Malware analysis: Details on LockBit ransomware

seguranca-informatica.pt/malware-analysis-details-on-lockbit-ransomware/

October 5, 2021

lampion

Introduction

LockBit is a data encryption malware in operation since September 2019 and a recent Ransomware-as-a-Service (RaaS), in which developers are in charge of the payment site and development and affiliates sign up to distribute the threat in the wild. This piece of malware was developed to encrypt large companies in a few hours as a way of preventing its detection quickly by security appliances and IT/SOC teams. According to McAfee, LockBit encrypted approximately 25 servers and 225 workstations in just three hours during a recent attack.

When executed, the ransomware renames the files with the extension "**.abcd**" after compromising a device. After this process, a text file – "**Restore-My-Files.txt**" is created in all affected folders.

LockBit in depth

This malware is usually launched by criminals after a network has been compromised as one of the final stages of infection. LockBit deployment is launched via a PowerShell command also observed on other mediatic ransomware, including Netwalker.

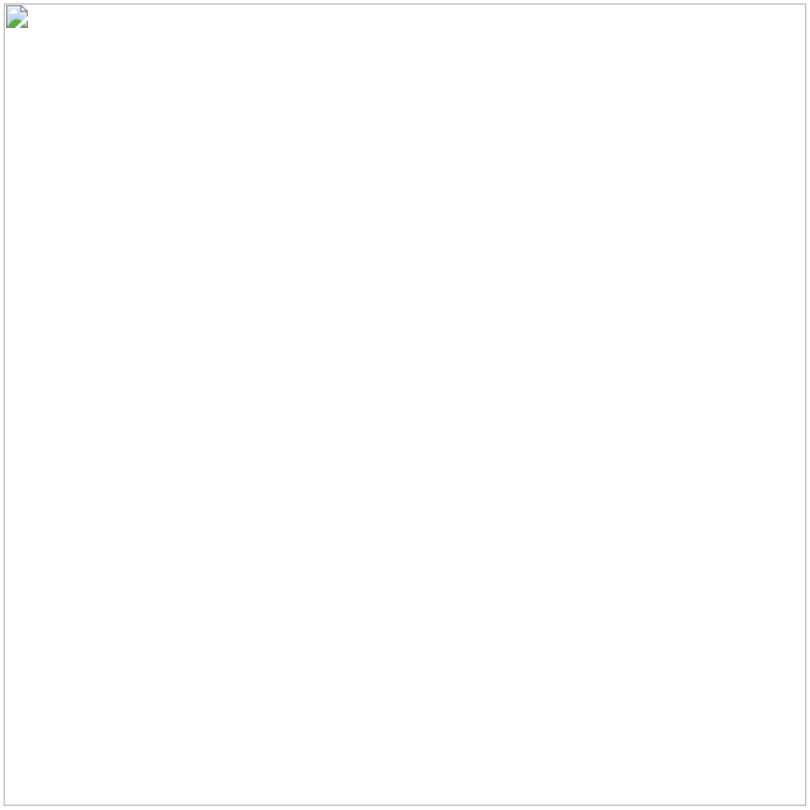

Figure 1: *PowerShell command launching the first stage of LockBit.*

In detail, the PowerShell command retrieves a .png file (rs40 and rs35 according to the .NET version installed on the infected device) from a website probably compromised by criminals, which starts the second stage. The second stage is a .NET downloader written in C# and compiled via Microsoft Visual Studio. As presented below, the binary has three sections and it is not packed, so it can be reversed for better understanding.

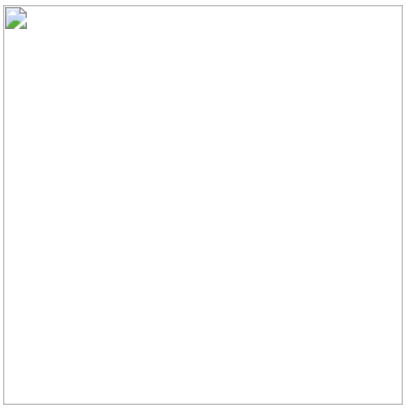

Figure 2: *LockBit second stage — number of sections.*

This file is a .NET loader that when executed downloads the final payload — LockBit — from the internet. Analyzing the **Main()** function, it shows that an array with the AES-encrypted base64 string contains the LockBit binary.

Figure 3: *String base64 encrypted with AES to protect the final payload — LockBit ransomware.*

In some observed samples, the symmetric encryption key ***ENCRYPTION29942*** was used to decrypt (and encrypt) the content observed in Figure 3.

