Team members:

Homoud Al-Ghanim \ 2132131306

Mohammad Al-Tarrah \ 2141113773

Phenomenom:

Exploratory Data Analysis(EDA) on cyber security using social media for the most active users and countries

Used Packages:

package name	uses
Pandas	read\create Dataframe
Tweepy	Import tweets from twitter
Altair	Data Visualization and plots
NLTK	analyze text and do sentiment analysis
Matplotlib	Data Visualization
Numpy	help with mathematical functions
WordCloud	Create a word cloud image
Basemap	Create map using the dataframe

Interesting Variables

In [212]:
```
df[['account_Name','userProfile_created_time','tweet','tweet_sent_Time','tweet_language','source']].head(10)
```

Out[212]:

	account_Name	userProfile_created_time	tweet	tweet_sent_Time	tweet_language	source
0	rachealgudaiti1	2017-03-25 20:47:39	RT @iFutureTek: Key #GDPR & #NewYork #Cybe...	2017-12-17 17:36:43	en	Twitter for Android
1	eescobar8127	2012-06-24 17:46:49	RT @iFutureTek: #CyberSecurity Attack is a glo...	2017-12-17 17:36:43	en	Twitter for Android
2	blystoneraclng	2014-04-15 04:06:25	RT @iFutureTek: #CyberSecurity Attack is a glo...	2017-12-17 17:36:40	en	Twitter for Android
3	JeremyhPh0to	2012-06-24 17:11:50	RT @iFutureTek: #CyberSecurity Attack is a glo...	2017-12-17 17:36:39	en	Twitter for Android
4	QueenB__Speakn	2015-03-02 02:31:01	RT @iFutureTek: #CyberSecurity Attack is a glo...	2017-12-17 17:36:38	en	Twitter for Android
5	stefenie_8206	2014-03-26 13:35:31	RT @iFutureTek: #CyberSecurity Attack is a glo...	2017-12-17 17:36:35	en	Twitter for Android
6	EmilyBrianne11	2011-12-18 01:17:52	RT @iFutureTek: #CyberSecurity Attack is a glo...	2017-12-17 17:36:33	en	Twitter for Android
7	CallttaH	2011-05-02 14:23:04	RT @iFutureTek: #CyberSecurity Attack is a glo...	2017-12-17 17:36:32	en	Twitter for Android
8	bPositive_	2009-04-01 01:45:53	After #FBI Director #ComeyFiring, any attempt ...	2017-12-17 17:36:31	en	Twitter Web Client
9	Omar_S558	2011-03-26 12:50:25	RT @iFutureTek: #CyberSecurity Attack is a glo...	2017-12-17 17:36:30	en	Twitter for Android

The number of tweets has been collected = 202,325

Result:

1.

Variables identification

Column Name	Variable definition	Data Type	Missing data report	distribution of the data	level of analysis
account_Name	The sender profile account name	String	False	No Distribution	User
userProfile_created_time	The time of when the users created their profile	continues(datetime64)	False	No Distribution	User
location	The location the user put on their profile page(not all the time true)	String	True(44,397)	No Distribution	User

Column Name	Variable definition	Data Type	Missing data report	distribution of the data	level of analysis
following_count	The number of account's the user follows	Continues(int64)	False	minimum(0),maximum(1148424)	User
followers_count	The number of account's follow this user	Continues(int64)	False	minimum(0),maximum(19181792)	User
account_total_tweets	The Total tweets the user sent from the creation of the profile	Continues(int64)	False	minimum(1),maximum(8271116)	User
tweet_sent_Time	The tweet creation or retweeted time	Continues(datetime64)	False	No Distribution	Tweet
tweet	The content of the tweet that has been sent	String	False	No Distribution	Tweet
tweet_language	The language of the tweet that has been sent	String	False	No Distribution	Tweet
hashtags	The hashtags inside the tweets if available	String	False	No Distribution	Tweet
mentions	The account's that has been mentioned in the tweet	String	False	No Distribution	User
reply_to_user	The user this tweet is being replied to	String	True(197,538)	No Distribution	Tweet
geo	The place or coordinates that the tweet been sent from(mostly accurate)	String	True(202,072)	No Distribution	Tweet
retweet_count	The number of times this tweet has been retweeted	Continues(int64)	False	minimum(0),maximum(102621)	Tweet
favorite_count	The number of times this tweet has been liked	Continues(int64)	False	minimum(0),maximum(10990)	Tweet
source	The source or the platform of the device the tweet been sent from	String	False	No Distribution	Tweet
url	The url of the tweet	String	False	No Distribution	Tweet

2.

Transformed Variables

Variable Names	Variable description	Steps in transformation	level of analysis
sentiment_analysis	The expression of the tweet if it was negative, neutral or positive	Used NLP SentimentIntensityAnalyzer provided by vaderSentiment package to get the sentiment expression	Tweets
sentiment_score	The score of the expression if it was extreme or not	Used NLP SentimentIntensityAnalyzer provided by vaderSentiment package to get the sentiment score	Tweets
top5_active_users	The most 5 active users by number of tweets they sent	used value count to get the most common users who sent tweets	Users
top3_source	The most used platform to sent a tweet	used value counts to get the most used platform from all the dataframe	Tweets

3.

insights:

insights:

Extreme values

In [215]:
```
df[['account_Name','tweet','sentiment_analysis','sentiment_score']].sort_values(by='sentiment_score',ascending=False).hea

# The 1st positive sentiment isn't related to cyber security
```

Out[215]:

	account_Name	tweet	sentiment_analysis	sentiment_score
39585	LilJamiee_	@niacerise HAPPY BIRTHDAY DEAR. WULLNP. AGE ...	positive sentiment	0.9906
161338	CyFlare_SOC	We are so excited to be working with Rochester...	positive sentiment	0.9850

In [218]:
```
print(df.tweet[39585]+'\n')

print(df.tweet[161338])
# This tweet was the extreme positive sentiment in the dataframe
```

```
@niacerise HAPPY BIRTHDAY DEAR.  WULLNP.  AGE ON WITH GRACE FOR EXPLOITS. I PRAY THAT DOUBLE PORTIONS OF BLESSINGS AND
LOVE AND FAVOUR AND EVERYTHING GOOD YOU WANT BE YOURS IN JESUS NAME,  AMEN AND I ALSO PRAY THAT YOU ACCOMPLISH THE ESSE
NCE OF YOUR EXISTENCE PREDESTINED BY GOD.  HBD

We are so excited to be working with Rochester Institute of Technology working with the best and brightest new talent i
n the cyber security market.  We cannot wait for the amazing new ideas and innovation to come. CyFlare is a place for t
he best and brightest to come and disrupt.
```

In [223]:
```
df[['account_Name','tweet','sentiment_analysis','sentiment_score']].sort_values(by='sentiment_score').head()[0:2]
# The 1st negative sentiment isn't related to cyber security
```

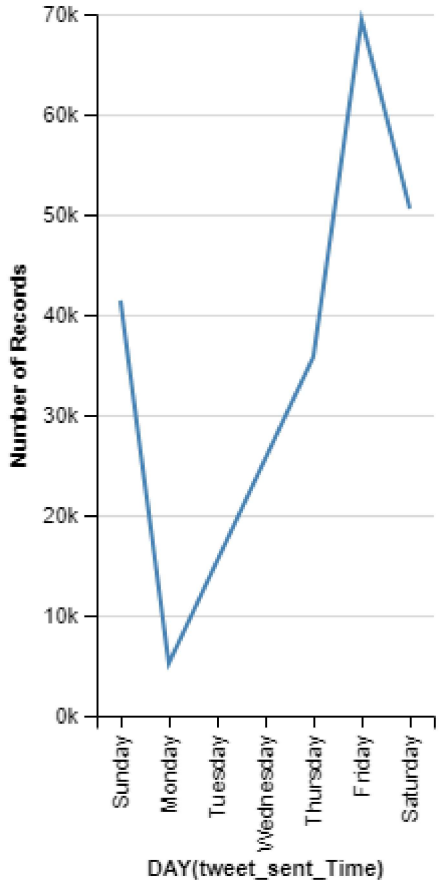
Out[223]:

	account_Name	tweet	sentiment_analysis	sentiment_score
31728	mlb5555ed	I think they're both DEAD don't you? Or very i...	negative sentiment	-0.9738
84283	susandavis178	@VP @POTUS Special thanks to our troups/law en...	negative sentiment	-0.9726

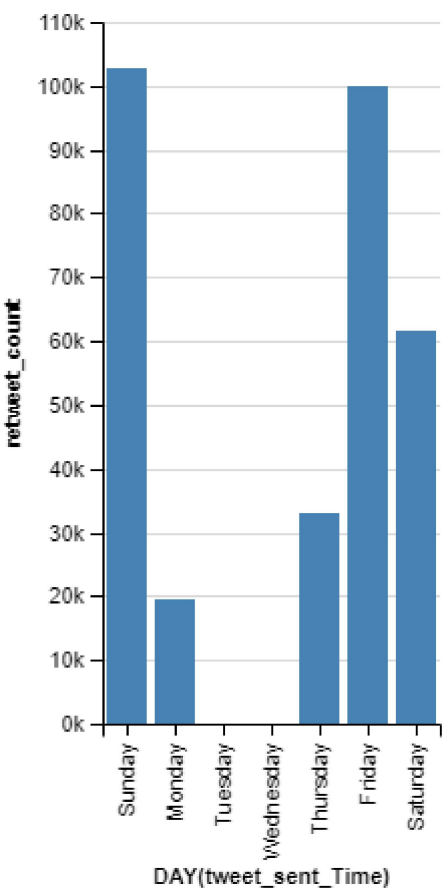
In [222]:
```
print(df.tweet[31728]+'\n')

print(df.tweet[84283])
# This tweet is the extreme negative sentiment in the dataframe
```

```
I think they're both DEAD don't you? Or very ill. Why silence #SeaWorld? You usually squeeze every last $$ out of 'cut
e' pics &amp;amp; PR from the captive baby. You're a disgrace. A cruel breeding program which kills &amp;amp; exploits #belugas
.You should stop now &amp;amp; close your doors 4ever https://t.co/cwYOF3jTRc (https://t.co/cwYOF3jTRc)

@VP @POTUS Special thanks to our troups/law enforcement/Intel Agencies. FBI? Cyber crime, terrorism, civil rights, orga
nized crime, violent crime, pharmaceutical theft, smuggling, espionage, bribery, extortion, bid-rigging, gangs, kleptoc
racy, child predators, mass shootings, serial murder
```

