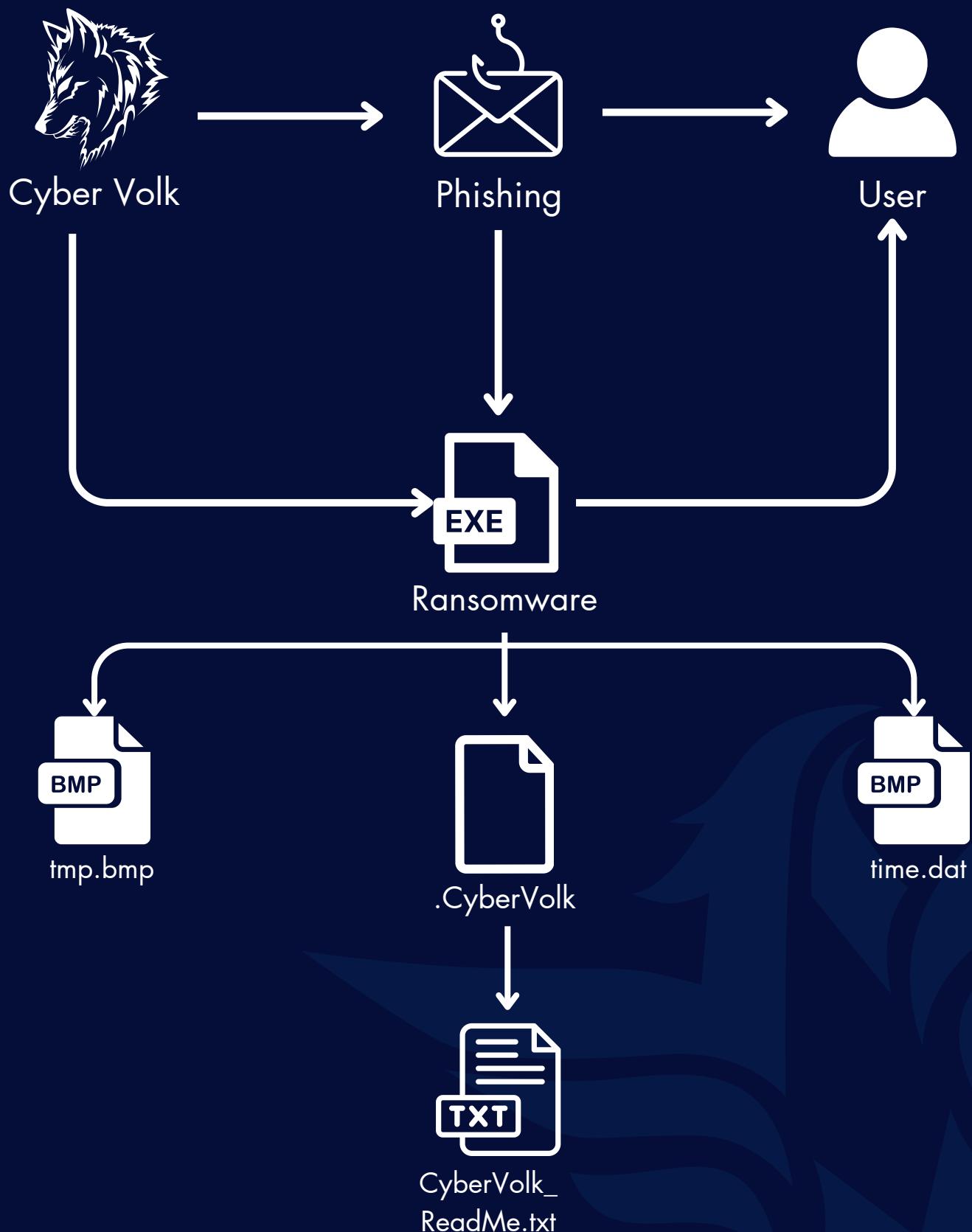
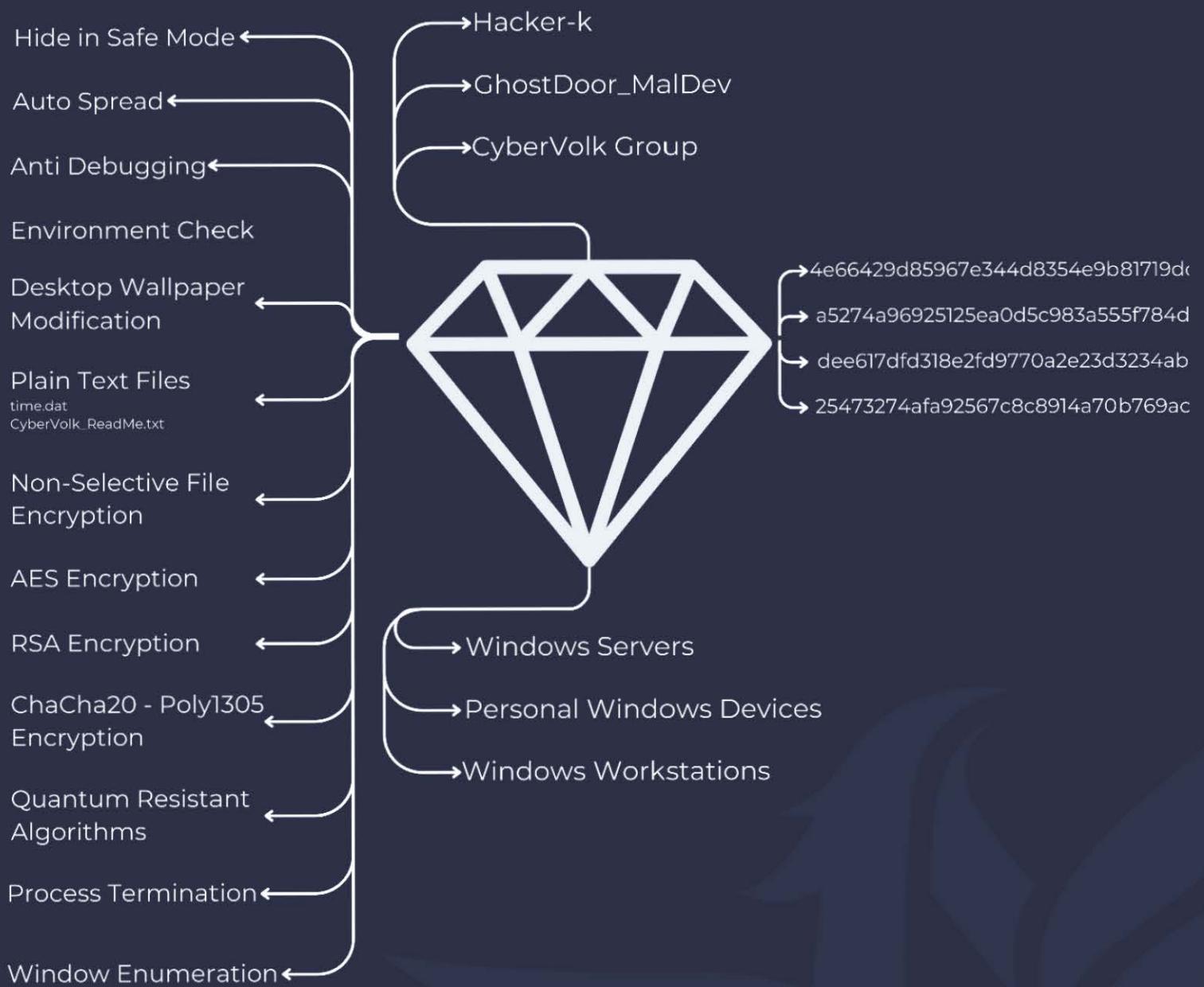


