

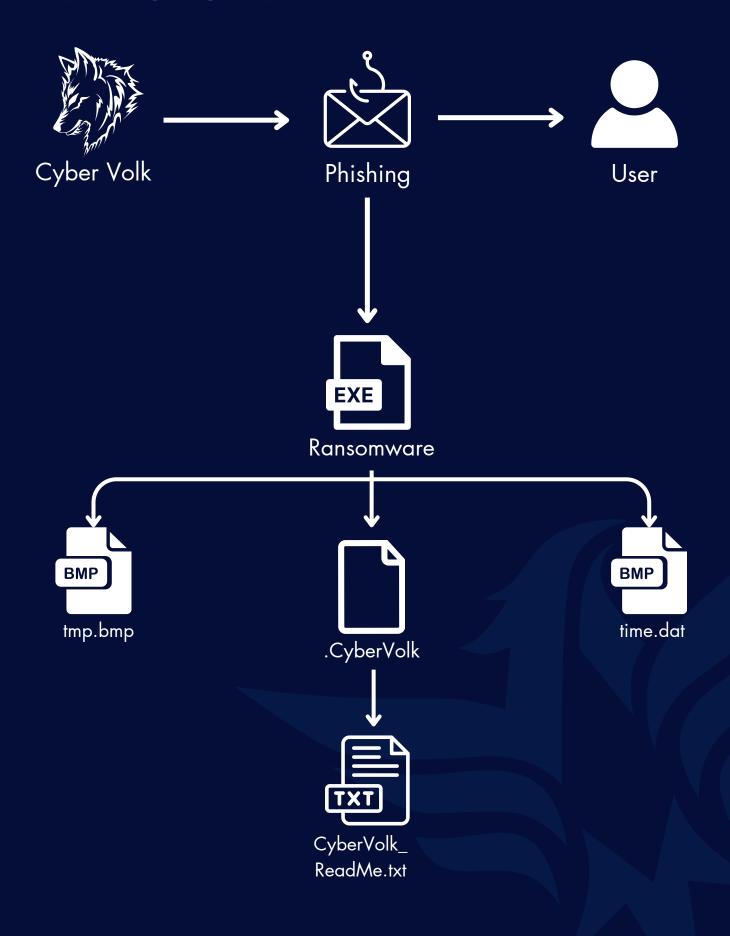


TABLE OF CONTENTS

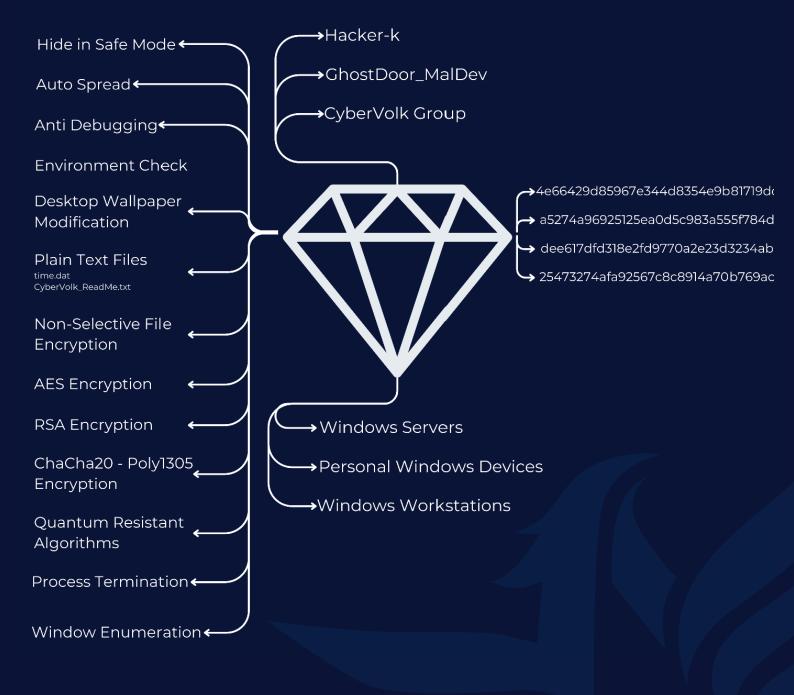
| Attack Chain | | |
|---|----|--|
| Diamond Model | | |
| Executive Summary & Key Findings | | |
| About CyberVolk Group | | |
| About Cybervolk Ransomware | 8 | |
| What Sets CyberVolk Ransomware Apart from the Others? | 10 | |
| CyberVolk Ransomware Contributors | 11 | |
| A Quick Look into the CyberVolk Ransomware | 12 | |
| Technical Malware Analysis | 13 | |
| Basic Characteristics of CyberVolk Ransomware | 13 | |
| Dynamic Analysis of CyberVolk Ransomware | 15 | |
| CyberVolk Ransomware Static Analysis | 18 | |
| CyberVolk Ransomware Vulnerabilities | 21 | |
| Mitigations | 23 | |
| Categorizations | 24 | |
| Mitre Att&ck Table | 24 | |
| Yara Rule | 25 | |
| IOC List | 26 | |
| Sigma Rules | 26 | |



ATTACK CHAIN



DIAMOND MODEL



Executive Summary & Key Findings

As ThreatMon, we strive to prevent potential malicious activities by informing individuals, companies, firms, institutions, and organizations about current threats through our reports, posts, and analyses.

