Money Ransomware: The Latest Double Extortion Group

yoroi.company/research/money-ransomware-the-latest-double-extortion-group/

April 13, 2023

04/13/2023

Introduction

Ransomware attacks have emerged as a predominant menace in recent years, with the strategies employed by malicious actors constantly evolving. Among the most effective and worrisome tactics is the "double extortion" model, which has rapidly gained popularity as a preferred business model for threat actors. Financially motivated perpetrators particularly favor the double extortion model, as it enables them to optimize their profits and bolster the likelihood of victims acquiescing to ransom demands.

In a double extortion assault, malefactors not only encrypt the targeted party's data but also exfiltrate sensitive information from the victim's system prior to encryption. The malicious actor subsequently issues a warning to publicize the purloined data unless the ransom is paid.

This deceptively simple yet exceedingly lucrative technique is increasingly being adopted by cybercriminals, leading to the emergence of new threats on a daily basis. One such example is the Money Ransomware group, which surfaced in March 2023. As of the time of writing, this nascent organization has already claimed two victims.

Figure 1: Leak site

```
Your files was encrypted by "Money message" profitable organization  and can't be accessed anymore.

If you pay ransom, you will get a decryptor to decrypt them. Don't try to decrypt files yourself - in that case they will be damaged and unrecoverable.

For further negotiations open this client4j6o5pilwvm2mwosianfwoqierbybff5uzd55hbz733jcvbpad.onion/chat.php?chatId=[redacted]
using tor browser https://www.torproject.org/download/

In case you refuse to pay, we will post the files we stole from your internal network, in our blog:

blogvl7tjyjvsfthobttze52w36wwiz34hrfcmorgvdzb6hikucb7aqd.onion

Encrypted files can't be decrypted without our decryption software.
```

Figure 2: Ransom Note Example

Technical Analysis

At the time of writing, we have been unable to completely unravel the infection chain of this emerging threat actor, primarily due to the limited number of targets attacked and the lack of evidence regarding their modus operandi. However, we do know that they employ a human-operated intrusion approach, evidenced by the method of data exfiltration and the execution of the malware sample.

We have managed to intercept a sample of the locker used to compromise the Bangladesh National Airport.

Hash	bbdac308d2b15a4724de7919bf8e9ffa713dea60ae3a482417c44c60012a654b
Threat	Money Ransomware
Brief Description	Locker of Money Ransomware

Table 1: File Info

Money Ransomware is engineered to accept either no parameters or just one during its execution. If more than one parameter is passed to the program, an error message will be logged. The program can be executed with or without parameters, but if multiple parameters are input, a log message will indicate that such execution is unsupported. This behavior suggests that the ransomware may be in the early stages of development.

The single parameter, if used, designates which drive where the sample will generate the Ransom Note "*money_message.log*".

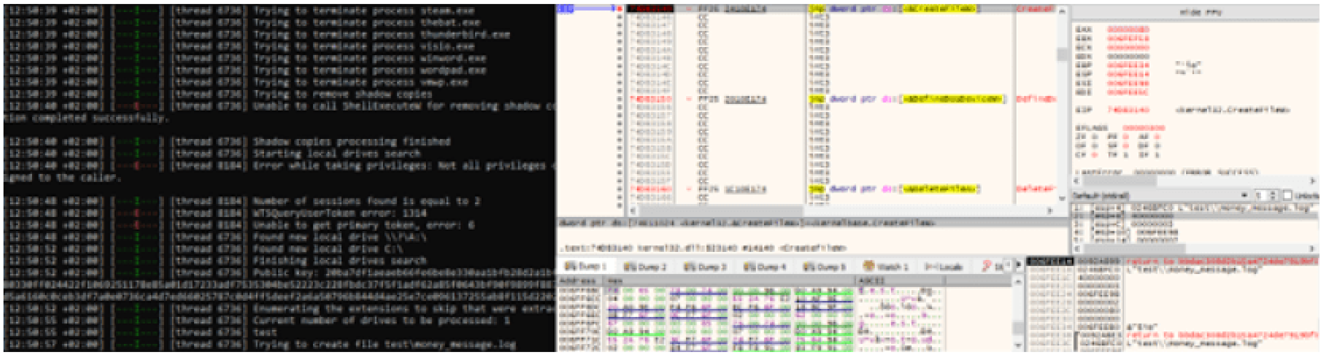

Figure 3: Execution of the Locker Sample

By performing static analysis, it becomes evident that the code is still in its infancy, as numerous code smells can be found within the binary. One notable example is the unobscured configuration data located in the overlay section of the compiled file.

