# TwitterSOARx Integration Design Document (v1.3)

## Key

Green Highlight = Integration XSOAR commands; Commands the user can enter in the War Room after installing the integration

Blue Highlight = Twitter commands; Commands that will be run on Twitter’s API.

Orange Highlight = Pre Existing XSOAR commands; Commands the user could enter in XSOAR, but are automatically executed by the integration to simplify the setup process.

Pink Highlight = Tweepy API methods.

Yellow Highlight = Example commands; Integration commands populated with sample values. Descriptions of the command output will be provided for reference

## Use Cases

**Fraud Account**

As a company with a social media presence, we need to prevent fraud accounts from impersonating our company in order to protect the corporate reputation. We also want to allow customers to do the same. In order to achieve this, we need a reliable way to detect and report fraudulent accounts. To achieve this, we will create an integration that allows XSOAR to connect to Twitter’s API. We will create a searching feature with customizable filters (such as searching by name, keywords, verified status), to allow us to better find the results we are looking for. As we search for fraud accounts, we will support the ability to add benign accounts to a list to be excluded from further searches. These two means of filtration will allow us to efficiently search for and report fraudulent accounts.

## Definitions

**Entities**: Provide metadata about context posted on Twitter. Examples of entities include hashtags, user mentions, links, stock images, symbols, polls, and attached media.

**Relevant Result** - A result that may be of interest to the person conducting the search;Any result that appears within the first 30 results of the customized search

**Non-Relevant Result** - Any result that is does not classify as a Relevant-Result

**Public Account** - A Twitter account that anyone can view. All of the information shared by the account is viewable by the general public.

**Public Metrics** - Public information about the account. Includes Followers Count, Following Count, Listed Count, and Tweet count.

**Private Account** - A Twitter account that only an approved list of people can view. All of the information shared by the account is viewable by only those who have sent requests to follow the private account and the owner of the account has accepted the request

## Requirements for Fraud Account Scan:

The goal of the account scan is to expedite the process of finding fraudulent accounts. In order to accomplish this, the scan must be simple to run. This will be accomplished by having three commands to run from the War Room (listed below). In order for the customer’s requests to be validated by Twitter’s API, they must provide their API key, API key secret, Access Token, Access token Secret, and Bearer Token.

Once authenticated, the user can begin using the integration. In order to identify potential fraudulent accounts, the “twitter-get-users” search will return the top 15 search results that meet the user’s search parameters. This is done by querying Twitter’s API [GET users/search](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/follow-search-get-users/api-reference/get-users-search) endpoint. The search is performed by Twitter, identifying accounts similar to the search term provided by the user (the q value, defined below ). The results of the search will be displayed neatly using markdown in the War Room on XSOAR. The user also has the option to use the “twitter-get-user-info” and “twitter-get-tweets” commands

Authentication

Twitter requires authentication for most of its endpoints. There are 5 main credentials within Twitter’s Authentication:

* **API Key** - Like a username, this allows you to make a request on behalf of your app when combined with API Key Secret.
* **API Key Secret** - Like a password, this allows you to make a request on behalf of your app when combined with API Key.
* **Access Token** - This token allows you to make a request on behalf of the Twitter account that owns the App when combined with the Access token secret.
* **Access Token Secret** - This token allows you to make a request on behalf of the Twitter account that owns the App when combined with the Access token.
* **Bearer Token** - A credential used to allow the app to authenticate requests that do not require the API Keys or Access tokens.

Clients will need to provide their own API Key, API Key Secret, Access token, Access Token Secret, and Bearer Token. Since Twitter has a request limit, the customer will need to use their own keys. This allows each user to have their own rate limit rather than the community sharing a rate limit. In cases where API keys are not needed, just the Bearer Token will suffice. Tweepy has a built-in OAuth handler; and Twitter uses OAuth to authenticate its users. By storing the user’s API Key, API Secret Key, Access token, and Access Token Secret in an integration parameter and passing them to Tweepy’s OAuth Handler, the user’s requests can get authenticated automatically. The stored keys/tokens will be encrypted to protect the confidentiality of the user’s information. After installing the integration, the user can navigate to Settings > Integrations > Twitter Integration > Add Instance, and entering their keys/tokens in the corresponding text boxes under the “Instance Settings” tab on the left of the window.

## Commands

### **twitter-get-users**

This command executes a search for users given a specified query. It will then return the results of the search in markdown format in the war room. This command uses Tweepy’s API to authenticate using OAuth 1.0 and then to perform the search function.

| **Argument name** | **Corresponding Twitter parameter** | **Description** | **Required** | **Values** |
| --- | --- | --- | --- | --- |
| name | q | The name to search; Twitter will return accounts with similar names to the one entered in this argument | required | String  Ex: “Target” |
| page | page | The search page to retrieve results from | optional | Integer |
| count | count | The maximum number of potential users to retrieve per page. The maximum value is 20. The default value is 15. | optional | Integer  Maximum: 20 |
| include\_entities | include\_entities | Theentities node will not be displayed when set to *false*. The default value is *false*. | optional | True/False |

#### Underlying design

Makes an API call to Twitter API v1.1 [GET users/search](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/follow-search-get-users/api-reference/get-users-search), which provides relevance-based search results of public user accounts. This search is performed through Tweepy using the api call API.search\_users. The results are returned in JSON format. The JSON results will then be neatly displayed in the War Room in Markdown format.