CyberVolk Group is a threat actor group originating from India and is one of the members of the Holy League organization, established by APT 44 and other Russian/Russian-aligned hackers to carry out attacks against NATO, Ukraine, and states opposing Russia. Such formations pose a global threat.

CyberVolk Ransomware was developed by the CyberVolk Financially Motivated Threat Actor Group and released for sale as Ransomware-as-a-Service (RaaS) on July 1, 2024. After the initial version of the ransomware was leaked on VirusTotal, the CyberVolk group developed a new version and continued their RaaS services with this new version on July 10, 2024.

Operates in an offline structure, encrypts files with the .CyberVolk extension and demands a payment of \$1,000 for the decryption key.

The ransomware employs ChaCha20-Poly1305, AES, RSA, and quantum-resistant algorithms for encryption, making it highly secure. If an incorrect decryption key is provided, instead of indicating that the key is wrong, it initiates the decryption process, and at the end, it writes 0-byte data into the encrypted files, leading to severe data loss.

CyberVolk ransomware has been found to block TaskManager in order to prevent the encryption process from terminating. By opening the task manager, the user cannot terminate the running ransomware through the task manager. However, as ThreatMon Malware team, we have identified critical vulnerabilities in CyberVolk ransomware that affect the encryption process and summarized them in detail.

The ransomware developed by the CyberVolk group is a current threat to all windows users (individuals, companies, institutions, organizations, etc.). Especially according to the intelligence information collected, the +20.000\$ that the cybervolk group admin claims to have earned through this ransomware demonstrates the seriousness of this threat level.

You can find more information and a technical analysis of the CyberVolk ransomware in the continuation of the report.



About Cybervolk Group

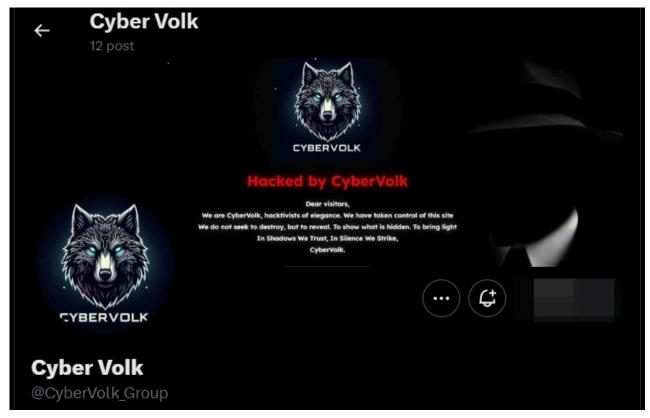


Image of Cybervolk Group Twitter Account

Cybervolk Hacker Group is an Indian cyber crime organization that was founded on March 28 2024 under the name GLORIAMIST India and later changed to Cybervolk.

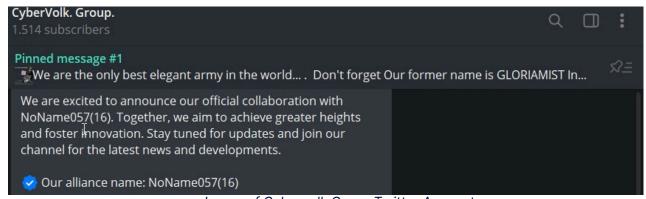


Image of Cybervolk Group Twitter Account

It was first identified by ThreatMon after their partnership with **Noname057(16).**

Russian-based hacker groups (**Noname057(16**) and the **cyber arm of russia**) have been attracting newly founded cybercrime organizations that can do successful work, and Cybervolk is one of these groups.



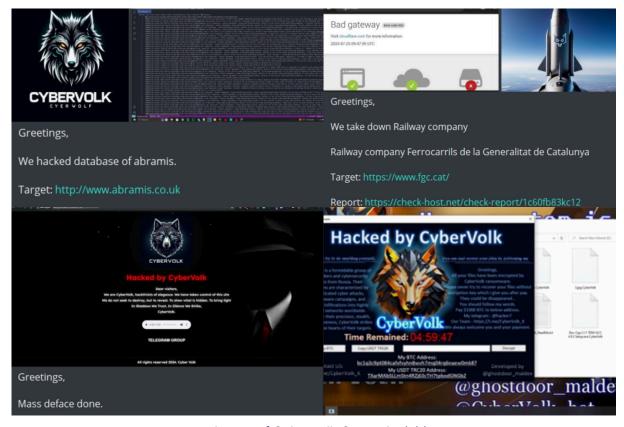


Image of Cybervolk Group Activities

According to the intelligence obtained by ThreatMon, the Cybervolk group has so far been involved in DDoS attacks, Website Defacement attacks, Data Leak attacks, Network Breach attacks and Ransomware attacks.

