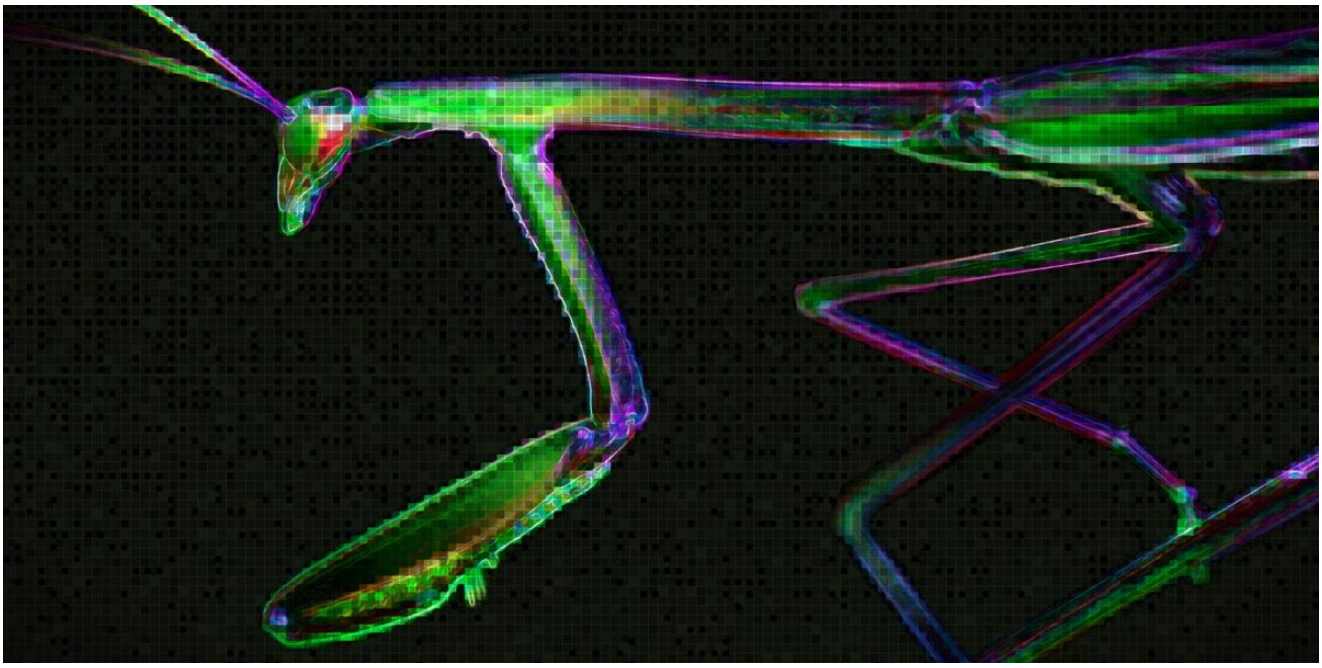
Roaming Mantis reaches Europe

SL securelist.com/roaming-mantis-reaches-europe/105596/



Authors

Expert Suguru Ishimaru

Part VI. 2021 sees smishing and modified Wroba.g/Wroba.o extend attacks to Germany and France

