Twitter Search

API

Twitter stellt die v2 API frei zur Verfügung, allerdings mit Einschränkungen je nach Accounttyp. Diese Einschränkungen beziehen sich auf die maximale Anzahl von Tweets, die pro Monat abgerufen werden dürfen. Für uns sind das 500.000 Tweets pro Monat, das sind ca. 11 Tweets pro Minute. Außerdem sind manche Endpoints nicht verfügbar ohne z.B. einer akademischen Lizenz.

Python Bindings für die API

tweepy bietet leicht nutzbare Bindings der API an. Vermutlich weichen andere Angebote nicht besonders stark davon ab, allerdings kapselt tweepy einige Konzepte wie z.B. Sessions, was manchmal Aufwand sparen kann.

Endpoints

Zur Abfrage von Tweets gibt es folgende interessante Endpoints, für uns ist allerdings nur einer wirklich nützlich:

- `/2/tweets/search/all`: Suche Tweets mit Inhalt der der Suchanfrage entspricht
- `/2/tweets/search/recent`: wie oben, allerdings auf die letzte Woche beschränkt
- `/2/tweets/`: Rufe Tweet per ID ab
- `/2/tweets/sample/stream`: "Livestream" von einem Prozent aller Tweets

Der erste und letzte Endpoint sind nur mit der akademischen Lizenz verfügbar (und vielleicht auch gar nicht so nützlich). Einen Tweet per ID aufzurufen kann hilfreich sein wenn dieser schon als relevant gespeichert ist. Um neue Tweets zu finden kommt also nur `search/recent` in Frage.

Dieser Endpoint nimmt eine Suchanfrage entgegen und gibt passende Tweets aus der letzten Woche zurück. In einer Suchanfrage können mehrere Begriffe kombiniert werden, sodass entweder alle gesuchten Begriffe enthalten sein müssen, oder so dass einer der Begriffe ausreicht. Außerdem kann die Anfrage auf eine Sprache und einen engeren Zeitraum eingeschränkt werden. Die Antwort enthält zumindest die Tweet-ID, es ist allerdings nicht nötig die Tweets einzeln mit einer weiteren Anfrage abzurufen, da man in der Anfrage auch weitere Attribute mit anfordern kann. Dazu gehören unter anderem der Inhalt, Account des Verfassers sowie Statistiken (Anzahl Likes, Retweets, Antworten).

Retweets werden generell nicht als eigene Tweets behandelt. Erhält ein Tweet 50 Retweets, ist das unter den Statistiken des Tweets vermerkt, es werden aber nicht 50 Duplikate in den Suchergebnissen erscheinen. Ist man trotzdem an den Accounts, die den Tweet retweeted haben interessiert, kann man diese mit einer seperaten Anfrage abrufen. Im Gegensatz dazu zählen Antworten auf Tweets als eigenständige Tweets, die auch in der Suche beachtet werden.

Funktionsweise des Programms

Das Ziel ist, möglichst passende Tweets zum Thema Phishing herauszufiltern um Informationsmaterialien zu verbreiten. Dazu sind zwei wesentliche Schritte notwendig: Zuerst müssen alle Tweets grob nach dem Thema gefiltert werden. Danach sollen diese Tweets nach ihrer Relevanz priorisiert werden.

Filtern

Aufgrund der zur Verfügung stehenden Endpoints sowie der enormen Datenmenge aller Tweets muss das Filtern auf der Seite von Twitter vorgenommen werden. Deshalb muss die Suchanfrage passend gewählt werden. Der Suchterm "Phishing" ist naheliegend und schließt auch den gleichnamigen Hashtag und andere Groß-/Kleinschreibung mit ein. Ansonsten sind vielleicht auch grammatikalische Variationen wie "phishen", "gefisht" interessant. Wählt man zu generelle Suchterme läuft man in Gefahr um Größenordnungen mehr Tweets zu finden, die dann das eigentliche Ergebnis übertönen. Zum Beispiel liefert die Suche nach `hacking OR hacker OR gehackt` über 100.000 Tweets in einer Woche. Andererseits bringen sehr spezielle Suchterme, die nur in sehr wenigen Tweets enthalten sind wenig Mehrwert. Daher vermute ich, dass an der Suche eher wenig zu optimieren ist und stattdessen mehr Wert auf die Priorisierung gelegt werden sollte.