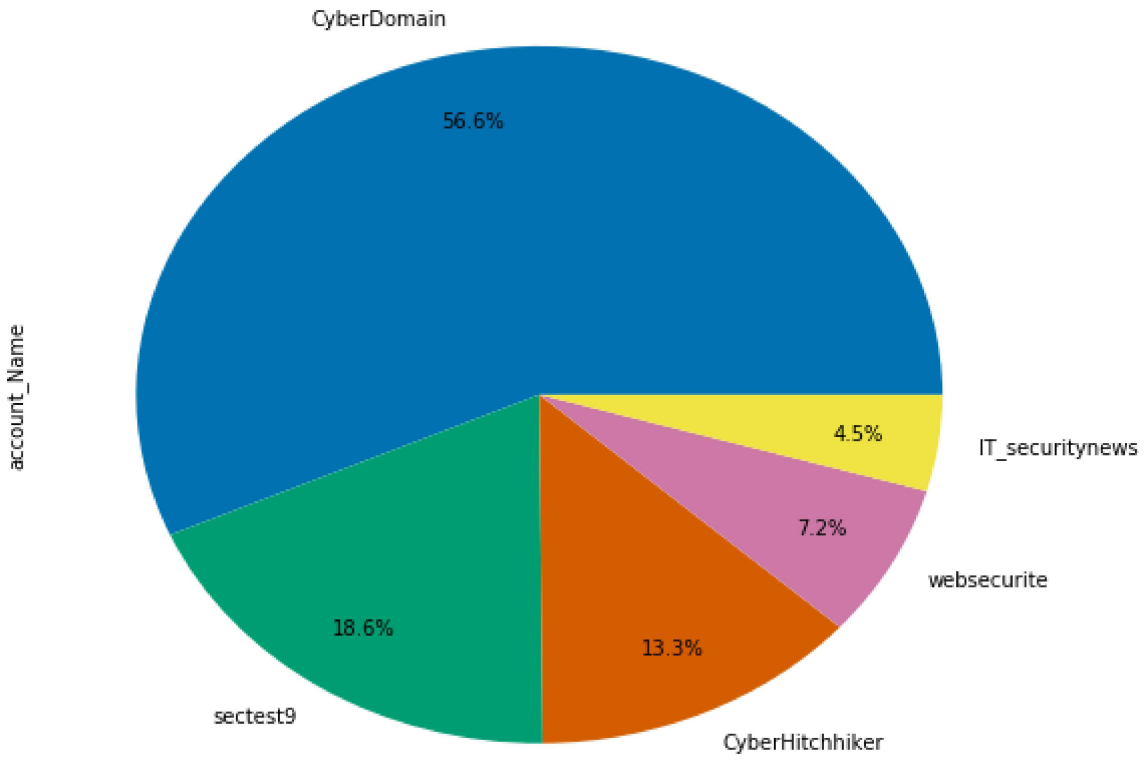
There's a big dicline in monday because it's the first work day in the week And the most active time is from friday to sunday because it's a weekend.



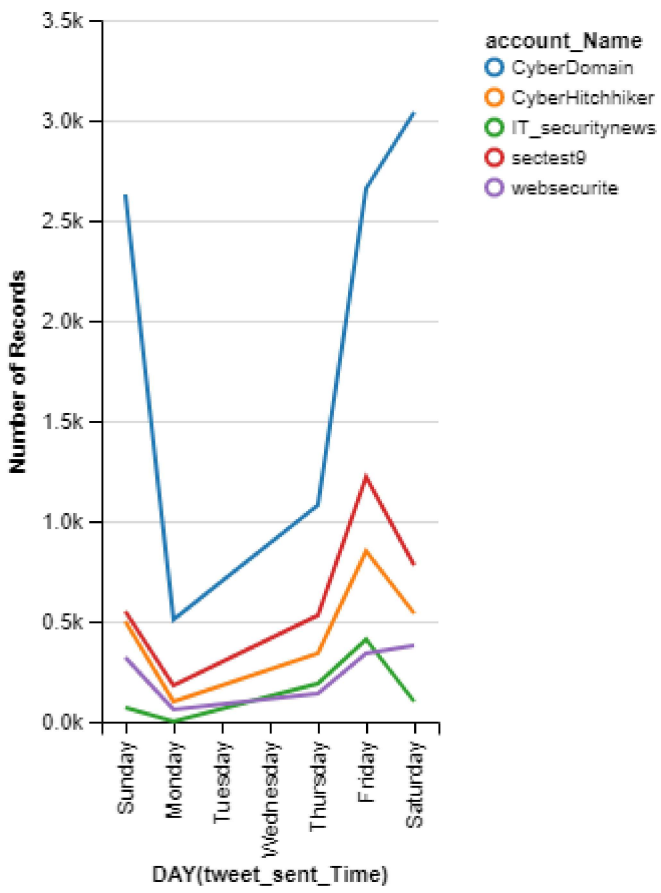
The most retweeted tweets was in sunday and friday. What was the most tweet or topic been retweeted in friday?



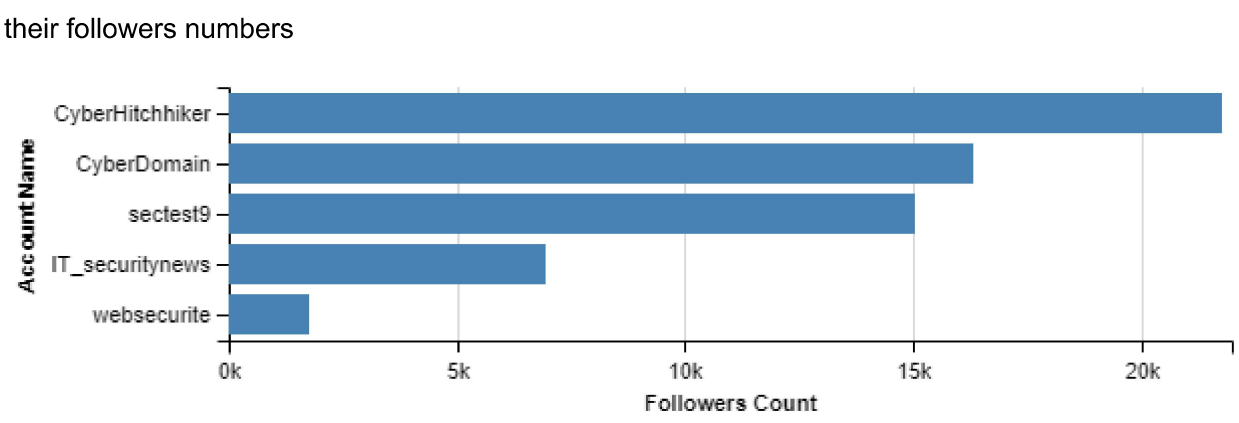
The most active accounts based on tweets number



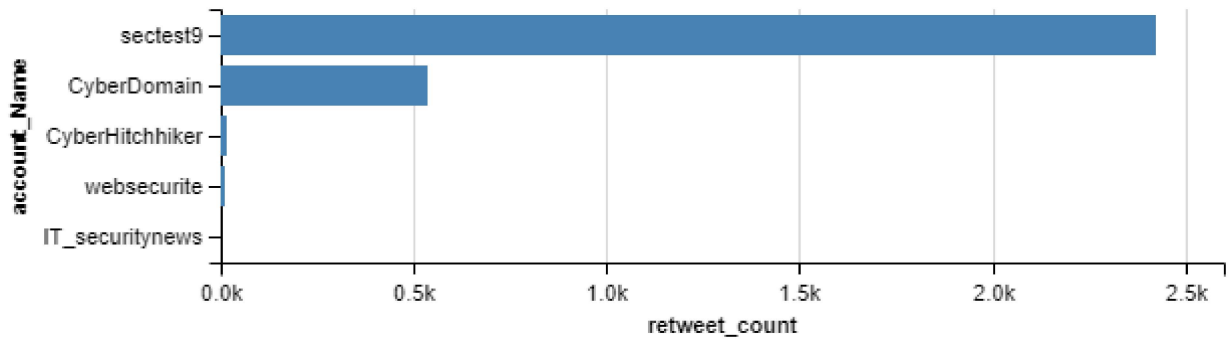
The most active user in Friday and Saturday is CyberDomain



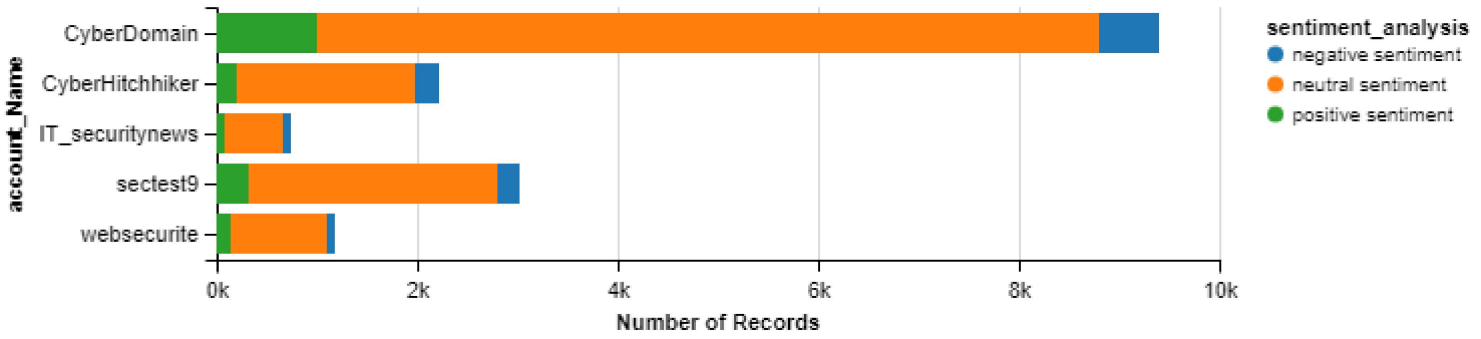
The account's based on their followers numbers



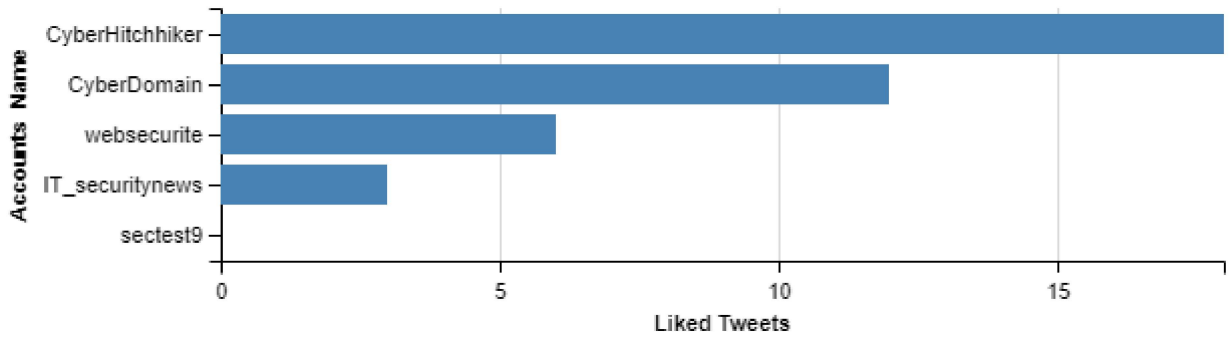
The account's basen on their retweeted tweets