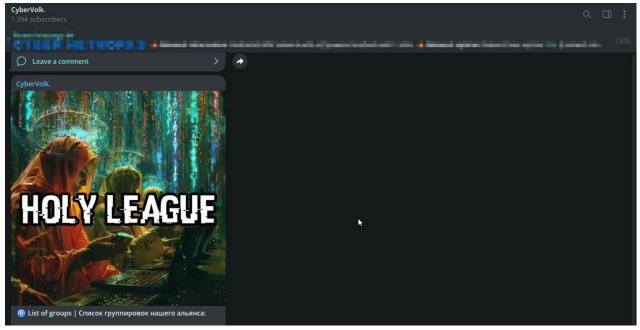


Image of Holy League Organization

At the same time, the Cybervolk group has been identified as one of the 45 hacker groups of the **Holy League** organization, which was recently created by Russian threat actors to attack **NATO**, **Ukraine** and **Israel**.



About Cybervolk Ransomware

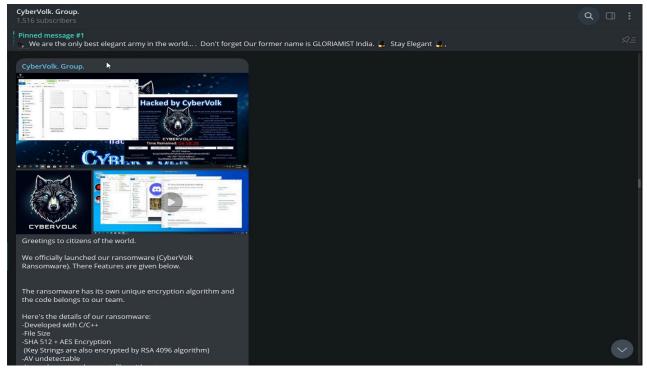


Image of CyberVolk Group Telegram Post

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.

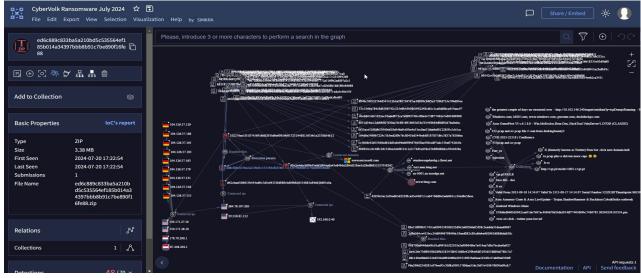


Image of CyberVolk Ransomware VirusTotal Leak

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.



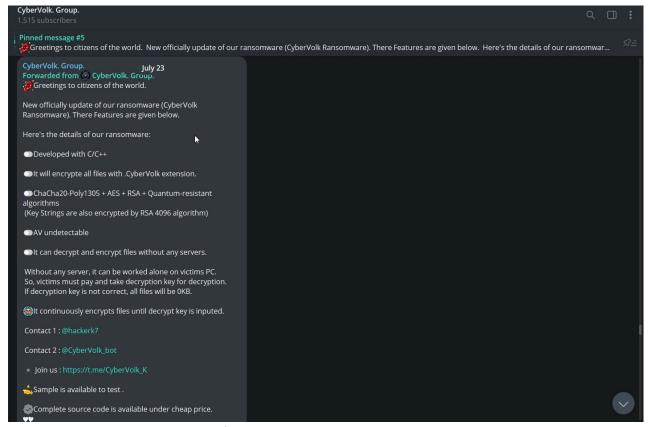


Image of CyberVolk Ransomware Telegram Post

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

The .cvenc extension has been replaced by the .CyberVolk extension.

The AES encryption algorithm has been replaced by ChaCha20-Poly1305 + AES + RSA + Quantum resistant algorithms.

It is claimed to be FUD (Fully UnDetectable).

It can encrypt and decrypt without the need for a C2 (Command and Control) server (offline ransomware).

If the wrong key is entered, the contents of the encrypted files are deleted, and if there is no backup of the data, it is lost forever.



What Sets CyberVolk Ransomware Apart from the Others?

In general, PQC/Quantum-resistant algorithms are not commonly used by ransomware. These algorithms are employed to be secure against cryptanalytic attacks by quantum computers. This is the first time a quantum-resistant algorithm has been observed being used within ransomware.

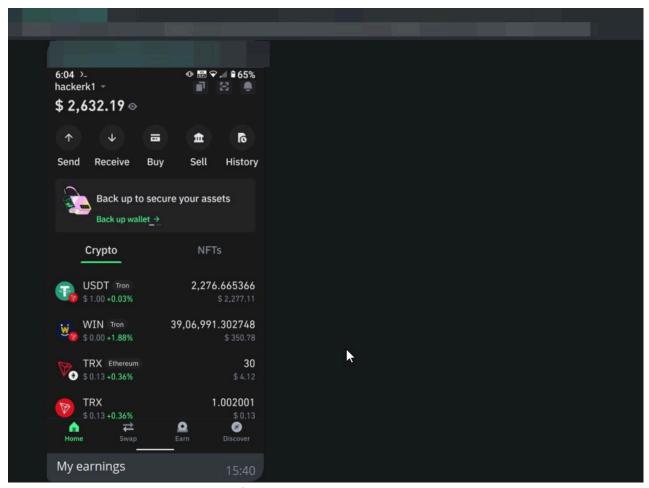


Image of CyberVolk Admin Leaked Telegram Chat

According to the intelligence obtained, it has been determined that the Cybervolk admin made a profit of \$2632 from the ransomware in the past.

