How Leaked Twitter API Keys Can be Used to Build a Bot Army

CloudSEK's BeVigil, the world's first security search engine for mobile apps, uncovered 3207 apps, leaking Twitter API keys, that can be utilized to gain access to or to take over Twitter accounts.

Key findings

- BeVigil discovered that 3207 apps were leaking valid Consumer Key and Consumer Secret.
- 230 apps, some of which are unicorns, were leaking all 4 Auth Creds and can be used to fully take over their Twitter Accounts to perform critical/sensitive actions such as:
    - Read Direct Messages
    - Retweet
    - Like
    - Delete
    - Remove followers
    - Follow any account
    - Get account settings
    - Change display picture

At first glance, it looks like what all can go wrong with just a bunch of API keys and tokens? Well, in short, "Everything!" Let's try and build an army of Twitter bots with this data and see where it goes.

The Misinformation War

The Twitter bot army that we will try to create can fight any war for you. But perhaps the most dangerous one is the misinformation war, on the internet, powered by bots. Time Berners-Lee, the founding father of the internet said that it is too easy for misinformation to propagate because most people get their news from a small set of social media sites and search engines that make money from people clicking on links. These sites' algorithms often prioritize content based on what people are likely to engage with, which means fake news can "spread like wildfire".

Twitter, is one such major social networking site, wherein its handles can be easily used to disseminate misinformation, thereby amplifying its reach. The flip side of this story is that scams and threats can be intricately weaved into this communication ploy, appearing to be legitimate.

Twitter was recently exploited to promote the “fake suspension notices” phishing scam. Verified handles were used to lend credence to the scam. It also played an active role in the US presidential elections in 2016. Twitter further raked up a storm when it was used to spread misinformation during the COVID 19 pandemic.

As with any social networking site, the buck does not stop at mere networking. Twitter takes it one step further because it is the sole medium of news and information for many of its users. Hence, multiple account takeovers can be used to sing the same tune in tandem, reiterating the message that needs to be disbursed.

Understanding The Tools

The tools needed for a bot army are forged using Twitter endpoints and services such as APIs.

What is Twitter API and How Does it Work?

API (Application Program Interface) in general is a means to extend an application’s data and functionality for other developers. These developers could be the internal team or the open-source developers. The beauty of an API is that a developer does not need to know how it is implemented, only how the interface works. So, all the messy coding is taken care of in the backend and the developer is presented with a neat interface.

In the same sense, the Twitter API enables access to the Twitter application. This allows a developer to access the core functionalities of Twitter such as reading and sending Tweets, Direct Messages, Following and Unfollowing users, etc. By allowing access to their APIs, Twitter ensures that developers can come up with their own unique ways of embedding Twitter’s data and functionality in their applications. For example, if a gaming app posts your high score on your Twitter feed directly, it is powered by the Twitter API.

Since the Twitter API provides direct access to a Twitter account, there must be some form of authentication involved. Sending passwords with each request to the API is not an efficient and secure

method. Hence, OAuth tokens are used by the Twitter API. OAuth ("Open Authorization") is an open standard for access delegation, commonly used as a means to grant API access without using the password each time. This standard is also used by Amazon, Google, Facebook, and Microsoft.

To understand it intuitively, suppose that you have the color red as your password. Now if you mix it with blue, you end up with purple color. Now when the app is presented with the color purple, it will know that red was involved too and hence it is valid. Similarly, you can make a new color combination and revoke a previous combination. Apart from OAuth token, the Twitter API also uses access controls such as:

App-Based Authentication: This provides access to all publicly available information such as tweets, shares, etc. Hence, it is not tied to any particular user and is known as App-Based Authentication. For this, an OAuth 2.0 Bearer Token is used. This can be obtained by passing the API Key and Secret through the POST oauth2/token endpoint.

User-Based Authentication: This can be used to access functionalities of a Twitter account, such as reading and sending messages. As the requests are tied to a particular account, it is known as User-Based Authentication. For this, the OAuth 1.0a authentication mechanism is used. This requires an Access Token combined with Access Secret. We can't have User-Based Authentication without App-Based Authentication. So, to perform a user-based authentication, we require all 4 valid authentication keys (Consumer Key + Consumer Secret + Access Token + Access Secret).

