Technical Analysis of Copybara | ThreatLabz

By Ruchna Nigam

Published: 2024-08-21 · Archived: 2026-04-05 14:49:39 UTC

Upon launching the application, the user is shown an attacker-defined message screen asking the user to enable the Accessibility Service permission for the application, as shown in the figure below. The Accessibility Service is a legitimate feature on Android phones to assist users with disabilities, however due to the inherent nature of the service, the feature may provide a threat actor with highly granular control over a victim's phone if enabled. If Copybara is installed and not granted the accessibility permission, the malware repeatedly shows notifications and toast messages (as shown in the figure below) to coerce the victim into enabling the service.

Figure 3 : Example Copybara launch screen without the accessibility permission enabled.

If the service is enabled, the user is shown another attacker-defined screen, as shown in the figure below.

Figure 4: Example screenshot of Copybara after the Accessibility Service feature is enabled.

Once the Accessibility Service feature is enabled, the application prevents the user from accessing some options in the Settings menu, ensuring they are unable to uninstall Copybara. In the background, the malware's behavior is determined by its configuration. Copybara is designed to download a list of phishing pages from the C2 server. The Copybara C2 responds with a ZIP file containing counterfeit login pages that mimic popular cryptocurrency exchanges and financial institutions. During our analysis, we discovered the existence of two operational C2 servers that were actively serving the phishing pages.

The figure below shows an open directory of a live C2 server hosting Copybara phishing pages.

Figure 5: Open directory of a live Copybara C2 server hosting phishing pages.

These phishing pages are designed to deceive unsuspecting users into entering their sensitive information. As depicted in the figure below, an example of one such phishing page imitates a login page for a prominent cryptocurrency exchange.

Figure 6: An example Copybara phishing page designed to look like a popular cryptocurrency exchange.

Finally, the application initiates a connection to an MQTT server on port 52997. Copybara subscribes to a specific queue named `commands_FromPC` on this server. This connection enables the application to listen for and receive various commands sent by the C2 server.

The specific commands and their descriptions are provided in the table below.

Command	Functionality
`open_app_setngs`	Opens Settings for the application (otherwise blocked for the user via the Settings menu).
`send_admn_lckdvcs_on`	Checks if the device admin feature is enabled. If it is not enabled, the user is prompted to enter a new lock screen password. Subsequently, the malware

	proceeds to lock the device screen.
`send_inj_lst`	The malware receives a list of package and filenames associated with injects from the C2 server. If a file with a matching name already exists, the malware first deletes the existing file. Subsequently, it proceeds to download a new file from the C2 server. The downloaded file is then written to disk.
`send_custom_opencam`	Initializes an MQTT connection to the C2 server and then starts the device's rear camera.
`send_custom_opencam2`	Initializes an MQTT connection to the C2 server and then starts the device's front camera.
`send_custom_opencam_close`	Ends camera activity.
`send_custom_fullbright`	Maximizes screen brightness.
`send_custom_lowbright`	Minimizes screen brightness.
`send_custom_openmics`	Transmits audio from the microphone to the C2 server.
`send_custom_openmics_close`	Stops transmitting microphone audio to the C2 server.
`send_custom_delallnoties`	Deletes all notifications from the victim's device.
`send_custom_donotdelallnoties`	Stops deleting notifications.
`send_custom_pagebuilder`	Creates a custom view using settings from the `PB_Data` object received from the C2 server. The object contains parameters specifying field types and text specifications to construct a custom webview on-the-fly.
`clickbyid`	Clicks on the screen at the location specified by `gesclick`, which is received from the C2 server.
`del_my_dv_fm_admnpnl`	Closes the connection to the MQTT server and stops the background service.

`Send_Open_Recents`	Shows an overview of recent applications.
`downextraapp`	Downloads an application from an `appurl` parameter provided by the C2 server, saves it under the filename `emptyapp.apk`, and installs it.
`openanyurl`	Opens a URL provided by the C2 server.
`Refrech_hvn_by_Noti`	Dismisses open notifications.
`GlobalParamsActions`	Performs an action specified by the C2 server. The IDs specified by the C2 server correspond to the global actions provided by the Accessibility Service.
`Enable_Noti`	Based on the value of the `Action` flag received from the server, the malware dismisses notifications.
`isAutoSystDalogClker`	Based on the value of the `Action` flag received from the server, the malware takes measures to restrict access to certain options in the Settings menu. This is done to prevent the uninstallation of the malware by the user.
`Request_TurnoffDeviceScreen_FromAndroid`	Turns off the screen capture feature on the victim's device.
`Send_DeviceScreenShot_Permission`	Streams the screen activity of the infected device to the MQTT server. The stream is published to the MQTT server in a queue named `med`.
`Send_Custom_LockScreen`	Downloads an image from the C2 server. The specific image name, referred to as `ImgName`, is provided by the server. Once downloaded, the image is saved as a file named `locscreen.jpg`. However, this functionality is not currently being utilized in the code.
`Send_LockScreen_Overlay`	Minimizes screen brightness and sets a black background.
`Send_LockScreen_Overlay_URL`	Displays a webview that opens a specific URL provided by the server through the `urllink` parameter.
`Send_LockScreen_Overlay_CO`	Displays a webview containing HTML content that is determined by objects received from the server, such as `toptitle`, `bottomtitle`, and `imgurl`.

