Proactive Acquisition using Bot on Discord

Niken Dwi Wahyu Cahyani[1], Daffa Syifa Pratama[2], Nurul Hidayah Ab Rahman[3]
Informatics Faculty, Telkom University, Bandung, Indonesia[1, 2]
Centre of Information Security Research-Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Parit Raja, Malaysia[3]

***Abstract*—Data deletion increases challenges in cybercrime investigation. To address the problem, proactive forensics for evidentiary collection is acknowledged to help investigators to acquire the potentially needed digital evidence. This study proposes a bot machine to record data from the Discord server in advance, hashing and saving it in proper storage for further forensic analysis. The recording process can be managed to collect activities and their related data (intact, modified, deleted), including text, pictures, videos, and audio. The Discord bot is designed by utilizing the main features of the Discord Social Networks Application Programming Interface (API). This paper examines how this approach is applicable by embedding the bot in a Discord server. Observation showed that the bot records the real-time data as it is always alive on the server, including the deleted or modified messages and their timestamps. All the recorded data is saved locally on the server's storage in easy-to-read formats, CSV and JSON. The results showed that the bot could conduct the data acquisition for 37 concurrent users with a 2.3% error rate and 97.7% accuracy.**

Keywords—Discord; bot; cybercrime; social networks API; digital evidence

I. Introduction

One of the challenges digital forensics investigators face is the inability of forensic tools to acquire deleted data [1]. The challenge makes investigative activities require more time and effort to find data remnants from other sources that can provide more evidence. In addition, from an information security perspective, there is a need to guarantee data deletion [2].

Proactive forensics supports investigators in collecting data before an incident occurs [3]. This approach will collect live data, thus enabling all data and changes that occur to the data to be collected [4]. For example, keystroke logging methods have been proposed to logically acquire keystrokes in cloud applications for forensic readiness [5]. The method shows how forensic activities can be assisted through data collection techniques initially considered malicious acts.

It is understood that many bots are known for their harmful impact. Even several guides for conducting bot crime investigations are available [6]–[8], and IoT-Botnet datasets have also been developed for network forensic analytics [9]. However, no known study in this area examines the benefits of bots for acquiring potential evidence data. This gap is identified by a previous study that stated some important future works in instant messaging forensic investigation, including users editing/deleting messages and Discord bots [10]. Therefore, this current research examines proactive forensics using bots by taking the example of the Discord application.

The Discord data is collected in advance to reduce the time and complexity of investigations to obtain more complete data. The proposed bot is tested for positive reasons and measured how far this acquisition method could help. Understanding its advantages and identifying its issues are essential to guide investigators' practice and academics in developing the proper bot system.

Discord enables users to indirectly publish substantial amounts of information, including voice calls, video calls, text messages, media, and file sharing. From cybercriminals' perspective, these features can be exploited to conduct cybercrimes. However, the offender could modify evidence of interest by deleting, editing, and clearing messages on the cache. The action is anti-forensics - data concealment by implementing data hiding and trail obfuscation techniques to remain anonymous or undetected [11], [12]. In addition to anti-forensic issues, it could lead to incomplete attributes of collected artifacts and affect the integrity of digital evidence [13].

Therefore, it is necessary to have an acquisition method that can acquire the attributes of digital evidence completely and does not affect its integrity. An example of the potential method is by using the Social Networks API. The API is used in the acquisition process by focusing on features that the Discord application API has fully provided, and it can be modified and adjusted according to acquisition needs [14]. These characteristics of the APIs can complement the lack of attributes from collecting digital evidence. In this study, a Discord Bot is developed based on the features of the Social Networks API's Discord to facilitate digital evidence acquisition. A test scenario of the Social Networks API based on Discord Bots was set up to simulate digital evidence collection from the Social Networks API method. Another key point is to improve the performance of the acquisition process and extract complete digital evidence.

The rest of this paper is organized as follows: Section II outlines the related work of proactive forensics, and Discord acquisition. Section III explains the bot design. Section IV presents the results and discussion. Section V contains the conclusion and suggestions for future work.

II. Related Works

Proactive forensics involves collecting data before or during the incident by promoting the automation of live investigation [3], [15]. Applying proactive forensics in incident response teams would equip the team to respond appropriately to an incident. At the same time, for investigators, advanced data collection would help speed up the investigation process because the data is already available. This proactive approach is essential in digital investigation, especially in environments like social networks where editing and deleting messages are common for users to remove their unwanted traces; also, in critical systems such as Industrial Control Systems where real-time analysis may enable rapid triaging and response to attack [16], [17]. Meanwhile, the live data collection approach can be conducted by recording data from the running activities. For example, previous authors proposed a method of keystroke logging to acquire keystrokes in cloud applications for forensic readiness [5] and installing software on the target system to preserve deleted files that might interest forensic examiners [3].

Motivated by the best communication feature for gamers, Anderson ran a qualitative ethnography study to understand if Discord helps enforce the values and behaviors of the Harbormen gaming community [18]. This study revealed that the ease of use and the convenience of interacting with different communities on various servers are experienced by its users. With this positive experience, the gamer's community and other fields, such as education, utilize Discord. It was used as a platform for physical education learning during the COVID-19 pandemic in a high school [19]. Some higher education institutions have innovatively used Discord to deliver teaching listening as an e-learning tool in the sciences and humanities [20].

Instant messenger provides communication via text, voice, videos, and photos. However, there is an increasing trend of cyber criminals who use instant messaging applications to do malicious acts. The ease of their registration and usage attracts many users, including criminals. The vast amount of user data inside the apps becomes potential evidence in digital forensic investigations. As different apps manage their data differently, it is essential to conduct application-specific forensics.

A previous study analyzed the Google Chrome cache structure inherited from Discord [21]. The study showed it could successfully get Discord-related metadata through the cache on Discord apps for the PC version. Nevertheless, further work still needs to be conducted to cover evidence from Discord Web Applications and Discord Mobile Applications. Another study examined Discord desktop applications on Windows 10 from a forensic value and cybersecurity perspective [22]. Similarly, the study demonstrated that Discord metadata could be successfully acquired through the cache on Discord Applications. A recent study on Discord forensics based on data from the Google Chrome browser also recovered various artifacts [23]. However, while much important information can be acquired through cached data, an issue of its deletion may prevent the acquisition.

Research on Linux OS computers found Discord-specific data, including messages, usernames, and passwords [10]. Examination of the broader platform by including the Discord mobile app identified locations of artifacts, such as received/sent messages, shared files, chat rooms, and user account information [24]. Conducting forensic analysis on client-based devices can successfully acquire interesting data remnant, but the analysis should be done individually for each device.

As Discord provides services on instant messaging and VOIP, there has been a significant interest in examining the forensic analysis of other similar tools. A study conducted a forensic survey and analysis of Tango VoIP for iOS and Android platforms [25]. A research environment was set up for different mobile devices by installing WhatsApp, Skype, Viber, and Tango, and a list of target artifacts was defined. In addition to the forensic analysis, this study investigated how cloning IM sessions and intercepting communications can facilitate data acquisition. It was observed that encrypted data presents challenges to the acquisition process.

