TECH BRIEF





LockBit Attempts to Stay Afloat with a New Version

Technical Appendix: LockBit-NG-Dev Detailed Analysis

The new LockBit version we analyzed was packed using the MPRESS packer, possibly to evade static file detections. After unpacking, we can see that this new version seems to have been written in .NET and possibly compiled using CoreRT, which is different from the usual C/C++ language used for past versions.



Figure 1. Shows the use of MPRESS packer using the Detect-It-Easy tool

Like past versions, it still has an embedded configuration that dictates the routines it can perform. The configuration, which is in JSON format, is decrypted at runtime and includes information like date range for execution, the ransom note filename and content, unique IDs for the ransomware, the RSA public key, and some other flags and lists for its other routines. A table with the full configuration options is included at the end of this brief.

Address	UNICODE
0000000143007F4C	MinDate": "2022-06-01", "MaxDate": "2023-09-16", "Ap
0000000143007FCC	pendedExtension": "locked_for_lockbit", "NoteFilename": "READ
000000014300804C	MEtxt", "NoteContent": null
00000001430080CC	, "ID": "', "ChangeFilenam
000000014300814C	e": true, "EncryptNetworkShares": true, "SkipHiddenFiles":
00000001430081CC	<pre>true, "DeleteVolumeShadowCopies": true, "DeleteWindowsSys</pre>
000000014300824C	<pre>temBackups": true, "EfficiencyMode": true, "SelfDelete": f</pre>
00000001430082CC	alse, "DropNoteBeforeEncryption": true, "DropNoteInEveryDi
000000014300834C	rectory": true, "DropNoteInSpecificDirectories": false, "D
00000001430083CC	irectoriesToDropNoteIn": [], "RegexDropNoteInSpecificDirector
000000014300844C	ies": false, "DirectoriesToDropNoteInRegexQueryString": "",
00000001430084CC	"StopProcesses": false, "ProcessesToStop": [], "StopServ
000000014300854C	<pre>ices": false, "ServicesToStop": [], "IncludeFiles": false,</pre>
00000001430085CC	"FileSet": ["READMEt
000000014300864C	<pre>xt"], "RegexIncludeFiles": false, "FilesRegexQueryStri</pre>
00000001430086CC	ng": "", "IncludeDirectories": false, "DirectoryList": [
000000014300874C	"C:\\Windows", "C:\\ProgramData", "C:\\Program Fil
00000001430087CC	es (x86)", "C:\\Program Files", "C:\\msys64"],
000000014300884C	"RegexIncludeDirectories": false, "DirectoriesRegexQueryStrin
00000001430088CC	g": "", "IncludeExtensions": false, "NoneSet": [".in
000000014300894C	1" ".dll", ".tmp", ".exe", ".url", ".l
00000001430089CC	nk", ".locked_for_lockbit"], "FastSet": [], "Inte
0000000143008A4C	rmittentSet": [".sql", ".csv"], "FullSet": [
0000000143008ACC	".txt"], "Buttersize": 4096, "Percent": 0.025,
00000014300884C	"Segmentation": 256, "Publickey": "BEGIN RSA PUBLIC KEY-

Figure 2. Decrypted configuration in JSON format

After decrypting the configuration, LockBit will then create a mutex using the value of *ID* field from the configuration as the mutex name. If the mutex already exists, the process will exit to avoid multiple instances of execution.

Figure 3. Mutex checking routine

One of the new behaviors of LockBit is its ability to check if the current date is within the date range set in the configuration. If the date is not within this range, the process will terminate. The analyzed sample only works between a specified start and end date. This is probably LockBit's way to limit affiliates from reusing their ransomware, forcing them to purchase a new version from the operators once the date expires. This can also be considered an anti-analysis and anti-sandbox technique — however, it is relatively simple for an analyst to bypass this during reverse engineering. On the other hand, it could would be more difficult for an affiliate to patch the binary before using it against a victim.

```
if ( SDWORD2(Date) < MinDate || SDWORD2(Date) > MaxDate )
{
    if ( qword_1402771A8[-1] )
        sub_140001105();
    LODWORD(qword_1402771A8[0]) = 0;
    sub_1400E0130();
    v7 = qword_1402771A8[0];
    sub_140006400(&v6);
    ExitProcess(v7);
}
```