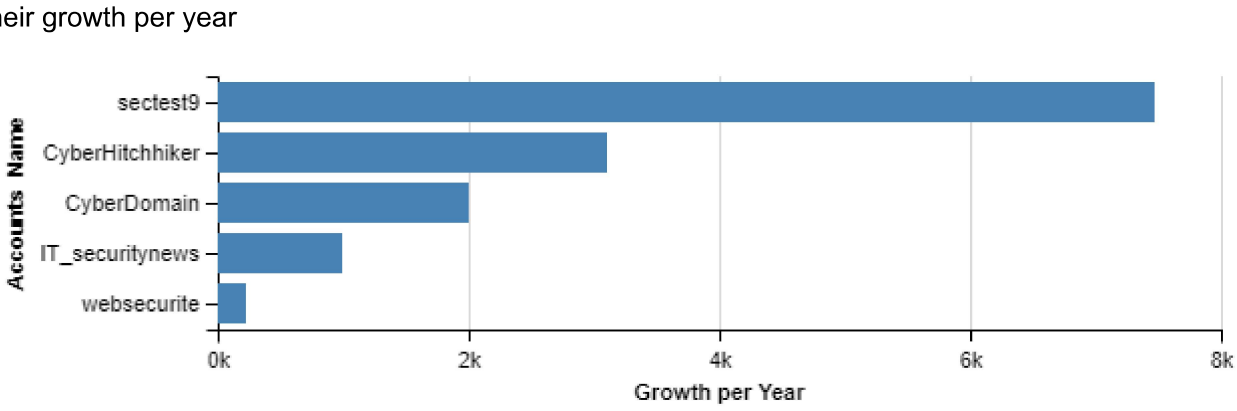
Here we can see each account and their sentiment analysis from their tweets



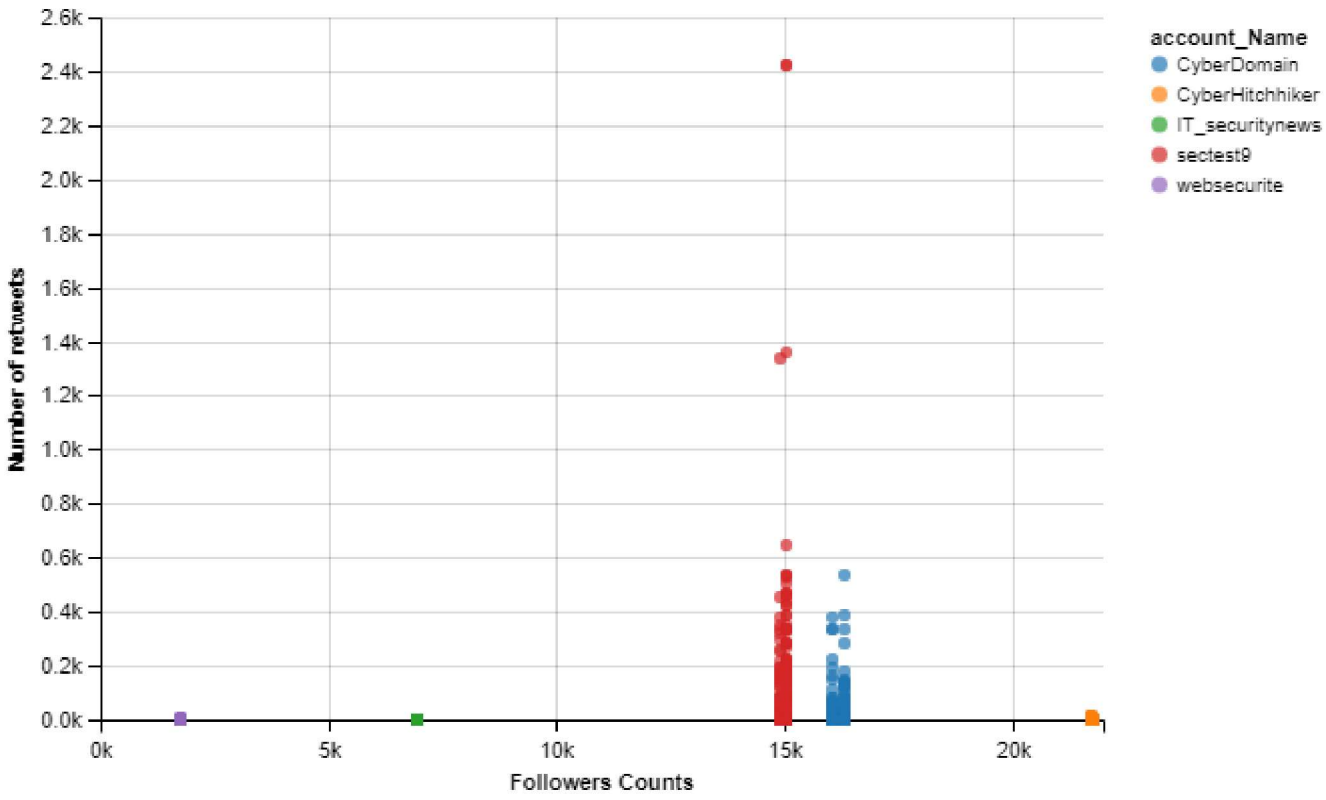
Most liked tweets from the top 5 users



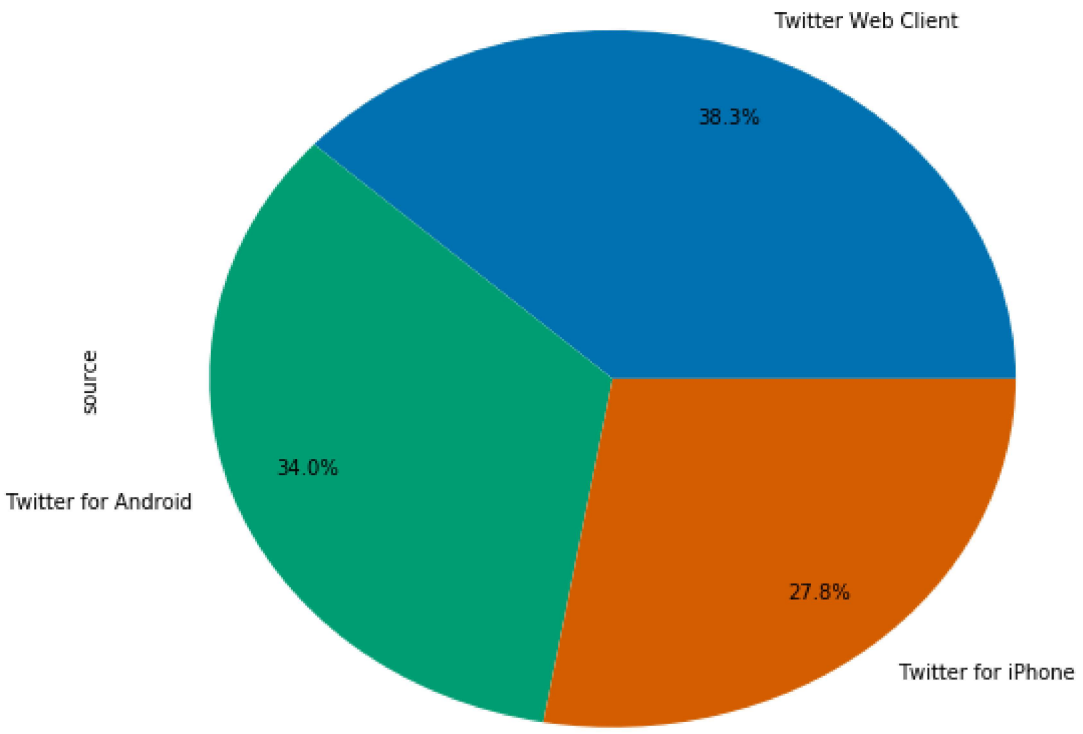
The account based on their growth per year



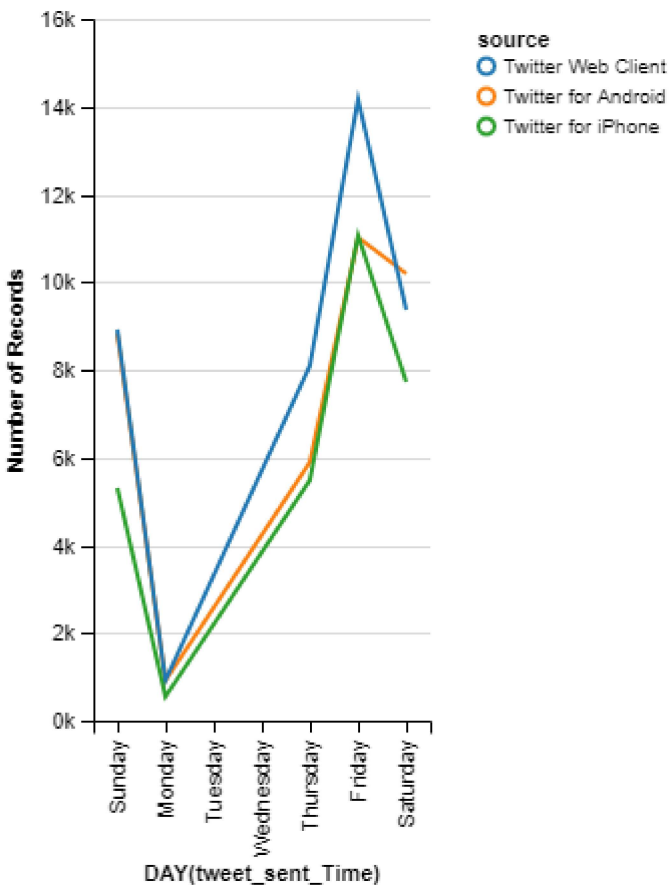
It's not important to have more followers to get retweeted more



top 3 used platforms

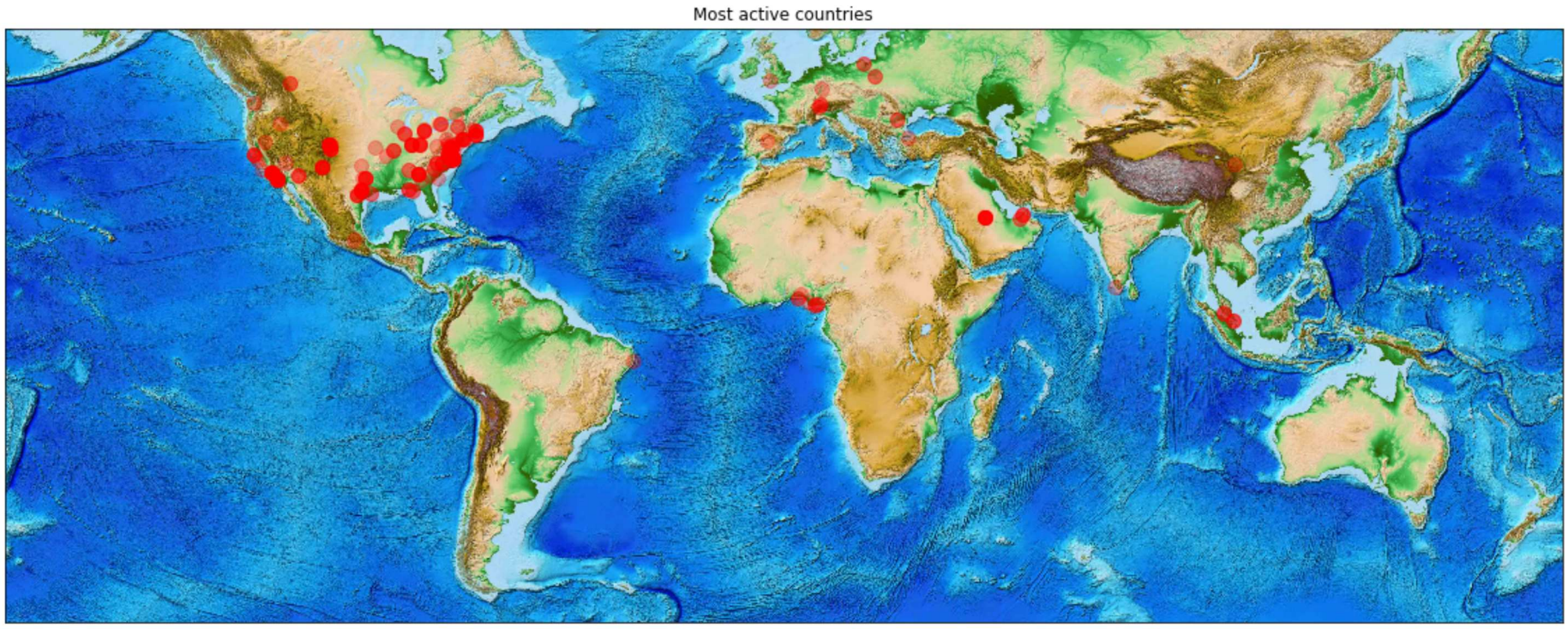


From all the spike in friday we see that the web client is the most used one



This is a word cload contain the most common words used in tweets And their size based on their frequency

The darker the red dot the higher number of users there interested in cyber security



Questions???

