Samuel Steinberg

November 20th, 2019

CS 366

Final Project: Scraping Twitter for Latest Cybersecurity Trends

Project Goal

The goal of this project was to analyze current cybersecurity trends, topics, and targets through crawling Twitter. It also finds relationships between types of cyber-attacks/topics and industries/sectors effected by cybercrime. With a world as busy and fast-paced as ours, Twitter is the perfect tool to analyze real-time trends in an ever-changing world, especially in the digital world. Home to hundreds of millions of users, it is the best tool available to see what cyber threats, trends, and targets are currently most popular worldwide. By extracting hashtags and tweet content with Twitter's own API, huge amounts of data are accessible as ever which this project took advantage of. Twitter was the ideal social media site to perform this project not only because of its vast API, but because content themes can easily be distinguished due to hashtags being a staple of the site.

Background

Before delving into the specifics of the project, there are a few terms and concepts that readers must be familiar with. First off, Twitter is a social media company based in San Francisco, CA and is the second most popular social media company and the seventh most visited website in 2019. Users can post, like, read, and retweet tweets from other users and Twitter handles over one billion search queries per day. Being up to date with what's being talked about on Twitter usually means being up to date with the latest trends in one's industry, hobby, etc. and allows for the gathering of real-time data and knowledge. The incredible amount of data from the site is all kept in storage by Twitter and is available to the public in the form of an API. An API allows applications to communicate with one another and is simply code that oversees the access point for the server (which then accesses the database). In other words, API's return data in response to a request by a client. Figure 1 is a visual provided via Medium.com:

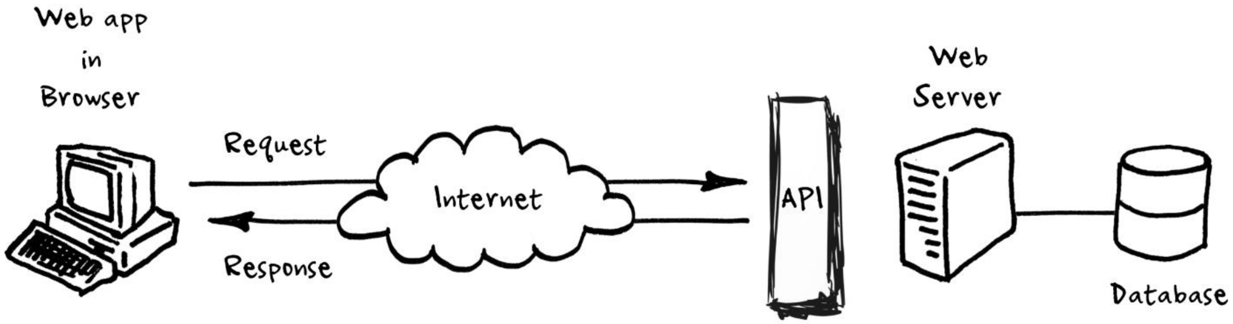
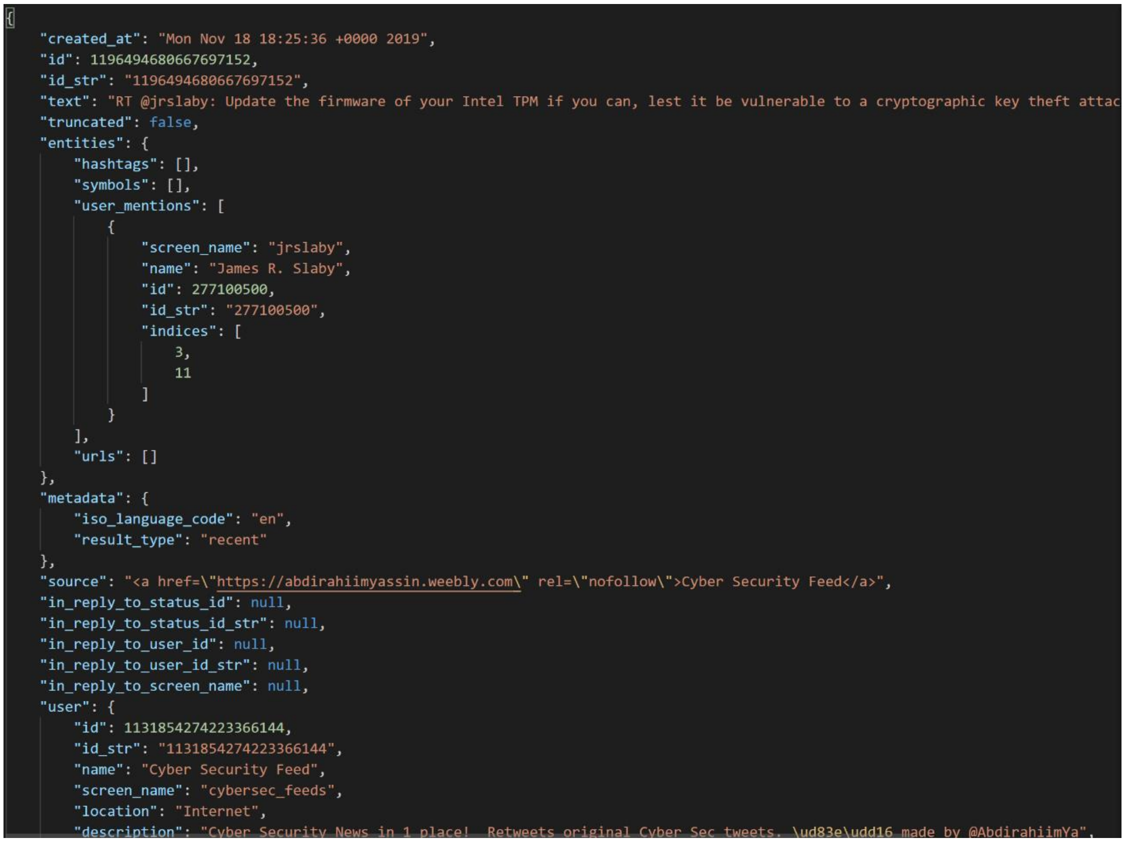


Figure 1: Courtesy of Medium.com, this graphic represents the communication between technologies occurring during an API request.

As can be obtained from Figure 1, the client also needs an API endpoint to receive the data. This will be explained in later sections of the paper, however the visual still serves to show the flow of communication. An API request is made by a client and is sent through the internet to the API trigger access to the web server to service the request with the database. This response is then returned with the reversed flow and is usually in XML of JSON form. These are human readable data documents/objects. The following is an example JSON response from this project. Note that it is only a small portion of the entire response:

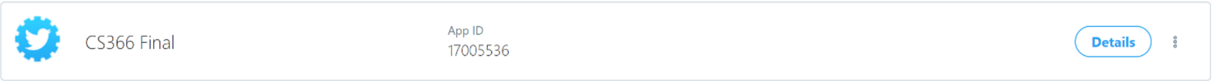


```
{
    "created_at": "Mon Nov 18 18:25:36 +0000 2019",
    "id": 1196494680667697152,
    "id_str": "1196494680667697152",
    "text": "RT @jrslaby: Update the firmware of your Intel TPM if you can, lest it be vulnerable to a cryptographic key theft attac
    "truncated": false,
    "entities": {
        "hashtags": [],
        "symbols": [],
        "user_mentions": [
            {
                "screen_name": "jrslaby",
                "name": "James R. Slaby",
                "id": 277100500,
                "id_str": "277100500",
                "indices": [
                    3,
                    11
                ]
            }
        ],
        "urls": []
    },
    "metadata": {
        "iso_language_code": "en",
        "result_type": "recent"
    },
    "source": "<a href=\"https://abdirahiimyassin.weebly.com\" rel=\"nofollow\">Cyber Security Feed</a>",
    "in_reply_to_status_id": null,
    "in_reply_to_status_id_str": null,
    "in_reply_to_user_id": null,
    "in_reply_to_user_id_str": null,
    "in_reply_to_screen_name": null,
    "user": {
        "id": 1131854274223366144,
        "id_str": "1131854274223366144",
        "name": "Cyber Security Feed",
        "screen_name": "cybersec_feeds",
        "location": "Internet",
        "description": "Cyber Security News in 1 place!  Retweets original Cyber Sec tweets. \ud83e\udd16 made by @AbdirahiimYa",
```

Figure 2: Sample JSON response from the Twitter API.

