



SECURITY RESEARCH REPORT

Stealth Mango & Tangelo

Selling your fruits to nation state actors

Contents

| | |
|---------------------------------------|-----------|
| Executive summary | 1 |
| Key findings | 2 |
| Threat summary | 3 |
| Exfiltrated content | 4 |
| Noteworthy exfiltrated content | 5 |
| Airport surveillance | 9 |
| Victim analysis | 9 |
| Infection vectors | 11 |
| Stealth Mango functionality (Android) | 12 |
| Tangelo functionality (iOS) | 15 |
| Threat actor activity | 16 |
| Associated Stealth Mango developers | 16 |
| Infrastructure | 20 |
| Conclusions | 24 |
| Appendix | 25 |
| Indicators of Compromise | 25 |
| Servers and IPs | 25 |
| APK hashes | 26 |
| APK package names and app names | 27 |
| iOS hashes | 27 |
| iOS Bundle ID | 27 |
| Miscellaneous | 27 |

Executive summary

Lookout Security Intelligence has discovered a set of custom Android and iOS surveillanceware tools we're respectively calling Stealth Mango and Tangelo. These tools have been part of a highly targeted intelligence gathering campaign we believe is operated by members of the Pakistani military. Our investigation indicates this actor has used these surveillanceware tools to successfully compromise the mobile devices of government officials, members of the military, medical professionals, and civilians. To date, we have observed Stealth Mango being deployed against victims in Pakistan, Afghanistan, India, Iraq, Iran, and the United Arab Emirates. The surveillanceware also retrieved sensitive data from individuals and groups in the United States, Australia, and the United Kingdom. These individuals and groups were not themselves targeted, but interacted with individuals whose devices had been compromised by Stealth Mango or Tangelo. We believe that the threat actor behind Stealth Mango is also behind [Op C Major](#) and [Transparent Tribe](#).

At present, we have analyzed over 15 gigabytes of data taken from compromised devices, the majority of which is information that would be relevant to a nation state actor performing espionage activities. Content includes:

- Letters and internal government communications
- Detailed travel information
- Pictures of IDs and passports
- GPS coordinates of pictures and devices
- Legal and medical documents
- Developer information including whiteboard sessions, account information, and test devices
- Photos of the military, government, and related officials from closed door meetings including U.S. Army personnel

The server infrastructure supporting this operation had several operational security shortcomings that allowed Lookout to retrieve compromised content. This included the presence of the web shell software WSO 2.5 hackers commonly use to maintain remote access to a server. It's unclear whether WSO was installed by a third-party actor that successfully compromised this site or if the operators behind Stealth Mango installed and configured it themselves.

As was the case with previous actors we've reported on, [such as Dark Caracal](#), the actor behind Stealth Mango has stolen a significant amount of sensitive data from compromised devices without the need to resort to exploits of any kind. The actors rely on social engineering to infect target devices and have previously hosted Stealth Mango samples on their own fake app store. Analysis of exfiltrated data suggests that physical access may also play a role.

We have also identified, as part of this investigation, several individuals who we believe are responsible for the development of other commodity Android spyware tools that share many similarities to Stealth Mango. These individuals all belong to the same freelance developer group for hire, which says it has a physical presence in India, Pakistan, and the United States.

Currently, it is unclear when Stealth Mango was first deployed, however it is being actively developed with the latest release as recently as April 2018. Additionally, we discovered an iOS component named Tangelo that may be responsible for the exfiltrated iOS data we observed, but we are unsure if it has been used directly in this campaign or if it is still in development and testing.

The Lookout Security Intelligence team alerted Google to the existence of Stealth Mango during our investigation. The company states: "Google identified the apps associated with this actor, none of the apps were on the Google Play Store. Google Play Protect has been updated to protect user devices from these apps and is in the process of removing them from all affected devices."

Key findings

Lookout researchers have identified a new mobile malware family called Stealth Mango.

- Our research shows that Stealth Mango is being actively managed by Pakistani based actors that are likely military.
- Stealth Mango is being used in targeted surveillance operations against government officials, members of the military, and activists in Pakistan, Afghanistan, India, Iraq, and the United Arab Emirates.
- We determined that government officials and civilians from the United States, Australia, the United Kingdom, and Iran had their data indirectly compromised after they interacted with Stealth Mango victims.
- The actors behind Stealth Mango typically lure victims via phishing, but they may also have physical access to victims' devices.
- The attacker has multi-platform capabilities. We know of the Android component and there is evidence of an iOS component. The evidence is as follows:
 - A sample Debian package on attacker infrastructure called Tangelo
 - EXIF data from exfiltrated content showed data from iPhones
 - WHOIS information from the attackers show registrations for the following domains:
iphonespyingsoftware[.]org, iphonespyingapps[.]org, and iphonespyingapps[.]info

We have identified over 15 gigabytes of compromised data on attacker infrastructure.

- Exfiltrated content includes call records, audio recordings, device location information, text messages, and photos.
- We found attacker infrastructure running the WSO web shell, which provides a third party with complete control over the server.
- The actor deploying Stealth Mango appears to have a primarily mobile-focused capability.


Stealth Mango and Tangelo appear to have been created by freelance developers with physical presences in Pakistan, India, and the United States.

- These individuals belong to the same developer group.
- We linked their tooling to several commodity mobile surveillance tools suggesting that they are either sharing code or have engaged with several distinct customers who are being delivered tooling based off similar source code.

As part of this investigation Lookout is releasing numerous IOCs related to this actor and the Stealth Mango surveillanceware family.

- 38 APK hashes
- 6 domain names
- 1 iOS Debian hash
- 14 IP addresses
- 2 iOS Mach-O hashes

Threat summary

| Name | Stealth Mango |
|-------------------|--|
| Country of origin | Pakistan  |
| Threat actor | Group or individuals that are believed to belong to the Pakistani military. The actor is possibly related to Op C Major and Transparent Tribe. |
| Developer group | Believed to be a group of freelance developers that are available for hire and operate under the same developer group. At least one government employee moonlights as a mobile app developer. |
| Infection vectors | <ul style="list-style-type: none"> • Phishing links to a fake third-party Android app store • Exfil suggests physical access may play a role |
| Targets | <p>Primary targets:</p> <ul style="list-style-type: none"> • Pakistan officials & citizens • Afghanistan officials & citizens • Other regional people from Balochistan and nearby cities <p>Individuals and groups whose data was inadvertently collected after they interacted with a Stealth Mango or Tangelo victim:</p> <ul style="list-style-type: none"> • United States officials & civilians • Australian diplomats • British diplomats • NATO members • Iranian officials & civilians |
| Highlights | <ul style="list-style-type: none"> • Developers freelance on mobile apps including commodity spyware (TheOneSpy) • Server had publicly accessible exfiltrated content including: <ul style="list-style-type: none"> • Images, recordings, messages, etc. • Server may have been compromised • Development is very active • Android and iOS implants |

Exfiltrated content

The actor behind Stealth Mango has been able to retrieve a considerable amount of sensitive data from compromised devices when considering how short-running this campaign has been. To date, this investigation has identified over 15 gigabytes of exfiltrated data on command and control servers. This data includes text messages, contact details, package info, geolocation data, audio recordings, photos, and videos from both victim and test devices. We believe Stealth Mango developers regularly release new versions of their tooling, with the most recent version dating April 2018. Throughout this period they've continued to retrieve data from compromised devices.

Analysis of exfiltrated data found it to be comprised of personal content but also more sensitive information that included:

- A letter from the High Commission for Pakistan to the United States Director of the Foreign Security Office Ministry of Foreign Affairs.
- A letter from the United States Central Command to the Afghanistan Assistant Minister of Defense for Intelligence inviting the Assistant Minister to the Annual Central and South Asia Directors of Military Intelligence Conference that takes place in Tampa, Florida in February 2018.
- A letter to the Pakistani Chief of Army Staff.
- Documents with the letterhead for the Strategic Intelligence Deputy, Islamic Republic of Afghanistan.
- Details of visits to Quetta, Balochistan, Pakistan by German Diplomats.
- Details of visits to Quetta, Balochistan, Pakistan by Australian Diplomats.
- Official Government of Pakistan notices that use the Confidential classification marking and relate to an internal corruption investigation.
- A user entering their personal information into the Federal Public Service Commision website of Pakistan and listing their profession as "Inspector customs / Intelligence Officer."
- Photos of Afghan and Pakistani military officials.
- Content for the Regional Directorate Anti Narcotics Force Islamabad.
- Passport photos.
- Photos of ID cards.
- Photos of what we believe are from devices belonging to the developer.

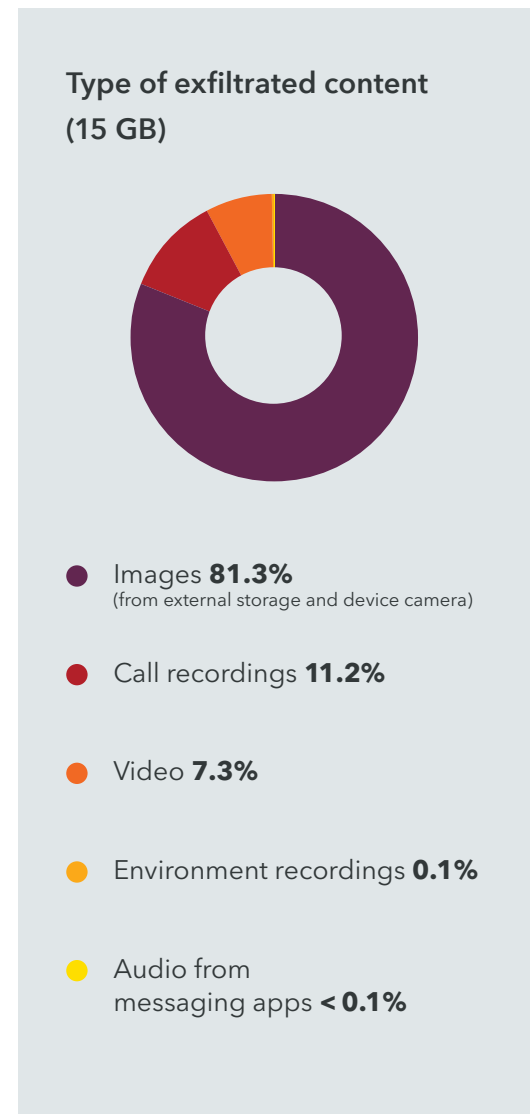


Figure 1: Breakdown of the media types of exfiltrated content.