Analysis:

In [1]:
```
import pandas as pd
import tweepy
from nltk.tokenize import regexp_tokenize
import altair as alt
%matplotlib inline
import nltk
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
#Addinh the keys of the api
consumer_key='X27CcqxmukbhG9zZxBcJYQf5L'
consumer_secret='G4sc1awVnqZg3BkzJR7P3cd6qMcTSxV7d87NKn9euD6Qz8kdIF'
access_token= '933745888777957376-hJQy24QHKdnozFIuV4xCgYMxjSeerPM'
access_token_secret= 'iT1MfT2Ur5ekTzjVha0JQV4MMKDwkDOezOc3qWBFbFWy9'

auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
# create the authenticated api object
api = tweepy.API(auth)
```

In [3]:
```
#The query i want to search
search_terms='"cybersecurity" OR "cyber security" OR "امن المعلومات" OR "السيبراني" OR "infosec" OR "cybercrime" OR "cyber crime"
```

In [4]:
```
#By using square bracket we did a list comprehension and put our loop inside
#each page will contain 200 tweet and we want 1,500 page if possible
#And by using wait_on_rate_limit and wait_on_rate_limit_notify the loop will stop by
#itself when we reach the limit and notify us.

tweets=[twet for twet in tweepy.Cursor(api.search,
                                q = search_terms,
                                tweet_mode='extended',
                                wait_on_rate_limit=True,
                                wait_on_rate_limit_notify=True,
                                show_user=True,
                                count=200).pages(1500)]
```

```
Rate limit reached. Sleeping for: 595
Rate limit reached. Sleeping for: 538
Rate limit reached. Sleeping for: 575
Rate limit reached. Sleeping for: 562
Rate limit reached. Sleeping for: 555
Rate limit reached. Sleeping for: 557
Rate limit reached. Sleeping for: 562
Rate limit reached. Sleeping for: 599
```

In [5]:
```
#Here we clean the list's and add None in the empty places
def val(x):
    if x=='' or x== None:
        return "None"
    else:
        return x
```

In [6]:
```
df = pd.DataFrame()

#Account name
acc=[]
[[acc.append(twet.user.screen_name) for twet in tweet] for tweet in tweets]
df['account_Name'] = acc

#User location
loc=[]
[[loc.append(val(twet.user.location)) for twet in tweet] for tweet in tweets]
df['location']=loc

#profile created time
pr_created_time=[]
[[pr_created_time.append(str(twet.user.created_at)) for twet in tweet] for tweet in tweets]
df['userProfile_created_time'] = pr_created_time


#Account following count
flng_count=[]
[[flng_count.append(twet.user.friends_count) for twet in tweet] for tweet in tweets]
df['following_count'] = flng_count


#Account followers count
flwrs_count=[]
[[flwrs_count.append(twet.user.followers_count) for twet in tweet] for tweet in tweets]
df['followers_count'] = flwrs_count


#Account total tweets
tltw=[]
[[tltw.append(twet.user.statuses_count) for twet in tweet] for tweet in tweets]
df['account_total_tweets'] = tltw


#The tweet has been sent or replied
twt=[]
[[twt.append(twet.full_text) for twet in tweet] for tweet in tweets]
df['tweet'] = twt


#tweet language
lng=[]
[[lng.append(twet.lang) for twet in tweet] for tweet in tweets]
df['tweet_language'] = lng


#tweet sent or replied time
crtd=[]
[[crtd.append(twet.created_at) for twet in tweet] for tweet in tweets]
df['tweet_sent_Time'] = crtd


#hashtags found in tweet
hashtags=[]
[[hashtags.append(regexp_tokenize(hashtag.full_text, r"#\w+")) for hashtag in tweet] for tweet in tweets]
df['hashtags']=hashtags
df.hashtags=df.hashtags.astype(str).str.replace('\[|\]|\'', '')

#Mentions in the tweet
mentions=[]
[[mentions.append(regexp_tokenize(mention.full_text, r"@\w+")) for mention in tweet] for tweet in tweets]
df['mentions']=mentions
df.mentions=df.mentions.astype(str).str.replace('\[|\]|\'', '')

#The tweet replied to that user
rply=[]
[[rply.append(val(twet.in_reply_to_screen_name)) for twet in tweet] for tweet in tweets]
df['reply_to_user']=rply


#The location of the tweet
geo=[]
[[geo.append(val(twet.geo)) for twet in tweet] for tweet in tweets]
df['geo']=geo

#Count of retweets
retwt_con=[]
[[retwt_con.append(twet.retweet_count) for twet in tweet] for tweet in tweets]
df['retweet_count']=retwt_con

#Count of twet likes
fav_con=[]
[[fav_con.append(val(twet.favorite_count)) for twet in tweet] for tweet in tweets]
df['favorite_count']=fav_con

#The source of the tweet
```

```
src=[]
[[src.append(twet.source) for twet in tweet] for tweet in tweets]
df['source']=src

#URL of the tweet
url=[]
[[url.append('https://twitter.com/'+twet.user.screen_name+'/status/'+twet.id_str) for twet in tweet] for tweet in tweets]
df['url'] = url
```

In [7]:
```
#Here we turned the date column to string type so it easier to plot in altair later
df.tweet_sent_Time=df.tweet_sent_Time.astype(str)
```

In [9]:
```
df.tail()
```

Out[9]:

	account_Name	location	userProfile_created_time	following_count	followers_count	account_total_tweets	tweet	tweet_language
126864	fastmetrics1	San Francisco, CA	2008-09-09 22:10:25	23966	22349	19118	RT @tfkohler: How does #Blockchain work - with...	en
126865	ehackingss	Internet	2014-09-30 17:39:56	1247	6160	75531	Manager - #Security and Safety https://t.co/Yb...	en
126866	geekable	Elsewhere	2007-02-06 00:09:56	420	848	31500	RT @aionescu: A... LUA VM... using JWT... insi...	en
126867	L_O_A_B	Lagos, Nigeria	2016-01-13 10:09:38	686	790	5820	RT @IMITAKCO: [#DigitalMarketing]\nUnderstan...	en
126868	iSandeepRaj	Worldwide	2012-05-01 08:53:53	2827	600	10405	RT @MumbaiPolice: Sharing your password is sim...	en

In [10]:
```
#The dataframe length or the number of tweets we have equals to 85,372
len(df)
```

Out[10]: 126869

In [11]:
```
#Here we used Json instead of csv because the was a problem using it the dataframe got mixed up when import later.
df.to_json('25_Data.json')
```

In [2]:
```
#Importing the json file and sort the index column because its unsorted when import json
df1=pd.read_json('17_Data.json')
df1=df1.sort_index(axis=0)

df2=pd.read_json('25_Data.json')
df2=df2.sort_index(axis=0)

df=pd.concat([df1, df2],axis=0)
df=df.reset_index(drop=True)
```

In [3]:
```
len(df)
```

Out[3]: 202325

In [4]:
```
df.head()
```

Out[4]:

	account_Name	account_total_tweets	favorite_count	followers_count	following_count	geo	hashtags	location	mentions	re
0	rachealgudaiti1	42	0	0	30	None	#GDPR, #NewYork, #CyberSecurity, #DataSecurity...	None	@iFutureTek	
1	eescobar8127	1984	0	90	5003	None	#CyberSecurity, #QuantumPhysics, #Quantum_Mech...	Gaithersburg,MD	@iFutureTek	
2	blystoneracIng	2301	0	100	4436	None	#CyberSecurity, #QuantumPhysics, #Quantum_Mech...	Commodore PA	@iFutureTek	
3	JeremyhPh0to	2192	0	99	5003	None	#CyberSecurity, #QuantumPhysics, #Quantum_Mech...	CDA, ID	@iFutureTek	
4	QueenB__Speakn	1769	0	33	355	None	#CyberSecurity, #QuantumPhysics, #Quantum_Mech...	Work & Home	@iFutureTek	

In [5]:
```
df.dtypes
```

Out[5]:
```
account_Name                        object
account_total_tweets                 int64
favorite_count                       int64
followers_count                      int64
following_count                      int64
geo                                 object
hashtags                            object
location                            object
mentions                            object
reply_to_user                       object
retweet_count                        int64
source                              object
tweet                               object
tweet_language                      object
tweet_sent_Time             datetime64[ns]
url                                 object
userProfile_created_time    datetime64[ns]
dtype: object
```

In [213]:
```
df.location[df.location=='None']=None
df.reply_to_user[df.reply_to_user=='None']=None
df.geo[df.geo=='None']=None


#in those column the null value assignewd to string 'None'
# if we want to sum all the null values it's better to change
# them back to null
```

In [214]:
```
print(df.isnull().sum())
```

```
account_Name                  0
userProfile_created_time      0
location                  44397
following_count               0
followers_count               0
account_total_tweets          0
tweet_sent_Time               0
tweet                         0
sentiment_analysis        23499
sentiment_score           23499
tweet_language                0
hashtags                      0
mentions                      0
reply_to_user            197538
geo                      202072
retweet_count                 0
favorite_count                0
source                        0
url                           0
latitude                 202072
longitude                202072
dtype: int64
```

In [7]:
```
df.describe()
```

Out[7]:

	account_total_tweets	favorite_count	followers_count	following_count	retweet_count
count	2.023250e+05	202325.000000	2.023250e+05	2.023250e+05	202325.000000
mean	9.580636e+04	0.781248	9.178760e+03	2.839356e+03	237.548691
std	2.118660e+05	33.431603	1.263182e+05	1.059291e+04	1113.455481
min	1.000000e+00	0.000000	0.000000e+00	0.000000e+00	0.000000
25%	2.222000e+03	0.000000	1.960000e+02	2.050000e+02	0.000000
50%	1.048700e+04	0.000000	9.130000e+02	7.910000e+02	3.000000
75%	5.552400e+04	0.000000	4.139000e+03	2.524000e+03	65.000000
max	8.271116e+06	10990.000000	1.918179e+07	1.148424e+06	102621.000000