Overall Plan

The first step for this project was to get access to the API. This process involved applying for and receiving a Twitter Developer account and took around two days. Next, Twitter demands that an App is made so that API transactions can be noted and stored, and API keys and tokens necessary for the transactions are generated.

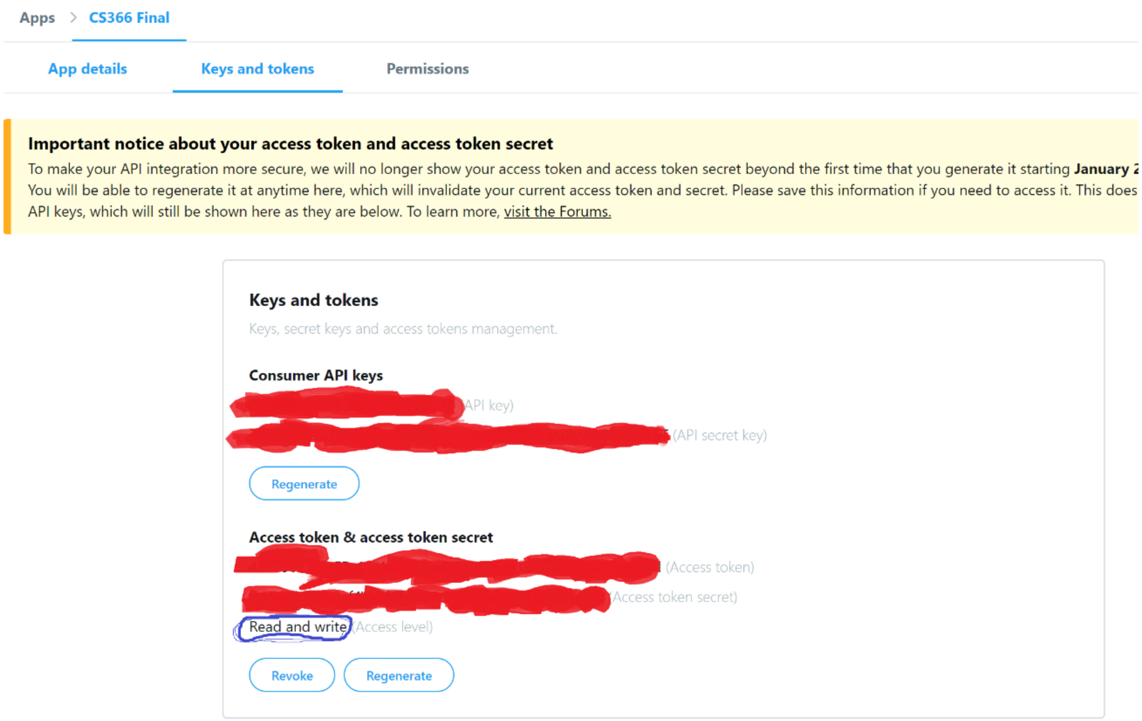


CS366 Final

App ID
17005536

Figure 3: Twitter Developer App Interface

An API key is a unique identifier to authenticate a user, developer, or program when calling the API. A secret API key proves the normal API is what it claims to be (normal key tells who you are, the secret key proves you are who you say you are). API tokens are unique identifiers of those requesting service to an application, which can then be matched to a stored one to authenticate. See Figure 4 below:



Important notice about your access token and access token secret

To make your API integration more secure, we will no longer show your access token and access token secret beyond the first time that you generate it starting **January 2** You will be able to regenerate it at anytime here, which will invalidate your current access token and secret. Please save this information if you need to access it. This does API keys, which will still be shown here as they are below. To learn more, visit the Forums.

Keys and tokens

Keys, secret keys and access tokens management.

Consumer API keys

[redacted] (API key)

[redacted] (API secret key)

Access token & access token secret

[redacted] (Access token)

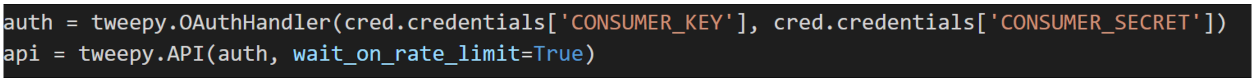
[redacted] (Access token secret)

Read and write (Access level)

Figure 4: Developer App information, including keys, tokens, and permissions. Keys and tokens are hidden for security purposes.

These API keys and tokens will be used for the client endpoint. Additionally, permissions are shown for the account, circled in blue in Figure 4. Even though read *and* write permissions were given for this App, only read were used since this project is mining data and not posting any.

After getting an account up and running, the plan was to first gather tweets based off specific themes. As previously mentioned, Twitter is ideal for gathering data specific to content themes/specifications due to heavy use of the hashtag and having their own Python package, tweepy. A hashtag (#) is a type of metadata tag allowing users to apply tagging to messages, easily allowing other users to discover them based off keywords. Twitter has their own Python package, which makes it easier to connect to the API and make requests (See Figure 5). As such, this was the approach taken. First, tweets were gathered based off of hashtag (and therefore specific topic) and then the message bodies would be extracted from the JSON returned from the API call and analyzed. Ultimately, this mined data would be the backbone of the project.

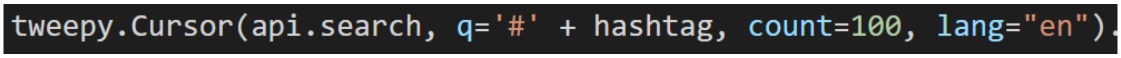


```
auth = tweepy.OAuthHandler(cred.credentials['CONSUMER_KEY'], cred.credentials['CONSUMER_SECRET'])
api = tweepy.API(auth, wait_on_rate_limit=True)
```

Figure 5: Twitter API authorization in code. Twitter's tweepy package makes connection and requests easy and simple to make.

Challenges

Most of the challenges to this project were involved in gathering meaningful data. Activating and getting the account and Twitter App up and running was a small barrier, but after these further challenges such as language, misspellings, and noisy data expressed themselves in full. Language was an issue that had to be faced head-on, since translations and different rhetoric differ from English. The solution to this was to extract only tweets written in English, which can be specified in the API call:



```
tweepy.Cursor(api.search, q='#' + hashtag, count=100, lang="en")
```

Figure 6: API call, with the "lang" field specifying the English language (en).