Research on WhatsApp discussed forensic approaches to creating real-time insights into WhatsApp communications [26]. This approach uses eavesdropping, decrypting WhatsApp databases, open-source information, and analyzing WhatsApp web communications. The research evaluated the method in various WhatsApp forensic scenarios to prove its feasibility and efficiency. It was found that various data, including profile pictures, user activities, location data, remote access to suspicious WhatsApp accounts, voice messages, shared contacts, documents, images, and videos, are accessible.

Tools are needed for application-specific forensics analysis to acquire and analyze the data. A study has presented a brief overview and a comparative analysis of various commercial and open-source mobile forensic tools [27]. The review used a cross-device and test-driven approach to predefined software parameters. Test scenarios addressed digital threats and assessed whether the tool has the expected functionality. The parameters used to compare are cost, MD5 hash mechanism, ease of use, and platform support.

To the extent of our review, existing studies on tools to support Discord forensic analysis depend on the data remnant of the apps, both on the client and server sides. There is no study on proactive evidence acquisition by recording all user activities at once using a bot. In this study, the usage of a bot installed on the Discord server is proposed to obtain all the exchanged messages, including text and multimedia data, for the intact and the deleted data.

A. Discord

Discord is a social networking app used by people (over 13) to discuss many topics, including games. It hosts communities of all sizes but is primarily used by small, active groups that interact regularly. It hosts communities of all sizes but is primarily used by small, active groups that interact regularly. Therefore, Discord comprises artifacts that contain vital information such as text, attachment files, and member lists in one event. Unlike other popular social network apps, no algorithm is involved in deciding what to watch, infinite scrolling, and no news feeds on Discord. Table I presents the commonly used Discord features by users.

B. Social Network API

Application Programming Interface (API or WEB API) is a module that enables interaction with application service resources. Examples of resources owned by application services are documents, images, and text messages [28]. Therefore, there is a potential to utilize Social Networks API to collect evidence artifacts from social network apps. Utilization is possible because the generated resources comprise metadata that describes the corresponding data. Furthermore, Social Networks API allows us to adjust the code according to the acquisition needs.

TABLE I. DISCORD FEATURES

Function	Description
Text channels	Feature for users to send messages to each other
Voice channels	Feature for users to communicate with each other
Share screen	Feature for users to share videos live with other users
Sharing images	Feature for user sharing images
Text channels	Feature for users to send messages to each other
Upload files	Feature for users to share documents

The Discord API provides another user account dedicated to automation called a bot account. Anyone can create a bot account from the app page and authenticate with a token (i.e., without a username and password). Unlike the regular Open Authorization (OAuth2) API, a bot account has full access to all API routes and can connect to a real-time gateway without an authentication token.

III. RESEARCH METHOD

Discord provides various functions via API. The bot uses the official Web API Discord to acquire the data stored on the Discord cloud server.

A. API Usage

The flowchart in Fig. 1 presents the two steps of using Discord API. Firstly, Discord API records every message and returns the log or cache in the bot. Before the log or cache returns, Discord Bot will check any edited or deleted message/data. Secondly, Discord API is used to request features and give responses. After these two main usages of the API, the bot performs read or write disk operations to save the messages in CSV or JSON format.

B. Discord Bot Design

Discord bot is an application created by the user with the admin group role. For example, user A as the admin, creates a discord server, and user B join the discord server created by user A. After that, user B sent message on the text channel. The message from user B will record by the discord bot and reproduce the cache. Discord bot is designed to hook websites as a function to get message channel history which can list return message attributes, including the deleted and edited messages. The Discord bot will save all record messages on private channel that can be seen by user A as the group admin. The use case of the designed Discord bot is presented in Fig. 2.

Before starting the Discord bot, the admin must create a guild and text chat channel. Next, the admin can create a bot, enter it into the created guild or channel, and share the guild. Users could then join the guild and interact with other users by sending messages, pictures, documents, videos, and audio. Subsequently, these activities are recorded, and their data will be acquired and saved in the Discord bot. This flow is presented in Fig. 3.

C. System Requirement

The bot is built using Python and utilizes the currently available Discord API. The details of our bot specification are presented in Table II.

To acquire Discord data from the server, the required functions are implemented on the bot to get the messages, data attachment, timestamp, and message id, calculate data hash value, and prepare the data saving on local storage. Table III lists these functions embedded in the Discord Bot.

D. Testing Scenarios

During the testing, 37 users accessed the Bot server simultaneously. The users simulated the common activities as follows:

- Sending:
  - text messages.
  - document files.
  - audio files.
  - video files.
  - picture files.
- Deleting the first sent text, document, audio, video, and picture files.

Comparison between the actual sent and deleted data and the successfully acquired data are examined to measure the bot's performance.

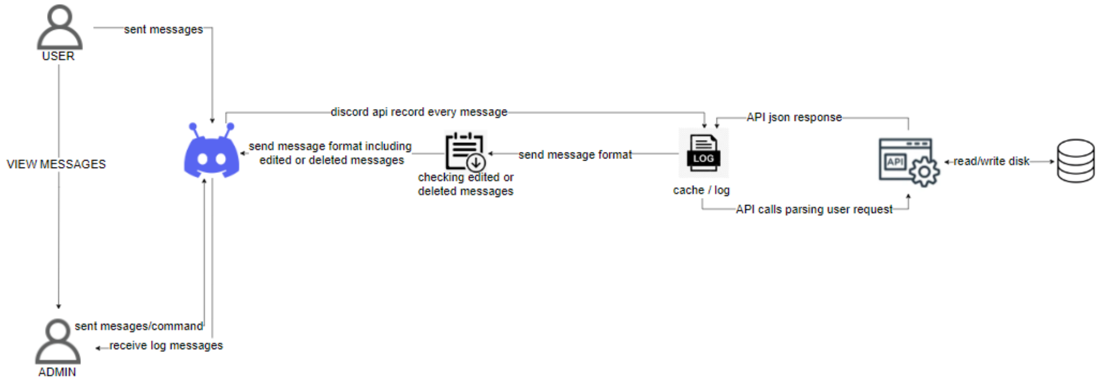


Fig. 1. The flow of API usage.

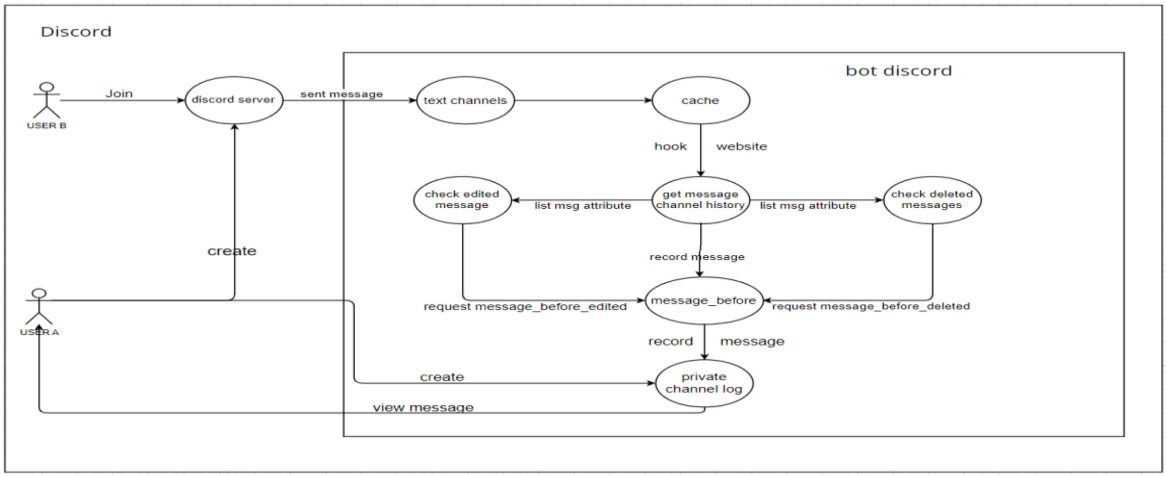


Fig. 2. The use case of the discord bot.

