

Chapter 1 — From Gozi to ISFB: The history of a mythical malware family.

 medium.com/csis-techblog/chapter-1-from-gozi-to-isfb-the-history-of-a-mythical-malware-family-82e592577fef

Benoit ANCEL

October 24, 2022



--

Disclaimer:

This article does not contain any IOCs or infrastructure details. Instead, the aim is to explain the whole business dynamic of a long-lasting malware family. This work is based on almost 10 years of research and intel gatherings and tries its best to stick to the truth and the facts observed around ISFB. Hopefully, it will give some insight on how the top cyber crime groups have been working over the years.

A 10 years journey

For the last 10 years, a certain malware family has caused lot of ink flow and left a lot of people confused: ISFB. With this series of articles, I aim to illustrate the whole journey from the early start, over the leak of Gozi 1, to the recent mutation of ISFB into LDR4. It is a very long and intense journey to describe, so I will try my best to be as rigorous as possible to finally document what ISFB has been doing since 2012.

In chapter 1, the first steps of our journey will describe where ISFB originates from. Its history has caused extreme public confusion, so we will break down the different branches of ISFB step-by-step and explain how we end up in the current situation of 2022.

Once the ISFB ecosystem is uncovered, I will focus on the business and the threat actors in a 2nd chapter. With the technical analysis of the different ISFB branches being already extremely well documented, I will go through years of daily operation of the ISFB crew, from the developer organisation to the affiliates and their connection. That will address some unanswered questions and show how the ISFB groups are tightly connected to high ranking groups like Evil Corp.

What is ISFB?

First of all, what is ISFB? ISFB is a malware family encapsulating a whole set of tools primarily used to defraud online banking accounts; it is a banking trojan. ISFB is not the first of its kind, the features available are somewhat like the Zeus trojan, just way more advanced.

ISFB evolved over the years, and we have seen features like:

- : the malware can collect local information about the victim computer, OS type, IP address, computer name, list of Anti-Virus applications, check if the computer is attached to a Domain Controller etc.
- It can load a 2nd stage attack as EXE or DLL or execute any command through CMD/Powershell.
- : ISFB can steal the clipboard and every keystroke from the infected PC. Furthermore, the malware injects itself into the web browser and capture every HTTP POST request, which makes credential and stealing credit card easy.

- The main feature of the module is injecting code into the web browser that will monitor which websites the victims are visiting. If a banking website is identified, ISFB injects a small snippet of JavaScript into the online banking website to steal the login credentials. This is commonly known as web injection, a technique commonly used by banking trojans. ISFB introduced an evolution to that attack with their Replacers. ISFB injects the web browser and checks for opened websites, if it identifies a bank from its configuration file, instead of injecting JavaScript, ISFB redirects the victim into a cloned website of the banking login, completely controlled by the ISFB operator while keeping the legit URL in the address bar. That allows the attacker to have direct access to a victim and for example bypass 2FA without having access to the victim's smartphone.
- To assist the web-injects/replacers, ISFB offers a VNC and reverse socks proxy module. Once the operator has obtained the online banking credentials, they need to log into the account while bypassing the anti-fraud protection. To achieve this, they wait for the victim to be online, and if the anti-fraud protection is only based on the IP, they use a socks proxy. If the anti-fraud is based on browser fingerprint, IP, and behavior (like most modern banks), they would use VNC and login directly from the victim computer remotely.
- ISFB can record the screen if a victim is visiting a specific website. The fraudsters can then study how a victim is usually moving money with the online banking account. Once they have gained enough knowledge about a target, they can reproduce the same behaviour to move money without triggering any suspicious behaviour anti-fraud mechanisms during the transfer.
- : not all the ISFB forks have it, but the email stealer is an important component. The stealer is used to detect email software like Outlook and steal the credentials and the contacts list. The operators are known to use that feature to maintain a constant feed of fresh login credentials for SMTP accounts, plus a list of valid emails to spam. Those corporate contact lists often include clients and partners of a company, which in turn, gives the spammer high value spam targets.
- A module that makes exfiltration of specific files from the victims hard drive possible. It is commonly used to steal BTC wallets as an example.

ISFB was created to defraud online banking but operators often use it as an entry point for a second stage attack such as ransomware or extortion.

Ursnif, CRM, Gozi, ISFB... what is going on ?

