TIKTOK FORENSIC SCRAPER TO RETRIEVE USER VIDEO DETAILS

by
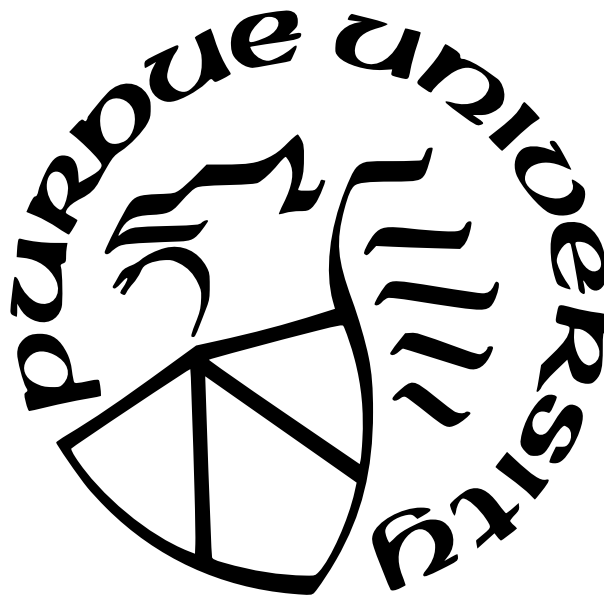
Akshata Nirmal Thole

A Thesis

Submitted to the Faculty of Purdue University
In Partial Fulfillment of the Requirements for the degree of

Master of Science

Department of Computer and Information Technology

West Lafayette, Indiana

December 2022

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Marcus K Rogers, Chair

Department of Computer and Information Technology

Dr.Umit Karabiyik

Department of Computer and Information Technology

Dr. Eugene H. Spafford

Department of Computer Science

Approved by:

Dr. John A Springer

ACKNOWLEDGMENTS

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Social media scraping [1] is a technique that helps us extract information of our interest from social media platforms such as Twitter, Facebook, TikTok, etc. TikTok [2] is a popular social media platform that allows a user to post short videos of different genres like pranks, stunts, tricks, jokes, dance, and entertainment. Criminal investigators have widely started analyzing the TikTok profile of a user to help them collect evidence to support their investigation, making it an important social media platform from OSINT (Open Source Intelligence) [3] perspective. Thus, it is forensically important for us to gather all the details of the videos posted by any user such as the date of posting to analyze the TikTok profile of that user. To fetch this data from TikTok, there is no publicly available API (application programming interface) that can directly provide all these details. TikTok also requires a CAPTCHA verification (verify yourself as a human) to load all the videos without which it will load a maximum of 30 videos. Due to this, popular techniques to perform web scraping such as using the Selenium library [4] in Python fails to get all the videos of TikTok and can fetch a maximum of 30 videos only. There are Python libraries [5] that claim to get all the user videos but with the addition of these security features in TikTok, these libraries also fail to provide all video details. There is various research work done on TikTok such as Forensic Analysis of the TikTok application [6], analysis of trending videos on TikTok [7], etc. but, there is no available research paper that talks about getting the video details for a TikTok user. Thus, the aim of this study is to propose a proof-of-concept tool that can automate the end-to-end flow to fetch the video details of a particular user, download the videos, and store all the data of interest by keeping the integrity intact to use them for further validation in future. The results show that details timestamp, caption, song, number of likes, shares, and comments can be retrieved from the user profile's HTML page, which can be stored in a database and used for further analysis. The sha256 [8] hash calculation of downloaded videos can be used to validate the integrity of the videos and make them potentially admissible in court.

1. INTRODUCTION

Web scraping [9] is a technique used to collect information that is of interest from a particular website. This is usually done using some tool that collects the specified information from the targeted website. Similar to web scraping, when useful information is retrieved from a social media platform such as Twitter, Facebook, TikTok, etc., it is called social media scraping. With the increasing popularity of these platforms, it provides criminal investigators a pathway to analyze a person from an OSINT perspective which is used as evidence in a court of law. One such popular social media platform is TikTok which allows a user to post short videos of different genres like pranks, stunts, tricks, jokes, dance, and entertainment. The focus of this study is performing social media scraping of TikTok to get all the details of the videos posted by a user, downloading the videos, and storing all the data of interest in a MySQL database.

1.1 Background

Web scraping was first introduced back in 1993 by the launch of JumpStation which was the first crawler-based web search engine [10]. Since then, web scraping has gained huge popularity and its use has extended to various domains such as E-Commerce Websites, Social Media Platforms, and Travel Portals to name a few. Different types of data of interest are extracted from various domains which are then used for further analysis using techniques like Data Mining, Machine Learning, and OSINT technologies. This information is then used for various purposes such as providing recommendations for travel or food, choosing the best marketing strategies, understanding the latest trends to make them popular, and many more. With the growing popularity of social media platforms, social media scraping has become extremely critical and vital to extract all sorts of data such as a users emotions, thoughts, hashtags, social media trends, etc. [11]

One such popular social media platform is TikTok which allows a user to post short videos of different genres like pranks, stunts, tricks, jokes, dance, and entertainment. It was originally launched by ByteDance in Beijing, China in September 2016 under the name *A.me* [12]. This was later renamed to *Douyin* in December 2016 but was available only in China.

The app finally made an international launch in September 2017 with the name TikTok. Since then, it has gained immense popularity and it is one of the most widely used apps in the world. It has been downloaded more than 130 million times in the United States and reached 3 billion downloads worldwide back in 2021 despite a ban on the app in India, one of its initial largest markets [12]. It is the first non-Facebook app to reach the download mark of 3 billion.

To create a video on TikTok, a user chooses background music from a variety of music genres and can save the video as a draft till they want to upload it for other users to view it. They can upload it using some trending hashtags that will get them popular on the platform. A user also has a section, "For You", where they can see other TikTok videos which are displayed either if the video is trending or the video genre is of the user's interest. To identify this, TikTok uses various AI algorithms to keep the user completely engrossed in the app. This has made TikTok one of the most addictive applications and users tend to spend much more time on TikTok as compared to other applications [13]. The app has had various negative impacts as well, especially on teenagers who have taken extreme steps to get popular on the app such as performing dangerous stunts. There have been cases of depression and even suicide if the person did not get the desired fame on the platform [14].

With TikTok now becoming widely popular, it is crucial to perform web scraping TikTok to get useful data from the application such as trending hashtags and videos, a user's emotions, likings, etc. Crime investigators have also widely started analyzing the TikTok profile of a user to help them collect evidence to support their investigation, making it an important social media platform from the OSINT (Open Source Intelligence) perspective. Thus, it is forensically important to gather all the details of the videos posted by any user such as the date of posting, comments, likes, shares, etc. to analyze the TikTok profile of that user.

1.2   Problem Statement

One of the popular techniques to perform web scraping is using the public API of the website which will provide all the required data. In python, this is mainly done using the

requests library [15] by making HTTP get and post requests. In the case of TikTok, there is no publicly available API that can provide all the required information for a user. There is an API that works well for TikTok to get the trending videos, but there is no such known API that can fetch the details of a particular user. Another method to perform web scraping is using popular tools and libraries such as Selenium in Python. TikTok requires a human verification called CAPTCHA without which it loads a maximum of 30 videos. TikTok identifies Selenium to be a test environment and fails the CAPTCHA verification while using Selenium. So, this technique also fails to get the details of all the videos posted by a user if the number of videos posted by a user is greater than 30. There are Python libraries that claim to get all the user video details but with the addition of these security features in TikTok, these libraries also fail to provide the details for all the videos. There are some publicly available GitHub repositories as well that support TikTok video downloading but they download either a single video or a maximum of 30 videos. Thus, there is no open-source tool or proof-of-concept currently available that can download all the videos of a user on TikTok and store all the metadata related to the user videos including date of posting, comments, likes, shares, caption, and song.

1.3 Research Questions

The research question for this study is as follows:

1. Can the forensic download of all the videos posted by users on TikTok be automated?

   - Can the metadata of each video posted by a user on TikTok be captured?
   - Can the information retrieved via the TikTok user profile be preserved after download for further validation purposes?
   - Can the videos which have been deleted by the user be identified?

1.4 Hypotheses

The hypotheses for this study are as follows:

1. H_1 : It is possible to automate the end-to-end flow to download all the user videos on TikTok.

2. H_2: It is possible to get the timestamp and other details of the video posted by the user.

3. H_3: It is possible to use the information retrieved from the TikTok user profile as a piece of evidence in court.

4. H_4: It is possible to identify videos being deleted or made private by the user.

1.5   Assumptions

The assumptions for this study are as follows:

1. TikTok application is accessible by the computer and the IP address has not been blocked by TikTok.

2. The username of TikTok is known before using the tool to get the videos of the user.

3. The user has downloaded all the required tools and set up the python libraries needed to successfully execute the tool.

