Born This Way? Origins of LockerGoga

By Mike Harbison

Published: 2019-03-26 · Archived: 2026-04-05 17:57:34 UTC

This Unit 42 blog provides insights into the ransomware attacks referred to as LockerGoga.

The LockerGoga ransomware was first publicly reported in January by Bleeping Computer, which tied the malware to an attack against French engineering company Altran Technologies. Several variants have since been found in the wild, where they were used in attacks against Norwegian aluminum manufacturer Norsk Hydro and two chemical companies: Hexion and Momentive. Unit 42 reviewed malware samples from these attacks and found evidence that caused us to question the origin of the threat name. "LockerGoga" was taken from a string that did not exist anywhere in the code used in the original attack on Altran. Bleeping Computer reported that the name came from this source code path discovered by MalwareHunterTeam:

X:\work\Projects\LockerGoga\cl-src-last\cryptopp\src\rijndael_simd.cpp for SHA256 bdf36127817413f625d2625d3133760af724d6ad2410bea7297ddc116abc268f

We were able to find this string referenced in earlier ransomware variants identified as Ransom.GoGalocker by Symantec, but not in the sample identified in Bleeping Computer's report. To avoid confusion, we will continue to use the LockerGoga name to refence the initial variant and its predecessors. Palo Alto Networks has identified 31 ransomware samples that are similar in behavior and code to the initial variant. In this report we will attempt to trace back to the origin of these samples, discuss their evolution, then expose some of their inner working capabilities and even faults.

First LockerGoga Sample

The earliest known sample of LockerGoga (SHA256: bdf36127817413f625d2625d3133760af724d6ad2410bea7297ddc116abc268f) we found was submitted to VirusTotal on January 24. We believe this sample was used in the attack on Altran. We do not know how the ransomware infected Altran's networks or propagated once there.

Propagation

Currently LockerGoga does not support any worm-like capabilities that would allow it to self-propagate by infecting additional hosts on a target network. We have observed LockerGoga moving around a network via the server message block (SMB) protocol, which indicates the actors simply manually copy files from computer to computer.

LockerGoga Characteristics

This sample requires administrative privileges in order to successfully execute, though the specific mechanism for initial code execution is unknown. Once it executes it seeks to encrypt files on the infected computer and any attached hard drives. Once it's fully executed, it leaves a ransom note on the user's desktop containing an email address to contact presumably for decryption and payment options.

The initial sample was written in the C++ programming language and employs publicly available libraries such as Boost, Cryptopp and regex. This sample includes a denylist that excludes the following files and directories from encryption:

- readme-now.txt
- files starting with ntsuser or usrclass
- File extensions: .dll, .lnk, .sys, .locked
- Microsoft\Windows\burn
- dat
- log

The malware does not support any self-propagating code to infect other hosts on the network and is signed by a certificate issued in the name MIKL LIMITED, which has been revoked. The following command line arguments are also supported:

Command Line	Description
-w	Work: This is the default command line parameter and encrypts all files on the host. It also wipes all free space by creating the following file c:\wipe and filling it with random data the size of the free space available on the drive. The file is then deleted. This parameter can also be used to encrypt a single file i.e. -w filename.exe
-r	Dry Run: Does not encrypt any files on the host.
-l	Log: Creates a log file off the root drive named cl.log.
-f	File: Used to encrypt a single file.

Table 1. Supported command line arguments

Encryption

LockerGoga uses the Cryptopp library to implement RSA, as implementing RSA from scratch would be very time consuming and error prone. To encrypt files, (Strong RSA) RSA-OAEP MGF1(SHA-1) is used. The RSA public key found in this sample is:

00000000 4D 49 47 64 4D 41 30 47 43 53 71 47 53 49 62 33 MIGdMA0GCSqGSIb3

00000016 44 51 45 42 41 51 55 41 41 34 47 4C 41 44 43 42 DQEBAQUAA4GLADCB

00000032 68 77 4B 42 67 51 43 46 66 33 43 54 59 79 41 79 hwKBgQCFf3CTYyAy

00000048 6F 79 5A 71 52 33 6E 48 63 4C 4A 2B 49 2F 71 69 oyZqR3nHcLJ+I/qi

00000064 2F 50 57 77 57 54 75 6C 20 6C 4D 69 4E 32 54 47 /PWwWTul lMiN2TG

00000080 4D 41 4D 62 34 39 75 58 51 32 79 43 34 4D 5A 76 MAMb49uXQ2yC4MZv

00000096 5A 76 4B 53 50 55 44 6F 33 61 4D 67 5A 4A 71 30 ZvKSPUDo3aMgZJq0

00000112 78 75 52 53 42 34 58 6F 73 6D 73 30 5A 39 51 4B xuRSB4Xosms0Z9QK

00000128 70 76 47 6C 6A 4E 48 36 79 34 50 59 4E 39 38 2F pvGljNH6y4PYN98/

00000144 76 20 79 31 7A 4F 6B 34 70 45 69 53 68 43 32 49 v y1zOk4pEiShC2I

00000160 47 46 50 4A 32 47 71 33 4F 63 2B 41 78 4F 37 57 GFPJ2Gq3Oc+AxO7W

00000176 6F 2F 62 42 76 35 34 32 52 51 30 67 50 55 77 7A o/bBv542RQ0gPUwz

00000192 79 54 53 66 71 6A 44 47 35 33 35 73 38 57 73 76 yTSfqjDG535s8Wsv

00000208 4B 73 79 77 49 42 45 51 3D 3D KsywIBEQ==

Table 2. RSA Public key found in initial LockerGoga sample

Although RSA-OAEP MGF1 has features that make it more secure, the added computational overhead causes encryption to take longer and has difficulty handling large files. To mitigate this, the developers launch multiple child processes that work in parallel to maximize encryption speed. To overcome large files, the developers decided to encrypt chunks of a file every 80,000 bytes and skip the next 80,000 bytes of a file. Example:

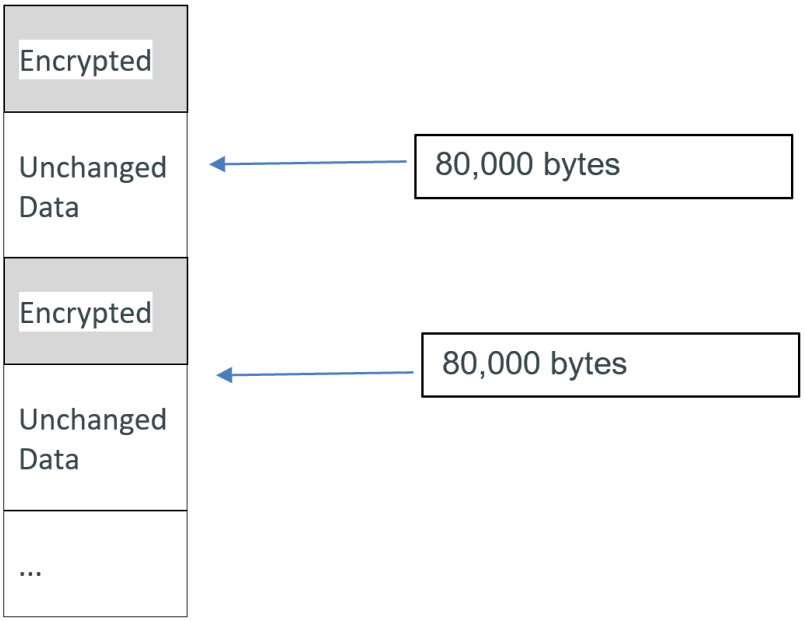

Figure 1. Data encrypted in 80,000 byte chunks

Because of this, partial recovery of large files might be possible even without the decryption key.

Although the developers attempt to use a denylist of files and directories to skip, it was observed encrypting core Windows operating system files, which caused the operating system to become unstable and crash. This was observed when running the ransomware on a Windows 2012 machine.

Early Development

Based on our analysis, we believe this variant is an early release of LockerGoga ransomware. The developers left behind command line parameters such as -r which allows the malware to run without encrypting anything. This can be used in conjunction with -l (log) to test how the ransomware behaves. Both of these parameters are suggestive of an initial, or test, build. The -r specifically was not observed in later variants.

According to the Bleeping Computer report, the ransomware appeared to encrypt only the following specific file extensions: doc|dot|wbk|docx|dotx|docb|xlm|xlsx|xltx|xlsb|xlw|ppt|pot|pps|pptx|potx|ppsx|sldx|pdf.

Our analysis found that the malware does not use the code block that checks for specific file extensions. Instead, we observed that the malware encrypts all files except those in the denylist. It also has issues with large files, which is addressed in later variants.

Development Cycle

Like other active software projects, the LockerGoga ransomware is under constant development with new variants being developed and used to attack victims. All these variants share similar characteristics and just like other professional development, each release contains improvements or new capabilities.

To counter this ongoing development cycle, security researchers have to focus on identifying new variants.