In [4]:
```
analyzer = SentimentIntensityAnalyzer()
def sent(x):
    sa = analyzer.polarity_scores(x)
    emotion=float(sa['compound'])
    if emotion >= 0.5:
        return 'positive sentiment'

    elif emotion > -0.5 and emotion < 0.5:
        return 'neutral sentiment'

    else:
        return 'negative sentiment'

def sent_per(x):
    sa = analyzer.polarity_scores(x)
    emotion=float(sa['compound'])
    return emotion
```

In [5]:
```
df['sentiment_analysis']=df.tweet[df.tweet_language=='en'].apply(sent)
df['sentiment_score']=df.tweet[df.tweet_language=='en'].apply(sent_per)
```

In [6]:
```
# The columns order i want in the dataframe
columns=['account_Name','userProfile_created_time','location','following_count','followers_count','account_total_tweets',
```

In [7]:
```
#Re-sort the columns order
df=df[columns]

#The first 5 rows of the dataframe
df.head()
```

Out[7]:

	account_Name	userProfile_created_time	location	following_count	followers_count	account_total_tweets	tweet_sent_Time	twee
0	rachealgudaiti1	2017-03-25 20:47:39	None	30	0	42	2017-12-17 17:36:43	R @iFutureTek Key #GDPI &amp #NewYor #Cybe.
1	eescobar8127	2012-06-24 17:46:49	Gaithersburg,MD	5003	90	1984	2017-12-17 17:36:43	R @iFutureTek #CyberSecurit Attack is glo.
2	blystoneraclng	2014-04-15 04:06:25	Commodore PA	4436	100	2301	2017-12-17 17:36:40	R @iFutureTek #CyberSecurit Attack is glo.
3	JeremyhPh0to	2012-06-24 17:11:50	CDA, ID	5003	99	2192	2017-12-17 17:36:39	R @iFutureTek #CyberSecurit Attack is glo.
4	QueenB__Speakn	2015-03-02 02:31:01	Work & Home	355	33	1769	2017-12-17 17:36:38	R @iFutureTek #CyberSecurit Attack is glo.

Extreme values

In [215]:
```
df[['account_Name','tweet','sentiment_analysis','sentiment_score']].sort_values(by='sentiment_score',ascending=False).hea

# The 1st positive sentiment isn't related to cyber security
```

Out[215]:

	account_Name	tweet	sentiment_analysis	sentiment_score
39585	LilJamiee_	@niacerise HAPPY BIRTHDAY DEAR. WULLNP. AGE ...	positive sentiment	0.9906
161338	CyFlare_SOC	We are so excited to be working with Rochester...	positive sentiment	0.9850

In [218]:
```
print(df.tweet[39585]+'\n')

print(df.tweet[161338])
# This tweet was the extreme positive sentiment in the dataframe
```

```
@niacerise HAPPY BIRTHDAY DEAR.  WULLNP.  AGE ON WITH GRACE FOR EXPLOITS. I PRAY THAT DOUBLE PORTIONS OF BLESSINGS AND
LOVE AND FAVOUR AND EVERYTHING GOOD YOU WANT BE YOURS IN JESUS NAME,  AMEN AND I ALSO PRAY THAT YOU ACCOMPLISH THE ESSE
NCE OF YOUR EXISTENCE PREDESTINED BY GOD.  HBD

We are so excited to be working with Rochester Institute of Technology working with the best and brightest new talent i
n the cyber security market.  We cannot wait for the amazing new ideas and innovation to come. CyFlare is a place for t
he best and brightest to come and disrupt.
```

In [223]:
```
df[['account_Name','tweet','sentiment_analysis','sentiment_score']].sort_values(by='sentiment_score').head()[0:2]
# The 1st negative sentiment isn't related to cyber security
```

Out[223]:

	account_Name	tweet	sentiment_analysis	sentiment_score
31728	mlb5555ed	I think they're both DEAD don't you? Or very i...	negative sentiment	-0.9738
84283	susandavis178	@VP @POTUS Special thanks to our troups/law en...	negative sentiment	-0.9726

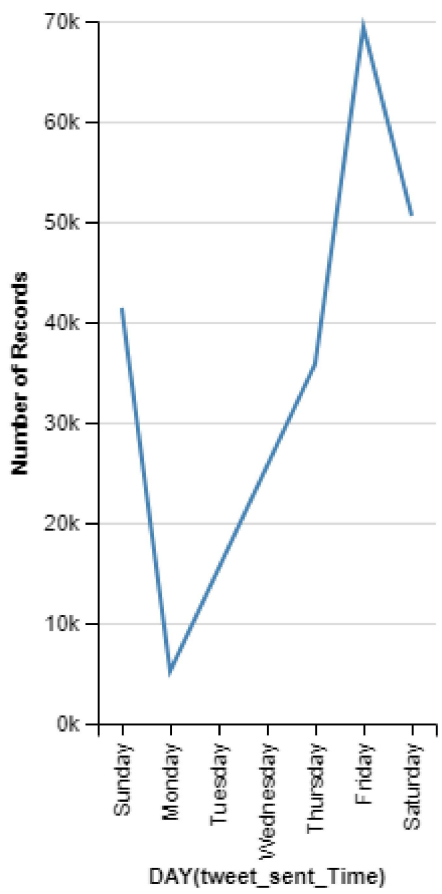
In [222]:
```
print(df.tweet[31728]+'\n')

print(df.tweet[84283])
# This tweet is the extreme negative sentiment in the dataframe
```

```
I think they're both DEAD don't you? Or very ill. Why silence #SeaWorld? You usually squeeze every last $$ out of 'cut
e' pics &amp;amp; PR from the captive baby. You're a disgrace. A cruel breeding program which kills &amp;amp; exploits #belugas
.You should stop now &amp;amp; close your doors 4ever https://t.co/cwYOF3jTRc (https://t.co/cwYOF3jTRc)

@VP @POTUS Special thanks to our troups/law enforcement/Intel Agencies. FBI? Cyber crime, terrorism, civil rights, orga
nized crime, violent crime, pharmaceutical theft, smuggling, espionage, bribery, extortion, bid-rigging, gangs, kleptoc
racy, child predators, mass shootings, serial murder
```

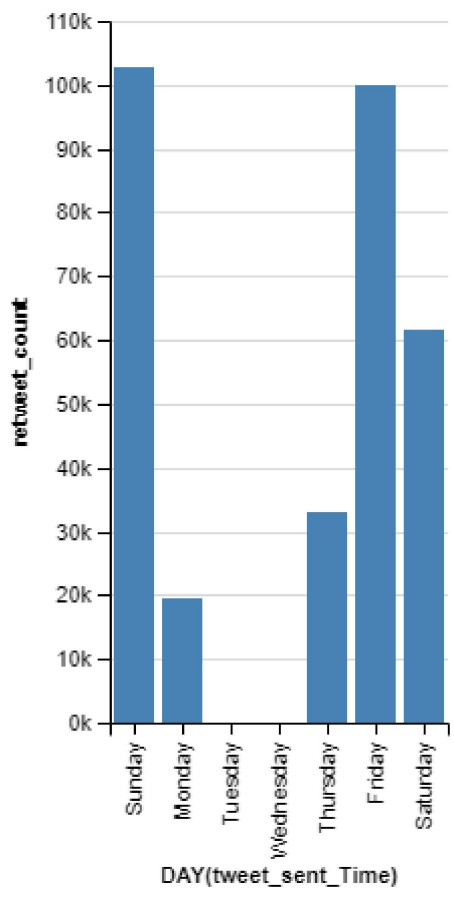
In [8]:
```
def to_altair(dataframe):

    '''This function take a dataframe and export it to json oriented as a records
        and the date is formated to iso, then return the name of the json file'''

    df_cp=dataframe.copy()
    for column in df_cp:
        if df_cp.loc[:,column].dtype == 'datetime64[ns]':
            df_cp.loc[:,column] = df_cp.loc[:,column].astype(str)
        else:
            pass
    df_cp.to_json('chart.json', orient='records',date_format='iso')
    return('chart.json')
pd.DataFrame.to_altair = to_altair
```

In [9]:
```
#Assign the maximum rows altair have to use (usually it's 5000 rows by default)
full_rows = alt.Chart(df.to_altair())
full_rows.max_rows = len(df)
```

In [42]:
```
alt.Chart(df.to_altair()).mark_line().encode(
    x=alt.X('tweet_sent_Time:T',timeUnit='day'),
    y='count(*):Q',
)
#there's a big dicline in monday because it's the first work day in the week
#And the most active time is from friday to sunday because it's a weekend.
```



In [21]:
```
full_rows.mark_bar().encode(
    x=alt.X('tweet_sent_Time:T',timeUnit='day'),
    y='retweet_count:Q',
)

#The most retweeted tweets was in sunday and friday.
#What was the most tweet or topic been retweeted in friday?
```



In [31]:
```
df.sort_values(by='retweet_count',ascending=False).head()[0:1]
#The most retweeted tweet was in sunday, 24th of DEC.
```

Out[31]:

	account_Name	userProfile_created_time	location	following_count	followers_count	account_total_tweets	tweet_sent_Time	tweet	senti
89049	InfoSec_Ash	2015-02-09 15:32:28	None	896	343	3034	2017-12-24 15:45:55	RT @elonmusk: Nuclear alien UFO from North Kor...	ne

In [19]:
```
df.tweet[89049]
# Talks about the UFO that has been seen in California
#SpaceX
```

Out[19]:
```
'RT @elonmusk: Nuclear alien UFO from North Korea https://t.co/GUIHpKkkp5' (https://t.co/GUIHpKkkp5')
```

In [29]:
```
df.sort_values(by='retweet_count',ascending=False).head()[1:2]
#The second most retweeted tweet was in Friday, 15th of DEC.
```

Out[29]:

	account_Name	userProfile_created_time	location	following_count	followers_count	account_total_tweets	tweet_sent_Time	tweet
67397	Infosec_Tourist	2010-04-20 14:52:29	Ottawa	655	1734	45185	2017-12-15 11:42:34	RT @MackenzieAstin: Hey, @AjitPaiFCC, today my...

