DragonForce Ransomware

Lat61 Threat Intelligence Team ⋮ ⋮ 9/3/2025

Introduction:

DragonForce ransomware is designed to encrypt files on compromised systems and requires ransom payments in cryptocurrency (Bitcoin) in return for the decryption key.

DragonForce is distributed via phishing emails, malicious websites, or exploiting vulnerabilities in systems.

DragonForce ransomware represents a Ransomware-as-a-Service (RaaS) attack. This ransomware threat actor, known as DragonForce, surfaced in mid-2023 and is believed to have originated from Malaysia. DragonForce employs a multi-extortion strategy, whereby they not only encrypt the data of their victims but also exfiltrate sensitive information.

In early 2025 DragonForce introduced white-label ransomware services under the brand 'Ransom Bay'. They also attacked major UK retailers including Marks & Spencer, Harrods in April–May 2025.

Infection Flowchart:

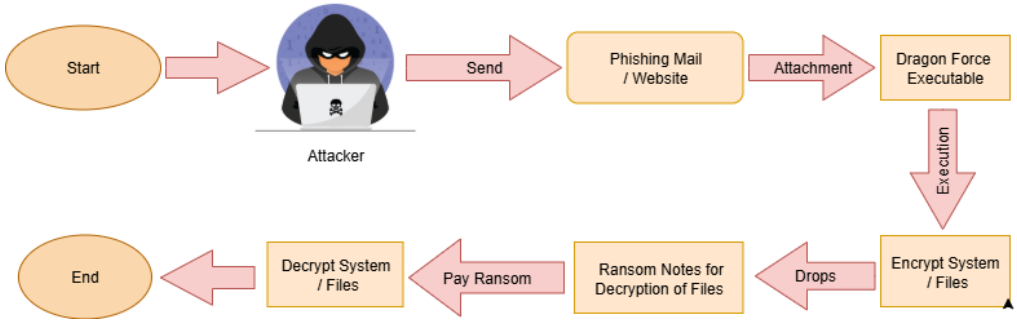


Figure 1: Infection Flowchart

Technical Analysis:

MD5: 9db8f7378e2df01c842cfcb617e64475

SHA-1 : eada05f4bfd4876c57c24cd4b41f7a40ea97274c

SHA-256 : c844d02c91d5e6dc293de80085ad2f69b5c44bc46ec9fdaa4e3efbda062c871c

Compiler : 32 bit C++ compiler executable file

Suspicious use ofAPI Call : CreateFileW, CreateMutexA, CopyFileW, Process32NextW, Process32FirstW, FindNextFileW, RegCreateKeyExW, CryptImportKey, CryptEncrypt, CryptGenRandom, RegOpenKeyExW, WriteFile, CreateProcessW.

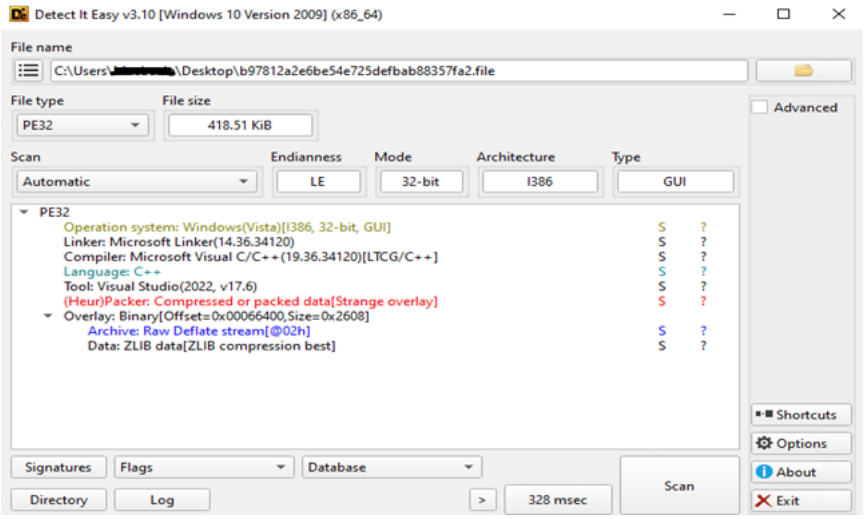


Figure 2: File Information

The log file created by ransomware at C:\Users\Public\log.log could potentially serve multiple purposes:

- Tracking Infection Details: It may record details about the victim's machine, such as the operating system, machine name, IP address, and other identifying features, which help the ransomware operators monitor infected systems.
- Monitoring Encryption Status: The log might track the progress of encryption, including which files or directories were successfully encrypted, any errors encountered, and whether the ransom payment process is proceeding.
- Logging Victim Interactions: If the victim interacts with the ransom note (e.g., trying to communicate with the attackers), the log could capture these actions, which could help guide future communication or attempts to extract payment.
- Debugging and Persistence: For the ransomware operators, this log could provide debugging information, helping them understand if the ransomware successfully encrypted files and if there are issues they need to address in further iterations.

