Flubot’s Smishing Campaigns under the Microscope

telekom.com/en/blog/group/article/flubot-under-the-microscope-636368

Blog.Telekom

09-14-2021
Thomas Barabosch

3 Comments

- Share Share
  Two clicks for more data privacy: click here to activate the button and send your recommendation. Data will be transfered as soon as the activation occurs.

- Print
- Read out

Our lives cannot be imagined without mobile phones. They know all our secrets and we use them to accomplish tasks like online banking or cryptocurrency trading. Most of them run the Android operating system resulting in billions of Android installations worldwide. Given Android’s ubiquity and the kind of sensitive data Android devices handle, cybercriminals routinely target them for financial gain.

Flubot is so successful for a reason - unfortunately.

One of the tools such attackers utilize is Flubot, which is a botnet primarily targeting Android mobile phones. Its first appearance was in late 2020. Flubot comprises information stealing capabilities (exfiltrate contact list, SMS exfiltration), spamming capabilities (sending of smishing SMS), and application manipulation capabilities (injecting HTML code in banking and cryptocurrency apps).

This botnet spreads by sending SMS like “Notification: (1) new voice message: LINK”, where LINK redirects the target to a lure server serving a website that convinces the target to install a third-party APK. This behavior is known as smishing, an artificial word derived from “SMS” and “Phishing”. An infected device exfiltrates its contact list to the command and control server, which commands it to try to infect hundreds or even thousands of other devices each day. Hence, the name “Flubot” as this botnet spreads like the flu. Since its inception in late 2020, this botnet has become a serious threat to end users and an annoying problem for carriers around the world.

Telekom Security has detected thousands of Flubot infections of Deutsche Telekom’s clients throughout 2021. Each infected client is notified and offered assistance during the cleaning process. The following figure shows the amount of unique infections per day from May 2021 until September 2021. Note that the population of infections heavily fluctuates due to continued effort to notify our clients and consequently by them removing the malware from their devices. The plot shows a period during June and July where less infections were detected. This is a typical seasonal fluctuation also known as “the threat actor’s summer break”. As of September 2021, we are noticing how the botnet activity is increasing and the current infection level is converging against a level we noticed in May 2021.

Figure 1 Unique Flubot infections per day of Deutsche Telekom customers

Furthermore, the Flubot operators keep on working to maintain this status quo. They keep changing their smishing SMS templates every couple of hours and change the links in these SMS every couple of minutes. In addition, they use mechanisms to circumvent simple SMS content filter engines. After several weeks, they usually switch the theme of their campaigns, e.g. from voicemail to parcel services as observed in the last week of August 2021. And notably they implement a mechanism to shut out security researchers who are running bot emulations from their botnet: they verify if new bots can send out SMS!

In this blog post, we’ll see how the smishing SMS work, how Flubot utilizes social engineering to convince targets to install APKs, discuss the payloads that Flubot recently has distributed (i.e. Flubot and Teabot), and show how the operators verify new bots. The goal of this blog post is to give the reader a detailed insight in how Flubot’s smishing campaigns work from end to end. It will not reiterate on the capabilities of the malware itself as there are already very detailed write-ups by Incibe CERT, SWITCH, and ProDaft.

In our Github repository, we share hashes of Flubot and Teabot payloads, YARA rules to hunt for the aforementioned malware families, as well as further analysis scripts.

It just starts with one SMS …

In general, the operators of a botnet (also known as botmasters) face at least two issues when running a botnet. First, users detect infections and clean up devices. This results in a loss of bots. Second, botmasters fully squeeze out bots by, for instance, steal all credentials, conduct wire fraud, etc. This is an indirect loss since it turns a valuable bot into a worthless bot. The solution to both problems is to acquire new bots.

The Flubot operators go to great lengths to acquire new bots: they are capable of circumventing carrier SMS content filters to increase the spreading probability as well as keep out security researchers from the botnet to minimize the scrutiny of their botnet.

The basis for this heavy SMS spamming are the contact lists of Flubot victims that the malware exfiltrates to the command and control server. The command and control server regularly sends bots the command “GET_CONTACTS”. Bots respond with all names and phone numbers of their contact list. In one case, we could determine that the time from contact exfiltration to the first smishing SMS is only a couple of hours. We utilized a phone number that was previously not known to the botnet. This example shows how fast the spamming machinery of Flubot is spinning.

