ToxicPanda: a new banking trojan from Asia hit Europe and LATAM

cleafy.com/cleafy-labs/toxicpanda-a-new-banking-trojan-from-asia-hit-europe-and-latam

,

Download your PDF guide to TeaBot

Get your free copy to your inbox now

Download PDF Version

Key Points

- In October 2024, the **Cleafy Threat Intelligence team identified an anomalous Android banking Trojan campaign**. The campaign was initially associated with TgToxic, a banking trojan family reported to be spread in Southeast Asia. Subsequent analyses revealed significant differences in the campaign's code, and we started tracking this family as **ToxicPanda**.
- ToxicPanda's main goal is to initiate money transfers from compromised devices via account takeover (ATO) using a well-known technique called **On-Device fraud** (ODF). It aims to bypass bank countermeasures used to enforce users' identity verification and authentication, combined with behavioral detection techniques applied by banks to identify suspicious money transfers.
- According to its source code, **ToxicPanda is in an early stage of development**, with some commands appearing as placeholders without a real implementation.
- Our investigation successfully **identified an active botnet with over 1500 infected devices** across Italy, Portugal, Spain, and Latin America, targeting 16 banking institutions.
- According to our findings, the **TAs (Threat Actors) behind this malware campaign are likely Chinese speakers**, similar to those responsible for the original TgToxic. Notably, it is uncommon for TAs from this geographical origin to conduct "banking fraud" operations targeting regions such as Europe and LATAM, indicating a potential shift or expansion in their operational focus.

Executive Summary

In late October 2024, Cleafy's Threat Intelligence team observed a significant spike in a new Android malware sample initially classified as TgToxic. However, further analysis revealed that while it shares some bot command similarities with the TgToxic family, the code diverges considerably from its original source. Many capabilities characteristic of TgToxic are notably absent, and some commands appear as placeholders without real implementation. Based on these findings, we started tracking this family as **ToxicPanda**.

ToxicPanda belongs to the modern RAT generation of mobile malware, as its Remote Access capabilities allow Threat Actors (TAs) to conduct Account Takeover (ATO) directly from the infected device, thus exploiting the On Device Fraud (ODF) technique. This consolidation of this technique has already been seen by other banking trojans, such as Medusa, Copybara, and, recently, BingoMod. Adopting a manual approach has several advantages: it requires less skilled developers, TAs can distribute the malware's target base to any banking customers, and bypass various behavioral detection countermeasures put in place by multiple banks and financial services.

Our analysis will reveal that the TAs behind ToxicPanda are Chinese speakers. Notably, it is uncommon for TAs from this geographical origin to conduct "banking fraud" operations targeting regions such as Europe and LATAM, indicating a potential shift or expansion in their operational focus.
Further analysis of the ToxicPanda botnet infrastructure granted our team access to comprehensive telemetry data, revealing the full extent of this campaign:

- **Over 1500 Android devices were infected** and remotely controlled during the ToxicPanda fraud campaign
- **Italy is the primary hotspot**, accounting for more than 50% of the infected devices, **followed by Portugal, Spain, France, and Perù.**

This geographical distribution underscores the ToxicPanda botnet's significant reach and adaptability. These numbers suggest that the operators are expanding their focus beyond primary European targets, hinting at a potential shift towards Latin America.

The following table represents a summary of the TTPs behind ToxicPanda campaigns:

First Evidence	**Early/Mid 2024**
State	Active
Affected Entities	Retail banking
Target OSs	Android Devices
Target Countries	Italy, Portugal, Spain, France, Perù
Infected Chain	Side-loading via Social Engineering
Fraud Scenario	On-Device Fraud (ODF)
Preferred Cash-Out	Instant Payments

First Evidence	Early/Mid 2024
Amount handled (per transfer)	Up to 10K EUR

Malicious App Overview

From a technical standpoint, this sample exhibits reduced capabilities, especially compared to modern banking trojans. However, **the notable differences between this sample and its “ancestor”, TGToxic, are intriguing**. Most commands are either not implemented or exhibit poor refactoring, suggesting that TGToxic served as a foundational template for this malware. The removal of the Automatic Transfer System (ATS) routine and reduced obfuscation routines indicates a downgrade in technical sophistication.

These changes may reflect the developers' inexperience with foreign targets and the challenges of stricter regulations in certain countries, such as PSD2 (Payment Services Directive). Additionally, the shift in primary targets from crypto wallets to financial institutions aligns with the larger demographic of individuals holding bank accounts, at least for the EMEA region. The embedded notes within the code could further imply unfamiliarity with certain technical aspects, highlighting the complexities of adapting and shifting in a “brand new” operational environment.

Our analysts identified the following icons during this investigation. It is evident that TA employs a mix of well-known brands (e.g., Google Chrome, VISA) and decoy icons resembling dating apps to enhance the malware's deceptive capabilities and broaden its reach.

Figure 1 - Identified ToxicPanda’s icons

Figure 1 - Identified ToxicPanda’s icons

The malware’s key features include:

- **Accessibility Service Abuse**: By exploiting Android's accessibility services, ToxicPanda can grant elevated permissions, manipulate user inputs, and capture data from other apps, making it particularly effective in targeting banking applications.
- **Remote Control Capabilities**: ToxicPanda enables remote control of the infected device, allowing attackers to perform various actions, including initiating transactions and modifying account settings without the user's knowledge. With these capabilities, ToxicPanda can enable TAs to perform the On-Device Fraud (ODF) scenario, one of the most dangerous types of banking fraud.
- **Interception of One-Time Passwords** (OTPs): it can intercept OTPs sent via SMS or generated by authenticator apps, allowing cybercriminals to bypass 2FA and authorise fraudulent transactions.

- **Usage of Obfuscation Techniques**: ToxicPanda continually evolves its obfuscation methods to avoid detection. It uses code-hiding techniques to make it difficult for security researchers to analyse the malware.

In this article, we will not delve into these features in detail, as they no longer introduce anything novel and are already widely adopted by modern banking trojans. As previously outlined, the actors behind this campaign likely prioritized their efforts on the operational aspects, given the linguistic barriers and regulatory challenges specific to these territories (e.g., PSD2), as well as the sophisticated countermeasures implemented by anti-fraud teams.

For this reason, in the following chapters, we will focus on some of the unique characteristics identified within the analyzed samples. We will then shift to a detailed examination of the command and control (C2) infrastructure, providing valuable insights into how the group manages and maintains the botnet on an operational level.

Technical Analysis

System configurations and app monitoring

Some interesting artefacts left on the APK are related to files called langs.json and XX.json (where XX is a language file, e.g., it.json, es.json, etc.).

Analyzing the langs.json JSON file, we could spot applications and classes associated with different Android systems or vendor-specific apps (e.g., Samsung, Xiaomi, Huawei, Oppo). These configurations focus on system-level management applications, backup or cleaning utilities as well as security permissions (all applications likely to interfere with or limit the purpose of the malware). Moreover, analyzing the whole structure is possible to catch quite interesting keys, such as pkg, text and action.

Figure 2 - Blocking interaction with unwanted applications

Figure 2 - Blocking interaction with unwanted applications

Those keywords are structured to contain specific information that will be parsed later on from the dedicated malware component. The figure above shows an example of “preventing” users from removing and generally accessing system settings, referring them back to the home screen.

In details:

- **action**: this field represents the actions that need to be performed.

- **pkg**: application interested in this action (e.g., **com.miui.securitycenter**, **com.android.systemui**, **com.android.settings.intelligence**, **com.coloros.safecenter**). It’s worth mentioning that those packages refers also to specific device manufacturers
- **text**: a Chinese string that will be used to match the XX.json file containing language translation for target devices.

Figure 3 - Parsing the ‘langs.json’ file during the executio

Figure 3 - Parsing the ‘langs.json’ file during the execution

Matching internal telemetries and the mechanism observed, it's also possible to infer target countries that are the main focus of this threat. Limiting targets to Europe, it's possible to observe **Italy**, **Spain**, **Portugal**, **France**, **Germany**, and the **UK**. However, considering the linguistic ties between Spanish and Portuguese and the LATAM region, we must recognize that this area could also be a significant target.

Collecting Phone Images

One notable characteristic of this malware, which aligns with practices commonly observed among Chinese-speaking developers, is its capability to access phone albums, convert images to BASE64, and transmit them back to the command and control (C2) server. While this technique is not entirely new— it has already been observed with malware like TrickMo — it represents a significant strategy for gathering potentially sensitive information (e.g., screenshots containing login credentials or virtual cards) from user devices.

Figure 4 - Collecting device’s images

Figure 4 - Collecting device’s images

Debug and Connection Info

In addition, it was possible to discover the following config.toml file inside the asset/ folder:

Figure 5 - Network configuration settings (config.toml)

Figure 5 - Network configuration settings (config.toml)

This file defines configuration settings for a communication or tunneling system, potentially facilitating connections between the malware’s infrastructure and remote devices or servers.

As the previous image shows, this file contains a hardcoded DNS service (114.114.114.114), a Chinese Free Public DNS service named 114DNS. While 114DNS is a legitimate public DNS, its use in malware or suspicious configurations can indicate a connection between TAs and China. Also, since this service is not commonly used outside the region, TAs still consider this region a testing ground for setting up their malware operations against new geographical regions.

Command-List

ToxicPanda significantly overlaps the command names utilised in the TgToxic malware family. Our analysis identified **61 commands** common to both, with highly distinctive names that suggest their presence in both malware is unlikely to be coincidental. This overlap indicates that the same TA (or closed affiliates) could be behind both malware.

Figure 6 - Malware commands

Figure 6 - Malware commands

Conversely, ToxicPanda introduces **33 new commands**, some lacking implementation. Additionally, several commands from TgToxic persist in this variant but remain unimplemented—particularly those associated with EasyClick, a framework enabling UI automation scripts via JavaScript. In TgToxic, this framework was exploited to hijack the Android device's user interface (UI), allowing for actions such as monitoring user input and automating clicks and gestures. In contrast, ToxicPanda does not rely on this framework, though its associated commands remain in the code with blank implementations.
The complete list of commands can be found in **Appendix A - Malware Commands**.

C2 Communication

ToxicPanda contains three hard-coded domains designated for establishing a connection with the Command and Control (C2) server:

- dksu[.]top
- mixcom[.]one
- freebasic[.]cn

Unlike more sophisticated malware that may employ advanced techniques such as Domain Generation Algorithms (DGA) or dynamic configuration updates to determine C2 endpoints, this malware relies on static, pre-defined domains embedded directly within its code.

Figure 7 - Hard-coded C2 server domains

Figure 7 - Hard-coded C2 server domains

In the analyzed sample, domain selection is managed through a switch statement, which defaults to the first domain (dksu[.]top) by setting a specific switch variable to 1. This approach simplifies the initial C2 connection process but reduces the malware's adaptability in cases where one or more of these domains are blocked. However, the C2 server can modify this behavior in real-time by leveraging the setCommandStyle command to change the C2 domain remotely, providing some degree of flexibility despite the hard-coded nature of the initial configuration. While the malware lacks sophisticated C2 domain generation or obfuscation techniques, combining hard-coded domains with selective remote configuration demonstrates a balance between simplicity and operational effectiveness, allowing the attackers to maintain control with minimal complexity.

The chosen domain is prefixed with the subdomain ctrl to establish communication, and an initial HTTP request is sent over HTTPS to initiate contact with the C2 server. This “handshake” request prompts a response containing a JSON payload, including connection parameters such as the port number. This port will subsequently be used for a persistent connection to the C2 server via the WebSocket protocol, which enables low-latency, bidirectional communication.

Figure 8 - Bot’s registration on the C2 server

Figure 8 - Bot’s registration on the C2 server

With the WebSocket protocol, the initial message exchange involves a “login” request from the infected device to the C2 server. This message includes a unique Device ID, allowing the C2 server to identify, register, and monitor each infected device within its botnet. Once the login is successful, the C2 server responds with specific commands based on the fraud campaign’s goals. These commands, outlined in prior sections, prompt the infected device to carry out malicious actions as instructed by the C2 server.

Figure 9 - WebSocket traffic

Figure 9 - WebSocket traffic

ToxicPanda employs AES encryption in ECB (Electronic Codebook) mode to secure network communication. The encryption key is hard-coded within the malware’s source code, derived from a specific byte array, and converted into a string format. In the sample under analysis, this hard-coded encryption key is **0623U2SKT3YY3QB9P**.

Figure 10 - AES encryption routine

Figure 10 - AES encryption routine

A deep dive into ToxicPanda C2 panel

Our analysts successfully obtained visibility into the botnet’s command and control (C2) panel during our investigation into the ToxicPanda Android banking trojan campaign. This visibility was a significant breakthrough, providing crucial insights into the operations of the TAs behind this ongoing banking fraud campaign.

Figure 11 - C2 panel login page

Figure 11 - C2 panel login page

Understanding the inner workings of a botnet control panel is vital in the broader context of **Threat Intelligence**, especially within the realm of Android banking trojans. Visibility into these C2 infrastructures allows analysts to gather invaluable intelligence regarding the techniques and procedures employed by TAs. It also helps us understand the scope of the compromised devices and the specific actions that operators can perform on infected devices.

Access to such information enhances our ability to develop **effective countermeasures, anticipate the attackers' next steps, and ultimately disrupt their operations**.

Figure 12 - C2 panel dashboard

Figure 12 - C2 panel dashboard

In this case, visibility into the botnet’s control panel confirmed that the **ToxicPanda** campaign was orchestrated by a Chinese-speaking group—a rare occurrence in Europe, where this campaign has primarily occurred. The insights gleaned from the panel have further deepened our understanding of this group's operational capabilities and methods of conducting fraud.

The “Machine Management” interface is one of the most important sections within the C2 panel. As shown in the following image, this section provides the fraud operators with a detailed overview of each infected Android device connected to the botnet.

Figure 13 - Victim’s list and details

Figure 13 - Victim’s list and details

This interface is organized into several columns, each representing various aspects of the compromised devices, including:

- **ID and Status**: Displays each compromised device's identification number and online/offline status.
- **Brand and Model**: Information about the device's make and model helps operators understand its technical specifications.
- **Geolocation**: Shows the geographical region based on the device’s time zone, helping the operators narrow down the location of the infected devices.
- **Version and Last Seen**: This details the software version running on the device and when it was last active on the network.

TAs also have various controls, including **updating or resetting scripts, clearing the cache**, or **removing devices** from the botnet. These controls enable fraudsters to maintain or upgrade their malware on the devices, ensuring long-term persistence or adjusting their tactics to remain undetected by anti-fraud measures.

A key feature of this botnet is the ability to initiate On-Device Fraud (ODF), a method increasingly favored by banking fraudsters. The “**Machine Management**” interface allows operators to request real-time remote access to any connected Android device. Once connected, the operator can perform fraudulent transactions directly from the victim’s certified device.

Further analysis of the **ToxicPanda** botnet infrastructure granted our team access to comprehensive telemetry data, revealing the full extent of this campaign. This dataset allowed us to map out the geographic distribution of **over 1,500 infected devices**, highlighting the regions currently experiencing the heaviest concentration of infections.

Figure 14 - Victims’ geographic distribution

Figure 14 - Victims’ geographic distribution

The aggregated data, visualized in the map above, clearly illustrates a pronounced targeting pattern:

- **Italy** is the primary hotspot, accounting for **56.8%** of the infected devices. This concentration suggests that Italy is a strategic focal point for the operators behind ToxicPanda.
- **Portugal** follows, with **18.7%** of compromised devices, indicating a secondary target within Europe.
- **Hong Kong** is the third most affected region, at **4.6%**, potentially reflecting either testing grounds or emerging targets within Asian markets.
- **Spain** and **Peru** are also featured on the list, though they have smaller shares of **3.9%** and **3.4%**, respectively. These numbers suggest that the operators are expanding their focus beyond primary European targets, hinting at a potential shift towards Latin America.

This geographical distribution underscores the significant reach and adaptability of the ToxicPanda botnet. By leveraging these insights, we better understand the botnet's operational focus and can more effectively strategize region-specific defenses. The visibility into regional infection patterns also helps financial institutions and local authorities in the most impacted areas prioritize mitigation efforts and fortify their anti-fraud measures accordingly.

Moreover, our analysts can provide valuable insights into the geographic origin of TA connections and the services they rely on to access the C2 panel. The following image gives an aggregated, high-level view of these extracted telemetries, highlighting key operational patterns:

Figure 15 - Threat Actors’ origin connections

Figure 15 - Threat Actors’ origin connections

Conclusions

Our telemetry data indicates that the threat posed by **ToxicPanda** is becoming increasingly prominent, with a botnet comprising thousands of devices, primarily across Europe. This TA actively targets Europe and potentially extends its reach into the LATAM region, leveraging linguistic and cultural ties.

ToxicPanda needs to demonstrate more advanced and unique capabilities that would complicate its analysis. However, artefacts such as logging information, dead code, and debugging files suggest that the malware may either be in its early stages of development or undergoing extensive code refactoring—particularly given its similarities with TGToxic.

More broadly, we observe a marked shift as Chinese-speaking TAs expand their focus into new geographical regions, especially targeting financial institutions and customers in pursuit of banking fraud opportunities. This trend underscores the mobile security ecosystem's escalating challenge, as the marketplace is increasingly saturated with malware and new threat actors emerge.

An important question arising from this analysis is not just how to defend against threats like ToxicPanda but why contemporary antivirus solutions have struggled to detect a threat that is, in technical terms, relatively straightforward. Although there is no single answer, the lack of proactive, real-time detection systems is a primary issue.

Current security approaches emphasize isolated point detections rather than establishing a **comprehensive "Early Warning system".** Such a system would enable continuous monitoring of suspicious applications, supporting timely classification and mitigation before a full-scale threat can materialize.

Appendix 1: Malware Commands

Command	Description	tgToxic	ToxicPanda
Awake	Keeps the device awake	x	x
adm	Request admin rights.	x	
admLock	Turn Off Screen	x	
admLockRule	Requires a password reset	x	
admPwd	N/A	x	
admSet	N/A	x	
antiDeleteOff	Deactivates anti-delete mode	x	x
antiDeleteOn	Activates anti-delete mode	x	x
ask_relay	N/A	x	
autoBoot	N/A	x	
autoRequestPerm	N/A	x	
back	Activates the back button using the Accessibility service	x	x

Command	Description	tgToxic	ToxicPanda
backstage	Check the status of the backstage service	x	x
black	Activates a black overlay on the screen	x	x
blackB	N/A	x	x
callAcc	N/A, Verifies if Android Accessibility service is active in TgToxic		x
callAppSetting	Opens the app settings	x	x
cancelAwake	Disables device awake mode	x	x
cancelWakeup	Maintains a dimmed screen	x	x
capture	Takes a screenshot	x	x
capturePic	Enables screenshot functionality	x	x
catAllViewSwitch	N/A	x	x
clickB	Clicks within a defined boundary	x	x
clickInput	Selects the input field	x	x
clickPoint	Clicks on a specific point on the screen	x	x
closeEnv	N/A, Sets the accessibility status to inactive in TgToxic		x
closeProtect	N/A	x	
doNotDisturb	Sets the Do Not Disturb mode on the device	x	
fetchIcon	Retrieves icons of wallet applications	x	
gestureB	Executes a series of gestures	x	x
gestureCapture	Captures user’s gesture	x	
googleAuth	N/A, Retrieves Google 2FA code via Accessibility in TgToxic		x
hideShortcuts	Hides the application icon on the device	x	
home	Activates the home button using the Accessibility service	x	x

Command	Description	tgToxic	ToxicPanda
init_data	Initializes specific application data	x	
inputSend	Captures text input	x	x
installApk	N/A, Downloads and installs an APK in TgToxic		x
installPermission	N/A	x	
light	Removes the black screen overlay	x	x
lightT	N/A	x	
lockScreen	Locks the device screen	x	x
logMode	Modifies the log mode	x	
openIntent	Displays a floating toolbar	x	x
openUrl	Open WebPage	x	
permission	Requests all necessary permissions	x	x
permissionB	Automatically grants permissions	x	x
power	N/A	x	x
reConn	N/A	x	
reOpenMe	Reopens the application	x	x
readAlbumLast	Retrieves the last album file name	x	
readAlbumList	Retrieves all album file names	x	x
readAlbumThumbnail	Retrieves thumbnails for all album images	x	x
readContactList	Retrieves all contact information	x	x
readSmsList	Retrieves all SMS messages	x	x
realtimeOnOff	N/A	x	
realtimeSet	N/A	x	
recent	Activates the recent button using the Accessibility service	x	x
releaseScreenCapture	N/A	x	

Command	Description	tgToxic	ToxicPanda
reqPerList	N/A	x	x
reqScreenPermission	N/A, Requests permission for screen capture in TgToxic		x
requestfloaty	N/A, Requests permission for floating windows in TgToxic		x
restartSc	Restarts the Easyclick script service	x	x
restartMe	Restarts the application	x	x
rightClick	Activates the back button	x	x
screen_relay	Configures screenshot settings	x	x
screenshot	N/A	x	
sendAlert	Sends an alert notification	x	
setAppStyle	Modifies the domain used to contact the C2 server	x	
setCam	Captures a photo	x	x
setDebugMode	Sets the debug mode	x	
setDebugOff	Disables debug mode	x	x
setDebugOn	Enables debug mode	x	x
setHideMode	Sets the hide mode	x	
setWakeup	N/A, Schedules a task to wake up the device in TgToxic		x
showShortcuts	Adds an icon to the home screen	x	x
startApk	Launch an application on the device	x	
startCam	Activates the camera	x	x
stopCam	Turns off the camera	x	x
stopHereTest	N/A	x	
swipePwdScreenOff	N/A, Disables enforced password mode in TgToxic		x

Command	Description	tgToxic	ToxicPanda
swipePwdScreenOn	N/A, Enforces password entry mode in TgToxic		x
takeScreen	Gets screen data	x	
touchDown	Initiates a downward swipe	x	x
touchMove	Initiates a move swipe	x	x
touchUp	Initiates an upward swipe	x	x
transparent	N/A	x	
uninstallApk	Uninstalls the application	x	
update	N/A, Updates Easyclick scripts in TgToxic		x
updateApk	N/A, Installs an APK in TgToxic		x
wakeup	Keeps the screen active	x	x
walletList	Uploads the list of installed wallet applications	x	x
wallpaper	N/A	x	x

TLP:AMBER version. This report is a TLP:WHITE version intended for public dissemination and is based on an original TLP:AMBER report. The TLP:AMBER report was previously shared privately with relevant financial CERTs, impacted banking institutions, and law enforcement agencies (LEAs). We encourage trusted researchers and analysts within the community to contact us via email at labs@cleafy.com to request access to the TLP:AMBER version. Access will be granted to those recognized as "trusted entities," allowing for a deeper insight into the findings and supporting data behind this analysis.

Appendix 2: Indicator of Compromise (IOCs)

ToxicPanda Sample:

Hash	App name
2f5c4325f77280b2b58be981f9051f04	Chrome
6e0a7e94ce0a1fe70d43fe727dc41061	dbltest
68139c9e7960d3eb956472bdc5ed5ad2	Chrome

Hash	App name
f5c44a7044572e39e8fb9fa8e1780924	Chrome
4295dfdd9d9fad74ee08d48d13e2b856	Chrome

C2 servers:

Domains
dksu[.]top
mixcom[.]one
freebasic[.]cn

Distribution:

Domain	Campaign/Decoy
fgta[.]lol	99 Spedmart
dpds[.]lol	Chrome
cgtp[.]lol	Amore Live
atnp[.]lol	SK-II
bnwu[.]lol	鑺辫姳鐩存抄 (Braided girl)
dblpap1[.]top	dbltest
dblpap2[.]top	dbltest
dblpap3[.]top	Chrome
dblxz[.]lol	eporner
dbltest[.]top	dbltest
dbltest6[.]top	dbltest
dbltest8[.]top	dbltest
cpt[.]lol	MindMate
unk[.]lol	MindMate
99spedmart[.]me	99 Spedmart

Domain	Campaign/Decoy
mwscg[.]top	Amore Live
ckysp[.]top	Amore Live
kmpct[.]top	Honey Peach

Landing pages:

Figure 16 - Example of Toxic landing pages

Figure 16 - Example of Toxic landing pages

Meet the authors:

Michele Roviello

Alessandro Strino

Federico Valentini

Meet the authors