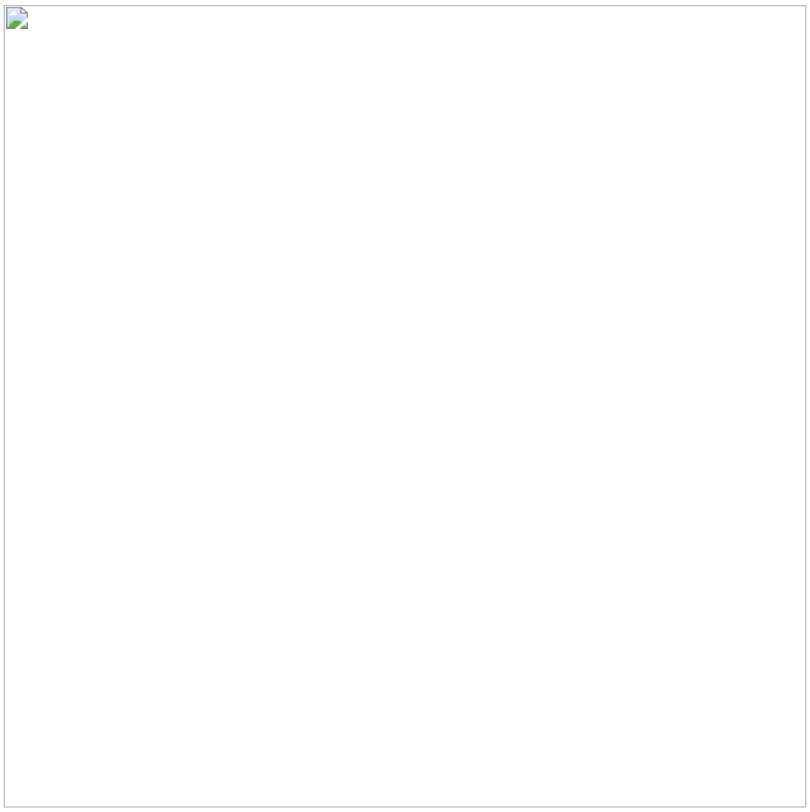

Figure 4: *AES ciphertext (the LockBit payload) and the symmetric decryption key.*

The .NET loader checks for the existence of vbc.exe on the infected device or downloads it from a compromised host (Figure 3). This binary is used for the process hollowing — a well-known technique for injecting LockBit into the memory.

In brief, the following calls were observed:

- **NtUnmapViewOfSection:** Unmap the original code in execution
- **NtWriteVirtualMemory:** Writing the base address of the injected image into the PEB
- **VirtualAllocEx:** Allocating the space before injecting the malicious payload

More details about this technique can be found here.

Regarding the VBC utility: it is the visual basic compiler for Windows. LockBit uses it to compile and execute the payload in runtime.

Figure 5: *VBC utility used to compile and execute the payload on the fly.*

After this, the ransomware is executed. It creates a new thread to manage services, terminate processes and delete the shadow volumes.

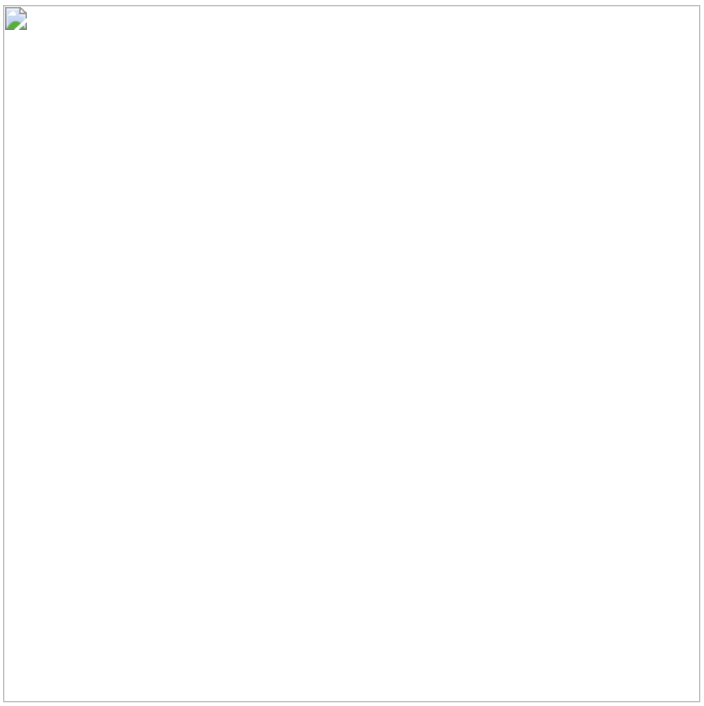

Figure 6: *Delete shadow copies from infected device.*

The ransomware has some entries hardcoded, related to services that are stopped during its execution.