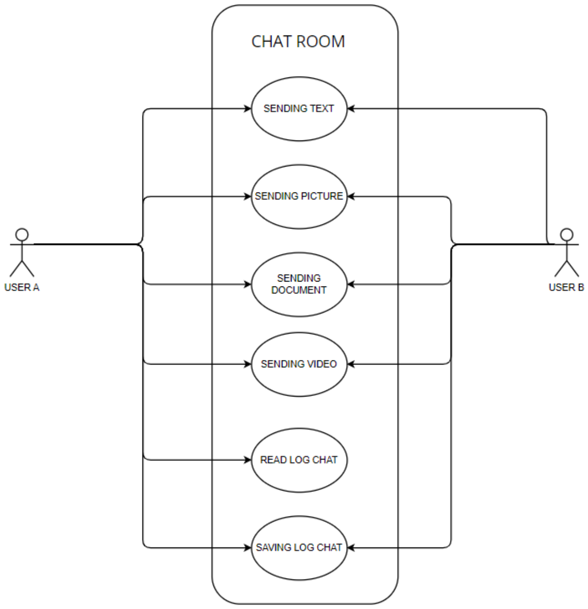


Fig. 3. User interaction in the bot.

TABLE II. THE BOT SPECIFICATION

<table>
<tr><th colspan="3">Discord Bot Application</th></tr>
<tr><th>No</th><th>Specification</th><th>Detail</th></tr>
<tr><td>1</td><td>Name</td><td>DEVICO BOT</td></tr>
<tr><td>2</td><td>Version Name</td><td>1.0</td></tr>
<tr><td>3</td><td>Application ID</td><td>936165836010426369</td></tr>
<tr><td>4</td><td>API Version</td><td>9.0</td></tr>
<tr><td>5</td><td>Token</td><td>OTM2MTY1ODM2MDEwNDI2MzY5.GMbjAH.SwcAiKb_s_1-kWLt53TQnz9KCUqxxxxxxxxxxxx</td></tr>
<tr><th colspan="3">Discord Account</th></tr>
<tr><th>No</th><th>Specification</th><th>Detail</th></tr>
<tr><td>1</td><td>Name</td><td>Simpleman (as admin)
Simpleman1 (as normal user)</td></tr>
<tr><td>2</td><td>Authorization</td><td>OTI5NjM4ODIxNzkwOTA0MzMx.YjHTaQ.TKBgkamryyoHa5G2EGMWnpyxxxx (authorization admin)
OTI5NjQxODk4NjczNTkwMzIy.YjHTrg.TIEmtB-sBtm-dzSg3OpvV5Rxxxx (authorization normal user)</td></tr>
<tr><th colspan="3">Python</th></tr>
<tr><th>No</th><th>Specification</th><th>Detail</th></tr>
<tr><td>1</td><td>Name</td><td>Python</td></tr>
<tr><td>2</td><td>Version</td><td>3.10.2</td></tr>
</table>

TABLE III. THE BOT FUNCTION DETAILS

No	Function	Detail
1	get_messages	get chat messages sent
2	get_images	get the picture sent
3	get_attachment	get documents sent
4	get_message author	get the author who sent the message
5	get_timestamp	get time sending message
6	get_editedtimestamp	get the time when the message was edited
7	get_guildname	get the name of the guild or group
8	get_channelname	get channel name in group
9	get_editedmessage	get edited message
10	get md5 and sha256	get md5 and sha256 hashing
11	get_deletedmessage	get deleted messages
12	get_url_attachment	get all URL where attachments are stored on the Discord database
13	save json.dumps	save all deleted or edited messages and regular messages
14	save embed_message	save all deleted or edited messages and regular messages in one private channel
15	save CSV	Save all messages CSV format

IV. RESULTS AND DISCUSSION

This section presents acquisition results as part of internal testing to test the functionality and external testing to measure the bot’s performance. It is followed by a discussion that explores how the bot supports available research to acquire Discord’s data.

A. Experimental Configuration

The tests in this paper were carried out using a custom-built experimental apparatus. The following information describes the system setup used for the investigations:

1) System configuration: The studies were carried out on an ASUS TUF Gaming FX504 laptop, which served as the study's principal hardware platform. This laptop model, noted for its durability and dependability, provided a suitable computing environment for experimental activities. The laptop, which included a 2.2 GHz Intel Core i7-8750H CPU with six cores, gave the processing capability to tackle difficult computations. With 16 GB of DDR4 RAM, it provided sufficient memory capacity to accommodate huge datasets and ensure the smooth execution of the experimental methods. The laptop also has a 512 GB NVMe SSD for quick and efficient data storage and retrieval. An NVIDIA GeForce GTX 1050 Ti graphics card with 4 GB provided the graphical capabilities.

2) Software environment: The Windows 10 operating system (version 10.0) was used as the platform for the studies in the experimental setup. For the study, Windows 10 offered a user-friendly and generally compatible environment. Python (version 3.9.6) was used as the primary programming language for carrying out the experiments and analyzing the results. Python's adaptability and vast library ecosystem made it an excellent choice for scientific computing jobs. The Discord API interfaced with the Discord platform to gather and analyze data. The most recent version of the Discord API was used, assuring compatibility with the most recent Discord features and functions. The Pandas package (version 1.3.4) was used for data processing and analysis in the studies. Pandas provide efficient data structures and handling tools.

B. Acquisition Results

Internal testing ensures all features are working as intended and ready to be used by external users.