Trotzdem ist es eventuell sinvoll, das Ergebnis nachträglich weiter zu filtern. Beim lesen des Suchergebnis ist mir aufgefallen, dass es ein paar Accounts gibt, die sehr viele unnütze Tweets senden. Mit einer Blacklist, die vom Nutzer des Programms manuell bearbeitet wird, könnte man in diesem Fall Accounts ausschließen, die sicherlich keine passende Tweets senden. Allerdings ist es auch gut möglich, dass diese Tweets durch die Priorisierung sehr weit am Ende platziert werden.

Ergebnisse zum Suchbegriff "Phishing"

Die Suche `phishing lang:de` hat 809 Tweets zwischen 2022-05-01 22:55:05+00:00 und 2022-05-08 21:26:33+00:00, also einer Woche geliefert.

Ich habe diese Tweets grob in vier Kategorien unterteilt. Es gibt vermutlich Überlappungen oder Fehlkategorisierungen, letztenendes gibt es keine eindeutigen Kriterien - aber die ungefähren Verhältnisse sollten stimmen.

Konversationen

Hier geht es um Gespräche zwischen individuellen Personen, die sich über Phishing austauschen. Mit individuellen Personen sind Accounts gemeint, die nicht eine Institution (z.B. Zeitungen, Forschungseinrichtungen) repräsentieren. Trotzdem könnte es sich beispielsweise um einen Journalist für eine Zeitung mit eigenem Account handeln. Erkannt habe ich diese Tweets z.B. an Klarnamen, Tweet war Antwort auf einen anderen Tweet. Beispiel:

> id: 1522138539428827137
> time: 2022-05-05 08:58:02+00:00
> user: 🖐🏼🏡💉💉💉😷🌈Kerstin dj @dujoxy
> text:
> RT @deroadebicher: Hab gerade ne Phishing Mail von "PayPal" bekommen. Die hatte sogar die Fußzeile mit den Tipps zum erkennen von Phishing...

News

Nachrichten über aktuelle Phishingversuche. Hauptsächlich von Accounts, die Onlinemagazine repräsentieren oder Accounts die exklusiv solche Nachrichten verbreiten. Diese Tweets sind hauptsächlich am Nutzernamen erkennbar. Außerdem enthalten sie oft Links zur Website die wohl ausführlicher Bericht erstattet. Beispiel:

> id: 1522560837202120704
> time: 2022-05-06 12:56:06+00:00
> user: heise online @heiseonline Verified
> text:
> NFT-Marktplatz Opensea: Phishing-Angriff über Discord-Server https://t.co/hh0eTDo4ca #Opensea #Phishing

Werbung

Werbung für Produkte, Kurse oder Informationen, die gegen Phishing helfen sollen. Auch hier sind die Tweets am Nutzernamen erkennbar. Die Kategorie grenzt sich aber von News dadurch ab, dass der Inhalt genereller ist und sich nicht auf ein spezielles Ereignis beschränkt. Wahrscheinlich ist die Unterscheidung zwischen News und Werbung aber sehr uneindeutig. Beispiel:

> id: 1521819420699549697
> time: 2022-05-04 11:49:58+00:00
> user: ALL4NET GmbH @all4net_gmbh
> text:
> Nur informierte Mitarbeiter wenden Schaden ab.
> Wir sensibilisieren Ihr Team gegenüber Bedrohungen und Betrugsversuchen:
> E-Mail-Sicherheit #phishing
> Identitätsbetrug #CEOFraud
> den sicheren Umgang mit Passwörtern, etc.
> Termine/Info: https://t.co/LLq8vKtZfw
> #Security #Mittelstand https://t.co/v7VMczjMMU