CYBERVOLK RANSOMWARE TECHNICAL & MALWARE ANALYSIS

ATTACK CHAIN



DIAMOND MODEL



About Cybervolk Ransomware

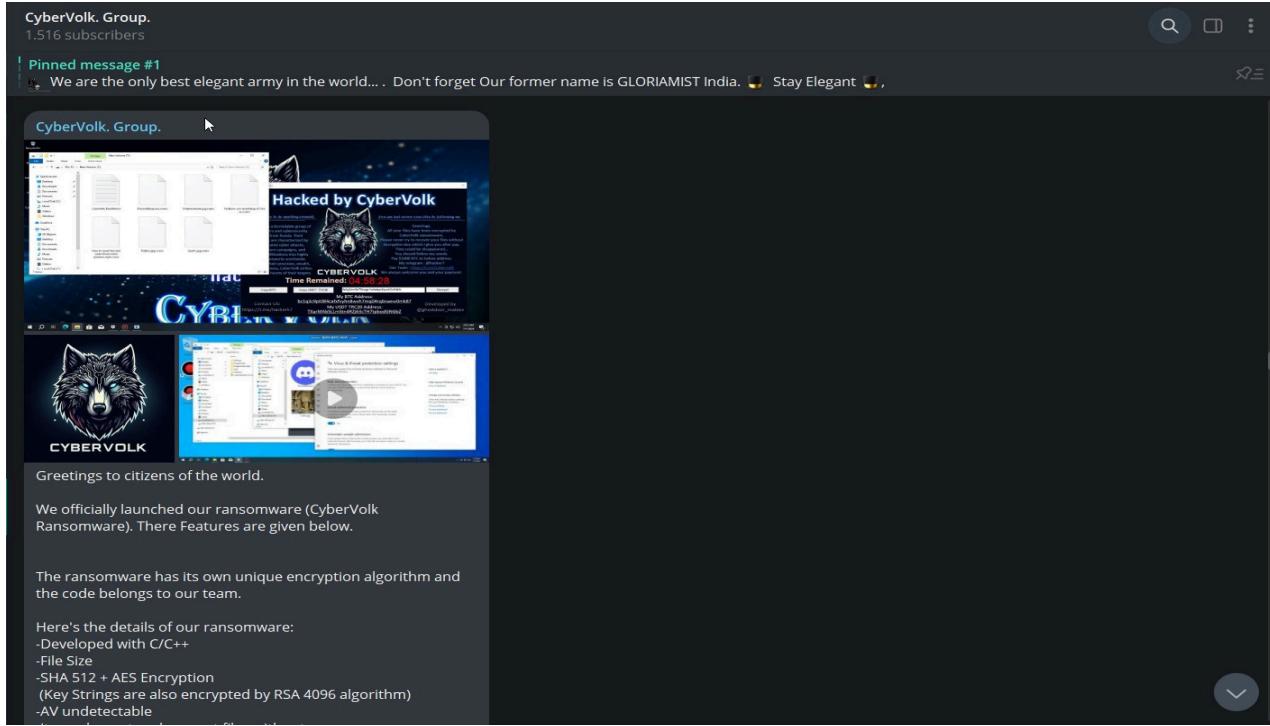


Image of CyberVolk Group Telegram Post

CyberVolk Ransomware was first completed on July 1, 2024, and it was detected being marketed as RaaS (Ransomware as a Service) on the dark web and Telegram on July 3, 2024. The initial ransomware was developed in the C++ language and, like most ransomware, uses the AES encryption algorithm. The SHA512 hash algorithm is used for AES key generation.

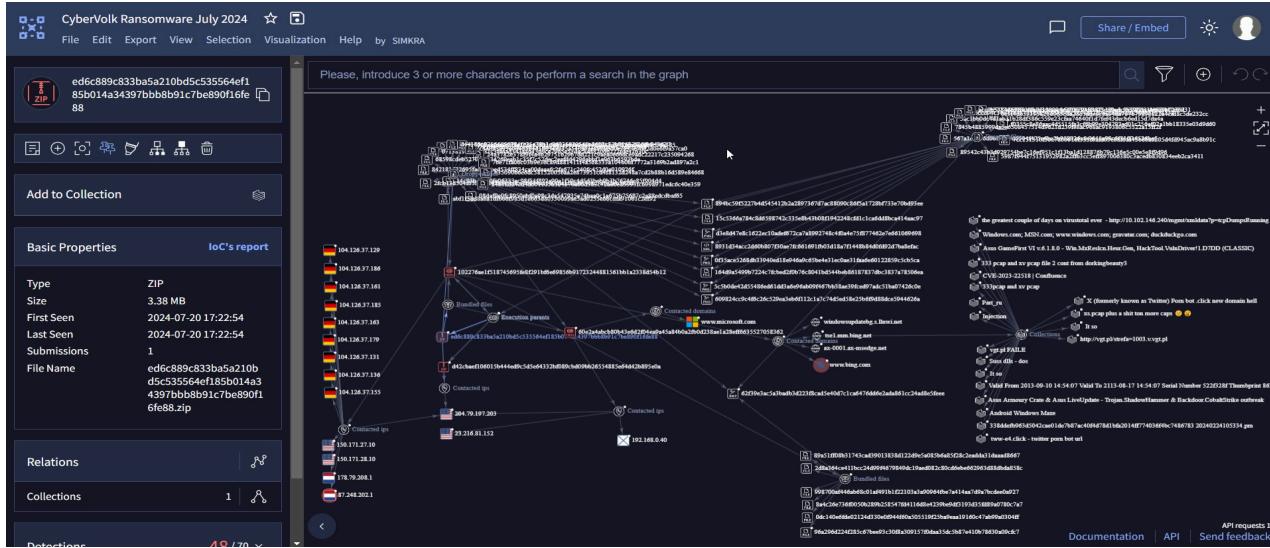


Image of CyberVolk Ransomware VirusTotal Leak

However, the initial ransomware (with the .cvenc extension) was leaked on VirusTotal and rendered non-functional. Consequently, CyberVolk subjected the ransomware to a significant update, making many changes within the ransomware.



A Quick Look into the CyberVolk Ransomware

After the successful unpacking of AzzaSec Ransomware, its basic characteristics have changed as follows:



Image of CyberVolk Ransomware

After running on the system, CyberVolk ransomware directly displays the payment screen and begins encrypting all files by restricting user activities within the system. It prevents applications like Task Manager from opening to ensure the encryption process is not interrupted, and it encrypts all files in a short time.

The ransomware gives the user a 5-hour window to make the payment. Additionally, it creates a Readme.txt file within the system.

```
Greetings.
All your files have been encrypted by CyberVolk ransomware.
Please never try to recover your files without decryption key which I give you after pay.
They could be disappeared?
You should follow my words.
Pay $1000 BTC to below address.
My telegram : @hacker7
Our Team : https://t.me/cubervolk
We always welcome you and your payment.
```

Image of CyberVolk Ransomware Readme.txt

In the **Readme.txt**, it is observed that a payment of **\$1,000** is demanded within this 5-hour.

If the **\$1,000** payment is not made, data loss occurs within the infected system.



Technical Malware Analysis

Basic Characteristics of CyberVolk Ransomware

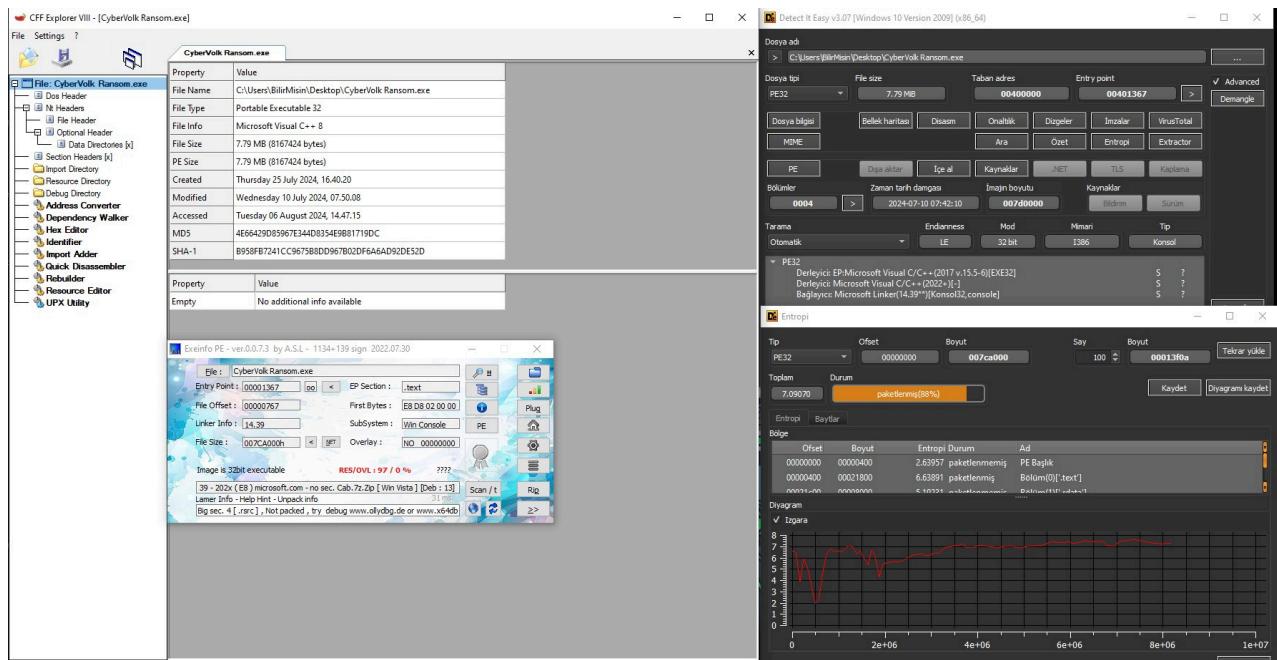


Image of CyberVolk Ransomware Characteristics

When examining the file features of the CyberVolk Ransomware, it is observed that it is developed in C++, has a size of 7.79MB, and does not use any packer.