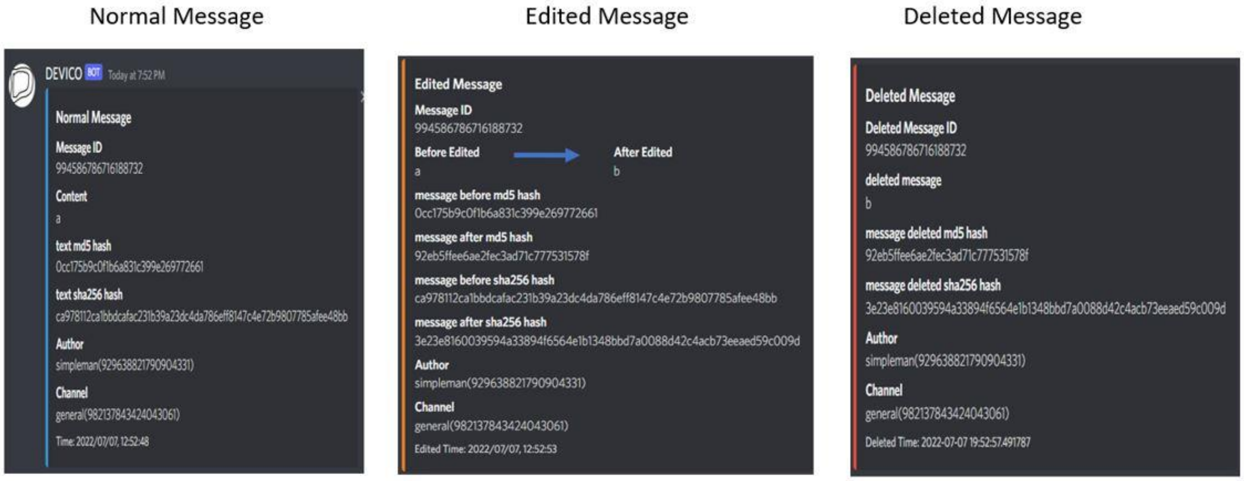
1) Text acquisition: The bot automatically saves all text messages in the Discord server, whether intact, deleted, or edited. The messages are captured on the admin's private channel text created previously and stored in the embedded_message format. The output of text acquisition is shown in Fig. 4.

The detailed content of the acquired text data is presented in Table IV. There is a tittle as the type of message, a Message ID as a unique id denoted by the message, and the hash value of the data is computed to support data integrity checking.

2) Document acquisition: The bot is set to be able to acquire various document formats such as pdf, docx, xlsx and others. Like text acquisition, the document acquisition results are stored directly in the private channel text. Fig. 5 presents two scenario results: the acquisition of the intact document and the deleted document. The hash value of the acquired documents was calculated for both scenarios. The detailed attributes of the documents are presented in Table V.

3) Audio acquisition: The bot collected its attributes for audio data, namely name, extension, resolution, and size. The acquired data is stored directly on the private channel text. The screen capture of the example result can be seen in Fig. 6, while the complete type of the acquired data from the audio is presented in Table VI.

4) Image and video acquisitions: Users can upload various images and video formats. Fig. 7 presents the details acquisition results of the files. The data are stored directly on the private channel text. The detailed data is shown in Table VII.



Normal Message

DEVICO BOT Today at 7:52 PM

Normal Message

Message ID
994586786716188732

Content
a

text md5 hash
0cc175b9c0f1b6a831c399e269772661

text sha256 hash
ca978112ca1bbdcafac231b39a23dc4da786eff8147c4e72b9807785afee48bb

Author
simpleman(929638821790904331)

Channel
general(982137843424043061)

Time: 2022/07/07, 12:52:48

Edited Message

Edited Message

Message ID
994586786716188732

Before Edited → **After Edited**
a → b

message before md5 hash
0cc175b9c0f1b6a831c399e269772661

message after md5 hash
92eb5ffee6ae2fec3ad71c777531578f

message before sha256 hash
ca978112ca1bbdcafac231b39a23dc4da786eff8147c4e72b9807785afee48bb

message after sha256 hash
3e23e8160039594a33894f6564e1b1348bbd7a0088d42c4acb73eeaed59c009d

Author
simpleman(929638821790904331)

Channel
general(982137843424043061)

Edited Time: 2022/07/07, 12:52:53

Deleted Message

Deleted Message

Deleted Message ID
994586786716188732

deleted message
b

message deleted md5 hash
92eb5ffee6ae2fec3ad71c777531578f

message deleted sha256 hash
3e23e8160039594a33894f6564e1b1348bbd7a0088d42c4acb73eeaed59c009d

Author
simpleman(929638821790904331)

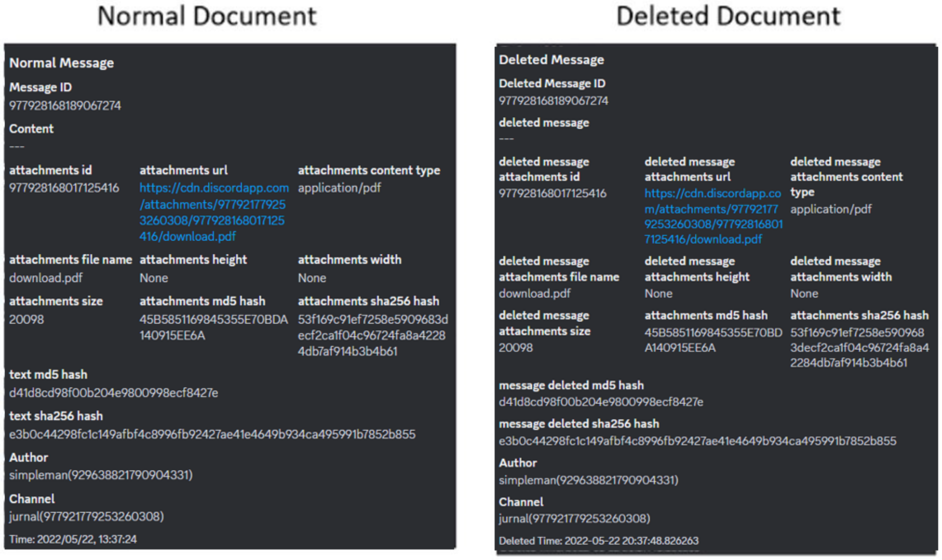
Channel
general(982137843424043061)

Deleted Time: 2022-07-07 19:52:57.491787

Fig. 4. Sample output of the text acquisition.

TABLE IV. SAMPLE OUTPUT OF THE TEXT CHAT ATTRIBUTES

Name	Normal Test	Deleted Text
Message ID	977928168189067274	977928168189067274
Content	---	---
Attachments id	977928168017125416	977928168017125416
Attachments URL	https://cdn.discordapp.com/attachments/977921779253260308/977928168017125416/download.pdf	https://cdn.discordapp.com/attachments/977921779253260308/977928168017125416/download.pdf
Attachments content type	application/pdf	application/pdf
Attachments file name	download.pdf	download.pdf
Attachments height	None	None
Attachments width	None	None
Attachments Size	20098	20098
Attachments MD5	45B5851169845355E70BDA140915EE6A	45B5851169845355E70BDA140915EE6A
Attachments Sha256	53f169c91ef7258e5909683decf2ca1f04c96724fa8a42284db7af914b3b4b61	53f169c91ef7258e5909683decf2ca1f04c96724fa8a42284db7af914b3b4b61
Author	simpleman(929638821790904331)	simpleman(929638821790904331)
Channel	jurnal(977921779253260308)	jurnal(977921779253260308)
Time	2022/05/22, 13:37:24	2022/05/22, 13:37:48



Normal Document

Normal Message

Message ID
977928168189067274

Content
---

attachments id
977928168017125416

attachments url
https://cdn.discordapp.com/attachments/977921779253260308/977928168017125416/download.pdf

attachments content type
application/pdf

attachments file name
download.pdf

attachments height
None

attachments width
None

attachments size
20098

attachments md5 hash
45B5851169845355E70BDA140915EE6A

attachments sha256 hash
53f169c91ef7258e5909683decf2ca1f04c96724fa8a42284db7af914b3b4b61

text md5 hash
d41d8cd98f00b204e9800998ecf8427e

text sha256 hash
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855

Author
simpleman(929638821790904331)