To that end, we highlight some of the evolutionary changes we have observed since the malware surfaced in January:

- Dropped the -r and -f command line arguments
- Log file renamed from cl.log to .log
- Ransomware note is no longer encoded in the binary
- Ransomware note renamed from README_NOW.txt to README_LOCKED.TXT
- Added the following command line parameters:

Parameter	Description
-i	Inter process communication (IPC)
-s	Slave (child process)
-m	Master Process

- Began using mutexes (example: SM-zzbdrimp) to identify processes for inter-process communication.
- Stopped using svch0st[numeric value].exe as a process name and began using distinct hard-coded names for each sample.

- Uses undocumented Windows API calls (for example NtQuerySection)
- Changes administrator password to HuHuHUHoHo283283@dJD
- Stopped creating a wipe file to erase free space and instead uses Windows cipher.exe with command parameter /w to wipe free space on the host
- Added importation of WS2_32.dll, possibly to support network communications
- Changed the format and collection of data recorded in the log file
- Updated encryption of files and directories. No longer encrypts all files on the host: instead targets specific file extensions and directories
- Calls into Windows Restart Manager session for possible token elevation to overwrite trusted installer files
- Uses the following digital signers:
    - ALISA LTD
    - KITTY'S LTD
    - MIKL LIMITED
- Logs off the current user
- The inclusion of the word “Goga” in the binary and encrypted files

Conclusion

LockerGoga’s developers continue to add capabilities and launch new attacks. The addition of WS2_32.dll and use of undocumented Windows API calls indicates a level of sophistication beyond typical ransomware authors. The former could lead to the eventual inclusion of C2 communication or automated propagation, and the latter requires some working knowledge of Windows internals.

These features raise more questions about the actor’s intent as ransomware is typically one of the least advanced forms of malware: Are they motivated by profits or something else? Has the motive change over time? Why would developers put such effort into their work only to partially encrypt files. Why do they include an email address and not seek payment through more frequently used cryptocurrencies?

We do not know if any of the victims have paid the ransom and were able to successfully retrieve their data. We do know that this ransomware has caused significant harm. The damage could increase significantly if the attackers continue to refine this ransomware. Unit 42 will continue to monitor LockerGoga and report on new activity.

WildFire properly identifies all of the malware samples listed in this report as malicious. Traps prevents execution of LockerGoga samples and AutoFocus customers can view LockerGoga samples using the LockerGoga tag.

Palo Alto Networks has shared our findings, including file samples and indicators of compromise, in this report with our fellow Cyber Threat Alliance members. CTA members use this intelligence to rapidly deploy protections to their customers and to systematically disrupt malicious cyber actors. For more information on the Cyber Threat Alliance, visit www.cyberthreatalliance.org.