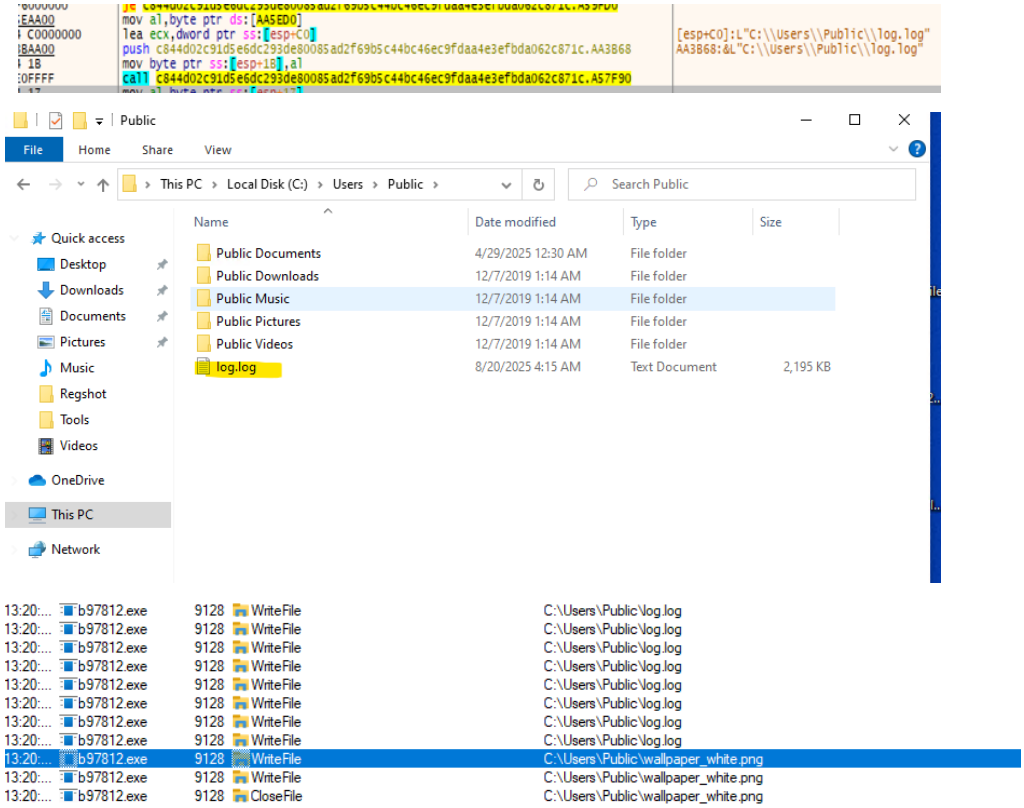


Figure 3: Dropped Log File

- The DragonForce ransomware binary used stack strings obfuscation technique where encrypted strings are stored on the stack and only decrypted to an array allocated at runtime. This approach helps to evade static