Channel
jurnal(977921779253260308)

Time: 2022/05/22, 13:37:24

Deleted Document

Deleted Message

Deleted Message ID
977928168189067274

deleted message
---

deleted message attachments id
977928168017125416

deleted message attachments url
https://cdn.discordapp.com/attachments/977921779253260308/977928168017125416/download.pdf

deleted message attachments content type
application/pdf

deleted message attachments file name
download.pdf

deleted message attachments height
None

deleted message attachments width
None

deleted message attachments size
20098

attachments md5 hash
45B5851169845355E70BDA140915EE6A

attachments sha256 hash
53f169c91ef7258e5909683decf2ca1f04c96724fa8a42284db7af914b3b4b61

message deleted md5 hash
d41d8cd98f00b204e9800998ecf8427e

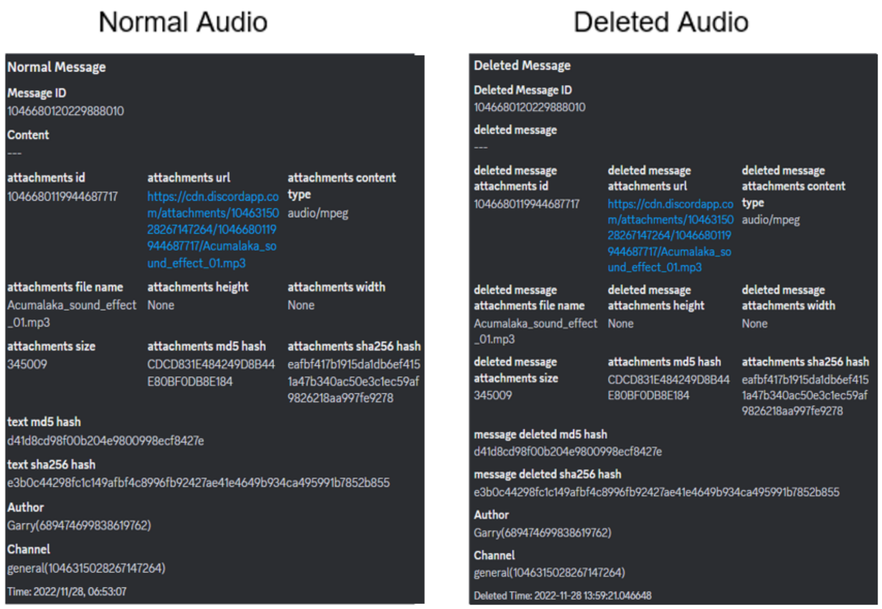
message deleted sha256 hash
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855

Author
simpleman(929638821790904331)

Channel
jurnal(977921779253260308)

Deleted Time: 2022-05-22 20:37:48.826263

Fig. 5. Sample output of the document acquisition.



Normal Message

Message ID
1046680120229888010

Content
---

attachments id
1046680119944687717

attachments url
https://cdn.discordapp.com/attachments/1046315028267147264/1046680119944687717/Acumalaka_sound_effect_01.mp3

attachments content type
audio/mpeg

attachments file name
Acumalaka_sound_effect_01.mp3

attachments height
None

attachments width
None

attachments size
345009

attachments md5 hash
CDCD831E484249D8B44E80BF0DB8E184

attachments sha256 hash
eafbf417b1915da1db6ef4151a47b340ac50e3c1ec59af9826218aa997fe9278

text md5 hash
d41d8cd98f00b204e9800998ecf8427e

text sha256 hash
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855

Author
Garry(689474699838619762)

Channel
general(1046315028267147264)

Time: 2022/11/28, 06:53:07

Deleted Message

Deleted Message ID
1046680120229888010

deleted message
---

deleted message attachments id
1046680119944687717

deleted message attachments url
https://cdn.discordapp.com/attachments/1046315028267147264/1046680119944687717/Acumalaka_sound_effect_01.mp3

deleted message attachments content type
audio/mpeg

deleted message attachments file name
Acumalaka_sound_effect_01.mp3

deleted message attachments height
None

deleted message attachments width
None

deleted message attachments size
345009

attachments md5 hash
CDCD831E484249D8B44E80BF0DB8E184

attachments sha256 hash
eafbf417b1915da1db6ef4151a47b340ac50e3c1ec59af9826218aa997fe9278

message deleted md5 hash
d41d8cd98f00b204e9800998ecf8427e

message deleted sha256 hash
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855

Author
Garry(689474699838619762)

Channel
general(1046315028267147264)

Deleted Time: 2022-11-28 13:59:21.046648

Fig. 6. Sample output of the audio acquisition.

TABLE V. SAMPLE OUTPUT OF THE DOCUMENT ATTRIBUTES

No	Attribute	Detail
1	Title	Normal Message; Edited Message; Deleted Message
2	Message ID	994586786716188732; 994586786716188732; 994586786716188732
3	Content	a; a(before edited) -> b(after edited); b
4	Md5 hash	0cc175b9c0f1b6a831c399e269772661; 0cc175b9c0f1b6a831c399e269772661(before edited) -> 92eb5ffee6ae2fec3ad71c777531578f (after edited); 92eb5ffee6ae2fec3ad71c777531578f
5	Sha256 hash	ca978112ca1bbdcafac231b39a23dc4da786eff8147c4e72b9807785afee48bb; ca978112ca1bbdcafac231b39a23dc4da786eff8147c4e72b9807785afee48bb(before edited) -> 3e23e8160039594a33894f6564e1b1348bbd7a0088d42c4acb73eeaed59c009d (after edited); 3e23e8160039594a33894f6564e1b1348bbd7a0088d42c4acb73eeaed59c009d;
6	Author	simpleman(929638821790904331).
7	Channel	general(982137843424043061)
8	Time	2022/07/07 12:52:48; 2022/07/07 12:52:53; 2022-07-07 12:52:57.491787 UTC

TABLE VI. SAMPLE OUTPUT OF THE AUDIO ATTRIBUTES

Name	Normal Audio	Deleted Audio
Message ID	1046680120229888010	1046680120229888010
Content	---	---
Attachments id	1046680119944687717	1046680119944687717
Attachments URL	https://cdn.discordapp.com/attachments/1046315028267147264/1046680119944687717/Acumalaka_sound_effect_01.mp3	https://cdn.discordapp.com/attachments/1046315028267147264/1046680119944687717/Acumalaka_sound_effect_01.mp3
Attachments content type	audio/mpeg	audio/mpeg
Attachments file name	Acumalaka_sound_effect_01.mp3	Acumalaka_sound_effect_01.mp3
Attachments height	None	None
Attachments width	None	None
Attachments Size	345009	345009
Attachments MD5	CDCD831E484249D8B44E80BF0DB8E184	CDCD831E484249D8B44E80BF0DB8E184
Attachments Sha256	eafbf417b1915da1db6ef4151a47b340ac50e3c1ec59af9826218aa997fe9278	eafbf417b1915da1db6ef4151a47b340ac50e3c1ec59af9826218aa997fe9278
Author	Garry(689474699838619762)	Garry(689474699838619762)
Channel	general(1046315028267147264)	general(1046315028267147264)
Time	2022/11/28, 06:53:07	2022-11-28 06:59:21

