Lockbit leak, research opportunities on tools leaked from TAs

By Eduardo Ovalle

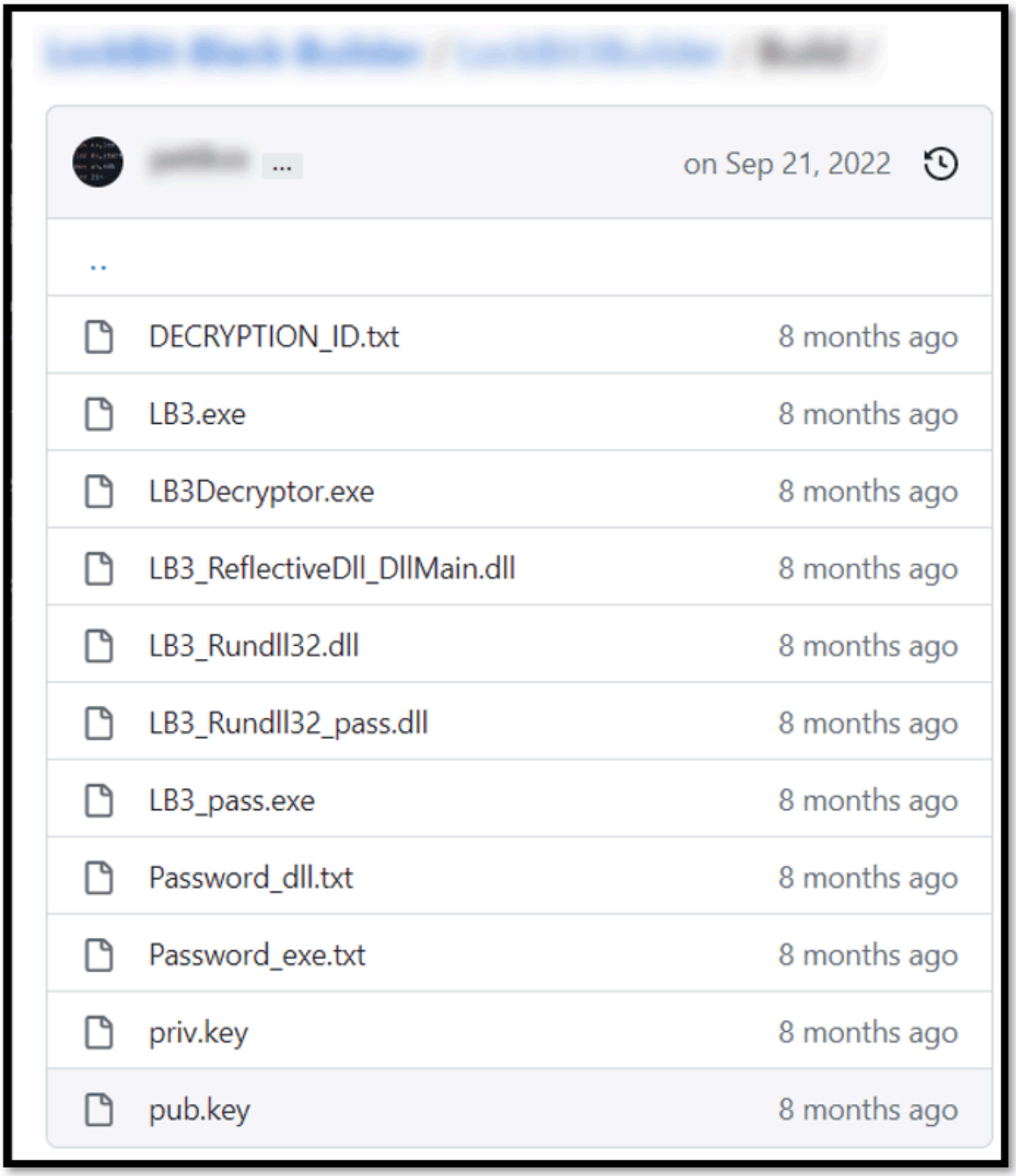
Published: 2023-08-25 · Archived: 2026-04-05 16:51:10 UTC

Lockbit is one of the most prevalent ransomware strains. It comes with an affiliate ransomware-as-a-service (RaaS) program offering up to 80% of the ransom demand to participants, and includes a bug bounty program for those who detect and report vulnerabilities that allow files to be decrypted without paying the ransom. According to the Lockbit owners, the namesake cybercriminal group, there have been bounty payments of up to 50 thousand dollars. In addition to these features, Lockbit has offered a searchable portal to query leaked information from companies targeted by this ransomware family, and even offered payment to those who get tattooed with a Lockbit logo on their body.

Lockbit v3, also known as Lockbit Black, was detected for the first time in June 2022 and represents a challenge for analysts and automated analysis systems. Among the most challenging characteristics, we can highlight the following:

- It supports the usage of encrypted executables with randomly generated passwords. This prevents execution and hinders automatic analysis unless the appropriate password is provided at the command line.
- The payload includes strong protection techniques against reverse-engineering analysis.
- It includes many undocumented kernel-level Windows functions.

In September of 2022, multiple security news professionals wrote about and confirmed the leakage of a builder for Lockbit 3. This tool allowed anyone to create their own customized version of the ransomware. Two different users published the files needed to create different flavors of this ransomware:



Lockbit builder uploaded to GitHub

According to our analysis, two different variants were spotted by the X's (previously known as Twitter) users **@protonleaks** and **@ali_qushji.** Our timestamp analysis confirmed that the binary, builder.exe, was slightly different in both leaks. The version from protonleaks registers the compilation date 2022/09/09. Meanwhile, the version from ali_qushji was compiled on 2022/09/13. A similar difference in compilation time was identified in the malware's template binaries (embedded and incomplete versions of the malware used to build the final version ready for distribution).