Noteworthy exfiltrated content

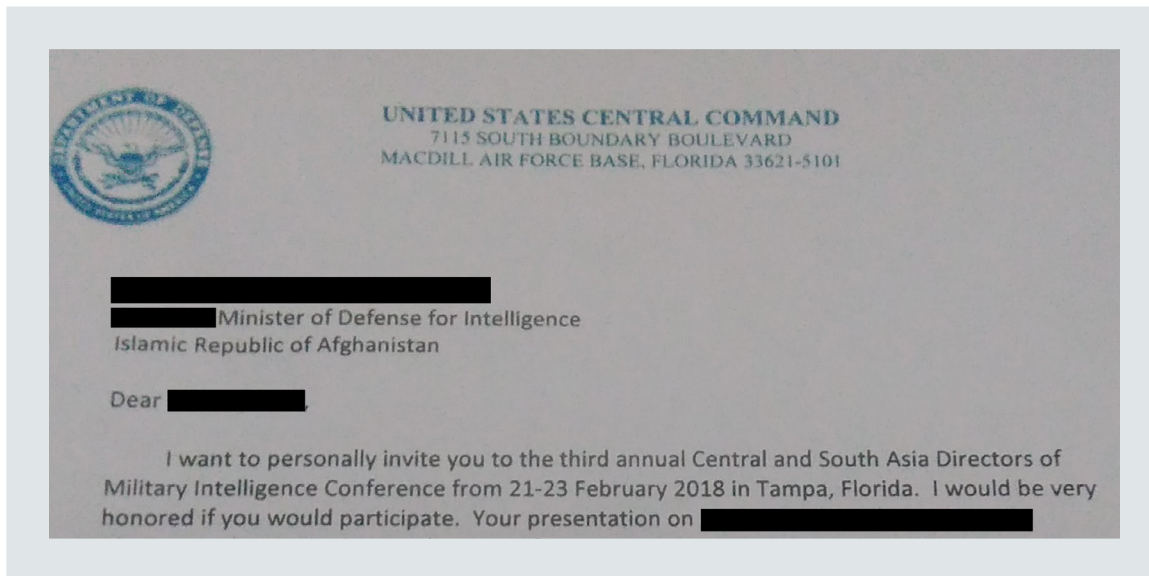


Figure 2: A reduced snippet of the original photo taken of exfiltrated image from the U.S. Central Command Afghan Assistant Minister of Defense.

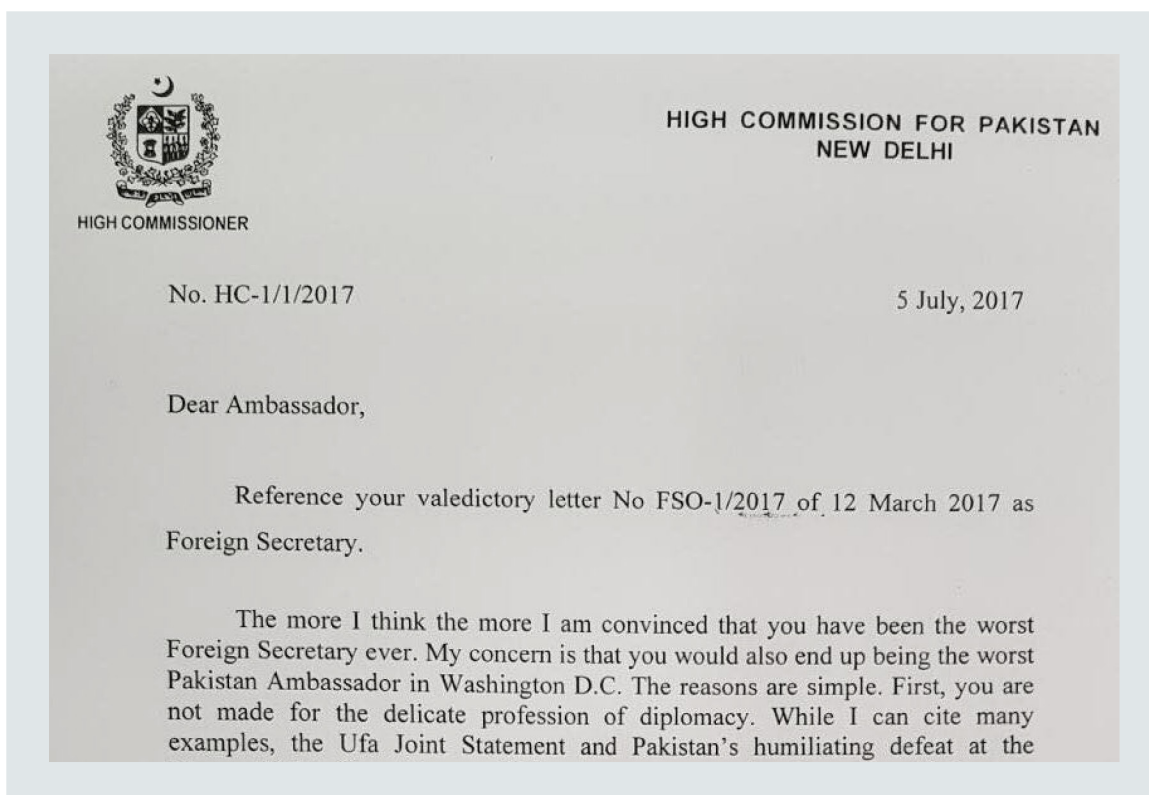


Figure 3: A snippet of a scathing letter from the Pakistan High Commission to the United States Ambassador, as previously made public in [numerous news publications](#).

Subject: - Visit of Australian Diplomats to

This is to inform that the following diplomats of Australian High Commission in Islamabad would be visiting. Contact number of the Mission is . The details of the visit are as under:

| Name & Designation of Member(s) of Mission | Date (s) of Visit (s) | Place (s) to be visited and mode of Travel |
|--|-----------------------|--|
| (accompanied by 01 person) | (03 Days) | Visit to: By Air Purpose of visit: |

Total Person(s) (-02-)

2. It is requested that full proof security arrangements may be made during the visit of the above mentioned Diplomat/Official to

Most Immediate

Government of Pakistan
Ministry of Foreign Affairs

Subject: - Visit of the German diplomat to

The Ministry of Foreign Affairs, Islamabad has informed vide note No.

| Name & Designation of Member(s) of Mission | Date (s) of Visit (s) | Place (s) to be visited and mode of Travel |
|--|-----------------------|--|
| | (for 3 days only) | |

Total Person(s)-02-

2. It is requested that a fool proof security arrangements may be made during

Figure 4: Details around travel in and around Pakistan from both Australian and German diplomats.

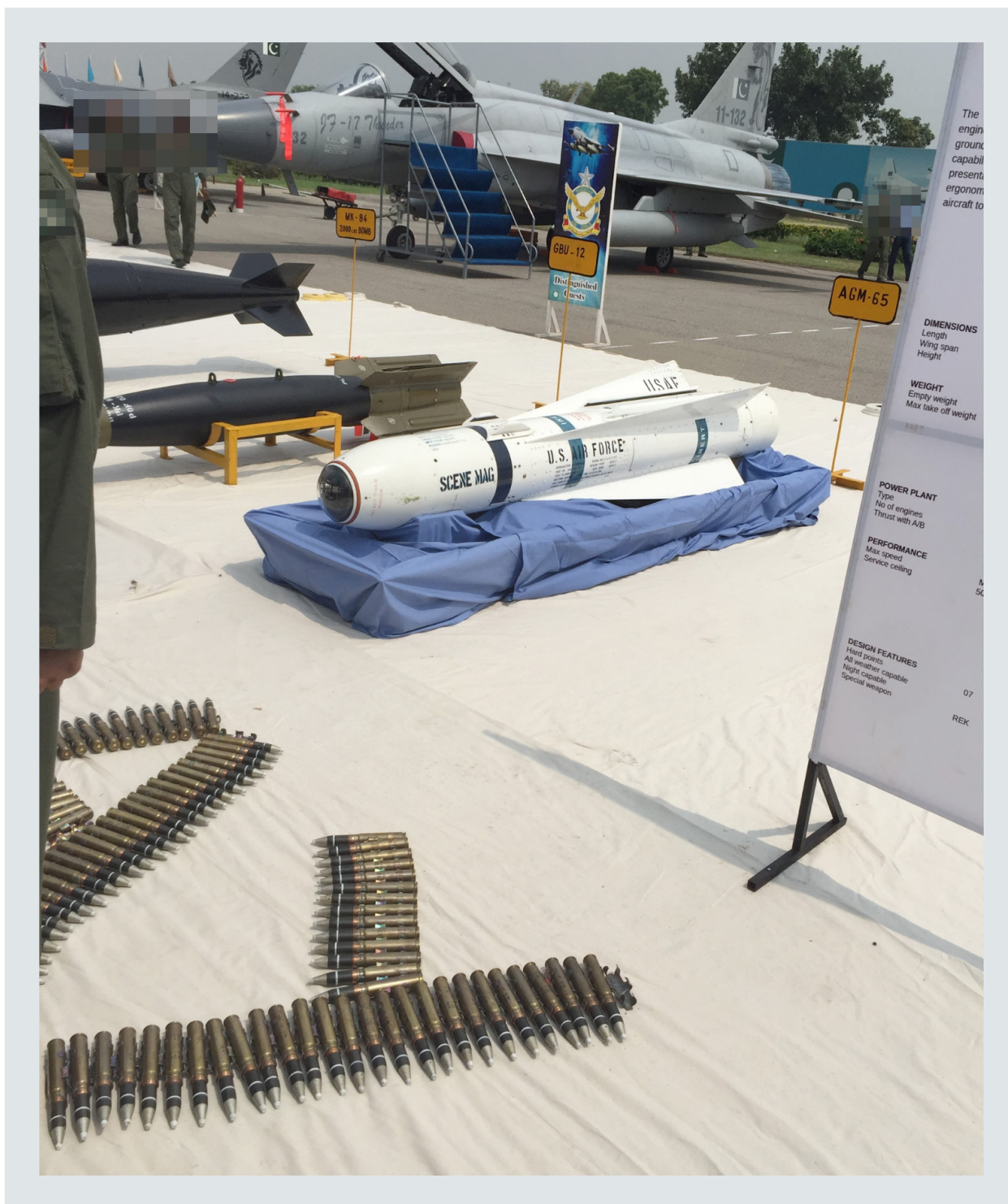


Figure 5: Pictures from what appeared to be a weapons show included images of U.S. military hardware such as the United States Air Force Scene Mag.



Figure 6: Exfiltrated content was found to contain military photos including a series of images from an event with military attendees from numerous countries including U.S. Army personnel.



Figure 7: Many images of members belonging to the Afghan military were also discovered among the content taken from compromised devices.

Airport surveillance

Stealth Mango has collected data from a number of different locations. In particular we discovered exfiltrated photos of passports and diplomatic IDs with GPS coordinates from the Kandahar airport in Afghanistan. This indicates either someone is actively exfiltrating this data knowingly to the C2 server or that Stealth Mango has infected someone's device that works at the airport unbeknownst to them.