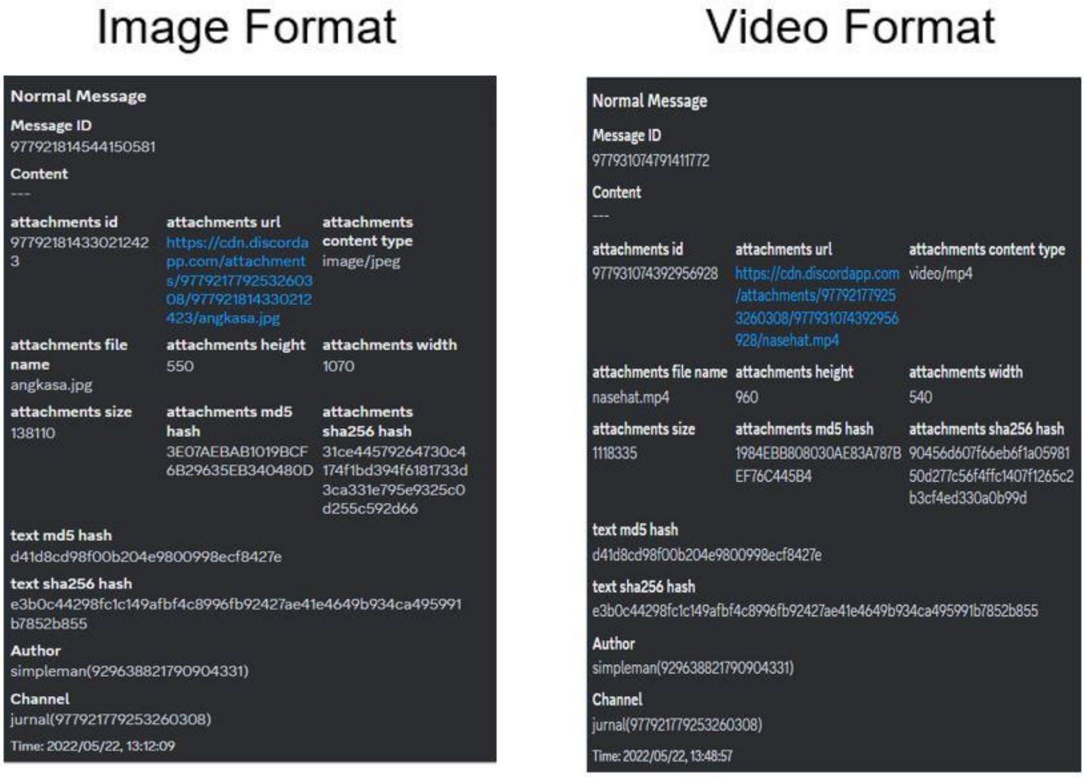


Image Format

Normal Message

Message ID
977921814544150581

Content
---

attachments id
977921814330212423

attachments url
https://cdn.discordapp.com/attachments/977921779253260308/977921814330212423/angkasa.jpg

attachments content type
image/jpeg

attachments file name
angkasa.jpg

attachments height
550

attachments width
1070

attachments size
138110

attachments md5 hash
3E07AEBAB1019BCF6B29635EB340480D

attachments sha256 hash
31ce44579264730c4174f1bd394f6181733d3ca331e795e9325c0d255c592d66

text md5 hash
d41d8cd98f00b204e9800998ecf8427e

text sha256 hash
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855

Author
simpleman(929638821790904331)

Channel
jurnal(977921779253260308)

Time: 2022/05/22, 13:12:09

Video Format

Normal Message

Message ID
977931074791411772

Content
---

attachments id
977931074392956928

attachments url
https://cdn.discordapp.com/attachments/977921779253260308/977931074392956928/nasehat.mp4

attachments content type
video/mp4

attachments file name
nasehat.mp4

attachments height
960

attachments width
540

attachments size
1118335

attachments md5 hash
1984EBB808030AE83A787BEF76C445B4

attachments sha256 hash
90456d607f66eb6f1a0598150d277c56f4ffc1407f1265c2b3cf4ed330a0b99d

text md5 hash
d41d8cd98f00b204e9800998ecf8427e

text sha256 hash
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855

Author
simpleman(929638821790904331)

Channel
jurnal(977921779253260308)

Time: 2022/05/22, 13:48:57

Fig. 7. Sample output of the image and video acquisitions

TABLE VII. SAMPLE OUTPUT OF THE IMAGE AND VIDEO ATTRIBUTES

Name	Image Format	Video Format
Message ID	977921814544150581	977931074791411772
Content	---	---
Attachments id	977921814330212423	977931074392956928
Attachments URL	https://cdn.discordapp.com/attachments/977921779253260308/977921814330212423/angkasa.jpg	https://cdn.discordapp.com/attachments/977921779253260308/977931074392956928/nasehat.mp4
Attachments content type	image/jpeg	video/mp4
Attachments file name	angkasa.jpg	nasehat.mp4
Attachments height	550	960
Attachments width	1070	540
Attachments Size	138110	1118335
Attachments MD5	3E07AEBAB1019BCF6B29635EB340480D	1984EBB808030AE83A787BEF76C445B4
Attachments Sha256	31ce44579264730c4174f1bd394f6181733d3ca331e795e9325c0d255c592d66	90456d607f66eb6f1a0598150d277c56f4ffc1407f1265c2b3cf4ed330a0b99d
Author	simpleman(929638821790904331)	simpleman(929638821790904331)
Channel	jurnal(977921779253260308)	jurnal(977921779253260308)
Time	2022/05/22, 13:12:09	2022/05/22, 13:48:57

C. *Error Rate and Accuracy*

The testing was conducted with 37 users running the bot simultaneously for about one hour. Our record showed that the total sent text messages, files, and deleted data during the testing phase were 74, 296, and 185, respectively. Meanwhile, the acquired data are 74, 289, and 177. Details of the acquired data are presented in Table VIII. It can be observed that the bot cannot identify seven intact and eight deleted files; therefore, these files cannot be acquired.

The error rate (ERR) and Accuracy (ACC) metrics are measured based on the dataset and the acquired data by using Eq. (1) and (2). ERR is calculated as the number of all incorrect predictions divided by the total number of data sets.

Accuracy (ACC) is calculated as the number of all correct predictions divided by the total number of data sets.

$$\text{Error Rate (ERR)} = \frac{\text{FP+FN}}{\text{TP+TN+FN+FP}} \tag{1}$$

$$\text{Accuracy (ACC)} = \frac{\text{TP+TN}}{\text{TP+TN+FN+FP}} \tag{2}$$

where,

FP = the application falsely predicts the true dataset

FN = the application falsely indicating the false dataset

TP = the application accurately predicts the dataset

TN = the application accurately predicts the incorrect dataset

Based on the ERR and ACC formulas above, Table IX presents the confusion matrix based on the performance test results.

