# 24/7 managed detection, response, and expert cybersecurity services - GoSecure



This blogpost summarizes cutting-edge research that uncovers an obfuscation-as-a-service platform for Android applications. From a thorough analysis of the obfuscation techniques to comprehending the service's usage, efficiency, and potential profitability, as well as placing the service in its wider market context, the research provides a practical deep dive into the modus operandi of malicious actors attempting to complicate the work of security analysts.

*The research is the result of a collaboration among Masarah Paquet-Clouston from GoSecure, Vit Sembera, from Trend Micro as well as Maria Jose Erquiaga and Sebastian Garcia from the Stratosphere Laboratory. Two related blogs about the research can be found, one on the Stratosphere Laboratory blog and one on the Trend Micro website.*
While confined in our homes studying the interactions of individuals involved in the spread of the Android banking Trojan botnet (known as Geost), we encountered **a unique opportunity: strip naked an automated obfuscation-as-a-service platform for Android malware authors**.

Indeed, in a leaked chat log that involved Geost botnet operators, two individuals talked about an obfuscation service used to "protect" their malicious Android Applications (APKs) from being detected by antivirus engines. We visited the website related to the "protection" service (protection from antivirus engines -so basically obfuscation), which raised a lot of questions: How does this obfuscation service work? Is it automated? Does it really obfuscate applications well enough to avoid malicious applications being detected? How well is the service known in the underground community?

**And so, a new research quest began. We scrutinized this service by stripping naked its obfuscation techniques, understanding its usage, and uncovering its clients, analyzing its efficiency, and estimating the potential revenue of the administrators. **

This blogpost summarizes the quest and the findings, which will also be presented at the 2020 Botconf Conference, happening from December 2nd to 4th. The conference is virtual, and registration is free.

*To avoid tipping off the obfuscation-as-a-service operators that we exposed their service, we do not mention the service's name. However, security researchers can find the service's link by doing a ROT13 of the string sggxvg.pbz*
This blogpost summarizes cutting-edge research that uncovers an obfuscation-as-a-service platform for Android applications. From a thorough analysis of the obfuscation techniques to comprehending the service's usage, efficiency, and potential profitability, as well as placing the service in its wider market context, the research provides a practical deep dive into the modus operandi of malicious actors attempting to complicate the work of security analysts.

*The research is the result of a collaboration among Masarah Paquet-Clouston from GoSecure, Vit Sembera, from Trend Micro as well as Maria Jose Erquiaga and Sebastian Garcia from the Stratosphere Laboratory. Two related*

*blogs about the research can be found, one on the Stratosphere Laboratory blog and one on the Trend Micro website.*

While confined in our homes studying the interactions of individuals involved in the spread of the Android banking Trojan botnet (known as Geost), we encountered **a unique opportunity: strip naked an automated obfuscation-as-a-service platform for Android malware authors**.

Indeed, in a leaked chat log that involved Geost botnet operators, two individuals talked about an obfuscation service used to "protect" their malicious Android Applications (APKs) from being detected by antivirus engines. We visited the website related to the "protection" service (protection from antivirus engines -so basically obfuscation), which raised a lot of questions: How does this obfuscation service work? Is it automated? Does it really obfuscate applications well enough to avoid malicious applications being detected? How well is the service known in the underground community?

**And so, a new research quest began. We scrutinized this service by stripping naked its obfuscation techniques, understanding its usage, and uncovering its clients, analyzing its efficiency, and estimating the potential revenue of the administrators. **

This blogpost summarizes the quest and the findings, which will also be presented at the 2020 Botconf Conference, happening from December 2nd to 4th. The conference is virtual, and registration is free.

*To avoid tipping off the obfuscation-as-a-service operators that we exposed their service, we do not mention the service's name. However, security researchers can find the service's link by doing a ROT13 of the string sggxvg.pbz*

## Step 1: Registering to the Service: Surfing the Underground

When we encountered the link, we attempted to register on the platform. However, as shown in Figure 1, we needed a coupon code, creating barriers of entry for curious individuals who were not serious customers, like us.
Figure 1 - Registration Webpage for the Obfuscation Service
Using the Sixgill and Flared Systems darknet monitoring platforms, we found coupon codes on several underground forums. Indeed, the service was advertised on HackForum, xss.is (previously DamageLab), procrd.top and the closed/registration forums alligator-cash and exploit.in. Figure 2 displays an advertisement for the service on the HackForum platform.
Figure 2 - Advertisement for the Obfuscation-as-a-Service on HackForums
As mentioned in the advertisement, the APK obfuscation service is supposed to offer a "fully automated service for protection of Android applications". It says it has an API integration and offers a wide range of obfuscation techniques to avoid detection and thus enhance the protection of APKs.

