pattern and associated credential count (.fi)*

Conclusion

The study effectively categorizes and analyzes data that was previously collected from various malware dump, online marketplace like genesis marketplace, and various other database marketplaces operating inside the Tor network. It identifies the prevalence of personal, financial, and online account information for illicit trading, financial fraud, and identity theft, and highlighting United States as a prime target for online attacks. The study also showcased price variations in different currencies and regions, emphasizing unique cyber threats in various countries. Although inconclusive, the ratio analysis of unique usernames to password and emails to password across countries provided a glimpse into the diversity and reuse patterns of digital credentials.

Reference

1. J. Nurmi, M. Niemelä, and B. B. Brumley, “Malware Finances and Operations: A Data-Driven Study of the Value Chain for Infections and Compromised Access,” in Proc. 18th Int. Conf. Availability, Reliability and Security (ARES '23), 2023, Article No. 108, pp. 1–12. [Online]. Available: https://doi.org/10.1145/3600160.3605047
2. https://www.wikiwand.com/en/OpenAI
3. https://www.wikiwand.com/en/Genesis_Market
4. https://2009-2017.state.gov/j/inl/rls/nrcrpt/2014/vol2/222695.htm
5. https://www.bloomberg.com/news/articles/2023-09-18/monaco-money-laundering-trial-bankers-accused-of-ignoring-red-flags?embedded-checkout=true
6. https://www.bbc.com/news/uk-65180488
7. https://www.securitymagazine.com/articles/92318-of-people-know-password-reuse-is-insecure-yet-75-do-it-anyway
8. https://www.wikiwand.com/en/List_of_most-visited_websites
9. https://www.wikiwand.com/en/Tor_(network)
10. https://www.torproject.org/
11. https://www.avast.com/c-tor-dark-web-browser
12. https://tb-manual.torproject.org/about/#:~:text=Tor%20is%20a%20network%20of,out%20onto%20the%20public%20Internet.
13. https://2019.www.torproject.org/about/torusers.html.en
14. https://www.wikiwand.com/en/Silk_Road_(marketplace)
15. https://go.recordedfuture.com/hubfs/data-sheets/malware-logs.pdf
16. https://www.wikiwand.com/en/Cryptocurrency
17. https://www.ivacy.com/blog/darknet-markets/
18. https://www.wikiwand.com/en/Darknet_market
19. https://flare.io/learn/resources/blog/dark-web-marketplaces/
20. https://www.wikiwand.com/en/GPT-4
21. https://platform.openai.com/docs/models/overview
22. https://openai.com/gpt-4

Appendices

Appendix 1: Top 10 Domains and Countries

Domains	Total Countries	Countries
facebook.com	208	San Marino, Qatar, Barbados, Côte d'Ivoire, Martinique, United Kingdom, Oman, ... (and others)
accounts.google.com	205	San Marino, Barbados, Qatar, Côte d'Ivoire, Martinique, United Kingdom, Oman, ... (and others)
login.live.com	200	San Marino, Barbados, Qatar, Côte d'Ivoire, Martinique, United Kingdom, Oman, ... (and others)
m.facebook.com	186	San Marino, Barbados, Qatar, Côte d'Ivoire, United Kingdom, Oman, ... (and others)
login.yahoo.com	184	Qatar, Barbados, Côte d'Ivoire, Oman, Poland, United Kingdom, Sudan, Congo, ... (and others)
twitter.com	184	Barbados, Qatar, Côte d'Ivoire, Oman, Poland, United Kingdom, French Guiana, ... (and others)
linkedin.com	181	Barbados, Qatar, Côte d'Ivoire, United Kingdom, Oman, Poland, French Guiana, ... (and others)
paypal.com	180	San Marino, Barbados, Qatar, Côte d'Ivoire, Martinique, Oman, Poland, United Kingdom, ... (and others)
amazon.com	179	Barbados, Qatar, Côte d'Ivoire, Oman, Poland, United Kingdom, Sudan, Congo, ... (and others)
instagram.com	178	Barbados, Qatar, Côte d'Ivoire, Oman, Poland, United Kingdom, French Guiana, ... (and others)

Disclaimer: This table has been generated by ChatGPT. The original content of the table has been generated by my code on the terminal. To display it properly, it has been inserted into ChatGPT to generate this table.

Appendix 2: Top Level Domains (TLDs) and their count

TLD	Count
.com	506426
.net	62189
.com.br	45501
.org	36104
.de	27218
.ru	24327
.fr	19261
.it	15543
.co.uk	15270
.in	11712
.pl	11376
.nl	10094
.edu	9619
.vn	8850
.ro	8745
.ca	8538
.es	8474
.hu	8302
.eu	7527
.ir	7394

Figure 36: Top Level Domains (TLDs) that has been compromised