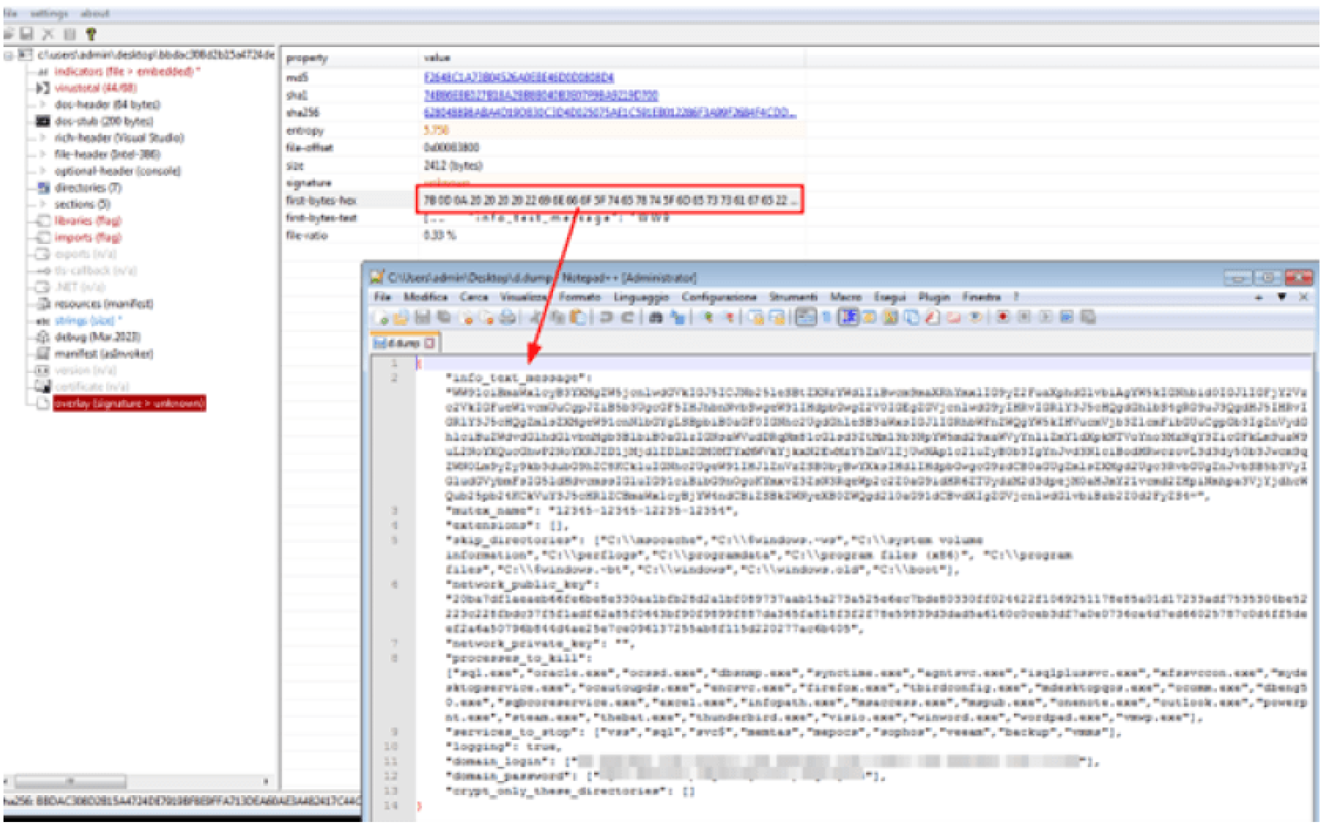

Figure 4: Configuration stored in clear in the overlay

In the following table we summarize all the parameters in the configuration file:

Key	Description
info_text_messag	Base64 Encoded Ransom Note
mutex_name	String used as Mutex
extensions	Extensions to skip
skip_directories	Directories to skip

network_public_key	
network_private_key	
processes_to_kill	Names of processes to kill
services_to_stop	Name of services to stop
logging	Boolean, to print logs
domain_login	List of domain usernames
domain_password	List of domain passwords
crypt_only_these_directories	List of directories to encrypt

Table 2: Description of the Config File

The first phase of the execution of the malware is to install a Mutex in order to keep track of the already locked machines. But, if the mutex creation fails, the infection goes on, with the risk to encrypt a second time the machine.

Then, Money Ransomware removes the shadow copies by executing vssadmin, but before doing that, it disables the redirection to WOW64 directory, in order to force the execution of the command from the System32 Directory.

The next phase of the locking process is to kill the processes which can get an handle to file to encrypt.

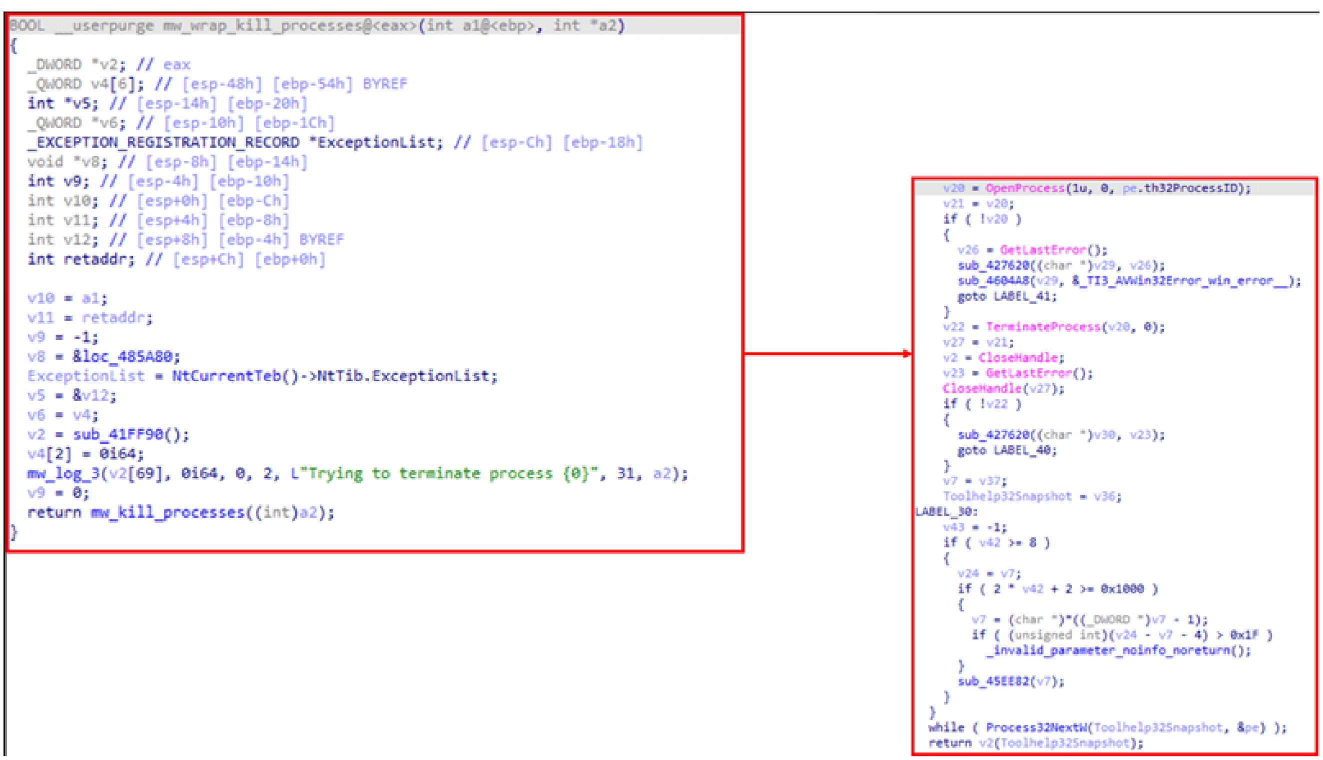

Figure 5: Kill Processes routine

The list of the processes to kill is the following:

- sql.exe
- oracle.exe
- ocssd.exe
- dbsnmp.exe
- synctime.exe
- agntsvc.exe
- isqlplussvc.exe
- xfssvccon.exe
- mydesktopservice.exe
- ocautoupds.exe
- encsvc.exe
- firefox.exe
- tbirdconfig.exe
- mdesktopqos.exe
- ocomm.exe
- dbeng50.exe
- sqbcoreservice.exe
- excel.exe
- infopath.exe
- msaccess.exe
- mspub.exe
- onenote.exe
- outlook.exe
- powerpnt.exe
- steam.exe
- thebat.exe
- thunderbird.exe
- visio.exe
- winword.exe
- wordpad.exe
- vmwp.exe

Subsequently, the malware proceeds to halt services that could potentially disrupt the encryption process. In this particular instance, not only are system utilities targeted, but also anti-malware software, such as Sophos. It is important to note that the processes and services to be terminated are contingent upon the configuration file. Thus, we can deduce that the threat actor is aware of the victim's use of Sophos as their anti-malware solution. For other victims, the attacker could customize the file to disable different services accordingly.

The list of services targeted in this specific case includes:

- vss
- sql
- svc$
- memtas
- mepocs
- sophos
- veeam
- backup
- vmms

Figure 6: Service Stop Routine

Figure 6 illustrates that the malware employs two distinct methods to attempt to halt Windows services: the first method utilizes Microsoft's WMIC utility, while the second relies on the SCManager* Windows APIs.

Following this, the ransomware extracts information from the configuration file to identify directories that should be exempt from encryption. In this specific instance, the folders to be bypassed include:

- C:\\msocache
- C:\\$windows.~ws
- C:\\system volume information
- C:\\perflogs
- C:\\programdata
- C:\\program files (x86)
- C:\\program files
- C:\\$windows.~bt
- C:\\windows
- C:\\windows.old
- C:\\boot

One of the most serious capabilities of the ransomware is the ability to propagate the locking process through the network. It uses two different ways to perform that operation. The first one is to iterate and inside all the connected devices of the machine.

The second one is sneakier, because it attempts to login to hardcoded domain accounts using the API function WNetAddConnection2W. WNetAddConnection2W is a Windows API function that allows a program to connect to network resources, such as shared drives or printers, by establishing a network connection. This function enables connection establishment using specified usernames and passwords, and it also permits the user to dictate whether the connection should be remembered and reconnected automatically in the future.

WNetAddConnection2W works by trying to connect to a network resource, like a network share or cloud storage service, using a series of compromised user credentials. These credentials are stored within Money Ransomware's configuration file. This behavior indicates that the ransomware operators have obtained compromised credentials from prior privilege escalation activities.

Once the connection is established, the ransomware can then encrypt the files stored on the network resource, in addition to those stored locally on the victim's computer.

Figure 7: Accessing to remote resources abusing compromised credentials and WNetAddConnection2W API