We used a coupon available in the advertising posts to register for the service. The service was available both in English and Russian (although only the Russian version worked properly) and the prices for obfuscation varied depending on the bundle chosen, from USD 20 for one APK to unlimited APKs for USD 850 per month.

## Step 2: Obfuscation and Deobfuscation

To uncover the obfuscation techniques, we selected the "three-applications bundle for USD 50" and obfuscated three applications; they are displayed in Table 1.

The Android Locker and SMS-Stealer were chosen for two reasons. First, their source code did not use any obfuscation, so it was easy for us to interpret the application's behavior. Second, they were both flagged as highly malicious on VirusTotal, which meant that it would make sense to use the service from a malware operator's perspective.

The third adware application was chosen because its malicious behavior was not obvious. When uploaded on VirusTotal, only one antivirus flagged it as adware. We wanted to assess if we would see different results depending on the maliciousness of the applications.

| **APK** | **SHA1 - Original** | **SHA1 - Obfuscated** |
|---|---|---|
| **Android Locker** | a48fea41f84dc357ff164b7f2f35e8f09bb8305d | 3d81adfef37e817ceb0a45d62d314af1eba27374 |
| **SMS-Stealer** | 98bb4315a5ee3f92a3275f08e45f7e35d9995cd2 | d9872e32b5f4cda4aea7beed32ae3f23c753987b |
| **Adware** | 4c3a1103960780cc890831280b37ea3a20754fad | 494e7942be0ca873ea49e5cf33bed10aa1e7faf7 |

Table 1 - Original and Obfuscated Applications Comparing the results of the three APKs, we noticed that the obfuscation was the same, which confirmed that the service was automated. Thus, for simplicity purposes, we present below the results for the first APK, the Android locker. ### Step 2.1 Automated Analysis to Compare Android Locker Original and Obfuscated Files

We started by conducting an automated analysis using MobSF to compare the results between the original Android Locker and the obfuscated one. The results are presented in Table 2.

The obfuscation process was good enough to hide vulnerabilities found in the original file, thus increasing the security score of the obfuscated file to 100/100, compared to 75/100 for the original file.

The service also changed the Package and Main Activity names to random strings, making the obfuscated file look suspicious at first sight. The number of services and activities also increased in the obfuscated file compared to the original one. The number of activities went from 2 to 10 and the number of services went from 1 to 3, making the obfuscated file more complex and potentially running more tasks in the background. Overall, the automated analysis hinted that the obfuscated file may look more suspicious than the original one.
Table 2 - Automated Analysis Comparing the Android Locker Original File with the Obfuscated one

### Step 2.2 Reverse Engineering the Obfuscated Locker Application

We were fortunate enough to have a great reverse-engineer in our team: Vit Sembera. Moreover, Vit had already uncovered part of the obfuscation process in March 2020 when he investigated an application related to the Android banking Trojan Geost.

At the time, we all thought that he was reverse engineering a Geost application, while what he was doing was: figuring out the obfuscation service to access the Geost application afterward.

In terms of the reverse engineering results, the service creates a Dalvik executable dropper and launcher (first stage) that opens and decrypts a second stage Dalvik executable (named radio.ogg and located in a /tracksfolder in all

obfuscated samples), representing the original APK with partially obfuscated symbol names.

All strings in the first stage are encrypted with the RC4 algorithm using a hard-encoded private key that changes on each APK. The obfuscated APK makes a special effort at hiding the RC4 decryption code and the second stage loader. Fortunately, the RC4 function does not keep its internal state, making it easier to decrypt all the strings separately after the key is found.

The various obfuscation techniques, to prevent a malware analyst from finding the second stage Dalvik executable, include complex string splitting, the use of decoy strings and methods to lure the analyst on low-hanging fruits, as well as nested junk flow control structures that increased the amount of code to analyze by a factor of nearly 200.