TABLE VIII. THE ACQUIRED DATA FROM THE PERFORMANCE TEST

User	Number and Type of Sent Data					Deleted
	Text	*Document*	*Audio*	*Video*	*Picture*	
1	2	2 (pdf)	2 (mpeg)	2 (mp4)	2 (png, jpeg)	5
2	2	2 (docx)	2 (mpeg)	2 (mov)	2 (jpeg)	4
3	2	1 (docx)	2 (mpeg)	2 (mp4)	1 (jpeg)	0
4	2	2 (docx)	2 (mpeg)	2 (mp4)	2 (jpg)	3
5	2	2 (docx)	2 (mpeg)	2 (mp4)	2 (png)	5
6	2	2 (pdf)	2 (mpeg)	2 (mp4)	2(png)	4
7	2	2 (docx)	2 (ogg)	2 (mp4)	2 (png, jpg)	5
8	2	3 (docx, txt)	2 (mpeg)	2 (mp4)	2 (png)	5
9	2	2 (pdf,docx)	2 (mpeg)	-	1 (jpg)	6
10	2	2 (txt)	2 (ogg)	2 (mov)	2 (png)	5
11	2	2 (docx)	2 (ogg)	2 (mov)	1 (jpeg)	3
12	2	3 (docx)	2 (mpeg)	2 (mp4)	4 (png, jpg)	5
13	2	2 (docx)	2 (mpeg)	2 (mp4)	2 (jpg)	5
14	2	2 (pdf)	2 (ogg)	2 (mp4)	2 (jpg)	5
15	2	2 (docx)	2 (mpeg)	2 (mp4)	2 (jpg)	11
16	2	-	2 (mpeg)	2 (mp4)	2 (jpeg)	3
17	2	2 (pdf, docx)	2 (mpeg)	2 (mov)	2 (jpg)	5
18	2	2 (docx)	2 (wav)	2 (mp4)	3 (jpg)	5
19	2	3 (csv,docx,txt)	1 (mpeg)	0	4 (gif, jpeg, png)	1
20	2	2 (docx)	2 (ogg)	2 (mp4)	2 (jpg)	5
21	2	5 (pdf)	3 (mpeg)	4 (mp4)	5 (jpg, png)	10
22	2	2 (docx, pdf)	2 (mpeg)	2 (mp4)	2 (png, jpg)	5
23	2	1 (txt)	-	-	2 (jpg, png)	3
24	2	2 (pdf)	2 (mpeg)	2 (mp4)	2 (jpg)	5
25	2	2 (docx)	2 (mpeg)	2 (mp4)	2 (jpg)	5
26	2	2 (txt)	2 (ogg)	2 (mp4)	2 (jpg)	5
27	2	1 (pdf)	2 (ogg)	2 (mp4)	2 (png)	5
28	2	2 (pdf)	2 (wav)	2 (mkv)	2 (png)	5
29	2	2 (txt)	2 (mpeg)	2 (mp4)	2 (jpg, jpeg)	5
30	2	2 (pptx ,txt)	2 (mpeg)	2 (mkv)	2 (png)	5
31	2	2 (pdf)	2 (mpeg)	2 (mp4)	2 (jpg)	5
32	2	-	-	-	-	2
33	2	2 (pdf, docx)	2 (mpeg)	2 (mp4)	2 (jpg)	5
34	2	2 (txt)	2 (mpeg)	2 (mp4)	2 (jpg)	5
35	2	2 (docx)	2 (mpeg)	2 (mp4)	2 (png, jpg)	6
36	2	2 (docx)	2 (mpeg)	2 (mp4)	3 (png, jpg)	6
37	2	2 (docx, pptx)	2 (mpeg)	2 (mkv)	2 (jpg)	5

TABLE IX. CONFUSION MATRIX FOR A BINARY CLASSIFIER PREDICTING ACCURACY AND ERROR RATE STATUS

	Predicted Positive	Predicted Negative
Actual Positive	289 (TP)	0 (FN)
Actual Negative	7 (FP)	0 (TN)

In Table IX, there are 289 true positives (TP), meaning the classifier correctly identified 289 messages sent by the user. However, there are seven false positives (FP), meaning the classifier predicts seven users have already sent the message, but the fact is not sent yet.

Looking at the values in the confusion matrix in Table IX, the number of error rates generated by the Bot Discord application is 2.3%, and its accuracy is 97.7%.

D. Discussion

The observed results show that the bot successfully recorded messages sent through the Discord app, including the edited and deleted data. However, it is noticed that some data could not be recorded and acquired from the performance results. A potential explanation is that this testing was conducted for the 37 concurrent users, and some data arrived simultaneously. Managing the buffer for storing the consecutive arriving data shall become our concern for future work. It is also necessary to consider storing the acquired data on special storage, including the cloud, because it is possible to get a vast amount of data.

The proposed bot is designed by utilizing the Discord API. The acquired data is presented in Table X. The implementation of the bot by using the Discord API approach gives flexibility because it can be modified according to the acquisition needs, for example, to acquire a guild that contains text messages, images, documents, videos, and audio. This approach opens the possibility of gathering more data as a digital forensic investigation is needed, as much as the app's API can access them.

The other benefit of the bot approach to conduct the acquisition is it can be used to proactively collect the data from all users at once, as it is conducted on the server side. Nevertheless, implement it without compromising user privacy [29] needs to be considered; this could be achieved by providing a notice to the users.

TABLE X. ARTIFACTS ACQUIRED FROM THE PROPOSED DISCORD BOT

Type	Discord Data	API-based Bot Discord
Guild	Name, name_channel, created_at, Id, Category name	✓
Messages	Id, Type, Content (intact, edited, deleted), Attachment (id, filename, size, url, for intact and deleted attachment), Chanel id, Author (id, username, avatar, discriminator, public flag), Embeds, Mentions (roles, everyone), Pinned, Tts, Timestamps (intact, edited, deleted), Flags, Hash (md5 and sha256)	✓

V. CONCLUSION

This study proposed a novel way to collect Discord data using a bot in proactive forensics. The Discord API-based bot saves the data as the embedded message card, stored in a private channel created by the admin. The real-time data can be recorded as the bot is always alive on the server. Therefore, intact, edited, and deleted data are available in advance to be analyzed as needed. All the recorded data is saved locally on the server's storage in easy-to-read formats (i.e., CSV and JSON). The bot is equipped with calculating hash values (i.e., md5 and sha256) for the individual data. Future works may focus on improving the bot's performance to handle massive users, adding remote/cloud storage access, and handling data acquisition for VoIP and encrypted messages.

REFERENCES

[1] V. Fernando, "Cyber Forensics Tools: A Review on Mechanism and Emerging Challenges," in 2021 11th IFIP International Conference on New Technologies, Mobility and Security, NTMS 2021, Institute of Electrical and Electronics Engineers Inc., Apr. 2021. doi: 10.1109/NTMS49979.2021.9432641.

[2] L. Yang, C. Li, T. Wei, F. Zhang, J. Ma, and N. Xiong, "Vacuum: Efficient and Assured Deletion Scheme for User Sensitive Data on Mobile Devices," IEEE Internet Things J, vol. 9, no. 12, pp. 10093–10107, Jun. 2022, doi: 10.1109/JIOT.2021.3119514.

[3] C. Shields, "Towards proactive forensic evidentiary collection," in Proceedings of the Annual Hawaii International Conference on System Sciences, 2010. doi: 10.1109/HICSS.2010.408.