Spam

Hauptsächlich kryptische Links und kaum Text, wird auch oft fälschlicherweise als deutsch klassifiziert obwohl es englisch ist. Dieser Account allein hat über 100 Tweets aus dem Suchergebnis erstellt. Beispiel:

> id: 1520915790975090688
> time: 2022-05-01 23:59:16+00:00
> user: KesagataMe @KesaGataMe0
> text:
> #Phishing #ETC #ETC利用照会
> #フィッシング詐欺
>
> IP:155.94.235.188
> (AS 8100 / ASN-QUADRANET-GLOBAL)
>
> hxxps://frxpfq.cn
> hxxps://ypfsg.cn
> hxxps://fqwiqi.cn
> hxxps://gzkop.cn https://t.co/So7C3JlkWp

Die 809 Tweets sind wie folgt aufgeteilt:

- Konversationen: 104

- News: 125
- Werbung: 93
- Spam: 487

Wenn man nach grammatikalischen Variationen des Begriffs sucht, erhält man weitere Antworten. Allerdings sind das in einem Zeitraum von einer Woche (aber nicht die selbe wie oben) nur ein bis zwei Tweets jeweils für die Terme "gefisht", "gefished", "phishen". Für "hacken" und entsprechende Variationen erhält man hingegen über 100.000 Tweets in einer Woche.

Priorisierung

Die zurückgegeben Tweets sollen dann anhand ihrer Relevanz sortiert werden. Die Twitter API bietet nur die Möglichkeit, nach Zeitpunkt des Tweets und nach Relevanz zu sortieren. Diese "Relevanz" bezieht sich allerdings nur auf die Suchterme direkt, wenn ein Tweet also mehrere gesuchte Terme enthält wird er höher priorisiert. Für eine bessere Sortierung können wir die mitgelieferten Daten zu den Tweets verwenden. Beispielsweise sind folgende Kriterien möglich:

Kriterien für die Priorisierung

- Anzahl der Likes
- Anzahl der Retweets
- Anzahl der Antworten
- Anzahl der Follower des Accounts
- Verifikationsstatus des Accounts
- Zeitpunkt des Tweets (ältere Tweets sind weniger relevant)

Für jeden Tweet kann dann ein Prioritätswert bestimmt werden. Dieser könnte zum Beispiel als gewichtetes Mittel aus den einzelnen Kriterien berechnet werden. Da die Werte mancher Kritierien stark gestreut sind (manche Tweets haben 5 Likes, andere 20000) ist es vielleicht sinvoll stattdessen den Logarithmus oder die Wurzel des Wertes zu verwenden. Anschließend werden die Tweets sortiert nach ihrer Priorität ausgegeben oder gespeichert.

Außerdem wäre es möglich die Ergebnisse abzuschneiden, sobald der Prioritätswert zu gering ist. Ein passendes Verfahren, das zuverlässig relevanten Tweets hohe Werte gibt, muss wahrscheinlich experimentell bestimmt werden.

Retweetet ein Account mit vielen Followern einen Account mit nur wenigen, hat das (fast) keinen einen direkten Effekt auf die Priorisierung (Allerdings führt dies wahrscheinlich zu mehr Likes, Retweets und Antworten). Es ist aber naheliegend, dem Tweet größere Priorität zuzuweisen. Dazu könnte man zusätzlich alle Accounts die den Tweet retweeted haben in die Berechnung mit einfließen lassen. Dafür müssen aber mehr Anfragen gestellt werden, was etwas aufwändiger wäre.

Eine weitere Möglichkeit wäre, über verschiedene Anfragen hinweg zu speichern, wie sehr ein Account priorisiert wurde und insbesondere, ob dem Account durch den Nutzer des Programms geantwortet wurde. Dieser Account könnte dann in Zukunft stärker priorisiert werden.

Offene Fragen

Wie soll das Programm verwendet werden?

- das Programm läuft als Service im Hintergrund und fügt neue relevante Tweets der Liste (.csv o.ä.) hinzu
- das Programm wird vom Nutzer gestartet wenn er/sie Tweets senden will. Dann sucht das Programm relevante Tweets und gibt sie aus