FileType	Portable Executable 32
Language	C++
FileSize	7.79 MB 8167424 bytes
PeSize	7.79 MB 8167424 bytes
Packer	Not Packed
MD5	4E66429D85967E344D8354E9B81719DC
SHA1	B958FB7241CC9675B8DD967B02DF6A6AD92DE52D
Sha256	de0b74917fe24c2b38e2d1172b7352f88bf8b3df64b6d44ca5f317db85aeb324
IMPHash	0982e392aba6a868dc7bda8b61e977ab



Dynamic Analysis of CyberVolk Ransomware

The screenshot shows the debugger interface with assembly code and memory dump panes. A red box highlights the assembly code for creating a temporary file. The memory dump pane shows the file path and the file content.

Image of CyberVolk Ransomware Dynamic Analysis I

It is observed that CyberVolk ransomware starts its process by writing a BMP file to the \$HOME\\AppData\\\\Temp directory. The BMP file is then set as the background image.

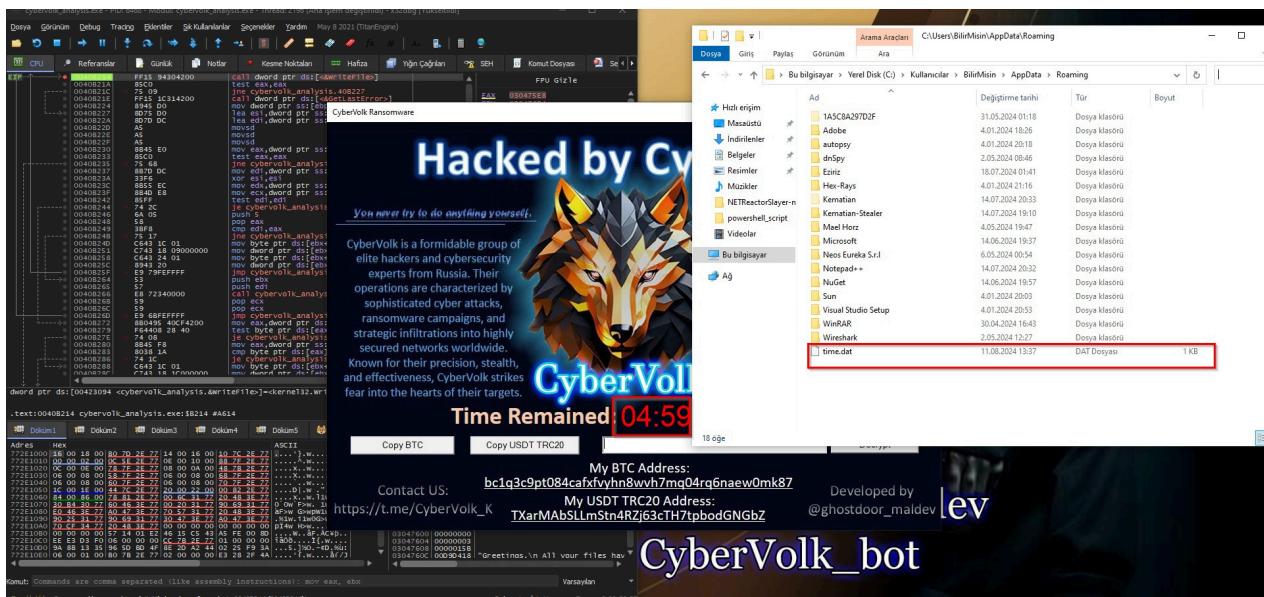


Image of CyberVolk Ransomware Dynamic Analysis II

Then it prints the "time.dat" file to the system and starts the GUI. A time of 5 hours is specified in "time.dat" and a timer is set on the GUI according to the data written there.



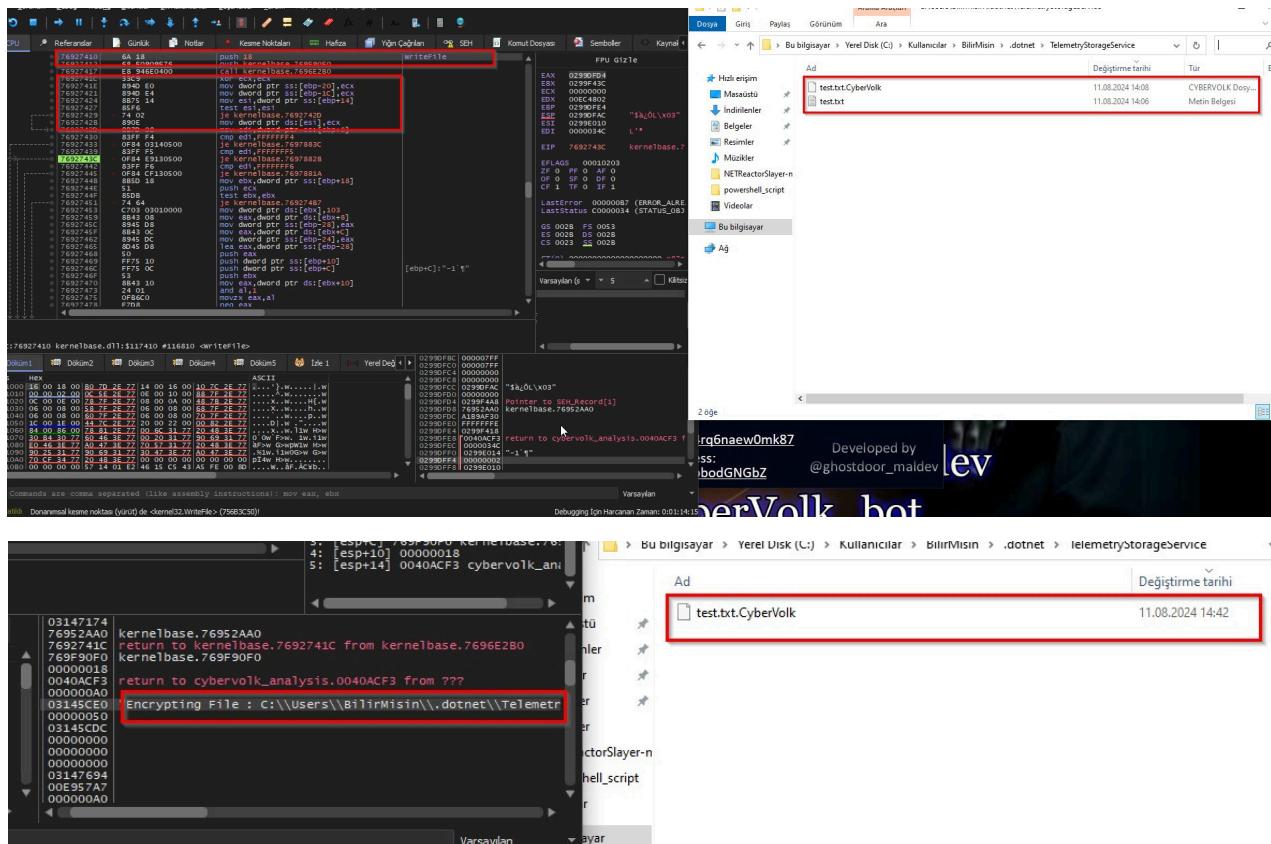


Image of CyberVolk Ransomware Dynamic Analysis III

After creating the **time.dat** file, it starts encryption from the first directory of the **\$HOME** directory. Firstly, it creates a file with **.CyberVolk** extension and then encrypts it by reading the contents of the file, then writes the encrypted data into the file with .CyberVolk extension. Then it deletes the unencrypted file from the system.

```
// Token: 0x0600002A RID: 42 RVA: 0x00002990 File Offset: 0x00000B90
public static object Math_Decryption_Algorithm_2(string input, string pass)
{
    RijndaelManaged rijndaelManaged = new RijndaelManaged();
    MD5CryptoServiceProvider md5CryptoServiceProvider = new MD5CryptoServiceProvider();
    object obj;
    try
    {
        byte[] array = new byte[32];
        byte[] array2 = md5CryptoServiceProvider.ComputeHash(MathChecker.KO(pass));
        Array.Copy(array2, 0, array, 0, 16);
        Array.Copy(array2, 0, array, 15, 16);
        rijndaelManaged.Key = array;
        rijndaelManaged.Mode = CipherMode.ECB;
        ICryptoTransform cryptoTransform = rijndaelManaged.CreateDecryptor();
        byte[] array3 = Convert.FromBase64String(input);
        string text = MathChecker.LALAK(cryptoTransform.TransformFinalBlock(array3, 0, array3.Length));
        obj = text;
    }
    catch (Exception ex)
    {
    }
    return obj;
}
```