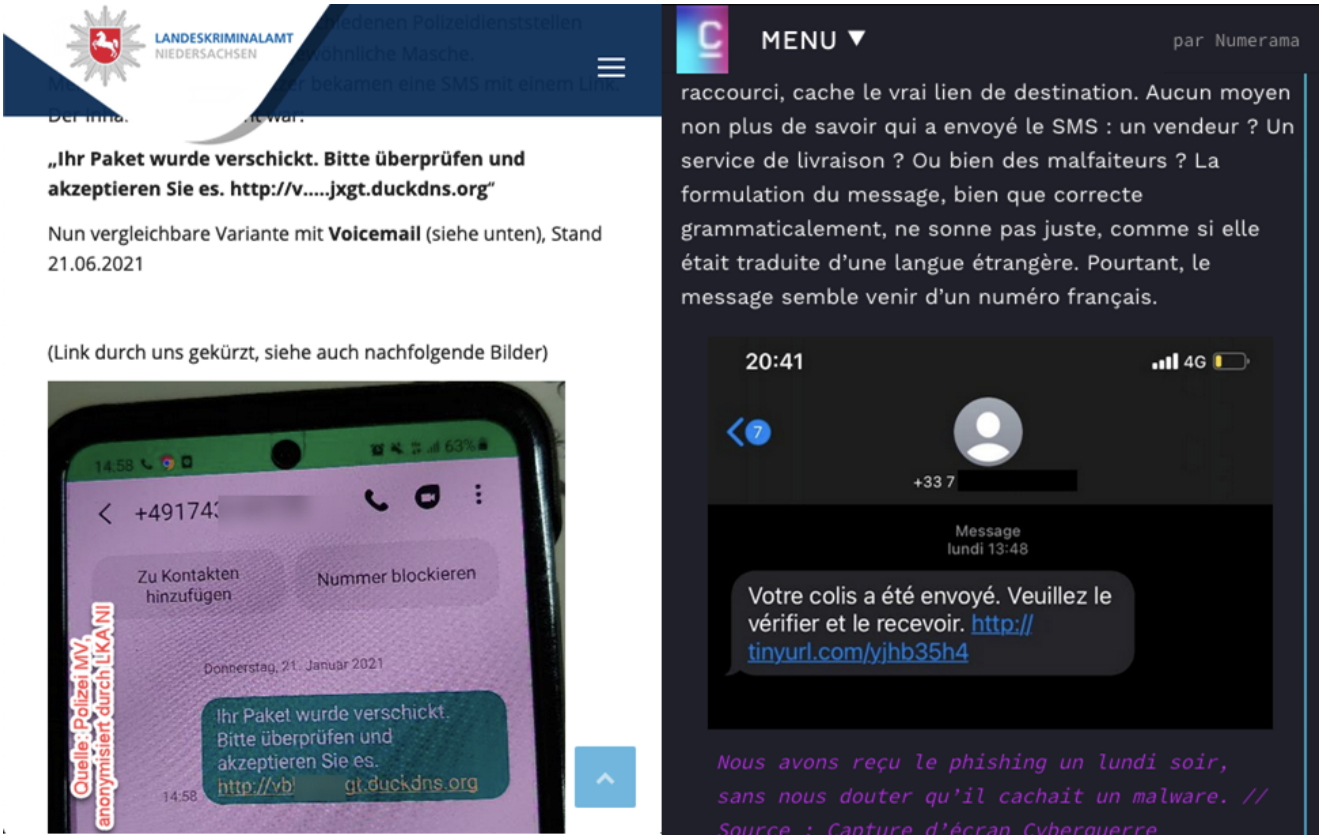
Roaming Mantis is a malicious campaign that targets Android devices and spreads mobile malware via smishing. We have been tracking Roaming Mantis since 2018, and published five blog posts about this campaign:

- Roaming Mantis uses DNS hijacking to infect Android smartphones
- Roaming Mantis dabbles in mining and phishing multilingually
- Roaming Mantis, part III
- Roaming Mantis, part IV
- Roaming Mantis, part V

It's been a while since the last blog post, but we've observed some new activities by Roaming Mantis in 2021, and some changes in the Android Trojan Wroba.g (or Wroba.o, a.k.a Moqhao, XLoader) that's mainly used in this campaign. Furthermore, we discovered that France and Germany were added as primary targets of Roaming Mantis, in addition to Japan, Taiwan and Korea.

Geography of Roaming Mantis victims

Our latest research into Roaming Mantis shows that the actor is focusing on expanding infection via smishing to users in Europe. The campaign in France and Germany was so active that it came to the attention of the German police and French media. They alerted users about smishing messages and the compromised websites used as landing pages.