ALI_QUSHJI leak builder

Resource	Content	Compile timestamp raw	Compile timestamp
rsrc101	EXE ransomware template	0x632112B1	Tue **Sep 13** 23:30:57 2022 UTC
rsrc103	DLL ransomware template	0x632112B7	Tue Sep 13 23:31:03 2022 UTC
rsrc106	ReflectiveDLL ransomware template	0x632112BE	Tue Sep 13 23:31:10 2022 UTC

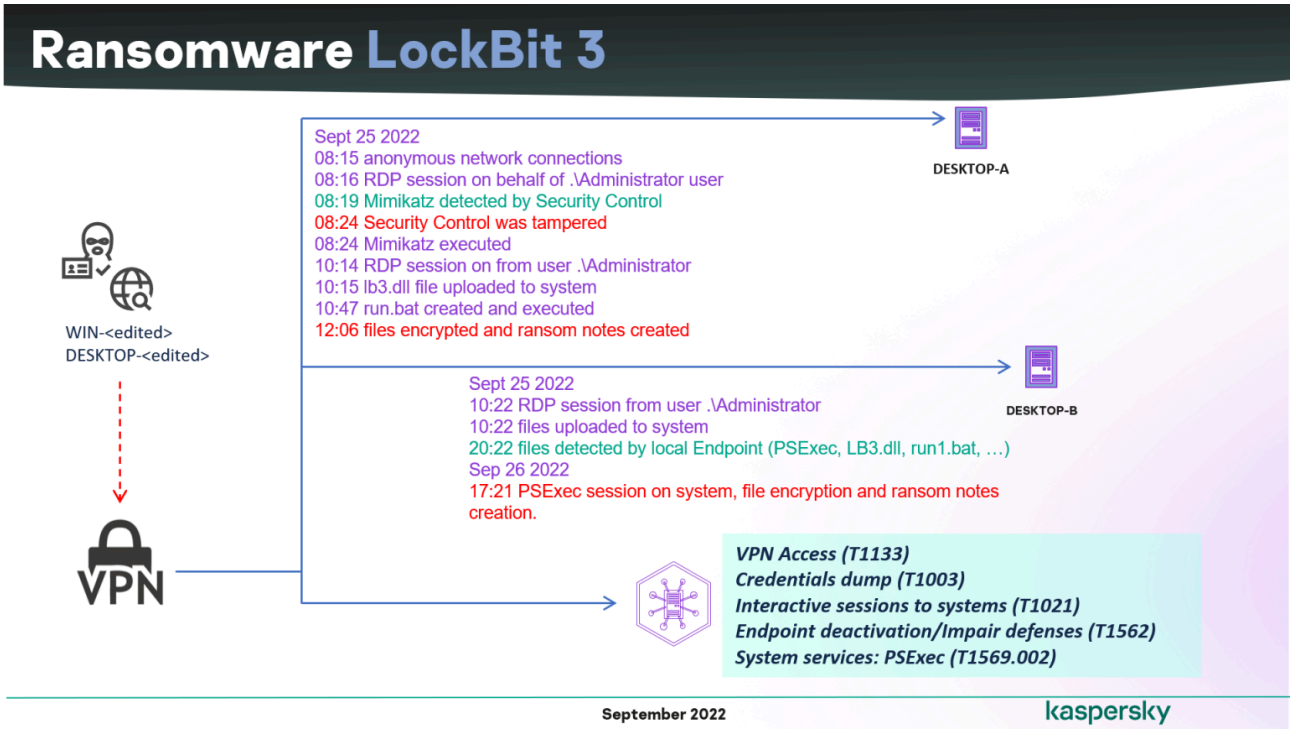
PROTONLEAKS leak builder

Resource	Content	Compile timestamp raw	Compile timestamp
rsrc101	EXE ransomware template	0x631A9665	Fri **Sep 09** 01:27:01 2022 UTC
rsrc103	DLL ransomware template	0x631A966C	Fri Sep 09 01:27:08 2022 UTC
rsrc106	ReflectiveDLL ransomware template	0x631A9643	Fri Sep 09 01:26:27 2022 UTC

Who abused these builders and how?