Misspellings were also a challenge. Since there is a limit of message characters on Twitter, users will often do whatever they can to restrict them while still getting their points across. This also means intentional (and unintentional) spelling mistakes, which can lead to the program missing these tweets since they do not contain any key cybersecurity terms. This problem is unavoidable, though *some* precaution can still be taken. For example, when searching for the popularity of SQL-injections, the following terms are some that a user might type:

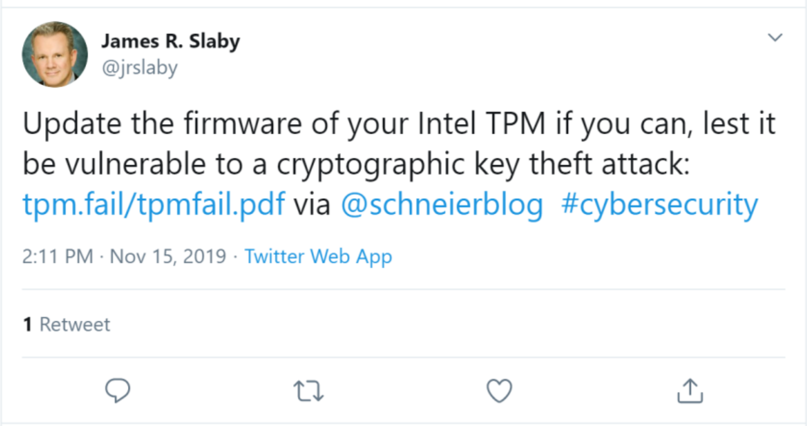
[SQL-injection, sql-injection, SQL injection, sql injection]

However, using the built-in Python method lower() will convert the Tweet text string to lowercase. This reduces the number of required searches by half, since now only “sql-injection” and “sql injection” will require a search, despite all four terms being flagged. This was repeated for all keywords.

A more challenging obstacle was noisy data. Since this project aims to find the relevancy of cyber trends, attacks, and targets each word in each tweet holds significant weight. While generally uncommon, there are “spam” tweets with multiple cyber terms embedded, though not really contributing anything relevant. This noise was unavoidable, though it would be potentially damaging to suppress since there are also many legitimate tweets with multiple terms making significant contributions. An additional challenge was finding actual trending attack types, for example, a trojan horse virus is usually used in ransomware attacks, which is itself a sub-group of malwares. This potentially led to more hits on malware and ransomware as opposed to the actual viruses and vulnerabilities.

Approaches and Techniques

The approach taken in this project was to write an algorithm to target tweets by content theme, and then to iterate over the mined tweets returned from the API while searching for key words of interest to create a dictionary of labeled data. While this approach might seem novel at first, after reviewing hundreds of tweets it turned out to be quite an accurate indicator. In other words, the extracted tweets were mostly meaningful (See Figure 2 text field) and fit the project goal:



James R. Slaby
@jrslaby

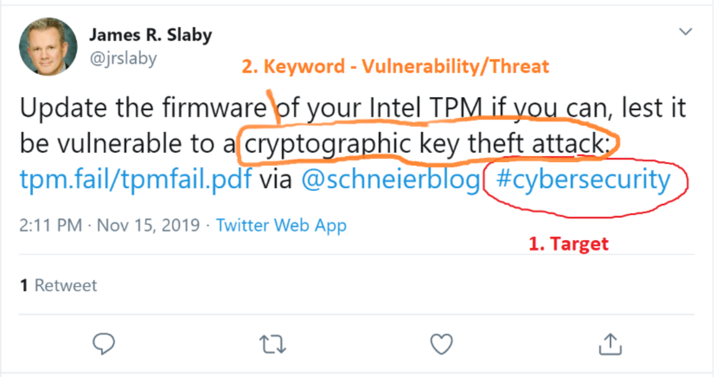
Update the firmware of your Intel TPM if you can, lest it be vulnerable to a cryptographic key theft attack: tpm.fail/tpmfail.pdf via @schneierblog #cybersecurity

2:11 PM · Nov 15, 2019 · Twitter Web App

1 Retweet

Figure 7: Example of mined tweet, the actual tweet that the JSON from Figure 2 represents.

In Figure 7, the tweet was targeted by its hashtag, "cybersecurity". After being flagged as containing a hashtag of interest, the tweet is searched for types of vulnerabilities and cyber threats. This is found in the "cryptographic key theft attack" substring within the text of the tweet. This algorithm allowed current cybersecurity trends, topics, and common attack targets to be recorded and are a representation of their popularity. Since the Twitter free access API only allows users to search the previous week, time allowed for two weeks of mining.



James R. Slaby
@jrslaby

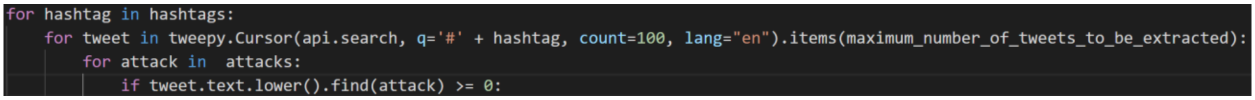
Update the firmware of your Intel TPM if you can, lest it be vulnerable to a cryptographic key theft attack: tpm.fail/tpmfail.pdf via @schneierblog #cybersecurity

2:11 PM · Nov 15, 2019 · Twitter Web App

1 Retweet

Figure 8: Algorithm on sample tweet

Figure 8 shows the effect of the algorithm on the same mined sample tweet from Figure 7. To reiterate, the tweet is targeted for a search with the "cybersecurity" hashtag. The search will find the "theft" substring and record it as a statistic.

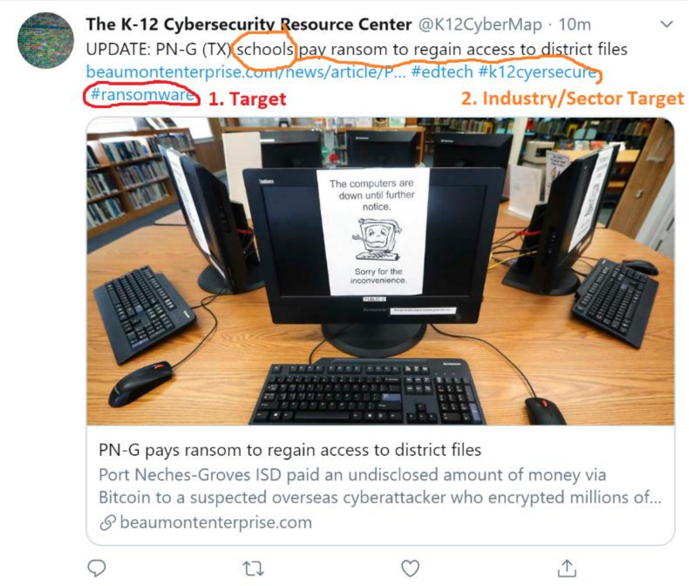


```
for hashtag in hashtags:
    for tweet in tweepy.Cursor(api.search, q='#' + hashtag, count=100, lang="en").items(maximum_number_of_tweets_to_be_extracted):
        for attack in  attacks:
            if tweet.text.lower().find(attack) >= 0:
```

Figure 9: Main search loop in source code.