There are various types of subscriptions to the Twitter API such as 'Standard', 'Premium', and 'Enterprise' which can be seen from the image given below.

Feature summary

Category	Product name	Supported history	Operator availability	Counts endpoint	Data fidelity
Standard	Standard Search API	7 days	Standard operators	Not available	Incomplete
Premium	Search Tweets: 30-day endpoint	30 days	Premium operators	Available	Full
Premium	Search Tweets: Full-archive endpoint	The entire archive	Premium operators	Available	Full
Enterprise	30-day Search API	30 days	Enterprise operators	Included	Full
Enterprise	Full-archive Search API	The entire archive	Enterprise operators	Included	Full

What are Twitter Webhooks?

A webhook generally works by sending data to a web app in case of an event. For example, in the case of Twitter, an event can be someone sending a message to you or someone starting to follow your Twitter account. For this, the Twitter Account Activity API which is a webhook-based API is used. Webhooks are generally faster and require less effort than polling. Webhooks uses HTTP (Hypertext Transfer Protocol) requests to send data from one endpoint to another.

The Account Activity API delivers a JSON event payload every time an event on your Twitter account occurs. For more information, you can refer to the Twitter account activity API. A webhook differs from an API in the sense that an API is a two-way method, where the user asks for something and then gets the result for it. A webhook on the other hand is a one-way communication channel where the data can only be received in case of an event.

The use of Twitter webhooks requires OAuth 1.0a which is sometimes also referred to as “user context authentication. Hence, to use webhooks, you require your developer App’s Consumer Keys (API Key and Secret), alongside a set of user access Tokens (Access Token and Secret), as part of the authorization header in the API request. However, to use the Twitter webhooks, you need the paid version of Twitter API.

Building a Twitter Bot Army

Recruiting Soldiers

An army needs a large number of soldiers to prepare for an attack. In this case, they come from the vulnerabilities present in mobile applications. Often this vulnerability is the result of an error on the part of the developer. While developing a mobile application, developers use the Twitter API for testing. While doing so, they save the credentials within the mobile application at locations such as:

- resources/res/values/strings.xml
- source/resources/res/values-es-rAR/strings.xml
- source/resources/res/values-es-rCO/strings.xml
- source/sources/com/app-name/BuildConfig.java

Sometimes, these credentials are not removed before deploying it in the production environment. Once the app gets uploaded to the play store, the API secrets are there for anyone to access. A hacker can simply download the app and decompile it to get the API credentials. Thus, from here bulk API keys and tokens can be harvested to prepare the Twitter bot army.

Roll Call

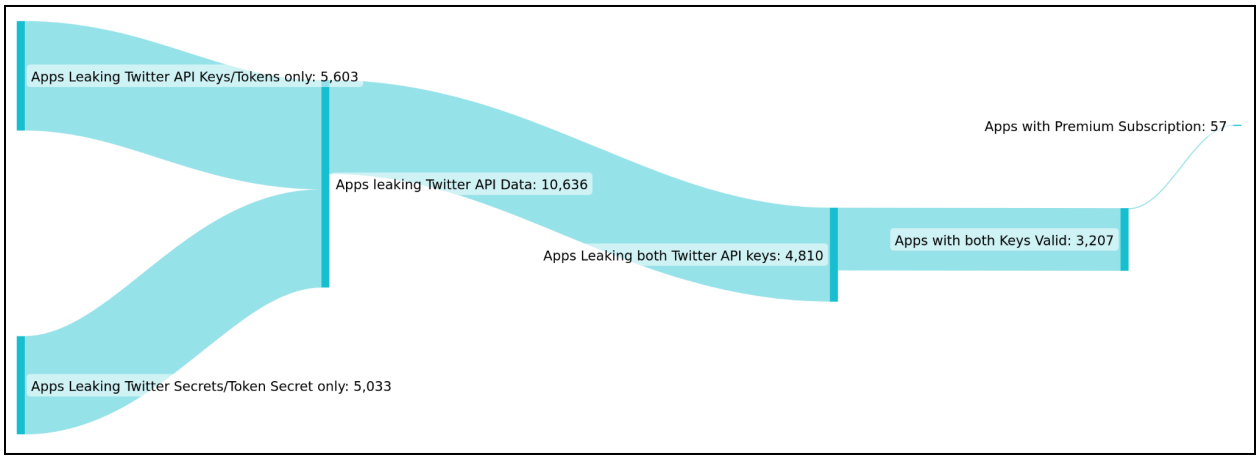
Researchers at Cloudsek inspected the mobile apps uploaded to BeVigil and observed that:

- 5603 companies were leaking Twitter API Keys/Tokens
- 5033 companies were leaking the Twitter Secrets/Token Secret only
- 4810 companies were leaking both the Twitter API Keys/Tokens and the Twitter Secrets/Token Secret

Out of 4810, 3207 companies had both the keys as valid ones. 57 of these companies had premium or enterprise subscriptions to the Twitter API, for which they were paying USD 149/month to Twitter.

In the case of the webhooks:

- Cloudsek researchers tested 3,300 companies for exposed Webhook access credentials
- Out of which, 57 Apps were found vulnerable



Breakdown of Vulnerable Companies

Out of 3207, 230 apps were leaking all the 4 Auth Creds. 39 of the apps had all 4 keys as valid. The Twitter accounts of these apps could be taken over to perform any critical/sensitive actions such as read DMs, Retweet, Like, Delete, remove followers, follow any account, get account settings, change DP, etc. Some of the leaked credentials belonged to verified Twitter accounts.

Attack

Now that the army is ready and prepared, the attack can be launched. The valid API keys and tokens can be embedded in a script to perform a variety of attack scenarios:

Scenario 1: Tweets and their subsequent retweets gain global attention. So, a Twitter bot army can be used to spread misinformation on any topic ranging from vaccines to elections. Thereby affecting millions across the globe.

Scenario 2: Twitter can be used to spearhead malware attacks through verified accounts passed on among legitimate followers. So, a Twitter bot army can run large-scale malware campaigns to infect systems, some of which could be critical infrastructure or SCADA systems.

Scenario 3: Spamming is another way to reach a massive audience and disseminate information related to cryptocurrency or the stock market. So, a Twitter bot army can be used to inflate or deflate the value of a cryptocurrency or the stock value of a corporation.

Scenario 4: Phishing is a strategy used by threat actors to obtain sensitive user information. And the collected PII can be used to launch other social engineering attacks or identity theft. So, a Twitter bot army automates phishing, on a large scale, to collect credentials. The trust that users have in verified accounts can be used to lure even the most educated of users.

Although webhooks may seem harmless, by just sending activity data they can be used to track a Twitter user. In some cases, they can also be used to read the messages of the Twitter account.

Defending Against Attacks

It is imperative that API keys are not directly embedded in the code. Developers should also follow secure coding and deployment processes such as:

Standardizing Review Procedures: Ensure accurate versioning. Publication requires the code base to be examined, reviewed, and approved prior to versioning. Complying with standardized procedures prevents key exposures.

Hiding Keys: Variables in an environment are alternate means to refer to keys and disguise them. Variables save time and increase security. Adequate care should be taken to ensure that files containing environment variables in the source code are not included.

Rotate API keys: Rotating keys can help reduce the threat posed by leaked keys. Unused keys reduce the severity of invalidation. It is recommended to rotate keys every six months as existing keys get deactivated while new ones get generated.

Conclusion

Real-time information being the USP of social networking platforms such as Twitter, it is difficult to differentiate between truth and lies, both deliberate and accidental. Hence, it is important for social media platforms to ensure that they are not misused to spread misinformation.

It is equally critical for organizations to secure their social media data and prevent their verified handles from being used to spread misinformation. And this can be done by ensuring securing coding and deployment policies, in addition to using tools like BeVigil to scan for exposed keys and credentials.

References

- https://www.newscientist.com/article/2124201-web-creator-tim-berners-lee-speaks-out-on-fake-news/
- https://economictimes.indiatimes.com/magazines/panache/twitter-accounts-of-bill-gates-jeff-bezos-elon-musk-hacked-in-bitcoin-scam/articleshow/76991797.cms?utm_source=contentofinterest&utm_medium=text&utm_campaign=cppst
- https://bevigil.com/blog/hardcoded-github-personal-access-tokens-leak-159-private-repositories