Indicators of Compromise

LockerGoga Samples

ae7e9839b7fb750128147a9227d3733dde2faacd13c478e8f4d8d6c6c2fc1a55

f474a8c0f66dee3d504fff1e49342ee70dd6f402c3fa0687b15ea9d0dd15613a

ffab69deafa647e2b54d8daf8c740b559a7982c3c7c1506ac6efc8de30c37fd5

c1670e190409619b5a541706976e5a649bef75c75b4b82caf00e9d85afc91881

65d5dd067e5550867b532f4e52af47b320bd31bc906d7bf5db889d0ff3f73041

31fdce53ee34dbc8e7a9f57b30a0fbb416ab1b3e0c145edd28b65bd6794047c1

32d959169ab8ad7e9d4bd046cdb585036c71380d9c45e7bb9513935cd1e225b5

e00a36f4295bb3ba17d36d75ee27f7d2c20646b6e0352e6d765b7ac738ebe5ee

6d8f1a20dc0b67eb1c3393c6c7fc859f99a12abbca9c45dcbc0efd4dc712fb7c

79c11575f0495a3daaf93392bc8134c652360c5561e6f32d002209bc41471a07

050b4028b76cd907aabce3d07ebd9f38e56c48c991378d1c65442f9f5628aa9e

1f9b5fa30fd8835815270f7951f624698529332931725c1e17c41fd3dd040afe

276104ba67006897630a7bdaa22343944983d9397a538504935f2ec7ac10b534

88d149f3e47dc337695d76da52b25660e3a454768af0d7e59c913995af496a0f

c97d9bbc80b573bdeeda3812f4d00e5183493dd0d5805e2508728f65977dda15

06e3924a863f12f57e903ae565052271740c4096bd4b47c38a9604951383bcd1

a845c34b0f675827444d6c502c0c461ed4445a00d83b31d5769646b88d7bbedf

7bcd69b3085126f7e97406889f78ab74e87230c11812b79406d723a80c08dd26

ba15c27f26265f4b063b65654e9d7c248d0d651919fafb68cb4765d1e057f93f

eda26a1cd80aac1c42cdbba9af813d9c4bc81f6052080bc33435d1e076e75aa0

7852b47e7a9e3f792755395584c64dd81b68ab3cbcdf82f60e50dc5fa7385125

14e8a8095426245633cd6c3440afc5b29d0c8cd4acefd10e16f82eb3295077ca

47f5a231f7cd0e36508ca6ff8c21c08a7248f0f2bd79c1e772b73443597b09b4

f3c58f6de17d2ef3e894c09bc68c0afcce23254916c182e44056db3cad710192

9128e1c56463b3ce7d4578ef14ccdfdba15ccc2d73545cb541ea3e80344b173c

c3d334cb7f6007c9ebee1a68c4f3f72eac9b3c102461d39f2a0a4b32a053843a

6e69548b1ae61d951452b65db15716a5ee2f9373be05011e897c61118c239a77

8cfbd38855d2d6033847142fdfa74710b796daf465ab94216fbbbe85971aee29

bdf36127817413f625d2625d3133760af724d6ad2410bea7297ddc116abc268f

5b0b972713cd8611b04e4673676cdff70345ac7301b2c23173cdfeaff564225c

c7a69dcfb6a3fe433a52a71d85a7e90df25b1db1bc843a541eb08ea2fd1052a4

Appendix

The latest variant of LockerGoga uses memory mapped files to communicate between processes. To illustrate this, we captured the memory of a section created by a child process.

00000000 02 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00

00000016 D0 02 01 00 CC 02 01 00 C8 02 01 00 30 00 00 00 Ð Ì È 0

00000032 80 02 01 00 F8 FF 0F 00 00 00 00 00 00 00 00 00 € øÿ

00000048 FF FF FF FF 00 00 00 00 01 00 00 00 24 00 00 00 ÿÿÿÿ $

00000064 20 00 00 00 1C 00 00 00 00 00 00 00 01 00 00 00

00000080 FC FF FF FF F8 FF FF FF F4 FF 01 00 0B 20 00 C0 üÿÿÿøÿÿÿôÿ À

00000096 DE FF FF FF 01 00 00 00 01 00 00 00 38 00 01 00 Þÿÿÿ 8

00000112 01 00 04 21 10 01 00 00 00 00 00 00 00 7A 70 63 ! zpc

00000128 55 48 4A 76 5A 33 4A 68 62 53 42 47 61 57 78 6C UHJvZ3JhbSBGaWxl

00000144 63 31 78 57 54 58 64 68 63 6D 56 63 56 6B 31 33 c1xWTXdhcmVcVk13

00000160 59 58 4A 6C 49 46 52 76 62 32 78 7A 58 47 64 73 YXJlIFRvb2xzXGds

00000176 61 57 49 74 4D 69 34 77 4C 6D 52 73 62 41 3D 3D aWItMi4wLmRsbA==

00000192 64 48 42 6A 63 48 4D 75 5A 47 78 73 63 6D 56 7A dHBjcHMuZGxscmVz

00000208 65 43 35 6B 62 47 77 75 62 58 56 70 4E 46 39 66 eC5kbGwubXVpNF9f

00000224 4F 48 64 6C 61 33 6C 69 4D 32 51 34 59 6D 4A 33 OHdla3liM2Q4YmJ3

00000240 5A 54 68 68 5A 44 46 68 4E 7A 51 77 4C 54 52 68 ZThhZDFhNzQwLTRh

00000256 4E 32 59 74 4E 47 45 78 59 53 30 35 4F 54 45 78 N2YtNGExYS05OTEx

00000272 4C 54 51 79 4D 57 51 30 4D 6A 49 34 4D 7A 52 6C LTQyMWQ0MjI4MzRl

00000288 5A 46 78 42 63 33 4E 6C 64 48 4E 63 55 6D 56 7A ZFxBc3NldHNcUmVz

00000304 62 33 56 79 59 32 56 7A 58 46 4A 6C 63 58 56 70 b3VyY2VzXFJlcXVp

00000320 63 6D 56 6B 55 48 4A 70 62 6E 52 44 59 58 42 68 cmVkUHJpbnRDYXBh

00000336 59 6D 6C 73 61 58 52 70 5A 58 4D 75 65 47 31 73 YmlsaXRpZXMueG1s

00000352 65 6D 55 74 4E 44 68 66 59 57 78 30 5A 6D 39 79 emUtNDhfYWx0Zm9y

00000368 62 53 31 31 62 6E 42 73 59 58 52 6C 5A 43 35 77 bS11bnBsYXRlZC5w

00000384 62 6D 63 3D 62 53 31 31 62 6E 42 73 59 58 52 6C bmc=bS11bnBsYXRl

00000400 5A 43 35 77 62 6D 63 3D 00 00 00 00 00 00 00 00 ZC5wbmc=

Table 3. Child process mapped file

The 1st byte (02) instructs the master process to process the data below. The data is base64 encoded in three parts and decodes to the following:

Part 1:

Unknown value

Part 2:

tpcps.dllresx.dll.mui4__8wekyb3d8bbwe8ad1a740-4a7f-4a1a-9911-421d422834ed\Assets\Resources\RequiredPrintCapabilities.xmlze-48_altform-unplated.png

Part 3:

m-unplated.png

Example of initial variant of LockerGoga running on a host:

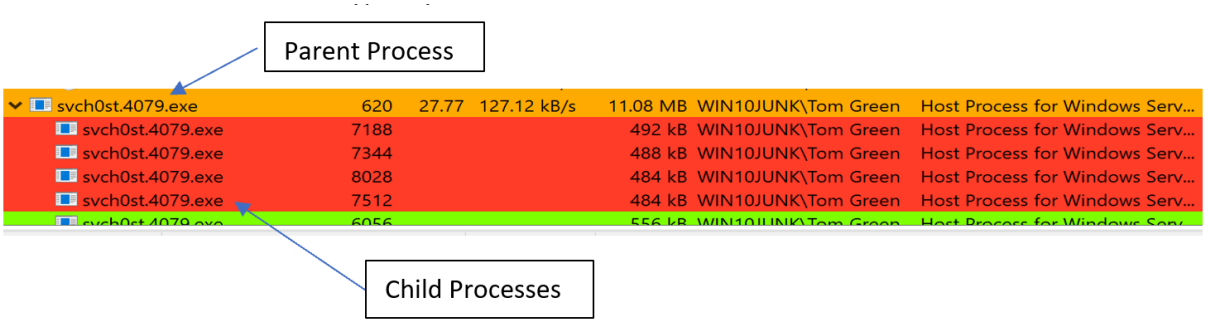

Figure 2. Initial LockerGoga running

Initial LockerGoga Ransom Note:

Greetings!

There was a significant flaw in the security system of your company.

You should be thankful that the flaw was exploited by serious people and not some rookies.

They would have damaged all of your data by mistake or for fun.

Your files are encrypted with the strongest military algorithms RSA4096 and AES-256.

Without our special decoder it is impossible to restore the data.

Attempts to restore your data with third party software as Photorec, RannohDecryptor etc.

will lead to irreversible destruction of your data.

To confirm our honest intentions.

Send us 2-3 different random files and you will get them decrypted.

It can be from different computers on your network to be sure that our decoder decrypts everything.

Sample files we unlock for free (files should not be related to any kind of backups).

We exclusively have decryption software for your situation

DO NOT RESET OR SHUTDOWN - files may be damaged.

DO NOT RENAME the encrypted files.

DO NOT MOVE the encrypted files.

This may lead to the impossibility of recovery of the certain files.

To get information on the price of the decoder contact us at:

CottleAkela@protonmail.com;QyavauZehyco1994@o2.pl

The payment has to be made in Bitcoins.

The final price depends on how fast you contact us.

As soon as we receive the payment you will get the decryption tool and

instructions on how to improve your systems security

Figure 3. Initial LockerGoga ransom note

Latest LockerGoga Ransom Note:

Greetings!

There was a significant flaw in the security system of your company.

You should be thankful that the flaw was exploited by serious people and not some rookies.

They would have damaged all of your data by mistake or for fun.

Your files are encrypted with the strongest military algorithms RSA4096 and AES-256.

Without our special decoder it is impossible to restore the data.

Attempts to restore your data with third party software as Photorec, RannohDecryptor etc.

will lead to irreversible destruction of your data.

To confirm our honest intentions.

Send us 2-3 different random files and you will get them decrypted.

It can be from different computers on your network to be sure that our decoder decrypts everything.

Sample files we unlock for free (files should not be related to any kind of backups).

We exclusively have decryption software for your situation

DO NOT RESET OR SHUTDOWN - files may be damaged.

DO NOT RENAME the encrypted files.

DO NOT MOVE the encrypted files.

This may lead to the impossibility of recovery of the certain files.

The payment has to be made in Bitcoins.

The final price depends on how fast you contact us.

As soon as we receive the payment you will get the decryption tool and

instructions on how to improve your systems security

To get information on the price of the decoder contact us at:

MayarChenot@protonmail.com

QicifomuEjijika@o2.pl

Figure 4. Latest LockerGoga ransom note