Flubot’s spreading just starts with just one innocently looking SMS that a target receives and which was typically sent by someone not known to the target. Such an SMS comprises an information (e.g. missed call) and a link to follow. The SMS texts changed their theme a couple of times throughout the last months. In August 2021, they matched a voicemail theme by resembling messages that network carriers send on missed calls. This changed in September 2021 when SMS texts started to abuse the DHL brand. Further changes of Flubot’s SMS themes are to be expected as its operators have an history of abusing various brands such as FedEx and Correos (the Spanish postal service) in the past.

The next figure illustrates such a smishing SMS as received by the target. The SMS text starts with random letters and numbers, which is a way to circumvent carrier SMS content filters. This is followed by a text notifying the target that they missed a call, and a suspicious link they should click on to listen to a voicemail. Links in these SMS fluctuate very often, which is another way of not getting blocked by SMS carriers. Each link is personalized. For instance, the parameter “5bb0bxt93” of the “r.php” script is likely utilized to track campaign success.

Figure 2 Flubot smishing SMS telling a target about a missed called, followed by a suspicious link to click on

An important piece of Flubot’s success is how it avoids carrier SMS content filters:

- (1) First, the operators have access to hundreds of hacked websites that they utilize as redirects. These hacked websites in their smishing SMS rotate around every ten minutes.
- (2) Second, they add random letters as prefix or postfix to their smishing SMS.

- (3) Third, they flip one or more letters in their smishing SMS. This may circumvent SMS content filters as it produces different messages but the messages are still readable by human beings. For instance, the word “Call” in “Missed Call” is changed to “Coll” as in “Missed Coll”.
- (4) Fourth, they change the capitalization of one or more words in their messages. For instance, the word “demain” (French: tomorrow) will be capitalized to “DEMAIN”.

The following table illustrates these three circumvention mechanism with several example smishing SMS.

Country	SMS	Method
Australia	New voice-message jecoived: hxxp://fyqz[.vip/m.php?REDACTED	(1), (3)
Australia	(!) New voice message rzcegved: hxxp://tantawy-group[.com/z.php?REDACTED	(1), (3)
The Netherlands	8hd9 Inkomende voice oproep: hxxps://sachizi[.com/r.php?REDACTED	(1), (2)
Austria	Eingehender Anruf: hxxps://www.internationalsengroup[.org/jcqsx9.php?REDACTED unqm	(1), (2)
Italy	Notifdca: (1) nuovo messaggic vocale: hxxps://bitcoinsociety.passionland[.vn/c.php?REDACTED 7jbl6qc	(1), (2), (3)
Germany (sent to the Netherlands)	U heeft 1 nieuwe voicemail. Ga naar hxxps://ospreymine[.co/k.php?REDACTED	(1)
Belgium	Votre commande sera LIVREE par DHL DEMAIN entre 11h26 et 14h26. Suivre le progres hxxps://bodrumenduro[.com/f.php?REDACTED	(1), (4)

The botmasters sometimes utilize Flubot infections of neighboring countries to boost the spreading of Flubot in countries where not yet a strong bot base is established. For instance, Telekom Security observed how German Flubot infections sent SMS with Dutch texts to Dutch phone numbers as well as Polish texts to Polish phone numbers in late August 2021. This is in line with Flubot's extension into these two countries around that time. As a rule of thumb, Telekom Security is observing that roughly ten percent of Flubot's SMS traffic is sent to foreign destination phone numbers.

The spreading of Flubot may cause severe financial damages for Flubot victims as well as network carriers. Infected devices may send a SMS message every couple of minutes, which may add up to significant amounts after some days. Especially if SMS messages are sent across borders, which is a typical tactic to boost the initial spreading rate in a new country. The bot queries the rate at which it should send SMS with the command SMS_RATE (see table of relevant commands). We observed values from 4 minutes up to 60 minutes. This value depends on the current spreading status of the botnet in a specific country. For instance, when the Flubot operators start to spread Flubot in a country, they set the SMS_RATE typically very low (less than 5 minutes). As of September 2021, Telekom Security observes a daily average of 1000 SMS per day per infected client. However, there are some extreme cases of up to 3000 SMS per day.

Command	Description
GET_SMS	Query the command and control server for smishing task, consisting of a telephone number and a SMS text
SMS_RATE	Query the command and control server for the delay time when sending mass SMS
LOG	Response of the bot to log data to the command and control server. The subcommand LOG, SMS logs individual SMS.
GET_CONTACTS	Asks the bot to exfiltrate all names and phone numbers of the victim's contact list.