As shown in Figure 9 above, for each hashtag (subject of tweet) a maximum number of tweets are mined, which for this project was set to 50,000 to prevent being banned but to still gather sufficient data. For each of the 50,000 tweets maximum caught with the hashtag, the text of the tweet was searched for a number of keywords. These keywords, stored in a list called "attacks" are searched with the find() Python function. This function *does* return true when substrings are found, however for the sake of this project it was not an issue. This is because the tweets found are already within the topic of cybersecurity, and there will not be many substrings of attacks found, if at all. This makes it highly unlikely that a substring such as "buffer overflow" will be found as a substring in a tweet within the realm of cybersecurity and not be relevant to the topic. Each attack type found in the text will be counted and added to a dictionary, which is saved after all iterations are completed.

The next step was to save the results stored in the dictionary as a text file for future data visualization and to create chart visuals. These allowed trends to be exposed and conclusions about the most trending and recent flaws and vulnerabilities. This process was repeated to find most common attack targets. In a day of the Internet of things and really most daily processes rely on the Internet. This makes industries such as healthcare, finance, education, and energy massive targets for hackers. There are large amounts of vulnerable personal and financial data stored by companies with security flaws in addition to services vital to the successful operation of the industry. The program was constructed to be able to go between querying common attacks and common targets by allowing the user to an option to choose before the main loop.



The K-12 Cybersecurity Resource Center @K12CyberMap · 10m

UPDATE: PN-G (TX) schools pay ransom to regain access to district files beaumontenterprise.com/news/article/P... #edtech #k12cyersecure #ransomware

1. Target

2. Industry/Sector Target

The computers are down until further notice.
Sorry for the inconvenience.

PN-G pays ransom to regain access to district files

Port Neches-Groves ISD paid an undisclosed amount of money via Bitcoin to a suspected overseas cyberattacker who encrypted millions of...

beaumontenterprise.com

Figure 10: Algorithm searching for targeted industries/sectors

Tools

Tools used in this project included the Twitter API, Twitter dev tools, Visual Studio Code, and Python with packaged libraries. The Twitter API and its tweepy package provided easy access to their database and delivered clean and easy-to-process results. This allowed data pre-processing to be at an absolute minimum.

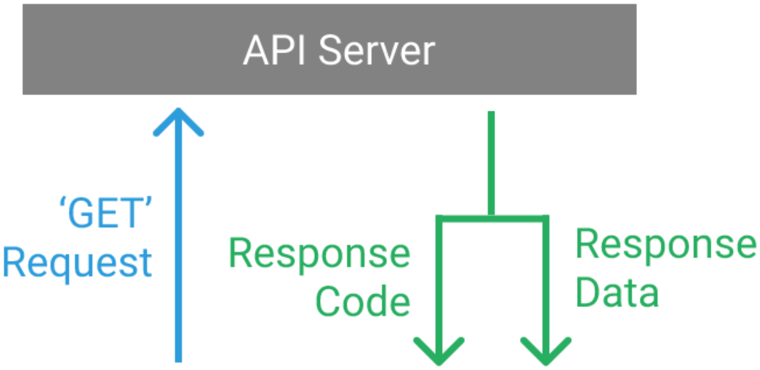
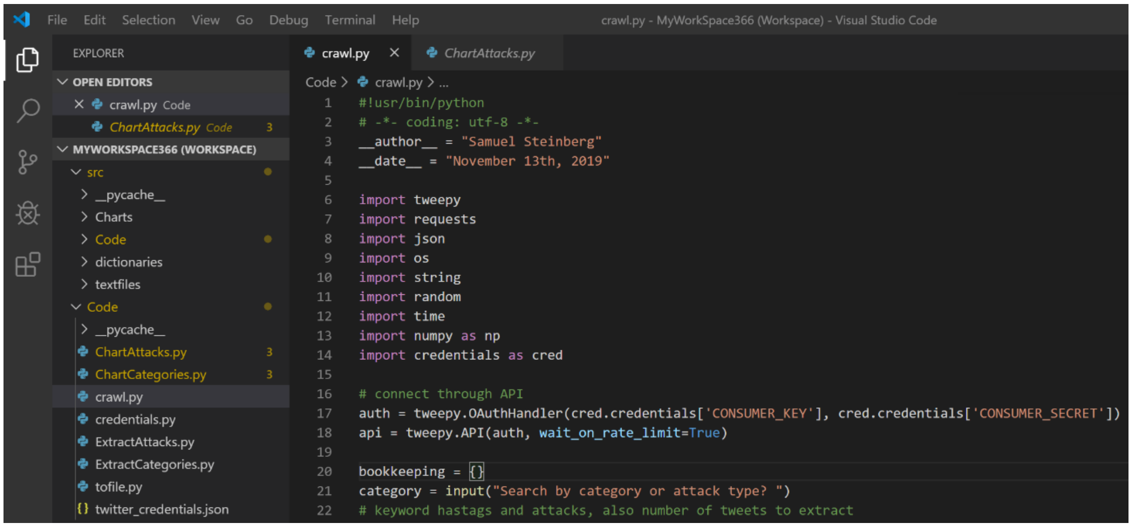


Figure 11: Simple REST call. This project only used the GET method to request data.

Figure 11 represents a simplified version of the GET method to request data, which Twitter's tweepy package handled. The JSON object returned from the call was easy to parse and was well organized. Twitter dev tools allow users to create an App and make rate-limited free calls to their resources. It is possible to go back further than one week with a subscription, though for the scope of this project it was not necessary. They offer secure authorization and a user interface to monitor account activity.

The code was developed on Visual Studio Code using Python. Visual Studio Code is an IDE developed by Microsoft for Windows, Linux, and MacOS and is one of the most widely used IDE's available. It allows for version control and the download of plugins for stable development, so it was ideal for this project.



```python
#!usr/bin/python
# -*- coding: utf-8 -*-
__author__ = "Samuel Steinberg"
__date__ = "November 13th, 2019"

import tweepy
import requests
import json
import os
import string
import random
import time
import numpy as np
import credentials as cred

# connect through API
auth = tweepy.OAuthHandler(cred.credentials['CONSUMER_KEY'], cred.credentials['CONSUMER_SECRET'])
api = tweepy.API(auth, wait_on_rate_limit=True)

bookkeeping = {}
category = input("Search by category or attack type? ")
# keyword hastags and attacks, also number of tweets to extract
```

Figure 12: VS Code interface and structure

Additionally, Visual Studio Code provides a terrific user interface. Files and file folders are easy to locate and are hierarchal, allowing for easy organization. Here, the project was put into a Workspace folder (a sort of container used by the IDE) and individual folders for generated charts, dictionaries, text files, and source code are placed inside. Python was the language of choice for this project because it works well with API's, has extensive library support and is easy to modularize. This is in addition to Twitter having a custom said Python-specific library, tweepy. Python's extensive library support includes packages such as matplotlib, allowing for custom charts and graphs to be programmed.

Results

Statistics were recorded for batches of 50,000 tweets at a time. This paints a fair, large picture while still being of a large size. Counts were taken for each attack type found and were analyzed to find trends and popular attacks. Unsurprisingly and noted in the Challenges section, larger category of attack sometimes overshadowed specific attacks. The statistics were gathered over a two week period, November 5th to November 20th, and were compared against one another to find trends in popularity.