To understand where ISFB has spawned from you have to go back to the CRM/Gozi malware, the Kuzmin Gang, and Service76. I am not going to dig into the Gozi story; however, you can find great documentation from [phishlabs](#) article or through the very interesting podcast Malicious life ([part1](#) and [part2](#)).

Long story short:

- Gozi v1 (Also called CRM1) was born around 2005 by combining several other malwares (Ursnif was one of them). The malware is part of an operation from the Kuzmin Gang.
- In 2010 The Kuzmin Gang evolved Gozi v1 (CRM1) into Gozi v2 (CRM2) that 2 years later became “Gozi 2 Pranimalka”, also called Vawtrak.
- When the Kuzmin Gang moved from Gozi v1 to Gozi v2, the v1 source code was sold, and later in 2010 the source code was leaked.

Let's take 2010 as a starting point. The Kuzmin Gang launches the big update of Gozi v2 and as usual in the cyber-crime industry, once they moved to the v2, they sold the source code of the v1. During that transaction, the source code unfortunately leaked due to a non protected download link. It doesn't matter much for our story who originally bought the Gozi v1. The important part is that the Gozi v1 source code was sold, somebody snagged the code, rewrote it, and started a new empire: ISFB.

ISFB was born in 2011/2012 as a highly modified version of the source code of Gozi v1. Over the years, the code has moved further and further away from the original Gozi v1, but this is where our story begins.

The infosec industry created several conventions to name ISFB, leaving the developer of ISFB quite irritated. During a discussion between a client asking for anti-reverse features and the ISFB developer, the developer said: “If they still call all the branches Gozi, although from Gozi there is only a request format with ID, group and server number [...], we don't need anti-reverse”. You may have heard of ISFB through the names URSNIF, PAPRAS, GOZI or even sometime ROVNIX, but, really, it is most of time ISFB. The signature names don't matter much in the end as long as the threat is blocked, but to easier understand our story I will call them by their internal names.

ISFB is born

Around 2011, a very skilled system developer is looking around for a new project. For some reason he is looking for a fresh start in the cyber-crime industry, he has solid development skills and luckily for him, the Gozi v1 source code had recently leaked. Let us call him ISFB_Coder.

ISFB_Coder retrieved the code of Gozi v1, started editing it, and created a whole business around it.

He started by versioning the code as if the fork of Gozi v1 was “CRM2” or ISFB. Both those names are officially still inside the source code today, and lots of references to CRM can be found. Note that it is not the same CRM2 as Vawtrak!

This whole phase of rewriting the code of Gozi v1 took a serious amount of work. We had to wait until 2013 to discover the first traces of ISFB in the wild. A fair amount of the gap between 2011 and 2013 was for sure due to the development and the testing of the malware, but also due to a very probable private usage of ISFB_Coder's for a few times.

ISFB Versioning: the nightmare

Once ISFB_Coder was ready to open his product up to partners and make way more money, he took ISFB (CRM2) and split it into several branches over the years.

ISFB is not one unique product, it's a range of products. Each major versions has its own features. Some ISFB versions are exclusive to one group, other branches have been totally resold as a service like IAP or the public branch of Dreambot. The main thing that ISFB_Coder tries to avoid is to share the source code of the bot. Sharing too much of the source code inevitably leads to a public leak, as he previously experienced in 2015 when one of his partners publicly leaked the source code of a branch of ISFB.

We suppose that ISFB versions started at version 2.00.000 up to now 3.x.xxx. Each major versions has its own features and each major versions has its own subbranches with custom features.

Some major branches as example:

- Used to be the IAP/Dreambot early branch after 2.12.xxx, with the first integration of Tor onion as available C2.
- The bot uses content compression (gzip).
- Uses a loader instead of a direct DLL.
- Used to be an evolution of 2.16 mainly (but not only) used by IAP2 and the end of Dreambot.
- Originally meant as a loader for WastedLocker, the code ended up used by the IAP2.
- complete rewrite of the network protocol, this branch was — and still is — the most exclusive and prestigious one.

Each major branch has subbranches, some private, some public, and despite having for example the branch 2.17, some customers still prefer to stay with the 2.15 or 2.16. Not everybody moves to the latest versions.

To recap, ISFB_Coder is a development company, developing a product that he also makes use of personally. The core of his product is ISFB, and he offers custom features for trusted partners across a wide range of ISFB versions.