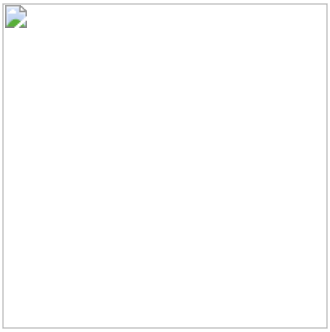

Figure 7: *Hardcoded services inside the binary.*

The completed list of services that LockBit tries to terminate are the following:

- DefWatch (Symantec Antivirus)
- ccEvtMgr (Norton AntiVirus Event Manager)
- ccSetMgr (Common Client Settings Manager Service of Symantec)
- SavRoam (Symantec Antivirus)
- sqlserv
- sqlagent
- sqladhlp
- Culserver
- RTVscan (Symantec Antivirus Program)
- sqlbrowser
- dbsrv12
- dbeng8 (Sybase's Adaptive Server
- SQLADHLP
- QBIDPService (QuickBooksby Intuit.)
- QuickBoooks.FCS (QuickBooksby Intuit.)
- QBCFMonitorService (QuickBooksby Intuit.)
- sqlwriter
- msmdsrv (Microsoft SQL Server)
- tomcat6 (Apache Tomcat)
- zhundongfangyu (this belongs to the 360 security product from Qihoo company)
- vmware-usbarbitator64
- vmware-converter
- Anywhere version 8 database program)
- wrapper

During the ransomware execution, a ransom note is created in each directory it encrypts with the following content:

Figure 8: *Ransom note dropped by LockBit during the encryption process.*

A TOR link is available and used to get additional information about the ransom payment and to establish a straight communication with the ransomware operators. As depicted below, a support chat is available, and also a feature named “Trial decrypt”, used to upload a damaged file and recover it. In short, this feature is just a proof-of-work provided by criminals.

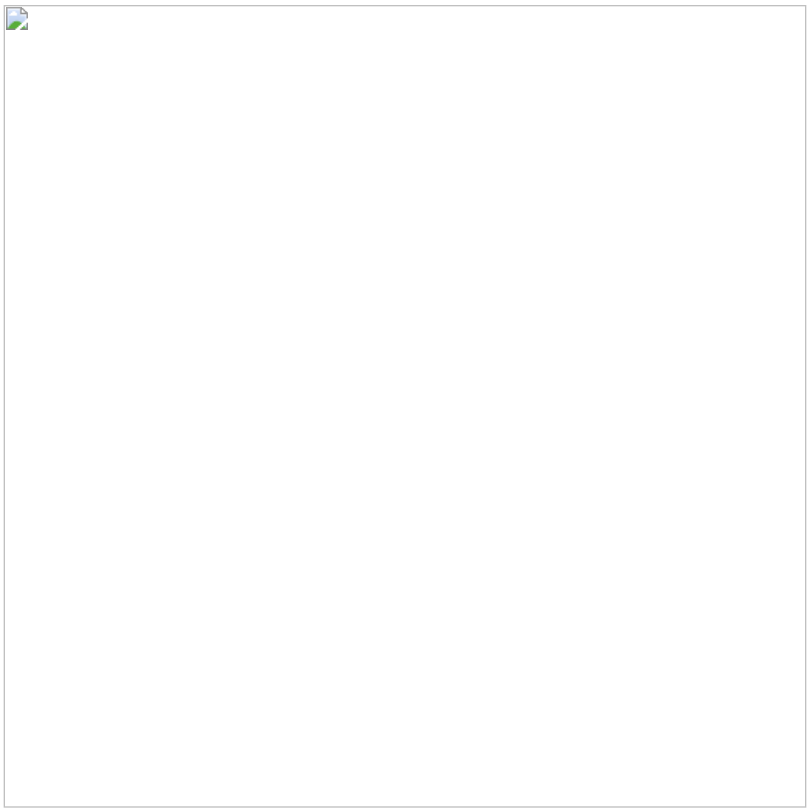

Figure 9: *Trial decrypt and chat — LockBit ransomware.*

The ransomware also performs lateral movement by using ARP protocol and via sending SMB requests on the network. If any device is accessed with success, it tries to execute the payload present in Figure 1 to infect the next device.

Another interesting detail is that the malware tries to eliminate itself at the end of the execution and creates two entries in the registry so that in potential infections it realizes that the computer has already been infected.

Figure 10: *Registry entries created during LockBit execution to prevent further infections.*

Prevention measures

This type of ransomware is not disseminated through social engineering attacks like phishing or spearphishing. It is deployed after a foothold on the network and to close the infection chain.

Because of this, fighting this threat is a headache! Monitoring can be a good friend, at least to detect potential intrusions on the network in an early stage. Monitoring network assets and to install endpoint protection agents in each device should be a mandatory rule present on the “security menu” for each company. This is an effective measure that can warn IT administrators about potential abnormal events on the entire ecosystem.

The article was initially published by Pedro Tavares on resources.infosecinstitute.com.

Pedro Tavares

Pedro Tavares is a professional in the field of information security working as an Ethical Hacker/Pentester, Malware Researcher and also a Security Evangelist. He is also a founding member at CSIRT.UBI and Editor-in-Chief of the security computer blog seguranca-informatica.pt.

In recent years he has invested in the field of information security, exploring and analyzing a wide range of topics, such as pentesting (Kali Linux), malware, exploitation, hacking, IoT and security in Active Directory networks. He is also Freelance Writer (Infosec. Resources Institute and Cyber Defense Magazine) and developer of the 0xSI_f33d – a feed that compiles phishing and malware campaigns targeting Portuguese citizens.

Read more here.