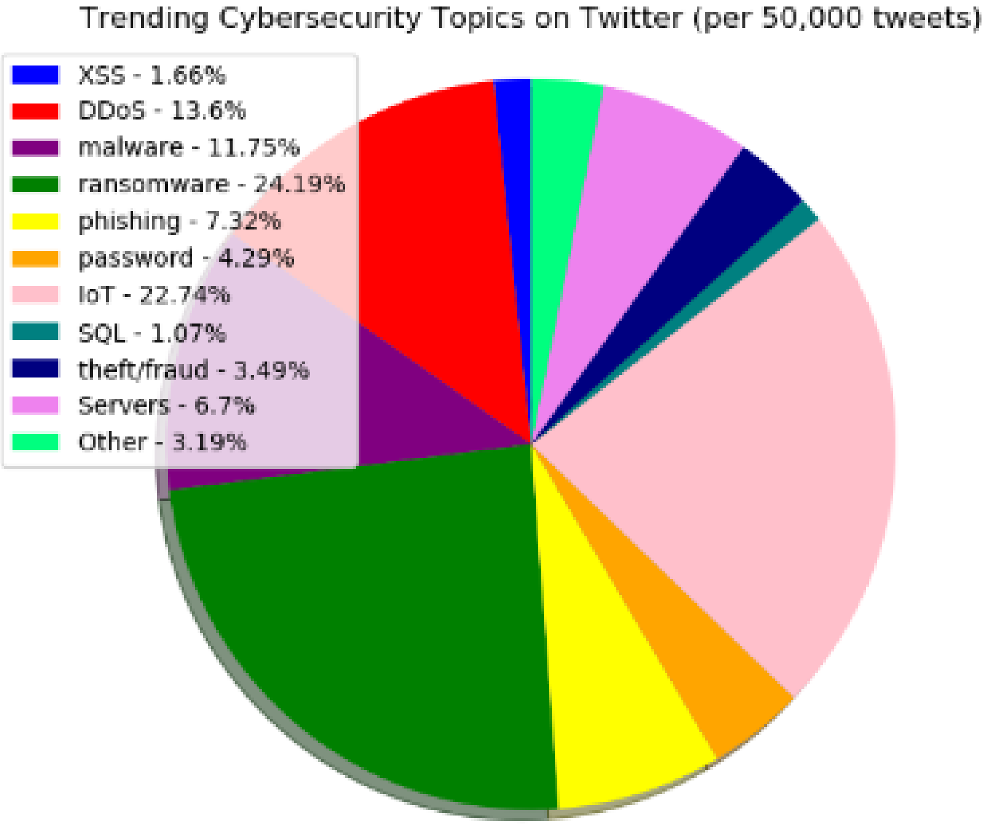


Figure 13: November 5th - 12th most trending cybersecurity attacks

Figure 13 is a visual representation of the most popular attacks during the first week. The majority of the trending attacks situated around malware, ransomware, DDoS, and IoT attacks.

This isn't a surprise and made sense, since these attacks are both increasingly popular and difficult to detect and fight against. They are also very widespread and are usually shocking to many people since they effect many people. Take the mined tweet in Figure 10; an entire school district was held hostage by ransomware, and it probably effected thousands of people which means it will be talked about much more often than a more low-key attack.

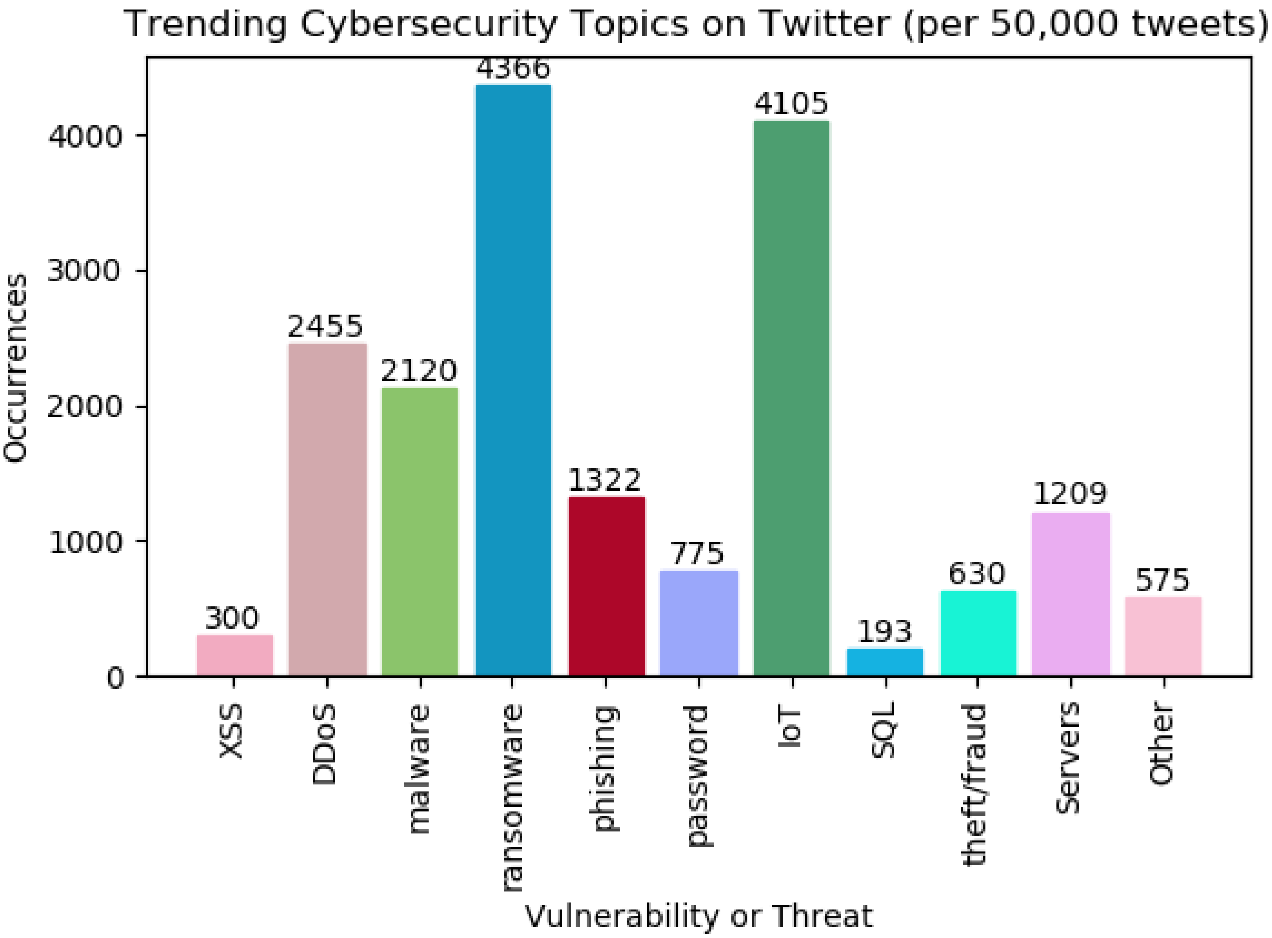


Figure 14: Numerical tallies supporting the percentages in Figure 13.

Even though this project did not center around web vulnerabilities, the algorithm developed in the project had a top 10 attacks that overlapped with the latest OWASP Top 10 Security Risks. These turned out to be Cross Site Scripting (XSS) and Injection (SQL-injection). Though more on OWASP's list were found, they were not as significant. The project focused more on cybersecurity vulnerabilities, trends, and topics which encompass but also surpass the realm of web vulnerabilities.