In addition to this fact, we know ISFB is not the only product on ISFB_Coder's shelves. As for example, he sold a ransomware in 2019 (WastedLocker) and his coding style is noted around other malware families like Caberp, but for our story, will stay with ISFB.

In the following graph you can see the major branches of ISFB. The graph does not cover other projects that spawned out of the ISFB leak in 2015 (Saigon, Goznm...) or the small groups using the malware, but it covers the most active campaigns around ISFB_Coder.

The important thing is: You cannot do threat intel on ISFB by solely looking at the bot versions. Completely different group of actors use the same versions, at the same time, and if you want to understand the threat correctly, you must look deeper into the binaries and beyond. Small details like for example the algorithm of the config encryption distinguish different versions. Some used RC6 some Serpent for the same bot version.

I have tried to recap the different versions for the major branches over the years:

Please bear in mind that we only cover the major branches. We have observed other actors using the v3, for example in Switzerland or Japan, or even a Goziat v3. Those campaigns being relatively small, and that story being long and confusing, I have chosen to leave them out for now.

CRM2: v2.xx.xxx

As explained, ISFB_Coder started his ISFB business from a fork of Gozi 1, that was branded as CRM2. At that time, CRM2 was the main and only version of ISFB available. CRM2.5 and CRM3 were both much later.

The plan was to have a core (CRM2) and to adapt it on demand for different partners. He wanted both private (exclusive) versions and an open “as-a-service” version, kind of like Zeus was back then.

The “as-a-service” version is the less advanced version and has been sold to so many different people over the time that it has become very complex to map out.

IAP /Dreambot/IAP2

Example of sample IAP:

```
ffcb650b28719d3bde1b032b14cfe7f5d7f2a73878d752737da0ba8a4f8bb70c
```

ISFB_Coder being a system developer, he is not building a panel for ISFB. He will sell you the bot, documentation for the API and then you must create a panel by yourself.

For a malware-as-a-service, as they tried to do it, that is an obvious problem. In 2014, the clients want a bot with a panel included. That’s where IAP came alive.

IAP, discovered in 2014, is the name of a panel project built to control ISFB bot’s version CRM2. In September 2014, Yurii Khvyl wrote a [blogpost](#) exposing the installation manual of IAP:

To develop this panel, ISFB_Coder partnered with a web developer. That developer was in the close circle of the ISFB core group, and he ended up developing several panels for ISFB. The one for IAP as mentioned but also the one for the “Global Network” branch. The developer was not really appreciated by the ISFB core groups, described by some of his partners as “a unique kind of failure”, and ended up being arrested in 2016 in Kiev for his work on ISFB. His arrest involved a shady story implicating ISFB_Coder and the partner owning RM3 but to jump to the point: Betraying the ISFB group leads to jail time.

IAP panel looked like this back then:

This version was very popular between 2014 and 2016 and has been seen in a lot of different countries around the world. IAP was not the first ISFB version used in the wild, but it was the first one to gain large attention.

Managing IAP took a lot of energy, so ISFB_Coder delegated the responsibility to a 3rd party manager to deal with the business and bug reports, but with the panel developer doing jail time, the project took a hit, and the business was in danger.

Meanwhile, another version of CRM2 had been detected in the wild. It was an exclusive partner version from 2013 that offered Tor onion C2s. This actor was looking to replace IAP and called it Dreambot.

Example of sample:

```
7e0bf604d3ab673a519feb5d5375f0f88cf46e7cd1d3aa301b1b9fb722e9cef7
```

The partner who owned Dreambot asked for a fork with small features on demand and then opened the previous version of Dreambot for public sale.

IAP and Dreambot were so alike, that at some point, the IAP bots were able to join Dreambot panels without any problems, because the malware network communication was exactly the same.

Dreambot is a good example of the capabilities of ISFB. The partner who owned Dreambot requested CRM2 to support Tor. ISFB_Coder strongly disagreed, explaining that deploying the Tor lib would make the AV detection rates explode compared to the clear web C2.

The Dreambot owner is targeting US and CA, he is used to receiving a huge amount of abuse reports on his C2 and he really needed that Tor option. The Dreambot owner had decided, with the help of the ISFB API Documentation, to develop his own Tor lib for ISFB and he incorporated it into Dreambot. ISFB is made in such a way that developing your own module is quite easy and well supported by ISFB_Coder.