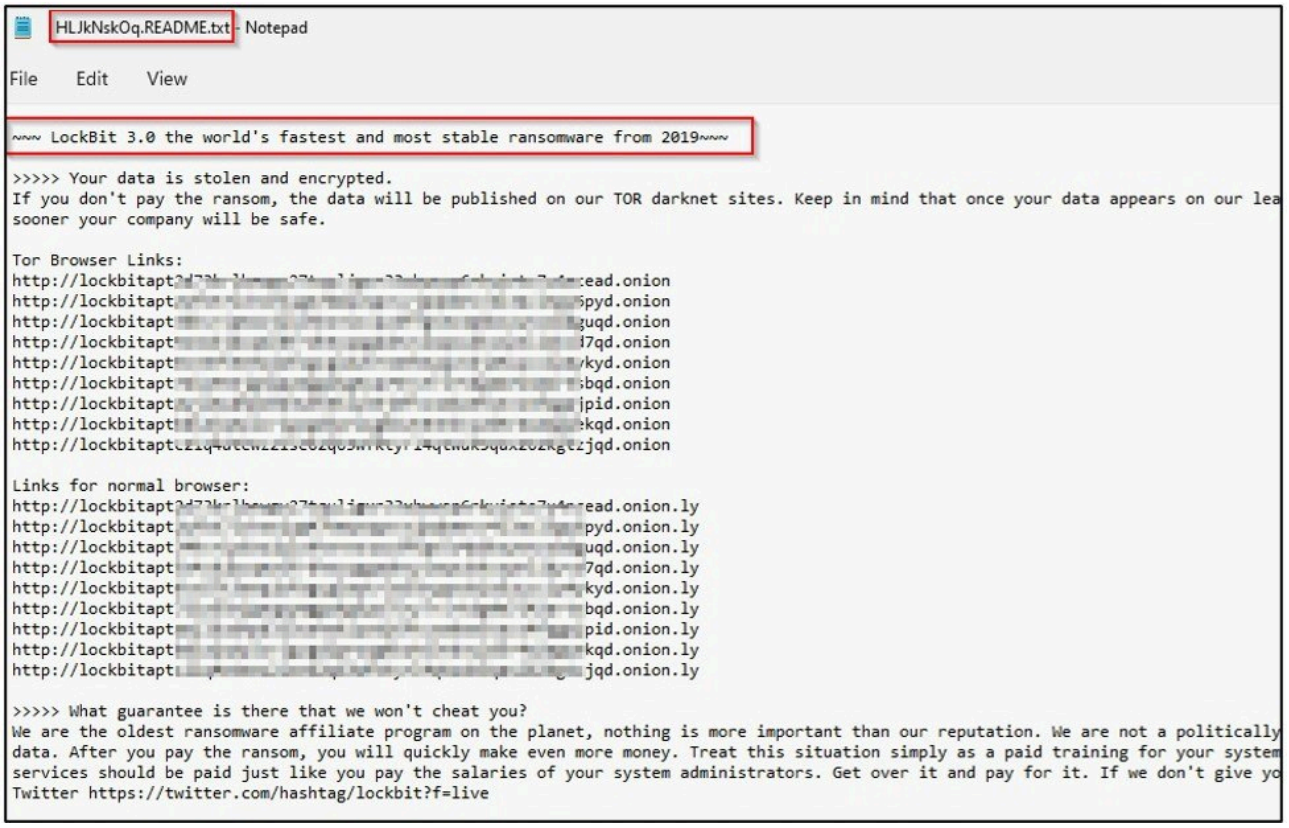
Immediately after the builder leak, during an incident response by our GERT team, we managed to find an intrusion that leveraged the encryption of critical systems with a variant of Lockbit 3 ransomware. Our protection system confirmed and detected the threat as “Trojan.Win32.Inject.aokvy”.

The intrusion included TTPs similar to those highlighted in [the report by Kaspersky Threat Intelligence team] from August 2022 about the eight main ransomware groups behind ransomware attacks, including tactics for reconnaissance, enumeration, collection and deployment.



Although this variant was confirmed as Lockbit, the ransom demand procedure was quite different from the one known to be implemented by this threat actor. The attacker behind this incident decided to **use a different ransom**

note with a headline related to a previously unknown group, called NATIONAL HAZARD AGENCY.



Original Lockbit ransom note

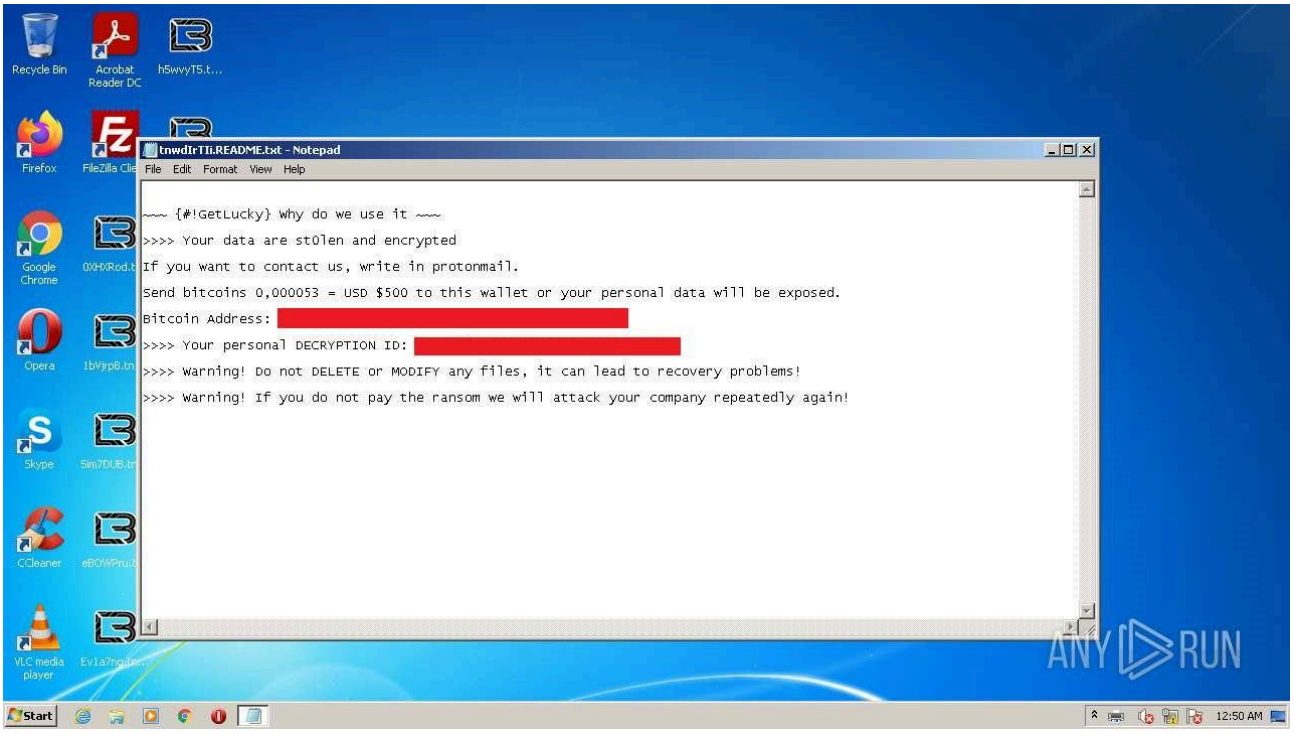
```
            ~~~ NATIONAL HAZARD AGENCY ~~~

>>>> Your data are encrypted

We do not steal data from victims however we can guarantee you that if you refuse to pay or if you do not start
negotiations we may give other hackers access to your company.

The amount of redemption is 3,000,000 USD!
BTC Wallet - [redacted]
MONERO Wallet - [redacted]

We can provide you with a test decryption of any data of your choice to prove that the data can be returned.
Contact us:
        Tox ID Fisheye: [redacted]
        Email: [redacted]

>>>> Your personal DECRYPTION ID: [redacted]

>>>> Warning! Do not DELETE or MODIFY any files, it can lead to recovery problems!

>>>> Warning! If you do not pay the ransom we will attack your company repeatedly again!
```