Of the 56 attack types searched, 26 were found in the mined tweets. Table 1 shows the popularity of those that were returned. Note that some of the members of the list are topics (such as attacks on servers denoted as 'server attack'):

Table 1: Attack Tweets November 5th – November 12th

Attack	Occurrences
ransomware	4366
IoT	4105
DDoS	2455
malware	2120
phishing	1322
server attack	1209
password	775
theft/fraud	630
XSS	300
SQL-Injection	193
third-party	125
state-sponsored	113
buffer overflow	71
sensitive data	67
social security	43
zero-day	39
drive-by	32
man-in-the-middle	28
XXE	19
misconfiguration	15
cookies	4
brute force	4
insider attacks	4
birthday	2
dictionary	1
Idle	1

As seen in the statistics, ransomware and IoT attacks are far and away the most popular. Ransomware is a type of malware that focuses on denying access to a computer system or data until a ransom is paid. They are a virus usually contracted through a phishing email (also a popular attack on Twitter) or by visiting an infected website and are becoming increasingly lucrative to hackers. Until a few years ago, malicious actors mainly targeted individuals but have since moved on to larger enterprise. By doing this, they can receive much more money and companies are less willing to hold out for fear of reprisals. This is commonly seen in the healthcare industry, such as the MedStar Health ransomware attack where it is a lose-lose situation for companies: if they decide to pay, they have not only set a dangerous precedent but will lose the confidence and faith of their customers. If they decide to hold out, the malicious actors might get tired of waiting and sell customer information on the dark web, also leading to a loss of faith and business. The biggest motivation for an attacker is that they hold all the cards. For example, if an attacker holds medical equipment hostage during emergency surgery, the hospital can either pay off the attacker or the patient's life is put an further risk. Hence, it makes sense why this attack is the most popular and trending on Twitter: it is both *extremely dangerous and relevant*. In a day and age where technology has never been so great, its reliance is also our greatest hinder. IoT devices, also known as Internet of Things (smart coffee makers, security systems, etc.) were the second most popular attack trending on Twitter. According to Forbes, IoT attacks rose 300% in 2019, partially because of the sheer number of devices and partially because their architectures have little to no security. Due to the lack of defense most of the attacks are automated with bots and/or scripts and malware and are an easy way to get valuable information about a person, household, or business.

The second week of mining yielded similar results, signaling that the results from November 5th to 12th were probably not a fluke, and these topics are in fact the most popular at the moment. The 10 most popular attacks stayed the most popular, and XSS and SQL-injection remained as the similarities with OWASP's web vulnerability list. There were not many major movers on the list, with most attacks gaining or losing a half percent in popularity. Since there were not any major cybersecurity events (such as a major DDoS attack or phishing scam) taking place over this time period, it makes sense that it stayed steady. For the purpose of this project it was desired that there were no events so that the numbers would be pure, with no outliers due to unforeseen circumstances. See Table 2 and Figures 15 and 16 for the most trending attacks and topics from November 12th – 20th:

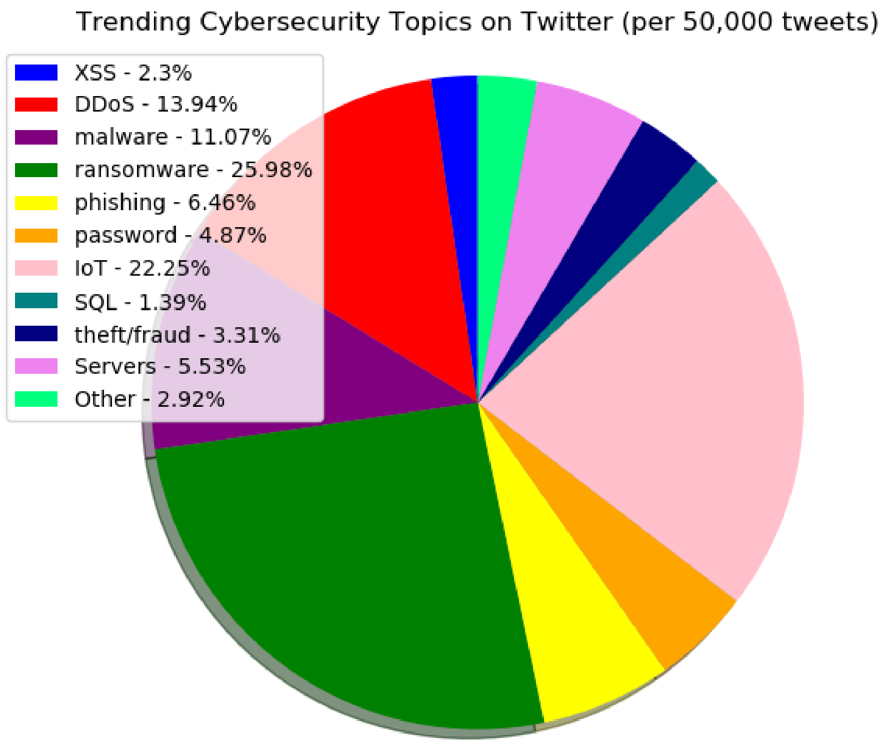


Figure 15: November 13th - 20th most trending cybersecurity attacks

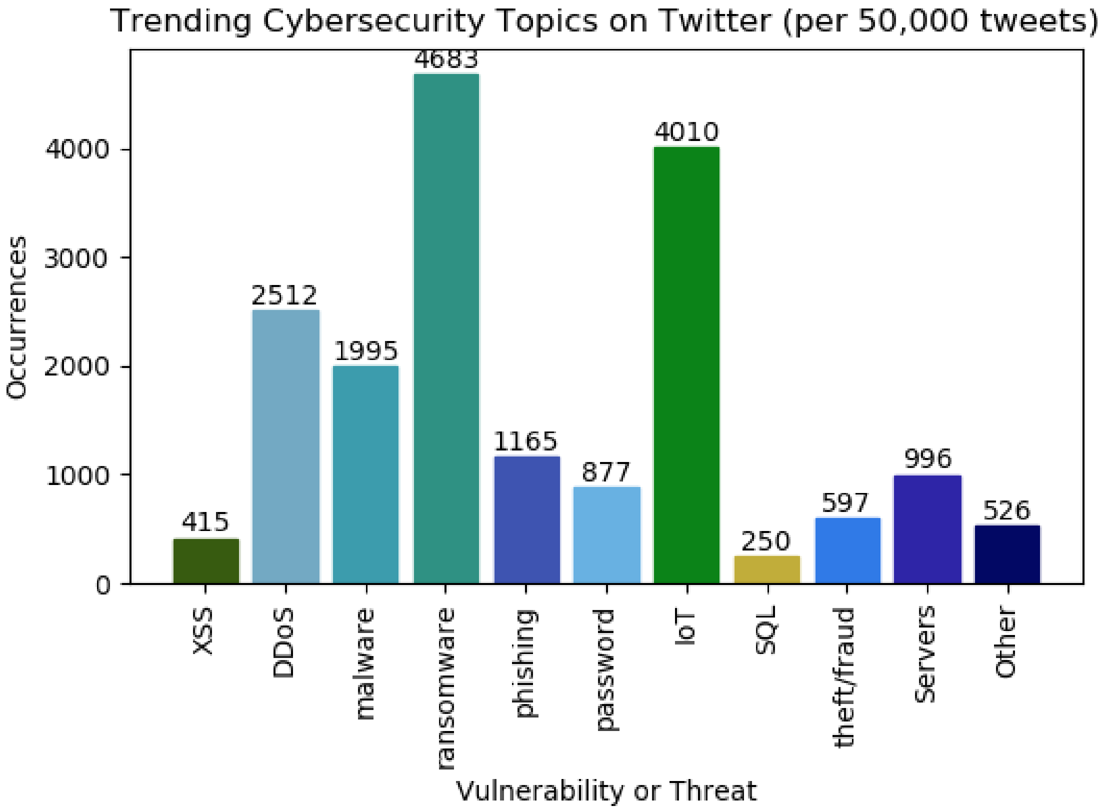


Figure 16: Numerical tallies supporting the percentages in Figure 15.