Image of CyberVolk Ransomware Dynamic Analysis IV



According to the CyberVolk Group's post on Telegram, on July 23 2024, significant updates occurred in the ransomware after the leak on VirusTotal.

"Greetings.

All your files have been encrypted by CyberVolk ransomware.

Please never try to recover your files without decryption key which I give you after pay.

They could be disappeared?

You should follow my words.

Pay \$1000 BTC to below address.

My telegram : @hacker7

Our Team : <https://t.me/cubervolk>

We always welcome you and your payment."

5:19...	cybervolt_analysis.exe	4180	Process Start		SUCCESS	Parent PID: 4100...
5:19...	cybervolt_analysis.exe	4180	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Control\Session Manager	REPARSE	Desired Access: Q...
5:19...	cybervolt_analysis.exe	4180	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Control\Session Manager	SUCCESS	Desired Access: Q...
5:19...	cybervolt_analysis.exe	4180	RegQueryValue	HKEY\SYSTEM\CurrentControlSet\Control\Session Manager\RaiseExceptionOnPossibleDeadlock	NAME NOT FOUND Length: 80	
5:19...	cybervolt_analysis.exe	4180	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Control\Session Manager\Segment Heap	REPARSE	Desired Access: Q...
5:19...	cybervolt_analysis.exe	4180	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Control\Session Manager\Segment Heap	SUCCESS	Desired Access: Q...
5:19...	cybervolt_analysis.exe	4180	RegQueryValue	HKEY\SYSTEM\CurrentControlSet\Control\Session Manager\Segment Heap	NAME NOT FOUND Length: 24	
5:19...	cybervolt_analysis.exe	4180	RegOpenKey	HKEY\Software\Microsoft\Wow64v86	SUCCESS	Desired Access: R...
5:19...	cybervolt_analysis.exe	4180	RegQueryValue	HKEY\SOFTWARE\Microsoft\Wow64v86\cybervolt_analysis.exe	NAME NOT FOUND Length: 520	
5:19...	cybervolt_analysis.exe	4180	RegQueryValue	HKEY\SOFTWARE\Microsoft\Wow64v86(Default)	SUCCESS	Type: REG_SZ, Le...
5:19...	cybervolt_analysis.exe	4180	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Control\Session Manager	REPARSE	Desired Access: Q...
5:19...	cybervolt_analysis.exe	4180	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Control\Session Manager	SUCCESS	Desired Access: Q...
5:19...	cybervolt_analysis.exe	4180	RegQueryValue	HKEY\SYSTEM\CurrentControlSet\Control\Session Manager\RaiseExceptionOnPossibleDeadlock	NAME NOT FOUND Length: 80	
5:19...	cybervolt_analysis.exe	4180	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Control\Session Manager\Segment Heap	REPARSE	Desired Access: Q...
5:19...	cybervolt_analysis.exe	4180	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Control\Session Manager\Segment Heap	SUCCESS	Desired Access: Q...
5:19...	cybervolt_analysis.exe	4180	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Control\Session Manager	REPARSE	Desired Access: Q...
5:19...	cybervolt_analysis.exe	4180	RegQueryValue	HKEY\SYSTEM\CurrentControlSet\Control\Session Manager\ResourcePolicies	SUCCESS	Desired Access: Q...
5:19...	cybervolt_analysis.exe	4180	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Services\bam\State\UserSettings\S-1-5-21-1690830767-3441749873-8510784...	NAME NOT FOUND Desired Access: Q...	
5:19...	cybervolt_analysis.exe	4180	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Services\bam\State\UserSettings\S-1-5-21-1690830767-3441749873-8510784...	SUCCESS	Desired Access: All...
5:19...	cybervolt_analysis.exe	4180	RegQueryValue	HKEY\SYSTEM\CurrentControlSet\Services\bam\State\UserSettings\S-1-5-21-1690830767-3441749873-8510784...	Type: REG_BINA...	
5:19...	cybervolt_analysis.exe	4180	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Control\Session Manager\BAM	REPARSE	Desired Access: Q...
5:19...	cybervolt_analysis.exe	4180	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Control\Session Manager\BAM	NAME NOT FOUND Desired Access: Q...	
5:19...	cybervolt_analysis.exe	4180	Process Create	C:\Windows\System32\Conhost.exe	SUCCESS	PID: 1876, Comma...
5:19...	cybervolt_analysis.exe	4180	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Control\StateSeparation\RedirectionMap\Keys	REPARSE	Desired Access: R...
5:19...	cybervolt_analysis.exe	4180	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Control\StateSeparation\RedirectionMap\Keys	NAME NOT FOUND Desired Access: R...	
5:19...	cybervolt_analysis.exe	4180	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Control\SafeBoot\Option	REPARSE	Desired Access: Q...
5:19...	cybervolt_analysis.exe	4180	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Control\SafeBoot\Option	NAME NOT FOUND Desired Access: Q...	
5:19...	cybervolt_analysis.exe	4180	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Control\Sip\GP\DLL	REPARSE	Desired Access: R...
5:19...	cybervolt_analysis.exe	4180	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Control\Sip\GP\DLL	NAME NOT FOUND Desired Access: R...	
5:19...	cybervolt_analysis.exe	4180	RegOpenKey	HKEY\Software\WOW6432Node\Policies\Microsoft\Windows\Safe\CodeIdentifiers	REPARSE	Desired Access: Q...
5:19...	cybervolt_analysis.exe	4180	RegOpenKey	HKEY\Software\WOW6432Node\Policies\Microsoft\Windows\Safe\CodeIdentifiers	SUCCESS	Desired Access: Q...
5:19...	cybervolt_analysis.exe	4180	RegQueryValue	HKEY\Software\Policies\Microsoft\Windows\safe\codeidentifiers\TransparentEnabled	NAME NOT FOUND Length: 80	
5:19...	cybervolt_analysis.exe	4180	RegOpenKey	HKEY\Software\Policies\Microsoft\Windows\Safe\CodeIdentifiers	NAME NOT FOUND Desired Access: Q...	
5:19...	cybervolt_analysis.exe	4180	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Control\FileSystem	REPARSE	Desired Access: R...
5:19...	cybervolt_analysis.exe	4180	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Control\FileSystem	SUCCESS	Desired Access: R...
5:19...	cybervolt_analysis.exe	4180	RegQueryValue	HKEY\SYSTEM\CurrentControlSet\Control\FileSystem\DefaultAuthnLevel	TYPE: REG_DWORD	

Image of CyberVolk Ransomware Dynamic Analysis V

When the process operations are monitored in the dynamic analysis, it is observed that the console "**conhost.exe**" for GUI support is started depending on the main process. No additional potentially harmful process, network connection, persistence or any other methods/techniques were detected.