That double version of Dreambot (private/public) was already observed by [Proofpoint](#) in 2016:

The config shown in the screenshot is for Dreambot private edition.

The public demand for the Tor lib ended up being so high that the Dreambot owner decided to give the source of his Tor lib to ISFB_Coder for free, for integration into the official sources. Peer pressure made ISFB_Coder bend on that one. We observed another ISFB partner with his own branch switching to Tor for the C2s and announcing losing around 30% of his bots due to Tor detection in corporate environments.

IAP being dead, it was easy for Dreambot to fill the gap as the “as a service” branch with a better UI for the command & control server.

The first version of the panel was developed in Perl, and it is highly suspected that the developer of the IAP panel helped with the first Dreambot panel. This panel was first disclosed by Maciej Kotowicz at Botconf 2017 in his talk ISFB, Still Live and Kicking

The first version of the panel:

It evolved into a better version, still in Perl:

Ending up as a completely re-branded panel in PHP:

We tried to give some insight into several of the Dreambot affiliates, but the turnover being too high, and the affiliates working with Dreambot end up being too far away from the core group to be interesting for this story. That doesn't undermine the colossal damage done by these affiliates over the years all over the world though, from banking fraud to ransomware attacks.

The story of Dreambot ends in 2020. The whole branch evolving from 2.12.xxx in 2014, to end-of-life in version 2.17.xxx. The IAP project came back to life in 2018 as IAP2 and later as 2.5, and after a test phase, all of the affiliates moved to IAP2 letting Dreambot to die in peace.

IAP2: back for good

Example of sample:

b74327fb49965c60d3d066788c5e0ece297187944e4336d6fea79135455f62fb

While Dreambot was busy dying, another version took the lead: IAP2. IAP2 started in the wild in 2 versions, 2.14.xxx or 2.17.xxx (Standalone DLL or DLL combined with a Loader) and has been extremely active in Poland, Germany, and Italy, but also USA or Canada. It is still distributed by spam in Italy today.

The first version of the panel was actually 2 panels, one for the loader (called “Lodiri”), and another for the worker (called “Newadminka” dealing with webinjects, VNC etc). The loader was deployed by the affiliates themselves, making it extremely exposed and vulnerable. The

affiliates being usually sloppy people that don't care much about the malware they use, they often end up deploying the panel in insecure way or simply leaving the source code in a panel.zip file.

If you monitored ISFB between 2018 and 2020 you will probably remember all the samples with the static encryption key "10291029JSJUYNHG". That was IAP2.

IAP2 "Lodiri" panel:

IAP2 "Newadminka" panel:

As for IAP1/Dreambot, the management was delegated to a 3rd party.

After 2 years of business, IAP2 evolved into IAP2.5 in 2020, using the base CRM2.5. This time, we noticed clues leading us to believe the developer of this panel is actually the admin of the Dreambot private / RM2 (goziAT) branch. This is the essence of the ISFB business; ISFB_Coder is the developer, but the whole empire consists of a strongly linked pool of affiliates helping each other, year after year.

The new panel of IAP2.5 in 2020 is very similar to the RM3 panel, and is based on the framework SmartAdmin. The early panel is called "Hyper". Later on it obtained the same name as the first RM3 panel: "P-II":

IAP2 was still active all around the world. In 2020 they even tried to develop their business in Colombia, but without much success.

This kind of work on new business areas is typical from the 2020 "crisis". Banking Trojan operators started to realize that collecting good volume of bots that can stay alive on a computer for long time is hard and that deploying ransomware was more profitable.

IAP2 is still today very active in the wild and we will likely be seeing it around for a while, probably until the RM3 branch is delegated to the as-a-service branch.

RM2/GoziAT

Example of sample (Dreambot private):

```
f815a76a46034e200a7be1ccc319174da6bebed8426df7adac6374b5abc94f47
```

Back to 2013. A very early adopter of ISFB is a partner with his own exclusive variant ISFB variant. Heavily involved in the whole ISFB project, this partner "Expro", is a professional developer and a quite talented botmaster. Expro started his adventure with ISFB_coder in 2012/2013 with the ISFB variant we call Dreambot private.

Expro leads a group of old-time carders whose focus is on attacking Canada and USA. Both are seen as dangerous countries that most of the other partners do not want touch.