Table 2: Attack Tweets November 13th – November 20th

Attack	Occurrences
ransomware	4683
IoT	4010
DDoS	2512
malware	1995
phishing	1165
server attack	997
password	877
theft/fraud	597
XSS	415
SQL-Injection	250
third-party	119
state-sponsored	87
buffer overflow	72
sensitive data	64
zero-day	45
social security	30
man-in-the-middle	27
XXE	25
drive-by	19
misconfiguration	13
deserialization	7
cookies	4
brute force	4
Idle	2
birthday	2
insider attacks	1
dictionary	1
eavesdropping	1

In addition to investigating the most trending cybersecurity attacks, vulnerabilities, and topics, this project also investigated which industries and sectors are the highest trending when the subject is brought up. As businesses and enterprises are being increasingly targeted and hit with more sophisticated attacks, it is of interest to see which ones are hit the hardest. Similar to the previous subject investigated, this took place over November 5th to November 20th. The gathered statistics were then compared against each other to develop trends. See Figures 17 and 18 for the results. These statistics were also gathered to draw parallels between the most frequent attacks and the most frequent targets. For example, if ransomware is the most popular attack genre, what industries do they attack most frequently? What data do these industries possess that might be vulnerable and serve as motivation for malicious actors?

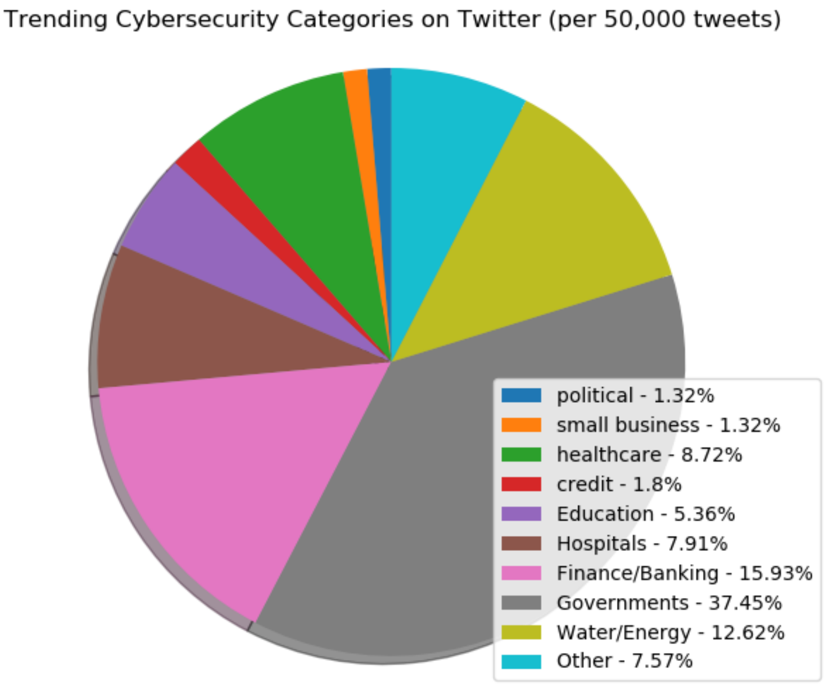


Figure 17: November 5th - 12th most trending cybersecurity attack targets

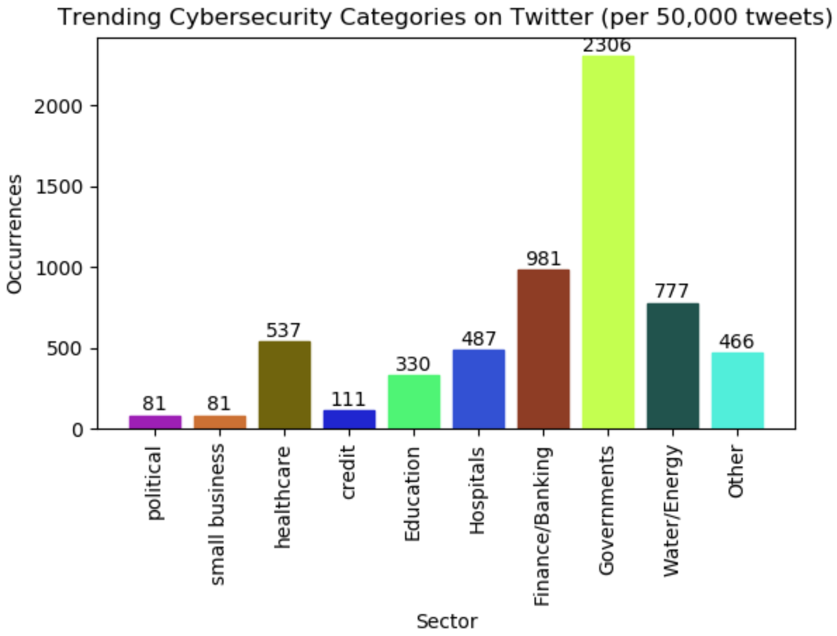


Figure 18: Numerical tallies supporting the percentages in Figure 17.

As can be seen with the numbers, governments were far and away the biggest cyber target. It should be noted that the "political" category are separate from the government since they are separate political institutions/groups, and "credit" is separated from the larger financial category due to being credit unions. Since nation states frequently target each other's infrastructure and data, it is unsurprising that they were the most trending targeted category. In addition, financial institutions, water and energy, healthcare, and hospitals were also frequent targets. All of these institutions are targets for IoT devices and ransomware. As for the government attacks, IoT devices offer an easy in for what should be private networks. Seemingly harmless devices such as a coffee maker or smart fridge usually have little to no security, which a malicious actor or nation-state can take advantage of. The other industries (especially healthcare, hospitals, and finance) literally rely on customer trust and an assumption that customer data will stay secure. Ransomware attacks can hold this data hostage, and more often than not significantly cost companies millions. This data is usually personal and private and can be damaging to individuals if it is sold and distributed. In sum, these attacks make their way down the ladder from large enterprise to individual customer. Below in Table 3 are the complete findings for the first week of attack targets:

Table 3: Attack Target Tweets November 5th – November 12th

Target	Occurrences
Governments	2306
Finance/Banking	981
Water/Energy	777
Healthcare	537
Hospitals	487
Education	330
Credit	111
Small business	81
Political	81
Automobiles	77
Online retail	74
Transportation	63
SSN	48
Telecommunications	35
Private sector	18
Public sector	18
Control systems	18
Refrigerator	6
Planes	1

The week of November 13th-20th changed drastically. Attacks in context of percentages only changed mildly: mined tweets on the subject of governments fell around 3%, while on the other hand water and energy gained about 3% and became the second most commonly found target. The big change came in the number of tweets mined. There were a few thousand more tweets mined on the subject of industries and enterprise when it came to the realm of cybersecurity. See Table 4 and Figures 19 and 20.