Managed incident ransom note

The ransom note used in this case directly described the amount to be paid to obtain the keys, and directed communications to a Tox service and email, unlike the Lockbit group, which uses its own communication and

negotiation platform.

According to other [analysts' publications](), different groups appeared using the exfiltrated builders, but with their own notes and communication channels:



GetLucky ransom note, Source: AnyRun

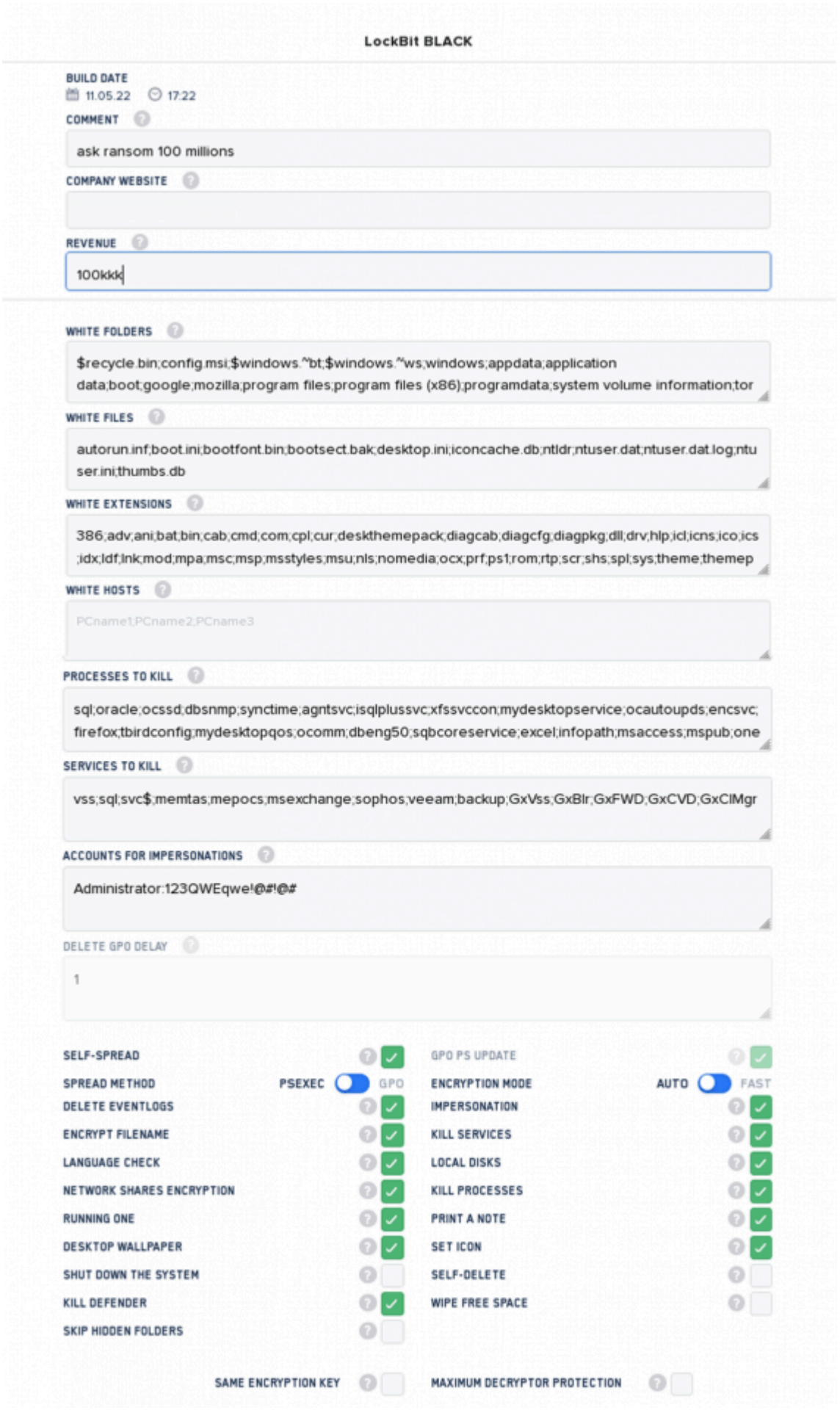
GERT's approach to analyzing the builder and payload

While many threat actors took advantage of the leak to propose new ransomware groups, Kaspersky's GERT team decided to analyze the builder to understand its construction methodology and define additional analysis opportunities.

The analysis of the builder addressed some of the challenges posed by the ransomware payload:

- The **builder contains no protection mechanisms** as it will be used by the actors and should not be exposed: no anti-debugging (at least in the builder itself), no anti-reversing, no code obfuscation, sample templates embedded as resource (decrypter, EXE, DLL, reflective DLL).
- We learned **how the configuration parameters are embedded within the payload** without requiring reverse engineering of the final binary.