	The `imgurl` object can either be a local file path or the name of a URI located on the server. In the case of a URI, it is fetched from the C2 server.
`Send_UnLockScreen_Overlay`	Removes an overlay from the screen.
`Request_HVNC_TableTexts_FromAndroid`	Sets a flag value based on the `isShowingOnlyTable` parameter received from the server. However, this functionality is not currently utilized in the code.
`Send_DeviceApps`	Retrieves a list of installed packages on the infected device and sends this information to the MQTT server by publishing it to a queue called `divap_topc`.
`Send_KeyLo_Views`	Enables or disables the keylogger functionality based on the value of the `IsKeyLo` parameter received from the C2 server.
`Send_Click_FromPCToAndroidDevice`	Carries out a gesture on the screen based on the values `clickstartx`, `clickstarty`, `clickx`, and `clicky` which are provided by the C2 server.
`Send_Text_FromPCToAndroidDevice`	Sets the text value, as specified by the `textvalue` parameter, to the currently focused node on the screen (equivalent to injecting keystrokes).
`Send_Important_Views_Only`	Sets a flag based on the value of the `isImportantViewsOnly` parameter received from the C2 server. However, this flag is not currently utilized in the code.
`FormatthisDevice`	Clears browser history and wipes data on the device.
`Send_CallPhoneNumber`	Initiates a phone call to a specific number provided by the C2 server through the `phonenumber` parameter.
`Send_Change_H_Quality`	Adjusts the image quality of screenshots sent to the C2 server based on the value provided by the `intqulaity` parameter received from the C2 server.
`Get_Device_CallLogs`	Publishes contact information from the device to the MQTT server at a queue named `Device_Calls_Logs_Save`.

`Send_GlobalAction_FromPCToAdroid`	Executes an Accessibility Service action on the phone, depending on the value of the `Action` parameter received from the C2 server.
`Send_ChangeVNCFPS`	Adjusts the frames per second (fps) value based on the `fpsdata` parameter received from the C2 server. This adjustment is made when sending images to the server.
`Hide_AppData_Info`	Hides or displays the application icon in the phone menu based on the value of the `isshouldshow` parameter received from the C2 server.
`Send_Wakeup_Device`	Disables the lock screen.
`Send_Request_Permissions`	Requests a specific permission based on the value of the `permission` parameter received from the C2 server.
`Send_Open_CertainApp`	Initiates the launch of a specific application as indicated by the `apppackage` parameter received from the C2 server.
`Send_Uninstall_CertainApp`	Deletes a specific application, as indicated by the `apppackage` parameter received from the C2 server.
`Send_blocknoti_CertainApp`	Enables the blocking of notifications for a specific application as indicated by the `apppackage` parameter received from the C2 server.
`Send_Block_Certain_App`	Blocks the user from opening a specific application as indicated by the `apppackage` parameter received from the C2 server.
`Send_Swipe_Action_ACS`	Performs a swipe action using the values `firstX`, `firstY`, `secondX`, `secondY`, and `intSpeed` provided by the C2 server.
`Send_Swipe_wheel_Action_ACS`	Performs a swipe action using the values for firstX, firstY, secondX, secondY, and intSpeed provided by the C2 server.

`Send_fromtblclick_ACS`	Performs a swipe action using the values for `firstX`, `firstY`, `secondX`, `secondY`, and `intSpeed` provided by the C2 server.
`Send_Pattren_Action_ACS`	Enters a pattern using the values `firstX`, `firstY`, `secondX,` `secondY`, and `intSpeed` provided by the C2 server.
`Send_PZ_Action_ACS`	Performs a gesture using the values for `movx1`, `movy1`, `line1X`, `Line1Y`, `movx2`, `movy2`, `line2X`, `Line2Y`, and `intSpeed` provided by the C2 server.
`Send_Create_Notification`	Creates a notification using the data received from the C2 server through the parameters `title`, `description`, `filename`, and `pkgname`. The `filename` object is utilized to download an icon image from the C2 server.
`Send_Show_Pattren_Buttons`	Sets a flag based on the value of the `IsPattren` parameter received from the C2 server. However, this flag is not currently used in the code.
`SendSMS_To_Admin`	Publishes SMS messages collected from the infected device to the MQTT server at a queue named `Send_SMS_To_Admin_From_Android`.
`del_SMS_FromAdmin`	Deletes a specific SMS from the phone as indicated by the `smsid` parameter received from the server.
`Send_SMSMessage_ToNumber`	Sends an SMS using the phone number and SMS body specified by the `phonenumber` and `SMSBody` parameters received from the C2 server.
`Admin_ConnectedToDevice`	Sends a heartbeat message to the C2 server.

Table 1: Copybara commands and functionalities.