Smishing alerts on German and French websites

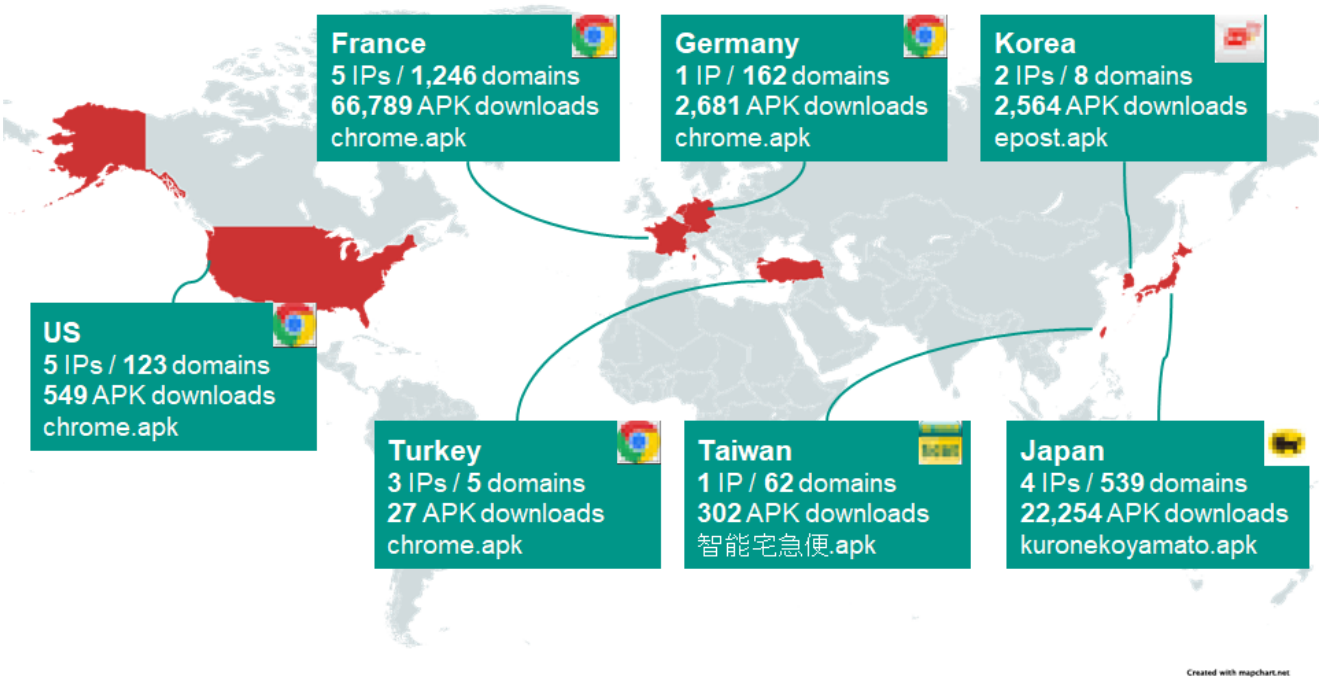
Typically, the smishing messages contain a very short description and a URL to a landing page. If a user clicks on the link and opens the landing page, there are two scenarios: iOS users are redirected to a phishing page imitating the official Apple website, while the Wroba malware is downloaded on Android devices.

Link from smishing message redirects to Wroba or phishing page

Based on the telemetry we gathered between July 2021 and January 2022, Wroba.g and Wroba.o have been detected in many regions. The most affected countries were France, Japan, India, China, Germany and Korea.

Territories affected by Trojan-Dropper.AndroidOS.Wroba.g and Trojan-Dropper.AndroidOS.Wroba.o (download)

We'd also like to point out some very interesting data on Roaming Mantis landing page statistics published on Internet Week 2021 and Github by @ninoseki, an independent security expert based in Japan. The data shows the number of downloaded APK files, landing page domains, and IP addresses located in the seven regions targeted most by Roaming Mantis using Wroba.g/Wroba.o on a particular day in September 2021.



The number of downloaded APK files and IPs/domains of landing pages

The following table is a ranking based on the number of APK file downloads. The most affected country is France, followed by Japan, Germany and others. Some targeted regions seem to overlap with our telemetry mentioned above.

	Region	Number of	Impersonated brand		
IPs	domains	downloads			
1	France	5	1,246	66,789	Google Chrome
2	Japan	4	539	22,254	Yamato transport
3	Germany	1	162	2,681	Google Chrome
4	Korea	2	8	2,564	ePOST
5	United States	5	123	549	Google Chrome
6	Taiwan	1	62	302	智能宅急便 (Yamato transport in Chinese)
7	Turkey	3	5	27	Google Chrome

Anti-researcher tricks in the landing page

Throughout 2020 and 2021, the criminal group behind Roaming Mantis made use of various obfuscation techniques in the landing page script in order to evade detection.