In [32]:
```
df.tweet[67397]
# Talks about the FCC and the net neutrality
```

Out[32]:
```
"RT @MackenzieAstin: Hey, @AjitPaiFCC, today my mom would have turned 71. But she didn't. Because she died in March of 2016. Can you please…"
```

In [83]:
```
df.sort_values(by='favorite_count',ascending=False).head()[1:2]
# since the most liked tweet has nothing to do with security
# That's why i showed the second most liked tweet
```

Out[83]:

	account_Name	userProfile_created_time	location	following_count	followers_count	account_total_tweets	tweet_sent_Time	tweet	senti
124246	SirPareshRawal	2011-08-16 01:05:43	Ahmedabad & Mumbai	160	1525399	3499	2017-12-23 07:12:47	Beware This is a fake news n I have never sai...	neg

In [84]:
```
df.tweet[124246]
# The user aware the others that he didn't
# say this talk about cyber security and it's a fake news
```

Out[84]:
```
'Beware This is a fake news n I have never  said or believed in such views. lodged complaint at cyber crime dept . https://t.co/fddOBhdqLl' (https://t.co/fddOBhdqLl')
```

In [85]:
```
df.url[124246]
```

Out[85]:
```
'https://twitter.com/SirPareshRawal/status/944465796226605056'
```

In [10]:
```
df.account_Name.value_counts().head()
```

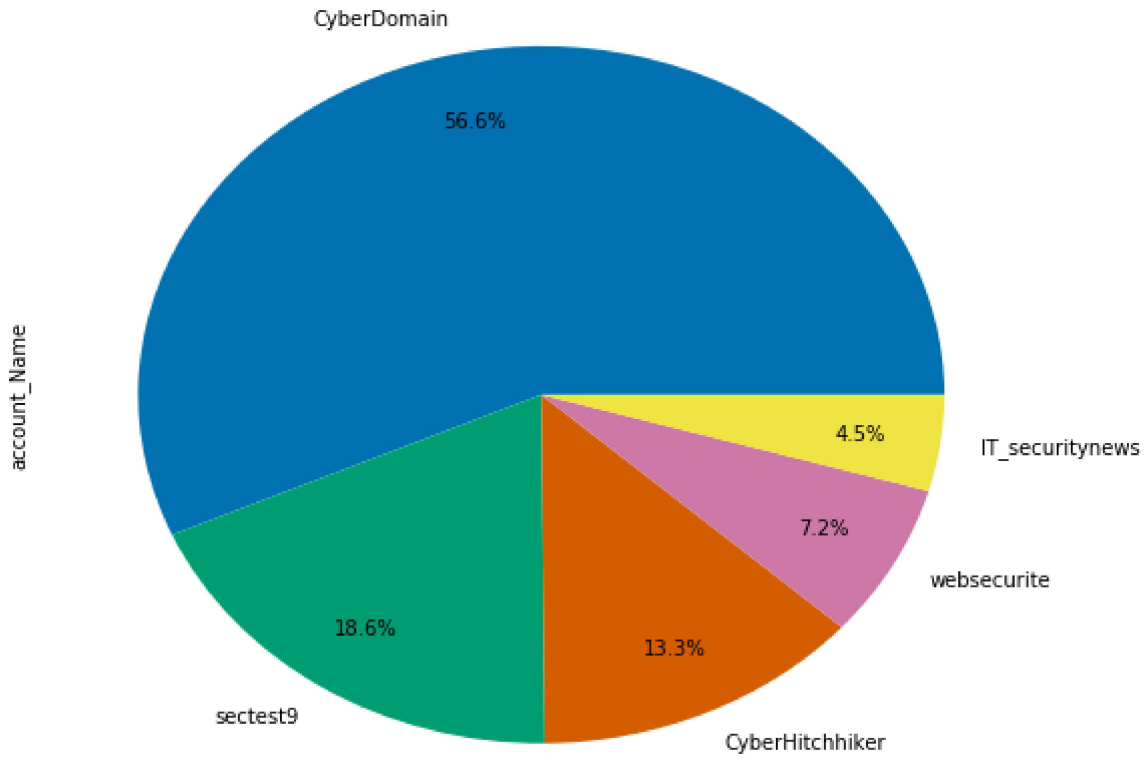
Out[10]:
```
CyberDomain        9923
sectest9           3254
CyberHitchhiker    2324
websecurite        1255
IT_securitynews     782
Name: account_Name, dtype: int64
```

In [11]:
```
top5_active_users=df[df.account_Name.isin([
    'CyberDomain',
    'sectest9',
    'CyberHitchhiker',
    'websecurite',
    'IT_securitynews']
    )]
# Top 5 active users
```

In [32]:
```
import matplotlib.pyplot as plt
plt.style.use('seaborn-colorblind')
#changing matplotlib style
```

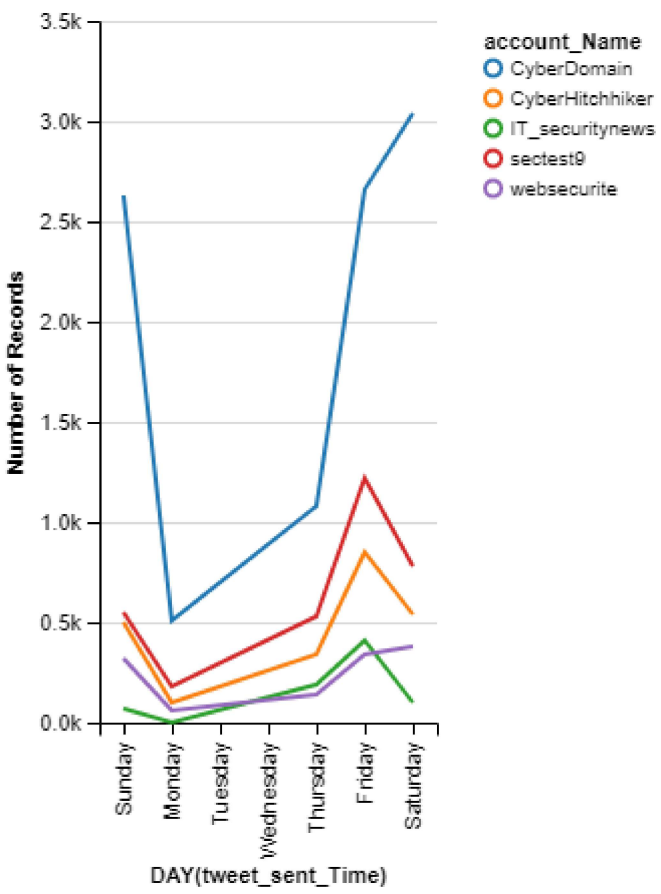
In [34]:
```
top5_active_users.account_Name.value_counts().plot(
    kind='pie',autopct='%1.1f%%',pctdistance=0.8,figsize=[9,8])
```

Out[34]: <matplotlib.axes._subplots.AxesSubplot at 0x1cfebb18a58>



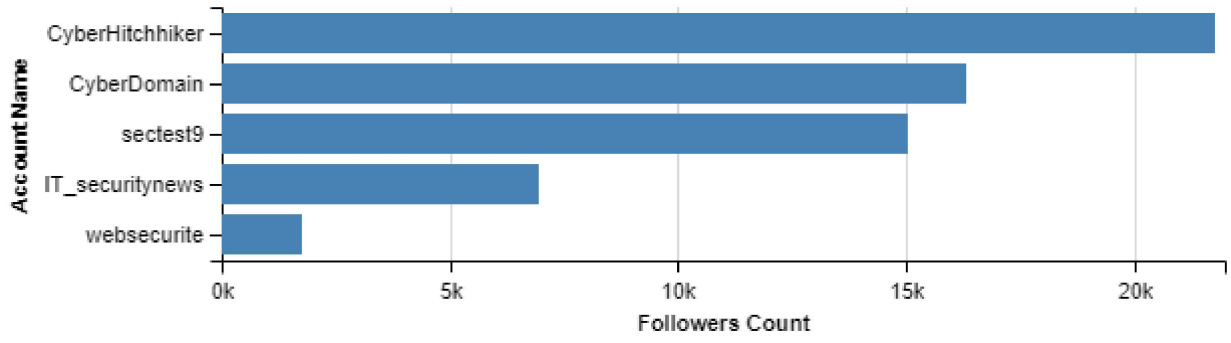
In [14]:
```
#Assign the maximum rows altair have to use (usually it's 5000 rows by default)
top_acc = alt.Chart(top5_active_users.to_altair())
top_acc.max_rows = len(top5_active_users)
```

In [43]:
```
top_acc.mark_line().encode(
    x=alt.X('tweet_sent_Time:T',timeUnit='day'),
    y='count(*):Q',
    color='account_Name:N'
)

# The most active user in Friday and Saturday is CyberDomain
```

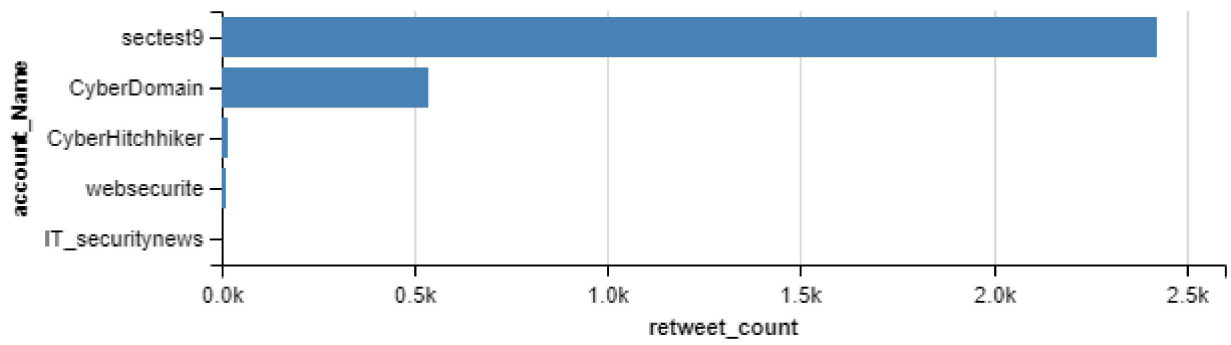


In [17]:
```
#Reset the top_acc Chart because we specified color before.
top_acc = alt.Chart(top5_active_users.to_altair())
top_acc.max_rows = len(top5_active_users)
```

In [50]:
```
top_acc.mark_bar().encode(
    x=alt.X('followers_count:Q',title='Followers Count'),
    y=alt.Y('account_Name:N',title='Account Name',
            sort=alt.SortField(
                field="followers_count",
                op="max",
                order="descending")
           ),
)

# The account's based on their followers numbers
```

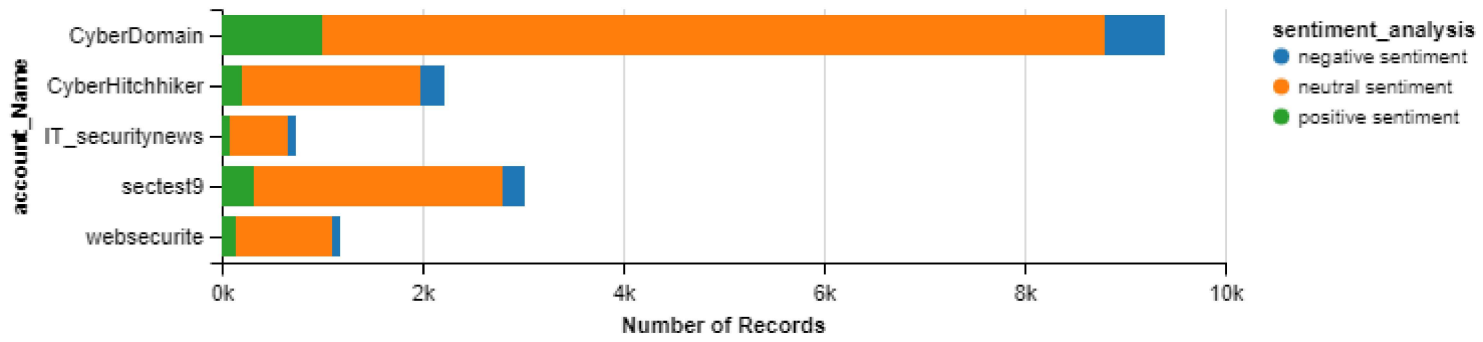


In [45]:
```
top_acc.mark_bar().encode(
    x='retweet_count:Q',
    y=alt.Y('account_Name:N',
            sort=alt.SortField(
                field="retweet_count",
                op="max", order="descending")
           ),
)

#The account's basen on their retweeted tweets
```