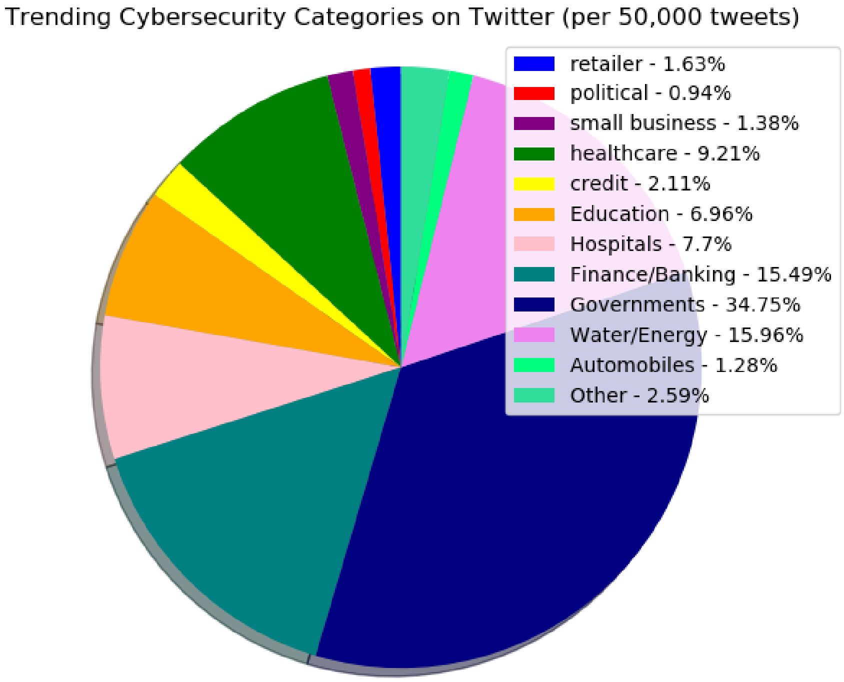


Figure 19: November 13th - 20th most trending cybersecurity attack targets

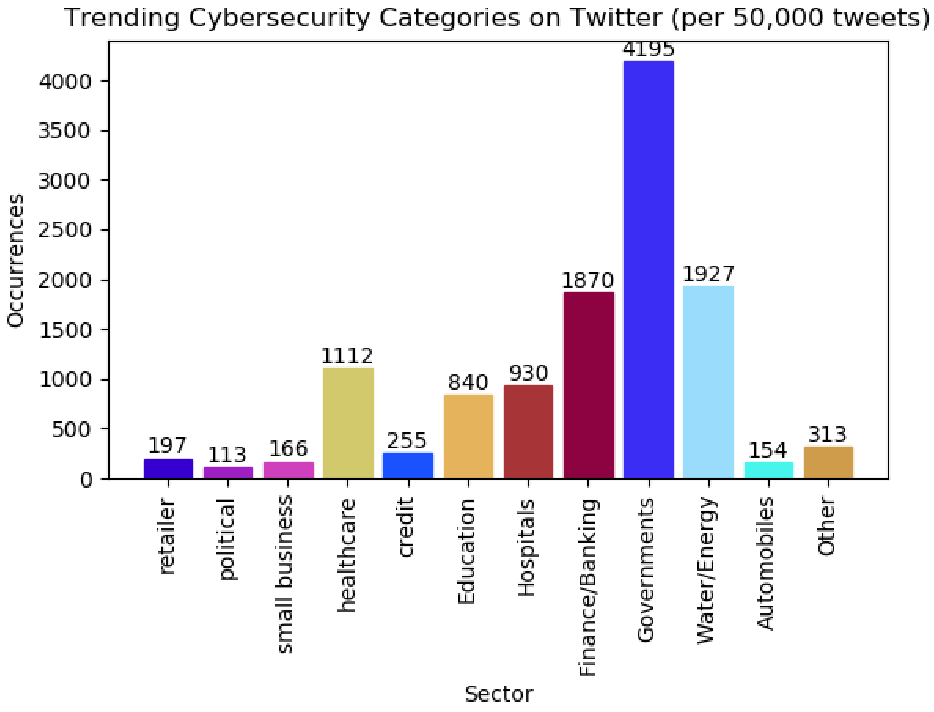


Figure 20: Numerical tallies supporting the percentages in Figure 19.

Table 4: Attack Target Tweets November 13th – November 20th

Target	Occurrences
Governments	4195
Water/Energy	1927
Finance/Banking	1870
Healthcare	1112
Hospitals	930
Education	840
Credit	255
Online retail	197
Small business	166
Automobiles	154
Political	113
Transportation	78
SSN	66
Telecommunications	35
Public sector	20
Private sector	18
Control systems	16
Planes	6
Refrigerator	4

The number of tweets rose by around 96%, from 6,157 to 12,072 tweets mined. Most of the gain was seen in the upper eight categories, with all of them nearly doubling in total occurrences. This rise correlates with slight rises in ransomware attacks, though does not heavily correlate with any of the other attacks. The steep rise could also be a result of CyberCon conference in Anaheim, which is a large event where industry leaders discuss solutions to protect public and private infrastructure. This encapsulates the government, water/energy, finance/banking, healthcare, and hospitals since they all rely upon digital infrastructure to function. The conference revolves around the US power grid. The influx and interest generated in the event would explain the high jump in numbers, though this cannot be known for sure.

Conclusion

This project provided a wealth of information and learning. I have used API's extensively before, however I never investigated how they actually worked behind the scenes. In preparation for this report, I studied the exchange of keys, tokens, and authentications and developed a new respect for the design of REST API's and their development and usage in the real world. Additionally, I learned very useful skills about gathering near real-time data and trends using an incredibly popular social media platform: Twitter. The wealth of knowledge they provide at the call of an API is astounding, as is the ease at which it can be achieved with a few simple lines of code.

Given more time to build up this project, I would have liked to implement machine learning algorithms to predict future trends with the data. For example, after gathering a few months' worth of labeled sample data a model would be trained with the purpose of predicting the trends for each month. Python offers a number of libraries to ease the burden of implementation, such as Scikit-learn, Natural Language Toolkit (NLTK), and keras. Since Twitter is a social media site and not everybody is an expert in the field, it would be smartest to use Sentiment Analysis due to the subjectivity of user's opinions.

Overall, this project was a great experience. Not only did it offer the opportunity to combine data analytics with relevant cybersecurity topics, it also showed me one way to keep on top of what is relevant in the field. Since Twitter is real time and is made up of millions of normal people, it is a positive to know what is currently being spoken about, especially in an ever-changing field for which I am trying to be employed in. This is perhaps the most important piece of knowledge obtained from this project, even more so than learning more about API's and behind the scenes exchanges.

Demo Video Link: https://www.youtube.com/watch?v=_pDgCXuQeJo&feature=youtu.be