Figure 4. Checking if the current date is within the valid date range

Similar to other ransomware, it terminates processes and stops services that may be accessing files it is attempting to encrypt or security-related processes and services that may hinder the execution of the ransomware to ensure the proper encryption of files,. To do so, it first needs to check if the *StopProcesses* or *StopServices* flags are true in the configuration. If true, it will terminate processes from the list of process names under the *ProcessesToStop* field and services from the list of services in the *ServicesToStop* field in the configuration.

```
if ( config->StopProcesses )
{
    sub_1401E44E0(v196, &qword_1402AAC48, *(config->ProcessesToStop + 8), 0i64, -1, -1, 0)
if ( v10->StopServices )
{
    sub_1401E44E0(v194, &qword_1402AAC48, *(v10->ServicesToStop + 8), 0i64, -1, -1, 0);
    Figure 5. Routine checking if processes and services will be stopped
```

LockBit also inhibits recovery from shadow copies and backups by performing the following routines before encryption:

It checks if *DeleteVolumeShadowCopies* is true in the configuration, and if it is, deletes shadow copies by executing the following command:

"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe vssadmin Delete Shadows /All /Quite"

To delete the Windows backups, it checks if *DeleteWindowsSystemBackups* is true in the configuration, and if it is, it executes the following command:

"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe wbadmin DELETE BACKUP -keepVersions:0 -quiet"

```
if ( config_14042AC88[3]->DeleteVolumeShadowCopies )
    execute_via_powershell_140075100(&str_vssadmin_140271AA8, 0x2710u);// vssadmin Delete Shadows /All /Quiet
if ( *(&qword_140276FD8 - 1) )
    sub_140001440();
if ( config_14042AC88[3]->DeleteWindowsSystemBackups )
    execute_via_powershell_140075100(&str_wbadmin_140271BC8, 0x2710u);// wbadmin DELETE BACKUP -keepVersions:0 -quiet
```

```
Figure 6. Routine executing PowerShell commands to delete shadow copies and backups
```

One of the routines it possesses that was also in past versions is the ability to rename the encrypted files with random filenames. It does this by checking if the *ChangeFilename* field is true in the configuration, then it generates a random filename using a randomizer function.

The original file name will then be placed within the content of the file after the encrypted blob. For files that are not encrypted with full encrypt mode, it will just be appended on the file. Meanwhile, those encrypted via full encrypt mode will have filenames that will be included in the RSA encrypted buffer.