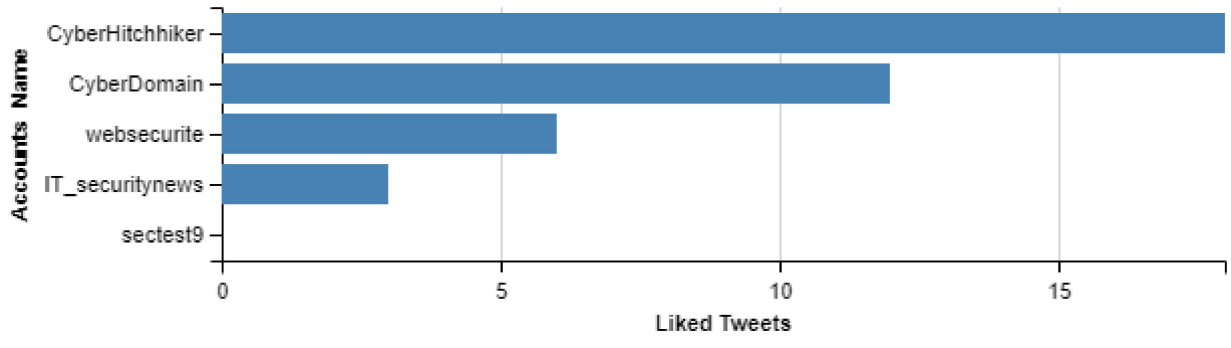


In [46]:
```
top_acc_sent = alt.Chart(
    top5_active_users[(top5_active_users.sentiment_analysis.notnull())
                      &(top5_active_users.sentiment_score.notnull())].to_altair())
top_acc_sent.max_rows = len(top5_active_users)
```

In [47]:
```
top_acc_sent.mark_bar().encode(

    x='count(*):Q',
    y='account_Name:N',
    color='sentiment_analysis:N'
)

#Here we can see each account and their sentiment analysis from their tweets
```



In [18]:
```
top_acc.mark_bar().encode(
    x=alt.X('favorite_count:Q',title='Liked Tweets'),
    y=alt.Y('account_Name:N',title='Accounts  Name',
            sort=alt.SortField(field="favorite_count",
                               op="max",
                               order="descending")
           ),
)

# Most liked tweets from the top 5 users
```



In [149]:
```
test=df.head()
```

In [164]:
```
test.head()
```

Out[164]:

	account_Name	userProfile_created_time	location	following_count	followers_count	account_total_tweets	tweet_sent_Time	twee
0	rachealgudaiti1	2017-03-25 20:47:39	None	30	0	42	2017-12-17 17:36:43	R @iFutureTek Key #GDPF &amp #NewYor #Cybe.
1	eescobar8127	2012-06-24 17:46:49	Gaithersburg,MD	5003	90	1984	2017-12-17 17:36:43	R @iFutureTek #CyberSecurit Attack is glo.
2	blystoneracIng	2014-04-15 04:06:25	Commodore PA	4436	100	2301	2017-12-17 17:36:40	R @iFutureTek #CyberSecurit Attack is glo.
3	JeremyhPh0to	2012-06-24 17:11:50	CDA, ID	5003	99	2192	2017-12-17 17:36:39	R @iFutureTek #CyberSecurit Attack is glo.
4	QueenB__Speakn	2015-03-02 02:31:01	Work & Home	355	33	1769	2017-12-17 17:36:38	R @iFutureTek #CyberSecurit Attack is glo.

In [200]:
```
df_growth=None
```

In [204]:
```
usrs=['CyberDomain','sectest9','CyberHitchhiker','websecurite','IT_securitynews']
crtd=[2010,2016,2011,2010,2011]
fllwrs=[16053,14935,21722,1739,6935]
df_growth=pd.DataFrame()
df_growth['account_Name']=usrs
df_growth['created_Year']=crtd
df_growth['followers_Count']=fllwrs

#Here we want to know the growth of the top 5 accounts by year
```

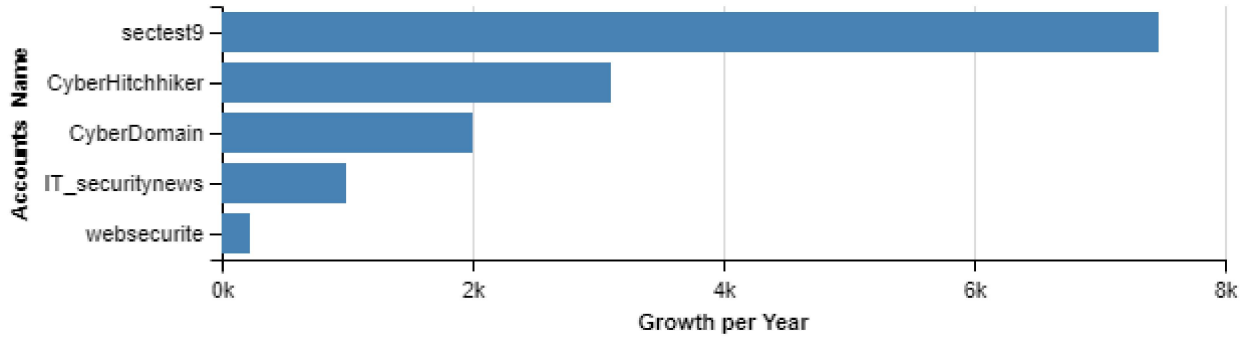
In [206]:
```
df_growth['Growth\Year']=2018-df_growth['created_Year']

df_growth['Growth\Year']=df_growth[
    'followers_Count']/df_growth['Growth\Year']

df_growth['Growth\Year']=df_growth['Growth\Year'].astype(int)
```

In [207]:
```
df_growth
```

Out[207]:

	account_Name	created_Year	followers_Count	Growth\Year
0	CyberDomain	2010	16053	2006
1	sectest9	2016	14935	7467
2	CyberHitchhiker	2011	21722	3103
3	websecurite	2010	1739	217
4	IT_securitynews	2011	6935	990

In [23]:
```
alt.Chart(df_growth).mark_bar().encode(
    x=alt.X('account_Growth:Q',title='Growth per Year'),
    y=alt.Y('account_Name:N',
            title='Accounts  Name',
            sort=alt.SortField(
                field="account_Growth",
                op="max", order="descending")
           ),
)

# The account based on their growth per year
```

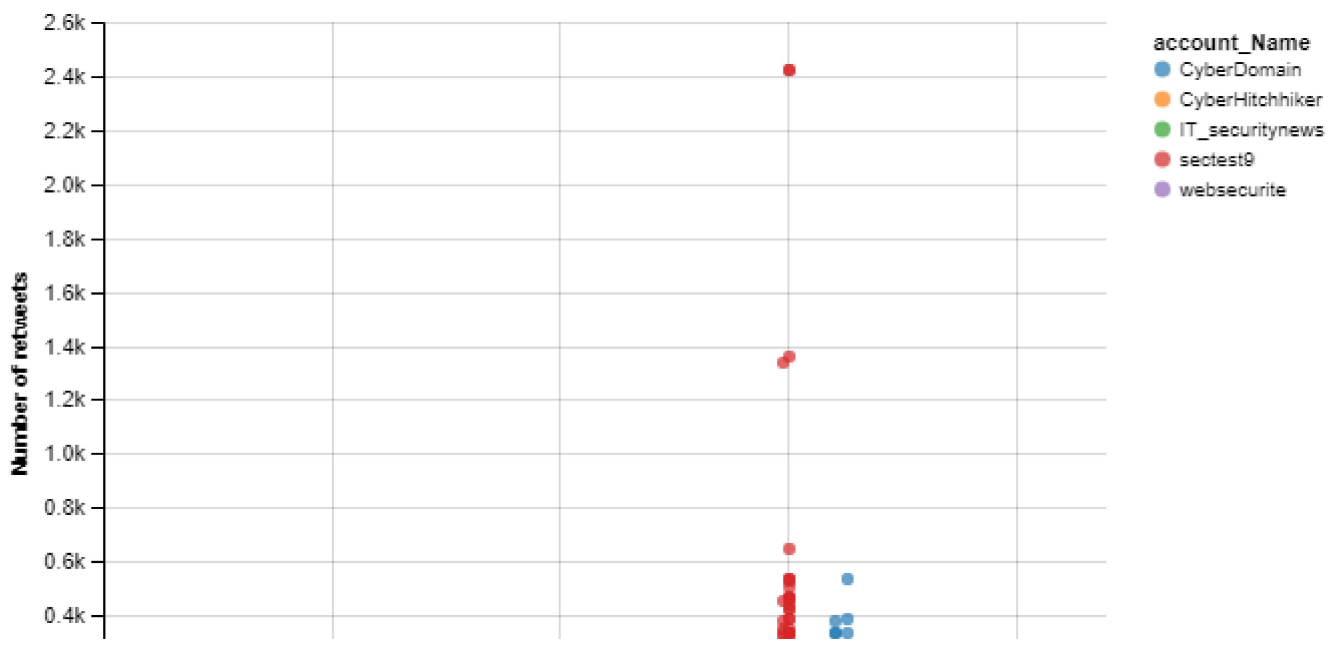


In [188]:
```
top5_active_users.mentions[
    top5_active_users.mentions!=''].value_counts().head()

#Those are the most mentioned accounts
```

Out[188]:
```
@CyberDomain        1297
@CyberHitchhiker     131
@Hackers_toolbox      68
@Fisher85M            60
@Sec_Headhunter       48
Name: mentions, dtype: int64
```

In [27]:
```
#Reset the top_acc Chart because we specified color before.
top_acc = alt.Chart(top5_active_users.to_altair())
top_acc.max_rows = len(top5_active_users)
```

In [31]:
```
top_acc.mark_circle().encode(
    x=alt.X('followers_count:Q',title='Followers Counts'),
    y=alt.Y('retweet_count:Q',title="Number of retweets"),
    color='account_Name:N'
)

# It's not important to have more followers to get retweeted more
```



In [54]:
```
df[['account_Name','retweet_count','tweet','tweet_sent_Time']].sort_values(
    'retweet_count', ascending=False).head(3)

# Most retweeted tweet talks aboat the
# internet neutrality problem and it was in friday
```

Out[54]:

	account_Name	retweet_count	tweet	tweet_sent_Time
89049	InfoSec_Ash	102621	RT @elonmusk: Nuclear alien UFO from North Kor...	2017-12-24 15:45:55
67397	Infosec_Tourist	100082	RT @MackenzieAstin: Hey, @AjitPaiFCC, today my...	2017-12-15 11:42:34
75221	aria_infosec	71046	RT @realDonaldTrump: Do you think Putin will b...	2017-12-15 05:45:55

In [187]:
```
df[['account_Name','account_total_tweets']].sort_values(
    'account_total_tweets', ascending=False).head(3)

# All of those had nothing to do with security but they retweeted
# or tweeted one and appeared in the dataframe
```

Out[187]:

	account_Name	account_total_tweets
197617	notiven	8271116
198266	notiven	8271116
67811	t_hisashi	3669506

In [56]:
```
df[['account_Name','followers_count']].sort_values('followers_count', ascending=False).head()

# Most of those are newspaper and not specialize in security
# ChelseaFC and RockstarGames appeared because they used "exploit" word in their tweets
```

Out[56]:

	account_Name	followers_count
198597	Reuters	19181792
57984	WSJ	15256793
26328	Forbes	14271882
159149	AlArabiya	13518291
167063	ABC	13069264