4. The user has a basic understanding of Python and how to execute python files.

5. In case of any issues/errors faced while execution, the user knows basic debugging skills to identify and fix environmental issues.

6. The user does not aim to misuse the functionalities provided by the tool.

1.6   Limitations

The limitations of this study are as follows:

1. Due to a limitation of the number of downloads a user can perform for TikTok, this tool will work for less than 1000 videos at one time. Post this the response received is blank. The user should be able to resume the download task after a few hours.

2. During the time of the research, July - December 2022, TikTok has no policy that makes scraping a user profile illegal. If TikTok includes such a policy in the future, this research will be inapplicable.

3. The web browser used for the study is Chrome web browser as this was the only browser that supported the copy of the HTML body contents without the need for any human interaction.

4. The tool has been tested and developed using Python version 3.10. It has not been tested for versions greater than 3.10.

1.7 Delimitations

The delimitations of this study are as follows:

1. Python was the language chosen to develop the script as python supports numerous inbuilt libraries that can be used for specific tasks.

2. The tool is designed, developed, and tested for TikTok on a Windows 10 platform. The available device had Windows 10 platform so this was the operating system chosen for the study. Also, Windows 10 is the most popular and widely used version in the current times.

3. The fields extracted for a video are limited to timestamp, number of likes, number of comments, number of retweets, caption, and song of a video.

1.8 Contribution of the Study

The contributions of this study are as follows:

1. This study provides a proof-of-concept tool that can automate the end-to-end flow of fetching the video details of a particular user, downloading the videos, and storing all the data of interest in a MySQL database using python and Chrome web browser on a Windows platform.

2. Preserve the information retrieved via TikTok user profile after download for further validation purposes.

2. LITERATURE REVIEW

With the growing popularity of TikTok, the app has been the focus of several research topics such as identifying trending videos and hashtags, identifying various TikTok challenges and their safety, forensic analysis of the application, etc. In this section, the previous work related to the TikTok application is highlighted. Along with TikTok, research work done in similar social media applications such as Instagram, Twitter, etc. is also overviewed where the focus of the research aligns with the focus of this study. Research related to the forensics analysis of TikTok is also analyzed.

2.1 Implementation of Web Scraping to Build a Web-Based Instagram Account Data Downloader Application

In the paper, "Implementation of Web Scraping to Build a Web-Based Instagram Account Data Downloader Application" [16], the authors Himawan and Murdiyanto have used web scraping methods to download Instagram account data and manage the data collection like deleting and exporting data through a web interface. Similar to TikTok, Instagram is a popular social media platform that allows a user to post photos, and videos, and perform other actions too. It is used in various domains such as economy/business, education, and politics. Many groups take advantage of the insights obtained by processing and analyzing Instagram data for various purposes like data mining. The authors claim that most of the data collection is done manually. The tools that perform scraping for Instagram are not free and the APIs for Instagram have access restrictions to fetch data.

The authors perform an in-depth analysis of the Instagram application such as the various features provided by the app which include posting videos and photos, following other Instagram accounts, chats, etc. They also analyzed the various attributes related to the photos and videos to identify which of that information is useful and relevant for extraction. The application is built in Python. In order to perform web scraping, a request to the Instagram web page is made and from the response received the JSON object is extracted which has all the required information. Then the python library *Beautiful Soup* [17] is used to parse the JSON object to get useful data. All the relevant information is stored in a data

repository using the NoSQL concept. The tool has a web user interface built using the Flask library in python. This interface works as a data handler and provides functionalities such as deleting the user data and exporting it in various formats such as CSV.

Using the web scraping method, a maximum of 2412 post details can be retrieved. The tool testing has been performed using two main techniques: black-box and download analysis test methods. The application is not intended to download the Instagram post itself but only fetch the data relevant to the post such as likes, comments, and location.

2.2 Social Media Web Scraping using Social Media Developers API and Regex

In the paper, "Social Media Web Scraping using Social Media Developers API and Regex" [18], the authors have used APIs provided by Facebook and Twitter in order to perform web scraping. The authors claim that the main issue with respect to social media scraping is information overload, redundancy, and relevancy to the user. So, by performing web scraping they aim to develop a system that can search for relevant information and store it in a proper format to be presentable to the user. To use the APIs, the authors created an account on Facebook and Twitter. The data is fetched using the APIs and then parsed and stored in a database. The language used by the authors in order to develop this is JAVA.

Although the paper is published by ScienceDirect, it was presented in the *4th International Conference on Computer Science and Computational Intelligence.* This does not seem to be a very well-known or trustworthy conference, but, due to the lack of relevant work done in web scraping social media platforms, this paper is directly relevant to the study being performed.

2.3 A Web Scraping Methodology for Bypassing Twitter API Restrictions

In this paper, "A Web Scraping Methodology for Bypassing Twitter API Restrictions" [19], the authors present a web scraping technique for Twitter that bypasses Twitter API restrictions. Using the Twitter API to fetch data has a limitation and can retrieve only a limited number of queries. Another problem with Twitter is that historical tweets are available only for three weeks and there is no method to overcome this problem. The authors

have bypassed the API restrictions by optimizing the query fields in the Twitter search endpoint. They were able to fetch an unlimited volume of tweets by bypassing the date range limitations. They fetch all the tweets by using the pagination identifier parameter maximum_position. They have used Python to develop the tool and Scrapy as the main library to build the HTTP request packets and get the HTTP response packets. They built a GUI as well for testing the web scraping using the python Django library. With the tool developed, they successfully retrieved 64531 tweets from 2010 to 2017, which were not retrieved by a regular Twitter search API.

Even though the research topic for this study was Twitter, the authors focus on how to bypass the restrictions applied by an application which is a major focus of this study as well as TikTok also has various security features that need to be bypassed too in order to fetch video URL details of a user. This makes the research very relevant to the ongoing study.

2.4   Social Media Forensics - A Holistic Review

In this paper, "Social Media Forensics - A Holistic Review" [20], the authors Basumatary and Kalita have analyzed the existing problems in Social Media Forensics (SMF) and presented a generic taxonomy of the problems they have identified. With the increasing popularity of social media, there has been a significant rise in cybercrimes involving social media platforms called Online Social Network (OSN) crimes. Thus, Social Media Forensics has become very important. In order to improve the quality of SMF, it is crucial to identify the problems and issues that occur while performing SMF.

The authors have classified the problems and challenges into three types: Fundamental open issues, Legal issues, and Miscellaneous issues. The fundamental issues mainly cover the issues related to the quantity, diversity, complexity, consistency, and correlation of the data that can be retrieved through social media. Different social media platforms store data in different formats which makes it difficult to perform forensics across different platforms. The legal issues mainly deal with admissibility, ethics, privacy, and jurisdictional issues related to the evidence that can be retrieved from social media platforms. Miscellaneous issues are

mainly related to the additional security provided by these platforms such as end-to-end encryption, and data privacy.

One of the main challenges faced while performing web scraping of TikTok is the security that it provides like the need for human verification in order to stop bots from accessing it. This very closely relates to the miscellaneous issues described by the authors that make this research paper relevant to the study.

2.5 Digital Forensic analysis TikTok Android App

In the paper, "Post-mortem digital forensic artifacts of TikTok Android App" [21] the authors have performed a forensic analysis of the TikTok application on Android. They have also developed a TikTok module for Autopsy software which supports the extraction of TikTok artifacts. This can be used by digital forensic practitioners to extract relevant TikTok artifacts. For data population and testing, various devices with different android versions and TikTok versions were used. The authors have analyzed the data available in the rooted and non-rooted formats. They have shown the details of the information available in each of these artifacts retrieved. They discussed various XML and database files that were available. Most of the application data is stored on the cloud which makes it difficult to access in a post-mortem analysis. Despite this, the authors were able to fetch a significant amount of data such as messages exchanged, photos, videos watched, etc.

In the paper, "Forensic analysis of TikTok application to seek digital artifacts on Android smartphone" [22], the authors have done similar research to forensically analyze the TikTok application on the Android platform. They have collected evidence from the smartphone that can retrieve the list of followers, searching keywords, favorites, messages that have been exchanged by users, etc. in this application. The main tools used by them were the ADB tool which is a flexible command-line tool that helps in communication with Android devices, SQLite Database Browser which is used for reading database files, and a code editor that can read XML files.

Although the focus of this research is TikTok, it analyses the Android application of TikTok to track/fetch the details of the user activity on the application. However, this

research does not focus on the videos posted by the user. Nonetheless, since the research is focused on TikTok, it makes the research relevant to the ongoing study.

2.6   Trick and Please. A Mixed-Method Study On User Assumptions About the TikTok Algorithm

In the paper, "Trick and Please. A Mixed-Method Study On User Assumptions About the TikTok Algorithm" [7] the authors have focused on the trending algorithm of TikTok and what are the factors that can push the video to the trending section. The authors consider the assumptions that users make about how this algorithm works such as using specific hashtags that are popular, posting the video at a particular time, and then validating if these factors affect the post to be pushed to the trending section. To test this, they collected information on 300,000 trending videos on TikTok using an open-source scraper [23] and conducted interviews with 28 users to get their understanding of the algorithm. Upon analysis, they concluded that the video engagement such as increased likes, comments, etc. on the video, the posting times, and adding relevant hashtags increased the chance of the video getting trending. This also aligned with the user's assumptions about how to push the video to the trending section.

This paper focuses mainly on TikTok data and its analysis. However, the focus is on the trending videos section and not any particular user. Upon trying and testing the open source tool used by them to collect TikTok information [23], no data was available for a particular user or even the trending videos. Since the paper focuses on TikTok and getting data from TikTok, it is relevant to the current study.

2.7   User-Generated Short Video Content in Social Media. A Case Study of TikTok

In the paper, "User-Generated Short Video Content in Social Media. A Case Study of TikTok" [24] the author Aliaksandra Shutsko analyzed the content of 1000 videos on TikTok to analyze and find which genre of the videos is most popular on TikTok. The study is focused on Germany as the app is required to choose a country to set up an account. The author used the content analysis technique to conduct the research to quantify the contents and achieve

results. The main research questions that the author aimed to answer were: Which content posts users on TikTok? Are there gender differences among creators in content preferences? Videos of which content are the most popular on TikTok? Do potential law infringements occur on TikTok? The author divided the videos into 26 content categories. Along with this, another category was added - "Sexy/Flirting" to identify any such activity going on in the video too. Videos could be assigned to multiple categories as well, depending on the video content.

The author concluded that the videos belonging to the Comedy and Joke genres were the most popular among both males as well as females. Categories like Beauty were more popular among females whereas videos categorized as "Reaction" to another video were more popular among males. In videos categorized as "Sexy/Flirting", it was observed that females appeared in the majority of those videos. The author has also tried to identify potential legal infringements related to TikTok. Personal rights violation was observed in 19 of the videos and copyright issues in 304 of the videos which is a very high count and of high concern.

This study focuses on what types of videos are popular on TikTok and understanding what users like and post on TikTok. This is a broader scope as compared to the scope of the ongoing study as the current study focuses on the videos posted by one particular user only and performs social media analysis of that user.

2.8   Security, Privacy and Steganographic Analysis of FaceApp and TikTok

In the paper, "Security, Privacy and Steganographic Analysis of FaceApp and TikTok" [25], the authors have done a forensic analysis of the FaceApp and TikTok apps from the Android play store to determine the security posture of these apps from different perspectives such as data ownership, management, and privacy and steganographic use. To get the details of the privacy and management of the Apps, they used Androguard to perform reverse engineering of the two apps. Androguard functions such as get_permissions(), get_details_permissions(), and get_app_name(). were used to get the required details. They analyzed what type of information was collected and stored by these applications. Sensitive information such as Credit card details was also retrievable. To analyze how media

was uploaded through the app, the Wireshark tool was used to capture the packet transfer. Upon analysis of this capture, the authors concluded that the images and videos were unencrypted and can easily be sniffed. To perform a steganographic analysis of the apps, the authors uploaded the videos with a hidden text file. While retrieving the apps from the devices, no such text file was found which means TikTok had algorithms to remove these embedded text files. This is important as malicious users will not be able to embed any malware or other harmful information in the video files while uploading them.

It is essential to understand how these applications protect the identities and the sensitive information of the user and keep them safe, which was the main aim of this research. However, this paper focuses on the android application of TikTok and does not analyze the web version of TikTok. Since this paper heavily focuses on TikTok and its analysis, making it directly relevant to the study.

2.9 Unofficial TikTok API in Python

This tool, Unofficial TikTok API in Python [5], is an open-source, unofficial API wrapper for TikTok in Python developed by David Teather. This python library TikTokApi provides various functionalities such as getting the most trending videos, downloading a video, and getting specific user information. Upon trying and testing this tool, it was discovered that the tool functionalities such as getting trending video details, downloading a particular video, etc. work as expected. However, the functionality to get user-specific information fails at CAPTCHA verification despite passing the parameters which the author suggests. Hence, this tool fails to provide user video details for a particular user. However, since the download video functionality works as expected, the library TikTokApi will be used to download the user videos in this study as well.

Similar to this, there is another public repository, owned by author Avinash [26] which also uses this TikTokAPI but claims to run it faster. This repository also fails at downloading all the videos of a user. There are numerous other Github repositories such as a TikTok video downloader script in PHP [27] and a fast light-weight scraper for TikTok written in Node js [28]. The PHP downloader downloads a single video only which requires the video

ID to be passed. The scraper repository retrieves data from a maximum of 30 videos only. Hence, these repositories also fail to provide the details of all the videos posted by a user.

2.10   4k TokKit

The tool "4k TokKit" [29] is a TikTok Downloader that downloads TikTok challenges, captions, whole accounts, hashtags, and song-related and single videos. It is a paid tool and without a subscription, downloads only 30 videos for a user. The total limit for a free user is only 50 videos per day. There are various types of paid users and for a pro user, the rate is $60, which is a one-time payment only. With the pro version a user can download unlimited videos each day and it successfully downloads all the videos of a user. The tool creates a database related to the download of the videos and the video creation date.

It does not capture any information regarding comments, likes, and shares of the video or if a video has been deleted later. This tool functions similarly to the ongoing study but the free version fails to download all the videos of a user.

3. METHODOLOGY

The aim of this study was to develop a proof-of-concept tool that can fetch all the videos for a particular user and store the details of the videos in a database file. This section describes the architecture of the tool divided into two main phases: The first phase is loading all the user videos on chrome and copying the HTML contents from the browser. This phase is referred to as the profile loading phase. The second phase is parsing the HTML page to get the video URLs and making further requests to get in-depth details about the videos. This phase is referred to as the data parsing phase. The details of the Python libraries used for development will also be discussed. The detailed procedure for setting up the environment and execution steps will be covered in this section. This tool is developed to run on the windows platform and using the Chrome Web Browser only.

3.1   Architecture Phase 1 - Profile Loading Phase

Fig: 3.1 shows the detailed architecture of the profile loading phase.
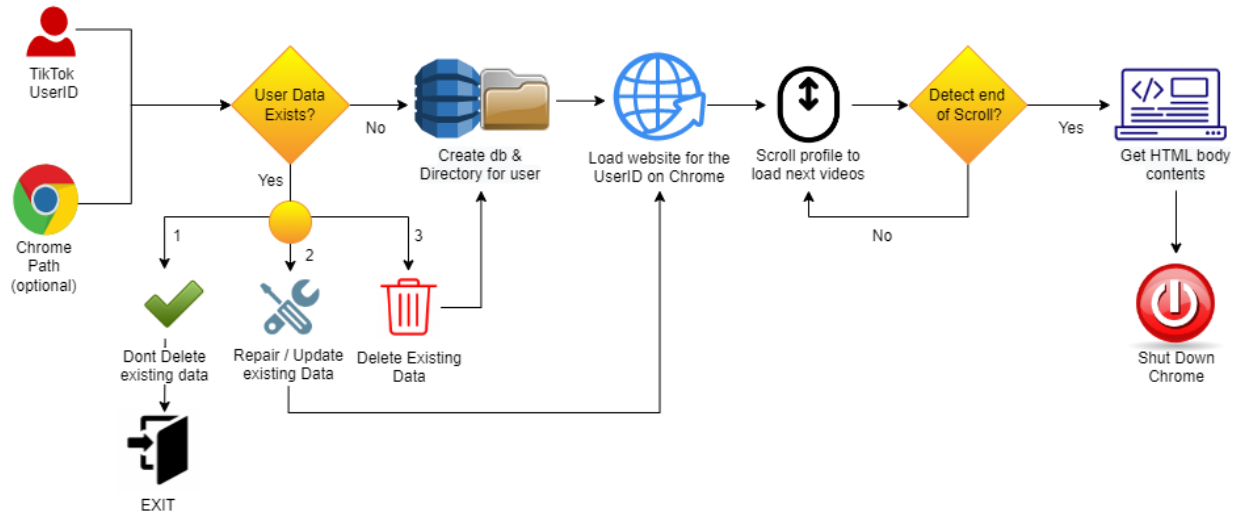
Figure 3.1. Design Architecture - Phase1

24

3.1.1 Step 1 - Argument parsing

The tool takes the TikTok userID as an input for which the scraping task is to be performed. Along with this, an optional chrome path parameter can also be passed to the tool. The input arguments provided to the script are parsed using the Python library *argsparser* [30]. Using this library, all the details of the arguments can be checked and validated. Optional as well as compulsory argument rules and details for the arguments can be set using this library.

3.1.2 Step 2 - Check for existing data

The next step is to check if there is existing data for the given user. If there is no existing data, then a new directory and database are created for the user. If existing data is found, the user can choose of the following three paths :

1. **Delete existing data:** In this case, the existing directory and database for the user are deleted and the user is treated as a new user. A new directory and database are created for the user.

2. **Repair/update existing data:** In this case, the user will be given two choices: The first choice is to update all the records for the user even if there is an existing data record for a particular user. Fields such as the number of comments, likes, etc. will be updated even if they exist. The second choice is to fix only those records that have missing values. If data already exists, then the video will not be checked for any updates on fields like the number of comments, likes, etc.

3. **Do not delete existing data:** In this case, the code will exit. This option is given in case the user does not wish to modify the existing records and wants to finish the script execution.

These checks, creations, and deletions are done using the Python libraries *os* [31] and *shutil* [32]. *Os* is a very popular Python library that provides the functionality to perform system-related operations. *Shutil* library is mainly used for copying and deleting files.

3.1.3 Step 3 - Load profile on Chrome

The next step is to load the TikTok profile of the user on the Chrome Web Browser. For this, the Python library *webbrowser* [33] is used. This library provides an interface to load web-based documents. It takes the path or the name of the browser where the website is intended to be loaded. If nothing is provided, it loads the website on the default browser of the computer.

3.1.4 Step 4 - Scroll all videos

The next step is scrolling the web page to load all the videos posted by the user. TikTok has an infinite scrolling facility to keep loading all the videos till the end. This action is performed by using the keyboard key-press technique. The Python library *pynput* [34] is used to perform the keyboard operations. It is a library that helps control and monitor mouse and keyboard operations. In order to identify the end of the scroll, image processing [35] has been used. This is a process of converting an image to digital form and then using the information for further usage. After a defined amount of scrolling, a screenshot is taken to capture the current contents of the screen. This is done using the *pyautogui* [36] library in Python. If there is a previous screenshot available, the current and the previous images are compared using l_0 norm [37]. The two image vectors are subtracted from each other and then the l_0 norm is computed by counting the number of non-zero numbers in the output vector. If the count is very less ≈ 0, then we conclude that there is no change on the website and all the videos have been loaded. The image processing is performed using the Python libraries *numpy* [38], *matplotlib* [39] and *scipy* [40]. Fig: 3.2 shows how the scrolling end is detected.

3.1.5 Step 5 - Get HTML contents

Once all the videos have been loaded, the next step is to get all the HTML content of the website. In order to do this, the inspect element tab on the Chrome Web Browser is opened. To directly open the elements tab, the shortcut *ctrl + shift + c* is used. The element tab

26
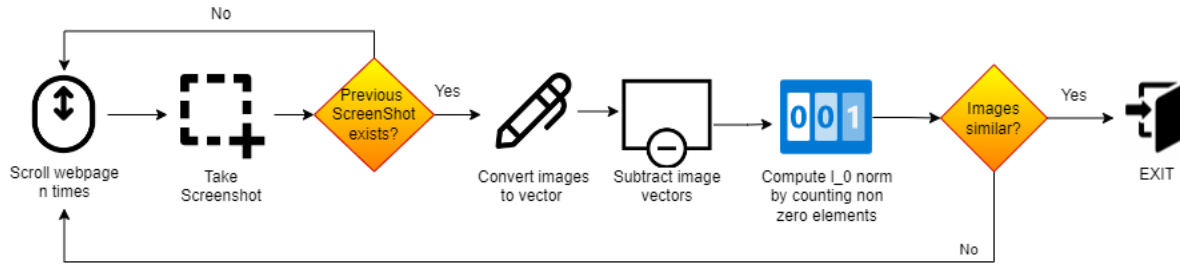
Figure 3.2. Identifying end of the scroll for website

selects the $< body >$ tag in the HTML web page by default. Copying this content gives us the entire HTML body content. This copy is done using the shortcut *ctrl + c*. These operations can be performed using the *pynput* library. This copy operation copies to the clipboard of the computer. This content is then passed to the tool so that it can be used for further processing. The Python library *pyperclip* [41] is used to paste the contents from the clipboard into the tool.

3.1.6 Step 6 - Shutdown Chrome Browser

All the contents of the HTML page that are needed for further processing are now inside the tool. The chrome browser is no longer needed and can be shut down. This is the final step of phase 1. It is performed using *pynput* library and using the shutdown shortcut *alt + f4*.

3.2 Architecture Phase 2 - Data Parsing Phase

Fig: 3.3 shows the detailed architecture of the data parsing phase.

3.2.1 Step 1 - Get Video URLs

The entire HTML body contents are now available for further parsing. The aim is to extract all the video URLs from the content available. Python library *Beautiful Soup* [17] is
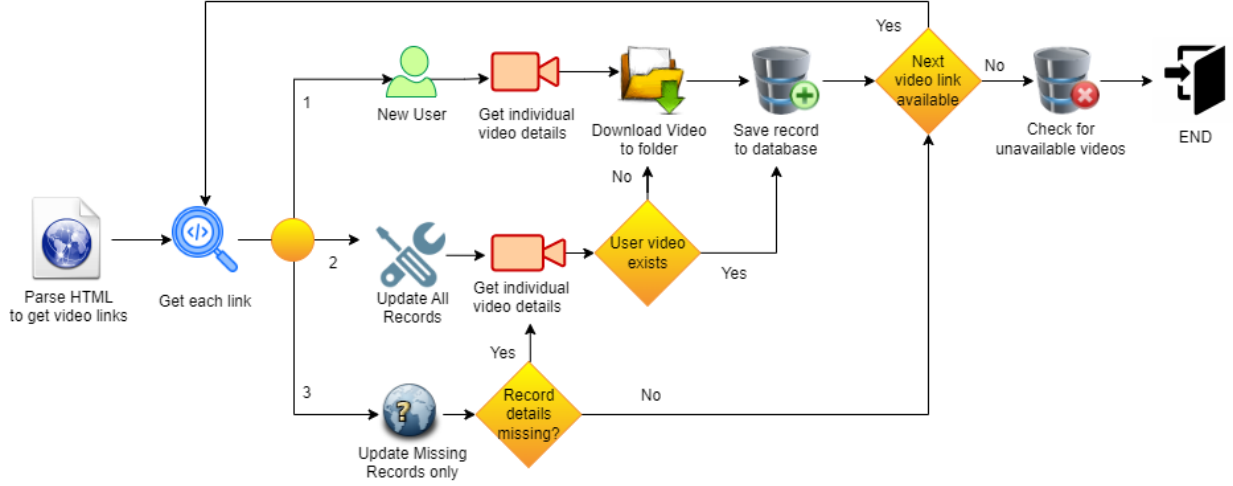
Figure 3.3. Design Architecture - Phase2

used to extract the video URLs from the entire web page. The Python regular expression library, *re* [42] is also used to match the exact structure of the video URL. Now, the list of videos is available which can be used to download the video and fetch further details about the video such as the number of likes, comments, etc. on the video.

3.2.2 Step 2 - Get further video details

From phase1, the user is identified as a new user or an existing user for which repairs need to be performed. Based on this the next step can be one of these paths:

1. **New User:** For a new user, the 1st step is to get further details about the video. For this, an HTTP get request is made using the *requests* [15] library in Python. It is a popular Python library mainly used for making various HTTP requests. The required details are then parsed using *beautiful soup* and *re* libraries in Python.

2. **Update all records**: If the user wishes to update the details of all the video records, (i.e., perform a thorough parse), the request is made to fetch further video details which are parsed using *beautiful soup* and *re* libraries.

28

3. **Update missing records only**: In this case, the records that have some type of missing detail such as missing timestamp is updated. If there is existing data available, it is ignored.

3.2.3   Step 3 - Download video

At this stage, all the details of the video are available. The next step is downloading the video and storing it on the local device. For a new user, the video is simply downloaded without any checks. This is done using the Python library, *TikTokAPI* [5]. It is a publicly available library dedicated to TikTok that provides functionalities for various tasks such as fetching trending videos, downloading videos, etc. In case of repair/update record flow, only if the video is not available in the directory of the user, the video is downloaded.

3.2.4   Step 4 - Update Database with video record

All the video details and the downloaded video are now available at this stage. The final stage is to update the database record for the video created for the user. For a new user, all the details are entered as a new record in the database. In case of update/repair flow, the existing record entries are updated with the latest details. All the database operations are performed using the Python library *sqlite3* [43]. It is a Python library that supports SQL operations for a MySQL database. In order to maintain the integrity of the video files downloaded, the sha256 hash of the video is computed and stored in the database. This is done using the Python library *hashlib* [44]. The script performs the phase2 steps for all the videos available.

3.2.5   Step 5 - Check for unavailable video records

Once all the video URLs have been parsed, the final step is to check if any existing video has been deleted or made private by the user. If a video link exists in the database which is not a part of the current scan (i.e., missing from the current scan), it can be concluded that the video is now unavailable (i.e. either deleted to made private). In order to keep a track of the unavailable videos, a database column is maintained indicating if a video has

been made unavailable. The script will terminate once all the unavailable video entries have been updated in the database.

Tables 3.1 & 3.2 show the detailed structure of the database tables.

Table 3.1. Database Table1 - ScanType

Sr. No.	Field Name	Field Details
1	id	Scan ID to identify the scan
2	scan_type	Type of scan performed for the user
3	timestamp	Time of the scan performed

Table 3.2. Database Table2 - Videos

Sr. No.	Field Name	Field Details
1	id	Unique video ID
2	timestamp	Stores the time of the video created
3	url	Video URL
4	video	Local path of the downloaded video
5	hash	Hash of the video downloaded
6	shares	Number of shares for the video
7	likes	Number of likes for the video
8	comments	Number of comments on the video
9	caption	Caption posted with the video
10	song	Song posted with the video
11	record_insert_scan_id	1st time the particular video has been scanned
12	last_updated_scan_id	Scan ID where video details have been updated
13	fields_updated	Fields updated for a repair/update scan
14	is_deleted	Boolean value to indicate unavailable videos

3.3 Tool Execution

In this section, the detailed steps for the execution of the tool will be discussed. It is important for the correct setup of the environment and installation of all the required libraries for the successful execution of the tool.

3.3.1 Downloading required softwares

For the execution of the tool, it is required that the user has the following software installed:

1. **Python**: The tool has been developed using Python version 3.10. In order to execute the script, Python version 3.x lesser than 3.11 must be installed. Python executable can be found at https://www.python.org/downloads/.

2. **Pip**: Pip is a package installer for Python. Thus, pip is required to install all the Python libraries. If pip is already installed, it is needed to upgrade pip to the latest version. Instructions to download and upgrade pip are available at https://pip.pypa.io/en/stable/installation/.

3. **Chrome Web Browser**: The chrome web browser is needed for loading the TikTok user profile. Chrome can be found at https://www.google.com/chrome/.

4. **DB Browser SQLite**: This tool is used to view the database file created post-script execution. It can be found at https://sqlitebrowser.org/dl/.

Table 3.3 shows the details of the software and device version details used.

3.3.2 Downloading required Python libraries

The Python libraries and dependencies required to execute the script need to be installed. It is advisable to set up a Python virtual environment to download all the dependencies and execute the script. The commands and instructions to create a virtual environment can be found at https://docs.python.org/3/library/venv.html#creating-virtual-environments.

Table **3.3.** Software Version Details

Sr. No	Software / Device	Version
1	Microsoft Surface Laptop 3	Windows 10 - 19044.2075
2	Chrome Web Browser	106.0.5249.62 (latest)
3	Python	3.10.7
4	DB Browser for SQLite	3.33.0

Command to create a virtual environment: **python3 -m venv /path/to/new/virtual/environment**

Once the virtual environment is created, it needs to be activated to be used. On the command prompt being used, be in the directory where the virtual environment is created. Instructions to activate the virtual environment can be found at https://python.land/virtual-environments/virtualenv.

Command to activate the virtual environment : **venv/Scripts/activate**

With the virtual environment now set up, the required Python libraries can be installed. The tool has an inbuilt **Requirements.txt** file which contains a list of all the libraries that need to be downloaded.

This file can be run with the command : **pip install -r Requirements.txt**

In order to download videos, Python playwright needs to be installed.

Command to install playwright : **python -m playwright install**

Table 3.4 lists the Python libraries used to develop the tool and their use in the tool.

3.3.3 Running the tool

With all the dependencies installed, the tool is ready to be run.

Command to run the tool: **Python Tiktok.py -u therock -p "C:/Program Files/Google/Chrome/Application/chrome.exe %s"**

Parameters :

Table 3.4. Python libraries used for tool execution

Sr. No.	Python Library	Usage in tool
1	argparse	Parse arguments passed to the script
2	sys	Exit the script execution
3	hashlib	Compute sha256 hash
4	pynput	Control keyboard operations
5	webbrowser	Load the profile on Chrome Browser
6	time	Add sleep while script execution
7	re	Match regular expressions
8	os	Perform system related operations
9	pyperclip	Copy/paste to and from clipboard
10	datetime	Get time stamp
11	pyautogui	Take screenshots
12	matplotlib	Image reading
13	scipy	Mathematical computations
14	numpy	Mathematical computations
15	shutil	Copy and delete files
16	BeautifulSoup	HTML parsing
17	requests	HTTP get request
18	TikTokApi	Download TikTok video
19	sqlite3	SQL operations
20	logging	Add logs related to execution

- **-u / −username**: It is a mandatory argument that indicates the user we want to get the videos for.

- **-p / Chrome path**: It is an optional parameter. If chrome is the default browser this does not need to be specified. Pass this parameter only if chrome is not set as the default parameter.

3.4 Choosing TikTok test data for tool validation

To validate the tool and its performance, a profile was created which is operated by the author. The videos are created with the following features to test:

1. Profile that has more than 30 videos to test if all video details are being fetched.

2. Videos that have added captions to test the captions being recorded correctly.

3. Comments added to videos to test the count of comments being recorded correctly.

4. Videos are created with well-known songs to test the song being recorded correctly.

5. Videos are shared by other users to test the share count being recorded correctly.

3.5 Pseudo code of the tool

Algorithm 1 shows the detailed algorithm used to develop the tool. The entire code can be viewed upon request at https://github.com/Akshata-Thole/tiktok-user-videos-scraper.

Algorithm 1 Tool Pseudo Algorithm

▷ Get username details from args
username, path ← read from args
set logging for terminal and file
initialize objects and variables being used

▷ Create a user directory
if user directory does not exist **then**
    Identify the user as a new user and create a directory for the user
else if user directory exists **then**
    choice ← 1. Delete data 2. Update all records 3. Update missing 4. Exit
    if choice is exit **then**
        Terminate execution
    end if
    if choice Delete existing data **then**
        Delete existing folder and database and create a new folder
    end if
end if

▷ Create a user database

if user database does not exist **then**

  Create database with scan_details and videos tables

end if

Insert scan record in scan_details to identify scan

▷ Open user profile on Chrome

if chrome path provided as arg **then**

  Use the given path to load the profile on chrome using webbrowser library

else

  Load profile on default web browser using webbrowser library

end if

▷ Load user profile on Chrome

while user profile not fully loaded **do**

  Scroll webpage using pynput library and down keypress

  if scroll_count = 500 **then**

    img ← Use the pyautogui library to take a screenshot

    if previous screenshot exists **then**

      Check if the two images are the same

      Use zero norm to compare images

      if images are same **then**

        Identify end of the scroll and terminate scrolling

      end if

    end if

  end if

end while

▷ Copy HTML body contents to the clipboard

Open inspection tab in chrome using pynput and chrome shortcut ctrl + shft + c

Copy HTML body using pynput and shortcut ctrl + c

▷ Send copied contents to the tool

file ← Use the pyperclip library to send clipboard contents to the tool

▷ Shutdown Chrome

Chrome is shut down using pynput and shortcut alt + f4

▷ Get the list of videos from HTML content

Pass HTML body content to beautiful soup object

video_list ← Filter video links from HTML content

---

if user is new **then**

    while next user video link exists **do**

        details ← Get video details by making HTTP Get request

        Fetch timestamp, likes, comments, share-count, caption, and song

        Download user video using the library TikTokApi

        hash ← Compute sha256 hash of the video

        Insert video details in a database

    end while

else if update all records **then**

    Execute DB query to check if video details exist in the database

    if video details do not exist in db **then**

        Insert new video details

    else

        details ← Get video details by making HTTP Get request

        Fetch timestamp, likes, comments, share-count, caption, and song

        Compare values with the existing values from the database

        if change in values observed **then**

            fields_updated ← Track changed values to be updated

        end if

        Check if the downloaded video exists

        if video not found **then**

            Download video

        end if

        Execute update query to insert updated values in the database

    end if

else if update missing values only **then**

    Execute DB query to check if video details exist in the database

    if video details do not exist in db **then**

        Insert new video details

    else

        Check for missing records in the database

        if missing further details **then**

            Get video details by making HTTP Get request

            Get missing values

            fields_updated ← Track missing values

        end if

        Check if the downloaded video exists

        if video not found **then**

            Download video

        end if

        Execute update query to insert missing values in the database

    end if

end if

$\triangleright$ Check for unavailable videos

if user is not identified as new user **then**
    old_videos ← Get existing video list from Database
    new_videos ← Get new videos list from the current scan
    unavail_videos ← old_videos - new_videos
    if unavailable videos found **then**
        Execute update query to mark videos as unavailable
    end if
end if

---

4. RESULTS

In this section, the results obtained by the execution of the tool will be analyzed. The details of accounts created and data collected using the tool will be validated and analyzed. Different aspects such as scroll identification, data validity, and testing on other platforms will be covered. It is important to understand and validate the tool results to make the data downloaded and provided by the tool valid and eligible evidence in a court of law.

4.1 Account details

A TikTok account with the username **samanthawatson745** was created to test the different features of the tool. A total of 35 videos were uploaded to the account. Other accounts were used to add likes and comments to the videos and share the videos. The videos uploaded are of different genres with different background songs. Some videos were added without captions and a known song to test if this is correctly identified. Fig: 4.1 shows the TikTok profile of user samanthawatson745.
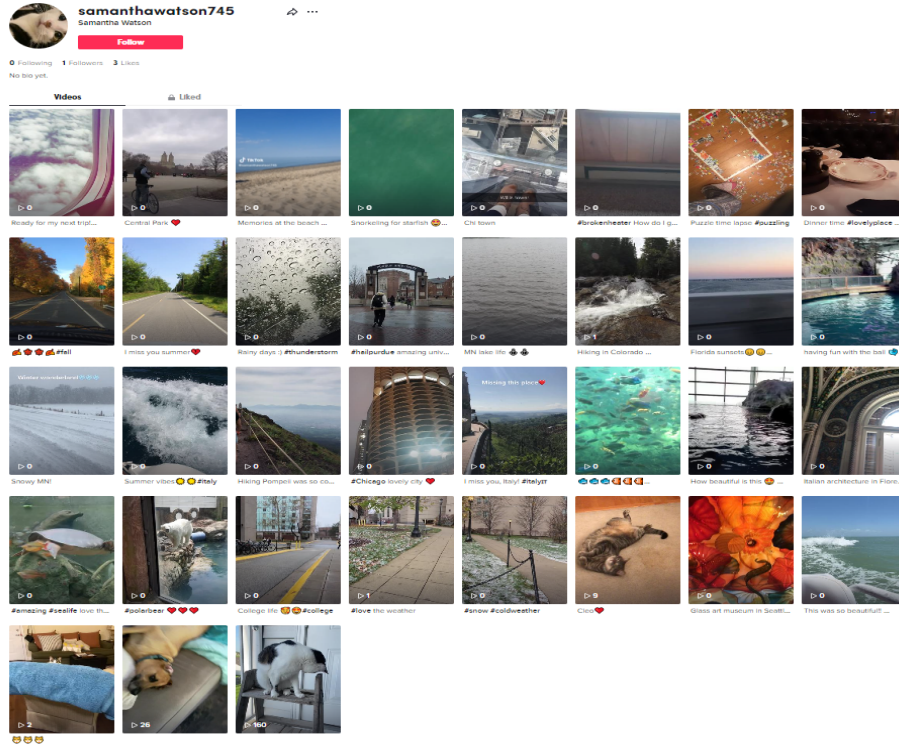

Figure 4.1. Screenshot of TikTok Profile of samanthawatson745

4.2 Analysis

In this section, the analysis of the tool will be discussed in detail. Various aspects will be covered such as correct identification of the end of the scroll, validation of the data downloaded, and tool success on other windows platforms other than windows 10.

4.2.1 Tool Execution

The tool was executed to fetch details of the TikTok account samanthawatson745 which was created for the purpose of this study. It was executed at multiple stages using the command provided in the Methodology chapter - python Tiktok.py -u samanthawatson745 -p "C:/Program Files/Google/Chrome/Application/chrome.exe %s". The purpose behind executing tools multiple times is to test the different features provided by the tool and validate if all the functionalities work as expected for the tool. Multiple runs are needed to validate features like deleted videos, changes in likes, comments, and shares, and if a new video is added after the initial execution of the tool. Fig. 4.2 shows the database details for the scans performed for the user.

id	scan_type	timestamp
Filter	Filter	Filter
1	First Scan	2022-11-10 23:31:38.885463
2	Repair All	2022-11-16 18:53:42.863157
3	Repair Missing	2022-11-16 19:05:37.171815
4	Repair Missing	2022-11-16 19:09:07.176007
5	Repair All	2022-11-16 19:21:25.735070
6	Repair All	2022-11-16 22:44:55.583516
7	Repair All	2022-11-16 23:00:32.965740

Figure 4.2. Tool Execution - Scan details of scans executed for samanthawatson745

4.2.2 End of scroll identification

It was noted that the page scrolling depended on the internet speed and the machine being used. For a good and high internet speed or a machine with high specifications, the videos loaded quickly and scroll correctly identifying the end of profile loading. For a slower

internet speed or machine with lower specifications, the videos loaded slowly and if newer videos did not load between two subsequent screenshots, the scroll ends without loading all the videos. To correctly load all the videos, the number of key-press after which a screenshot is taken needs to be adjusted according to the internet speed and machine specifications.

4.2.3 Timestamp analysis

The timestamp details for a TikTok video are stored at two different locations and both can be used to correctly identify the accurate timestamp of the video being posted. The two techniques that can be used are:

- Using video ID: The video ID for TikTok is a 64-bit snowflake ID which consists of the timestamp plus a random value added at the end. The timestamp and the random value are of equal length and if we remove the random value from the ID, we get the exact timestamp at which the video was uploaded to TikTok. This method of creating timestamps was used only after 2017 (2 years of TikTok in use) and for videos prior to 2018, this method fails to give the accurate timestamp of the video being posted.

- Parsing JSON response: The response of the HTTP GET request for a particular video ID contains a JSON object which can be fetched with the key: "createTime:". This field is correctly populated for all TikTok videos and can be used to fetch the exact timestamp of the video being uploaded to TikTok. The timestamp is UNIX time which needs to be converted to human time.

Since the JSON object gives us accurate results in all cases, this method is chosen to fetch the accurate timestamp. Once the timestamp is fetched, this field does not change the value as the upload time for a video cannot change.

4.2.4 Fetching video details

In order to make bots ineffective, TikTok randomly returns a blank response to an HTTP request. Due to this, in the initial run for a user, multiple null entries are seen. To get all the possible details, the tool needs to be executed multiple times for the user. Fields like

timestamp, caption, and song cannot be updated once inserted whereas fields likes, shares, and comments are changeable and can be updated in subsequent scans.

4.2.5 Video download and hash computation

It is observed that arbitrarily the video files downloaded are empty and need to be re-downloaded. This can happen if the TikTok server fails to provide the location of the file required to download similar to receiving an empty response to an HTTP request. Identification of empty files is out of the scope of this study and needs to be identified manually. The tool relies on the physical location of the video to check if the video is downloaded or not. If the video is deleted from the device, the next run for the user will identify the video as missing and attempt to download the video.

4.2.6 Tracking change in fields

The database columns last_updated_scan_id and fields_updated track the updates in the video posted by the user. Fields timestamp, caption, and song are constant and cannot be updated once retrieved. Fields like the number of likes, shares, and comments are varying and are bound to change. The column fields_updated keeps a track of the field_name and the scan_id in which it was last updated and the column last_updated_scan_id keeps a track of the latest scan_id in which any field was updated. However, as TikTok arbitrarily returns an empty response for HTTP requests, fields other than likes, shares, and comments can also be seen in updated fields. Fig. 4.3 shows the database entry that shows the fields last_updated_scan_id and fields_updated.

record_insert_scan_id	last_updated_scan_id	fields_updated ▲1
Filter	Filter	Filter
2	7	{"video": 4, "hash": 4, "likes": 7}
2	6	{"shares": 6}
2	7	{"likes": 7}

Figure 4.3. Tool Execution - Track updates fields in videos table for samanthawatson745

4.2.7 Testing for video count less than 30

The tool is aimed at loading the entire profile of a user and providing the details of all the videos posted by a user. To do so, the tool uses image processing to identify the end of the scroll and then fetch the HTML contents of the page loaded. However, if a user has very few videos that do not require scrolling, the tool must identify no need to scroll and return the HTML contents of the page. To test this, the tool was executed with only two videos posted for the user samanthawatson745 as this did not need any scrolling. The tool successfully provided the results for the two videos posted by the user. Fig. 4.4 shows the database details for the two videos posted by the user.

id	timestamp	url	likes	comments	shares	caption	song	video	hash	record_insert_scan_id	last_updated_scan_id	fields_updated	i
Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	F
1 7148514253505334571	2022-09-28 15:26:30	https://...	0	1		Share	original sound ...	samanthawatson745\video...	5d8b01b7ae181aa7cbf01ca8c6...	1	1 NULL		0
2 7148894534003428654	2022-09-29 16:02:12	https://...	0	2		Share	original sound ...	samanthawatson745\video...	706584bae9701109d99f0e486...	1	1 NULL		0

Figure 4.4. Tool Execution - Less than 30 videos posted for samanthawatson745

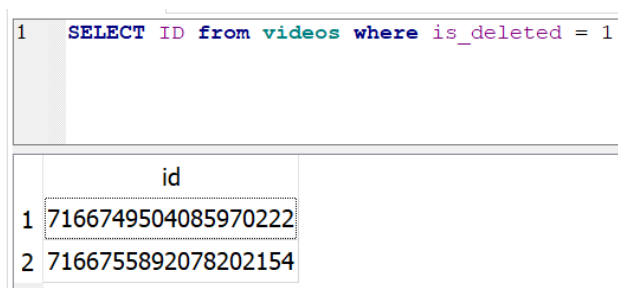### 4.2.8 Testing for video count greater than 30

By default, without human verification, TikTok loads a maximum of 30 videos for a user. The main aim of this study is to bypass this check and load the entire profile of the user aiming to fetch details of all the videos posted by the user. With 35 videos uploaded to the profile, the tool was executed for samanthawatson745 to validate all 35 video details being downloaded accurately. The tool successfully identified all the video links and fetched the details of all the videos posted by the user. Fig. 4.5 shows the result of a database query executed to count the number of entries in the videos table for the user.

```
1   SELECT count (ID) from videos
```

	count (ID)
1	35

Figure 4.5. Tool Execution - Count of all videos retrieved for user samanthawatson745

42

4.2.9 Testing for unavailable video

The tool identifies unavailable videos only after the initial scan is complete. The videos posted by the user during the first scan are considered the base for all subsequent scans. The tool has no knowledge of videos made unavailable prior to the initial scan. After the first scan, the list of initially posted videos is available. If a video does not appear in the subsequent scans, it is identified and marked as unavailable. The tool also fails to identify a video that has been posted and made unavailable in between subsequent scans. Hence, unless a video's details are entered in a prior scan, the tool cannot identify any video as being made unavailable by the user. To validate unavailable videos, two of the videos posted by the user samanthawatson745 were deleted and then the tool was executed again. The tool successfully detected the two videos as made unavailable by the user. Fig. 4.6 shows the result of a database query executed to count the number of videos identified as unavailable.

Figure 4.6. Tool Execution - Count of unavailable videos for user samanthawatson745

4.2.10 Testing on different platforms

The tool was run on the windows 11 platform to test and validate the results on the latest windows version available during the course of the study. The tool successfully executed and provided accurate results on windows 11 as well as compared to windows 10. To validate the tool performance of different screen resolutions, it was executed on larger monitors available in the university lab KNOY 374. The tool is unstable with execution on larger screens and at times fails to scroll correctly or copy HTML contents accurately. The reason behind this is unknown. Hence, the tool works correctly for laptops and small PCs only.

4.2.11 Miscellaneous Findings

Most social media sites need the user to login before being able to use it. However, this tool successfully gets all the video details without the need to login into TikTok. It is recommended that the tool is run using the incognito mode of Chrome web browser as with all the security features provided by TikTok, it can identify potential bot activity for this tool. To fetch all the videos, the cache needs to be cleared every time, prior to the execution of the tool. To bypass this, it is recommended to execute the tool using incognito mode which does not store cookies. This can be done by passing the chrome path with the incognito mode parameter set.

To correctly identify the end of scrolling activity, it is recommended to open the chrome web browser in full-screen mode, especially on large screen resolutions. Although this is not a requirement of the execution, this made a difference in the accurate execution of the script. Without chrome being in full screen, the tool failed to identify the correct end of scroll. The reason behind this is not known.

5. DISCUSSION

With the results obtained by the tool and analyzing it, it can be concluded that it is possible to automate the end-to-end flow to forensically download all the user videos on TikTok. We accept the hypothesis $H_{0,1}$ defined in the introduction. With the video links now available, and getting a response to the HTTP response for them, it was possible to fetch all the video details and relevant metadata. So, we accept the hypothesis $H_{0,2}$. Computing the sha256 hash of the video file downloaded is a check on the integrity of the video making it a piece of valid evidence in court. Hence, we accept the hypothesis $H_{0,3}$. Finally, using the baseline scans deleted videos were also successfully identified provided the video has been identified in previous scans. So, we accept the hypothesis $H_{0,4}$.

The results obtained by this study successfully validate a tool that can automate the download of all the videos posted by a user on TikTok. Analyzing the TikTok profile of a user is important from an OSINT perspective as the profile and videos potentially indicate the user's likes, interests, actions, behavior, and much more. All this information can become very strong evidence in a court of law. Thus, it is important to download valid and authentic data which can be obtained from a user profile. Along with the user, the TikTok profile of their close people with whom they have featured in videos is also important to completely analyze the social media of the user.

Timestamp plays an important factor to understand when the user posted the video. It can be inferred that the video was shot either on the day it was posted or sometime prior to that. For example, if an underage person posts a video of drinking on TikTok but claims that he never drank, using this timestamp it can be concluded that the person indeed lied and drank, being under-aged. This becomes important evidence to prove a claim to be wrong and can be used in court. The timestamp is not uploaded by the user and is generated by the TikTok server. This makes it difficult for any malicious user to manipulate the timestamp of the video once it is created and the video is uploaded to the TikTok server. So, the timestamp value fetched from the HTTP response can be trusted to be authentic. This makes it a piece of valid evidence in court.

Analysing the caption a user writes along with posting the video summarizes the potential content of the video. Users add hashtags to push their videos to the trending section. If a user posts an inappropriate caption promoting violence, danger, or sexual content, it is misusing the app and violating its terms and conditions of use. Hence, it is essential to store and analyze the caption posted by a user to understand any misuse or promotion of any wrong activity. Once the caption is posted, a user is unable to modify or alter the caption. So, once the caption is captured by the tool, it can be analyzed and trusted to be authentic and unaltered post-download.

Similar to the caption, the user also uses a song or his own sound as music to create and post a video. TikTok has trending songs and users generally tend to use these songs to create their videos. However, this can be misused to promote any wrong or illegal activities too. Hence, analysis of the song also plays an important role in the identification of any misuse or promotion of inappropriate content. The song also cannot be changed or altered once the video is posted. Once the tool parses the content to fetch the song, it can be used for further analysis.

Fields likes, shares, and comments are useful to interpret the popularity of the video and understand if the video is trending. Likes and shares are generally activities of other TikTok users on the video. Comments are a mix of activities of the user and other TikTok users. Users post their opinion about the video and in return, the user acknowledges their opinion on their video. Comments can turn out to be harsh, aggressive, sexual, etc. too and it is important to analyze the comments posted on the videos. Downloading and storing all the comments on the post is beyond the scope of this study. Fields likes, shares and comments keep changing according to user activities.

Computing the hash of the video downloaded is important to keep the integrity of the video intact. Since TikTok does not provide the hash of the video in the JSON response or there is no known trusted 3rd party to validate the video integrity, we trust the video downloaded by the tool to be correct. We compute the hash only after the video download and can use this for further validation purposes. There are well-known hashing techniques like md5 and sha256 that can be used to check the integrity of the video. The key feature that decides the strength of a hash function is collision resistance. Lesser the collisions of the

hash function, the stronger it is. md5 is known to have attacks against collision resistance and can be broken using brute force methods. sha256, on the other hand, does not have any known attack and is considered to be secure at the time of the study. Hence, the sha256 hash of the video is calculated to keep a check on the integrity of the video. By re-calculating the hash of the file downloaded, the hash in the database and the new hash can be compared. If they are the same then we can conclude that the file has not been tampered with and can be used as a piece of valid evidence in court.

The fields captured by the scan prove to be important to analyze the user profile from the OSINT perspective. The method used to store and capture these fields is authentic with the data integrity maintained and can be used for validation at any later stage.

6. CONCLUSION

The aim of this study was to propose a proof-of-concept tool that can automate the end-to-end flow to fetch the video details of a particular TikTok user, download the videos, and store all the data of interest. The methodology used to design the tool has not been used before to the best of the author's knowledge and can be used to perform web scraping of other social media platforms as well to gain vital information. In recent years, social media scraping has become an important aspect to analyse a person's behavior and get proof of potential inappropriate activities too. Hence, it has started to gain a lot of popularity in recent times.

Due to various security features of TikTok, methodologies used in research done on other social media platforms like Facebook and Twitter could not be applied to TikTok and needed a new and novel technique to get all the videos posted by a user. TikTok being a popular social media platform has been a topic of interest for a lot of researchers. Various studies have been discussed in detail in the literature review section such as the analysis of trending videos. However, there is a significant lack of research done to fetch user video details which was the main motivation behind the study.

The data obtained using the tool is trusted to be preserved with integrity and can be used by law enforcement and criminal investigators for validation and any other use at any later stage. The tool can be effective to scrape TikTok and reduce the manual efforts needed to analyze a profile and provide quicker results.

7. FUTURE WORK

The focus of this study was on creating a back-end service to download and store the details of videos posted by a user on TikTok. The next step to make the results more usable and accessible to the user is to develop a front-end system. Through this, a user will be able to visualize the details in the database and perform actions like searching for keywords to identify any inappropriate activities. Additional functionalities like running a scan, and scheduling a scan can also be added to provide a user-friendly experience.