During the observation process, the "SafeBoot" key draws attention. CyberVolk ransomware is observed to be tampering with the safe mode settings of the windows device. It is also observed that it reads dec_key.dat in the **\$HOME\\AppData\\\\Roaming** directory. The file is not created because it does not write.



```

kernelbase.76927453
mov dword ptr ds:[ebx],103
mov eax,dword ptr ds:[ebp+8]
mov eax,dword ptr ds:[ebp+C]
mov eax,dword ptr ds:[ebp+C]
mov dword ptr ss:[ebp-24],eax
lea eax,dword ptr ss:[ebp-28]
push eax
push dword ptr ss:[ebp+10]
push dword ptr ss:[ebp+C] ; [ebp+C]:"Decrypting File : C:\\\\Users\\\\BilirMisin\\\\.ghidra\\\\10.3.3_PUBLIC\\\\o
push ebx
mov eax,dword ptr ds:[ebp+10]
and eax,103
movzx eax,al
neg eax
sbb eax, eax
not eax
and eax,ebx
push eax
push ebx
push dword ptr ds:[ebp+10]
push edi
call dword ptr ds:<_NtWriteFile>
mov ecx, eax
mov edx,c0000000 ; edx:"\\niu9golo9do81x9uonhugjkjnjljrongoxdzxoligxuukjzoukipn9jhgzgxohluzkizjduplicoh9rrrggx8jn1x
and ecx,edx ; edx:"\\niu9golo9do81x9uonhugjkjnjljrongoxdzxoligxuukjzoukipn9jhgzgxohluzkizjduplicoh9rrrggx8jn1xno8
cmp eax,103
jne kernelbase.76927453

```

Image of CyberVolk Ransomware Dynamic Analysis VI

During the decryption process, the situation of checking with the original key was examined in detail, but no such comparison was found. CyberVolk ransomware does not compare the provided decryption key with the original decryption key.

Instead, after acquiring the key from the dec_key.dat file, it uses the WriteFile API to create an empty file with the actual name of the .CyberVolk extension file. For example, for the file file.txt.CyberVolk, it writes an empty file named file.txt on the disk. Then, using the NtWriteFile API, it processes the decryption key and writes the decrypted content of the encrypted file into file.txt. However, during this process, the buffer memory is not checked. If the provided key is incorrect, instead of writing corrupted data into the file, it writes 0-byte data. But if the provided key is correct, since the generated data won't be corrupted, it writes the decrypted file content correctly.



Image of CyberVolk Ransomware Dynamic Analysis VII

CyberVolk Ransomware detects whether it is running in safe mode using [GetOSSafeBootMode](#) and the [SafeBoot](#) registry key. The [HideInSafeMode](#) function is used to hide or stop certain functions when safe mode is detected.



CyberVolk Ransomware Static Analysis

```

        24 48      MOVAPS   xmmword ptr [ESP + 0x50],XMM0
00421ca1 0f 29 44    MOVS    AL,byte ptr [ECX]
00421ca6 ff 15 bc    CALL    dword ptr [->USER32.DLL::GetDlgItemTextA]
31 42 00
00421cac 8d 4c 24 30 LEA     ECX,[ESP + 0x30]
00421cb0 8d 51 01    LEA     EDX,[ECX + 0x1]
00421cb3 8a 01    MOV     AL,byte ptr [ECX]
00421cb5 41    INC     ECX
00421cb6 84 c0    TEST    AL,AL
00421cb8 75 f9    JNZ    LAB_00421cb3
00421cb9 2b ca    SUB     ECX,EDX
00421cbc 83 f9 24  CMP     ECX,ECX
00421cbf 74 1b    JZ     LAB_00421cdc
00421cc1 6a 00    PUSH    0x0
00421cc3 6a 00    PUSH    0x0
00421cc5 68 1c 91  PUSH    s_Decryption_Key_is_not_correct!_004291c
42 00
00421cca 6a 00    PUSH    0x0
00421ccc ff 15 d0  CALL    dword ptr [->USER32.DLL::MessageBoxA]
31 42 00
00421cd2 33 c0    XOR     EAX,EAX
00421cd4 5f    POP     EDI
00421cd5 5e    POP     ESI
00421cd6 8b e5    MOV     ESP,EBP
00421cd8 5d    POP     EBP
152
L"Are you sure this is right decryption key? If not, you can loose all
files..." L"Start Decryption",0x24);
if (iVar2 == 6) {
    uStack_594 = 0;
    auStack_5b4 = ZEXT16(0);
    uStack_590 = 0;
    auStack_5a4 = auStack_5b4;
    GetDlgItemTextA(param_1,0x3e9,auStack_5b4,0x25);
    pcVar11 = auStack_5b4;
    do {
        cVar1 = *pcVar11;
        pcVar11 = pcVar11 + 1;
    } while ((cVar1 != '\0') && ((int)pcVar11 - (int)(auStack_5b4 + 1) != 0x24));
    if (((int)pcVar11 - (int)(auStack_5b4 + 1) != 0x24)) {
        MessageBoxA((HWND)0x0,"Decryption Key is not correct!",(LPCSTR)0x0,0);
        return 0;
    }
    FUN_00421f10(auStack_5b4);
    DAT_0042b918 = '0';
    SHGetFolderPath(0,0x1a,0,0,auStack_3bc);
    processParametersAndExecute(apppuStack_4d8,0x428f9c);
    pVar5 = _open((char *)apppuStack_4d8,"w");
    if (pVar5 != (FILE *)0x0) {
        FUN_0040bf6c(&pVarStack_5c8,1,0x24,pVar5);
        FUN_0040bb30(pVar5);
    }
    return 0;
}

```

Image of CyberVolk Ransomware Static Analysis II

The function, for the string "**Decryption Key is Not Correct**" was analyzed due to its potential relation to the encryption key. It was found that it does not check the actual encryption key. Instead, it calculates a **36-character value**. If the entered value is not exactly 36 characters, it shows the "**Decryption Key is Not Correct**" message and returns 0. However, if the string is 36 characters, it proceeds with the decryption process **without validating the actual encryption key.**

```

00421ac0 8a 01    MOV     AL,byte ptr [ECX]
00421ac2 41    INC     ECX
00421ac3 84 c0    TEST    AL,AL
00421ac5 75 f9    JNZ    LAB_00421ac0
00421ac7 56    PUSH    ESI
00421ac8 2b ca    SUB     ECX,EDX
00421aca 8d 84 24  LEA     EAX,[ESP + 0x24]
00421aca 04 00 00 00
00421ad1 51    PUSH    ECX
00421ad2 6a 01    PUSH    0x1
00421ad4 58    PUSH    EAX
00421ad6 e8 92 a4  CALL    FUN_0040bf6c
00421ad6 fe ff
00421ada 56    PUSH    ESI
00421ade e8 50 a0  CALL    FUN_0040bb30
00421ade fe ff
00421ae0 83 c4 14  ADD     ESP,0x14
00421ae3 33 c0    XOR     EAX,EAX
00421ae5 5f    POP     EDI
00421ae6 5e    POP     ESI
00421ae7 8b e5    MOV     ESP,EBP
00421ae9 5d    POP     EBP
00421aea c2 10 00  RET     0x10
157
LAB_00421aed
00421aed 8b 3d e0  MOV     EDI,dword ptr [->USER32.DLL::GetDlgItemTextA]
31 42 00
00421af3 6a 00    PUSH    0x0
158
uStack_590 = 0;
auStack_5a4 = auStack_5b4;
GetDlgItemTextA(param_1,0x3e9,auStack_5b4,0x25);
pcVar11 = auStack_5b4;
do {
    cVar1 = *pcVar11;
    pcVar11 = pcVar11 + 1;
} while ((cVar1 != '\0') && ((int)pcVar11 - (int)(auStack_5b4 + 1) != 0x24));
if (((int)pcVar11 - (int)(auStack_5b4 + 1) != 0x24)) {
    MessageBoxA((HWND)0x0,"Decryption Key is not correct!",(LPCSTR)0x0,0);
    return 0;
}
FUN_00421f10(auStack_5b4);
DAT_0042b918 = '0';
SHGetFolderPath(0,0x1a,0,0,auStack_3bc);
processParametersAndExecute(apppuStack_4d8,0x428f9c);
pVar5 = _open((char *)apppuStack_4d8,"w");
if (pVar5 != (FILE *)0x0) {
    FUN_0040bf6c(&pVarStack_5c8,1,0x24,pVar5);
    FUN_0040bb30(pVar5);
    return 0;
}
}
else if ((UVar9 == 0x3ec) {
    hMem = GlobalAlloc(2,0x23);
}