analysis and signature-based detection.

```
83C4 04          add esp,4
8B0D 7C588A00    mov ecx,dword ptr ds:[8A587C]
C685 4CFEFFFF 00 mov byte ptr ss:[ebp-1B4],0
C685 4DFEFFFF 27 mov byte ptr ss:[ebp-1B3],27                    27:'''
C685 4EFEFFFF 10 mov byte ptr ss:[ebp-1B2],10
C685 4FFEFFFF 5E mov byte ptr ss:[ebp-1B1],5E                    5E:'^'
C685 50FEFFFF 10 mov byte ptr ss:[ebp-1B0],10
C685 51FEFFFF 56 mov byte ptr ss:[ebp-1AF],56                    56:'V'
C685 52FEFFFF 10 mov byte ptr ss:[ebp-1AE],10
C685 53FEFFFF 58 mov byte ptr ss:[ebp-1AD],58                    58:'X'
C685 54FEFFFF 10 mov byte ptr ss:[ebp-1AC],10
C685 55FEFFFF 7D mov byte ptr ss:[ebp-1AB],7D                    7D:'}}'
C685 56FEFFFF 10 mov byte ptr ss:[ebp-1AA],10
C685 57FEFFFF 25 mov byte ptr ss:[ebp-1A9],25                    25:'%'
C685 58FEFFFF 10 mov byte ptr ss:[ebp-1A8],10
C685 59FEFFFF 2D mov byte ptr ss:[ebp-1A7],2D                    2D:'-'
C685 5AFEFFFF 10 mov byte ptr ss:[ebp-1A6],10
C685 5BFEFFFF 29 mov byte ptr ss:[ebp-1A5],29                    29:')'
C685 5CFEFFFF 10 mov byte ptr ss:[ebp-1A4],10
C685 5DFEFFFF 0C mov byte ptr ss:[ebp-1A3],C                     0C:'\f'
C685 5EFEFFFF 10 mov byte ptr ss:[ebp-1A2],10
C685 5FFEFFFF 61 mov byte ptr ss:[ebp-1A1],61                    61:'a'
C685 60FEFFFF 10 mov byte ptr ss:[ebp-1A0],10
C685 61FEFFFF 7A mov byte ptr ss:[ebp-19F],7A                    7A:'z'
C685 62FEFFFF 10 mov byte ptr ss:[ebp-19E],10
C685 63FEFFFF 53 mov byte ptr ss:[ebp-19D],53                    53:'S'
C685 64FEFFFF 10 mov byte ptr ss:[ebp-19C],10
C685 65FEFFFF 30 mov byte ptr ss:[ebp-19B],30                    30:'0'
C685 66FEFFFF 10 mov byte ptr ss:[ebp-19A],10
C685 67FEFFFF 5B mov byte ptr ss:[ebp-199],5B                    5B:'['
C685 68FEFFFF 10 mov byte ptr ss:[ebp-198],10
C685 69FEFFFF 34 mov byte ptr ss:[ebp-197],34                    34:'4'
C685 6AFEFFFF 10 mov byte ptr ss:[ebp-196],10
C685 6BFEFFFF 58 mov byte ptr ss:[ebp-195],58                    58:'X'
C685 6CFEFFFF 10 mov byte ptr ss:[ebp-194],10
C685 6DFEFFFF 58 mov byte ptr ss:[ebp-193],58                    58:'X'
C685 6EFEFFFF 10 mov byte ptr ss:[ebp-192],10
C685 6FFEFFFF 75 mov byte ptr ss:[ebp-191],75                    75:'u'
```

```
if ( !GetTokenInformation(TokenHandle, TokenUser, Block, ReturnLength, &ReturnLength) )
{
  v16 = 0;
  qmemcpy(v17, " 8V8v8#8=8Q8[8p8", 16);
  v17[16] = 3;
  v17[17] = 56;
  v17[18] = 85;
  v17[19] = 56;
  v17[20] = 81;
  v17[21] = 56;
  v17[22] = 87;
  v17[23] = 56;
  v17[24] = 20;
  v17[25] = 56;
  v17[26] = 87;
  v17[27] = 56;
  v17[28] = 124;
  v17[29] = 56;
  v17[30] = 3;
  v17[31] = 56;
  v17[32] = 5;
  v17[33] = 56;
  v17[34] = 44;
  v17[35] = 56;
  v17[36] = 36;
  v17[37] = 56;
  v17[38] = 91;
  v17[39] = 56;
  v17[40] = 126;
  v17[41] = 56;
  v17[42] = 3;
  qmemcpy(&v17[43], "8W8#8|8$8~8", 11);
  v17[54] = 1;
  qmemcpy(&v17[55], "8Q8&8\n8#8{8&888", 15);
  for ( k = 0; k < 0x46; ++k )
    v17[k] = (62 * (56 - (unsigned __int8)v17[k]) % 127 + 127) % 127;
  v10 = GetLastError();
  sub_421530(v17, v10);
  CloseHandle(hObject);
```