However, it is now claimed that this profit has exceeded over \$20,000. This situation highlights the high-level threat posed by CyberVolk ransomware in the black-market(Screenshot not shared knowingly).



CyberVolk Ransomware Contributors

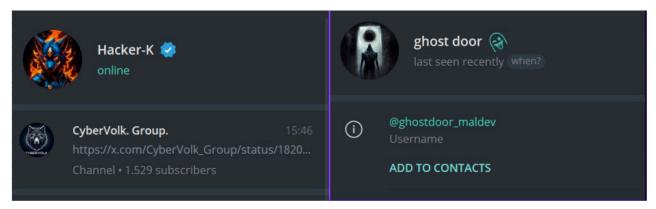


Image of CyberVolk Ransomware Contributors

The threat actor known by the alias Hacker-K is known to be of Indian origin and is the leader of the CyberVolk group.

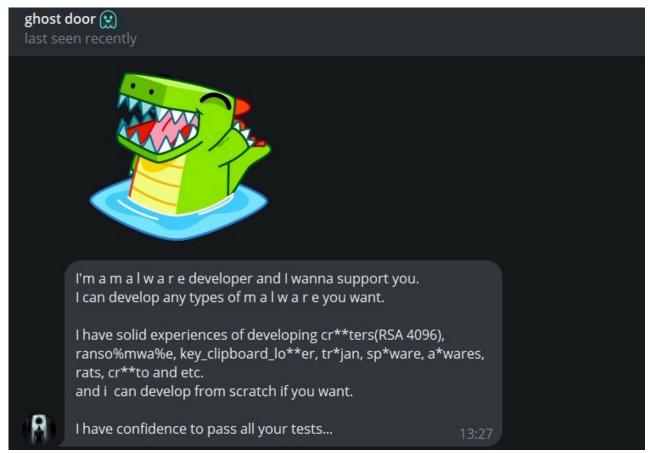


Image of Ghost Door Leaked Telegram Chat

The threat actor known by the alias **ghostdoor_maldev** is of china origin and is not directly associated with any group. This actor finds threat actor groups and makes requests for malware development to them. It has been identified as a threat actor in the expert class, particularly in the areas of cryptography and 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.

Tou Should Tollow 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

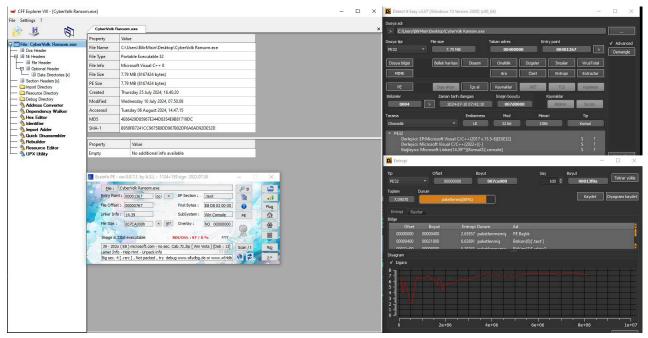


Image of CyberVolk Ransomware Characteristics

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

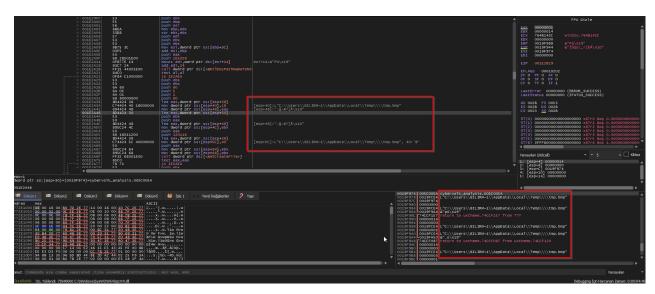


Image of CyberVolk Ransomware Dynamic Analysis I

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.