```

Image of CyberVolk Ransomware Static Analysis III

When it detects a 36-digit value, it is observed that it starts the decryption process. At the same time, a write operation is performed in the **_open** code structure. Here, the 36 byte of value received as input from the user is printed on **dec_key.dat**, which was displayed within the **dynamic analysis**.



```

    01 00 01 00
    004017b5 8b 40 fc  MOV    EAX,dword ptr [EAX + local_4]
    004017b8 6a 50  PUSH   0x50
    004017ba 89 85 90  MOV    dword ptr [EBP + local_274],EAX
    fd ff ff
    004017c0 8d 45 a8  LEA    EAX->local_5c,[EBP + -0x58]
    004017c3 6a 00  PUSH   0x0
    004017c5 50  PUSH   EAX
    004017c6 e8 05 0c  CALL   _memset
    .00 00
    004017cb 8b 45 04  MOV    EAX,dword ptr [EBP + local_res0]
    004017ce 83 c4 08  ADD    ESP,0xc
    004017d1 c7 45 a8  MOV    dword ptr [EBP + local_5c],0x40000005
    15 00 00 40
    004017d8 c7 45 ac  MOV    dword ptr [EBP + local_58],0x1
    01 00 00 00
    004017df 89 45 b4  MOV    dword ptr [EBP + local_50],EAX
    004017e2 ff 15 f4  CALL   dword ptr [->KERNEL32.DLL::IsDebuggerPresent]
    30 42 00
    004017e8 8b f0  MOV    ESI,EAX
    004017ea 8a 45 04  LEA    EAX=>local_5c,[EBP + -0x58]
    004017ed 89 45 f8  MOV    dword ptr [EBP + local_c],EAX
    004017f0 8d 85 dc  LEA    EAX=>local_328,[EBP + 0xfffffc0]
    fc ff ff
    004017f6 6a 00  PUSH   0x0
    004017f8 89 45 fc  MOV    dword ptr [EBP + local_8],EAX
    004017fb ff 15 fc  CALL   dword ptr [->KERNEL32.DLL::SetUnhandledExceptionFilter]
    30 42 00
  
```

Image of CyberVolk Ransomware Static Analysis IV

It is observed that CyberVolk Ransomware can detect debuggers with the **"IsDebuggerPresent"** API. If the debugger is detected, the function is terminated, but if the debugger is not detected, the program continues with the **resetGlobalVariable()** function.

```

    00401aa0 6a 0a  PUSH   0xa
    00401aa2 ff 15 04  CALL   dword ptr [->KERNEL32.DLL::IsProce
    31 42 00
    00401aa8 85 c0  TEST   EAX,EAX
    00401aaa 0f 84 ac  JZ    LAB_00401ic5c
    .01 00 00
    00401abb 83 65 f0 00  AND   dword ptr [EBP + local_14],0x0
    00401ab4 33 c0  XOR    EAX,EAX
    00401ab6 53  PUSH   EBX
    00401ab7 56  PUSH   ESI
    00401ab8 57  PUSH   EDI
    00401ab9 33 c9  XOR    ECX,ECX
    00401abb 8d 7d dc  LEA    EDI->local_28,[EBP + -0x24]
    00401abe 53  PUSH   EBX
    00401abf 0f a2  CPUID
    00401ac1 8b f3  MOV    ESI,EBX
    00401ac3 5b  POP    EBX
    00401ac4 90  NOP
    00401ac5 89 07  MOV    dword ptr [EDI->local_28],EAX
    00401ac7 89 77 04  MOV    dword ptr [EDI + local_24],ESI
    00401aca 89 4f 08  MOV    dword ptr [EDI + local_20],ECX
    00401acd 33 c9  XOR    ECX,ECX
    00401ac 89 57 0c  MOV    dword ptr [EDI + local_1c],EDX
    00401ad2 8b 45 dc  MOV    EAX,dword ptr [EBP + local_28]
    00401ad5 8b 7d e0  MOV    EDI,dword ptr [EBP + local_24]
    00401ad8 89 45 f4  MOV    dword ptr [EBP + local_10],EAX
    00401ad8 81 f7 47  XOR    EDI,0x756e6547
  
```

Image of CyberVolk Ransomware Static Analysis V

"IsProcessorFeaturePresent" API determines whether the specific processor feature is supported by the computing environment in which it is running.

It is also observed that the Ransomware accesses information related to the CPU. the **CPUID** instruction is utilized to distinguish between virtual and physical environments. **CPUID** queries the processor's attributes and checks virtualization indicators to determine if the environment is a virtual machine.



```

FLARE-VM 08/13/2024 01:13:47
PS C:\Users\... > Get-Process
Handles NPM(K) PM(K) WS(K) CPU(s) Id SI ProcessName
227 12 2816 22672 20,70 7044 1 CyberVolk_odz9rj5efm3yat2vb7w40cq16nx8hkplug

FLARE-VM 08/13/2024 01:13:49
PS C:\Users\... > Stop-Process -Name CyberVolk_odz9rj5efm3yat2vb7w40cq16nx8hkplug
FLARE-VM 08/13/2024 01:13:50
PS C:\Users\...

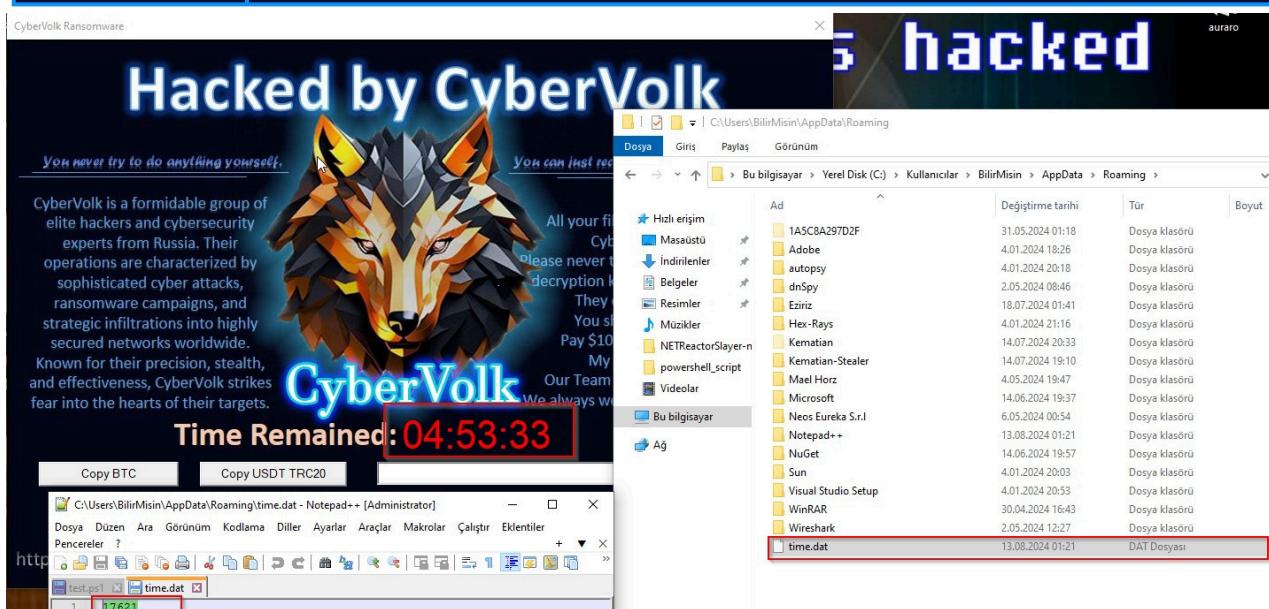
```

Image of CyberVolk Ransomware Vulnerabilities II

As soon as the GUI is launched and the necessary commands are given in PowerShell to terminate the process, the encryption process is interrupted.

Additionally, since it does not contain any persistence features within its structure, the Cybervolk ransomware does not reactivate or attempt to re-encrypt files if the device is restarted.

Command	Get-Process
Command	Stop-Process -Name <CyberVolk_Ransomare.exe>

*Image of CyberVolk Ransomware Vulnerabilities III*

Additionally, the CyberVolk ransomware operates by continuously counting down from 18,000 seconds, as written in the time.dat file. The timer can be manually adjusted by modifying the time.dat file, which allows the countdown to be extended indefinitely. This capability can facilitate the work of reverse engineering, forensic, and malware analysis teams by providing more time for analysis.