The builder presents different configuration parameters that are compulsorily embedded in the malware.


LockBit BLACK
BUILD DATE
11.05.22
17:22
COMMENT
ask ransom 100 millions
COMPANY WEBSITE
REVENUE
100kkk
WHITE FOLDERS
$recycle.bin;config.msi;$windows.~bt;$windows.~ws;windows;appdata;application data;boot;google;mozilla;program files;program files (x86);programdata;system volume information;tor
WHITE FILES
autorun.inf;boot.ini;bootfont.bin;bootsect.bak;desktop.ini;iconcache.db;ntldr;ntuser.dat;ntuser.dat.log;ntuser.ini;thumbs.db
WHITE EXTENSIONS
386;adv;ani;bat;bin;cab;cmd;com;cpl;cur;deskthemepack;diagcab;diagcfg;diagpkg;dll;drv;hlp;icl;icns;ico;ics;idx;ldf;lnk;mod;mpa;msc;msp;msstyles;msu;nls;nomedia;ocx;prf;ps1;rom;rtp;scr;shs;spl;sys;theme;themep
WHITE HOSTS
PCname1;PCname2;PCname3
PROCESSES TO KILL
sql;oracle;ocssd;dbsnmp;synctime;agntsvc;isqlplussvc;xfssvccon;mydesktopservice;ocautoupds;encsvc;firefox;tbirdconfig;mydesktopqos;ocomm;dbeng50;sqbcoreservice;excel;infopath;msaccess;mspub;one
SERVICES TO KILL
vss;sql;svc$;memtas;mepocs;msexchange;sophos;veeam;backup;GxVss;GxBlr;GxFWD;GxCVD;GxCIMgr
ACCOUNTS FOR IMPERSONATIONS
Administrator:123QWEqwe!@#!@#
DELETE GPO DELAY
1
SELF-SPREAD
SPREAD METHOD
PSEXEC
GPO
DELETE EVENTLOGS
ENCRYPT FILENAME
LANGUAGE CHECK
NETWORK SHARES ENCRYPTION
RUNNING ONE
DESKTOP WALLPAPER
SHUT DOWN THE SYSTEM
KILL DEFENDER
SKIP HIDDEN FOLDERS
GPO PS UPDATE
ENCRYPTION MODE
AUTO
FAST
IMPERSONATION
KILL SERVICES
LOCAL DISKS
KILL PROCESSES
PRINT A NOTE
SET ICON
SELF-DELETE
WIPE FREE SPACE
SAME ENCRYPTION KEY
MAXIMUM DECRYPTOR PROTECTION

Embedded resources

The encrypter and decrypter templates are embedded into the builder's resource section:

```
-------------------------------------------------------------------------------------------
Resources
  [RCDATA]  100                    en-us  -->   RVA: 000090f0-00016af0   (da00 bytes)
  [RCDATA]  101                    en-us  -->   RVA: 00016af0-0003aaf0   (24000 bytes)
  [RCDATA]  103                    en-us  -->   RVA: 0003aaf0-0005e4f0   (23a00 bytes)
  [RCDATA]  106                    en-us  -->   RVA: 0005e4f0-00076af0   (18600 bytes)
```

- 100: LockBit 3.0 Decryptor (EXE)
- 101: LockBit 3.0 Encryptor (EXE)
- 103: LockBit 3.0 Encryptor (DLL)
- 106: LockBit 3.0 Encryptor (Reflective DLL)

An approach was proposed – based on the methodology of constructing the configuration parameters and how they were added to the selected payload – to figure out:

- How parameter configuration parsing is performed
- How data transformation is applied
- How the configuration is encrypted and then stored within the final binary

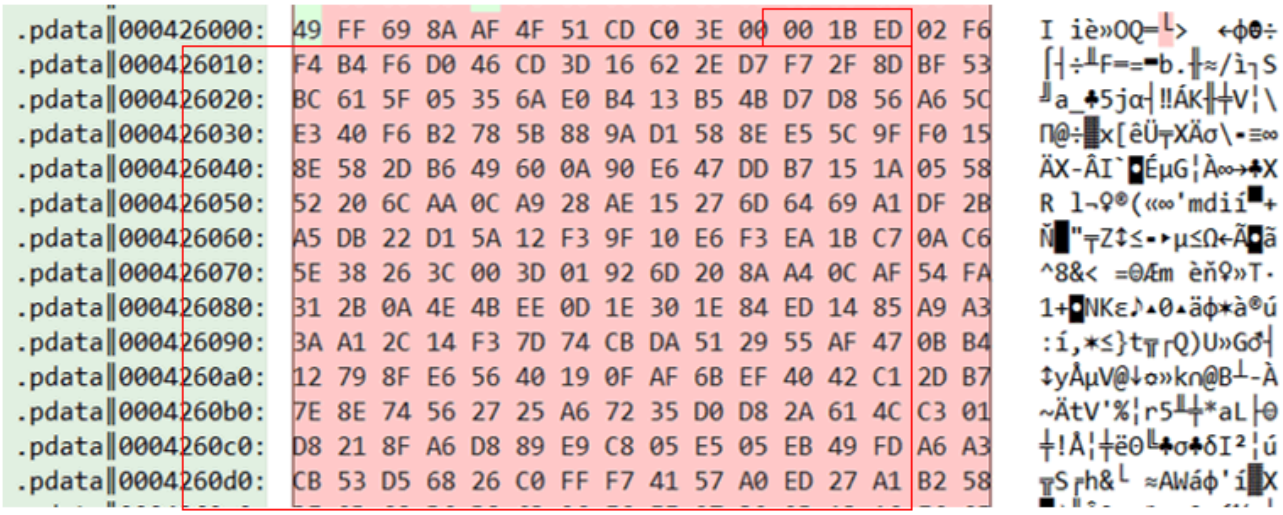
The payload-embedded configuration

The reverse-engineering analysis identified that the configuration is embedded in a section named .pdata, which is first encrypted using an XOR function with a key derived from a random seed and then compressed to embed it in the payload.