This is an amusing thing observed among several Russians threat actors in the carding industry. They seem to be very afraid of the FBI, saying that they have big resources and they are determined. Those criminals are dependent on many different American services like Apple, Microsoft or even Google and they are very aware that the US law enforcement could have access to that data quite easily.

Combined with the fact that the US has a massive amount of cyber security companies looking for fresh IOCs everywhere, it makes the threat actors more comfortable operating in Europe or Australia like ISFB. This misunderstanding of international cooperation between law enforcement has created some kind of legend that if you don't touch the US you will have less problems with their law enforcement.

Besides targeting the US and CA, Expro distributed his malware in a very broad way, distributing large volumes (of spam, exploit kit, bundle...) to low quality targets (people with very outdated OS, gaming computers, all the anti-malware sandboxes of the industry) instead of a low volume to high quality targets.

The tactic somehow works but it caused Expro a lot of problems. As mentioned before, attacking the US in a noisy way attracted abuse reports on his infrastructure from AV companies, and so, he realized the need for specific features like Tor.

Expro is a close partner to ISFB_coder. He develops and sells several panels for ISFB like the one for RM3 or LDR4 (not for free) and is sharing resources with the ISFB partners.

The RM3 group and Expro are also very close to each other. The way they cooperate has made them work together on a daily basis since 2012. For a long time, Expro hosted his infrastructure inside the RM3 infrastructure, and both groups exchanged their experiences with ISFB. They test and report bugs together, give feedback about each others variants, unionize together to push new feature requests to ISFB_coder, they even help each other with OPSEC and backend protection issues.

You can still observe this relationship disclosed by Mandiant in the article "[From RM3 to LDR4: URSNIF Leaves Banking Fraud Behind](#)" where Expro is the developer of the LDR4 panel.

The relation between RM3 and Expro is very interesting to study. Both doing business on their own and supporting each other. In part 2, I will dig deeper into the details of their relationship, given that RM3 has such a dense life, it deserves its own article.

In 2018 Expro moved from Dreambot to a new update. [Checkpoint named it GoziAT](#), as reference to the frequent usage of .at tld used as C2.

Example of sample (GoziAt):

21a03d9c845e446cb96eba7c93aa6403b8a9aaa744801e77468bf73c0507d028

GoziAT is now the common name for Expro's branch, but if we should respect the internal naming convention, the most probable name would simply be "RM2".

The RM2 branch used to have a static CnC beacon format. The bot sends its requests to a URL path like

"/images/[encoded data].[avi|bmp|gif|jpeg]"

As mentioned in Checkpoint's great analysis, in 2020 GoziAt was using a custom path, changing time to time for example to /wpapi/, /rpc/, /wpx/...

This is because GoziAt is distributed in such a wild way, it's very quickly detected by the AV engines. Expro tried to add some dynamic to ISFB.

Checkpoint is also raising a valid point where they noticed "*These campaigns tend to hang on to the same domains and IP addresses for a relatively long time, which may not be the best choice opsec-wise*". This is explained by the fact that Expro is putting all his efforts in the onion C2s and not the clear web domains. In several campaigns none of the clear web domains were working and only the onions were up.

While Expro distributes in a very noisy way, he concentrated a lot of his efforts into hiding his backend behind a layer of proxies, making him able to survive for very long time.

With the CRM2.5 available in 2020, Expro jumped on that boat and shifted to his own branch, being basically GoziAt2.5/RM2.5.

Example of sample (GoziAt2.5):

1c2fd2e6d4f1e0e2ee23f4b9ae0ea061cc1f4b41a28ec184ce7e70d5be263e8f

Expro is continuing his journey; you can still catch samples from his botnet spamming campaigns in the US.

Global Network / RM3

Last but not least, we have the Global Network / RM3 branches. These branches are probably the most interesting group actor.

Managed by a tyrannic boss, RM3_boss, this group works with the best. If they need to send spam, they hire "Sagrid" (TA543) or TA547. If they need to cash out money, they use QQAZZ. If they need help, they call Maksim Yakubets from Evil Corp. It's a powerful group with a sub-affiliates system generating a large amount of money.

They work more or less with the same business model since day one. In 2012/2013 RM3_boss agreed to join the ISFB_coder project and bought his own branch dubbed "Global Network".

