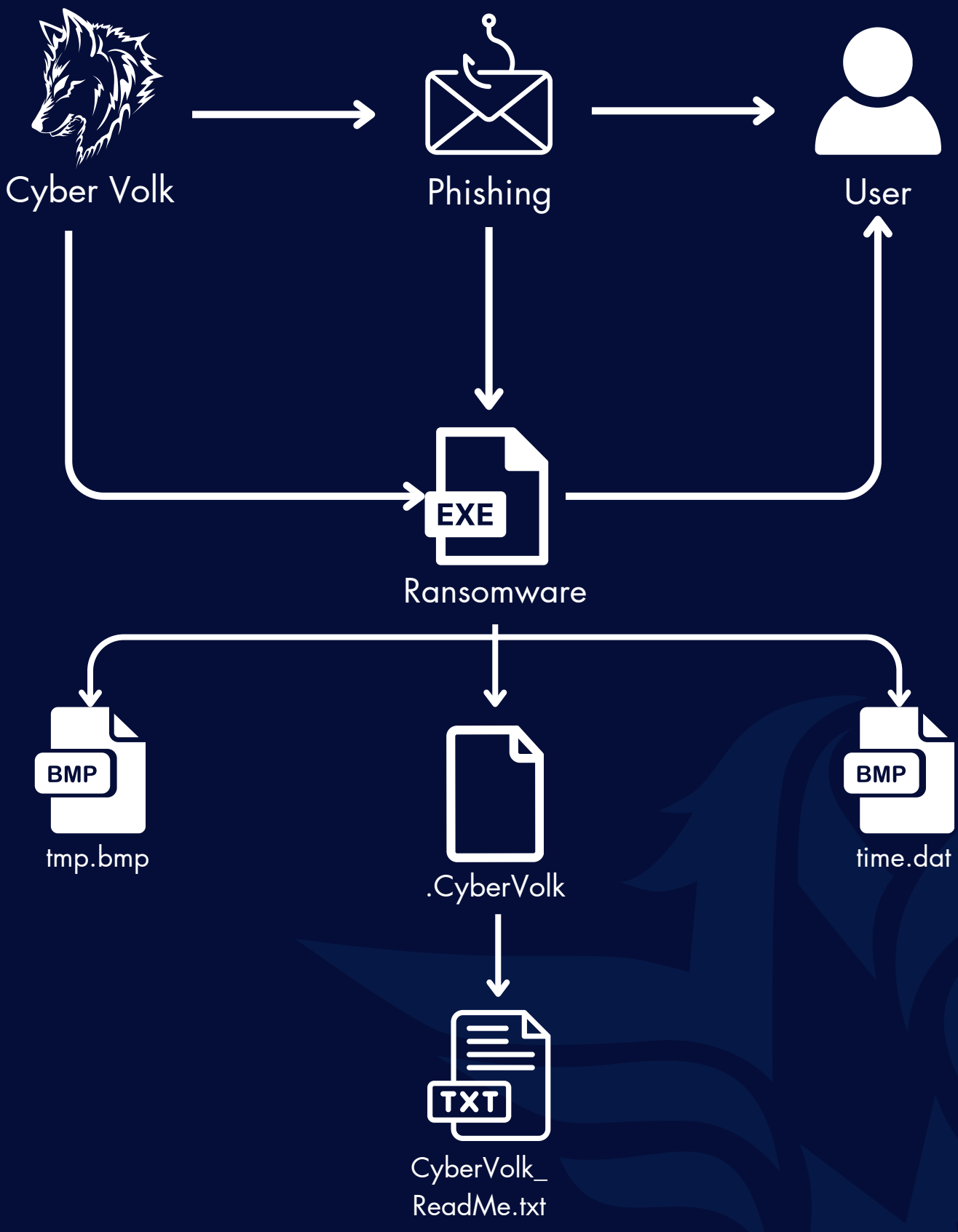
CYBERVOLK RANSOMWARE

TECHNICAL & MALWARE ANALYSIS

ATTACK CHAIN



DIAMOND MODEL

- Hide in Safe Mode
- Auto Spread
- Anti Debugging
- Environment Check
- Desktop Wallpaper Modification
- Plain Text Files
  - time.dat
  - CyberVolk_ReadMe.txt
- Non-Selective File Encryption
- AES Encryption
- RSA Encryption
- ChaCha20 - Poly1305 Encryption
- Quantum Resistant Algorithms
- Process Termination
- Window Enumeration

- Hacker-k
- GhostDoor_MalDev
- CyberVolk Group

- 4e66429d85967e344d8354e9b81719d
- a5274a96925125ea0d5c983a555f784d
- dee617dfd318e2fd9770a2e23d3234ab
- 25473274afa92567c8c8914a70b769ac