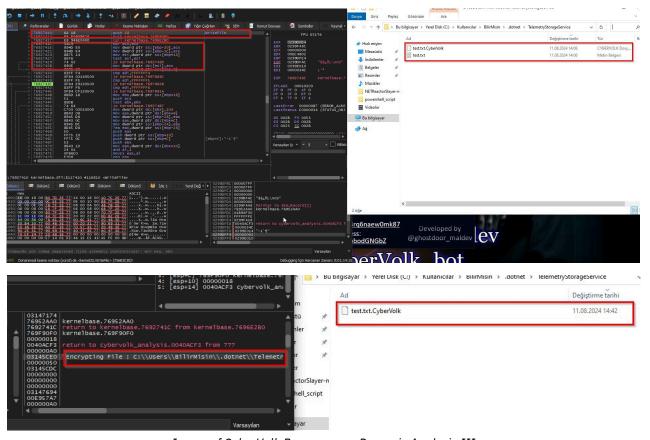


Image of CyberVolk Ransomware Dynamic Analysis III

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: 0x00000890
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. II Sychervok, analysis eve 4180 of Process Start 5.19. II Sychervok, analysis eve 4180 of PepChenkey 5.19. II Sychervok, analysis eve 6180 of PepChenkey 5.19. II Sycher
```

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:[ebx+s]
mov dword ptr ds:[ebx+s]
mov dword ptr ss:[ebp-28],eax
mov eax,dword ptr ss:[ebp-24],eax
loss dword ptr ss:[ebp-24],eax
push eaver dword ptr ss:[ebp-24],eax
push dword ptr ss:[ebp+10]
push dword ptr ss:[ebp+10]
push dword ptr ss:[ebp+10]
push dword ptr ss:[ebp+10]
push dword ptr ds:[ebx+10]
and al,1
movex eax,al
neg eax
sto eax, eax
not eax, eax
push eex
push dword ptr ds:[ebx+10]
and eax,ebx
push eex
push dec
push dword ptr ds:[ebx+10]
push edi

call dword ptr ds:[ebx+10]
push edi
call dword ptr ds:[ebx+10]
push edi
call dword ptr ds:[ebx+10]
push edi
eax, eax
push eex
push ex
pu
```

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.

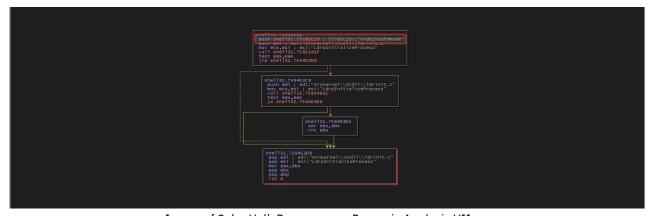


Image of CyberVolk Ransomware Dynamic Analysis VII

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

```
L*Are you sure this is right decription key? If not, you can loose all files...*
00421ca1 0f 29 44
                                      MOVAPS
                                                        xmmword ptr [ESP + 0x50],XMM0
                                                                                                                                153
154
155
156
157
24 50
00421ca6 ff 15 bc
                                                                                                                                                     if (iVar2 == 6) {
    uStack_594 = 0;
    auStack_5b4 = ZEXT816(0);
    uStack_590 = 0;
                                      CALL
                                                        dword ptr [->USER32.DLL::GetDlqItemTextA]
00421cb3 8a 01
00421cb5 41
00421cb6 84 c0
00421cb8 75 f9
00421cba 2b ca
00421cbc 83 f9 24
30421cbf 74 1b
                               LAB_00421cb3
                                                                                                                                                          GetDlgItemTextA(param_1,0x3e9,auStack_5b4,0x25);
                                                         AL, byte ptr [ECX]
                                                                                                                                                                                                                     1) != 0x24) {
| correct! (LPCSTR)0x0,0);
00421cc1 6a 00
00421cc3 6a 00
00421cc3 6a 1c 91
                                      PUSH
                                                        s Decryption Key is not correct! 00429110
                                                                                                                                                         FUN_00421f10(auStack_5b4);
                                                                                                                                                         DAT_0042b918 = '\0';
SHGetFolderPathA(0,0x1a,0,0,auStack_3bc);
processParametersAndExecute(apppuStack_4d
00421cca 6a 00
00421ccc ff 15 d0
                                                        dword ptr [->USER32.DLL::MessageBoxA]
                                                                                                                                                         processParametersAndExecute(apppuStack_4d8,6
pFVar5 = _fopen((char *)apppuStack_4d8,"w");
if (pFVar5 != (FILE *)0x0) {
                                      CALL
                                                                                                                                                                                                                          8.0x428f9c):
31 42 00
00421cd2 33 c0
00421cd4 5f
00421cd5 5e
00421cd6 8b e5
                                                        EAX, EAX
                                                                                                                                                             FUN_0040bf6c(&pvStack_5c8,1,0x24,pFVar5);
                                                                                                                                                            FUN_0040bb30(pFVar5);
                                                                                                                                C<sub>f</sub> Decompile: HandleMessa... × □ Defined Strings ×
```