Social Engineering with Lures

The links of the smishing SMS forward targets to a lure server. The following figures show a generated website using the Voicemail theme that was presented to targets in August 2021 and a website using a DHL brand abuse that was presented to targets in September 2021. The lure is generated per target as it shows the real phone number ("Your phone number") to

increase credibility and hence the probability for the target to fall for this scam. A message instructs the target to download a third-party application by clicking a download button. As of September 2021, these downloads are either Flubot or in some cases Teabot. The secondary payload is often distributed through on another (hacked) server as a proxy.

The lure servers are just hacked WordPress instances. The Flubot gang has access to a considerable amount of hacked websites that they utilize in their smishing SMS campaigns. Blocking the domains of these websites is not feasible as they may change as often as every ten minutes. Additionally, these are legit blogs that may be cleaned up in the future, which would require a timely unblocking. Telekom Security verified in more than 300 cases that the hacked websites were WordPress blogs. Furthermore, it seems that these domains are not shared across their geographical campaigns, i.e. the sets of domains per country they operate in are likely to be disjoint.

The lure servers host a heavily obfuscated PHP file (e.g. called “[a-z].php”). Its main task is to contact another server of the Flubot infrastructure, generate the lure, and forward the target to the payload APK once they’ve pressed the download button. As stated in the previous section, there is a personalized ID in the links: hxxps://some-url[.com/k.php?PERSONALIZED_ID. On the one side, this personalized ID serves as a way to show the correct phone number of the target in the lure website. On the other side, it likely serves to track campaign success. The download is proxied through another server.

Figure 3 Lure website with Voicemail theme presented to targets in August 2021.

Figure 4 Lure website with DHL brand abuse as presented to targets in September 2021

Overview of the Payload Distribution

As of September 2021, Telekom Security has observed how Flubot distributed two different malware families: Flubot itself in most of the cases and sometimes Teabot (also known as Toddler or Anatsa), another Android banking Trojan.

The names of the distributed APKs – be it Flubot or Teabot – consistently follow the same pattern for several weeks. Most time of August 2021 the distributed APKs had a Voicemail theme, which was in line with the smishing SMS the victims received (e.g. “Notification: (1) new voice message:” as observed on 2021-08-24 in Ireland). Each APK was called “Voicemail{SMALL_INTEGER}.apk”, where SMALL_INTEGER was in the range of 1 to 100 typically. In late August 2021, Telekom Security observed another shift in the lure theme. They shifted back to the parcel theme (e.g. “Visit LINK to manage your delivery. Your order ORDER_NUMBER will BE delivered SOON.”), which they utilized during the first and second quarters of 2021 mostly. Distributed APKs were called “DHL{SMALL_INTEGER}.apk”.

Flubot's distribution servers utilizes hash busting. This applies to Flubot and Teabot payloads. Hashes of the packed APKs change every hour up to a couple of days as of August 2021. However, there are counter-examples where hashes where distributed on one day and then on another day. There could be several explanations for this behavior like the generation of a set of binaries and rotating them or caching issues. Hashes appear to be always different in each country they operate in. For instance, the hashes of APKs distributed in Germany and the Netherlands were always different. The following table illustrates this using examples from several countries. The two examples of Germany and Ireland show how hashes reappear within an hour. However, the Australian example serves as a counter-example that hash busting is conducted every hour since one hash appeared on two different days.

Country	Hash	Date
Germany	ecd12174b28729a0b8c708c14c0a086b	2021-08-21-02:13:47
Germany	ecd12174b28729a0b8c708c14c0a086b	2021-08-21-03:04:13
Ireland	6a75deb9e909ae8a6ef836cf232ae8f2	2021-08-24-17:31:31
Ireland	6a75deb9e909ae8a6ef836cf232ae8f2	2021-08-24-17:56:20
Australia	ff772d18979f1e9d70f3324b3e1a25e6	2021-08-22-16:08:05
Australia	ff772d18979f1e9d70f3324b3e1a25e6	2021-08-22-17:14:31
Australia	ff772d18979f1e9d70f3324b3e1a25e6	2021-08-24-03:00:28

However, the hashes of the unpacked payloads do not change that often. The unpacked payloads are regenerated every couple of hours up to a couple of days per country. Let's visualize this with an example from Australia. We obtained 62 packed payloads with unique hashes from within Australia during the period from 2021-08-19 until 2021-08-24. The unpacked payloads have only a handful of unique hashes. We were able to observe only some of them within a couple of hours. However, we observed the hash acb9cc224edb2c834a58912ed5e97a31 for more than one day. Another interesting fact is how the unpacked hashes acb9cc224edb2c834a58912ed5e97a31 and 867329419ab81b51ded9040352ba8717 were distributed during the same time period.