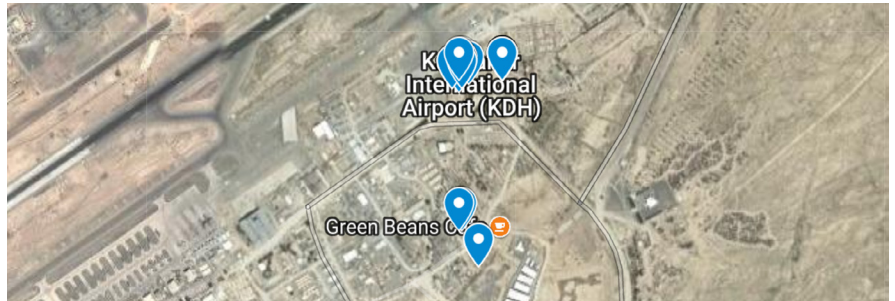


Figure 8: EXIF GPS locations of exfiltrated photos of diplomatic IDs and passports from/near the Kandahar airport.

Victim analysis

The attacker infrastructure had several operational security shortcomings that included the presence of a WSO web shell. This allowed us to build out a comprehensive picture of this actor's success, what they had been able to steal, and the groups they were targeting. While the presence of this web shell allowed for the retrieval of usernames and passwords for databases and various services, we deemed the use of these credentials as falling outside the Computer Fraud and Abuse Act (CFAA). Consequently, we only retrieved and performed analysis on files that were publicly accessible and did not connect or log in to services requiring authentication.

With these limitations in mind, we identified victims as likely being a mix of Pakistani government and military officials who had access to sensitive material. Exfiltrated content also suggests a smaller subset of victims reside in India, Iraq, Afghanistan, and the United Arab Emirates. We further identified content from other countries' officials and diplomats, including the United States, Australia, the United Kingdom, and Iran, however we believe this data may have been stolen when these individuals interacted with Stealth Mango victims.

Many of the images found on attacker servers did not contain latitude and longitude metadata however, for those that did we were able to identify where some of the compromised images were taken.

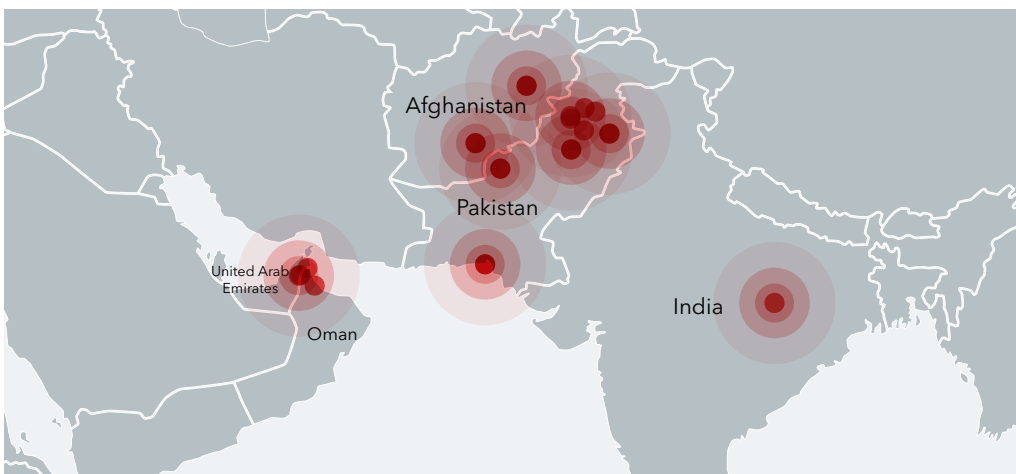


Figure 9: GPS coordinates pulled from the EXIF data of exfiltrated images is centered around Pakistan, Afghanistan, India, and the United Arab Emirates.

Further analysis on the EXIF data of images provided some useful insights, including the identification of a surprising amount of iPhone metadata. While we haven't found evidence of an iOS component installed on infected devices, we did discover the existence of an iOS implant called Tangelo that was developed by these actors. This further backs up our theory that the attackers have multi-platform capabilities.

It's possible that the images containing iOS metadata were sent from uninfected iPhones to compromised Android devices and subsequently uploaded to attacker infrastructure. That said, the EXIF data pointed to a range of iOS devices and WHOIS information for this attacker that showed them registering domains such as `iphonespyingsoftware[.]org`, `iphonespyingapps[.]org`, and `iphonespyingapps[.]info`. This suggests the intention to create an iOS capability. The existence of Tangelo discovered on infrastructure owned by the developers further backs up our theories that there is an iOS component.

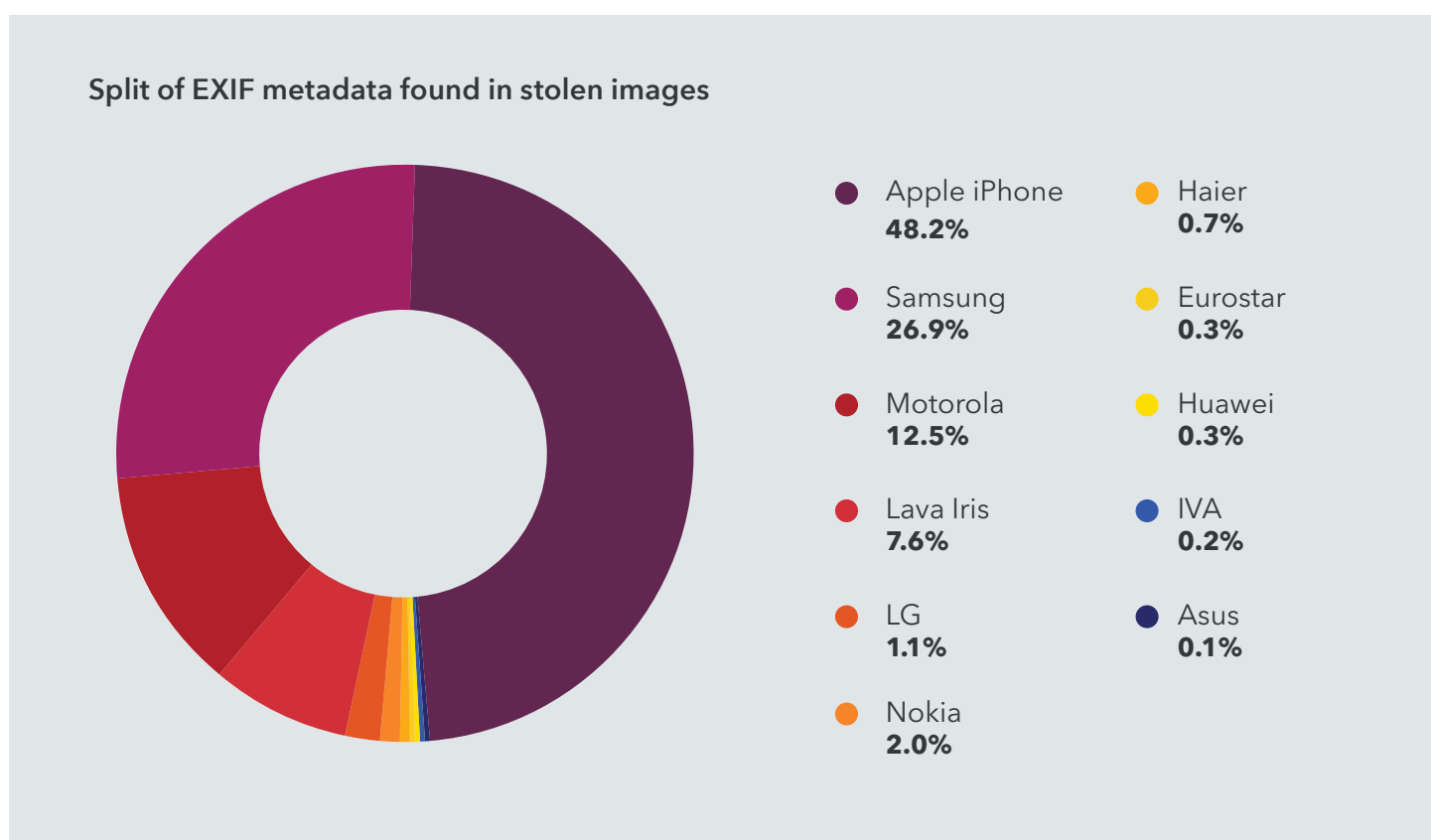


Figure 10: Analysis of the EXIF metadata contained in stolen images found that many contained information identifying the phone's make and model on which they were taken. While this doesn't definitely mean victims were using these makes and models, it is interesting to note that the majority are from iPhones.

Infection vectors

As we have seen by other threat actors in the past, the mechanism for infecting a device involves what we believe to be primarily phishing with the possibility of physical access. From examining the exfiltrated data, we have been able to pinpoint at least one of the watering holes at `secure-apps.azurewebsites.net`. The watering hole, used to distribute the malware, pretends to be the third-party Android app store APKMonk, but it is not. All links on the site to other apps will fail or re-direct to the Stealth Mango APK. The information about the APK – including the package name, version info, and past versions – is all fake information created to get the user to download the app.

The figure below shows what the watering hole looks like. The actors are using `azurewebsites.net` hosted by Microsoft to serve the malicious page. We initiated a takedown with Microsoft on the watering hole. The actors complied with the removal request from Microsoft, but then uploaded a newer version of the malware. The account was ultimately suspended.

The watering hole URL could have been sent to targets in a phishing attack. In at least one case it was distributed via Facebook Messenger suggesting the attackers are using fake personas to connect with their targets and coerce them into installing the malware onto their devices.