RM3_boss uses his branch in 2 ways:

- For his own fraud team. Always focusing on AU and NZ and nothing else.
- For a very selective set of affiliates always under his control.

If you want to use Global Network or RM3 from RM3_boss, you must deal with the sysadmin of RM3_boss and his infrastructure. You will never get access to any source code or server, RM3_boss provides access to a stub and credentials for a panel without root access. The whole business is based on trust, all affiliates must cash the money stolen out via the RM3_boss cash out network.

By forcing affiliates to launder the stolen money via RM3_boss, it allows him to bill his affiliates with a percentage of every fraud conducted via RM3. RM3_boss has full visibility on every fraud, takes his cuts and gives the rest back to the affiliates. He doesn't care how many victims the affiliates infect; he only cares about the stolen money. I will get deeper in these terms in the part 2, but the way RM3_boss manages his business partners is a very interesting case.

In 2013 ISFB CRM2, dubbed Global Network, was found spreading in Australia and New Zealand. The focus on this area is because RM3_boss was convinced for years that the AVs industry has no interest in these countries, despite them being very profitable targets for banking Trojans. The low media attention allowed him to stay hidden for a long time.

The Global Network panel was looking like:

By opening Global Network up to a short list of affiliates, the malware extended its propagation to places like PL, UAE, IT, DE, UK, CN, CH, and CA. The affiliates around UK and IT ended up cooperating with RM3_boss from early 2013 and are still today in 2022 with LDR4 in UK.

RM3_boss is a tyrant and a control freak; and that turns out to be the strength of the Global Network branch. He makes certain that his malware campaigns never leak publicly, and if that should happen anyway, RM3_boss would drop financial penalties on the responsible load seller. He spends time reading every news article mentioning any ISFB products, to make sure that he and only he, is using this particular version and that ISFB_Coder is not cheating on their exclusive agreement. RM3_boss seems relatively well connected and he made it clear with all his direct partners: if you try to betray him, he can send you to jail very easily.

The counter point to a work environment based on fear and threats, is that RM3_boss pays the people respecting the rules a lot, and that is how he managed to keep skilled third party people working for him.

Global Network followed the life of ISFB until 2017 where the anti-virus detection rates of Global Network (and CRM2 in general) became too good and started affecting the profits. To resolve that issue, RM3_boss put an order out to ISFB_Coder on a new variant: RM3.

He asked for a rewrite of the ISFB code and specifically the bot network part. The development of RM3 lasted from 2017 to 2018, and once ready, the panel looked like:

The move from Global Network to RM3 took longer than RM3_boss expected. In 2017 the AV detection of Global Network became way too good, and the profits of the team were dropping quickly. In order to temporarily fix this problem until RM3 was ready, RM3_boss decided to ask one of his partners for help; the member of Evil Corp: Maksim Yakubets.

Global Network had become a pain to work with and RM3_boss decided to rent Dridex for a few months until RM3 became ready. The next chapter will cover that transaction in more details, but between June 2017 and beginning of 2018 the RM3 group was observed using [Dridex botnet 2302](#).

It is not the only time that RM3_boss had been involved with Evil Corp. We will dig into that chapter two but as already suspected by Fox-IT, ISFB_Coder ends up being the actual developer of [WastedLocker](#).

After some time, RM3 also ended up being well detected by the AVs, and in 2020 the group started to seriously look for new opportunities. At this point the group was reselling bots for ransomware operations like Conti, Doppelpaymer and even Darkside.

After a catastrophic year 2021 in terms of business, RM3 was barely surviving in UK and only used for loading CobaltStrike. Microsoft removing Internet Explorer ([RM3 needs IE to work](#)) officially in 2022 killed every hope of future for the banking trojan.

The group re-emerged in 2022 with an evolution of an old loader in v2.5, a loader dubbed **LDR4**.

Example of sample:

```
2502a3f8c9a6a8681f9222e93b14e077bf879e3009571c646ee94275bc994d01
```

[Described by Mandiant](#) recently, LDR4 is the new loader used by the RM3 group. In development since 2021, the loader is finally ready in 2022. So far it is the UK affiliate that is the primary user but we expect them to expand in more countries any time soon.

LDR4 panel (you can recognize SmartAdmin framework):

First version of the panel in 2021

Update 2022 of the panel

And as mentioned by Fumiko, the username Expro is leaking from the gate domain:

Expro is the developer of the panel (based on the RM3 panel). He deploys and manages the sources. You can even find the traditional file 123.txt usually used by RM3 since at least 2018 to store the botnet name of the gate:

The LDR4 loader is composed of the usual loading features, plus VNC/Socks and keylogging modules. Ready for a more advanced 2nd stage.

The group is an amazing mix of great skills and an old way of doing things. Carders from the past trying their best to keep up with technology, but like many fragile businesses, RM3 are having trouble with a come back after the Covid-19 crisis. LDR4 seems to be their last chance now.

Conclusion

ISFB is today facing the same crisis as every other banking Trojan. The level of security in corporate environments is way higher now, and malware like ISFB is now well detected by all AVs. To commit bank fraud, you need to keep a victim infected sometimes for several days. With Windows Defender on Windows 10, or Azure endpoints deployed in corporate environments, combined with an anti-spam gateway, it is very hard for crimeware tools to stay undetected long enough to commit bank fraud. Phishing is way more cheap and profitable.

Tools like ISFB are today very outdated in terms of bypassing security measures, making the distribution of the malware very hard. ISFB_Coder still codes his products like if it was 2007 and always refuses to introduce solid anti detection measures in the core of ISFB. Using the justification that bypassing security measure is the job of the packer and not the bot. Those outdated views on the security products are now incompatible with an operation relying on bots to staying on a computer several days in a row.

The distribution of banking trojans has an extra step of complexity. The victims must have access to online banking data, which really is only a fraction of online people. Back in 2016, infecting 1000 victims a day with Global Network was a thing. In 2020, specially with COVID-19, infecting 50 victims was a good day. If the malware distributor is doing good and manages to target a corporate accounting department, the operator can only hope for around 30% of his victims to actually have access to bank accounts. If you leave out people who cannot access money or cases where the security is too high, there is not much money is left to steal.

ISFB is facing the same reality as Trickbot, Zloader, Ramnit, Dridex etc. Most of them gave up on the banking fraud part and became just loaders for loading 2nd stage attacks (often ransomware). With LDR4, ISFB_Coder tries to make his old partners stay in business with

the move towards ransomware. But it's 2022 already, and despite still being a big problem, the party is over and there is objectively not much money left on the ransomware field anymore.

IAP2, RM3, LDR4 are surviving so far, but if ISFB_Coder doesn't have an advanced tool hidden in his pockets, everything points to the end of the reign of ISFB.

I have tried through this first chapter to present the overall operation behind ISFB. The way the versions have been distributed through affiliates is not something commonly seen. ISFB_coder tries to evolve like a software development company and managed to earn his living from his developments. With each ISFB branch being sold for between 50,000 and 100,000 USD, without counting the support and the custom requests, jumping on the leak of Gozi v1 was a really smart move from ISFB_Coder.

The next chapter (coming up soon), will go deeper into the whole banking trojan crisis via the group Global Network/RM3, showing how the business of RM3_boss have evolved and what kind of issues they have faced. Infrastructure, distributors (Spam, Adwords...), cryptors, cashout, and how the defrauded money is divided between the group members. I am looking forward to presenting the actors running the whole operation, where they come from and their actual role within the organisation. We will also look at the deeper relationship between Expro and RM3_boss and I will review the different affiliates that fall under the RM3_boss umbrella since 2013.

I hope the ISFB mess is a little bit clearer and easier to apprehend in a more structured way. As much as it is not really important to publicly name a malware with its internal name, it is really important to document the structure of the groups behind that malware. Defining the global structure of a threat is a mandatory step to help law enforcement to understand the situation.

If you have any comments or leads around ISFB don't hesitate to ping me, I will be more than happy to exchange information.

Stay tuned for the next episode!

Acknowledgments

I would like to extend my thanks to Maciek Kotowicz for opening the way into ISFB intel with his inspiring work. [Kafeine](#) and Sammy for the huge work on campaigns classification and all the support given. Fumiko and Sandor Nemes for the great reversing work, [fr3dhk](#) for his patience following IAP2. Fumiko again for all his support, his analysis, the tools he provided, and the amount of time spent on this case, and of course everybody who worked around ISFB and who have allowed us to finally have a clear overview of the malware family.

Illustration: [wombo.art](#)

Annexes

Documentation:

IAP

Dreambot

IAP2

Global Network:

RM3

LDR4

Misc