The tool is focused on TikTok only. It can be further extended to other social media platforms and create a cross-platform consolidated result for a user to provide a detailed overview of the activities of a user across multiple platforms. This will greatly help criminal investigators to analyze user profiles from an OSINT perspective. Consolidated results from one tool will greatly save their time and efforts to perform multiple scans for multiple platforms.

The current study does not focus on additional activities of the user on TikTok like posting comments, liking other users' videos, and sharing their videos. These activities are also important to note and can prove to be of use in any investigation. Hence, increasing the scope and depth of analysis for a user can add value to the results obtained by the execution of this tool.

REFERENCES

[1]   *Social media scraping: Legality, how-to, and advantages.* [Online]. Available: https://research.aimultiple.com/social-media-scraping/.

[2]   *Tiktok.* [Online]. Available: https://www.tiktok.com/en/.

[3]   *Open-source intelligence*, Sep. 2022. [Online]. Available: https://en.wikipedia.org/wiki/Open-source_intelligence.

[4]   *Selenium with python.* [Online]. Available: https://selenium-python.readthedocs.io/.

[5]   D. Teather, *TikTokAPI*, version 5.2.2, Jul. 2022. [Online]. Available: https://github.com/davidteather/tiktok-api.

[6]   N. Hoang Khoa, P. The Duy, H. Do Hoang, D. Thi Thu Hien, and V.-H. Pham, "Forensic analysis of tiktok application to seek digital artifacts on android smart-phone," in *2020 RIVF International Conference on Computing and Communication Technologies (RIVF)*, 2020, pp. 1–5. DOI: 10.1109/RIVF48685.2020.9140739.

[7]   D. Klug, Y. Qin, M. Evans, and G. Kaufman, "Trick and please. a mixed-method study on user assumptions about the tiktok algorithm," in *13th ACM Web Science Conference 2021*, ser. WebSci '21, New York, NY, USA: Association for Computing Machinery, 2021, pp. 84–92, ISBN: 9781450383301. DOI: 10.1145/3447535.3462512. [Online]. Available: https://doi.org/10.1145/3447535.3462512.

[8]   B. K. Jena, *What is sha-256 algorithm: How it works and applications [2022 edition]: Simplilearn*, Nov. 2022. [Online]. Available: https://www.simplilearn.com/tutorials/cyber-security-tutorial/sha-256-algorithm.

[9]   *What is web scraping? - definition from techopedia. techopedia.com.* [Online]. Available: https://www.techopedia.com/definition/5212/web-scraping.

[10]   *Brief history of web scraping*, May 2021. [Online]. Available: https://webscraper.io/blog/brief-history-of-web-scraping.

[11]   X.-B. Enterprise, *The history and rise of web scraping last ten years.* [Online]. Available: https://www.xbyte.io/the-history-of-web-scraping.php.

[12]   *Tiktok*, Sep. 2022. [Online]. Available: https://en.wikipedia.org/wiki/TikTok.

[13]   -, *How tiktok is addictive*, Sep. 2020. [Online]. Available: https://medium.com/dataseries/how-tiktok-is-addictive-1e53dec10867.

[14]   A. E. Dastagir, *Children are dying in the tiktok 'blackout' challenge. how social media is changing peer pressure.* Jul. 2021. [Online]. Available: https://www.usatoday.com/story/life/health-wellness/2021/04/27/tiktok-challenge-kills-12-year-old-how-peer-pressure-has-evolved/4853507001/.

[15]   *Requests.* [Online]. Available: https://pypi.org/project/requests/.

[16] A. Himawan, A. Priadana, and A. Murdiyanto, "Implementation of web scraping to build a web-based instagram account data downloader application," *IJID (International Journal on Informatics for Development)*, vol. 9, no. 2, pp. 59–65, Dec. 2020. DOI: 10.14421/ijid.2020.09201. [Online]. Available: https://ejournal.uin-suka.ac.id/saintek/ijid/article/view/09201.

[17] *Beautiful soup documentation.* [Online]. Available: https://beautiful-soup-4.readthedocs.io/en/latest/.

[18] L. C. Dewi, Meiliana, and A. Chandra, "Social media web scraping using social media developers api and regex," *Procedia Computer Science*, vol. 157, pp. 444–449, 2019, The 4th International Conference on Computer Science and Computational Intelligence (ICCSCI 2019) : Enabling Collaboration to Escalate Impact of Research Results for Society, ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2019.08.237. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050919311561.

[19] A. Hernandez-Suarez, G. Sanchez-Perez, K. Toscano-Medina, V. Martinez-Hernandez, V. Sanchez, and H. Perez-Meana, *A web scraping methodology for bypassing twitter api restrictions*, 2018. DOI: 10.48550/ARXIV.1803.09875. [Online]. Available: https://arxiv.org/abs/1803.09875.

[20] B. Basumatary and H. K. Kalita, "Social media forensics - a holistic review," in *2022 9th International Conference on Computing for Sustainable Global Development (INDIACom)*, 2022, pp. 590–597. DOI: 10.23919/INDIACom54597.2022.9763129.

[21] P. Domingues, R. Nogueira, J. Francisco, and M. Frade, "Post-mortem digital forensic artifacts of tiktok android app," Aug. 2020, pp. 1–8. DOI: 10.1145/3407023.3409203.

[22] N. Hoang Khoa, P. The Duy, H. Do Hoang, D. Thi Thu Hien, and V.-H. Pham, "Forensic analysis of tiktok application to seek digital artifacts on android smartphone," in *2020 RIVF International Conference on Computing and Communication Technologies (RIVF)*, 2020, pp. 1–5. DOI: 10.1109/RIVF48685.2020.9140739.

[23] Drawrowfly, *Drawrowfly/tiktok-scraper: Tiktok scraper. download video posts, collect user/trend/hashtag/music feed metadata, sign url and etc..* [Online]. Available: https://github.com/drawrowfly/tiktok-scraper.

[24] A. Shutsko, "User-generated short video content in social media. a case study of tiktok," in *Social Computing and Social Media. Participation, User Experience, Consumer Experience, and Applications of Social Computing*, G. Meiselwitz, Ed., Cham: Springer International Publishing, 2020, pp. 108–125, ISBN: 978-3-030-49576-3.

[25] A. Neyaz, A. Kumar, S. Krishnan, J. Placker, and Q. Liu, "Security, privacy and steganographic analysis of faceapp and tiktok," Jun. 2020.

[26] avilash, *avilash/TikTok API Python Wrapper.* [Online]. Available: https://github.com/avilash/TikTokAPI-Python.

[27] TufayelLUS, *Tufayellus/tiktok-video-downloader-PHP: A simple but effective one page TikTok video downloader(Watermark Free)*. [Online]. Available: https://github.com/TufayelLUS/TikTok-Video-Downloader-PHP.

[28] Naseif, *NASEIF/tiktok-scraper*. [Online]. Available: https://github.com/naseif/tiktok-scraper/tree/d01bb3397f55c53c44644f10b87f0b40914c3e48.

[29] Alasad, Usama, and Mike, *4k tokkit: Tiktok videos, hashtags and accounts downloader*, Sep. 2022. [Online]. Available: https://www.4kdownload.com/products/tokkit/17.

[30] *Argparse - parser for command-line options, arguments and sub-commands*. [Online]. Available: https://docs.python.org/3/library/argparse.html.

[31] *Os - miscellaneous operating system interfaces*. [Online]. Available: https://docs.python.org/3/library/os.html.

[32] *Shutil - high-level file operations*. [Online]. Available: https://docs.python.org/3/library/shutil.html.

[33] *Webbrowser - convenient web-browser controller*. [Online]. Available: https://docs.python.org/3/library/webbrowser.html.

[34] *Pynput*. [Online]. Available: https://pypi.org/project/pynput/.

[35] Simplilearn, *What is image processing? meaning, techniques, segmentation and important facts to know*, Jul. 2022. [Online]. Available: https://www.simplilearn.com/image-processing-article.

[36] *Welcome to pyautogui's documentation!* [Online]. Available: https://pyautogui.readthedocs.io/en/latest/.

[37] *L0 norm in machine learning*. [Online]. Available: https://livebook.manning.com/concept/machine-learning/l0-norm.

[38] [Online]. Available: https://numpy.org/.

[39] *Visualization with python*. [Online]. Available: https://matplotlib.org/.

[40] *Scipy*. [Online]. Available: https://scipy.org/.

[41] *Pyperclip*. [Online]. Available: https://pypi.org/project/pyperclip/.

[42] *Re - regular expression operations*. [Online]. Available: https://docs.python.org/3/library/re.html.

[43] *Sqlite3 - db-api 2.0 interface for sqlite databases*. [Online]. Available: https://docs.python.org/3/library/sqlite3.html.

[44] *15.1. hashlib - secure hashes and message digests*. [Online]. Available: https://docs.python.org/3.5/library/hashlib.html.