- Windows Servers
- Personal Windows Devices
- Windows Workstations

About Cybervolk Ransomware

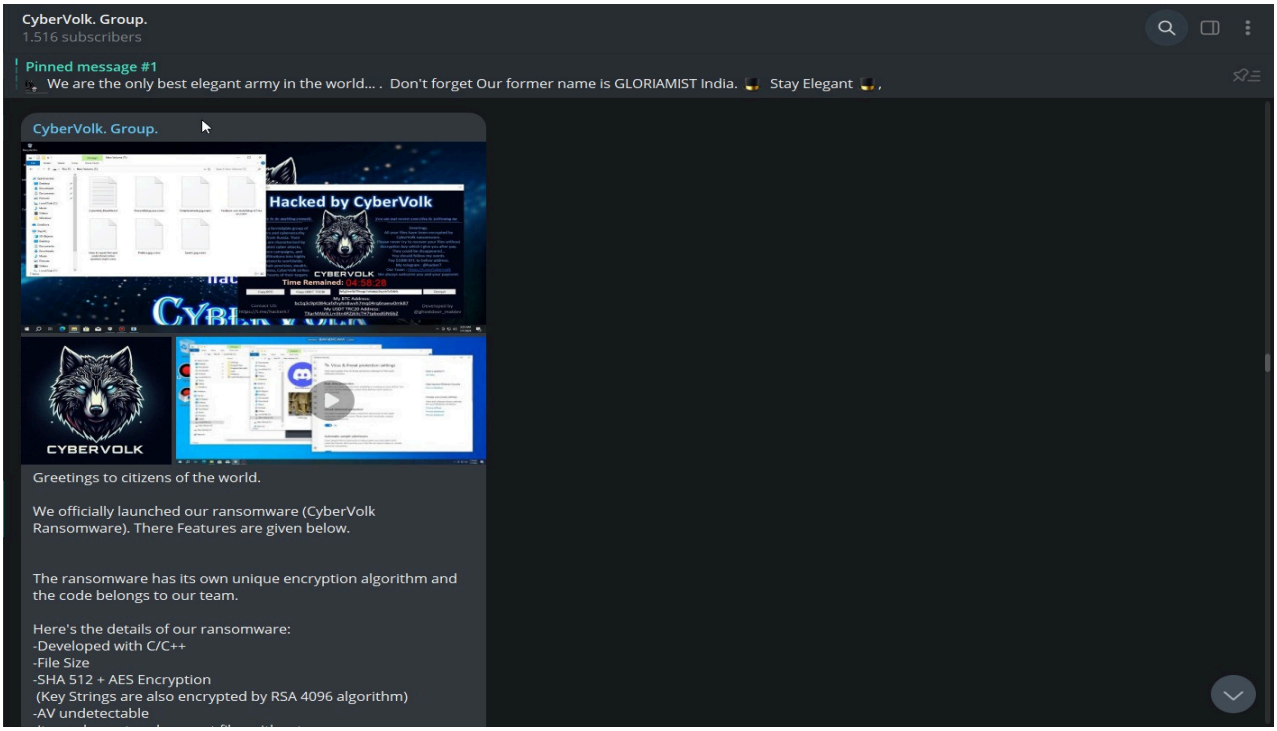


Image of CyberVolk Group Telegram Post

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.

