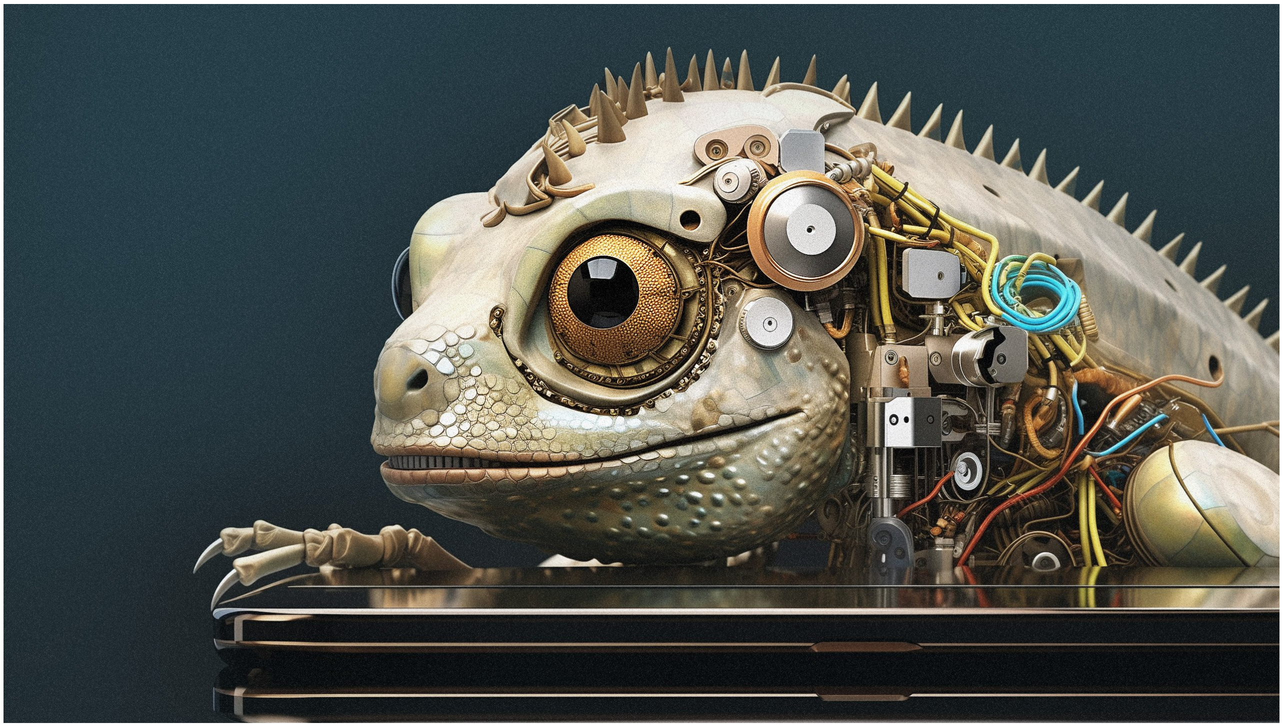
Android Banking Trojan Chameleon can now bypass any Biometric Authentication

threatfabric.com/blogs/android-banking-trojan-chameleon-is-back-in-action



Jump to

Introduction

In January 2023, the Chameleon Banking Trojan emerged as a significant threat, employing various distribution methods to infiltrate the Android ecosystem, with a specific focus on users in Australia and Poland. Aptly named "*Chameleon*," this Trojan showcases its adaptability through multiple new commands, including the examination of app package names. Its primary targets are mobile banking applications, with distribution through phishing pages disguising itself as a legitimate app.

In line with our earlier research (see also our SecuriDropper blog), during this investigation we were able to track and analyze samples related to the updated **Zombinder**.

These Zombinder samples utilize a sophisticated two-staged payload process. They employ the **SESSION_API** through PackageInstaller, deploying the Chameleon samples along with the Hook malware family.

This article takes a deep dive into the newly discovered Chameleon malware variant, distributed via Zombinder. Representing a restructured and enhanced iteration of its predecessor, this evolved Chameleon variant excels in executing Device Takeover (DTO) using the Accessibility Service, all while expanding its targeted region. Our research showcases the new advanced features and capabilities embedded in its malicious payload, highlighting the evolution of this mobile threat.