The screenshot shows a static analysis interface with two panes. The left pane displays assembly code from address 004216ed to 0042170e. The right pane shows corresponding C code. A red box highlights a section of the C code where it checks for drive types (UVar1) and creates threads for drives that can be auto-spreaded.

```

004216ed bf 61 00    MOV    EDI,0x61
004216f2 c7 45 ec    MOV    dword ptr [EBP + local_18],0x3a0063
004216f9 0f 57 c0    XORPS  XM60,XM60
004216fc c7 45 f8    MOV    dword ptr [EBP + local_14],0x5c
00421703 66 0f d6    MOVQ   qword ptr [EBP + local_10],XM60
00421708 45 f4      MOV    EBX,EDI
0042170a c7 45 fc    MOV    dword ptr [EBP + local_8],0x0
0042170e 00 00 00

00421711 8d 45 ec    LEA    EAX=>local_18,[EBP + _0x14]
00421714 66 89 7d ec    MOV    word ptr [EBP + local_18],DI
00421718 50          PUSH   EAX
00421719 ff 15 dc    CALL   dword ptr [->KERNEL32.DLL:GetDriveType]
0042171f 83 e0 02    SUB    EAX,0x2
00421722 74 12      JZ    LAB_00421736
00421724 83 e8 01    SUB    EAX,0x1
00421727 74 0d      JZ    LAB_00421736
00421729 83 e8 01    SUB    EAX,0x1
0042172c 74 08      JZ    LAB_00421736
0042172e 8b 0d 1c    MOV    ECX,dword ptr [DAT_0042f81c]
0042172f f8 42 00

LAB_00421711:
00421711 8d 45 ec    LEA    EAX=>local_18,[EBP + _0x14]
00421714 66 89 7d ec    MOV    word ptr [EBP + local_18],DI
00421718 50          PUSH   EAX
00421719 ff 15 dc    CALL   dword ptr [->KERNEL32.DLL:GetDriveType]

0042171f 83 e0 02    SUB    EAX,0x2
00421722 74 12      JZ    LAB_00421736
00421724 83 e8 01    SUB    EAX,0x1
00421727 74 0d      JZ    LAB_00421736
00421729 83 e8 01    SUB    EAX,0x1
0042172c 74 08      JZ    LAB_00421736
0042172e 8b 0d 1c    MOV    ECX,dword ptr [DAT_0042f81c]
0042172f f8 42 00

132     FUN_0040bb30(_File);
133     )
134     UVar4 = 0x61;
135     local_18 = 0x3a0063;
136     local_14 = 0x5c;
137     local_10 = 0;
138     UVar3 = 0x61;
139     local_8 = 0;
140     do {
141         local_18 = CONCAT22(local_18, 2, 2, UVar4);
142         UVar1 = GetDriveTypeW((LPCWSTR)&local_18);
143         if (((UVar1 == 2) || (UVar1 == 3)) || (UVar1 == 4)) {
144             lpParameter = (LPWSTR)FUN_004010f4(4);
145             wsprintfW(lpParameter,L"%c%c",UVar3,0x65);
146             ppvVar2 = (HANDLE *)lpParameter;
147             CreateThread((LPSECURITY_ATTRIBUTES)0x0,0,threadRoutine,lpParameter,0,(LPDWORD)0x0);
148             (&lpHandles_00430c30)[DAT_0042f81c] = ppvVar2;
149             DAT_0042f81c = DAT_0042f81c + 1;
150             DAT_00430c20 = DAT_00430c20 + 1;
151             }
152             UVar4 = UVar4 + 1;
153             iVar3 = iVar3 + 1;
154         } while ((UVar4 < 0x7b));
155         WaitForMultipleObjects(DAT_0042f81c,&lpHandles_00430c30,1,0xffffffff);
156         return;
157     }
158

```

Image of CyberVolk Ransomware Static Analysis VI

CyberVolk Ransomware has been found to include activity similar to a worm virus. It scans all drive letters between "a" and "z". If these drives are of the type where it can spread itself (removable, hard, network), it creates a multi thread to execute on these drives. This structure has an auto spread feature like a worm.

The screenshot shows assembly code for searching for a Task Manager window and sending a WM_CLOSE message to it. The code uses FindWindowA to find the window and PostMessageW to send the message. A red box highlights the call to FindWindowA.

```

sub_422500 proc near
    lpThreadParameter=dword ptr 4
    push ebx
    mov ebx, ds:FindWindowA
    push esi
    mov esi, ds:Sleep
    push edi
    mov edi, ds:PostMessageW

loc_422505: ; lpWindowName
    push 0
    push offset ClassName,"TaskManagerWindow"
    call ebx ; FindWindowA
    test eax, eax
    jz short loc_4225E8

    push 0 ; lParam
    push 0 ; wParam
    push 10h ; Msg
    push eax ; hWnd
    call edi ; PostMessageW

loc_4225E8: ; dwMilliseconds
    push 30h
    call esi ; Sleep
    fend short loc_422505

```

Image of CyberVolk Ransomware Static Analysis VII

CyberVolk ransomware continuously searches for the window named "**TaskManagerWindow**" via the "**FindWindowA**" API by waiting for 1 second in an infinite loop running as a different thread. When it finds it, it sends **0x0010 (WM_CLOSE)** via the **PostMessageW** API to close the window. This prevents the user from terminating the cybervolk ransomware process via the task manager.



```

00421e2e 8b e5    MOV    ESP,EBP
00421e30 5d        POP    EBP
00421e31 c2 10 00  RET    0x10

LAB_00421e34      CMP    EAX,0x113
00421e34 3d 13 01  JNZ    LAB_00421e63
00421e39 75 28    CMP    dword ptr [EBP + param_3],0xd80
00421e3b 81 7d 10  JNZ    LAB_00421e63
00421e44 80 0d 00 00 CMP    byte ptr [DAT_00442b918],0x0
00421e42 75 1f    JNZ    LAB_00421e63
00421e44 80 3d 18  CMP    byte ptr [DAT_00430c20],0x0
00421e42 b9 42 00 00 JNZ    LAB_00421e63
00421e4b 75 09    JNZ    LAB_00421e56
00421e4d 83 3d 20  CMP    dword ptr [DAT_00430c20],0x0
00421e54 8c 43 00 00 JZ     LAB_00421e6d

LAB_00421e56      PUSH   0x0
00421e56 6a 00    PUSH   0x0
00421e5a ff 75 08  PUSH   dword ptr [EBP + param_1]
00421e5d ff 15 fc  CALL   dword ptr [->USER32.DLL::InvalidateRect]
31 42 00

LAB_00421e63      XREF[1]: 00421e4b(j)
                        00421a92(j), 00421c79(j),
                        00421d34(j), 00421dc4(j)

XREF[8]: 00421831(j), 00421a92(j),
          00421c5a(j), 00421c79(j),
          00421d34(j), 00421dc4(j)

262 }
263 else if (param_2 == 0x110) {
264     iVar2 = GetSystemMetrics(0);
265     iVar3 = GetSystemMetrics(1);
266     SetWindowPos(param_1,(HWND)0x0,iVar2 / 2 + -0x1e0,iVar3 / 2 + -0x14a,0x3b5,0x294,0x14);
267     SetTimer(param_1,0x68,1000,(TIMERPROC)0x0);
268     return 0;
269 }
270 }
271 else if ((param_2 == 0x113) && (param_3 == 0xd80)) {
272     if ((DAT_0042b018 == '0') && (DAT_00430c20 == 0)) {
273         KillTimer(param_1,0x0d0);
274         SHGetFolderPathA(0x1a,0,0,aStack,30c);
275         processParametersAndExecute((char *)appuStack_4d8,0x428f9c);
276         FID_cfile1__mdir((char *)appuStack_4d8);
277         processParametersAndExecute(appuStack_4d8,0x428f90);
278         FID_conflict_mdir((char *)appuStack_4d8);
279         MessageBoxW((HWND)0x0,(LPCWSTR)&ptext_00428fd8,L'Decrypt Completed!',0);
280         /* WARNING: Subroutine does not return */
281         _exit(1);
282     }
283     InvalidateRect(param_1,(RECT *)0x0,0);
284 }
285 return 0;
286 }

MessageboxW((HWND)0x0,(LPCWSTR)&ptext_00428fd8,L'Decrypt Completed!',0);
/* WARNING: Subroutine does not return */

_exit(1);
InvalidateRect(param_1,(RECT *)0x0,0);
return 0;

```

Image of CyberVolk Ransomware Static Analysis VIII

When the decryption process is completed, the program terminates itself using the `_exit(1);` function. However, since it does not involve any persistence, writing itself to a process, or utilizing any other technique/method, it does nothing else in the self-cleaning stage other than terminating itself.