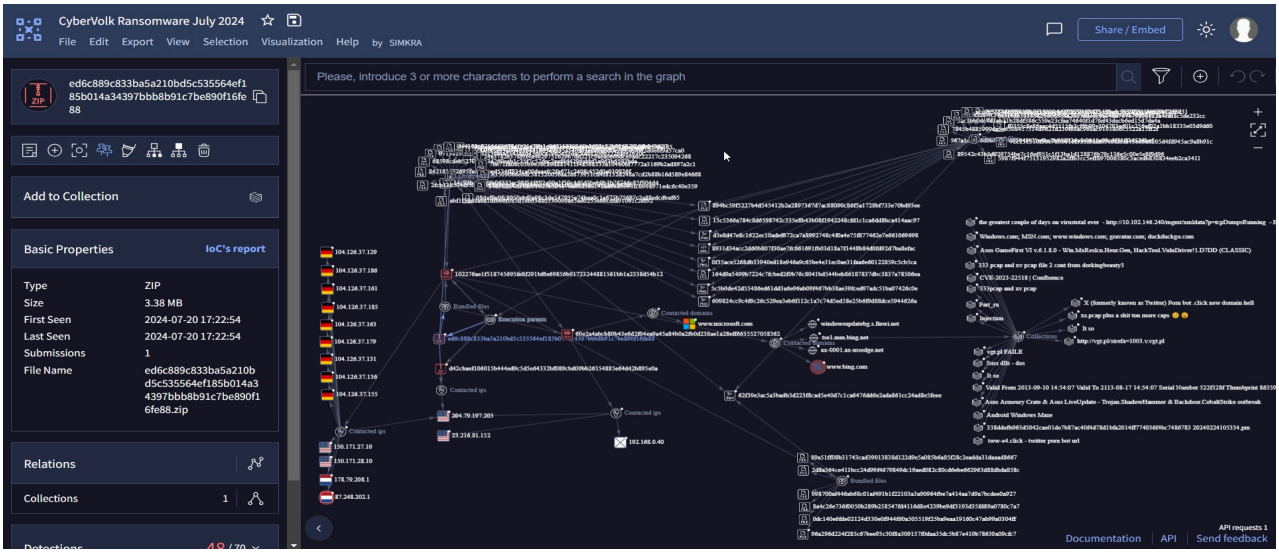


Image of CyberVolk Ransomware VirusTotal Leak

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

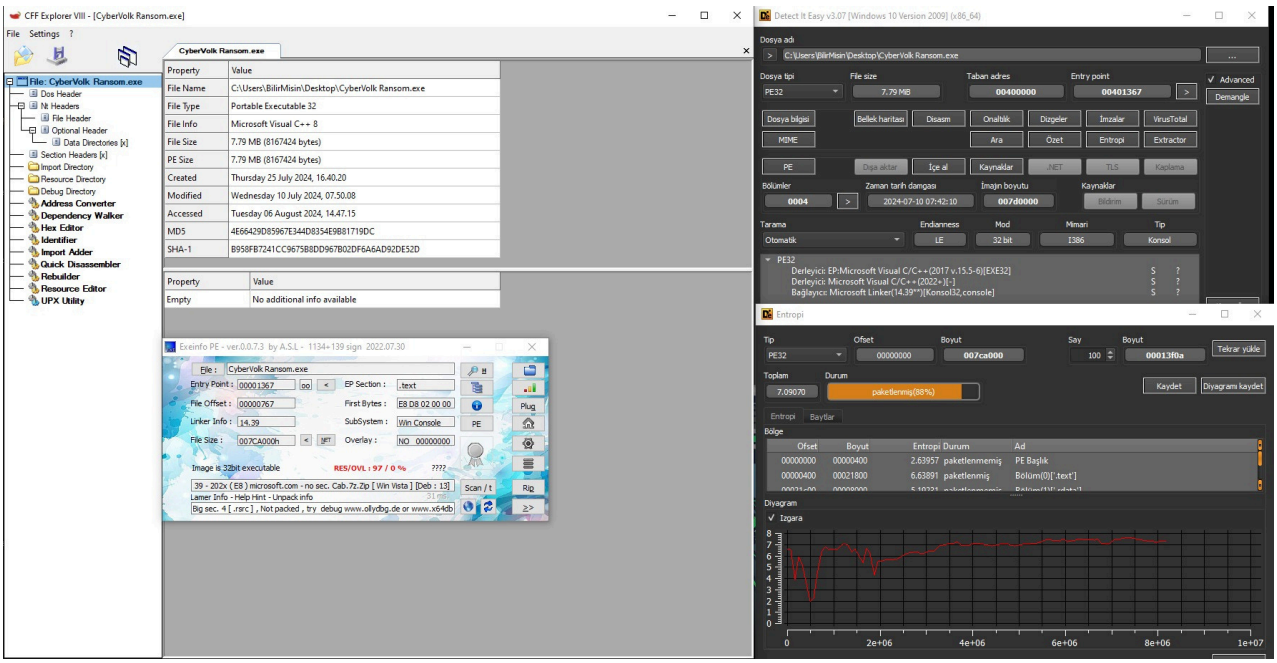


Image of CyberVolk Ransomware Characteristics

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

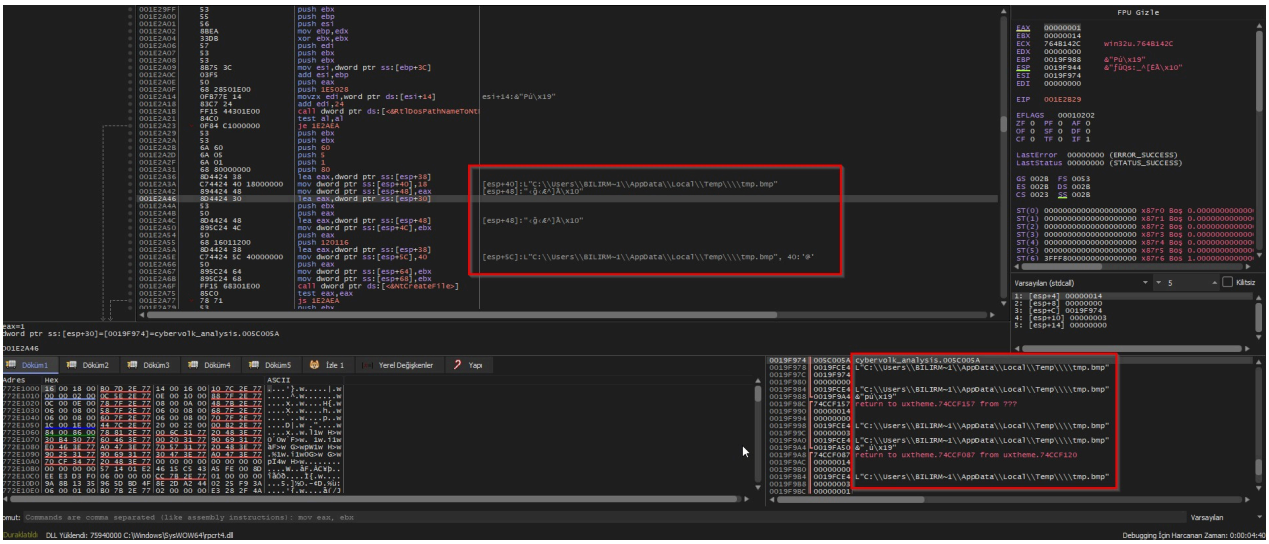


Image of CyberVolk Ransomware Dynamic Analysis I

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.