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 <u>without validating the actual encryption key</u>.

```
00421ac0 8a 01
00421ac2 41
00421ac3 84 c0
00421ac3 85 c0
00421ac5 75 f9
00421ac8 2b ca
00421aca 8d 84 24
a4 00 00
00421ad1 51
00421ad2 6a 01
00421ad4 50
                                                          TEST
                                                                                                                                                                                                                        GetDlgItemTextA(param_1,0x3e9,auStack_5b4,0x25);
pcVar11 = auStack_5b4;
                                                                                  AL,AL
LAB_00421ac0
                                                                                                                                                                                                                                                                                                                                      ect!".(LPCSTR)0x0.0):
   00421ad4 50
00421ad5 e8 92 a4
fe ff
00421ad5 56
00421adb e8 50 a0
fe ff
00421ae0 83 c4 14
00421ae3 33 c0
00421ae5 5f
00421ae6 5e
00421ae7 8b e5
00421ae9 5d
00421ae9 5d
00421ae2 c2 10 00
                                                                                 FUN_0040bf6c
                                                                                                                                                                                                                         FUN_00421f10(auStack_5b4);
                                                         PUSH
CALL
                                                                                                                                                                                                                        DAT_0042b918 = '\0';
SHGetFolderPathA(0,0x1a,0,0,auStack_3bc);
                                                                                FUN_0040bb30
                                                                                                                                                                                                                       SNeetholeerMath(0,0k1a,0,0,0mstack,30c);

processParametersAndExecute(apppuStack,4d8,0x428f9c);

pFVat5 = fopen((char ')apppuStack_4d8,"w");

if (pFVat5 | (FLE ')800,0

(FLU 0,0000ffc(&pvStack_5c8,1,0x24,pFVat5);

FUL 0,0000b30(pFVat5);

return 0;
                                                LAB_00421aed
                                                                                EDI, dword ptr [->USER32.DLL::GetDlgItem]
                                                                                                                                                                                                                         = GlobalAlloc(2,0x23)
00421af3 6a 00
```

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 **_fopen** 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 0
004017b5 8b 4b fc
004017b8 6a 50
004017ba 89 85 90
fd ff ff
004017c0 8d 45 a8
004017c3 6a 00
                                                                                                                                                                          EAX, dword ptr [EAX + local_4]
                                                                                                                                                                          dword ptr [EBP + local_274], EAX
                                                                                                                                                                       EAX=>local_5c,[EBP + -0x58]
                                                                                                                                                                                                                                                                                                                                                                                                                                                   (*pcVari)();
) resetClobalVariable();
| memset(local_328,0,0x2cc);
| local_328[0] = 0x10001;
| memset(alocal_5c,0,0x50);
| local_5c_ExceptionCode = 0x40000015;
| local_5c_ExceptionFlags = 1;
| pvar2 = 150buggerPresent();
| local_c_ExceptionRecord = 210cal_5c;
| local_c_ExceptionRecord = 210
 004017c6 e8 05 0c
00 00
004017cb 8b 45 04
004017ce 83 c4 0c
004017d1 c7 45 a8
15 00 00 40
004017d8 c7 45 ac
                                                                                                                                                                     EAX, dword ptr [EBP + local_res0]
                                                                                                                     MOV dword ptr [EBP + local_5c],0x40000015
                                                                                                                                                   dword ptr [EBP + local_58],0x1
                                                                                                                     MOV
   01 00 00 00
004017df 89 45 b4
004017e2 ff 15 f4
                                                                                                                   MOV dword ptr [EBP + local 50], EAX

CALL dword ptr [->KERNEL32.DLL::ISDebuggerPresent]
 004017e2 Ff 15 F4
30 42 00
004017e8 8b f0
004017e8 8d 45 a8
004017e4 89 45 F8
004017f6 6a 00
004017f6 6a 00
004017f6 6f 15 Fc
004017f6 ff 15 fc
30 42 00
                                                                                                                                                                                                                                                                                                                                                                                                                                                             local_c.ContextRecord = (PCONTEXT)local_328;
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           EXCEPTION FILTER(0x0);
                                                                                                                                                                     ESI,EAX

EAX==local_5c,[EBP + -0x58]

dword ptr [EBP + local_c],EAX

EAX=>local_328,[EBP + 0xfffffcdc]
                                                                                                                                                                                                                                                                                                                                                                                                                                                                          ar3 = UnhandledExceptionFilter(&local_c);
((LVar3 == 0) && (BVar2 != 1)) {
                                                                                                                                                                                                                                                                                                                                                                                                                                                        if ((LVar3 == 0) && (BVar:
resetGlobalVariable();
                                                                                                                                                                     0x0
dword ptr [EBP + local_8],EAX
dword ptr [->KERNEL32.DLL::SetUnhandledExcepti...
                                                                                                                                                                                                                                                                                                                                                                                                                                        Gr Decompile: HandleProce... × Oth Defined Strings ×
```

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.

```
00/01220 62 02
00401aa2 ff 15 04
                                                      dword ptr [->KERNEL32.DLL::IsProce
                                                                                                                        DAT_0042c9dc = 0;

DAT_0042b010 = DAT_0042b010 | 1;

BVar4 = IsProcessorFeaturePresent(10);

uVar5 = DAT_0042b010;
                                     CALL
             31 42 00
00401aa8 85 c0
00401aaa 0f 84 ac
                                     TEST
                                                     LAB_00401c5c
                                    JZ
                                                                                                                         if (BVar4 != 0) {
   piVar1 = (int *)cpuid_basic_info(0);
   puVar2 = (uint *)cpuid_Version_info(1);
            01 00 00
00401ab0 83 65 f0 00
                                                      dword ptr [EBP + local_14],0x0
00401ab4 33 c0
                                     XOR
                                                      EAX, EAX
                                                                                                                                watz - (unit )-(pub_vesson_minot),

are = puvarz[3];

{(((fuvars] ^ *puvarz & wxff3ffe, uvars == 0x106c0 || (uvars == 0x20660)) ||

(((((uvars = *puvarz & wxff3ffe, uvars == 0x106c0 || (uvars == 0x20660))) || (uvars == 0x20670))) || (uvars == 0x30670)))
00401ab6 53
                                     PUSH
                                                      FRX
00401ab8 57
                                     PUSH
                                                                                                                  80
00401ab9 33 c9
00401abb 8d 7d dc
                                                     ECX, ECX
                                                                                                                  81
82
                                                      EDI=>local_28,[EBP + -0x24]
                                                                                                                  83
84
85
                                                                                                                               DAT 0042c9e0 = DAT 0042c9e0 | 1;
00401abe 53
                                     PUSH
00401abf 0f a2
00401ac1 8b f3
                                     CPUID
                                                                                                                             if (*piVar1 < 7) {
00401ac3 5b
                                     POP
                                                                                                                  86
87
                                                                                                                               uVar7 = 0;
00401ac4 90
00401ac5 89 07
                                     NOP
MOV
                                                     dword ptr [EDI] =>local_28,EAX
                                                                                                                             88
89
90
91
92
93
94
95
96
00401ac7 89 77 04
00401aca 89 4f 08
00401acd 33 c9
                                                     dword ptr [EDI + local_24],ESI
dword ptr [EDI + local_20],ECX
                                     MOV
                                     MOV
XOR
                                                                                                                             if ((uVar7 & 0x200) != 0) {
                                                     dword ptr [EDI + local_1c], EDX
EAX, dword ptr [EBP + local_28]
EDI, dword ptr [EBP + local_24]
00401acf 89 57 0c
                                     MOV
                                                                                                                                   DAT_0042c9e0 = DAT_0042c9e0 | 2;
00401ad2 8b 45 dc
00401ad5 8b 7d e0
                                                     dword ptr [EBP + local_10],EAX
EDI.0x756e6547
                                                                                                                            DAT_0042c9dc = 1;
uVar5 = DAT_0042b010 | 2;
00401ad8 89 45 f4
00401adb 81 f7 47
```

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.



```
004216ed bf 61 00
                                                                                                                        FUN_0040bb30(_File);
dword ptr [EBP + local_18],0x3a0063
                                                                                                             136
137
                                                  dword ptr [EBP + local_14],0x5c
                                                                                                             138
139
140
141
142
                                                 qword ptr [EBP + local_10],XMM0
                                  MOVQ
                                  MOV
                                                 EBX,EDI
dword ptr [EBP + local_8],0x0
                                                                                                             143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
                                                                                                                            (((UVar1 == 2) || (UVar1 == 3)) || (U
lpParameter = (LPWSTR)FUN_004010f4(4);
                           LAB_00421711
                                                                                                                                                    ter.L"%c%c".iVar3.0x65):
                                                                                                                           00421711 8d 45 ec
                                                 EAX=>local_18,[EBP + -0x14]
word ptr [EBP + local_18],DI
                                                  EAX
dword ptr [->KERNEL32.DLL::GetDri
0042171f 83 e8 02
00421722 74 12
00421724 83 e8 01
                                                  EAX,0x2
LAB_00421736
                                 JZ
SUB
JZ
SUB
JZ
MOV
                                                 EAX,0x1
LAB_00421736
EAX,0x1
                                                                                                                     Overa - Overa + 1,

IVan = 1Van + 1;

) while (UVar4 < 0x7b);

MaitForMultipleObjects(DAT_0042f81c,&lpHandles_00430c30,1,0xffffffff);