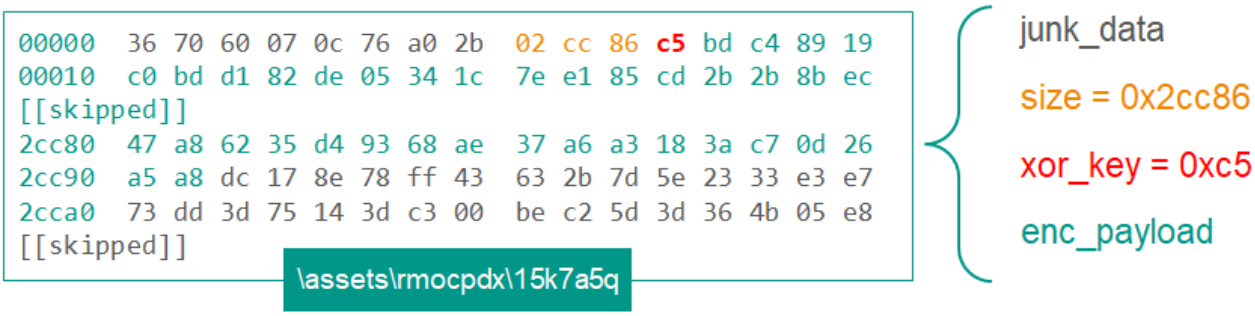
Variety of obfuscation techniques in the landing page script

In addition to obfuscation, the landing page blocks the connection from the source IP address in non-targeted regions and shows just a fake “404” page for these connections.

The user agent checking feature has not been changed in the landing page since 2019; it evaluates the devices by user agent, redirecting to the phishing page if the device is iOS-based, or delivering the malicious APK file if the device is Android-based.

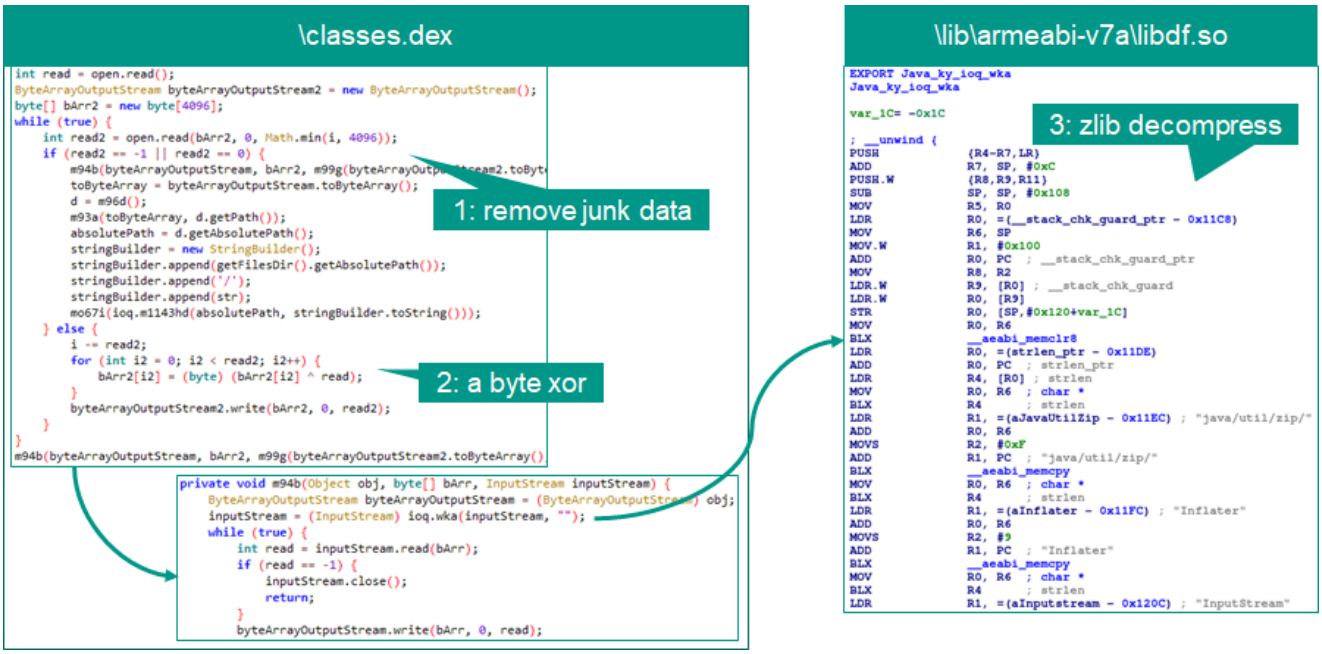
Technical analysis: loader module of Wroba.g/Wroba.o