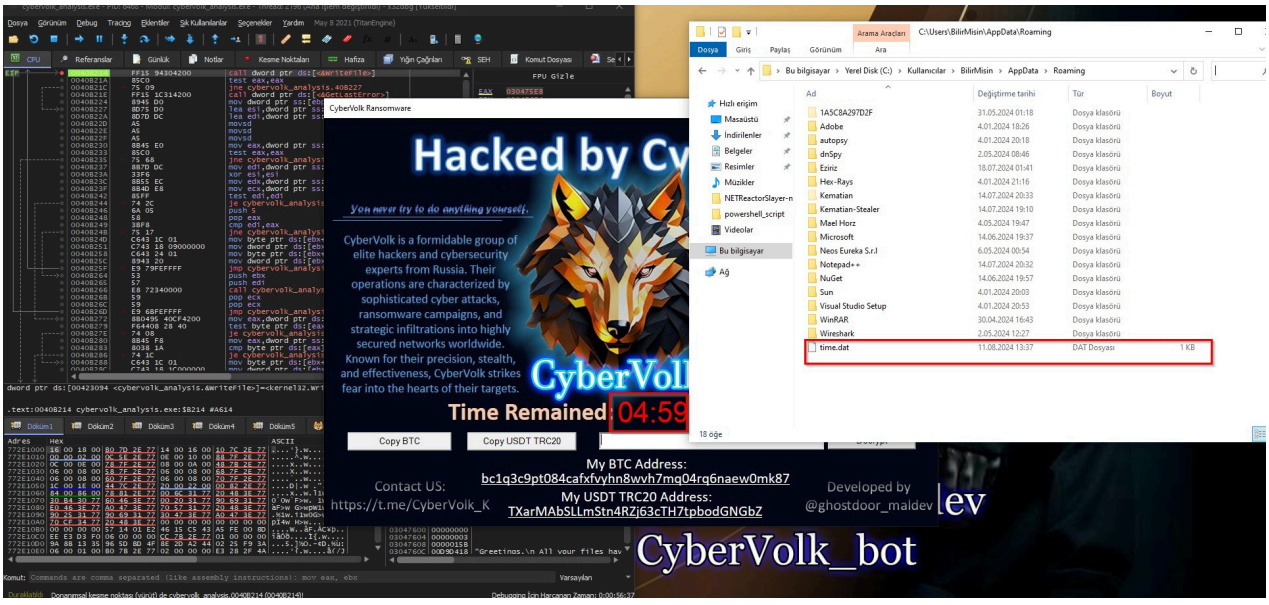


Image of CyberVolk Ransomware Dynamic Analysis II

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.

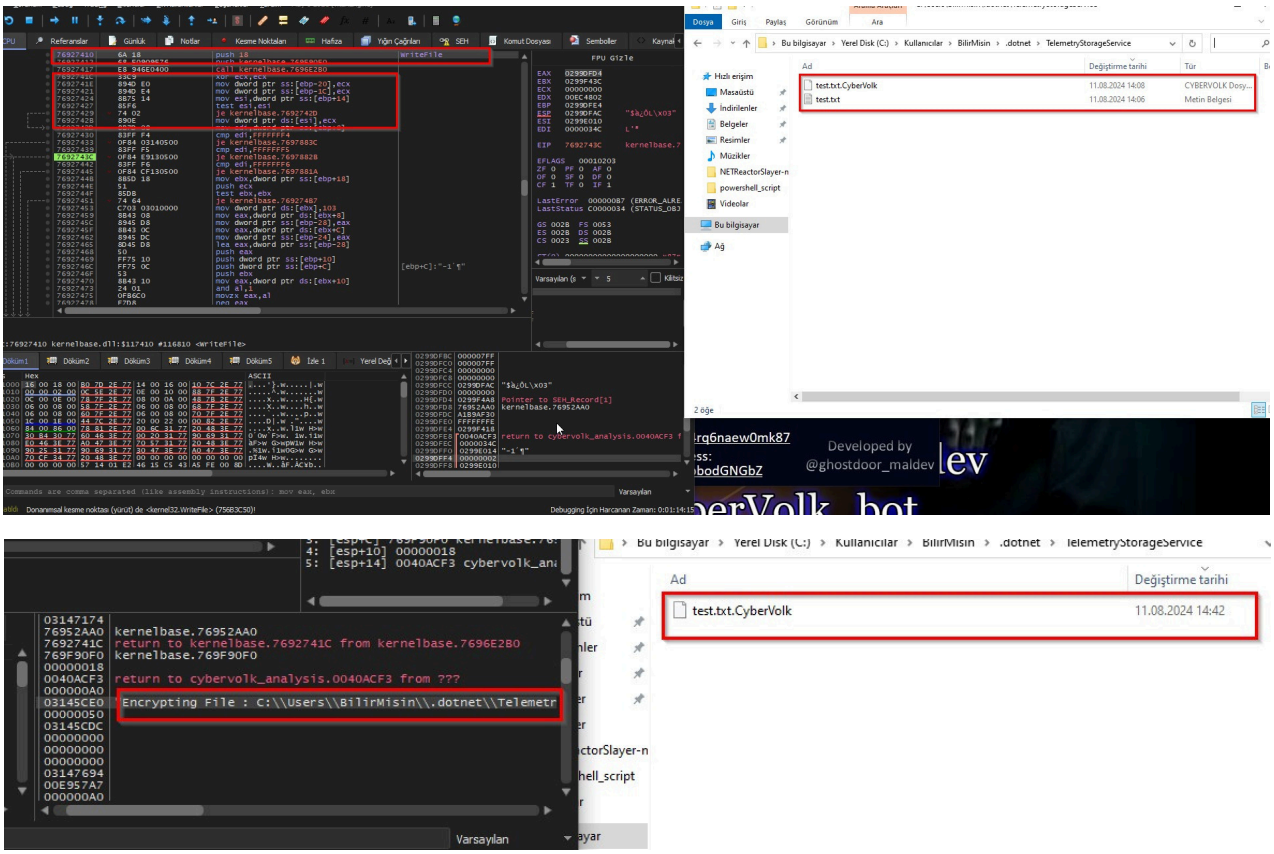


Image of CyberVolk Ransomware Dynamic Analysis III

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."