**For further information, Vit summarized the reverse engineering results on the blog titled From Geost to Locker: Monitoring the Evolution of Android Malware Obfuscation.**

Following this analysis, we wanted to dig deeper, which led us to expose the service's popularity, efficiency, and potential profitability.

## Step 3: Exposing the Service's Potential Popularity, Efficiency and Profitability

### Service Popularity

Through the reverse engineering results, it was easy to fingerprint the service. We leveraged the power of VirusTotal and conducted two Retrohunt jobs that asked:

*Are there files currently submitted on VT that are: "apks", "zip" or "jar" with an embedded "radio.ogg" file?*

**Yes! There are!**

The first job was launched on June 22nd, 2020, and another on October 14th, 2020. The first job yielded 2,172 files, and we could decompile 1,051 of them. The second job yielded 2,051 files from which we could decompile 2,006 of them. We thus had 3,057 APKs potentially related to the service.

**Are these Obfuscated APKs? Validating the Dataset**
We decompiled all the applications and extracted information on each of them. We found that all manifest files had Package name, Main activity as well as activity and service names that were long random strings just like our own obfuscated files. The radio.ogg file was also stored in a /tracks folder for all of them, just like our three applications. We concluded with confidence that these applications were all related to the obfuscation service. For future reference, their SHA1 hashes are available on GitHub.

**Yes, these 3,057 applications are related to the obfuscation service!**

Figure 3 shows the number of applications found on VirusTotal through time. The x-axis represents the latest date an application was last scanned for analysis on VirusTotal. The flat line represents the pause between the two Retrohunt jobs.
Figure 3 - Applications Scanned on VirusTotal related to the Obfuscation Service
Figure 3 illustrates that applications related to the obfuscation service are submitted to VirusTotal almost daily, with the peak 216 applications happening on July 24th, 2020.

## Uncovering the Clientele

Leveraging this dataset, we aimed at digging deeper: are these applications from the same group? We noticed that specific files in the /res/values folders of the obfuscated applications leaked information (such as strings.xml and ids.xml files). In these files, the variable names were randomly generated, but specific strings displayed in the applications were in clear text, as shown in Figure 4. The structure of the ids.xml files also did not change.
Figure 4 - Extract of strings.xml File of an Obfuscated Application
We leveraged these files, extracting the first strings of characters displayed in the "strings.xml" file and evaluating the structure of ids.xml files. We then grouped APKs together if they had the same first string in strings.xml or the same structure in the ids.xml file or both.

This yielded seven distinct groups and an eighth group, encompassing all outliers. We investigated further by taking a sample of a dozen APKs from each group and analyzed them in apklab.io. Apklab.io is a mobile-threat intelligence platform created and maintained by Avast, that displays results of APK dynamic analysis.

From each group, the dozen of APKs analyzed via apklab.io ended up behaving similarly: connecting to the same domains or to similar domains (like ccc1ccc.ru and bbb1bbb.ru) and once the second stage was launched, exposing similar patterns of behaviors.

Table 2 briefly presents the results for each group, including the application names (most likely faked), the number of APKs in each group, and some insights from the dozens of applications investigated. Obviously, we cannot confirm that each group found in the sample represents a specific client using the service, but the preliminary analysis shows that the service is used by several actors involved in spreading malicious applications related to malware. For example, the fourth group represents APKs that connect to the rakason.ru domain, which was found to be related to the flexnet malware, and one group of APKs is associated with domains that were linked to another Android Banking Trojan botnet. The seven groups could also represent several types of APKs belonging to one client.