We performed in-depth analysis of Wroba.g/Wroba.o samples and observed several modifications in the loader module and payload, using *kuronekoyamato.apk* as an example. First, the actor changed the programming language from Java to Kotlin, a programming language designed to interoperate fully with Java. Then, the actor removed the multidex obfuscation trick. Instead of this, the data structure of the embedded payload (\assets\rmocpdx\15k7a5q) was also modified as follows:



Modified data structure of embedded payload

The first eight bytes of the data are junk code (gray), followed by the size of payload (orange), a single-byte XOR key (red), the encrypted payload (green) and more junk code (gray). Furthermore, an ELF file, \lib\armeaib-v7a\libdf.so, was embedded in the APK file: it uses Java Native Interface (JNI) for the second stage payload, for decryption and also part of the loading feature. The decryption process and algorithms are just three steps as follows:



Various obfuscation techniques in the landing page script

First, the loader function takes each section of data from the embedded data, except the junk data. Then, the encrypted payload is XORed using the embedded XOR key. After the XOR operation, as with previous samples, the data is decompressed using zlib to extract the payload, a Dalvik Executable (DEX) file.

The following simple Python script helps to extract the payload:

```
#!/usr/bin/env python3

import sys
import zlib
import base64

data = open(sys.argv[1], "rb").read()
key = data[11]
size = data[10] | data[9] << 8 | data[8] << 16
enc = data[12:12+size]
dec_x = bytes(enc[i] ^ key for i in range(len(enc)))
dec_z = zlib.decompress(dec_x)

with open(sys.argv[1]+".dec","wb") as fp:
    fp.write(dec_z)
```

In this sample, the decrypted payload is saved as \data\data\ggk.onulfc.jb.utxdtt.bk\files\d and executed to infect the malicious main module on victim devices.

Technical analysis: payload of Wroba.g/Wroba.o

Regarding the updates to the Wroba.g/Wroba.o payload, Kaspersky experts only observed two minor updates in the payload part. One of them is the feature for checking the region of the infected device in order to display a phishing page in the corresponding language. In the old sample, it checked for three regions: Hong Kong, Taiwan and Japan. However, Germany and France were added as new regions. From this update, together with the map above, it is clear that Germany and France have become the main targets of Roaming Mantis with Wroba.g/Wroba.o.

Another modification is in the backdoor commands. The developer added two backdoor commands, “get_photo” and “get_gallery”, as well as removing the command “show_fs_float_window”. Overall, there are 21 embedded backdoor commands.

List of embedded backdoor commands with the two new commands 'get_gallery' and 'get_photo'

These new backdoor commands are added to steal galleries and photos from infected devices. This suggests the criminals have two aims in mind. One possible scenario is that the criminals steal details from such things as driver's licenses, health insurance cards or bank cards, to sign up for contracts with QR code payment services or mobile payment services. The criminals are also able to use stolen photos to get money in other ways, such as blackmail or sextortion. The other functions of the payload are unchanged. For more details, please see our previous blogposts mentioned above.

Conclusion

It has been almost four years since Kaspersky first observed the Roaming Mantis campaign. Since then, the criminal group has continued its attack activities by using various malware families such as HEUR:Trojan-Dropper.AndroidOS.Wroba, and various attack methods such as phishing, mining, smishing and DNS poisoning. In addition, the group has now expanded its geography, adding two European countries to its main target regions. We predict these attacks will continue in 2022 because of the strong financial motivation.

MD5 hashes of Wroba.o

527b5eebb6dbd3d0b777c714e707659c
19c4be7d5d8bf759771f35dec45f267a
2942ca2996a80ab807be08e7120c2556
4fbc28088b9bf82dcb3bf42fe1fc1f6d
0aaf6aa859fbdb84de20bf4bf28a02f1
5bafe0e5a96b1a0db291cf9d57aab0bc
ddd131d7f0918ece86cc7a68cbacb37d

- Google Android
- Malware Descriptions
- Malware Statistics
- Malware Technologies
- Mobile Malware
- smishing

Authors

Suguru Ishimaru

Roaming Mantis reaches Europe

Your email address will not be published. Required fields are marked *