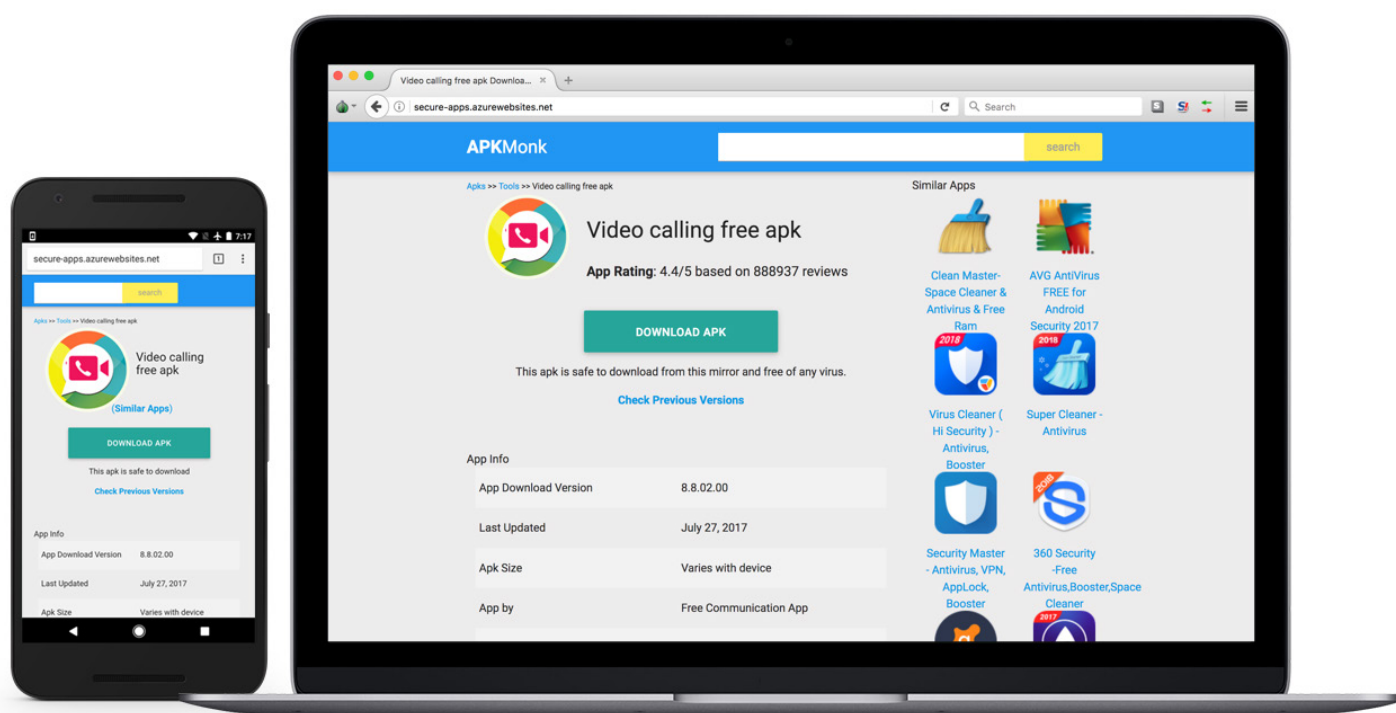


Figure 11: The watering hole as seen from desktop and mobile browsers.

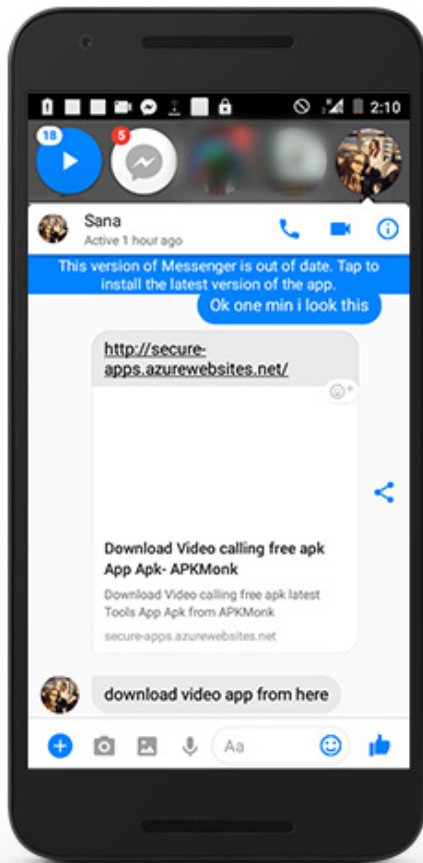


Figure 12: A screenshot of an infected Android device that shows the watering hole and the phishing attempt via Facebook Messenger.

We also believe that physical access to some devices may have also been used as an infection vector. When examining the exfiltrated data we came across a packing slip to a possible phone repair shop that contains the name of one of the people in the exfiltrated data.



Figure 13: A shipping slip used to send a cellphone to a possible repair shop. The sender of the phone is a person in the exfiltrated data that communicated with a U.S. contact, possibly indicating malware was installed via physical access to the device.

Stealth Mango functionality (Android)

Stealth Mango's surveillance capabilities are spread out over a number of sub-packages within the application and are separated into modules depending on the type of data they collect. The developer appears to be experienced in software development as abstraction of data models and other object-oriented techniques are used to develop structured communication with command and control infrastructure as well as database design.

All Stealth Mango samples launch their services at the highest priority possible and contain at least two background services which initially upload all data from an infected device and then track all changes that occur as soon as they happen. The malware also has categories for each type of information that, in later variants, is used as a model to create databases. These categories are also used to upload data into similarly named folders on command and control infrastructure. These are described below:

- CR - Call Records
- GL - Gallery
- SR - Surrounding recordings
- VD - Video
- AU - Audio

On a device infected with Stealth Mango, the following data is uploaded and tracked:

- Heartbeats to server infrastructure every 10 seconds.
- Installed packages and device information.
- Changes in SIM card or phone numbers on the device.
- Picture, video and audio files stored on the device.
- Calendar events and reminders.
- Contact lists for various third-party applications such as Yahoo, AIM, GoogleTalk, ICQ, Jabber, MSN, NetMeeting, QQ, and Skype.
- Call logs as well as recording incoming and outgoing calls. This includes blocking calls from phone numbers in a server-provided "block list" and then deleting them from the call logs of the user.
- SMS logs and deleting incoming messages from server-specified numbers including those that contain particular strings.
- Actor-specified commands received from text messages which are then promptly removed. These include commands to start/stop recording from the camera or microphone and take photos from the front and back camera.
- GPS location tracking as well as capturing coordinates as soon as an SMS or call is received.
- Functionality to detect when a victim is driving and has the ability to turn off internet and SMS reception during this time.

```
static {
    DbConstants.DB_FACEBOOK_PATH = Environment.getDataDirectory() + "/data/com.facebook.orca/databases/";
    DbConstants.DB_SKYPE_PATH = Environment.getDataDirectory() + "/data/com.skype.raider/files/";
    DbConstants.DB_SKYPE_PATH2 = Environment.getDataDirectory() + "/data/com.skype.raider/databases/";
    DbConstants.DB_INSTAGRAM_PATH = Environment.getDataDirectory() + "/data/com.instagram.android/cache/images/";
    DbConstants.DB_INSTAGRAM_DB_PATH = Environment.getDataDirectory() + "/data/com.instagram.android/databases/";
    DbConstants.DB_TINDER_PATH = Environment.getDataDirectory() + "/data/com.tinder/databases/";
    DbConstants.DB_VIBER_PATH = Environment.getDataDirectory() + "/data/com.viber.voip/databases/";
    DbConstants.DB_ZALO_PATH = Environment.getDataDirectory() + "/data/com.zing.zalo/databases/";
    DbConstants.DB_IMO_PATH = Environment.getDataDirectory() + "/data/com.imo.android.imoim/databases/";
    DbConstants.AUDIO_IMO_PATH = Environment.getDataDirectory() + "/data/com.imo.android.imoim/files/audio/";
    DbConstants.DB_TUMBLR_PATH = Environment.getDataDirectory() + "/data/com.tumblr/databases/";
    DbConstants.DB_HANGOUTS_PATH = Environment.getDataDirectory() + "/data/com.google.android.talk/databases/";
    DbConstants.DB_HIKE_PATH = Environment.getDataDirectory() + "/data/com.bsb.hike/databases/";
    DbConstants.DB_KIK_PATH = Environment.getDataDirectory() + "/data/kik.android/databases/";
    DbConstants.DB_WHATSAPP_PATH = Environment.getDataDirectory() + "/data/com.whatsapp/databases/";
}
```

Figure 14: Stealth Mango attempts to access and exfiltrate the above databases of third-party social media applications if they are installed on the infected device.

Later versions of Stealth Mango contain heightened functionality to further track victims in real time including obtaining root access to the device in order to ensure persistence as well as access the message databases of third-party social media applications. Stealth Mango creates multiple geo-fences which then monitors whenever the infected device crosses fence lines. It then communicates these logs back to command and control infrastructure. In February 2018, Stealth Mango also extended key logging, screenshot, and screen record functionality, which is also actor-controlled.

```
private void insertTypedKeys() {
    try {
        SQLiteDatabase v1 = DbManager.getInstance().openDatabase();
        ContentValues v0 = new ContentValues();
        v0.put("date", Util.formatDate(System.currentTimeMillis() + ""));
        v0.put("name", AppUtil.getAppNameFromPackage(AccessibilityProtection.lastPackageName));
        v0.put("data", AccessibilityProtection.lastTypedText);
        v0.put("is_sent", Integer.valueOf(0));
        v1.insert("KEY_LOGGER_TABLE", null, v0);
        DbManager.getInstance().closeDatabase();
    }
    catch (Exception v2) {
        Util.Log("error inserting keys data");
    }
}

public void onAccessibilityEvent(AccessibilityEvent arg13) {
    switch (arg13.getEventType()) {
        case 1: {
            goto label_3;
        }
        case 16: {
            goto label_77;
        }
        case 32: {
            goto label_34;
        }
    }
}
```

Figure 15: Stealth Mango uses the Accessibility Service to log keystrokes and insert them into a local database. This is stored along with the application package name which was in the foreground when the keystrokes were logged.

Tangelo functionality (iOS)

Throughout the course of this investigation we have seen a large percentage of images in the exfil data that are from iOS devices without much more of an indication that there exists an iOS implant. However, from our investigation into more infrastructure believed to be owned and operated by the same developer, we encountered an iOS implant for jailbroken devices. The implant comes in the form of a Debian package and was released in December 2016. The malware goes after private databases (only accessible on a jailbroken iOS device) from WhatsApp, Viber, Skype, and Line. Additionally, it contains functionality to get the following data:

- SMS messages
- Call logs
- Cellular IDs
- Browser history
- Pictures
- Videos
- GPS coordinates
- Call recordings
- Environmental recordings

```

57 v54 = self;
58 v53 = a2;
59 _objc_msgSend(self, "readThreadids");
60 _objc_msgSend(
61     &objc_class___NSString,
62     "stringWithFormat:",
63     CFSTR("%@/%@"),
64     CFSTR("/var/mobile/Library/SMS"),
65     CFSTR("sms.db"));
66 v2 = _objc_msgSend(
67     &objc_class___NSString,
68     "stringWithFormat:",
69     CFSTR("-----readAllSMS-----Exe mode"));
70 NSLog(CFSTR("%@"), v2);
71 v52 = _objc_msgSend(v54, "getLocalSMSDatabaseFilePath");
72 v3 = _objc_msgSend(
73     &objc_class___NSString,
74     "stringWithFormat:",
75     CFSTR("-----readAllSMS-----SMS DB Path-----%@"),
76     v52);
77 NSLog(CFSTR("%@"), v3);
78 v4 = _objc_msgSend(v52, "UTF8String");
79 v50 = sqlite3_open_v2(v4, &v51, 2, 0);
80 v5 = _objc_msgSend(
81     &objc_class___NSString,
82     "stringWithFormat:",
83     CFSTR("-----readAllSMS-----DB open Result -----%@"),
84     v50);
85 NSLog(CFSTR("%@"), v5);
86 if ( !v50 )
87 {
88     v6 = _objc_msgSend(&objc_class___UtilityClass, "valueForKey:", CFSTR("kLastSMSTag"));
89     v49 = _objc_msgSend(v6, "integerValue");
90     v7 = _objc_msgSend(
91         &objc_class___NSString,
92         "stringWithFormat:",
93         CFSTR("-----last SMS Record Rowid -----%@"),
94         v49);
95 NSLog(CFSTR("%@"), v7);
96 v48 = _objc_msgSend(
97     &objc_class___NSString,
98     "stringWithFormat:",
99     CFSTR("select case is_sent when 0 then 'Received' when 1 then 'Sent' else 'Unknown' end as is_sent, handle_id, text, datetime(date, 'unixepoch',
100         v48);
101 v50 = sqlite3_prepare_v2(v50, v47, -1, &v46, 0);
102 v50 = sqlite3_step(v50);
103 v8 = _objc_msgSend(
104     &objc_class___NSString,
105     "stringWithFormat:",
106     CFSTR("-----readAllSMS111-----SMS compiledStatement open Result -----%@"),
107     v50);
108 NSLog(CFSTR("%@"), v8);
109 if ( !v50 )
110 {
111     v9 = _objc_msgSend(
112         &objc_class___NSString,
113         "stringWithFormat:",
114         CFSTR("-----SMS before iterating -----"));
115 NSLog(CFSTR("%@"), v9);
116 v45 = sqlite3_step(v46);
117 while ( sqlite3_step(v45) == 100 )
118 {
119     v10 = _objc_msgSend(
120         &objc_class___NSString,
121         "stringWithFormat:",
122         CFSTR("-----SMS Start-----"));
123 NSLog(CFSTR("%@"), v10);
124 v11 = sqlite3_column_text(v46, 0);
125 v46 = _objc_msgSend(&objc_class___NSString, "stringWithUTF8String:", v11);
126 v12 = _objc_msgSend(&objc_class___NSString, "stringWithFormat:", CFSTR("Message type: %@"), v46);
127 NSLog(CFSTR("%@"), v12);
128 v13 = sqlite3_column_text(v46, 1);
129 v43 = _objc_msgSend(&objc_class___NSString, "stringWithUTF8String:", v13);
130 v43 = _objc_msgSend(@"(id)%" + v56 + 2, "valueForKey:", v43);
131 v14 = _objc_msgSend(&objc_class___NSString, "stringWithFormat:", CFSTR("Phone Number: %@"), v43);
132 NSLog(CFSTR("%@"), v14);
133 v15 = sqlite3_column_text(v46, 1);
134 v42 = _objc_msgSend(&objc_class___NSString, "stringWithUTF8String:", v15);

```

Figure 16: Code from Tangelo that reads SMS data.

Like the Android implant, the iOS implant contains connections to TheOneSpy and MStealthAgent.

Additionally, we discovered that one of the IP addresses associated with the Tangelo implant was found in two apps in the iOS App Store. We don't have reason to suspect that these apps are malicious in nature at this time. The apps are from a company called MULTITEL LLC, which looks to have contracted the developers of Stealth Mango at some point to create their mobile apps for both iOS and Android. However, one of the MULTITEL apps called MultiTEL, does send data to a server owned by the developer where Tangelo was found.

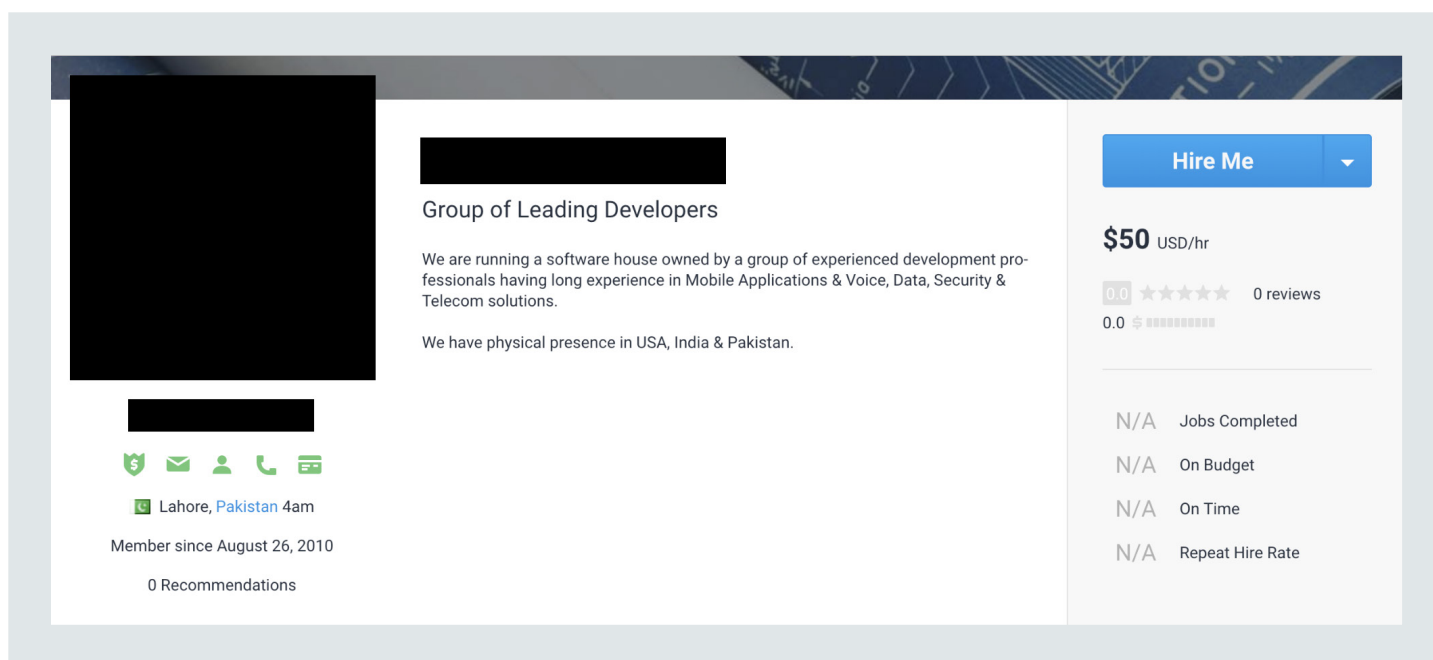
Threat actor activity

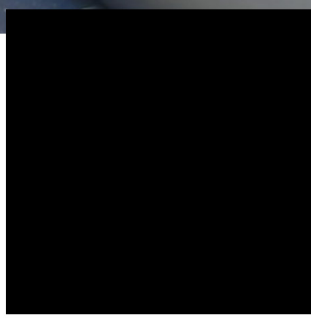
Throughout this investigation, Lookout researchers gained unique insight into the global operations of the actor deploying Stealth Mango. This has primarily been possible due to the actor's poor operational security, which allowed public access to data stolen from compromised devices. The presence of a web shell on the primary command and control server further increased this visibility and caused scripts, configuration files, and additional exfiltrated content to be publicly exposed.


In analyzing this dataset, we identified what appears to be numerous test devices from the developers who we believe work for the same freelance developer group. Audio recordings and content from these devices strongly suggest Stealth Mango is being actively demoed to prospective customers and that the developers behind it are succeeding in selling it. Not only have we been able to build out a picture of adversary behavior from this, but content from victim devices has helped us better understand why the actor has chosen its specific targets.






Associated Stealth Mango developers


During our code analysis of Stealth Mango we found multiple similarities to other commodity spyware families that fall into the category of "spouseware," or apps marketed as software that allow individuals to track and monitor the mobile devices of significant others or their children. Research into the infrastructure behind these families has consistently linked back to several key individuals that we have identified as belonging to the same freelance developer group associated with Stealth Mango and Tangelo.









Lahore, Pakistan 4am

Member since August 26, 2010

0 Recommendations



Group of Leading Developers

We are running a software house owned by a group of experienced development professionals having long experience in Mobile Applications & Voice, Data, Security & Telecom solutions.

We have physical presence in USA, India & Pakistan.

Hire Me

\$50 USD/hr

0.0 ★★★★★ 0 reviews

0.0 \$

N/A Jobs Completed

N/A On Budget

N/A On Time

N/A Repeat Hire Rate

Figure 17: The developer group advertises itself as having a physical presence in the United States, India, and Pakistan, and offers its services for only \$50 USD an hour.

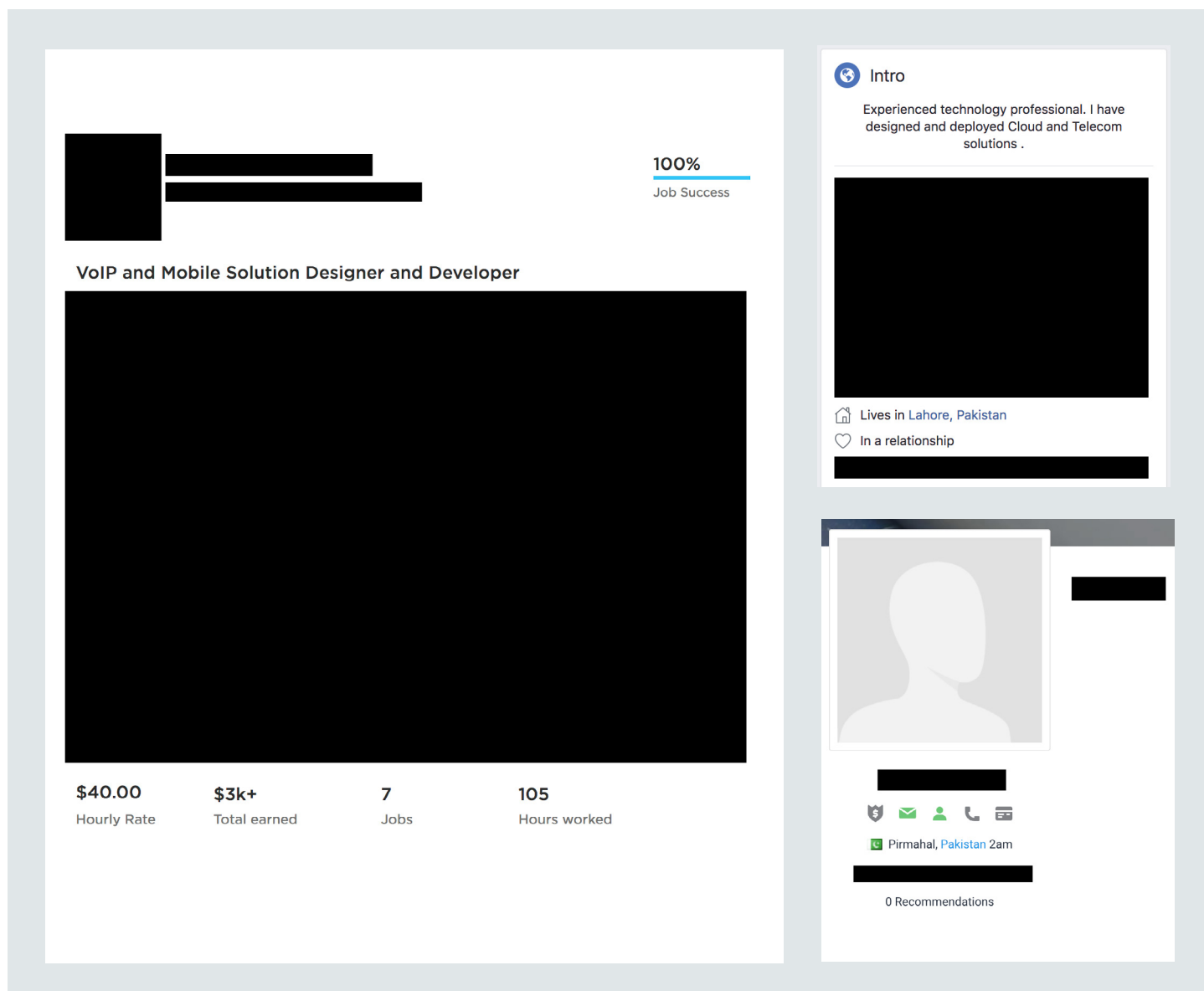


Figure 18: Other individuals associated with Stealth Mango, also individually advertise their services on Freelancer.