For the encryption process, the ransomware employs a combination of the Elliptic Curve Diffie-Hellman (ECDH) and ChaCha20 algorithms. By doing so, the malware effectively harnesses the robust asymmetric encryption capabilities provided by ECDH, along with the high performance of ChaCha20, to swiftly encrypt all files within the victim's machine.

```
v5 = a1;
v29 = a1;
v34[4] = *a4;
v34[5] = a4[1];
v34[6] = a4[2];
v34[7] = a4[3];
v34[8] = a4[4];
v34[9] = a4[5];
v34[10] = a4[6];
v34[11] = a4[7];
qmemcpy(v34, "expand 32-byte k", 16);
v35 = *a5;
v36 = a5[1];
v37 = a5[2];
v38 = a5[3];
for ( result = a3; result; a3 = result )
{
  v7 = 64;
  if ( result < 0x40 )
    v7 = result;
  v30 = v7;
  sub_424000(&v31, v34);
  v8 = 0;
  if ( v7 )
  {
    if ( v7 < 0x40 || (v9 = v7 + v5 - 1, v5 <= (unsigned int)&v30 + v7 + 3) && v9 >= (unsigned int)&v31 )
    {
      v10 = a2;
    }
    else
    {
      v10 = a2;
      if ( v5 > v7 + a2 - 1 || v9 < a2 )
      {
        v11 = a2 + 48;
```

```
if ( !sub_447810((int)v24, (int)Src) )
{
  sub_414FE0(v20, "Unable to generate keypair with ecdh_generate_keys");
  LOBYTE(v27) = 1;
  sub_42B370(v20);
  sub_42B6E0((char *)v17, v20);
  sub_4604A8(v17, &_TI3_AVECDH_Exception_shared_secret__);
  goto LABEL_12;
}
```

Figure 8: Encryption Algorithm

Another technique adopted by the ransomware to manage the file encryption process involves checking the file's footer. By using the SetFilePointerEx API call, the ransomware moves the file pointer to -172 from the end, searching for the hexadecimal pattern "90 00 00 00", which indicates the start of the footer. This approach helps prevent the encryption of the same file twice. Following this pattern, the ransomware writes 168 bytes, which encompass the necessary information to enable the decryption of the encrypted file.