Chameleon Banking Trojan

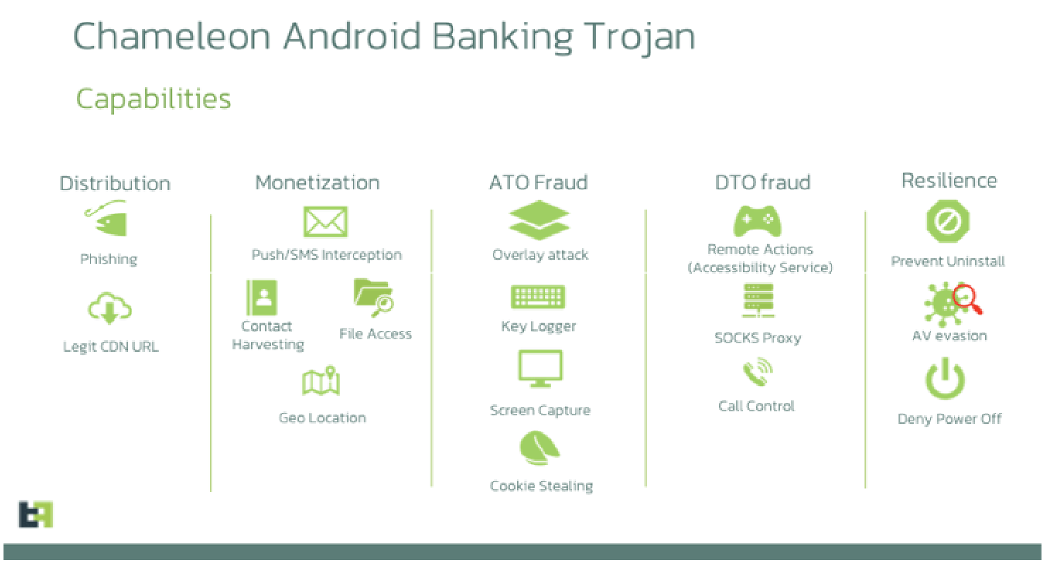
First seen in the wild in early 2023, the Chameleon banking trojan was discovered during its initial development phase. Marked by the use of various loggers, limited malicious functionalities, and well-defined but unused commands, it hinted at a clear potential for further evolution and impact.

This banking trojan displayed a distinctive capability to manipulate a victim's device, executing actions on the victim's behalf through a proxy feature. This feature enables advanced maneuvers like Account Takeover (**ATO**) and Device Takeover (**DTO**) attacks, particularly targeting banking applications and cryptocurrency services. These functionalities relied on the abuse of Accessibility Service privileges.

The earlier variant of the Chameleon banking trojan also employed a diverse set of distribution methods, with a preference for disguising itself as legitimate applications through phishing pages and using a legitimate Content Distribution Network (CDN) for file distribution.

Notably, it predominantly targeted Australia and Poland, where it disguises itself as institutions like the Australian Taxation Office (ATO) and popular banking apps in Poland.

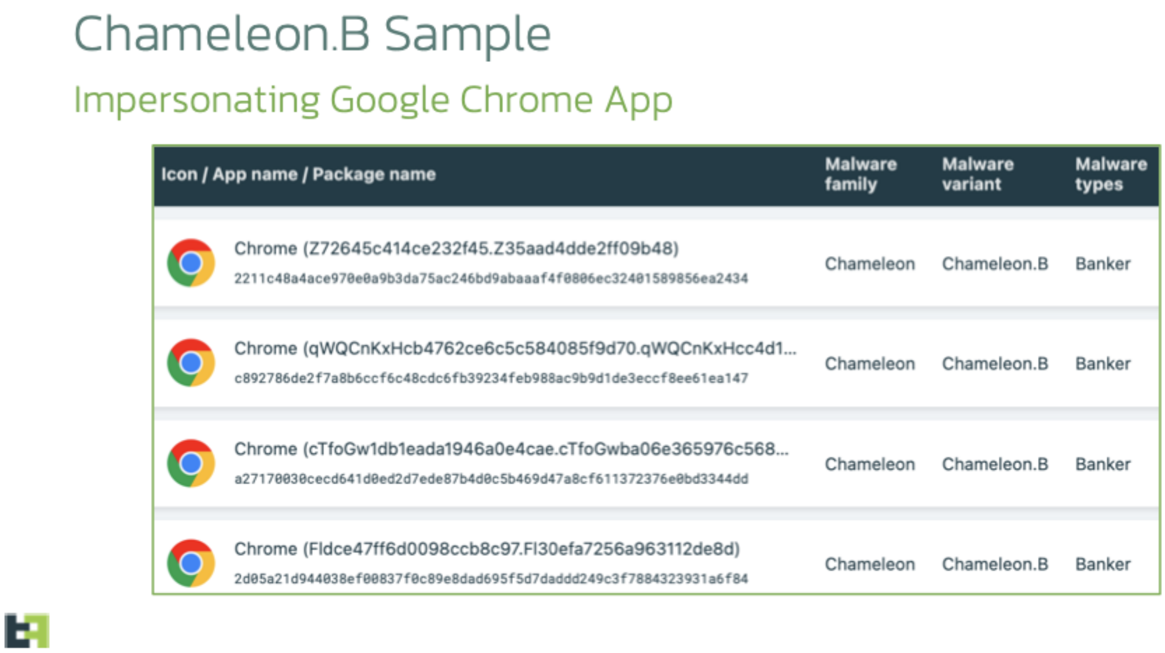
This targeted strategy raises substantial concerns for banks and other financial institutions in these regions. The trojan's adeptness at impersonating trusted apps enhances its potential for widespread impact, underscoring the significance of its threat to the mobile security landscape.



Unveiling the Enhanced Chameleon Variant

As predicted by ThreatFabric's earlier research, after the initial work-in-progress versions, a refined iteration of the Chameleon banking trojan has emerged, carrying over characteristics from its predecessor. This new variant has also expanded its target region to include Android users in **the United Kingdom (UK)** and **Italy**.

This newly discovered variant continues its malicious pursuits, including the Device Takeover (DTO) capability through the Accessibility Service. Distributed through Zombinder, samples of this new variant exhibit a consistent modus operandi while introducing advanced features. Notably, they are often distributed by posing as Google Chrome apps.



Chameleon.B Sample

Impersonating Google Chrome App

Icon / App name / Package name	Malware family	Malware variant	Malware types
Chrome (Z72645c414ce232f45.Z35aad4dde2ff09b48) 2211c48a4ace970e0a9b3da75ac246bd9abaaaf4f0806ec32401589856ea2434	Chameleon	Chameleon.B	Banker
Chrome (qWQCnKxHcb4762ce6c5c584085f9d70.qWQCnKxHcc4d1... c892786de2f7a8b6ccf6c48cdc6fb39234feb988ac9b9d1de3eccf8ee61ea147	Chameleon	Chameleon.B	Banker
Chrome (cTfoGw1db1eada1946a0e4cae.cTfoGwba06e365976c568... a27170030cecd641d0ed2d7ede87b4d0c5b469d47a8cf611372376e0bd3344dd	Chameleon	Chameleon.B	Banker
Chrome (Fldce47ff6d0098ccb8c97.Fl30efa7256a963112de8d) 2d05a21d944038ef00837f0c89e8dad695f5d7daddd249c3f7884323931a6f84	Chameleon	Chameleon.B	Banker

New Features of the Modified Chameleon Variant

Two new features stand out in the updated Chameleon variant: The ability to bypass biometric prompts, and the ability to display an HTML page for enabling accessibility service in devices implementing Android 13's "Restricted Settings" feature.

These enhancements elevate the sophistication and adaptability of the new Chameleon variant, making it a more potent threat in the ever-evolving landscape of mobile banking trojans.

In the next sections, we will take a closer look at the intricacies of the new Chameleon banking trojan, unraveling its capabilities, tactics, and the potential risks it poses to the cybersecurity landscape.

1. Android 13: HTML Prompt to Enable Accessibility Service

Amongst the features of the new Chameleon variant, one feature in particular stands out: This feature involves a device-specific check activated upon the receipt of the command "**android_13**" from the Command and Control (C2) server.

For instance, when the resurfaced Chameleon variant detects installation on an Android 13-based device with applied restrictions on applications, it responds dynamically. The malware displays an HTML page, prompting users to enable the Accessibility service. This step is crucial for the Chameleon malware family, as it relies on this service for the successful execution of Device Takeover (DTO) attacks.

The following code snippet outlines the method employed by the malware to check the restricted settings status of the infected device:

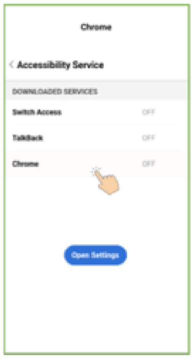
```
if (!class.devicebuild() && (class2.commandlist("android_13", Boolean.valueOf(true)) && Build.VERSION.SDK_INT >= 33 &&
!class2.commandlist("restrict_opened", Boolean.FALSE))
{
    this.startActivity(new Intent(this, class0).putExtra("action", "restriction"));
}
```

Upon receiving confirmation of Android 13 Restricted Settings being present on the infected device, the banking trojan initiates the loading of an HTML page. The page is guiding users through a manual step-by-step process to enable the Accessibility Service on Android 13 and higher. The visual representation below provides an overview of the new Chameleon variant's adaptation in response to the Android 13 environment.

This new functionality demonstrates once again how underground actors respond to and continuously seek to bypass the latest security measures designed to thwart their efforts.

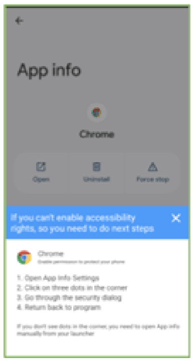
Android 13: HTML Prompt

Guides users through a detailed step-by-step process via an HTML page to enable AccessibilityService within Restricted Settings



“Normal” AccessibilityService request

“Updated” AccessibilityService request, guided through restricted setting



If you can't enable accessibility rights, so you need to do next steps

Chrome
Enable permission to protect your phone

1. Open App Info Settings
2. Click on three dots in the corner
3. Go through the security dialog
4. Return back to program

If you don't see dots in the corner, you need to open App info manually from your launcher

Chameleon is an example of a trend where threat actors adapt droppers and integrate Android 13 restriction checks in malware to bypass security measures.

2. Disrupting Biometric Operations

The new Chameleon variant introduces a feature aimed at interrupting the biometric operations of the targeted device. This feature is enabled by issuing the command "**interrupt_biometric**". Upon receiving this command, the malware executes the "*InterruptBiometric*" method.

This method employs the KeyguardManager API and AccessibilityEvent to assess the screen and keyguard status. It evaluates the keyguard's state concerning various locking mechanisms, such as pattern, PIN, or password. Upon meeting the specified conditions, the malware utilizes the AccessibilityEvent action to transition from biometric authentication to PIN authentication. This bypasses the biometric prompt, allowing the trojan to unlock the device at will.

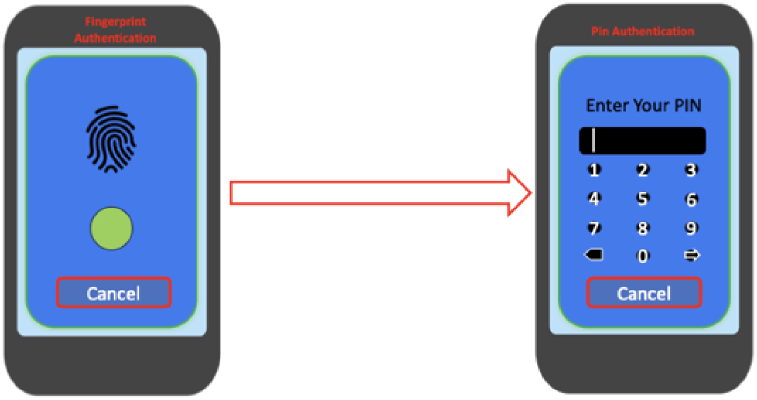
Forcing a fallback to 'standard' authentication provides underground actors with two advantages. Firstly, it facilitates the theft of PINs, passwords, or graphical keys through keylogging functionalities, because biometric data remains inaccessible to these threat actors.

Secondly, leveraging this fallback enables those same actors to unlock devices using previously stolen PINs or passwords. This is achieved through Accessibility actions.

So although the victim's biometric data remains out of reach for actors, they force the device to fall back to PIN authentication, thereby bypassing biometric protection entirely.

Disrupting Biometric Operations

Transition from Biometric authentication to PIN authentication in Infected Device



The following code snippet shows the malware evaluating the KeyGuard state:

```
public final void interruptBiometric(AccessibilityEvent accessibilityEvent0) {
    if (accessibilityEvent0.getPackageName() != null) {
        if (bCBFNOgmB2372b7065b5f58f8f9f.screenstatus != 1 && (KeyguardManager != null &&
(KeyguardManager.isKeyguardSecure()))) {
            if (getInstance.findViewByContainsID(getInstance.getRootInActiveWindow(), "lockPatternView") != null) {
                return;
            }
            if (getInstance.findViewByContainsID(getInstance.getRootInActiveWindow(), "pinEntry") != null) {
                return;
            }
            if (getInstance.findViewByContainsID(getInstance.getRootInActiveWindow(), "passwordEntry") != null) {
                return;
            }
        }
```

This functionality to effectively bypass biometric security measures is a concerning development in the landscape of mobile malware.

3. Task Scheduling & Activity Control

In addition to the features discussed earlier, the updated Chameleon variant introduces a capability also found in many banking other trojans: task scheduling using the AlarmManager API, a feature not present in its earlier "work-in-progress" variant.

While task scheduling is common among trojans, what sets this implementation apart is its dynamic approach, efficiently handling accessibility and activity launches in line with standard trojan behaviour.

The updated Chameleon version supports a new command, "**inejction_type**" [sic]. This command brings a unique element to the trojan's task-scheduling mechanism. It automatically switches from *"a11y*" (a11y is shorthand for accessibility) to "*usagestats*" upon receiving a specific command, depending on whether accessibility is disabled or not. If it is enabled, the malware launches overlay attacks through the "Injection" activity.

If the accessibility service is disabled, the malware seamlessly switches from "*a11y*" to "*usagestats*", collecting information on user app usage on Android devices with Android 23 or higher. This data, including the foreground app, provides an alternative method for determining the foreground application and deciding whether to initiate overlay or injection activity.

```
public final void run() {
    ((AlarmManager)class.this.getApplicationContext().getSystemService("alarm")).set(0, System.currentTimeMillis() +
60000L, PendingIntent.getBroadcast(class.this.getApplicationContext(), 5333, new Intent(class.this.getApplicationContext(),
class2.class), 0xC000000));
    if (!class.accessibility_enabled(class2.class) || (class.list("inejction_type", "a11y").equals("usagestats"))) {
        if (class.usage_stats()) {
    String s = class.this.currentActivity();
            if ((class.commandlist("injection", Boolean.TRUE)) && (class.config(s)) && !false) {
                new Handler(Looper.getMainLooper()).post(new Runnable() {
    @Override
                    public final void run() {
                    ActivityThread.startActivity(new Intent(ActivityThread, class2.class).putExtra("app",
this.val$lastPackage).addFlags(0x10000000).addFlags(0x8000));
                }
```

Conclusion

The emergence of the new Chameleon banking trojan is another example of the sophisticated and adaptive threat landscape within the Android ecosystem. Evolving from its earlier iteration, this variant demonstrates increased resilience and advanced new features. With an expanded focus on users in the UK and Italy, the Trojan employs multiple distribution methods, including deployment via Zombinder and masquerading as legitimate applications such as the Chrome app.

Noteworthy is the trojan's new features, such as HTML pages for Android 13 device evaluations and the disruption of biometric operations, both of which demonstrate its adaptive capabilities. The manipulation of accessibility settings and dynamic activity launches further underscore that the new Chameleon is a sophisticated Android malware strain.

In an ever-evolving threat landscape, understanding the intricacies of the new Chameleon variant is important in formulating effective defensive strategies. ThreatFabric remains committed to unveiling the subtleties of such threats, providing insights that empower users and security professionals to safeguard their digital domains. As threat actors continue to evolve, this dynamic and vigilant approach proves essential in the ongoing battle against sophisticated cyber threats.

Fraud Risk Suite

ThreatFabric’s Fraud Risk Suite enables safe & frictionless online customer journeys by integrating industry-leading mobile threat intel, behavioural analytics, advanced device fingerprinting, and over 10,000 adaptive fraud indicators. This will give you and your customers peace of mind in an age of ever-changing fraud.

Appendix

Indicators of compromise

New Variant of Chameleon Samples

HASH (SHA256)	APP NAME	PACKAGE NAME
2211c48a4ace970e0a9b3da75ac246bd9abaaaf4f0806ec32401589856ea2434	Chrome	Z72645c414ce232f45.Z35aad4dde2ff09b48
0a6ffd4163cd96d7d262be5ae7fa5cfc3affbea822d122c0803379d78431e5f6	Chrome	com.busy.lady