Among the data taken from test devices is personal information for an individual who we believe may be the main developer. This data included referral letters for two separate software development companies called Vopium and Appstertech. We identified other developer identities in this exfiltrated data who may have also been previously employed at these companies, further connecting these identities to Stealth Mango.

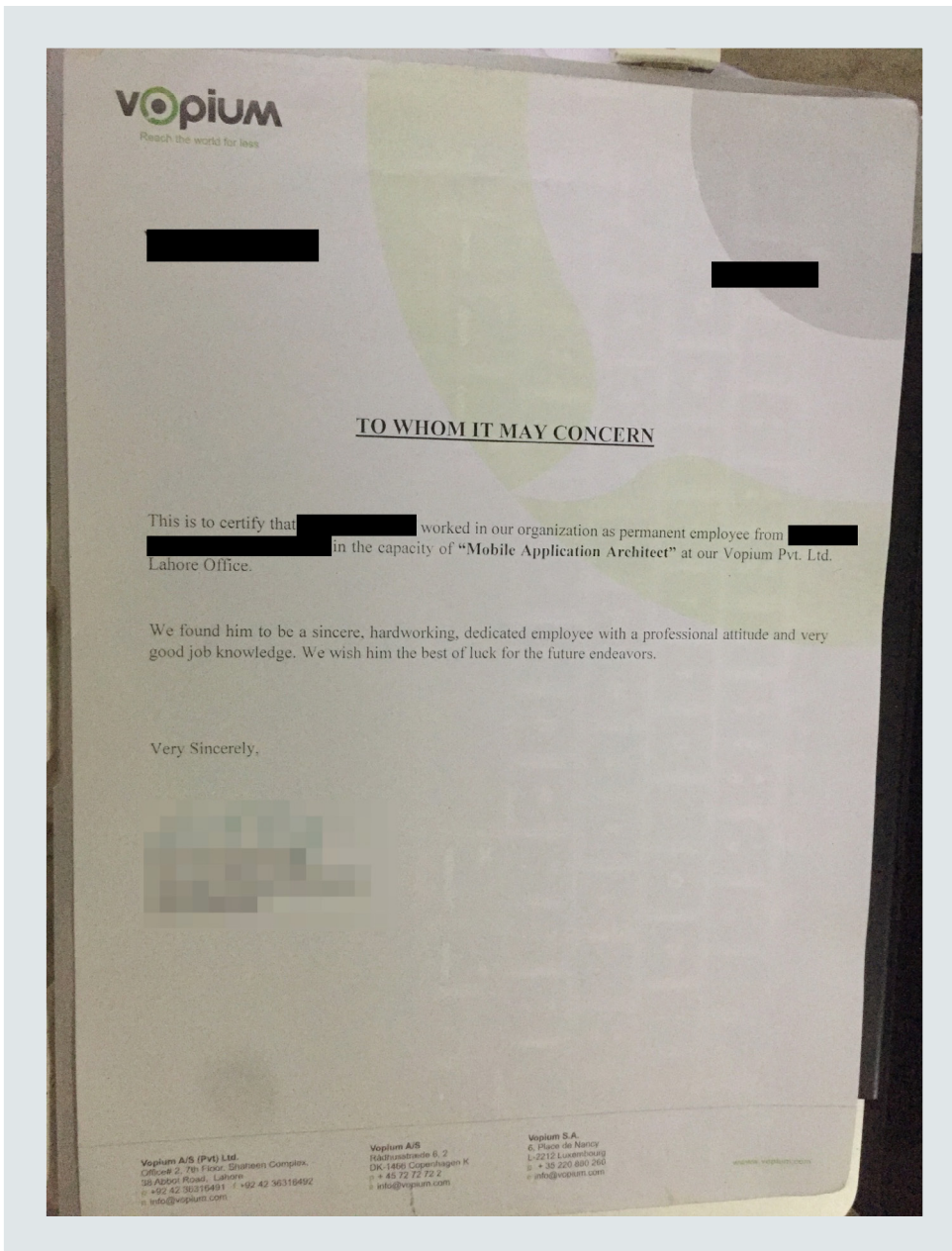


Figure 19: A large amount of data was found on attacker infrastructure that belonged to the main developer. This included references from Vopium and Appstertech as well as identity cards and personal photos.

We also observed the main developer using his email address to register domains associated with spyware families that share many code similarities and heuristics with Stealth Mango. The most significant of these is TheOneSpy spyware from a company called Ox-i-Gen Inc. that we suspect the main developer may have worked on in some capacity. While Ox-i-Gen has its headquarters in Sydney, Australia most of the connected employees on LinkedIn that work for Ox-i-Gen are located in Lahore, Pakistan which along with the developer group connections may explain overlap between Stealth Mango and TheOneSpy. Furthermore, the main developer previously owned the domain `mstealthagent[.]com`, which we believe is an earlier iteration of Stealth Mango. This indicates he has been developing spyware tooling like Stealth Mango in official and unofficial capacities for a few years.

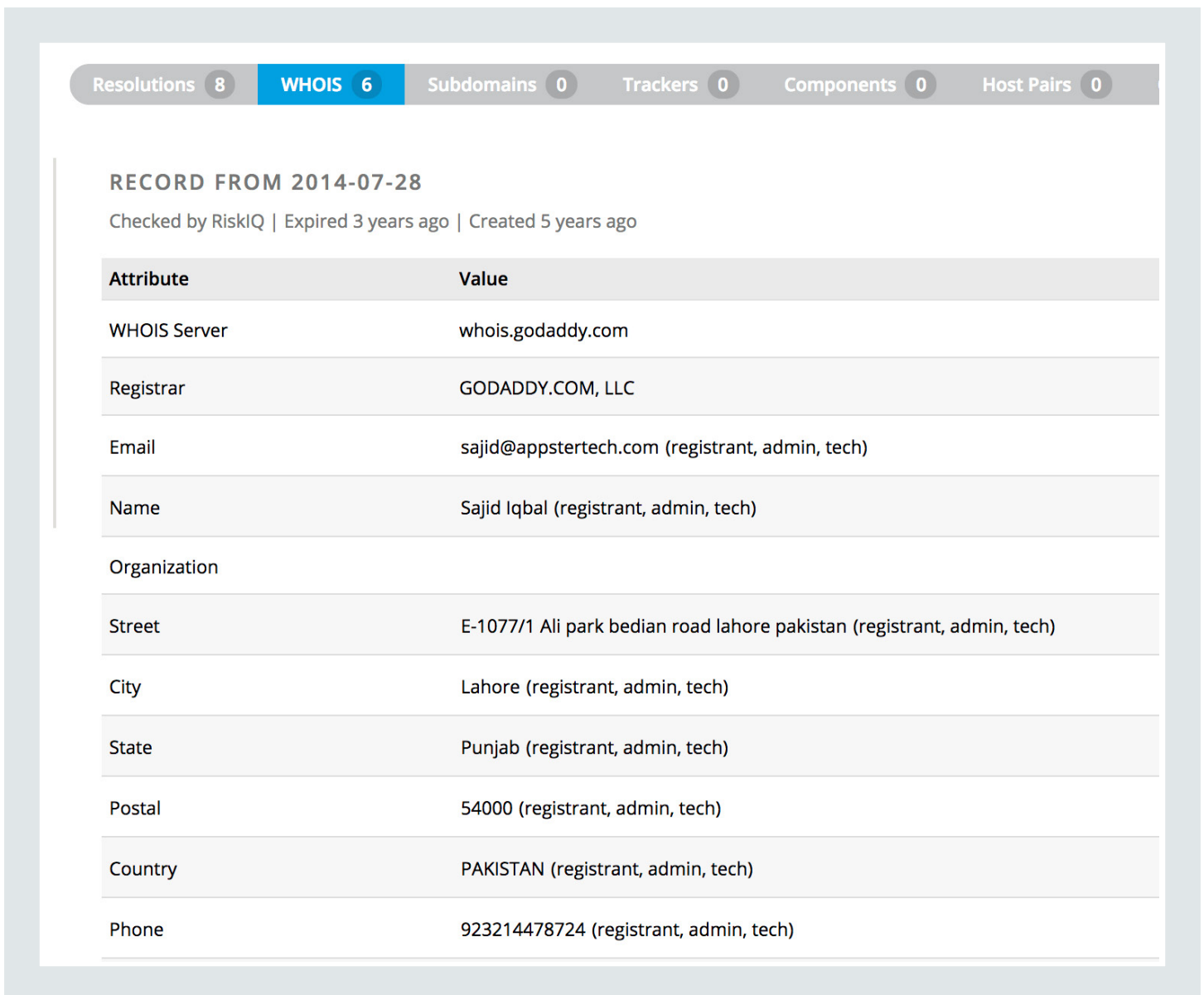


Figure 20: The main developer's email address was previously used to register the domain mstealthagent[.]com for which spyware with code similar to Stealth Mango has been identified.

From the information available, our working theory is that the main developer is a full-time app developer primarily focused on creating legitimate apps, but who is also moonlighting on the side. He is also part of a group of developers selling mobile surveillanceware.

Further analysis of server side logs on attacker infrastructure showed three IPs that geolocate to a specific area of the G-8 area in Islamabad, Pakistan accessed the infrastructure. This is shown below, however at this point in time it is unclear whether this is infrastructure being accessed by an administrator or some other actor.

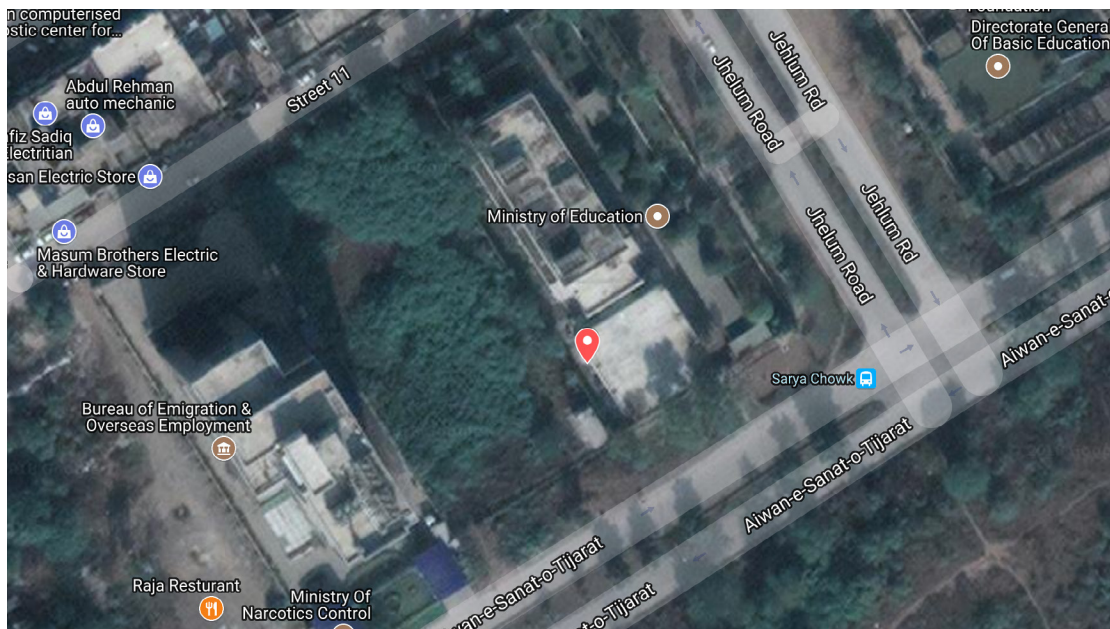
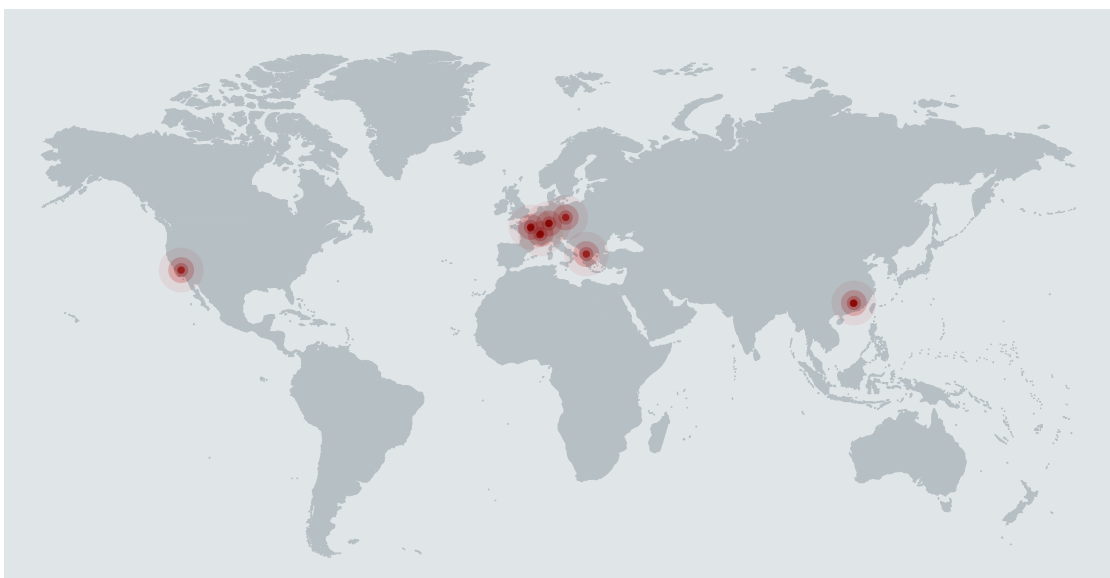


Figure 21: Geo-located IP Addresses that were observed to have logged into the C2 server from the G-8 area in Islamabad, Pakistan.

Infrastructure

Infrastructure for Stealth Mango uses two IP address. The server itself (217.182.147[.]171) is hosted in France with a jump box to that server located in Canada (158.69.159[.]57). However, as the campaign and actors are very active, we discovered the actor created several more malicious APKs that point to additional jump boxes, all of which resolve back to the main server:

- 178.33.140[.]197 • 149.56.237[.]148 • 164.132.182[.]141 • 137.74.221[.]199 • 137.74.147[.]190
- 137.74.221[.]193 • 164.132.182[.]142 • 178.33.140[.]198 • 158.69.159[.]58 • 51.255.13[.]89



The IPs map to servers hosted in the United Kingdom, United States, Germany, Italy, Hong Kong, and France.

Figure 22: Locations of the C2 infrastructure for Stealth Mango.

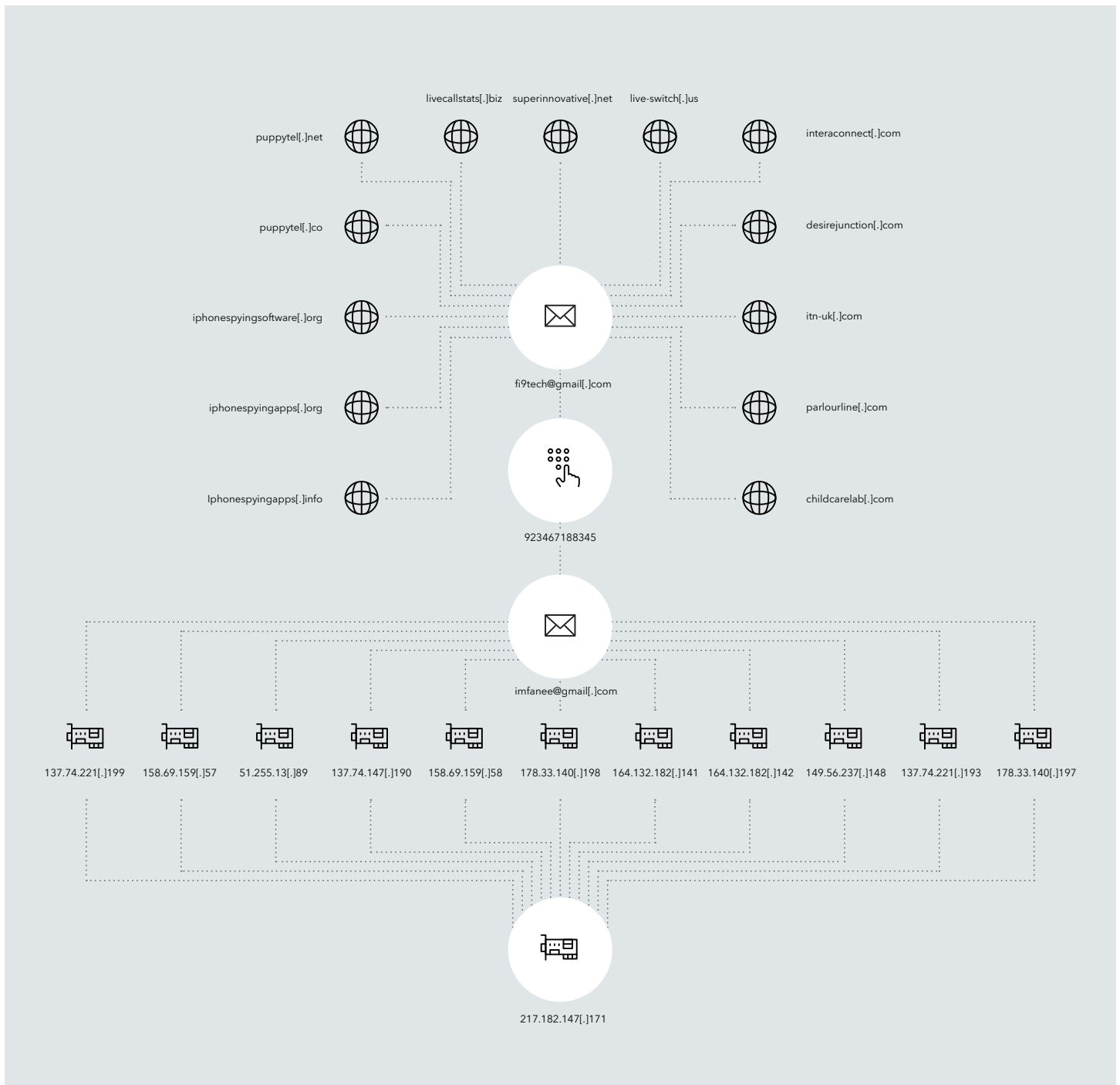


Figure 23: Mapping key pieces of attacker infrastructure and associated information.

Lookout researchers believe the main server may have been compromised by a unknown third party as it was found to be running WSO shell. While WSO shell does require a user to authenticate themselves if they attempt to access the main user dashboard, it was possible to login unauthenticated simply by browsing to a specific URL. Navigating to this URL provided complete access to the WSO web shell and allows an operator to run various commands and browse server infrastructure as the www-data user. A user can also use it to connect to local MySQL databases, execute arbitrary PHP scripts, run various file operations, retrieve server details, and run console commands.

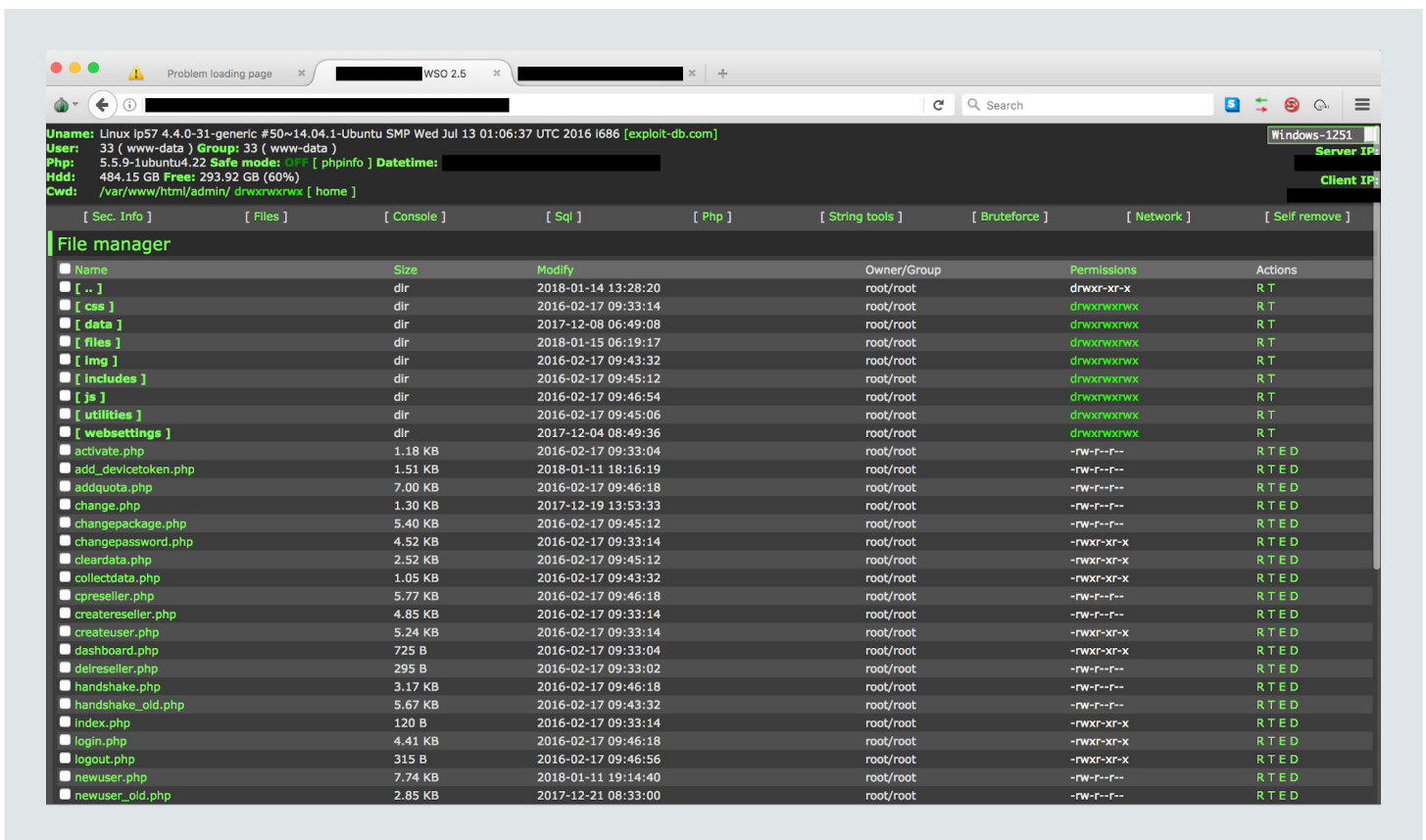


Figure 24: A screenshot of the WSO shell found to be running on infrastructure used by Stealth Mango. This provides an actor with a considerable amount of control over the command and control server. Much of the supporting PHP scripts are dated from 2015 and 2016 however exfil is very recent.

We also found the infrastructure hosting several APKs belonging to the Stealth Mango family which were only very recently added. This indicates that not only is this operation still ongoing but that the tooling is being actively improved. We further confirmed this via audio conversations from the developers that were captured when they were testing the malware on their own devices.

Conclusions

Stealth Mango and Tangelo is yet another example among the numerous campaigns we have uncovered ([Dark Caracal](#), [ViperRAT](#), [FrozenCell](#), etc.) where threat actors are developing in-house custom surveillanceware. The actor behind Stealth Mango has stolen a significant amount of sensitive data from compromised devices without the need to resort to exploits of any kind. The actors that are developing this surveillanceware are also setting up their own command and control infrastructure and in some cases encountering some operational security missteps, enabling researchers to discover who the targets are and details about the actors operating it that otherwise are not as easily obtained. Relevant data has already been shared with the appropriate authorities. Lookout customers are protected against Stealth Mango and Tangelo and have been for several months since the beginning of the investigation.

Appendix

Indicators of Compromise

Servers and IPs

| | |
|-------------------|----------|
| 158.69.159[.]57 | jump box |
| 178.33.140[.]197 | jump box |
| 137.74.221[.]193 | jump box |
| 149.56.237[.]148 | jump box |
| 164.132.182[.]142 | jump box |
| 164.132.182[.]141 | jump box |
| 178.33.140[.]198 | jump box |
| 137.74.221[.]199 | jump box |
| 158.69.159[.]58 | jump box |
| 137.74.147[.]190 | jump box |
| 51.255.13[.]89 | jump box |

| | |
|------------------|-------------|
| 128.199.53[.]121 | iOS related |
| 178.238.226[.]34 | iOS related |

| | |
|-------------------|-------------|
| 217.182.147[.]171 | main server |
|-------------------|-------------|

| | |
|-------------------------------------|---------------------|
| secure-apps.azurewebsites[.]net | watering hole |
| www.playstorepk[.]com | older watering hole |
| facebook-snaps.azurewebsites[.]net | watering hole |
| secure-google.azurewebsites[.]net | watering hole |
| proworld.azurewebsites[.]net | watering hole |
| secure-facebook.azurewebsites[.]net | watering hole |

APK hashes

| | | |
|-------|---|-----------------------|
| SHA1: | a910ad50e1fcf65f56f69b0efae1de6ff29e3998 | |
| SHA1: | 133fa44cd54d6f684035145a3010a9eca4926d56 | |
| SHA1: | d033425ade25c90ce00ca7503cb92300d91f9477 | |
| SHA1: | 37805ffbbbed21128189c87dff5bd1f5c81c9b841 | |
| SHA1: | 0c64547cbb3b556f2d48902e80385a80895e7ae0 | |
| SHA1: | 8fb0b7fee1ce2a4a7bf935334ef07bcd741c7958 | |
| SHA1: | 408550f277b166e60328314619ca414e7acc8953 | |
| SHA1: | 2428061f9f46388f6d8fcebba40b5dd54701dd8cc | |
| SHA1: | edd25c64e34bef582cd4d209d8dfe2c25c085495 | |
| SHA1: | 7d1b14727aa529bbf9a3dbeba9cd881b91dff96c | |
| SHA1: | 9f0ad56300e6d782325896479bd704761a5e3c45 | |
| SHA1: | 58d515e04ce02f47988068f3505bd2a9b85ea700 | |
| SHA1: | 73b54dc04b93449a137401da7e08bb35e2467e4a | |
| SHA1: | a59389656510a49fb8533e113d39300e032efb28 | |
| SHA1: | ccba51dbd36d0586a32c458bcd769201975fa4d7 | |
| SHA1: | 4db9beb5c6e83a124993060fe1ea235413b93a68 | |
| SHA1: | fd12615afc23ee780b21c825a8202e9b4768bca0 | |
| SHA1: | ce41bc550998270b8d271abaef3261661ae64349 | |
| SHA1: | 2a78145905e5cd5a7b8085bcfae0balld032850a | |
| SHA1: | 74987b1a3aa139348c179e86364abee1b4f9dadcd | different IP jump box |
| SHA1: | 07bfeb6cf511e3110ae7ac92c39ff4732bd74619 | different IP jump box |
| SHA1: | dd3629e11943948d77679971564a12032e1c2c99 | different IP jump box |
| SHA1: | 2b3dced1712c838c283137baa132871280b10e97 | different IP jump box |
| SHA1: | 5e54a85d8baaa24a8ee644fd397ba4301f7e0152 | different IP jump box |
| SHA1: | 3f87a6e09a7c095ecf192aa8932500795ae4201c | different IP jump box |
| SHA1: | 166449cdc909ce303ba4bd2f8a154c6a23bf5915 | different IP jump box |
| SHA1: | 9bfedcbd7968e0e56d9d16e42f0e46115ece1bc5 | different IP jump box |
| SHA1: | a31f3b7fblcb5c3bef967a599e0cbe7d4fcc26b8 | |
| SHA1: | 815900850aa1bb1ab7f383e88bf81e6a4be41fb7 | |
| SHA1: | ff321433e88986f0cb6782be640e11edf7d4fb03 | |
| SHA1: | faa6223fe6c58d24a5c099239b8bc2ea40d99fdb | |
| SHA1: | d51556a13a5e781a3c23fd91e0858e31e056c214 | |

| | | |
|-------|---|--|
| SHA1: | 89e3bf4b097acadbcb89fc39ca9daac6f5c574e00 | |
| SHA1: | 2369b8407ca0e2c30ab66ca74895ea3c0a157ec7 | |
| SHA1: | 5992bf8d560b75b44bade3997bf4c6470f798e31 | |
| SHA1: | 0736bf049415cc0804ce54538d28db2206e52b90 | |
| SHA1: | 9f8c3af2d4a8ad19852108b0f46c68504adb8245 | |
| SHA1: | a57b6f262ed0a9b3d3cb5338cb968593c490b6e3 | |

APK package names and app names

| | | | |
|---------------|----------------------------|-----------|---------------|
| Package Name: | com.gbooking.googleupdater | App Name: | GoogleUpdater |
| Package Name: | com.update.system | App Name: | System |
| Package Name: | com.ittelephone.dialer | App Name: | Dialer |
| Package Name: | com.due.gplayer | App Name: | GPlayer |
| Package Name: | com.maps.lgmaps | App Name: | gmaps |
| Package Name: | com.booking.gvoice | App Name: | GVoice |
| Package Name: | com.gsync | App Name: | Gsync |
| Package Name: | com.play.pservices | App Name: | Pservices |
| Package Name: | com.lgoogle.playupdate | App Name: | Playupdater |
| Package Name: | com.gsearch.ichrome | App Name: | iChrome |

iOS hashes

| | | |
|-------|--|----------------|
| SHA1: | 1261189102aee97fe5d811e8114e282e86876ea | Debian package |
| SHA1: | 221a884f6f9c8428033123861d95aa7e7445c40a | Mach-O binary |
| SHA1: | 921d6e701a39f2a70eea2c94cf4ba21d61c2ceb5 | Mach-O binary |

iOS Bundle ID

| | |
|------------|----------------------------|
| Bundle ID: | com.mobilekare.notifierrrr |
|------------|----------------------------|

Miscellaneous

| | |
|--------------|------------|
| iOS Team ID: | GUDCEEC5K9 |
|--------------|------------|

About Lookout

Lookout is a cybersecurity company for a world run by apps. Powered by the largest dataset of mobile code in existence, Lookout is the security platform of record for mobile device integrity and data access. Lookout is trusted by hundreds of millions of individuals, hundreds of enterprises and government agencies, and such ecosystem partners as AT&T, Deutsche Telekom, and Microsoft. Headquartered in San Francisco, Lookout has offices in Amsterdam, Boston, London, Sydney, Tokyo, Toronto and Washington, D.C.

Lookout Website

www.lookout.com

Blog

blog.lookout.com

Email

threatintel@lookout.com

Twitter

[@lookout](https://twitter.com/lookout)

Contributors

Andrew Blaich, *Head of Device Intelligence*

Apurva Kumar, *Sr. Security Intelligence Engineer*

Michael Flossman, *Head of Threat Intelligence*

Robert Nickle, *Staff Security Intelligence Engineer*

All security research conducted by Lookout employees is performed according to the Computer Fraud and Abuse Act (CFAA) of 1986. As such, analysis of adversary infrastructure and the retrieval of any exposed data is limited to only that which is publicly accessible. Any sensitive information obtained during this process, such as usernames or passwords, is never used in any authentication-based situations where its use would grant access to services or systems.