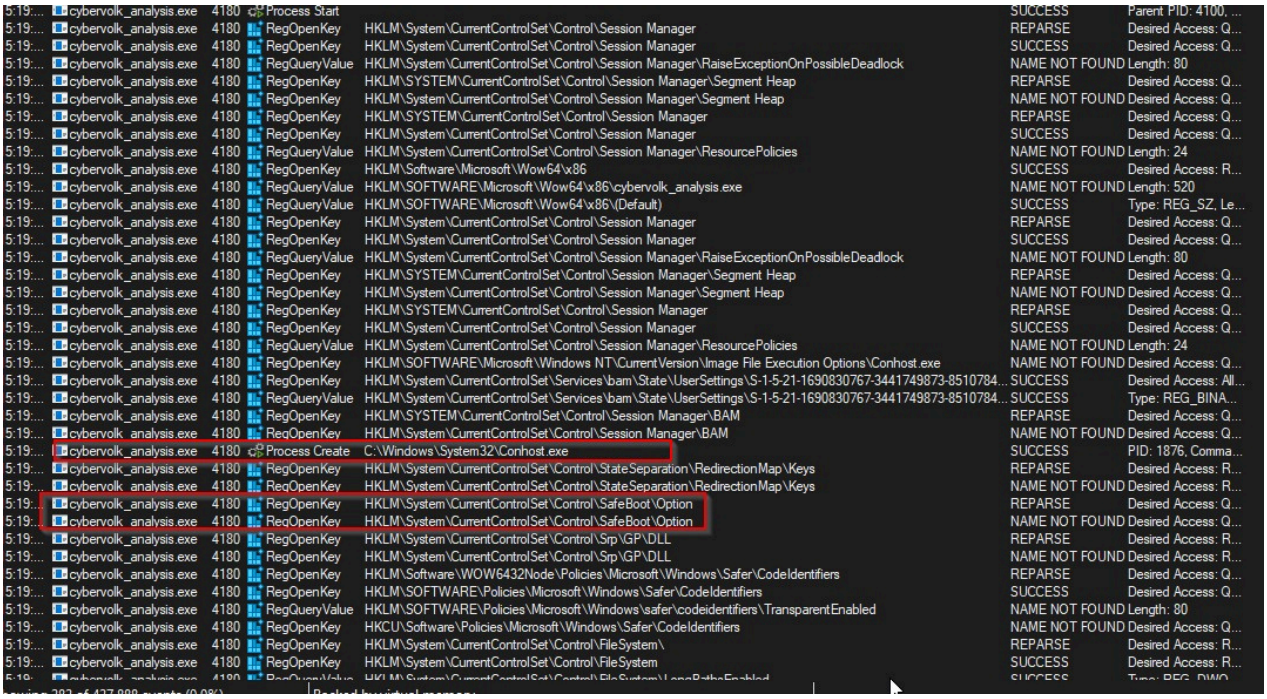


Image of CyberVolk Ransomware Dynamic Analysis V

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.

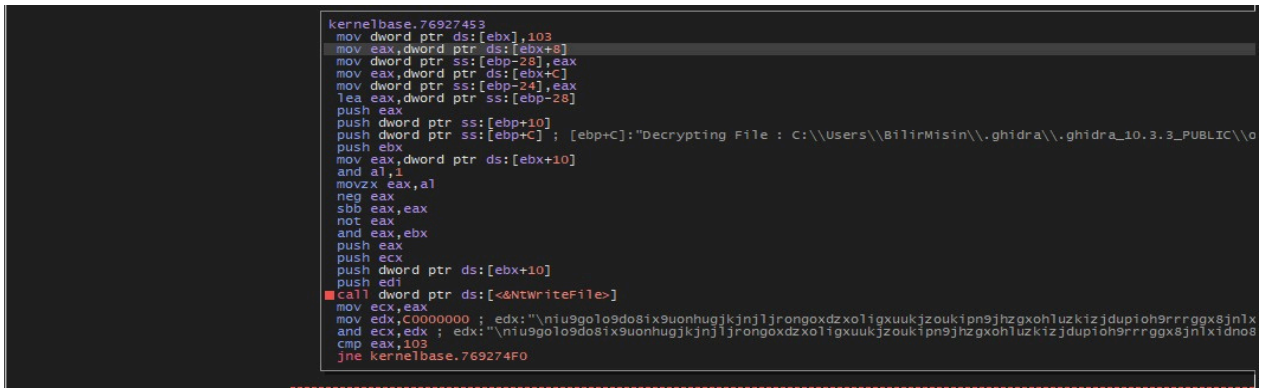


Image of CyberVolk Ransomware Dynamic Analysis VI

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.

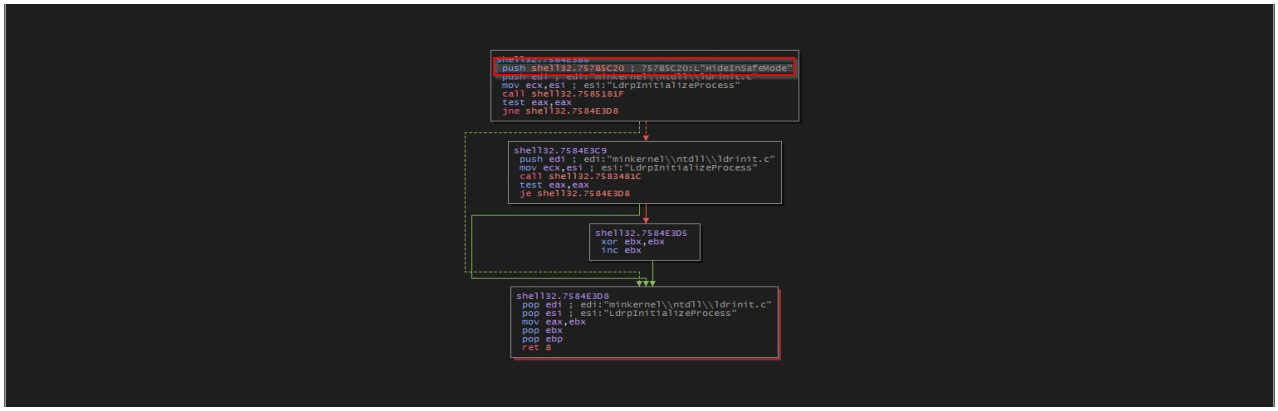


Image of CyberVolk Ransomware Dynamic Analysis VII

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

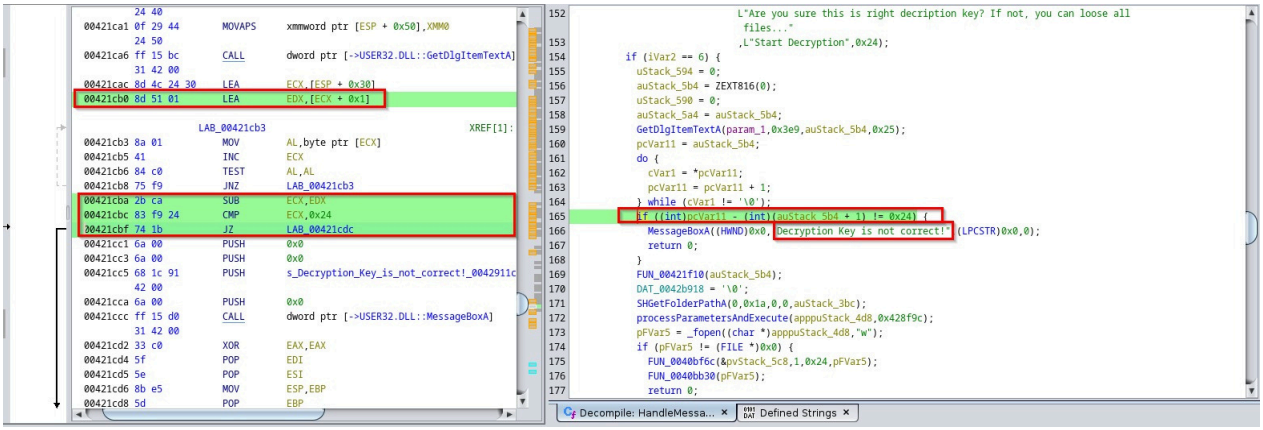


Image of CyberVolk Ransomware Static Analysis II

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.*

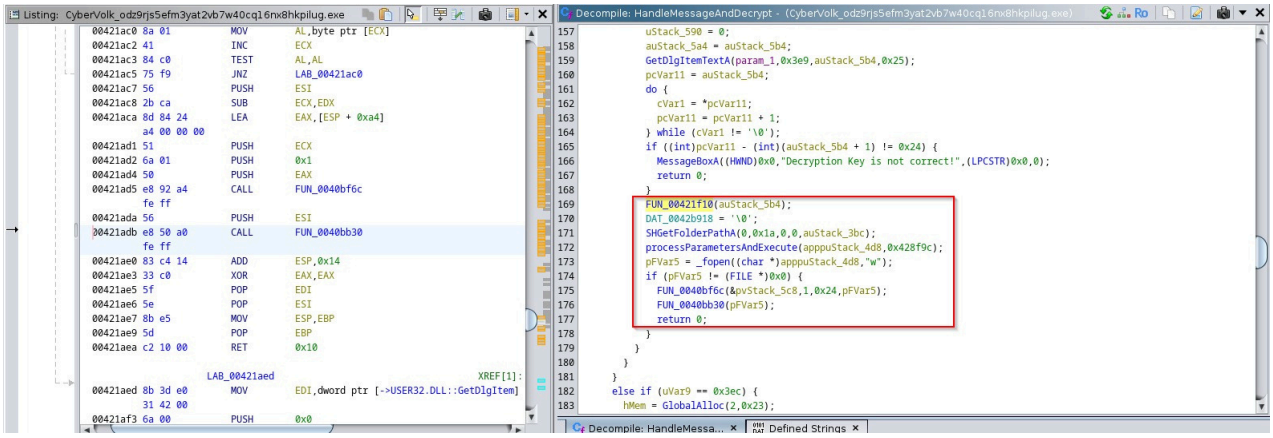


Image of CyberVolk Ransomware Static Analysis III

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**.

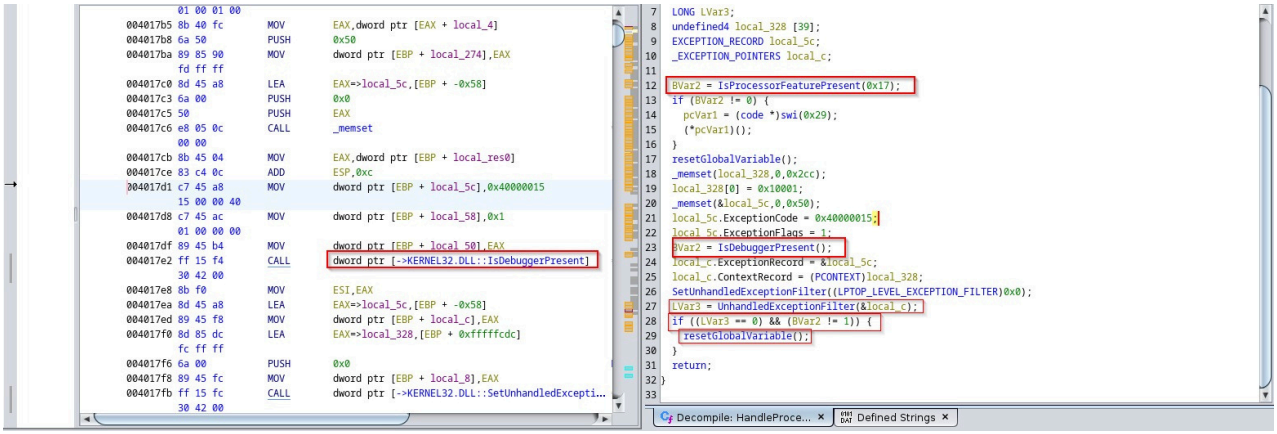


Image of CyberVolk Ransomware Static Analysis IV

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.

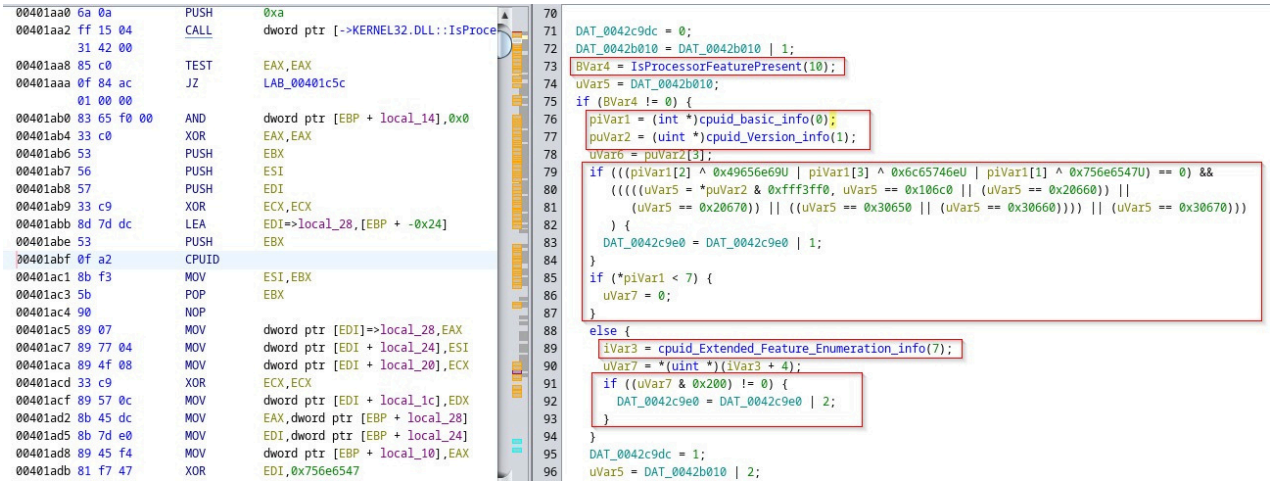


Image of CyberVolk Ransomware Static Analysis V

"**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.

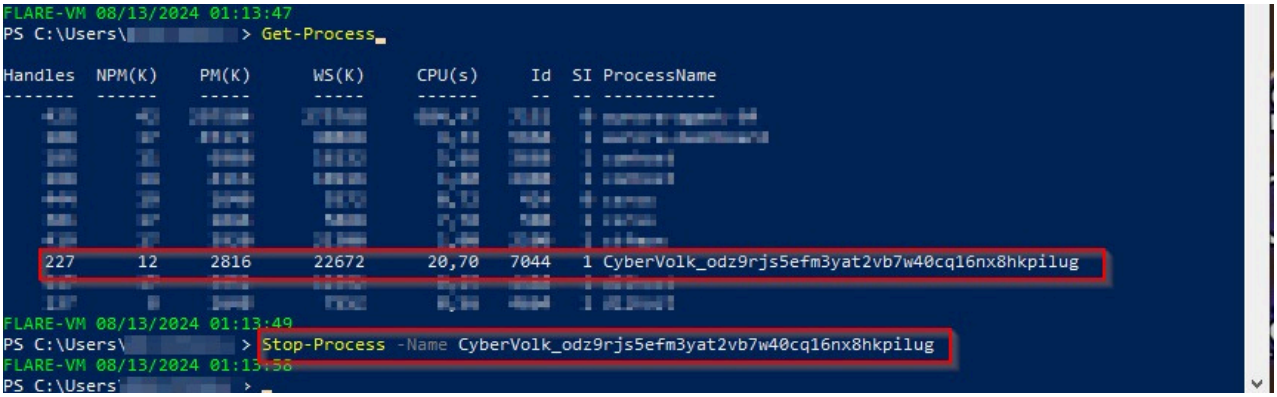


Image of CyberVolk Ransomware Vulnerabilities II

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>

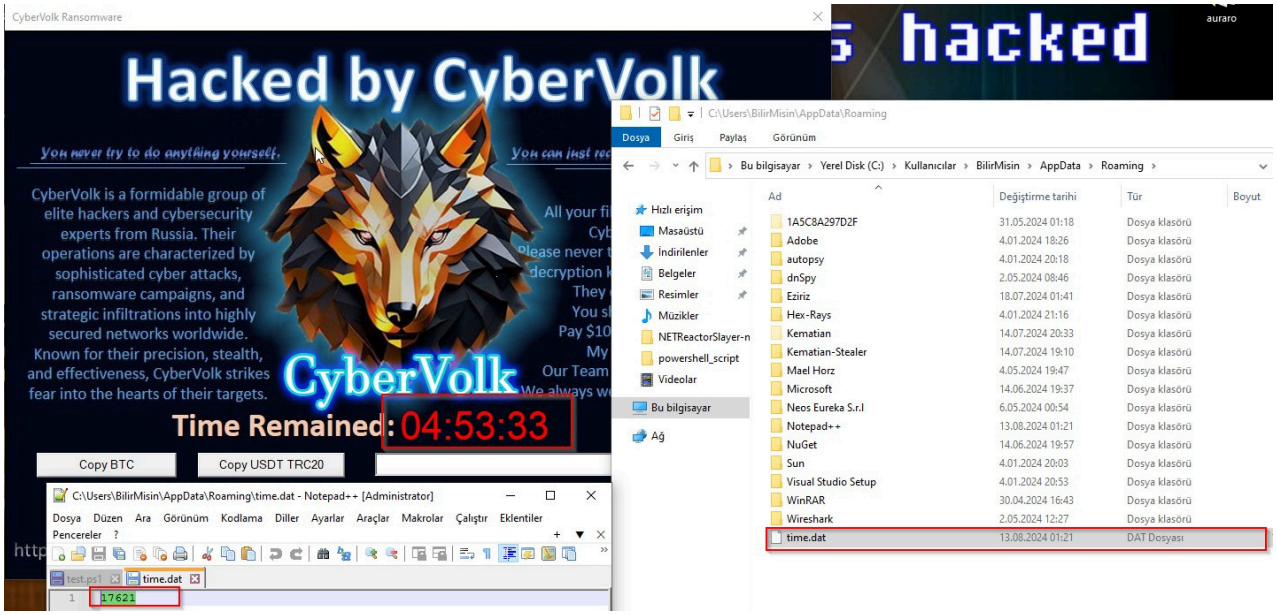


Image of CyberVolk Ransomware Vulnerabilities III

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.

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.*

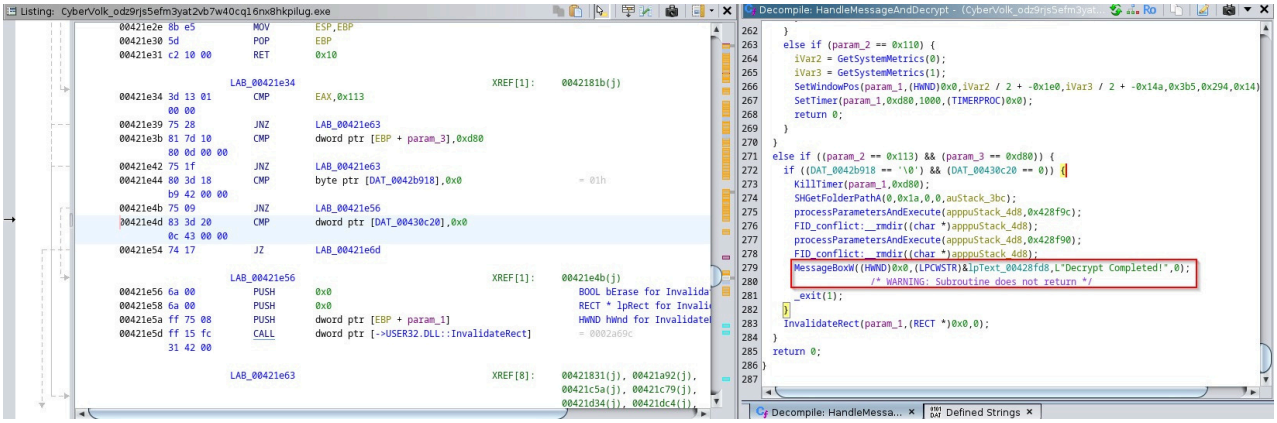


Image of CyberVolk Ransomware Static Analysis VIII

When the decryption process is completed, the program terminates itself using the **_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 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.

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]
                //e8c79cfeff                       |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
                //68800d0000                       |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
        $OP14 = { 25 73 5c 64 65 63 5f 6b 65 79 2e 64 61 74 00 }
                //dec_key.dat
    condition:
        uint32(uint32(0x3C)) == 0x00004550 or
                (filesize > 4 and uint32(0) == 0x464C457F) or
                (uint32(0) == 0xCEFAEDFE or uint32(0) == 0xCFFAEDFE) and
        (11 of ($OP*))
}
```

IOC List

Sha256	de0b74917fe24c2b38e2d1172b7352f88bf8b3df64b6d44ca5f317db85aeb324 70257c48ed8e1a3b57a7d6a5bed17837f60d630bdda0b22b048a3721569fe038 7d294c60c44b8b776c45e46e904a2de70ff4820e7e7863adb9f191c6554f9fb5 74b5a0ed14c7b8e26d51d4b9242e73686bad2e63cd11d9cbdb52e08fa34158c1

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
```