What we can conclude from this analysis is that these APKs do not belong to thousands of clients, but rather a few number of them (most likely less than ten).

| **Group** | **Application Name** | **N. APKs** | **Insights on samples investigated** |
|---|---|---|---|
| **1** | Flash Player, Instagram Shared | 1,697 | Samples investigated communicated with DNS address static.66.170.99.88.clients.your-server.de via HTTP |
| **2** | Sistem Güncelleştirmesi (System Update) | 416 | Samples investigated connected via HTTP to one of these domains: orucakacdkkaldi.com (104.217.127.209), ba2a.com (108.187.35.84), selammigo34.com (34.91.209.109) and gunaydinmorroc.com (104.217.127.131) . |
| **3** | Android Guncelleme (Android | 251 | Samples investigated connected via HTTPS to one of these domains: hnoraip.world, kalyanshop.best, dontworryman.club, Placeoftomcat.club. All hosted on IP 46.227.68.99. The domain |

| | | | |
|---|---|---|---|
| | Update), Browser Guncellemesi (Browser Update) | | kalyanshop.best was associated with an [Android Banking Trojan] (https://twitter.com/rebensk/status/1269233752397557760). |
| **4** | МОД (много денег) - MOD (a lot of money) | 49 | All the samples communicated with the domain rakason.ru (81.177.139.80) via HTTP. These APKs seem to be related to the [flexnet] (https://twitter.com/verovaleros/status/1268244469251543042) malware. |
| **5** | FlashPlayer, Romance Mod, Spotify++ | 115 | Samples investigated connected to ccc1ccc.ru, eee5eee.ru (both hosted on IP 194.58.112.174) and twitter.com via HTTPS. |
| **6** | Flash Player, GoogleGPS, Android Guncellemesi (Android Update), Google Update & more | 462 | Some APKs did not connect to any domains; others connected via HTTP to 217.8.117.15. |
| **7** | Notification (sms app) | 4 | Samples investigated connected to 142.250.102.188 and myluckycorp.com (107.161.23.204, 209.141.38.71 and 192.161.187.200) |
| | Other | 60 | Various APKs |
| | Install (Android Locker), Swimming Pool (Adware), Spy Mouse (sms stealer) | 3 | *These are the three APKs we submitted to the obfuscation service and uploaded on Virus Total.* |

Table 3 - Obfuscated Applications Grouped

### How efficient is the service?

Then, we assessed the service's efficiency by looking at the rate of detection in VirusTotal. We first compared the original and obfuscated files related to our investigation. The results are shown in Table 5.

| | **Original** | **Obfuscated** |
|---|---|---|
| **Android Locker APK** | 27/65 engines | 16/65 engines |

| | | |
|---|---|---|
| **SMS-Stealer APK** | 29/65 engines | 13/65 engines |
| **Adware APK** | 1/65 engine | 10/65 engines |

Table 5 - Antivirus Detections Obfuscated and Non-Obfuscated Files As shown in Table 5, the Android Locker and the SMS-stealer APKs were detected by 42% and 45% of the antiviruses respectively. Using the obfuscation service decreased the detection to 25% and 20% respectively. Thus, the service is efficient at reducing the detection rate when the file is malicious by nearly half. On the other hand, the detection rate for the adware APK increased from one to ten.

Seeing an increase in detections for the adware APK raised questions. Thus, to investigate further, we obfuscated an application that was absolutely not malicious: the function of the application was to print "Hello World". **The original application had no detection, but once obfuscated, 8 antiviruses out of 65 flagged it as malicious**, even tagging it as a 'dropper' and a 'banker'. Such findings confirmed that non-malicious applications could see their detection rate increase when using the obfuscation service, leading us to hypothesize that the potential clients of the service are individuals involved in developing highly malicious applications. We then looked at the applications found on VirusTotal and inquired: **are these APKs flagged as malicious as well? **

**Turned out that yes, without a doubt!**

They were all flagged as malicious, the minimum number of detections was 8, just like the benign application above, and the maximum number of detections was 32. On average, the APKs found on VirusTotal were flagged as malicious by 18 antiviruses (with a standard deviation of 4.79). Figure 6 shows the range of detections depending on groups.
Figure 6 - Detection Variation by Groups

Figure 6 shows that the average detection rate differed depending on APK groups. To ensure that this finding was not just a visual approximation, we computed a series of tests of mean differences and found that there exist significant mean differences in detection rates between each group!