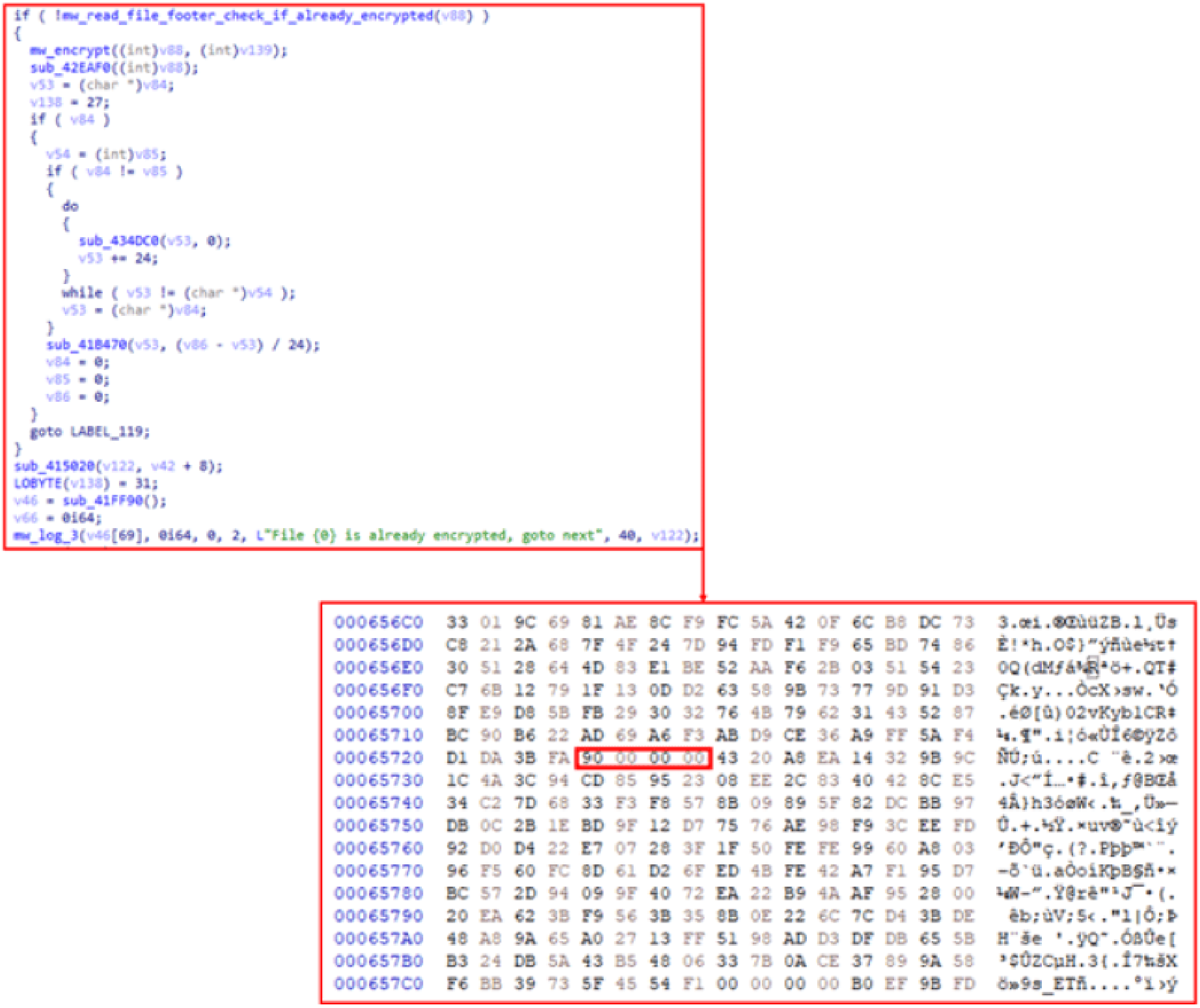

Figure 9: Already Encrypted Check

In the end, we can summarize the malware control flow in the following figure:

Figure 10: Money Ransomware Control Flow

Conclusion

Money Ransomware is part of a growing trend of ransomware attacks that have been on the rise since 2019, targeting the encryption, theft, and exfiltration of sensitive data. It is crucial to examine, as discussed in the technical details, the way these attacks are executed. Ransomware payloads do not necessarily require high levels of sophistication if a well-organized and optimized intrusion underlies the ransomware's deployment.

Additionally, another issue that has emerged in this case and others is the problem of propagation, which involves the abuse of legitimate API calls. For example, the infamous BlackCat/AlphV ransomware demonstrated the misuse of API calls to elevate its privileges during execution; in the case of Money Ransomware, API calls have been abused to propagate within remote shared resources. This poses a significant concern for organizations, as a single infected system can rapidly result in extensive damage and data loss.

To mitigate this risk, it is vital for organizations to adopt a proactive approach to network security. This includes regularly patching and updating software, employing firewalls and other network security tools, and educating employees on how to recognize and avoid common phishing and social engineering attacks. By taking these measures, organizations can reduce their risk of succumbing to ransomware attacks and safeguard their valuable data from harm.

Indicators of Compromise

Hash:

bbdac308d2b15a4724de7919bf8e9ffa713dea60ae3a482417c44c60012a654b

Yara Rules

```
rule money_ransomware
{
        meta:
                author = "Yoroi Malware ZLab"
        description = "Rule for Money Ransomware"
        last_updated = "2023-03-28"
        tlp = "WHITE"
        category = "informational"
        strings:
                // 0x00445F00 mw_remove_shadow_copies
                $1 = { 68 ?? ?? ?? ?? 68 ?? ?? ?? ?? c7 45 e8 00 00 00 00 ff 15 ?? ??
?? ?? 50 ff 15 ?? ?? ?? ?? 8b f0 85 f6 0f 84 ?? ?? ?? ?? eb ?? 8b 4d e0 8b 01 ff 50
04 89 45 e4 8d 45 e4 50 83 ec 08 8b c4 c7 00 ?? ?? ?? ?? c7 40 04 3e 00 00 00 e8 ??
?? ?? ?? 83 c4 0c b8 ?? ?? ?? ?? c3  }

                // 0044352D -> 00443566 mw_parse_config
                $2 = {8d 47 30 3b c6 74 ?? 8b c8 e8 ?? ?? ?? ?? 8b 0e 89 4f 30 8b 46
04 89 47 34 8b 46 08 89 47 38 c7 06 00 00 00 00 c7 46 04 00 00 00 00 c7 46 08 00 00
00 00 8d ?? 14 ff ff ff e8 ?? ?? ?? ??}
        condition:
                uint16(0) == 0x5A4D and ($1 or $2)
}
```

This blog post was authored by Luigi Martire, Carmelo Ragusa of Yoroi Malware ZLAB