return;
00421727 74 0d
00421727 74 0d
00421729 83 e8 01
0042172c 74 08
0042172e 8b 0d 1c
f8 42 00
                                                 LAB_00421736
ECX,dword ptr [DAT_0042f81c]
```

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.

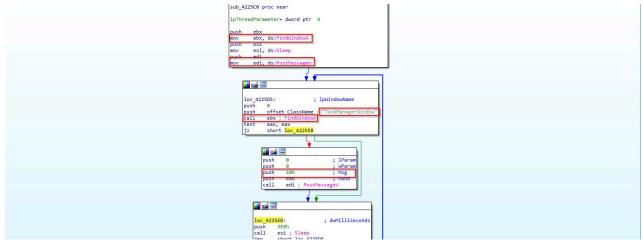


Image of CyberVolk Ransomware Static Analysis VII

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.



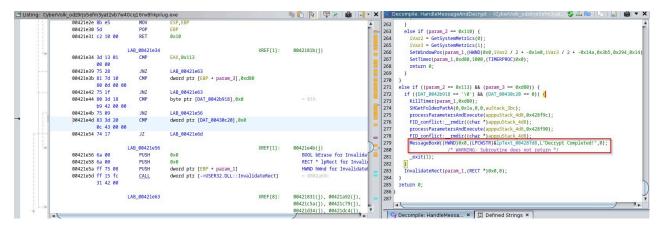


Image of CyberVolk Ransomware Static Analysis VIII

When the decryption process is completed, the program terminates itself using the <code>_exit(1)</code>; 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 multithreats. In this time, it was found that the task manager was blocked to prevent the process from being interrupted, but powershell was not blocked.



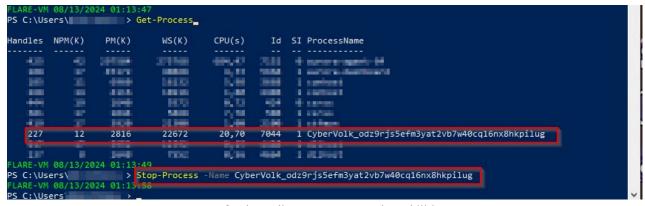


Image of CyberVolk Ransomware Vulnerabilities II

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.

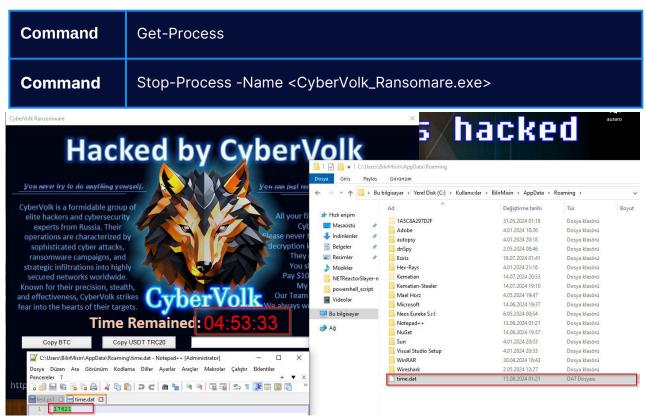


Image of CyberVolk Ransomware Vulnerabilities III

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.



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{
        description = "Yara rule for detecting CyberVolk Ransomware"
        author = "Aziz Kaplan"
        email = "aziz.kaplan@threatmonit.io"
        file_hash = "d08243e976e01baa5479a134577a1407daf4bec89a5f47bf2b803c0919917f5b"
        $0P1 = {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]
                                                   |PUSH<start_of_encryption>
               //685c904200
        $OP3 = {6a 24 68 5c 90 42 00}
                //6a24
                                                    PUSH0x24
                //685c904200
                                                    |PUSH<start_of_decryption>
        $0P4 = {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
                                                    LEAECX, [ESP+0x30]
                                                    |character_replacement
                //e82b0200
        $0P7 = {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]
                //e8c79cfeff
                                                    CALL_fopen
                //8d44<u>2438</u>
                                                    |LEAEAX,[ESP+0x38]
                //e823a2feff
                                                    |file_operation
        $0P9 = {68 80 0d 00 00 ff 75 08 ff 15 e8 31 42 00}
                //Timer Killer
                //68800d0000
                //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
        $0P13 = { 25 73 5c 74 69 6d 65 2e 64 61 74 00 }
                //time.dat
        $0P14 = { 25 73 5c 64 65 63 5f 6b 65 79 2e 64 61 74 00 }
                //dec_key.dat
        uint32(uint32(0x3C)) == 0x000004550 \text{ or}
                (filesize > 4 and uint32(0) == 0x464C457F) or
                (uint32(0) == 0xCEFAEDFE or uint32(0) == 0xCFFAEDFE) and
        (11 of ($OP*))
```



IOC List

Sha256
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
                        (x86)\'
            Files
            '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
```



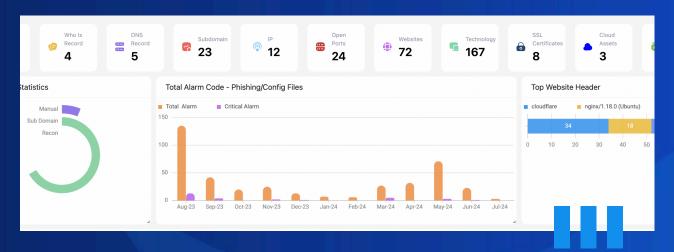
```
title: .CyberVolk Extension Detected
id: 37b2c73a-f147-4d93-842e-0b853b55de49
status: stable
description: Detects changes in file extensions where files are renamed to use
the .CyberVolk extension, typical in ransomware activity.
author: Aziz Kaplan <aziz.kaplan@threatmonit.io>
logsource:
    category: 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
```





More Information About ThreatMon



One Platform for all intelligence needs.

ThreatMon End-to-end intelligence is a cutting-edge, cloud-based SaaS platform that continuously monitors the dark and surface web, providing early warnings and actionable insights into emerging threats.

We are a SaaS platform designed to help businesses proactively detect and address threats before a cyber attack occurs. Unlike traditional cyber threat intelligence, we provide comprehensive and holistic cyber intelligence.

- Attack Surface Intelligence
- Fraud Intelligence
- Dark and Surface Web Intelligence
- Threat Intelligence



Contact Us:



Email Address team@threatmonit.io



https://x.com/MonThreat



https://www.linkedin.com/company/threatmon