[4] A. Sivaprasad, "Secured Proactive Network Forensic Framework," in 2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC), 2017, pp. 695–699. doi: 10.1109/CTCEEC.2017.8455003.

[5] S. M. Makura, H. S. Venter, R. A. Ikuesan, V. R. Kebande, and N. M. Karie, "Proactive Forensics: Keystroke Logging from the Cloud as Potential Digital Evidence for Forensic Readiness Purposes," in 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT), 2020, pp. 200–205. doi: 10.1109/ICIoT48696.2020.9089494.

[6] D. M. Beskow and K. M. Carley, "Bot-hunter: A Tiered Approach to Detecting & Characterizing Automated Activity on Twitter."

[7] R. U. Rahman and D. S. Tomar, "A new web forensic framework for bot crime investigation," Forensic Science International: Digital Investigation, vol. 33, Jun. 2020, doi: 10.1016/j.fsidi.2020.300943.

[8] X. Zhang, O. Upton, N. L. Beebe, and K. K. R. Choo, "IoT Botnet Forensics: A Comprehensive Digital Forensic Case Study on Mirai Botnet Servers," Forensic Science International: Digital Investigation, vol. 32, Apr. 2020, doi: 10.1016/j.fsidi.2020.300926.

[9] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," Future Generation Computer Systems, vol. 100, pp. 779–796, Nov. 2019, doi: 10.1016/j.future.2019.05.041.

[10] M. Davis, B. McInnes, and I. Ahmed, "Forensic investigation of instant messaging services on linux OS: Discord and Slack as case studies," Forensic Science International: Digital Investigation, vol. 42, p. 301401, Jul. 2022, doi: 10.1016/j.fsidi.2022.301401.

[11] İ. Yıldırım, E. Bostancı, and M. S. Güzel, "Forensic Analysis with Anti-Forensic Case Studies on Amazon Alexa and Google Assistant Build-In Smart Home Speakers," in 2019 4th International Conference on Computer Science and Engineering (UBMK), 2019, pp. 1–3. doi: 10.1109/UBMK.2019.8907007.

[12] M. A. Wani, "Privacy Preserving Anti-forensic Techniques," in Multimedia Security: Algorithm Development, Analysis and Applications, S. A. and B. R. and M. K. Giri Kaiser J. and Parah, Ed., Singapore: Springer Singapore, 2021, pp. 89–108. doi: 10.1007/978-981-15-8711-5_5.

[13] M. K. Rogers, K. C. Seigfried-Spellar, S. Bates, and K. Rux, “Online child pornography offender risk assessment using digital forensic artifacts: The need for a hybrid model,” J Forensic Sci, vol. 66, no. 6, pp. 2354–2361, 2021, doi: https://doi.org/10.1111/1556-4029.14820.

[14] Lokesh Gupta, “What is REST,” https://restfulapi.net/, Apr. 07, 2022.

[15] J. and T. I. Alharbi Soltan and Weber-Jahnke, “The Proactive and Reactive Digital Forensics Investigation Process: A Systematic Literature Review,” in Information Security and Assurance, H. and R. R. J. and B. M. Kim Tai-hoon and Adeli, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 87–100.

[16] M. Cook, A. Marnerides, C. Johnson, and D. Pezaros, “A Survey on Industrial Control System Digital Forensics: Challenges, Advances and Future Directions,” IEEE Communications Surveys and Tutorials, 2023, doi: 10.1109/COMST.2023.3264680.

[17] M. Azzam, L. Pasquale, G. Provan, and B. Nuseibeh, “Forensic readiness of industrial control systems under stealthy attacks,” Comput Secur, vol. 125, Feb. 2023, doi: 10.1016/j.cose.2022.103010.

[18] M. Anderson, “Discord and the Harbormen Gaming Community,” 2019.

[19] Mashud, H. Warni, S. Arifin, M. Ferry, Pebriyandi, and A. Kristiyandaru, “The application of discord as an effort to increase students’ wellbeing in physical education learning during the COVID-19 emergency,” Journal Sport Area, vol. 6, no. 3, pp. 335–348, 2021, doi: https://doi.org/10.25299/sportarea.2021vol6(3).6612.

[20] Y. E. Dayana, O. M. Andre, and L. Andrade-Arenas, “Design of the Discord application as an E-learning tool at the University of Sciences and Humanities,” in Proceedings of the LACCEI international Multi-conference for Engineering, Education and Technology, Latin American and Caribbean Consortium of Engineering Institutions, 2021. doi: 10.18687/LACCEI2021.1.1.9.

[21] M. Motylinski, A. MacDermott, F. Iqbal, M. Hussain, and S. Aleem, “Digital Forensic Acquisition and Analysis of Discord Applications,” in Proceedings of the 2020 IEEE International Conference on Communications, Computing, Cybersecurity, and Informatics, CCCI 2020, Institute of Electrical and Electronics Engineers Inc., Nov. 2020. doi: 10.1109/CCCI49893.2020.9256668.

[22] K. Moffitt, U. Karabiyik, S. Hutchinson, and Y. H. Yoon, “Discord Forensics: The Logs Keep Growing,” in 2021 IEEE 11th Annual Computing and Communication Workshop and Conference, CCWC 2021, Institute of Electrical and Electronics Engineers Inc., Jan. 2021, pp. 993–999. doi: 10.1109/CCWC51732.2021.9376133.

[23] K. Gupta, C. Varol, and B. Zhou, “Digital forensic analysis of discord on google chrome,” Forensic Science International: Digital Investigation, vol. 44, Mar. 2023, doi: 10.1016/j.fsidi.2022.301479.

[24] S. Shin, E. Park, S. Kim, and J. Kim, “Artifacts Analysis of Slack and Discord Messenger in Digital Forensic,” Journal of Digital Contents Society, vol. 21, no. 4, pp. 799–809, Apr. 2020, doi: 10.9728/dcs.2020.21.4.799.

[25] M.-T. and L.-K. N.-A. Sgaras Christos and Kechadi, “Forensics Acquisition and Analysis of Instant Messaging and VoIP Applications,” in Computational Forensics, F. Garain Utpal and Shafait, Ed., Cham: Springer International Publishing, 2015, pp. 188–199.

[26] D. Wijnberg and N.-A. Le-Khac, “Identifying interception possibilities for WhatsApp communication,” Forensic Science International: Digital Investigation, vol. 38, p. 301132, 2021, doi: https://doi.org/10.1016/j.fsidi.2021.301132.

[27] R. Padmanabhan, K. Lobo, M. Ghelani, D. Sujan, and M. Shirole, “Comparative analysis of commercial and open source mobile device forensic tools,” in 2016 Ninth International Conference on Contemporary Computing (IC3), 2016, pp. 1–6. doi: 10.1109/IC3.2016.7880238.

[28] Discord Teams, “Developer Portal,” https://discord.com/developers/docs/change-log, Apr. 06, 2023.

[29] A. R. Javed, W. Ahmed, M. Alazab, Z. Jalil, K. Kifayat, and T. R. Gadekallu, “A Comprehensive Survey on Computer Forensics: State-of-the-Art, Tools, Techniques, Challenges, and Future Directions,” IEEE Access, vol. 10, pp. 11065–11089, 2022, doi: 10.1109/ACCESS.2022.3142508.