#### Example

!twitter-get-users name=”Palo Alto Networks” page=”1” count=”5” include\_entities=”True”

Returns user accounts with names matching or similar to “Palo Alto Networks” located on page 1 of Twitter’s search results, and a maximum of 5 accounts will be displayed.

### Schedule

The twitter-get-users command will be implemented by July 16th, 2021

#### Constraints

Requires Authentication from the user to use the integration.

There is a rate limit of 900 requests per 15 minutes per user auth. If the rate limit is met, Twitter will throttle the remaining requests until the 15 minute interval is up. XSOAR’s performance is also of concern, generating and displaying 1000 results can negatively impact server performance. A default value of 30 will be set, and a maximum value of 300 will be coded into the integration to prevent the user from slowing down XSOAR’s operations.

The search results also only yield public accounts. This means the integration cannot detect private accounts. This is an acceptable risk however, because customers looking for the company are going to be searching for a public account rather than a private account. Private accounts do not have their information displayed to the public, so it also makes it difficult for private accounts to create a negative public-image.

| **Argument name** | **Corresponding Twitter parameter** | **Description** | **Required** | **Values** |
| --- | --- | --- | --- | --- |
| q | q | The tweet content to search; Twitter will return tweets containing strings matching or similar to the one entered in this argument | Required | String  500 character maximum |
| geocode | geocode | Returns tweets by users located within a specified radius of a specified latitude/longitude | optional | latitude,longitude,radius(mi/km)  Ex:  28.738659 -111.193820 1mi |
| lang | lang | Only displays tweets of the specified language, given in ISO 639-1 code. | optional | String (2 letters)  Ex:  Russian = ru  Swedish = sv |
| result\_type | result\_type | Specifies which type of search results the user would like to display. | optional | Recent: Returns only the most recent results  Popular: Returns only the most popular results  Mixed: Returns a mix of both popular and recent results. |
| count | count | The number of tweets to return per page. | optional | Integer  Maximum = 100 |
| include\_entities | include\_entities | Theentities node will not be displayed when set to *false*. The default value is *true*. | optional | True/False |
| from\_user | N/A | The name of the user to search tweets by. All tweets returned will only have been made from the specified account. | optional | string |
| to\_user | N/A | The name of the user to search replies by. All tweets returned will only have been made in reply to the specified account | optional | string |

#### Underlying design

Makes an API call to Twitter API v1.1 search tweets, (https://api.twitter.com/1.1/search/tweets.json) which provides relevance-based search results of public tweets. The results are returned in JSON format. The JSON results will then be neatly displayed in the War Room in Markdown format.

#### Example

!twitter-get-tweets content=”xsoar” geocode=”28.738659 -111.193820 10mi” lang=”ru” result\_type=”recent” count=”10” include\_entities=”True”

Returns the 10 most recent tweets (in the russian language) containing the string “xsoar” that are located within 10 miles of 28.738659 -111.193820.

#### Schedule

The twitter-get-tweets command will be implemented by July 16th, 2021

#### Constraints

Requires Authentication from the user to use the integration.

There is a rate limit of 180 requests per 15 minutes per user auth. If the rate limit is met, Twitter will throttle the remaining requests until the 15 minute interval is up. XSOAR’s performance is also of concern, generating and displaying 100 tweets with up to 500 characters each can negatively impact server performance. A default value of 20 will be set, and a maximum value of 100 will be coded into the integration, as Twitter limits the maximum value to be 100.

The search results also only yield tweets from public accounts. This means the integration cannot detect private accounts. This is an acceptable risk however, because customers looking for the company are going to be searching for a public account rather than a private account. Private accounts do not have their information displayed to the public, so it also makes it difficult for private accounts to create a negative public-image.

The API call used in this command (/1.1/search/tweets) is from Twitter API v1.1. This version will eventually be deprecated and no longer be supported. The v1.1 version had to be used as v2 is still in development and does not support a feature equivalent to search tweets.

### **twitter-get-user-info**

This command returns detailed account information about specified accounts.

| **Argument name** | **Corresponding Twitter parameter** | **Description** | **Required** | **Values** |
| --- | --- | --- | --- | --- |
| name | N/A | The username of a given account to search for additional information | Required | String |

#### Underlying design

Makes an API call to Twitter API v2 [GET](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/follow-search-get-users/api-reference/get-users-search) /2/users/ which provides information about a specified user. The information provided includes the user’s description, entities, date of creation, id, location, name, pinned tweet id, profile image url, protected status, public metrics, url, username, and verified status.

#### Example

!twitter-get-user-info usernames=”PaloAltoNtwks”

Returns the description, entities, date of creation, id, location, name, pinned tweet id, profile image url, protected status, public metrics, url, username, and verified status of the account whose username is “PaloAltoNtwks”.

#### Schedule

The twitter-get-users command will be implemented by July 23rd, 2021

#### Constraints

Requires Authentication from the user to use the integration.

There is a rate limit of 900 requests per 15 minutes per user auth. If the rate limit is met, Twitter will throttle the remaining requests until the 15 minute interval is up. The search results also only display information about public accounts. This means the integration cannot provide additional information about private accounts. This is an acceptable risk however, because customers looking for the company are going to be searching for a public account rather than a private account. Private accounts do not have their information displayed to the public so it also makes it difficult for private accounts to create a negative public-image.