In [57]:
```
df.account_Name[(df.following_count>2000)&(df.followers_count<20)].head()

# After i checked these account i realized that they are bot's because
#they retweet same tweets even though their profile picture and name doesnt say that.
```

Out[57]:
```
40587     EmileeCatherin1
40865            Dinsaren
41726            JerryFaw
42809            maxcity_
137206     Live_In_Quotes
Name: account_Name, dtype: object
```

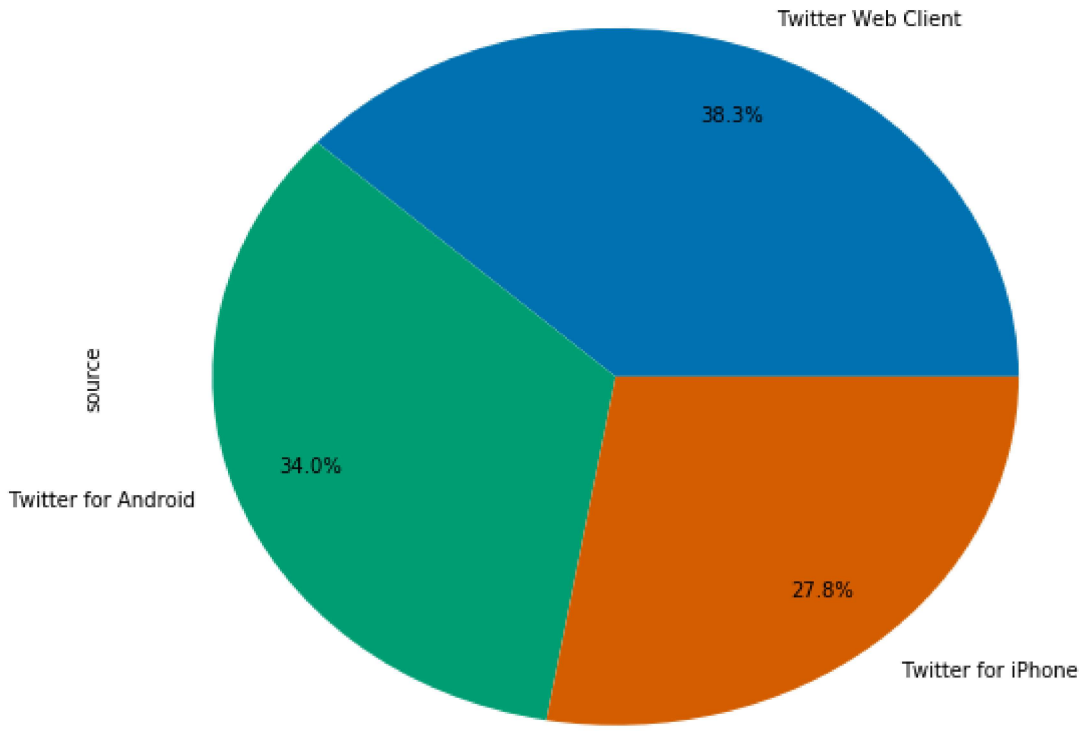
In [58]:
```
df.source.value_counts().head(3)
```

Out[58]:
```
Twitter Web Client     41501
Twitter for Android    36851
Twitter for iPhone     30095
Name: source, dtype: int64
```

In [36]:
```
top3_source=df[df.source.isin([
'Twitter Web Client',
'Twitter for Android',
'Twitter for iPhone'])]

#Here we have the top 3 used platform
```

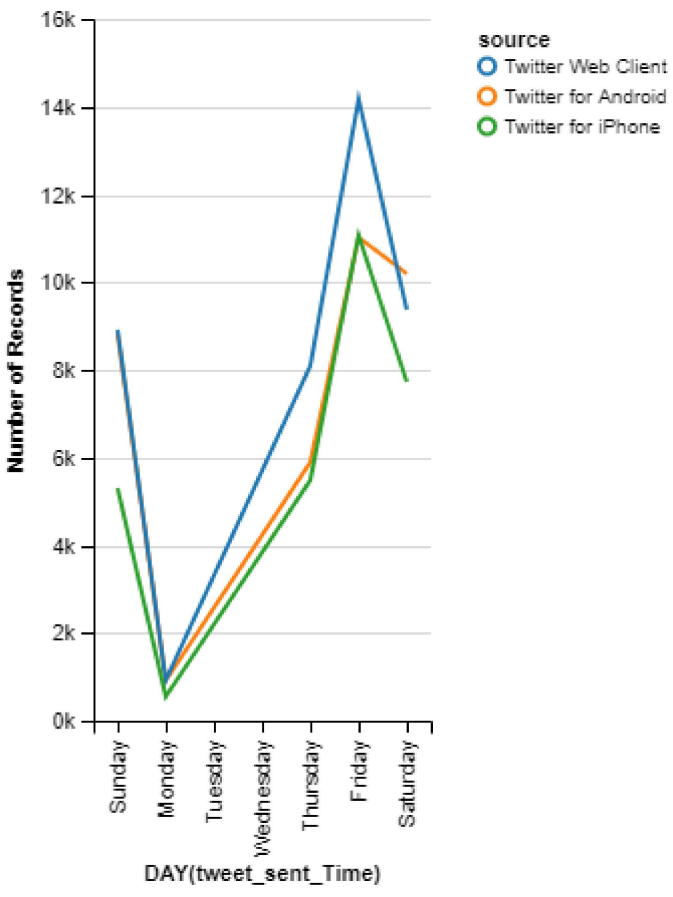
In [37]:
```
top3_source.source.value_counts().plot(kind='pie',autopct='%1.1f%%',pctdistance=0.8,figsize=[9,8])
```

Out[37]: <matplotlib.axes._subplots.AxesSubplot at 0x1cfebadc8d0>



In [63]:
```
#Assign the maximum rows altair have to use (usually it's 5000 rows by default)
top_src = alt.Chart(top3_source.to_altair())
top_src.max_rows = len(top3_source)
```

In [61]:
```
top_src.mark_line().encode(
    x=alt.X('tweet_sent_Time:T',timeUnit='day'),
    y='count(*):Q',
    color='source:N'
)

#From all the spike in friday we see that the web client is the most used one
```

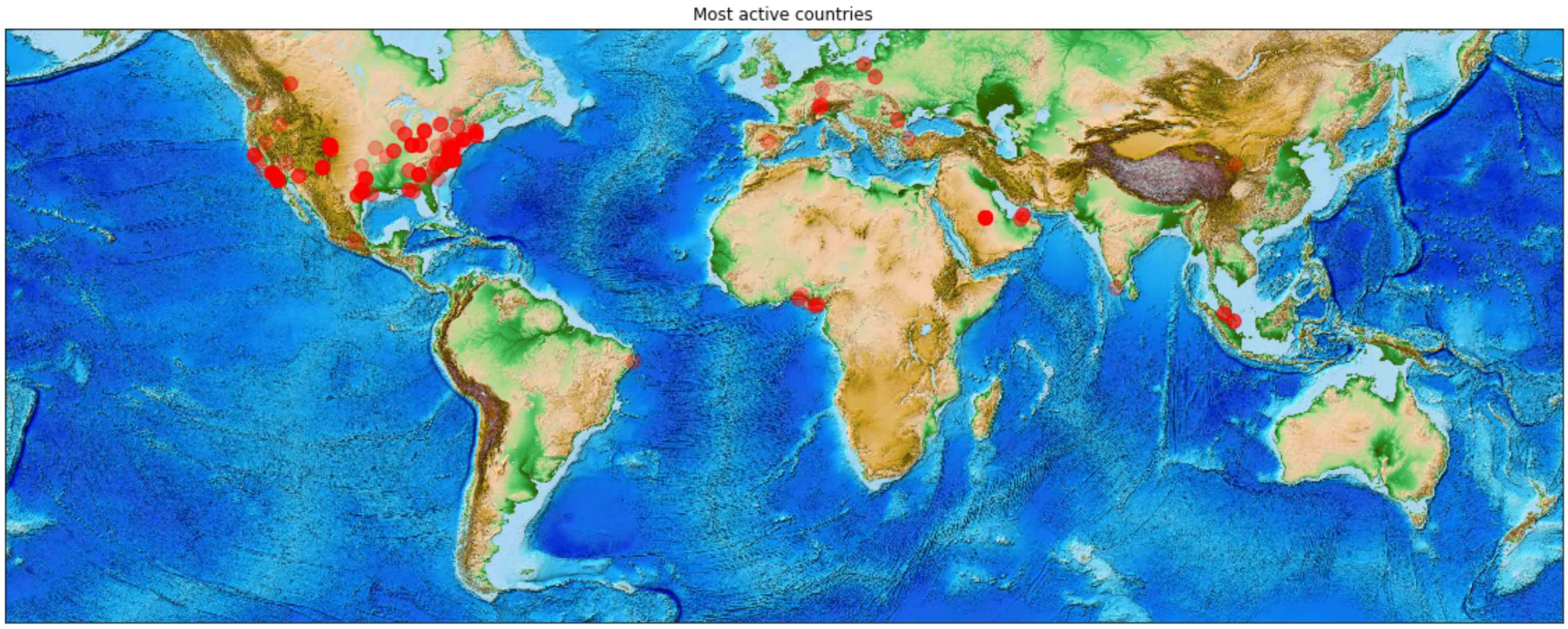


In [39]:
```
from nltk.collections import Counter
stopwords = nltk.corpus.stopwords.words('english')
stopwords.append('-')
stopwords.append('rt')

# RegEx for stopwords
RE_stopwords = r'\b(?:{})\b'.format('|'.join(stopwords))

# replace '|'-->' ' and drop all stopwords
words = (df.tweet
           .str.lower()
           .replace([r'\|', RE_stopwords], [' ', ''], regex=True)
           .str.cat(sep=' ')
           .split()
)

common_words=Counter(words).most_common(10)
common_words=[i for i in common_words if i[0][0] != '-']
#Here we assigned the most common words used in tweets to common_words
```

In [186]:
```
from wordcloud import WordCloud

wordcloud = WordCloud(background_color='white',width=1024,
                      height=768).generate_from_frequencies(common_words)
plt.imshow(wordcloud.recolor(random_state=6), interpolation="sinc",cmap='winter')
plt.axis('off')
plt.show()

# This word cload contain the most common words used in tweets
# And their size based on their frequency
```

In [122]:
```
df['latitude']=df['geo'][df.geo.notnull()].apply(lambda x: x['coordinates'][0])
df['longitude']=df['geo'][df.geo.notnull()].apply(lambda x: x['coordinates'][1])

# We created a new columns to make map plotting easier
```

In [157]:
```
import matplotlib.pyplot as plt
from mpl_toolkits.basemap import Basemap

plt.figure(figsize=(20,16))

m = Basemap(projection='mill',llcrnrlat=-60,urcrnrlat=60,\
            llcrnrlon=-180,urcrnrlon=180,resolution=None)

x, y = m(list(df["longitude"].astype("float")), list(df["latitude"].astype(float)))
m.plot(x, y,'.', markersize = 20, alpha = 0.3, color = "red")

m.etopo()
plt.title('Most active countries')
plt.show()

#The darker the red dot the higher number of users there interested in cyber security
```



In [ ]: