VB2020
localhost

30 September - 2 October, 2020 / vblocalhost.com

THE EYE ON THE NILE: EGYPT'S CIVIL SOCIETY UNDER ATTACK

Aseel Kayal

Check Point Software Technologies, Israel

aseelk@checkpoint.com

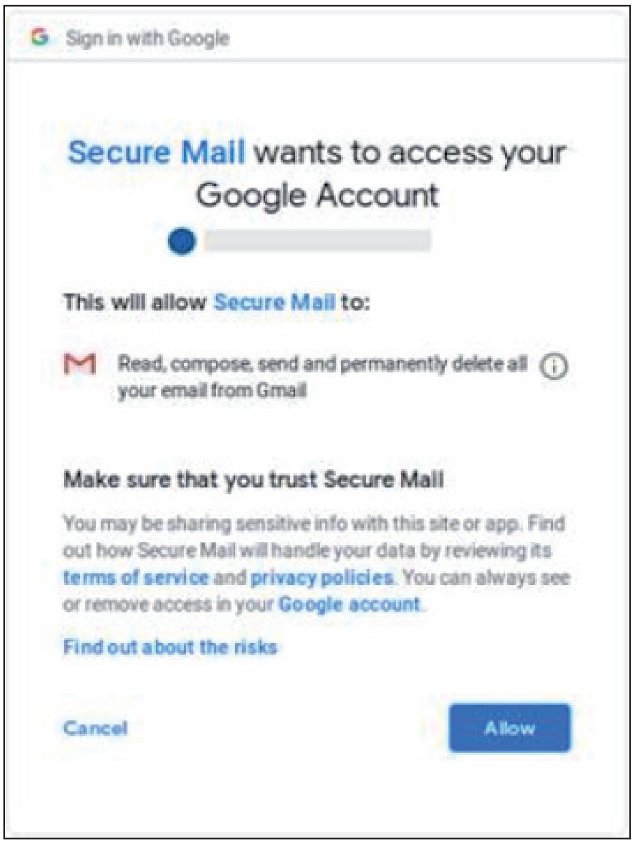
ABSTRACT

This paper details how we tracked a surveillance operation targeting Egypt’s civil society. The attackers took advantage of *Android* applications, phishing websites and third-party email applications to monitor their victims’ correspondences and whereabouts. An OPSEC mistake made by the attackers allowed us to gain insight into their activity and expose their tactics.

INTRODUCTION

Back in March 2019, *Amnesty International* published a report [1] that uncovered a targeted attack against journalists and human rights activists in Egypt. The victims even received an email from *Google* warning them that government-backed attackers had attempted to steal their passwords.

According to the report, the attackers did not rely on traditional phishing methods or credential-stealing payloads, but rather utilized a stealthier and more efficient way of accessing the victims’ inboxes: a technique known as ‘OAuth phishing’. By abusing third-party applications for popular mailing services such as *Gmail* and *Outlook*, the attackers manipulated victims into granting them full access to their emails.



Sign in with Google

Secure Mail wants to access your Google Account

This will allow Secure Mail to:

Read, compose, send and permanently delete all your email from Gmail

Make sure that you trust Secure Mail

You may be sharing sensitive info with this site or app. Find out how Secure Mail will handle your data by reviewing its terms of service and privacy policies. You can always see or remove access in your Google account.

Find out about the risks

Figure 1: Previous OAuth phishing campaign.

After taking a closer look at *Amnesty*’s findings, we were able to find previously unknown or undisclosed malicious artifacts belonging to this operation. A new website we attributed to this malicious activity revealed that the attackers are going after their prey in more than one way, and might even be hiding in plain sight: developing mobile applications to monitor their targets, and hosting them on *Google*’s official *Play Store*.

INFRASTRUCTURE: THE EARLY DAYS

The full list of indicators belonging to this campaign and shared by *Amnesty* on *GitHub* [2] showed multiple websites that used keywords such as ‘mail’, ‘secure’, or ‘verify’, possibly so as not to arouse any suspicions and to masquerade as legitimate mailing services.

By visualizing the information available about these websites, we saw clear connections between them. The addresses they resolved to shared the same IPv4 range or netblock (185.125.228[.]0/22), which belongs to a Russian telecommunications company called *MAROSNET*.

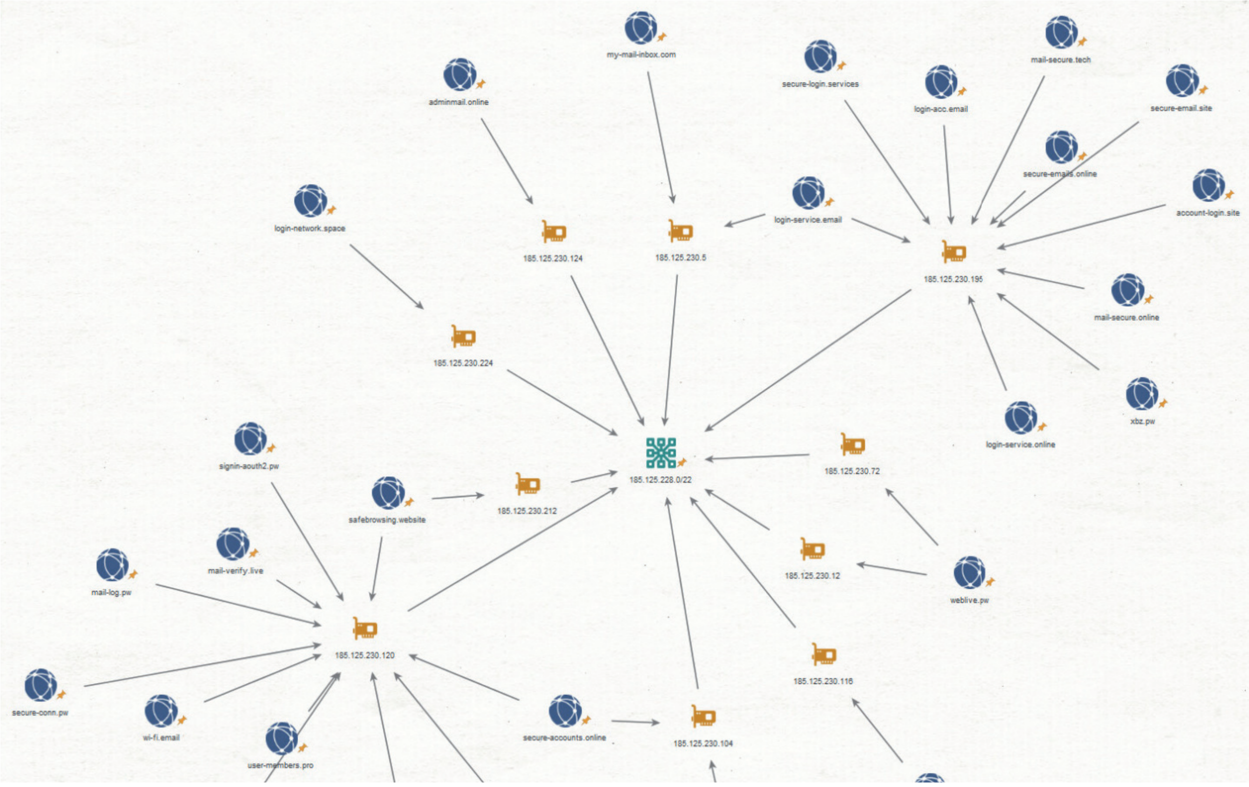
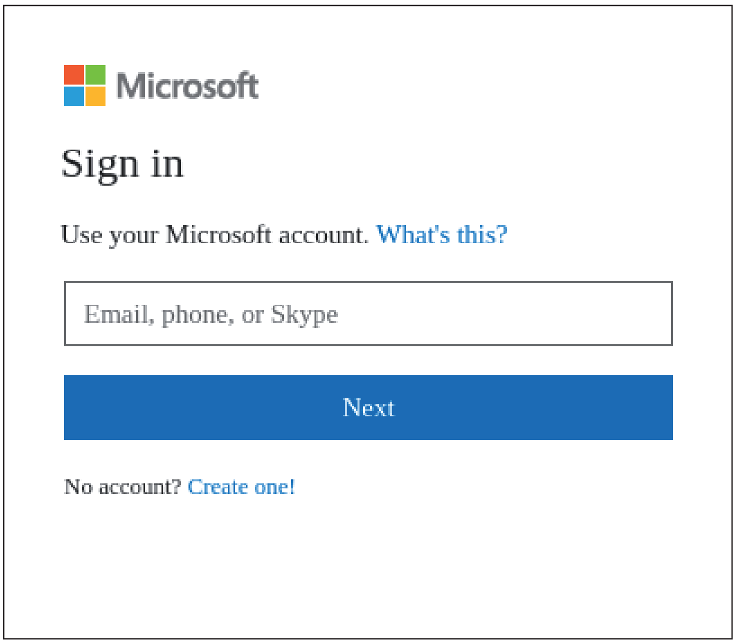


Figure 2: Maltego visualization of campaign infrastructure.

Naturally, the websites cannot be accessed nowadays, but by looking over public scans [3] available for some of them we could see that, in addition to being used for OAuth phishing, they hosted phishing pages that impersonated *Outlook* or *Facebook* and tried to steal login credentials for those services.


Microsoft

Sign in

Use your Microsoft account. What's this?

Email, phone, or Skype

No account? Create one!

Figure 3: Outlook phishing page example.

Interestingly enough, the phishing page shown in Figure 3 accessed a *GitHub* repository from an account called ‘Tarsotelvab’ to retrieve a CSS file it uses. ‘Tarsotelvab’ was mainly active in 2017, and had five repositories in total, as shown in Figure 4.

Another repository under this account that caught our attention included different CSS files belonging to *Qatar Airways*, *AlJazeera*, and a Qatari newspaper called *AlArab*. The files could have been part of a separate phishing campaign impersonating those websites, although we cannot say this for sure as we did not find such an attack.

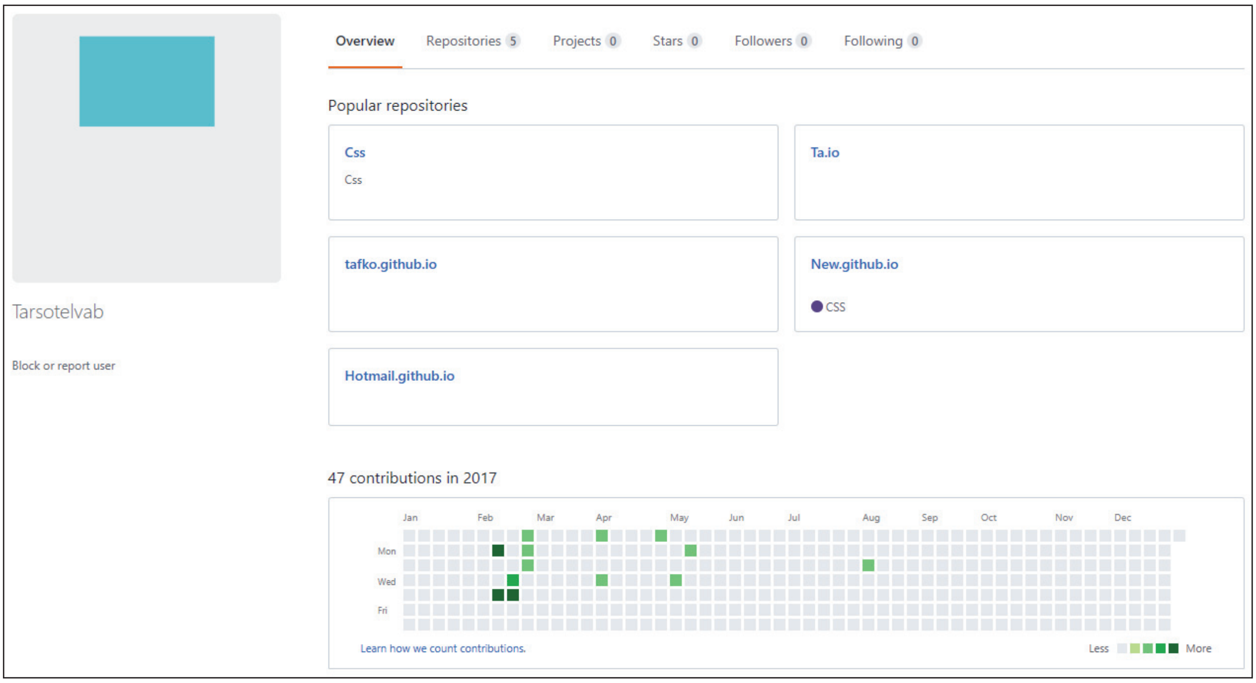


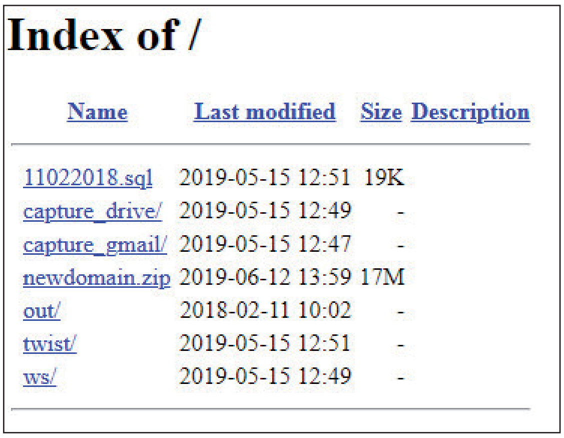
Figure 4: Tarsotelvab's repository on GitHub.

However, what we can say for sure is that the same infrastructure was utilized for both traditional and OAuth phishing attacks, and that it is still being used by the attackers today.

INFRASTRUCTURE 2.0

Many newly registered and unrelated domains resolved to the same IP addresses that were associated with this malicious activity. Looking for relevant indicators was not an easy task, but searching for domains that followed the same naming conventions as the malicious ones and had the words 'mail' or 'verify' in their addresses helped filter the surrounding noise.

One of the domains we came across, maillogin[.]live, had an open directory with files uploaded during May and June, allowing us to download and view all of its back-end scripts.



Index of /

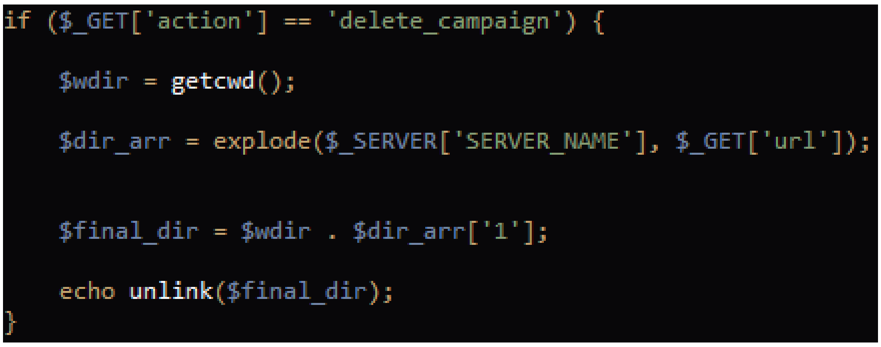
Name	Last modified	Size	Description
11022018.sql	2019-05-15 12:51	19K	
capture_drive/	2019-05-15 12:49	-	
capture_gmail/	2019-05-15 12:47	-	
newdomain.zip	2019-06-12 13:59	17M	
out/	2018-02-11 10:02	-	
twist/	2019-05-15 12:51	-	
ws/	2019-05-15 12:49	-	

Figure 5: Open directory on maillogin[.]live.

In addition to the similar name and shared IP address, we were quickly able to confirm that maillogin[.]live is indeed connected to the previous activity, both because the names of the directories it included were observed in older domains, but also because some of the files referenced those domains.

By downloading the contents of this directory, we were able to get our hands on many PHP scripts, API clients, SQL files and configuration files from the server. Looking into them revealed several aspects about the inner workings of this operation, the functionalities that were implemented on this server and possibly others, and lastly some information about the perpetrators behind it all.

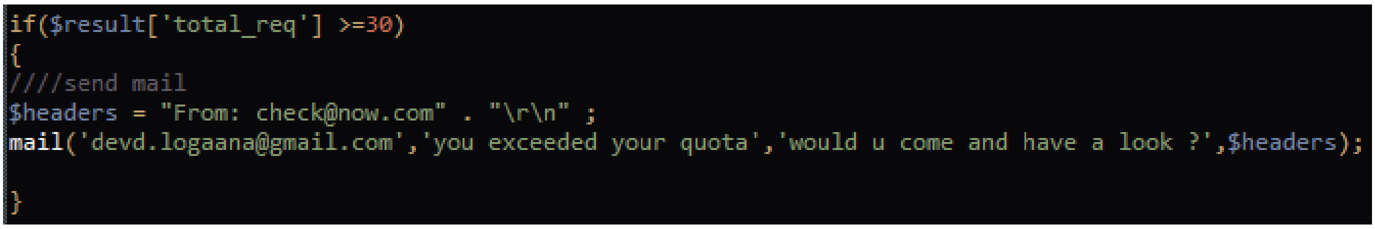
For example, we realized that the attackers can control the operation by sending commands to one of the PHP scripts. The script allowed the attackers to query the data stored on the server, but it had self-destructing capabilities as well, such as removing an existing campaign or deleting the information collected from victims.



```
if ($_GET['action'] == 'delete_campaign') {

    $wdir = getcwd();

    $dir_arr = explode($_SERVER['SERVER_NAME'], $_GET['url']);


    $final_dir = $wdir . $dir_arr['1'];

    echo unlink($final_dir);
}
```

Figure 6: Campaign deletion functionality.

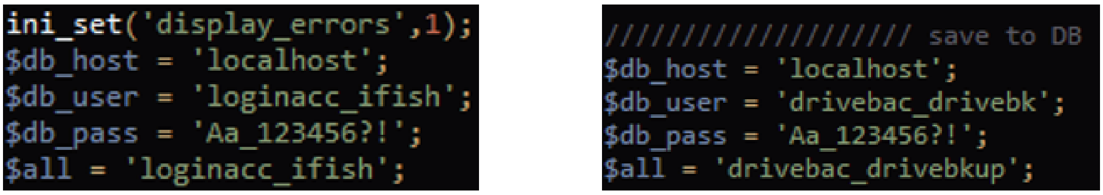
To try and prevent any undesired access, the script counts the total number of requests it receives in one hour. If that number exceeds 30, an email is sent to the address 'devd.logaana@gmail[.]com', alerting the attackers of the suspicious activity on the server and asking them to come take a look, as shown in Figure 7.



```
if($result['total_req'] >=30)
{
////send mail
$headers = "From: check@now.com" . "\r\n" ;
mail('devd.logaana@gmail.com','you exceeded your quota','would u come and have a look ?',$headers);

}
```

Figure 7: Anomalous activity alert functionality.

There were multiple databases that the attackers created to store such statistics from the server, data stolen from victims, blacklisted User-Agents, and more. The credentials for each database were hard coded into some of the scripts, and included usernames such as 'drivebac_drivebk', 'mailveri_shorten' and 'loginacc_ifish'.

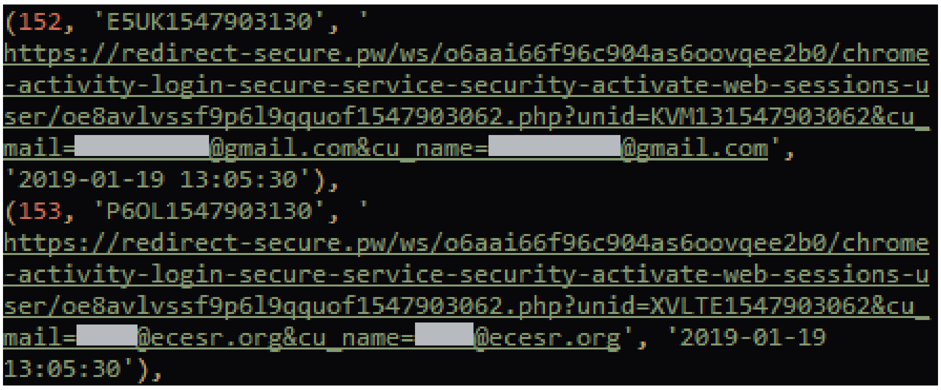


```
ini_set('display_errors',1);
$db_host = 'localhost';
$db_user = 'loginacc_ifish';
$db_pass = 'Aa_123456?!';
$all = 'loginacc_ifish';
```

```
//////////////////// save to DB
$db_host = 'localhost';
$db_user = 'drivebac_drivebk';
$db_pass = 'Aa_123456?!';
$all = 'drivebac_drivebkup';
```

Figure 8: Hard-coded database credentials.

One of the databases even included a list of phishing URLs generated by the attackers, and each link mentioned the email address of the victim it was sent to. This exposed the possible targets of those phishing attempts, which were mainly prominent journalists, human rights activists, and members of non-profit organizations from Egypt.

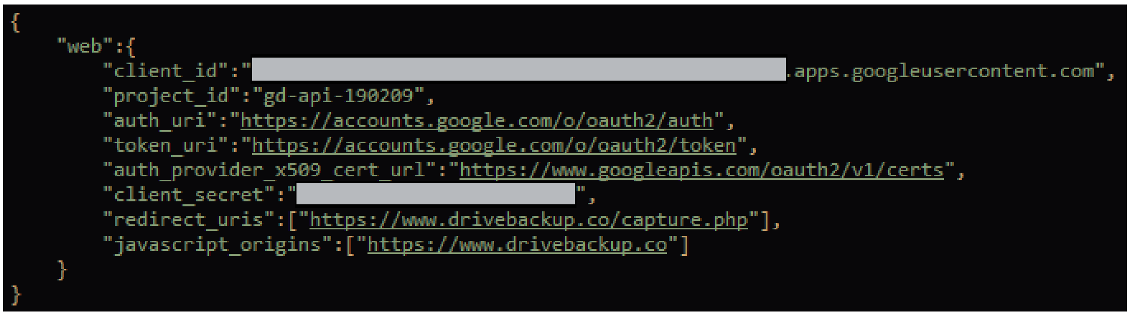


```
(152, 'E5UK1547903130', '
https://redirect-secure.pw/ws/o6aai66f96c904as6oovqee2b0/chrome
-activity-login-secure-service-security-activate-web-sessions-u
ser/oe8avlvssf9p619qquof1547903062.php?unid=KVM131547903062&cu_
mail=[redacted]@gmail.com&cu_name=[redacted]@gmail.com',
'2019-01-19 13:05:30'),
(153, 'P6OL1547903130', '
https://redirect-secure.pw/ws/o6aai66f96c904as6oovqee2b0/chrome
-activity-login-secure-service-security-activate-web-sessions-u
ser/oe8avlvssf9p619qquof1547903062.php?unid=XVLTE1547903062&cu_
mail=[redacted]@ecesr.org&cu_name=[redacted]@ecesr.org', '2019-01-19
13:05:30'),
```

Figure 9: Generated phishing URLs.

THIRD-PARTY APPLICATIONS

In addition to exposing the victims, some of the files in the server helped us understand the full flow of the OAuth phishing attacks. Configuration files of the attackers' third-party applications referred to a script called 'capture.php' that was hosted on drivebackup[.]co. Although this website is no longer accessible, a copy of the same script existed on maillogin[.]live:



```
{
    "web":{
        "client_id":"[redacted].apps.googleusercontent.com",
        "project_id":"gd-api-190209",
        "auth_uri":"https://accounts.google.com/o/oauth2/auth",
        "token_uri":"https://accounts.google.com/o/oauth2/token",
        "auth_provider_x509_cert_url":"https://www.googleapis.com/oauth2/v1/certs",
        "client_secret":"[redacted]",
        "redirect_uris":["https://www.drivebackup.co/capture.php"],
        "javascript_origins":["https://www.drivebackup.co"]
    }
}
```

Figure 10: OAuth phishing configuration.

The 'capture.php' script was in charge of taking the victims to the third-party applications. After being granted the requested permissions, the applications will redirect back to this script, which will then log stolen information about the victims, and eventually redirect them to *Google*.

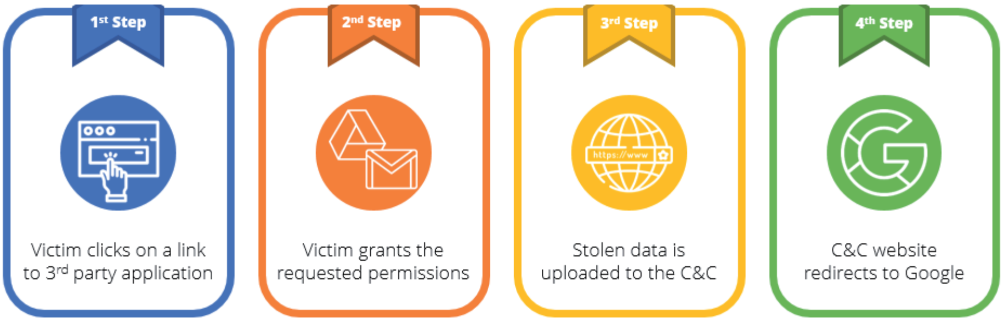
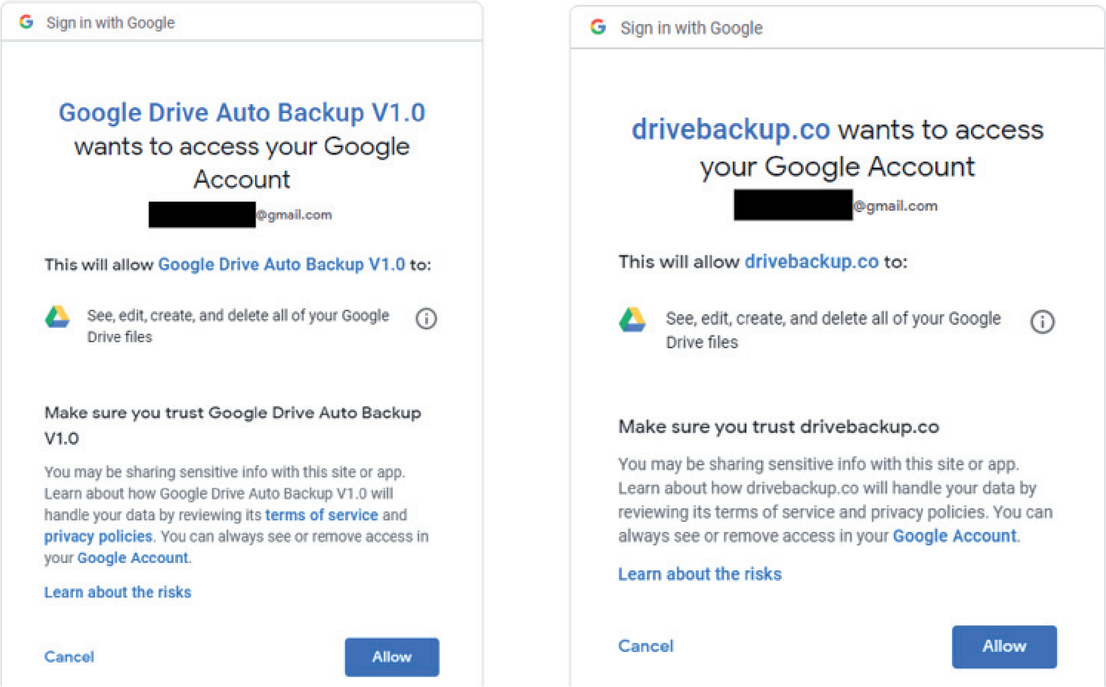

1st Step
Victim clicks on a link to 3rd party application
2nd Step
Victim grants the requested permissions
3rd Step
https://www
Stolen data is uploaded to the C&C
4th Step
C&C website redirects to Google

Figure 11: OAuth phishing compromise flow.

We discovered two new applications that requested permissions to view the victim's basic information and email address, but also to have access to their *Google Drive*, as shown in Figure 12.



Sign in with Google

Google Drive Auto Backup V1.0 wants to access your Google Account

[redacted]@gmail.com

This will allow Google Drive Auto Backup V1.0 to:

See, edit, create, and delete all of your Google Drive files

Make sure you trust Google Drive Auto Backup V1.0

You may be sharing sensitive info with this site or app. Learn about how Google Drive Auto Backup V1.0 will handle your data by reviewing its terms of service and privacy policies. You can always see or remove access in your Google Account.

Learn about the risks

Cancel Allow

Sign in with Google

drivebackup.co wants to access your Google Account

[redacted]@gmail.com

This will allow drivebackup.co to:

See, edit, create, and delete all of your Google Drive files

Make sure you trust drivebackup.co

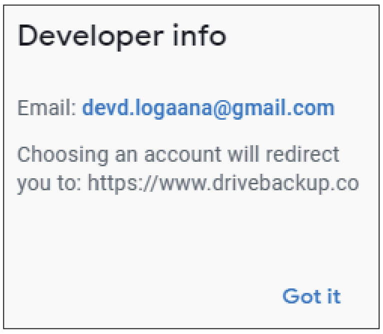
You may be sharing sensitive info with this site or app. Learn about how drivebackup.co will handle your data by reviewing its terms of service and privacy policies. You can always see or remove access in your Google Account.

Learn about the risks

Cancel Allow

Figure 12: Newly discovered OAuth Google applications.

Clicking on the application name reveals details about the developer, who has the same email address as the one we observed in the back-end scripts:



Developer info

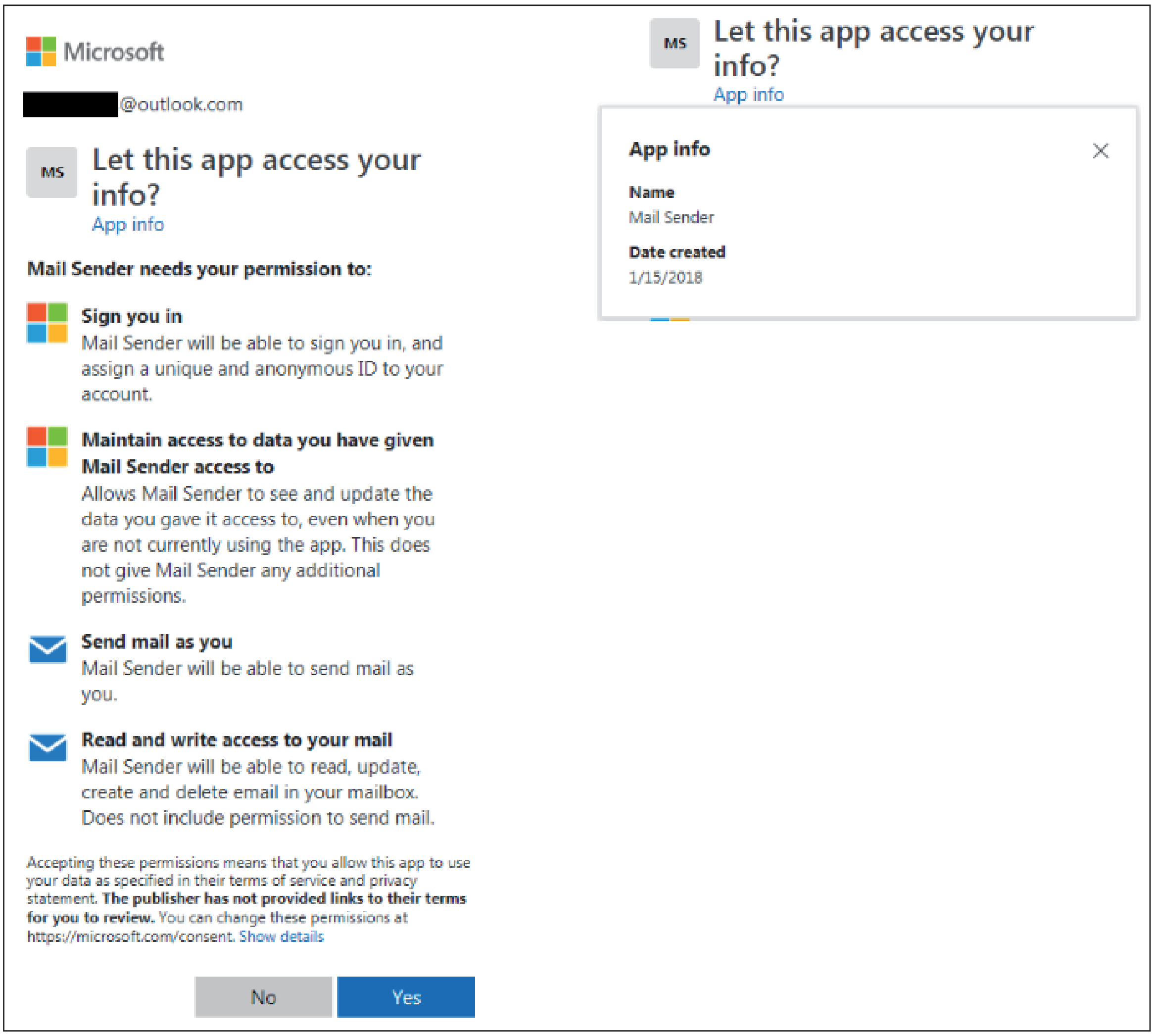
Email: devd.logaana@gmail.com

Choosing an account will redirect you to: https://www.drivebackup.co

Figure 13: Developer information.

The applications that were covered in *Amnesty*'s report accessed the victims' inboxes, but it seems the attackers are not satisfied with only viewing exchanged emails, and are trying to expand their scope to reach the victims' personal files as well.

Other mailing services besides *Gmail* were also targeted, as we found another external application for *Outlook* called 'Mail Sender'. The available information about 'Mail Sender' shows that it was created back in January 2018.



Microsoft

[redacted]@outlook.com

MS Let this app access your info?

App info

Mail Sender needs your permission to:

Sign you in
Mail Sender will be able to sign you in, and assign a unique and anonymous ID to your account.

Maintain access to data you have given Mail Sender access to
Allows Mail Sender to see and update the data you gave it access to, even when you are not currently using the app. This does not give Mail Sender any additional permissions.

Send mail as you
Mail Sender will be able to send mail as you.

Read and write access to your mail
Mail Sender will be able to read, update, create and delete email in your mailbox. Does not include permission to send mail.

Accepting these permissions means that you allow this app to use your data as specified in their terms of service and privacy statement. **The publisher has not provided links to their terms for you to review.** You can change these permissions at https://microsoft.com/consent. Show details

MS Let this app access your info?

App info

App info

Name
Mail Sender

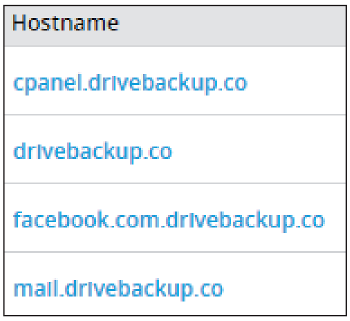
Date created
1/15/2018

Figure 14: OAuth Outlook application from 2018.

The new applications exemplified how the attackers are investing efforts into coming up with new ways to track their targets on multiple platforms. But they also raised questions about the website they redirected to, which had no previous bad reputation or malicious affiliations, and which we can no longer access, drivebackup[.]co.

THE ARCHIVES OF THE WEB

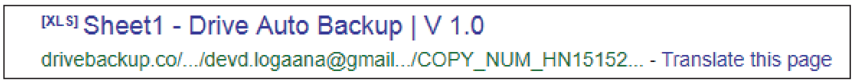
Back in 2018, drivebackup[.]co resolved to an IP address that was already associated with the OAuth phishing attacks: 185.125.130[.]195. It had subdomains such as 'facebook.com.drivebackup[.]co', which could have been used to host phishing pages.



Hostname
cpanel.drivebackup.co
drivebackup.co
facebook.com.drivebackup.co
mail.drivebackup.co

Figure 15: Passive DNS records for 185.125.130[.]195.

Google results for 'drivebackup[.]co' show that it had an open directory when it was active, and one of the files it hosted was called 'COPY_NUM_HN1515242134_OF_عربي.xlsx'. There is no backup for this file in *Google*'s cache, but even the result itself can give away many details.



[XLS] Sheet1 - Drive Auto Backup | V 1.0
drivebackup.co/.../devd.logaana@gmail.../COPY_NUM_HN15152... - Translate this page

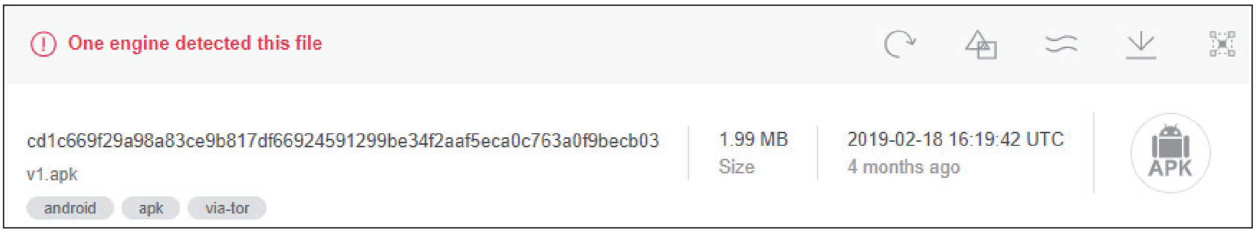
Figure 16: Cached Google result for drivebackup[.]co.

First, the URL from which the file was downloaded, '/downloaded/devd.logaana @gmail[.]com', mentions the same email address again. The title of the document, 'Sheet1 – Drive Auto Backup | V 1.0', matches the name of the *Google Drive* application. And lastly, the number that appears in the filename, '1515242134', is a UNIX timestamp of 6 January 2018, and might be the date on which this file was created.

Not much else was known about drivebackup[.]co other than this, but that was when we discovered a connection to an *Android* application.

TRACKING YOUR EVERY MOVE

An *Android* application that was submitted to *VirusTotal* during February communicated with drivebackup[.]co. This application has one detection only, and is called 'v1.apk':



One engine detected this file

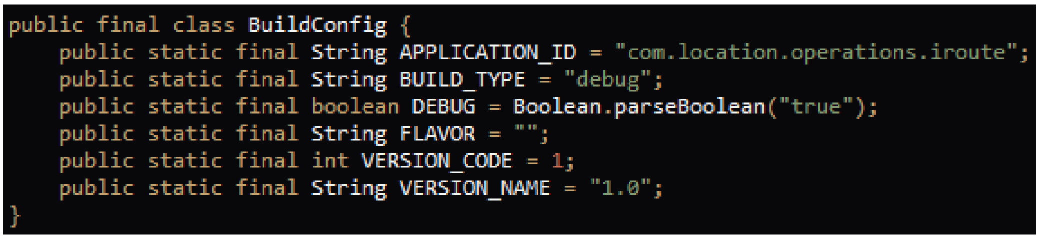
cd1c669f29a98a83ce9b817df66924591299be34f2aaf5eca0c763a0f9becb03
v1.apk
android apk via-tor

1.99 MB
Size

2019-02-18 16:19:42 UTC
4 months ago

Figure 17: Android application communicating with drivebackup[.]co.

The sample has no icon, its build type is set to 'debug', and its internal version number is set to '1.0'. All of this may indicate that it is still in early testing phases.



```
public final class BuildConfig {
    public static final String APPLICATION_ID = "com.location.operations.iroute";
    public static final String BUILD_TYPE = "debug";
    public static final boolean DEBUG = Boolean.parseBoolean("true");
    public static final String FLAVOR = "";
    public static final int VERSION_CODE = 1;
    public static final String VERSION_NAME = "1.0";
}
```

Figure 18: Internal configuration of the application.

When the application runs, it asks the user to enter an activation key, which is then sent to drivebackup[.]co. This may be a way to filter any undesired usage of the application, since the key has to be approved by the server.

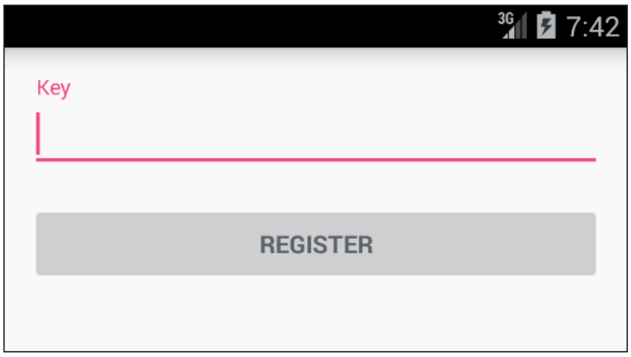
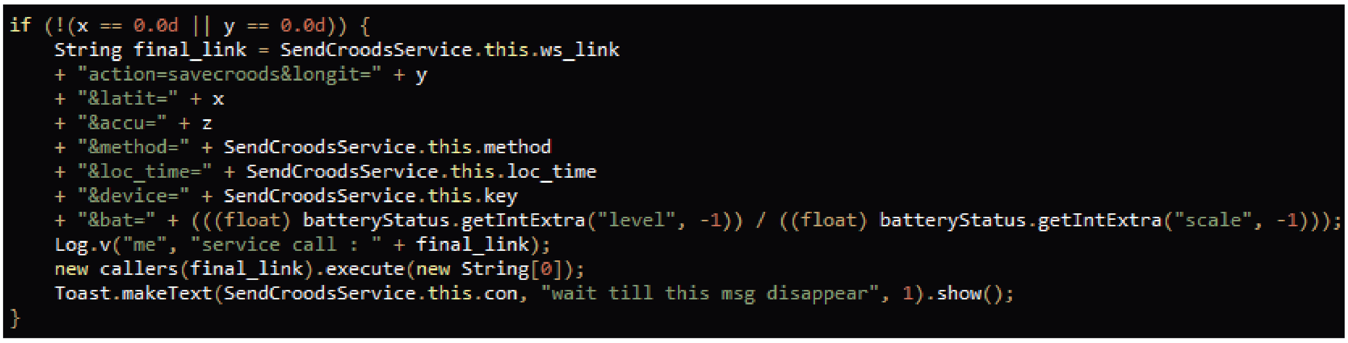


Figure 19: Application's interface.

Once the key is approved, the application proceeds to implement its only functionality: collecting information about the user's location. The device's exact coordinates, the accuracy of those coordinates, local time and battery status are all uploaded to drivebackup[.]co as well.



```
if (!(x == 0.0d || y == 0.0d)) {
    String final_link = SendCroodsService.this.ws_link
    + "action=savecroods&longit=" + y
    + "&latit=" + x
    + "&accu=" + z
    + "&method=" + SendCroodsService.this.method
    + "&loc_time=" + SendCroodsService.this.loc_time
    + "&device=" + SendCroodsService.this.key
    + "&bat=" + (((float) batteryStatus.getIntExtra("level", -1)) / ((float) batteryStatus.getIntExtra("scale", -1)));
    Log.v("me", "service call : " + final_link);
    new callers(final_link).execute(new String[0]);
    Toast.makeText(SendCroodsService.this.con, "wait till this msg disappear", 1).show();
}
```

Figure 20: Information collected by the application.

This functionality by itself is not malicious, but combined with other characteristics of the application, it raises many red flags. Although the chosen package name for the application 'com.location.operations.iroute' provides an accurate description of what it does, the displayed name tells a different story.

The application calls itself 'iLoud 200%' and it displays messages to the user saying that the 'ringtone is now 100% louder'. The application masquerades as a service that increases the device's volume, although it implements no such functionality.

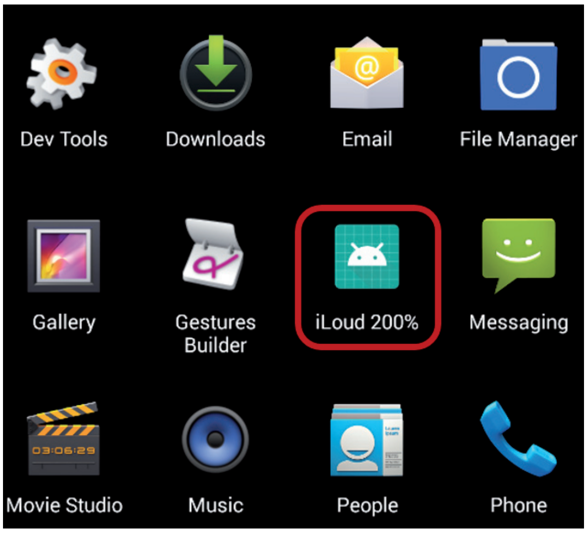
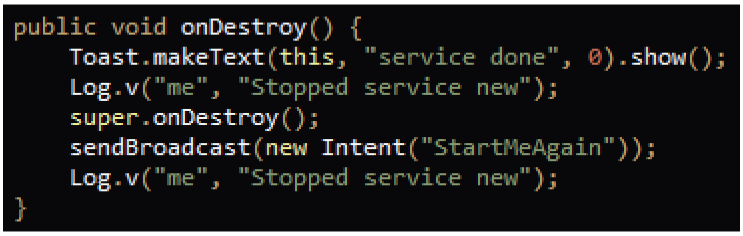


Figure 21: Application's icon and name.

Another warning sign was the service that collects the coordinates, which automatically runs when the device is started. If it is stopped for any reason, a broadcast is sent to another class, which is in charge of restarting this service and making sure it keeps running.



```
public void onDestroy() {
    Toast.makeText(this, "service done", 0).show();
    Log.v("me", "Stopped service new");
    super.onDestroy();
    sendBroadcast(new Intent("StartMeAgain"));
    Log.v("me", "Stopped service new");
}
```

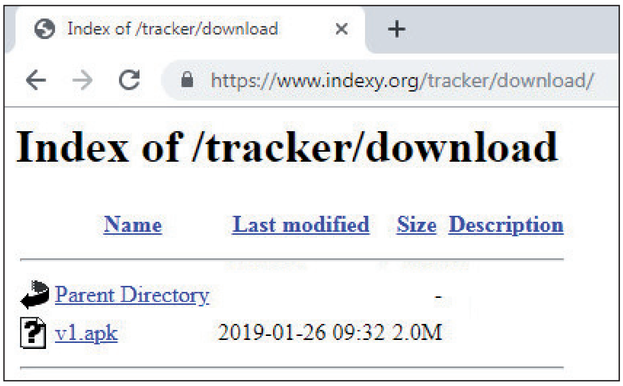
Figure 22: Service functionality.

In addition to using a website we have seen in a malicious context, we realized that this application is meant to monitor the location of the device it is installed on at all times, and to do so while hiding its true intentions. Therefore, it seems that it was created for a purpose that coincides with the surveillance motives behind the phishing activity, and that drivebackup[.]co is not the only connection between the two.

Hunting for related samples with the unique strings that appeared in the application led us to an older variant, which uses the package name 'com.whatsapp' to look like a legitimate service. This variant did not communicate with drivebackup[.]co, and instead referred to the website from which 'v1.apk' was downloaded: indexy[.]org.

MORE INDICES

Although 'v1.apk' was submitted to *VirusTotal* during February, we could see that it was uploaded to indexy[.]org in January:



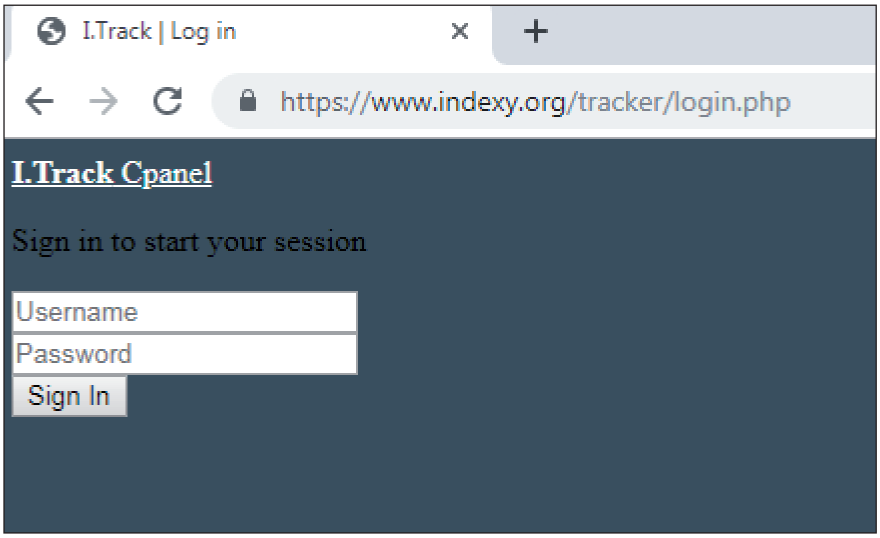
Index of /tracker/download
https://www.indexy.org/tracker/download/

Index of /tracker/download

Name	Last modified	Size	Description
Parent Directory		-	
v1.apk	2019-01-26 09:32	2.0M	

Figure 23: Application's upload timestamp.

An administration panel to manage this application exists on indexy[.]org as well, and the application is referred to as 'I.Track' in the login page:

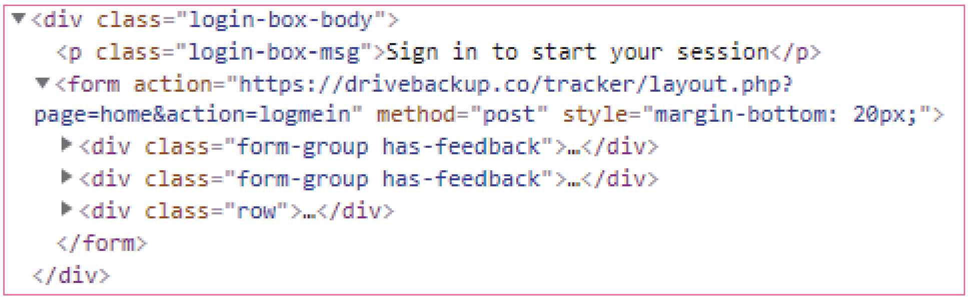


I.Track Cpanel

Sign in to start your session

Figure 24: C&C web login panel.

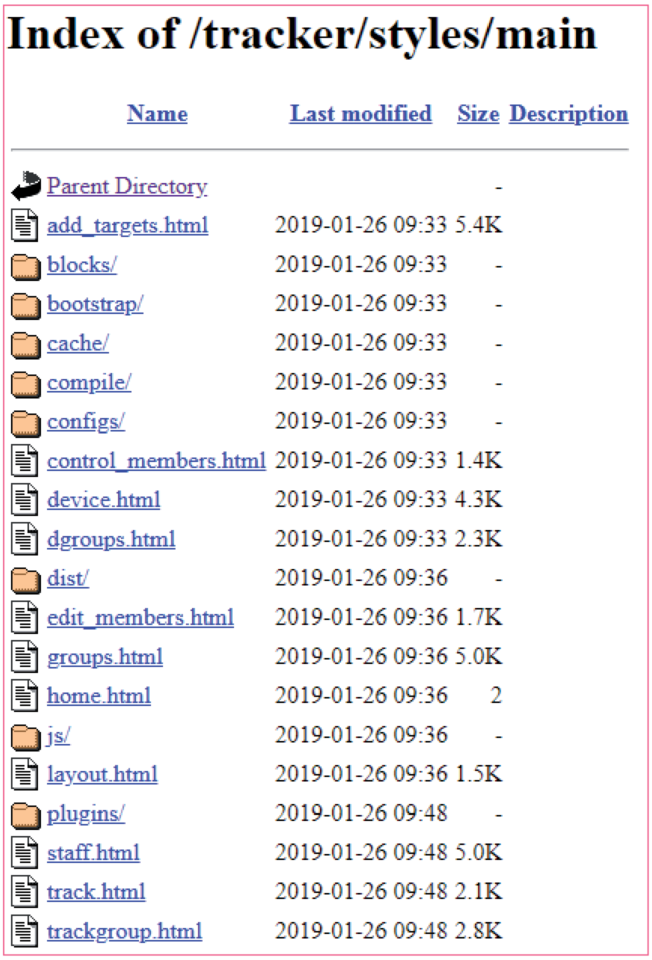
Many of the JS and CSS files that the panel needs are missing and cannot be loaded, since the application imports most of its resources from drivebackup[.]co.



```
▼<div class="login-box-body">
   <p class="login-box-msg">Sign in to start your session</p>
  ▼<form action="https://drivebackup.co/tracker/layout.php?
  page=home&action=logmein" method="post" style="margin-bottom: 20px;">
    ▶<div class="form-group has-feedback">…</div>
    ▶<div class="form-group has-feedback">…</div>
    ▶<div class="row">…</div>
   </form>
  </div>
```

Figure 25: Source code artifacts point to drivebackup[.]co.

Although we did not log into this panel, the 'styles' directory included the HTML templates of the pages available for the administrators after logging in.

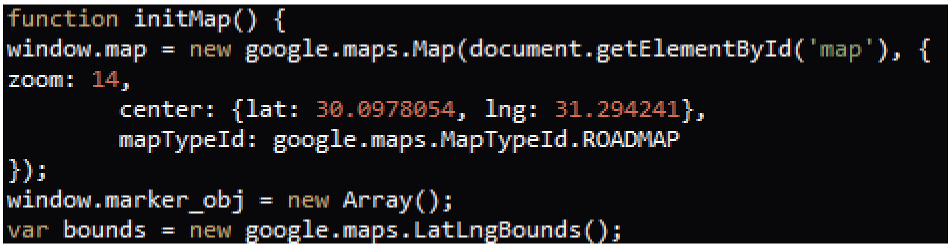


Index of /tracker/styles/main

Name	Last modified	Size	Description
Parent Directory		-	
add_targets.html	2019-01-26 09:33	5.4K	
blocks/	2019-01-26 09:33	-	
bootstrap/	2019-01-26 09:33	-	
cache/	2019-01-26 09:33	-	
compile/	2019-01-26 09:33	-	
configs/	2019-01-26 09:33	-	
control_members.html	2019-01-26 09:33	1.4K	
device.html	2019-01-26 09:33	4.3K	
dgroups.html	2019-01-26 09:33	2.3K	
dist/	2019-01-26 09:36	-	
edit_members.html	2019-01-26 09:36	1.7K	
groups.html	2019-01-26 09:36	5.0K	
home.html	2019-01-26 09:36	2	
js/	2019-01-26 09:36	-	
layout.html	2019-01-26 09:36	1.5K	
plugins/	2019-01-26 09:48	-	
staff.html	2019-01-26 09:48	5.0K	
track.html	2019-01-26 09:48	2.1K	
trackgroup.html	2019-01-26 09:48	2.8K	

Figure 26: Additional artifacts on indexy[.]org.

Examining the names of the HTML pages can reveal what actions the administrators can perform using the panel: add targets, control existing ones, and track them all. One of the pages called 'trackgroup.html' initializes a *Google Map*, adds the collected coordinates to it, and defines the default location which this map zooms into.



```
function initMap() {
window.map = new google.maps.Map(document.getElementById('map'), {
zoom: 14,
        center: {lat: 30.0978054, lng: 31.294241},
        mapTypeId: google.maps.MapTypeId.ROADMAP
});
window.marker_obj = new Array();
var bounds = new google.maps.LatLngBounds();
```

Figure 27: Embedded initialization coordinates.

Surprisingly, those coordinates point to an unnamed building in Cairo:

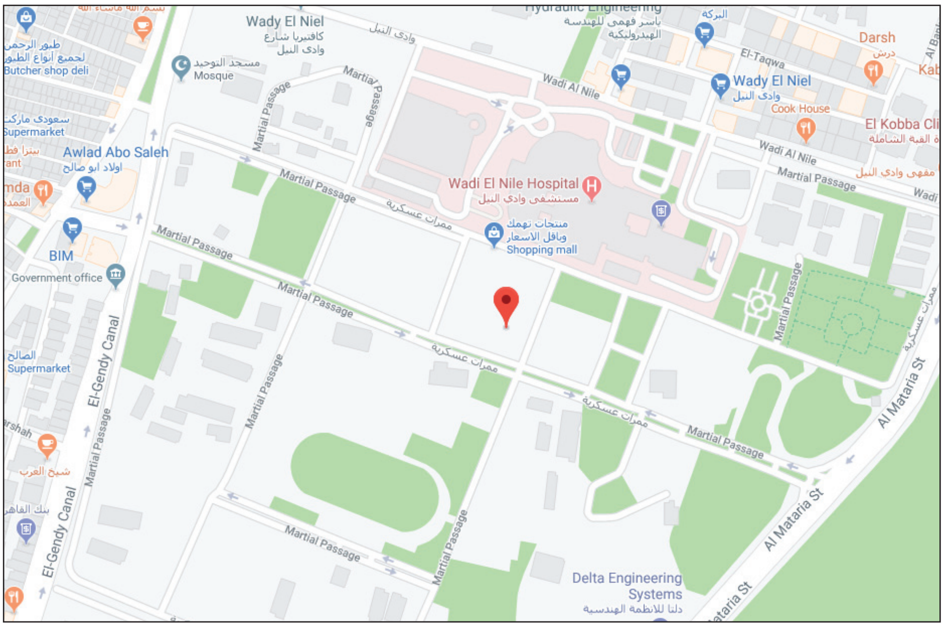


Figure 28: Google Maps view of the coordinates.

Another interesting finding was in the 'dist' directory under 'styles', which contains images that are provided by default from the AdminLTE bootstrap template. But there were two additional icons called 'logo.png' and 'logo_ifish.png' in this directory both of which mentioned the word 'iFish':

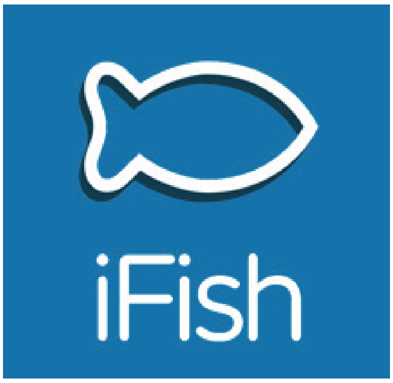
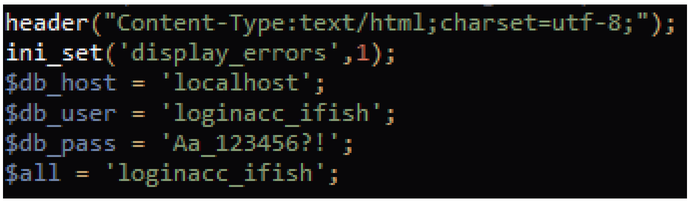


Figure 29: iFish logo under the 'dist' directory.

The word 'iFish' appeared before in one of the credential pairs used to access a database in maillogin[.]live, and might refer to the internal name of the project:



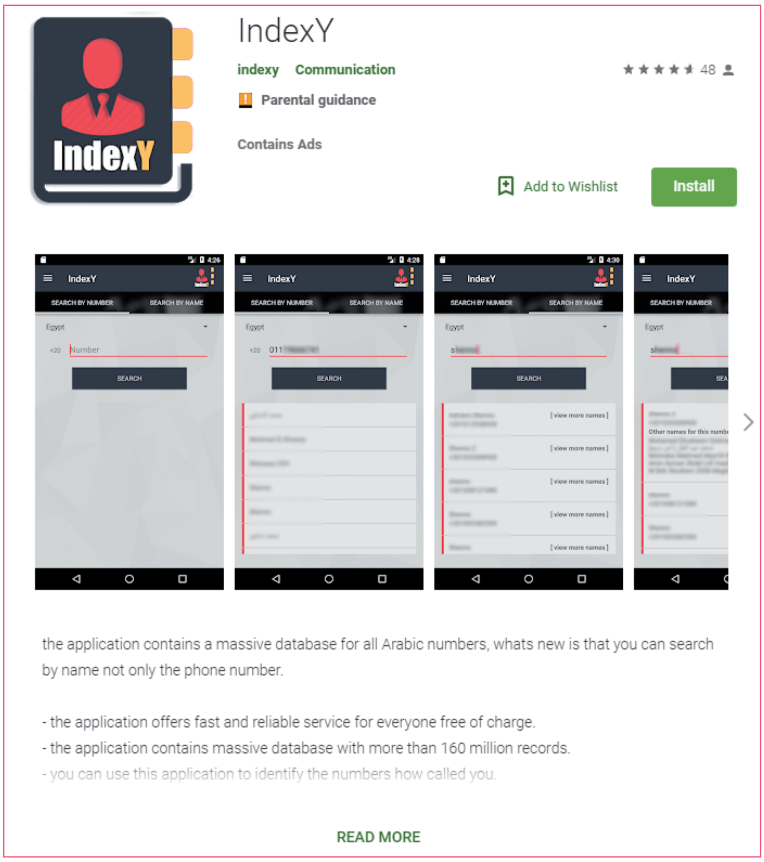
```
header("Content-Type:text/html;charset=utf-8;");
ini_set('display_errors',1);
$db_host = 'localhost';
$db_user = 'loginacc_ifish';
$db_pass = 'Aa_123456?!';
$all = 'loginacc_ifish';
```

Figure 30: Previous iFish connection.

This was yet another connection that we could establish between the *Android* application and the phishing attacks based on the findings from indexy[.]org. As it turns out, this website had other ties to the world of mobile, besides the location-tracking applications.

INDEXY

The website indexy[.]org appeared in an unlikely location: an *Android* application called 'IndexY' with more than 5,000 downloads, found in *Google*'s official *Play Store*.



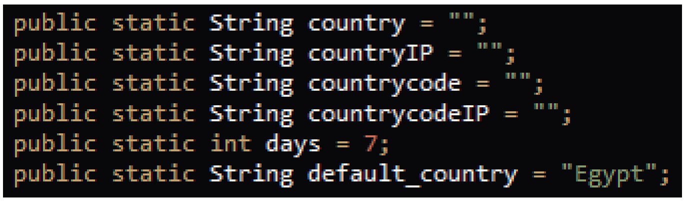
the application contains a massive database for all Arabic numbers, whats new is that you can search by name not only the phone number.

- the application offers fast and reliable service for everyone free of charge.
- the application contains massive database with more than 160 million records.
- you can use this application to identify the numbers how called you.

Figure 31: IndexY application on Google Play.

IndexY offers a service that is similar to the well known *TrueCaller* app, allowing users to look up details about phone numbers or their owners. It promises a large database of more than 160 million phone numbers, but seems to be intended for a specific target audience according to its description: Arabic-speaking users.

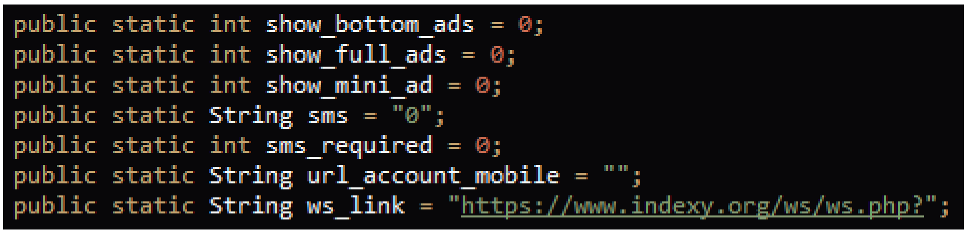
'Index Masr' is mentioned in the package name, which means 'Index Egypt'. In addition, the default country for IndexY users shows that the service is mainly aimed at Egyptians:



```
public static String country = "";
public static String countryIP = "";
public static String countrycode = "";
public static String countrycodeIP = "";
public static int days = 7;
public static String default_country = "Egypt";
```

Figure 32: Embedded location configuration.

The variable 'ws_link' is used to set the website address IndexY communicates with:

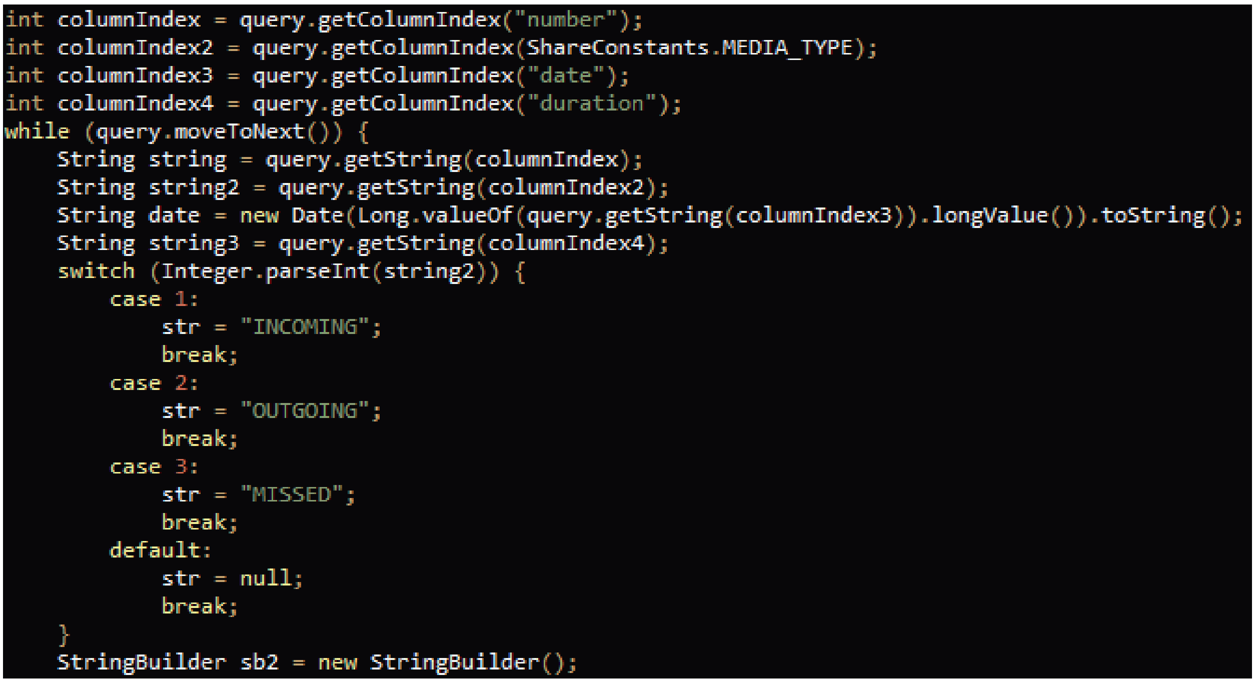


```
public static int show_bottom_ads = 0;
public static int show_full_ads = 0;
public static int show_mini_ad = 0;
public static String sms = "0";
public static int sms_required = 0;
public static String url_account_mobile = "";
public static String ws_link = "https://www.indexy.org/ws/ws.php?";
```

Figure 33: Recurring 'ws_link' variable name.

When the application is installed, it receives access to the user's contacts and call history. While this is considered sensitive data, it would make sense for an application of this nature to try and collect as many phone numbers as possible to enhance the service it offers.

But instead of only exporting phone numbers from the call history, the application logs the direction of each call (incoming, outgoing or missed), the date on which it was received and its duration.



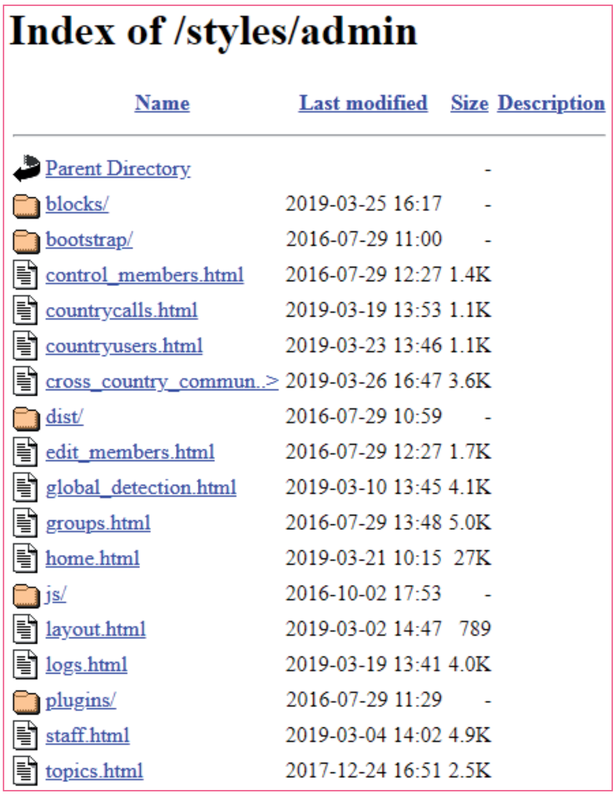
```
int columnIndex = query.getColumnIndex("number");
int columnIndex2 = query.getColumnIndex(ShareConstants.MEDIA_TYPE);
int columnIndex3 = query.getColumnIndex("date");
int columnIndex4 = query.getColumnIndex("duration");
while (query.moveToNext()) {
    String string = query.getString(columnIndex);
    String string2 = query.getString(columnIndex2);
    String date = new Date(Long.valueOf(query.getString(columnIndex3)).longValue()).toString();
    String string3 = query.getString(columnIndex4);
    switch (Integer.parseInt(string2)) {
        case 1:
            str = "INCOMING";
            break;
        case 2:
            str = "OUTGOING";
            break;
        case 3:
            str = "MISSED";
            break;
        default:
            str = null;
            break;
    }
    StringBuilder sb2 = new StringBuilder();
```

Figure 34: Call information recorded by the application.

Thus, IndexY does not seem to be accessing this data just to improve the service it promises and is actually exporting much more information than it needs to.

STYLES ONCE MORE

There was other supporting evidence on indexy[.]org that confirmed our suspicions. Similarly to the previous case, there was an administration panel to manage this application. Once again, we were able to view the HTML templates for the pages the administrators get after they log in, using the 'styles' directory:



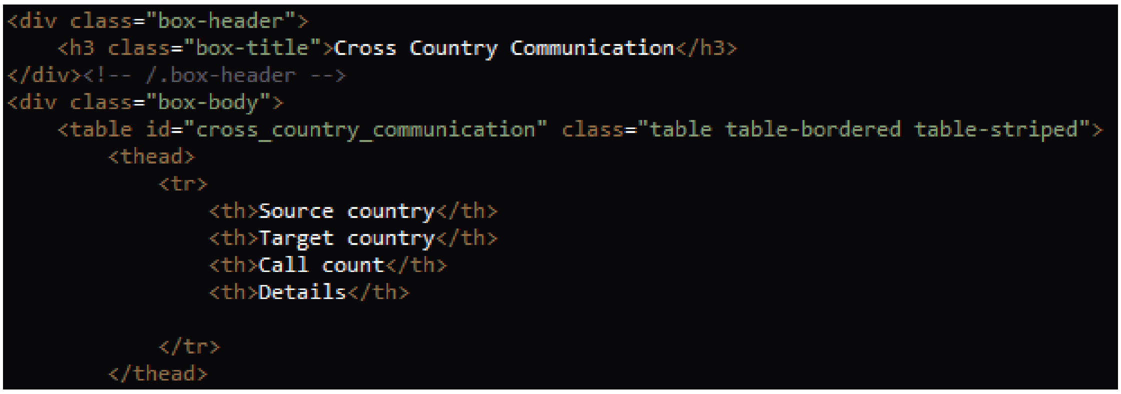
Index of /styles/admin

Name	Last modified	Size	Description
Parent Directory		-	
blocks/	2019-03-25 16:17	-	
bootstrap/	2016-07-29 11:00	-	
control_members.html	2016-07-29 12:27	1.4K	
countrycalls.html	2019-03-19 13:53	1.1K	
countryusers.html	2019-03-23 13:46	1.1K	
cross_country_commun..>	2019-03-26 16:47	3.6K	
dist/	2016-07-29 10:59	-	
edit_members.html	2016-07-29 12:27	1.7K	
global_detection.html	2019-03-10 13:45	4.1K	
groups.html	2016-07-29 13:48	5.0K	
home.html	2019-03-21 10:15	27K	
js/	2016-10-02 17:53	-	
layout.html	2019-03-02 14:47	789	
logs.html	2019-03-19 13:41	4.0K	
plugins/	2016-07-29 11:29	-	
staff.html	2019-03-04 14:02	4.9K	
topics.html	2017-12-24 16:51	2.5K	

Figure 35: Folder view of indexy[.]org.

The directory included HTML pages that were uploaded during 2016, and two of them were even identical to the ones that were found and reused in the location-tracking panel.

The pages show that the administrators store and inspect the information they collect from the users. This involves lists of the number of users per country, detailed call logs, and even lists of calls that were made by users from one country to another.



```
<div class="box-header">
    <h3 class="box-title">Cross Country Communication</h3>
</div><!-- /.box-header -->
<div class="box-body">
    <table id="cross_country_communication" class="table table-bordered table-striped">
        <thead>
            <tr>
                <th>Source country</th>
                <th>Target country</th>
                <th>Call count</th>
                <th>Details</th>

            </tr>
        </thead>
```

Figure 36: Code to display collected information.

Based on this, it seems that whoever is responsible for IndexY is abusing the access they have gained. They do not simply store the data that the application collects, but are rather analysing it carefully, and even looking for suspicious activity within it, far beyond the scope of an innocent *TrueCaller*-like service.

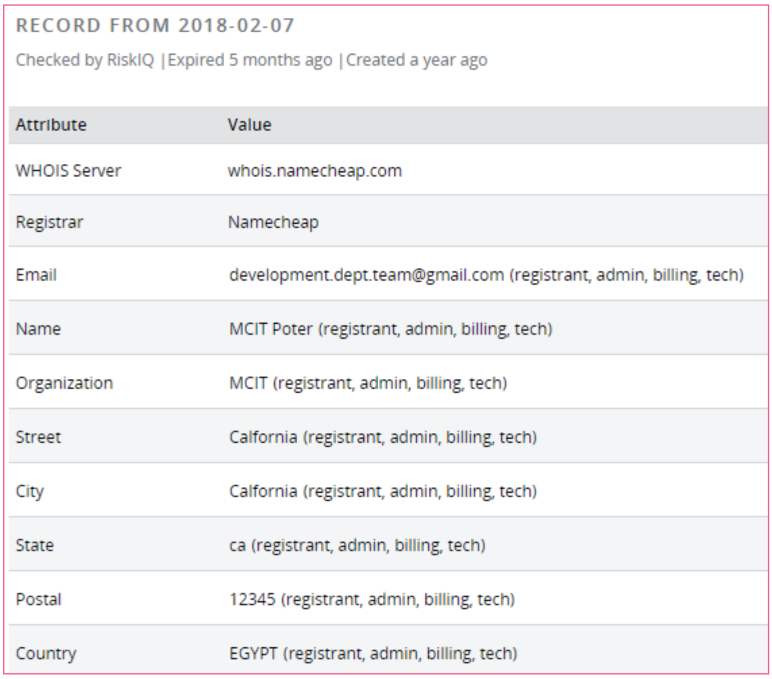
While we considered the possibility that IndexY is not connected to the phishing attacks, the more we looked into both things the more probable it became that they belong to the same operation, and are motivated by the same purpose: surveillance of Egyptians.

BACK TO EGYPT

We came across several clues during our investigation of the different components and layers that make up this attack that could suggest it was coming from an Egyptian source or someone who invested much effort in planting false flags in an extremely sophisticated way. We will divide those findings into three categories.

First: Amnesty International IoCs

We already mentioned that the domains in *Amnesty*'s report shared similar characteristics, but only one of them (account-login[.]site) had a WHOIS record with details about the registrant. According to this record, the registrant was from Egypt:



RECORD FROM 2018-02-07

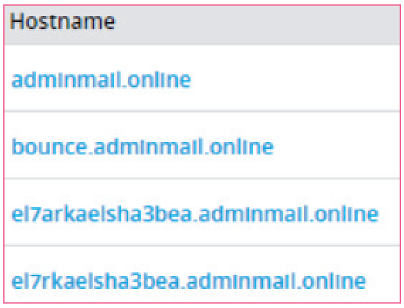
Checked by RiskIQ | Expired 5 months ago | Created a year ago

Attribute	Value
WHOIS Server	whois.namecheap.com
Registrar	Namecheap
Email	development.dept.team@gmail.com (registrant, admin, billing, tech)
Name	MCIT Poter (registrant, admin, billing, tech)
Organization	MCIT (registrant, admin, billing, tech)
Street	Calfornia (registrant, admin, billing, tech)
City	Calfornia (registrant, admin, billing, tech)
State	ca (registrant, admin, billing, tech)
Postal	12345 (registrant, admin, billing, tech)
Country	EGYPT (registrant, admin, billing, tech)

Figure 37: WHOIS information for account-login[.]site.

Moreover, the registrant’s name and organization mention ‘MCIT’, which might be an acronym for the Ministry of Communications and Information Technology in Egypt.

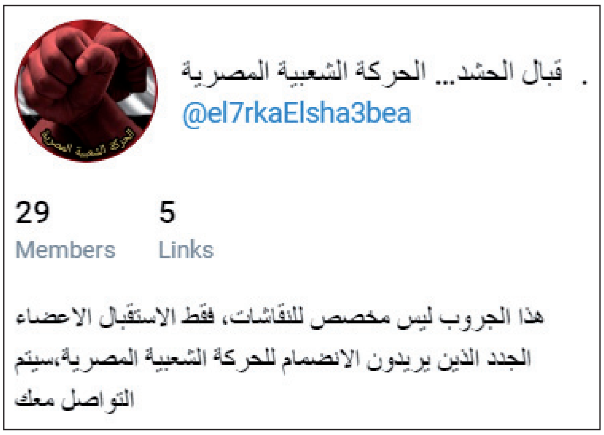
Another website from the IoC list, adminmail[.]online, had some interesting subdomains:



Hostname
adminmail.online
bounce.adminmail.online
el7arkaelsha3bea.adminmail.online
el7rkaelsha3bea.adminmail.online

Figure 38: Subdomain records for adminmail[.]online.

‘el7arkaelsha3bea’ and ‘el7rkaelsha3bea’ are the English transliterations of the Arabic phrase ‘الحركة الشعبية’, which means ‘The Popular Movement’. A domain that mentions this phrase (el7rkaelsha3bea.ddns[.]net) also resolved to an IP address that is related to this attack. Looking up this phrase online led us to one result only, a *Telegram* channel with the same name:



. قبال الحشد... الحركة الشعبية المصرية
@el7rkaElsha3bea

29
Members

5
Links

هذا الجروب ليس مخصص للنقاشات، فقط الاستقبال الاعضاء
الجدد الذين يريدون الانضمام للحركة الشعبية المصرية،سيتم
التواصل معك

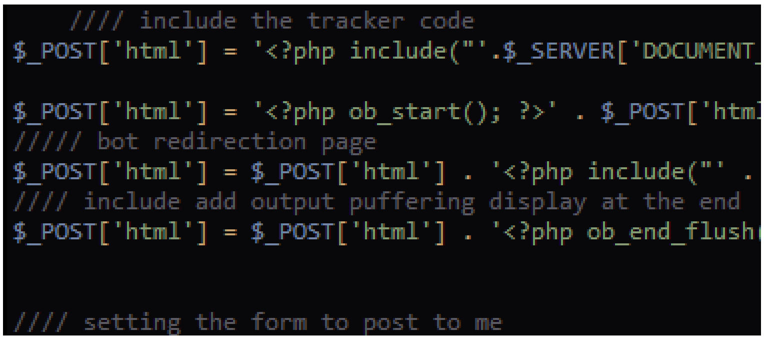
Figure 39: Telegram group ‘el7arkaelsha3bea’.

The channel was created in March, and seems to be intended for activists looking to join ‘a social movement that is opposed to the regime in Egypt’. It has several links to a *Facebook* page calling for a second revolution, and asks new members to contact the admins and identify themselves.

The channel itself might be used to track Egyptian dissidents, although we have no conclusive evidence of the real intentions behind it, nor can we confirm that it is related to the subdomains we have observed, as they were inactive by the time we checked them.

Second: the open directory

Scripts that we downloaded from maillogin[.]live were heavily documented, and included comments that explained the purpose of almost each section or branch of the code:

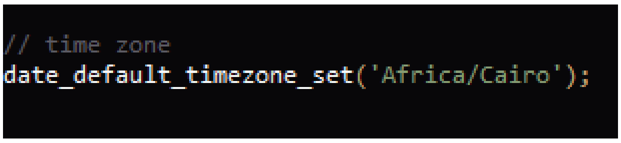


```
    //// include the tracker code
$_POST['html'] = '<?php include("".$_SERVER['DOCUMENT_

$_POST['html'] = '<?php ob_start(); ?>' . $_POST['htm
///// bot redirection page
$_POST['html'] = $_POST['html'] . '<?php include("' .
//// include add output puffering display at the end
$_POST['html'] = $_POST['html'] . '<?php ob_end_flush(


//// setting the form to post to me
```

Figure 40: Extensive code comments.

The comments contained many grammatical mistakes and misspelled words, and, surprisingly, they might even give away the origin of the attackers. In one of the comments, the word 'Puffering' is used instead of 'Buffering'. The two words would be phonetically similar to a native Arabic speaker, since the Arabic alphabet does not have a 'P' sound.

This can be confirmed by another configuration file, which sets the default time zone of the server to that of Cairo.

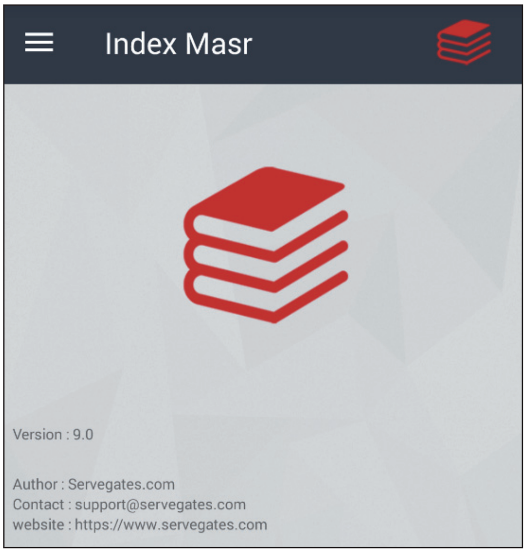


```
// time zone
date_default_timezone_set('Africa/Cairo');
```

Figure 41: Predefined default time zone in the code.

Third: IndexY developer

The latest version number of IndexY in *Google*'s *Play Store* was 11.08, but we were able to obtain 10 older variants that were published during 2018. The application did not always communicate with indexy[.]org, and it used arabindex[.]info or indexmasr[.]com instead in the previous generations. In addition, one of the versions mentions a fourth website in its 'About' section: servegates[.]com.

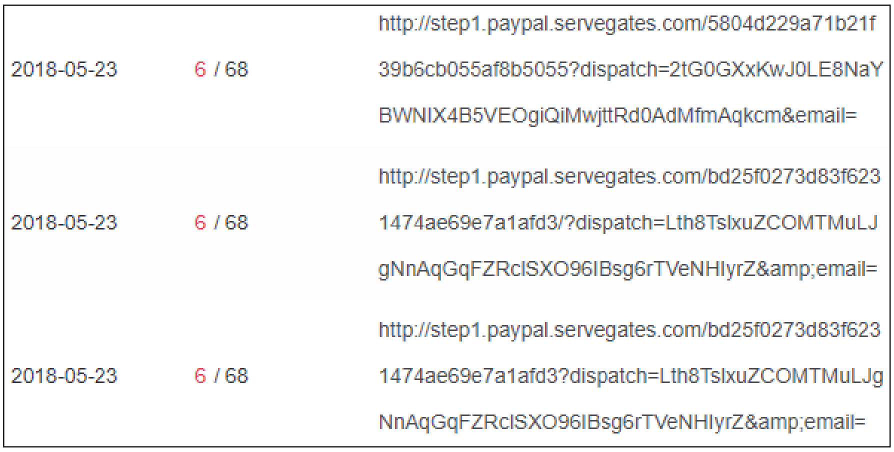


Version : 9.0

Author : Servegates.com
Contact : support@servegates.com
website : https://www.servegates.com

Figure 42: Older IndexY application.

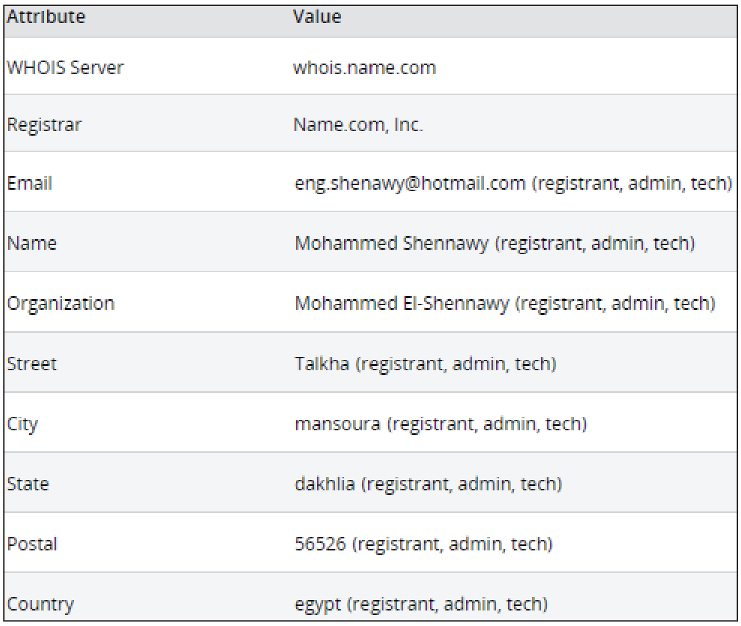
All of the above websites resolved to the same IP address, and although it was not connected to *MAROSNET* or to the netblock that was constantly used in the phishing attacks, we could see, for example, that servergates[.]com hosted phishing URLs in the past:



2018-05-23	6 / 68	http://step1.paypal.servegates.com/5804d229a71b21f39b6cb055af8b5055?dispatch=2tG0GXxKwJ0LE8NaYBWNIX4B5VEOgiQiMwjttRd0AdMfmAqkcm&email=
2018-05-23	6 / 68	http://step1.paypal.servegates.com/bd25f0273d83f6231474ae69e7a1afd3/?dispatch=Lth8TslxuZCOMTMuLJgNnAqGqFZRclSXO96IBsg6rTVeNHIyrZ&amp;email=
2018-05-23	6 / 68	http://step1.paypal.servegates.com/bd25f0273d83f6231474ae69e7a1afd3?dispatch=Lth8TslxuZCOMTMuLJgNnAqGqFZRclSXO96IBsg6rTVeNHIyrZ&amp;email=

Figure 43: Netblock previously used for phishing attacks.

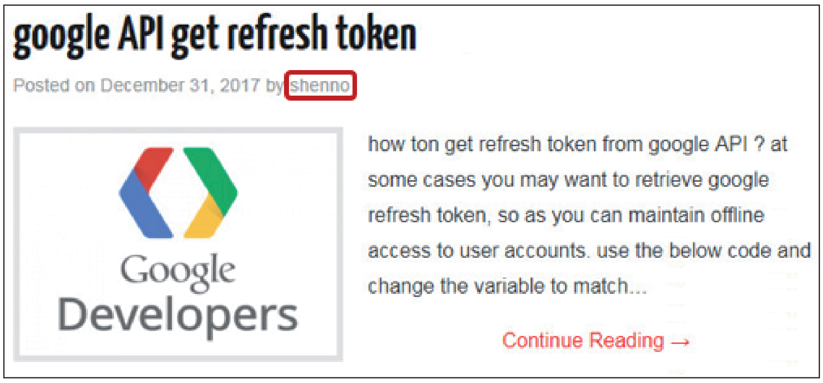
The WHOIS records of servegates[.]com and indexmasr[.]com show that they were both registered using the email address eng.shenawy@hotmail[.]com, which supposedly belongs to an individual from Egypt.



Attribute	Value
WHOIS Server	whois.name.com
Registrar	Name.com, Inc.
Email	eng.shenawy@hotmail.com (registrant, admin, tech)
Name	Mohammed Shennawy (registrant, admin, tech)
Organization	Mohammed El-Shennawy (registrant, admin, tech)
Street	Talkha (registrant, admin, tech)
City	mansoura (registrant, admin, tech)
State	dakhlia (registrant, admin, tech)
Postal	56526 (registrant, admin, tech)
Country	egypt (registrant, admin, tech)

Figure 44: WHOIS information for indexmasr[.]com.

A website with the same WHOIS information (txtips[.]com), which publishes blog posts with technical tips, uses 'shenno' as the name of the author.



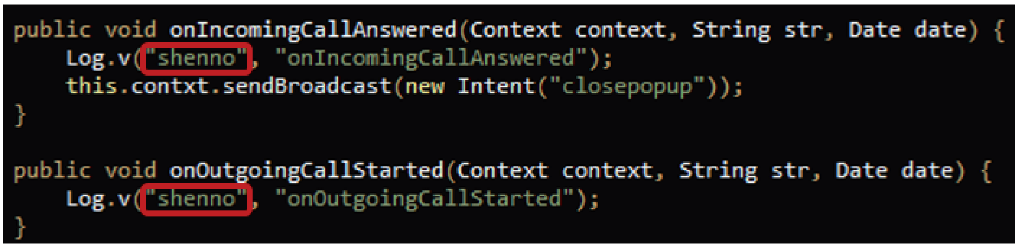
google API get refresh token

Posted on December 31, 2017 by shenno

how ton get refresh token from google API ? at some cases you may want to retrieve google refresh token, so as you can maintain offline access to user accounts. use the below code and change the variable to match...

Figure 45: Post published by 'shenno' on a connected website.

This name appeared in the source code of the IndexY application, and it was used as a tag for all the displayed log messages:



```
public void onIncomingCallAnswered(Context context, String str, Date date) {
    Log.v("shenno", "onIncomingCallAnswered");
    this.contxt.sendBroadcast(new Intent("closepopup"));
}

public void onOutgoingCallStarted(Context context, String str, Date date) {
    Log.v("shenno", "onOutgoingCallStarted");
}
```

Figure 46: Previous connections to 'shenno' observed in the IndexY application.

'Shenno' may be a nickname or an abbreviation of the name in the WHOIS record, 'Shennawy'. Although this might be a fake name, it led us to believe that whoever was in charge of registering the websites was the same person as was in charge of developing the IndexY application, and appears to be from Egypt.

CONCLUSION

Following up on the investigation first conducted by *Amnesty International*, we revealed new aspects of the attack that has been after Egypt's civil society since at least 2018. We discovered a list of victims that included hand-picked political and social activists, high-profile journalists and members of non-profit organizations in Egypt.

Whether it is phishing pages, legitimate-looking applications for *Outlook* and *Gmail*, or mobile applications to track a device's communications or location, it is clear that the attackers are constantly coming up with creative and versatile methods to reach victims, spy on their accounts, and monitor their activity.

The information we gathered from our investigation suggested that the perpetrators are Arabic speakers and well familiar with the Egyptian ecosystem. Because the attack might be government-backed, it means that we are looking at what could be a surveillance operation of a country against its own citizens.

REFERENCES

[1] Phishing attacks using third-party applications against Egyptian civil society organizations. Amnesty International. March 2019. https://www.amnesty.org/en/latest/research/2019/03/phishing-attacks-using-third-party-applications-against-egyptian-civil-society-organizations/.

[2] 2019-03-06_egypt_oauth. https://github.com/AmnestyTech/investigations/tree/master/2019-03-06_egypt_oauth.

[3] https://urlscan.io/result/f6241609-6dac-483f-8fbb-851388477aec/.