Unpacked MD5 hash	From	Until
81777f62d66f59a1aba5d006836ef080	2021-08-19-18:19:58	2021-08-20-10:17:06
062847f8333e235813ca0fdc3a50650a	2021-08-20-14:04:30	2021-08-21-07:09:56
c6e0f2808d9a5062a4b8ed64445ca36d	2021-08-21-09:38:04	2021-08-22-11:10:15
acb9cc224edb2c834a58912ed5e97a31	2021-08-22-16:08:05	2021-08-24-03:00:28
867329419ab81b51ded9040352ba8717	2021-08-22-20:29:34	2021-08-23-18:57:40
bd72a3dcd754d36cc097563a7b65b7d5	2021-08-24-12:18:09	2021-08-24-14:05:51

Keeping the Botnet Running BUT Without Security Researchers

From a botmaster's point of view, security researchers are an unnecessary evil who threaten their operation. What they really hate but hardly can thwart are bot emulations, i.e. partial emulations of a bot's networking protocol that query real command and control servers, for instance, for commands and secondary payloads.

Botmasters implement several mechanisms to thwart security researcher analysis attempts. These mechanisms can be on the client side or on the server side. An example for client side anti-analysis mechanisms is application packing. And an example for server side anti-analysis mechanisms is a long wait time before the first task is served to a bot in order to shut out short running sandboxes from receiving tasks immediately.

Apart from the usual application packing that Flubot's operators utilize (as seen in the previous sections), they've implemented a particularly interesting way to check if a new bot can send SMS and hence it isn't a bot emulation. But how does this exactly work?

Periodically, bots query the command and control (CC) server for new smishing SMS tasks via the GET_SMS command. The CC server responds with a destination phone number and a SMS text (e.g. "New voice-message jecoived: hxxp://fyqz[.vip/m.php?REDACTED"). However, Telekom Security observed how this behavior changed in August 2021. Instead of receiving smishing tasks from the beginning, new bots that queried the CC with GET_SMS for the first time received something else.

In place of a valid SMS text that included a link to a lure server, the SMS text became a random string consisting of lower ASCII characters and numbers (e.g. “xxqgx323550k09yhdplziqnuwkv58r17cmtqe838475r4ynbmgm4qsz9yg7”). In case a bot is blocked from sending out SMS but requests new tasks via GET_SMS, then it’ll receive another task with such a random string but never a valid smishing task.

The destination phone numbers of these invalid smishing tasks are an interesting piece of information here. Looking up several of them in who-called-me portals reveals that most of these numbers are known to send out smishing SMS. The following figure shows two example lookups. The user comments in these screenshots sound like a general description of a Flubot infection (e.g. “text with a link”). So let’s assume that these destination phone numbers are indeed Flubot infections.

So what’s the point of sending a SMS with some random gibberish to a destination phone number that is likely to be infected with Flubot as well? Well, if we don’t think of the string as gibberish but as of some form of token, then this is clearly a verification protocol. It verifies if a bot can send out SMS to another bot in the botnet. The CC server detects this because the bot of the destination phone number logs SMS to the CC server with the LOG,SMS command on incoming SMS. This enables the CC server to verify that the destination phone number indeed received the token sent out by the source phone number. Once a bot completes this verification protocol, it receives valid smishing tasks via the GET_SMS command.

In conclusion, the Flubot operators use their botnet infrastructure to verify if new bots can send out real SMS to already known bots. This proves that new bots are not a bot emulation. As a consequence, the bot is deemed valuable to receive valid SMS tasks via GET_SMS.

Figure 5 Screenshots from www.phonenumbers.ie and www.wemgehoert.at with comments on two Flubot infections that the botnet utilized to verify further infections.

Conclusion

In this blogpost, we’ve seen how Flubot’s smishing campaigns work, how the operators circumvent simple SMS content filters, and what countermeasures (e.g. geo-fencing, bot verification, application packing) they keep on adding to increase the analysis difficulty of the botnet for security researchers. This threat is a huge but often overlooked problem for a variety of stakeholders including but not limited to the users, the carriers, the Android ecosystem, and law enforcement. One could say that Flubot is for SMS what Emotet was for email: a spam kingpin!

Further information:

Android FluBot enters Switzerland

Incibecert flubot analysis study 2021

Prodaft report Flubot