If the sample is configured to be encrypted using a password, the configuration will be similarly embedded in the binary first and then the sample will be encrypted with a unique key.

```
Sections
  .text    [R·X·····C··]  -->  Memory: 00401000-00418DE5 (   17de6 bytes)    Disk: 00000400-000181ff (   17e00
bytes)
  .itext   [R·X·····C··]  -->  Memory: 00419000-00419549 (     54a bytes)    Disk: 00018200-000187ff (     600
bytes)
  .rdata   [R·········I·]  -->  Memory: 0041A000-0041A47D (     47e bytes)    Disk: 00018800-00018dff (     600
bytes)
  .data    [RW·······I·]  -->  Memory: 0041B000-00425DBF (    adc0 bytes)    Disk: 00018e00-00022dff (    a000
bytes)
  .pdata   [RW·······I·]  -->  Memory: 00426000-00429ECB (    3ecc bytes)    Disk: 00022e00-00026dff (    4000
bytes)
  .reloc   [R·····D·I·]  -->  Memory: 0042A000-0042AFCF (     fd0 bytes)    Disk: 00026e00-00027dff (    1000
bytes)
```

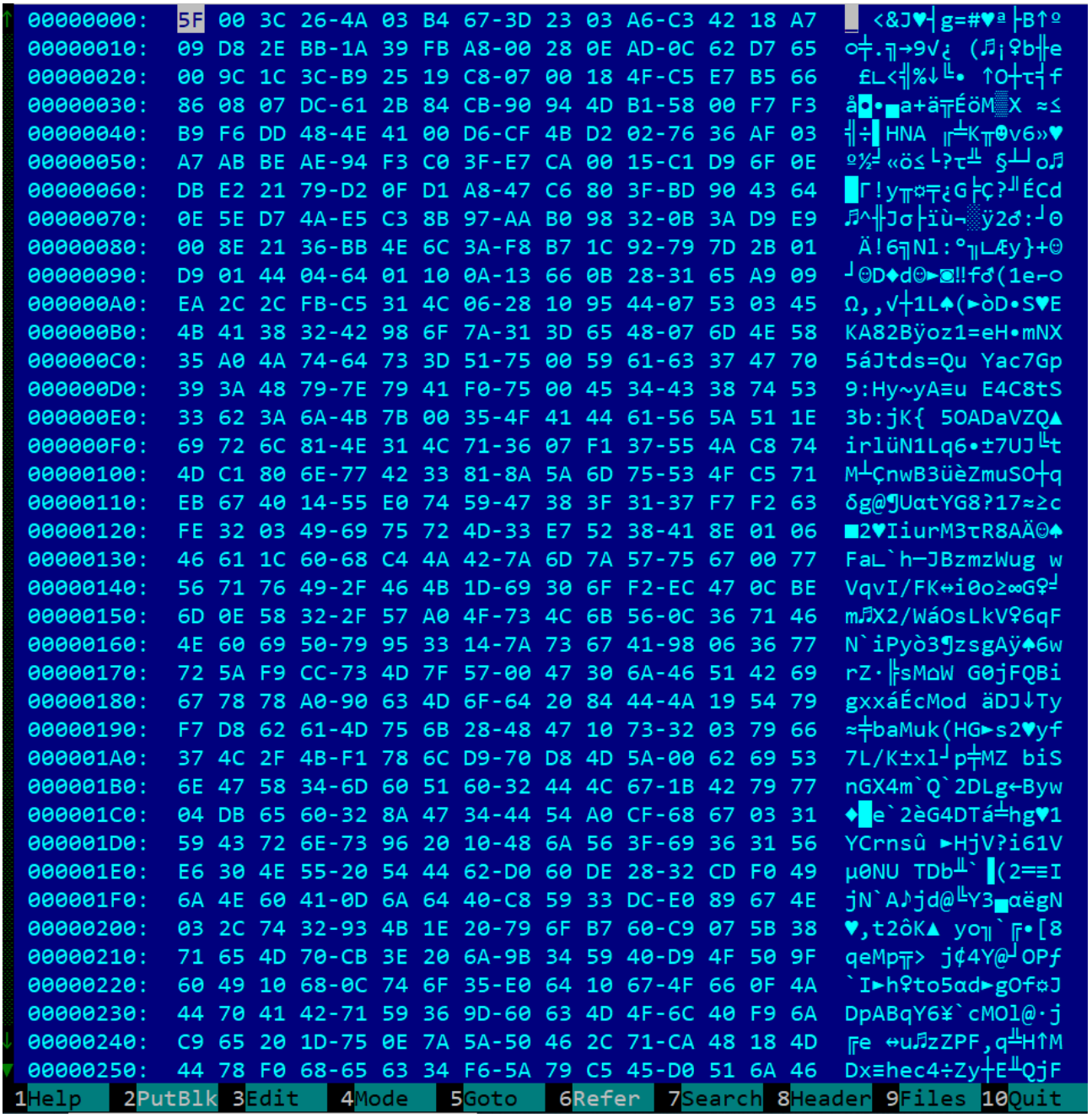
.pdata – this section contains the embedded configuration



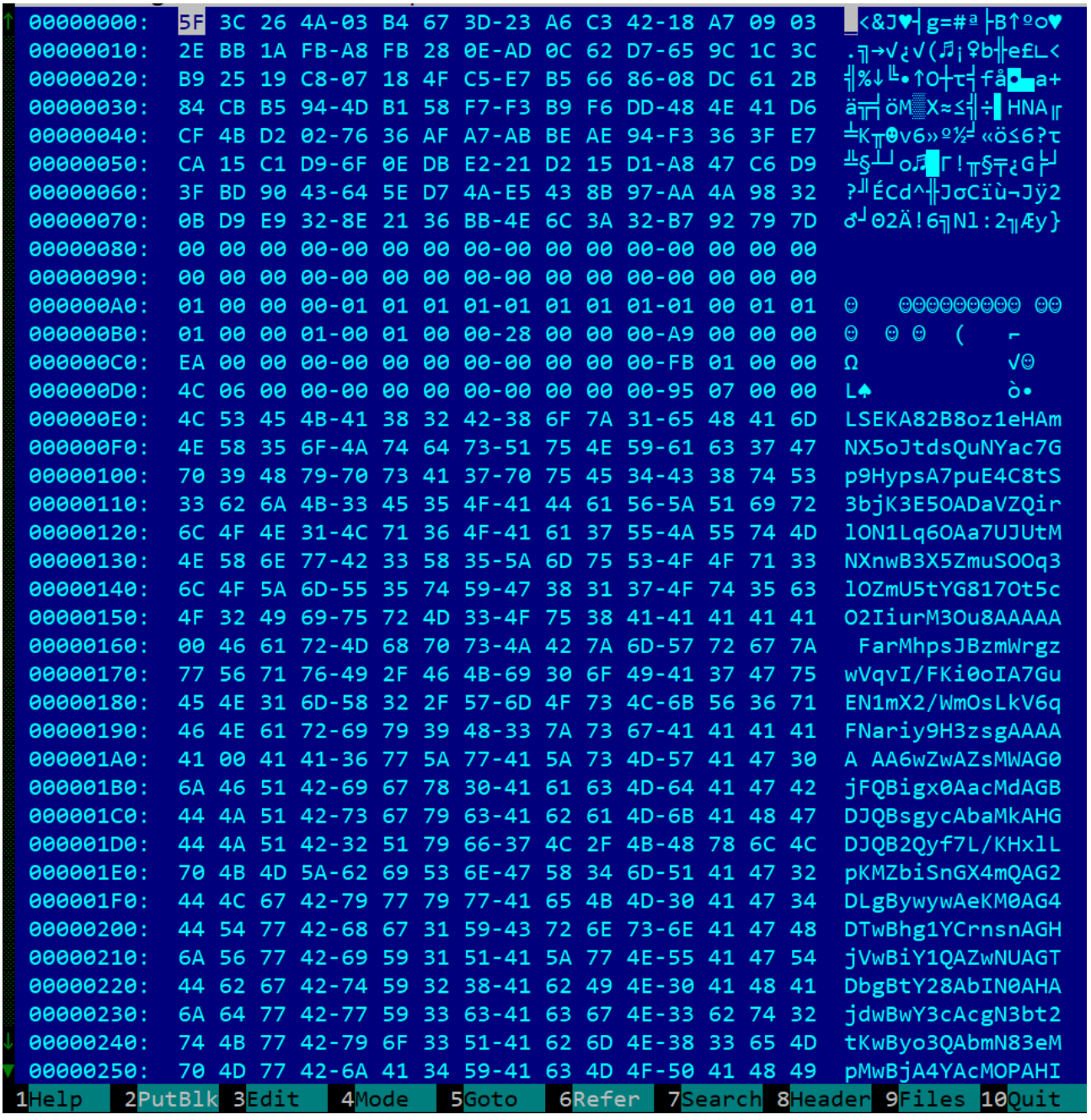
Embedded data (encrypted and compressed)

The creation of the XOR key, used to decrypt the content embedded in the section, depends on two random keys **along with other fixed values embedded in the binary source code.**

Decryption and subsequent decompression results in a set of sample configuration parameters, some of them with easily identifiable encryption mechanisms.



Decrypted section



Decompressed section

The next step is to interpret the fields and apply the required decryption to each of them to transform them into intelligible values.

The builder uses a custom hashing function that produces a 4-byte value for each of the values entered in the configuration parameters white_folders, white_files, white_extens and white_hosts. Other fields are stored with Base64 and ROR13.

Finally, interpreting the meaning of the fields in the config.json file and the relationship between the fields allows us to confirm that:

- Most configuration fields are easy to interpret based on their name and content.

- Some fields accept values only from a list of values.
- Many fields with string values are stored using ROR13 before being loaded into the payload configuration.
- Some fields accept multiple list values, using the “;” separator.
- Credentials must be stored in the format <user>:<password>.

```
{
"bot": {
       "uid": "<32digit-hexvalue>",
       "key": "<32digit-hexvalue>"
},
"config": {
       "settings": {
              "encrypt_mode": "<auto|fast>",
              "encrypt_filename": <false|true>,
              "impersonation": <false|true>,
              "skip_hidden_folders": <false|true>,
              "language_check": <false|true>,
              "local_disks": <false|true>,
              "network_shares": <false|true>,
              "kill_processes": <false|true>,
              "kill_services": <false|true>,
              "running_one": <false|true>,
              "print_note": <false|true>,
              "set_wallpaper": <false|true>,
              "set_icons": <false|true>,
              "send_report": <false|true>,
              "self_destruct": <false|true>,
              "kill_defender": <false|true>,
              "wipe_freespace": <false|true>,
              "psexec_netspread": <false|true>,
              "gpo_netspread": <false|true>,
              "gpo_ps_update": <false|true>,
              "shutdown_system": <false|true>,
              "delete_eventlogs": <false|true>,
              "delete_gpo_delay": <int>
       },
       "white_folders": "<list of foldername separated by ;>",
       "white_files": "<list of filename separated by ;>",
       "white_extens": "<list of extension separated by ;>",
       "white_hosts": "<list of hostname separated by ;>",
       "kill_processes": "<list of process name separated by ;>”,  used if KILL_PROCESSES flag is TRUE
       "kill_services": "<list of service name  separated by ;>",
       "gate_urls": "<list of URL separated by ; ex. https://test.white-datasheet.com/>",
       "impers_accounts": "<list of couple username:password separated by ;>",
       "note": "<ransom note>"
}
```

Config.json – what the fields mean

Based on these results, we defined a sample analysis procedure and applied it to multiple samples to determine the type of actors, objectives and construction preferences of the payloads.

Statistics of samples reported in our intelligence platforms

The objective of this analysis is to understand the parameters applied by different actors to build the malware as configured in samples detected in the wild.

During our research, **396** distinct samples were analyzed. According to the timestamps, mostly samples created by the leaked builders were detected, but other unknown builders dated June and July 2022 were also identified.

BUILDER/Build_timestamp	
Row Labels	**Count of sample_md5**
ALI_QUSHJI_LEAKED_BUILDER_VERS	**211**
14/09/2022 01:30	192
14/09/2022 01:31	19
PROTONLEAKS_LEAKED_BUILDER_VERS	**101**
09/09/2022 03:27	85
09/09/2022 03:26	16
UNKNOWN_BUILDER_VERS	**84**
14/07/2022 12:29	76
14/07/2022 12:29	5
14/07/2022 12:28	2
27/06/2022 16:56	1
Grand Total	**396**

General statistics of the embedded configuration:

- **Many of the detected parameters correspond to the default configuration of the builder**, only some contain minor changes. This indicates the samples were likely developed for urgent needs or possibly by lazy actors.
- The most recurrent encryption targets are local disks and network shares, avoiding hidden folders.
- The samples generally run a single instance and enable the following parameters:
  - **kill service**
  - **kill process**
  - **kill defender**
  - **delete logs**
  - **self-destruct**
- Most of the samples identified do not enable the system shutdown option.
- Network deployment by PSEXEC is configured in 90% of the samples, while deployment by GPO is configured in 72%.
- Very few samples enable communication to C2.

ENCRYPTION MODE	
Row Labels	Count of sample_md5
auto	367
fast	29
Grand Total	396

PSEXEC NET SPREAD	
Row Labels	Count of sample_md5
FALSE	357
TRUE	39
Grand Total	396

KILL SERVICES	
Row Labels	Count of sample_md5
TRUE	390
FALSE	6
Grand Total	396

KILL PROCESS	
Row Labels	Count of sample_md5
TRUE	390
FALSE	6
Grand Total	396

ENCRYPT FILENAME	
Row Labels	Count of sample_md5
FALSE	306
TRUE	90
Grand Total	396

GPO NET SPREAD	
Row Labels	Count of sample_md5
TRUE	286
FALSE	110
Grand Total	396

KILL DEFENDER	
Row Labels	Count of sample_md5
TRUE	387
FALSE	9
Grand Total	396

SEND REPORT	
Row Labels	Count of sample_md5
FALSE	386
TRUE	10
Grand Total	396

SKIP HIDDEN FOLDERS	
Row Labels	Count of sample_md5
FALSE	386
TRUE	10
Grand Total	396

LANGUAGE CHECK	
Row Labels	Count of sample_md5
FALSE	303
TRUE	93
Grand Total	396

DELETE EVENT LOG	
Row Labels	Count of sample_md5
TRUE	387
FALSE	9
Grand Total	396

PRINT NOTE	
Row Labels	Count of sample_md5
TRUE	340
FALSE	56
Grand Total	396

LOCAL DISK	
Row Labels	Count of sample_md5
TRUE	390
FALSE	6
Grand Total	396

RUNNING ONE	
Row Labels	Count of sample_md5
TRUE	384
FALSE	12
Grand Total	396

WIPE FREE SPACE	
Row Labels	Count of sample_md5
FALSE	316
TRUE	80
Grand Total	396

SET WALLPAPER	
Row Labels	Count of sample_md5
TRUE	359
FALSE	37
Grand Total	396

NETWORK SHARES	
Row Labels	Count of sample_md5
TRUE	390
FALSE	6
Grand Total	396

SHUTDOWN SYSTEM	
Row Labels	Count of sample_md5
FALSE	385
TRUE	11
Grand Total	396

SELF-DESTRUCT	
Row Labels	Count of sample_md5
TRUE	384
FALSE	12
Grand Total	396

SET ICONS	
Row Labels	Count of sample_md5
TRUE	363
FALSE	33
Grand Total	396

Detailed statistics

The C2 communication configuration showed it was rarely used and included three test domains. No suspicious or malicious domains were identified in the analyzed samples, showing there's no interest for establishing C2 communications using the leaked payloads.

Moreover, inside the configuration, the **impersonation data list** (credentials registered within the payload configuration) records general data with a default brute-force list. But it was possible to detect other binaries with specific data that allow identifying the organizations or individuals attacked.

It is important to keep in mind that Lockbit payloads and other ransomware actors integrate this type of information inside samples, and the handling of such samples must be done properly to avoid information leaks.

Finally, some statistics relate to the usage of leaked builders by actors other than the "original" Lockbit. We found that 77 samples make **no reference to a "Lockbit" string (case-insensitive) in the ransom note**, which is quite unexpected according to LB TTP.

Ransomnote content string (case-insensitive)	
Row Labels	**Count of sample_md5**
LockBit 3	264
-	77
LockBit	55
Grand Total	**396**

The modified ransom note **without reference to Lockbit** or with a **different contact address (mail/URL)** reveals probable **misuse of the builder by actors other than the "original" Lockbit.**

Source: https://securelist.com/lockbit-ransomware-builder-analysis/110370/