X429prj	52e	7474	466	008.10	cked	fo	or_lock	kbit	ŧ .					↓FRO
00001B94:	4E	00	6F	00-74	00	6F	00-53	00	61	00-6E	00	73	00	NotoSans
00001BA4:	42	00	75	00-67	00	69	00-6E	00	65	<u>00-</u> 73	00	65	00	Buginese
00001BB4:	2D	60	52	00-65	00	67	00-75	00	6C	00-61	00	72	00	- Regular
00001BC4:	2E	60	74	00- 74	00	66	00-38	00	00	00-82	7C	DE	6F	.ttf8 é o
00001BD4:	F8	84	FB	72-01	DF	41	64-2B	BC	65	D7-E8	EC	84	46	°ä√r©Ad+≝e-Φ∞äF
00001BE4:	44	7C	FA	0C-79	43	ØD	A7-4A	B7	DØ	47-46	E6	FØ	CD	D ·♀yC♪♀J╢ [⊥] GFμ≡=
00001BF4:	E1	65	E6	B1-6E	36	81	30-AD	F6	AD	F1-00	C4	68	BB	Beµ≣n6ü0;÷;± −h╗
00001C04:	A5	DØ	90	5E-62	4C	BB	9F-0C	6A	23	0B-6C	43	C1	ØE	Ň [⊥] É^bL╗f♀j#ðlC⊥⊅
00001C14:	03	88	97	21-E6	24	AF	E4-B9	74	52	04-E3	02	BØ	7C	♥êù!µ\$»Σ╣tR♦π⊕
00001C24:	48	F7	98	1C-9D	26	22	21-69	5D	64	62-D1	43	DB	4E	H≈ÿ∟¥&"!i]db = C <mark>N</mark>
00001C34:	15	ED	04	3A-EA	B 9	1E	77-74	41	ØE	DA-2E	F5	2F	8D	§φ♦:Ω=▲wtA⊅ _Γ ./ì
00001C44:	D0	95	96	2B-BD	BA	23	68-09	BØ	59	93-09	2B	EE	66	[⊥] òû+ [⊥] #ho Yôo+εf
00001C54:	2F	AE	99	B3-FF	46	2F	44-4D	72	02	10-4E	86	ЗD	6E	/«Ö F/DMr⊕►Nå=n
00001064:	00	36	00	41-39	C5	A6	F8-77	38	BA	FF-B7	4F	05	1 B	262A9∔ª°₩8∥ ■ ₩N ‡ ←
Figure 7.	File	er	ıcry	/pted	wit	h li	nterm	itte	ent	mode	e ha	as t	he	original filename in its conte



Figure 8. File encrypted with full encrypted mode has the original filename in the RSA encrypted buffer

LockBit has three encryption modes: fast, intermittent and full. Files are usually encrypted under fast mode to speed up encryption (an option commonly favored by affiliates), but it can be configured to perform different modes based on file extensions.

The sample we analyzed has set ".txt" file extensions to full encryption mode, while ".csv" and ".sql" are encrypted with intermittent mode. The three modes to encrypt files are as follows:

- **Fast** encrypts the first 0x1000 bytes of the file (files listed in Fast Set will use *Buffersize* value to determine the size to encrypt).
- **Intermittent** only encrypts a certain percentage of the file based on the value set in the configuration under the *Percent* field. Also, the field *Segmentation* determines the distance between encrypted blocks.
- **Full** encrypts the whole file.

Like other ransomware, LockBit-NG-Dev avoids encrypting certain directories, files and file extensions. These files are listed in the configuration under the *DirectoryList*, *FileSet*, and *NoneSet* fields. Also, the configuration fields *IncludeFiles*, *IncludeDirectories*, and *IncludeExtensions* need to be set to false. It also has a regex option for the files and directories to avoid under the fields *FilesRegexQueryString* and *DirectoriesRegexQueryString*.

For network encryption, if *EnableNetworkShares* is true, it also encrypts files on available network shares.

LockBit-NG-Dev encrypts files using the AES algorithm and encrypts the AES key using the embedded RSA public key that can also be found in the configuration. The AES keys are randomly generated for each file to be encrypted.

The ransom note content and file name are also in the configuration. It can also be set in the configuration if the ransom note will be dropped on all directories or only in specific directories by exact path/s that match a regular expression.

An option exists wherein the ransom note would first be dropped on target directories (or all traversed directories) before encryption begins. If this is enabled, dropping the same text file on multiple directories could be flagged by behavior monitoring tools as a suspicious routine and may terminate the execution process before the actual encryption begins.

```
if ( v28->DropNoteBeforeEncryption )
{
    v29 = sub_140077C40();
    sub_140004BF0(&v28->NoteContent, v29);
    v30 = v12[3];
    if ( v30->DropNoteInSpecificDirectories )
    {
        v31 = v30->DirectoriesToDropNoteIn;
        v32 = *(v24 + 32);
        if ( !v32 )
        {
            v32 = s_1400049E0(qword_1402ACDD0);
            sub_1400027AB(v32, *(v24 + 8));
            sub_140004BF0((v24 + 32), v32);
        }
```