CyberVolk Ransomware Vulnerabilities

ThreatMon Malware Team has identified several vulnerabilities in the CyberVolk ransomware that have a critical impact on its infection process.



Image of CyberVolk Ransomware Vulnerabilities I

Unlike most ransomware, CyberVolk ransomware first launches the GUI and then starts encrypting the system with multithreads. In this time, it was found that the task manager was blocked to prevent the process from being interrupted, but powershell was not blocked.



MITIGATION

- ◆ Ensure that data is backed up regularly, and keep multiple copies, including one offline or in a cloud service.
- ◆ Educate employees on recognizing phishing emails, suspicious links, and social engineering tactics.
- ◆ Keep all systems, software, and firmware up-to-date with the latest security patches.
- ◆ Deploy and regularly update security software across all endpoints.
- ◆ Use CTI to set up early warning alerts for ransomware campaigns that are targeting your industry or region. These alerts can help your organization prepare for potential attacks before they reach you.
- ◆ Use advanced spam filtering to reduce the risk of phishing emails reaching end users.
- ◆ Enforce the principle of least privilege (PoLP) to limit user access to only what is necessary for their role.
- ◆ Subscribe to threat intelligence feeds that provide information on emerging ransomware threats.
- ◆ Implement application whitelisting to allow only approved programs to run on your systems, preventing unauthorized or malicious software from executing.

Categorization

APT Group	It is not an APT group, but it has affiliations with APT 44
Threat Category	Ransomware
Malware Family	GandCrab Ransomware

Mitre Att&ck Table

Tactics	Technique ID	Technique Name
Initial Access	T1566	Phishing
Execution	T1106 T1204.002	Native API User Execution: Malicious File
Defense Evasion	T1562.001 T1562.009	Impair Defenses: Disable or Modify Tools Impair Defenses: Safe Mode Boot
Discovery	T1010 T1622 T1083 T1012 T1124 T1497	Application Window Discovery Debugger Evasion File and Directory Discovery Query Registry System Time Discovery Virtualization / Sandbox Evasion
Impact	T1486 T1485 T1565	Data Encrypted For Impact Data Destruction Data Manipulation

Yara Rule

Download the Yara Rule From ThreatMon [Github Page](#).

```
rule CyberVolk_Ransomware_Yara{
    meta:
        description = "Yara rule for detecting CyberVolk Ransomware"
        author = "Aziz Kaplan"
        email = "aziz.kaplan@threatmonit.io"
        file_hash = "d08243e976e01baa5479a134577a1407daf4bec89a5f47bf2b803c0919917f5b"

    strings:
        $OP1 = {8d 84 24 b0 04 00 00 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? 6a 00}
        $OP2 = {8b 7d 08 6a 24 68 5c 90 42 00}
            //8b7d086a24                                |MOVEDI,dwordptr[EBP+arg]
            //685c904200                                |PUSH<start_of_encryption>
        $OP3 = {6a 24 68 5c 90 42 00}
            //6a24                                         |PUSH0x24
            //685c904200                                |PUSH<start_of_decryption>
        $OP4 = {8d 51 01 ?? ?? ?? ?? ?? ?? ?? ?? ?? 2b ca 83 f9 24 74 1b}
            //Check of "if" condition of decryption process
            //8d5101                                     |LEAEDX,[ECX+0x1]
            //2bca                                         |SUBECX,EDX
            //83f924741b                                |CMPECX,0x24
        $OP5 = {ff 15 d0 31 42 00}
            //Call of API after the if condition
            //ff15d0314200                               |dwordptr[->USER32.DLL::MessageBoxA]
        $OP6 = {8d 4c 24 30 e8 2b 02 00}
            //Character replacment after the decryption key is provided
            //8d4c2430                                     |LEAECX,[ESP+0x30]
            //e82b0200                                     |character_replacement
        $OP7 = {8d 84 24 20 01 00 00 ?? ?? ?? ?? ?? ?? ?? e8 c7 9c fe ff}
        $OP8 = {8d 44 24 38 ?? ?? ?? e8 23 a2 fe ff}
            //File Creation dec_key.dat
            //8d842420010000                             |LEAEAX,[ESP+0x120]
            //e8c79cff                                 |CALL_fopen
            //8d442438                                     |LEAEAX,[ESP+0x38]
            //e823a2feff                                |file_operation
        $OP9 = {68 80 0d 00 00 ff 75 08 ff 15 e8 31 42 00}
            //Timer Killer
            //68000d0000                                |PUSH0xd80
            //ff7508                                     |PUSHdwordptr[EBP+param_1]
            //ff15e8314200                               |CALLdwordptr[->USER32.DLL::KillTimer]
        $OP10 = {83 f8 0f ?? ?? 3d 10 01 00 00}
            //Conditions for decryption process
            //83f80f7468                                |CMPEAX,0xf
            //3D10010000                                |CMPEAX,0x110
        $OP11 = {84 c0 74 10 ff 75 08 ff 15 08 31 42 00 ?? ff 15 0c 31 42 00}
            //Terminating itself if a condition is met
        $OP12 = { 54 61 73 6b 4d 61 6e 61 67 65 72 57 69 6e 64 6f 77 00 00 00 }
            //TaskManagerWindow
        $OP13 = { 25 73 5c 74 69 6d 65 2e 64 61 74 00 }
            //time.dat
        $OP14 = { 25 73 5c 64 65 63 5f 6b 65 79 2e 64 61 74 00 }
            //dec_key.dat

    condition:
        uint32(uint32(0x3C)) == 0x00004550 or
            (filesize > 4 and uint32(0) == 0x464C457F) or
            (uint32(0) == 0xCEFAEDFE or uint32(0) == 0xFFAEDFE) and
            (11 of ($OP*))
}
```



IOC List

Sha256

```
de0b74917fe24c2b38e2d1172b7352f88bf8b3df64b6d44ca5f317db85aeb
324
70257c48ed8e1a3b57a7d6a5bed17837f60d630bdda0b22b048a3721569f
e038
7d294c60c44b8b776c45e46e904a2de70ff4820e7e7863adb9f191c6554f
9fb5
74b5a0ed14c7b8e26d51d4b9242e73686bad2e63cd11d9cbdb52e08fa341
58c1
```

Sigma Rules

Download the Sigma Rules From ThreatMon Github Page.

```
title: Suspicious File Creation Detected
id: 8a5a94e2-5a2e-4b1a-bb97-03c7d5cf9a93
status: experimental description: |

Checks for BMP and DAT file creation within specific directories.

author: Aziz Kaplan <aziz.kaplan@threatmonit.io>
logsource:
    category: file_access
    product: windows
detection:
    selection:
        fileName|contains:
            - '\AppData\Local\
            - '\AppData\Roaming\
            - '\AppData\Local\Temp\
        fileName|endswith:
            - '.bmp'
            - '.dat'
    filter_system_folders:
        Image|startswith:
            - 'C:\Program Files\
            - 'C:\Windows\' - 'C:\Program
                Files (x86)\
            - 'C:\Windows\system32\
            - 'C:\Windows\SysWOW64\
    condition: selection and not 1 of filter_system_folders
falsepositives:
    - Legitimate software installed that creates BMP file in Temp directory
level: medium
```



```
title: .CyberVolk Extension Detected
id: 37b2c73a-f147-4d93-842e-0b853b55de49
status: stable
description: Detects changes in file extensions where files are renamed to use
the .CyberVolk extension, typical in ransomware activity.
author: Aziz Kaplan <aziz.kaplan@threatmonit.io>
logsource:
    category: file_event
    product: windows
detection:
    selection:
        TargetFilename|endswith: '.CyberVolk'
    condition: selection
falsepositives:
    - Unlikely
level: critical
```

```
title: CyberVolk Ransomware ImpHash Detected
id: e45cf64a-8af9-4e69-9b55-278f44f2b1d1
status: test
description: Detects CyberVolk Ransomware from import hash (imphash)
author: Aziz Kaplan <aziz.kaplan@threatmonit.io>
logsource:
    category: process_creation
    product: windows
detection:
    selection:
        - Imphash:
            - 0982e392aba6a868dc7bda8b61e977ab # CyberVolk
        - Hashes|contains:
            - IMPHASH=0982e392aba6a868dc7bda8b61e977ab
    condition: selection
falsepositives:
    - Legitimate use
level: high
```