**There are significant differences in detection rates for each APK group, hinting again that these groups could represent different clients.**

## Estimating Revenue Potential of Maintaining an Obfuscation-as-a-Service Platform

One of our final tasks was to assess the service potential revenue considering only the applications found on VirusTotal. The obfuscation service offered different price categories. The most expensive option was to obfuscate one file for USD 20. The cheapest option (if one had many applications to obfuscate) allowed unlimited access to the service for USD 850 per month. Figure 7 shows the different price bundles offered in Russian.
Figure 7 – Obfuscation-as-a-Service Prices
To estimate revenues, we followed this strategy: for all groups that have hundreds of scanned APKs scattered throughout the period of study, we consider the number of months each group operated and count an API price bundle of $850 per month. For the remaining groups that have only a few APKs uploaded at different points in time, we considered the highest price: USD 20 per APK.

**Following this pricing strategy, we estimate that the operators behind the obfuscation-as-a-service platform would have made USD 22,490 for the APKs found on **VirusTotal**. **

Another strategy was to create an interval taking into account the highest and the lowest prices. Considering that all the applications would have been purchased at the price of USD 20 per APK, then those behind the obfuscation-as-a-service platform would have made USD 61,060. Considering, on the other hand, that all obfuscated applications would have been purchased with one API access through six months of operation, then those behind the obfuscation-as-a-service platform would have made only USD 5,100. The first approximation, USD 22 490**,** is situated closer to the lower bound of the interval and seems to be a better approximation of what the administrators would have made because it considers API accesses for APKs grouped together due to their similarities. It is unlikely that malware authors would pay a full price of USD 20 per APK for each file in these groups. Whether this amount can be considered substantive depends on where one is positioned in the world.

### Market Ecosystem

Lastly, we wanted to position the obfuscation-as-a-service within its market ecosystem. Using the Sixgill DarkNet monitoring tool as well as the Flare.System one, we searched potential competitors leveraging the keyword "crypt" (which was the "slang" word used by Geost operators to talk about the service). We found, since January 2020, six potential competitors, advertising "APK crypt service" on different underground forums, as shown in Table 6.

The prices advertised by each competitor were higher than the service we investigated. Moreover, none of them offered an automated platform with API access. Instead, they all asked potential clients to contact them via messaging applications like Jabber or Telegram.

These competitors seem to conduct their obfuscations manually, rather than automatically (explaining their higher prices). This also means that the obfuscation-as-a-service investigated may have had a competitive edge by offering an "automated service".

| **Service or User** | **Forum** | **Date** | **Prices** |
|---|---|---|---|
| **Competitor 1** | XSS | August 2020 | $30 for 1 APK $80 for 4 APKs (1 week) $135 for 12 APKs (1 week) $250 for 25 APKs (1 week) $300 for 45 APKs (1 week) |
| **Competitor 2** | XSS Club2crd Dark Market Devil Team CenterClub | January 2020 | $20 for 1 APK $100 for weekly submission with max 10 APKs/day |
| **Competitor 3** | Club2crd | August 2020 | $100 for 1 APK |
| **Competitor 4** | Hackforums | July 2020 | $25 for 1 APK $70 for 3 APKs $99 for 5 APKs |
| **Competitor 5** | Ufolabs | October 2020 | $30 for 1 APK $150 for 4 APKs (1 week) $350 for 12 APKs (1 week) $550 for 25 APKs (1 week) $1000 for 45 APKs (1 week) |

| | | | |
|---|---|---|---|
| **Competitor 6** | SKYNETZONE CHAT telegram group | November 2020 | $150 for 4 APKs (1 week) $350 for 12 APKs (1 week) $550 for 25 APKs (1 week) $1000 for 45 APKs (1 week) |

Table 6 - Obfuscation-as-a-Service Competitors

## Conclusion

Overall, we conclude that the obfuscation-as-a-service platform provides a medium quality obfuscation service. Indeed, a lot of work has been put into automating the obfuscation process, yet a few mistakes made it easier to fingerprint the obfuscation. We also believe that the platform's clientele is formed of individuals developing highly malicious applications. Indeed, the service is only efficient at reducing detection for such applications.

Currently, the platform has been down since late August 2020. Yet, obfuscated APKs with the same obfuscation techniques are still being uploaded on VirusTotal as of November 2020. Possibly, the applications scanned by VirusTotal are applications obfuscated before the platform's shutdown or the operators are still active without the web platform.

As a last note, we hope that this work can be helpful to security analysts and reverse engineers who face obscure applications every day.

*The hash of each obfuscated APK found on *VirusTotal* is available on *GitHub*. *

---