Figure 9. Checking if the ransom note will be dropped before encryption

Finally, if *Self-delete* is true in the configuration, it will remove traces of the ransomware by zeroing out its contents using the following command:

powershell.exe -Stop-Process -Id {process id} -Force; fsutil.exe file setZeroData offset=0 length=9999999999 "{Path of ransomware}"

00000001400	72302		5D							p	op i	rbp														6 I	- 6-1		h 113		
00000001400	72303		5 E							p	op 1	rsi											ns:		De	Tault	t (xe	94 Tas	tcail)		
00000001400	72304		5 F							p	n qq	rdi													1:	n	cx /	0000	0001	4556	570E
0000001400	72305		C3	_						ne	t													~	2:	n	dx (0000	0000	0000	0138
<																								>	3:	r	8 O	0000	0000	01D2	22B0
red-000000	000103	200			_																				- 4	_ <u>r</u>	9 0	0000	0000	2695	9090
ecx=455670E	8	260																							1.2	- 17	sp	+20]	000	0000	0000
CCX-1550/02	·																														
.MPRESS1:00	000001	400	722/	49 e	enc.	. ex	e:\$	722	A9 i	#71	4A9	•													<						
Dump 1	m D	imp 2			Dum	n 3			Dum	.4			Dum	5	6	ti) v	Vato	h 1	La	=11 o	cale	4	ð 🕫	ruct							
e-e bump 1	8-8 DI	ump 2	•	0-0	Dun	pэ		0-0	Duni	7		0-0	baing	55			valu	11	14	-160	cais	6	2 31	ucu		_	_				
Address		нех																ASC	II												
00000001455	670E8	A8	7D 🛛	28 4	10	01	00	00	00	67	00	00	00	53	00	74	00	11	e	g	5	5.t.									
0000001455	670F8	6F	00	70 C	00	2D	00	50	00	72	00	6F	00	63	00	65	00	o.p	·	P.r	.0.0	e.									
00000001455	67108	73	00	73 C	00	20	00	2D	00	49	00	64	00	20	00	33	00	5.5		I	.d.	.3.									
00000001455	67118	30	00	38 C	00	38	00	20	00	2D	00	46	00	6F	00	72	00	0.8	. 8.		.F.0	o.r.									
00000001455	67128	63	00	65 C	00	3B	00	20	00	66	00	73	00	75	00	74	00	c.e	. : .	. f	.s.u	i.t.									
00000001455	67138	69	00	6C (ool:	2E	00	65	00	78	00	65	00	20	00	66	00	i.1		e.x	.e.	.f.									
00000001455	67148	69	00	6C (65	00	20	00	73	00	65	00	74	00	5A	00	i.1	.e.	. 5	.e.1	.z.									
00000001455	67158	65	00	72 0	00	6F	00	44	00	61	00	74	00	61	00	20	00	e.r	. 0.	D.a	.t.a	ı									
00000001455	67168	6F	00	66 0	ool	66	00	73	00	65	00	74	00	3D	00	30	00	o.f	f.	s.e	.t.=	. 0.									
00000001455	67178	20	00	6C (00	65	00	6E	00	67	00	74	00	68	00	3D	00	1	.e.	n.a	.t.)	1. =.									
00000001455	67188	39	00	39 Ö		39	00	39	00	39	00	39	00	39	00	39	00	9.9	. 9.	9.9	. 9. 9	9.9.									
00000001455	67198	39	00	39 0		20	00	22	00	43	00	3A	00	5C	00	SE.	00	9.9		".c											

Figure 11. Executing commands via PowerShell

Full configuration

Field	Description
MinDate	Minimum Date where ransomware will execute (format. MM/DD/YYYY)
MaxDate	Maximum Date where ransomware will execute (format. MM/DD/YYYY)
AppendedExtension	Extension that will be appended on encrypted files (ex. locked_for_LockBit)
NoteFilename	Filename of the ransom note
ID	Unique identifier for the ransomware
ChangeFilename	If true, change filename of encrypted files to a random one.
EncryptNetworkShares	if true, include network drives in encryption
SkipHiddenFiles	if true, will not encrypt files with attribute hidden
DeleteVolumeShadowCopies	If true, delete shadow copies by executing vssadmin command
DeleteWindowsSystemBackups	If true, delete windows backup by executing wbadmin command
EfficiencyMode	Flag to check if will use enough or more resources during encryption
SelfDelete	If true, overwrite contents of ransomware with null bytes
DropNoteBeforeEncryption	If true, ransom note will be dropped first before encrypting
DropNoteInEveryDirectory	if true, drop ransom note on every directory
DropNoteInSpecificDirectories	if true, drop note only on the specified directory in field DirectoriesToDropNoteIn
DirectoriesToDropNoteIn	List of file path to drop ransom note in
RegexDropNoteInSpecificDirectories	if true, drop note on directory if it matches the regex expressions under the DirectoriesToDropNoteInRegexQueryString
DirectoriesToDropNoteInRegexQueryString	List of regex expression of desired directories to drop ransom note
StopProcesses	If true, terminate processes under the <i>ProcessesToStop</i> field.
ProcessesToStop	List of process names to stop
StopServices	If true, stop services under the ServicesToStop field.
ServicesToStop	List of service name to stop
IncludeFiles	If true, include the files under <i>FileSet</i> in encryption routine, otherwise avoid encrypting if set to false
FileSet	List of files that would be excluded in encryption if <i>IncludeFiles</i> is set to false.
RegexIncludeFiles	If true, include the files that matches the regular expression under <i>FilesRegexQueryString</i> in encryption routine, otherwise avoid encrypting if set to false
FilesRegexQueryString	Regular expression of files to include or avoid in encryption

IncludeDirectories	If true, include the directories under <i>DirectoryList</i> in encryption routine, otherwise avoid encrypting if set to false
DirectoryList	List of directories that would be excluded in encryption if <i>IncludeDirectories</i> is set to false
RegexIncludeDirectories	If true, include the directories that matches the regular expression under <i>DirectoriesRegexQueryString</i> in encryption routine, otherwise avoid encrypting if set to false
DirectoriesRegexQueryString	Regular expression of directories to include or avoid in encryption
IncludeExtensions	If true, include the files with extension under <i>NoneSet</i> in encryption routine, otherwise avoid encrypting if set to false
NoneSet	List of file extension to avoid during encryption, if <i>IncludeExtensions</i> is False
FastSet	List of extension of files that are targeted to only be encrypted with the first x bytes where x is value set in <i>BufferSize</i>
IntermitttentSet	List of extension of files that would only be partially encrypted depending on the value of <i>Percent</i>
FullSet	List of extension of files that would be fully encrypted regardless of size
BufferSize	Size to encrypt for files listed in <i>FastSet</i> (ex. 4096)
Percent	Percentage value of intermittent encryption to perform
Segmentation	Value to compute offset of blocks to encrypt under Intermittent Encryption (ex. 256)
PublicKey	RSA public key.

Indicators of Compromise

Detected as Ransom.Win64.LOCKBIT.YXDLS

SHA256

f56cba51a4e86f3be5208dfce598d0d6a86cbbc820b214d5d5df7d327e580b82

TLSH

T1615533707F603835DB3BD27B546D0D8892FB39789A198BFAC0661F87185691F0907A8F



Trend Micro, a global cybersecurity leader, helps make the world safe for exchanging digital information. Fueled by decades of security expertise, global threat research, and continuous innovation, our unified cybersecurity platform protects hundreds of thousands of organizations and millions of individuals across clouds, networks, devices, and endpoints.

With 7,000 employees across 56 countries, and the world's most advanced global threat research and intelligence, Trend Micro enables organizations to simplify and secure their connected world.

TrendMicro.com

©2024 by Trend Micro, Incorporated. All rights reserved. Trend Micro and the Trend Micro t-ball logo are trademarks or registered trademarks of Trend Micro, Incorporated. All other company and/or product names may be trademarks or registered trademarks of their owners.