Figure 4: Stack String Decryption

- It generates a unique system ID for the compromised system to store collected information, as shown in the screenshot below. Attacker helps to identify the victim's machine by unique system ID.

```
68 00080000      push 800
8D8424 C4010000  lea eax,dword ptr ss:[esp+1C4]
50               push eax
8D8424 88000000  lea eax,dword ptr ss:[esp+88]      [esp+88]:L"[03:26:46] [TH:1F5C] build_key: A3AD3D0EDC8EAFFF\r\n"
50               push eax
E8 DCD30100      call C844d02c91d5e6dc293de80085ad2f69b5c44bc46ec9fdaa4e3efbda062c871c.87E7D5
8B3D 40318900    mov edi,dword ptr ds:[<WriteFile>]  edi:WriteFile
```

Ransomware often uses the current system time of the victim's machine for several reasons, which could include:

- System Identification:

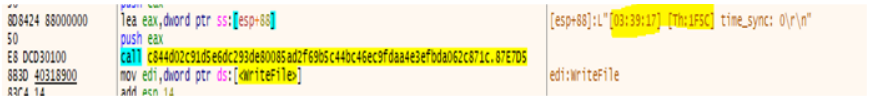
  The timestamp of the victim's machine can be used as a unique identifier for that system, especially when combined with other details like IP address or machine name. This helps the attackers track infected systems and may assist in customizing ransom demands based on the time zone or region.

- Encryption Time-Stamping:

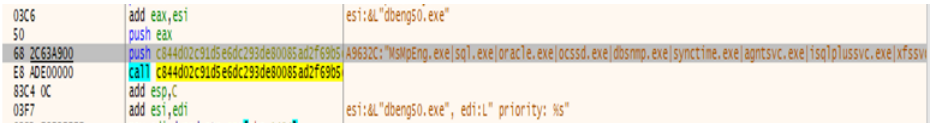
  The current system time might be used to timestamp encrypted files. This could help the attackers keep track of when the encryption took place, making it easier to monitor the victim's response over time (e.g., how long it takes the victim to pay the ransom).

- Time-Based Ransomware Behaviour:

  Some ransomware strains use time-based logic, such as delaying certain actions until a specific time or date, or accelerating the encryption process after a certain period. By capturing the system time, the ransomware can act more precisely based on the victim's clock.



- The ransomware terminates specific processes to prevent file-locking conflicts that would interfere with its encryption routine. The specific list of processes killed is shown in the image below.



- DragonForce maps multiple DLLs into the address space of the current process using the CreateFileMappingW and MapViewOfFile APIs.
- The DLLs were mapped as kernel32.dll, ws2_32.dll, advapi32.dll, RstrtMgr.dll, ole32.dll, netapi32.dll, IPHLPAPI.dll, shlwapi.dll, shell32.dll, and ntdll.dll. The malware uses the content of those fresh-mapped DLLs as a bypass technique, in order to replace the hooks placed from security vendors in these DLLs.

```
LoadLibraryA(LibFileName);
GetModuleFileNameW(hModule, Filename, 0x104u);
FileW = CreateFileW(Filename, 0x80000000, 1u, 0, 3u, 0x80u, 0);
v4 = FileW;
if ( !FileW )
  return (char)FileW;
GetFileSizeEx(FileW, &FileSize);
LowPart = FileSize.LowPart;
if ( !FileSize.LowPart || (FileMappingW = CreateFileMappingW(v4, 0, 2u, 0, 0, 0), (v7 = FileMappingW) == 0) )
{
  LOBYTE(FileW) = CloseHandle(v4);
  return (char)FileW;
}
v8 = MapViewOfFile(FileMappingW, 4u, 0, 0, LowPart);
v9 = (int)v8;
v25 = v8;
```

Figure 5: Mapping of dll

- The Dragonforce verifies if one of the DLLs listed above contains hook functions or not. If present it modifies the protection of the function code to 0x40 (PAGE_EXECUTE_READWRITE) and writes the unhooked code from the fresh-mapped DLLs . It then restores the old memory protection, as below image.

```
FileW = (int)GetProcAddress(hModule, v17);
v21 = (_QWORD *)FileW;
if ( FileW )
{
  LOBYTE(FileW) = *(_BYTE *)FileW;
  if ( *(_BYTE *)v21 == 0xE9 || (_BYTE)FileW == 0xFF && *((_BYTE *)v21 + 1) == 37 )
  {
    if ( !v20 )
      goto LABEL_39;
    v22 = 0;
    FileW = 0;
    while ( 1 )
    {
      v23 = v20[FileW];
      if ( v23 < *((_BYTE *)v21 + FileW) )
        break;
      if ( v23 > *((_BYTE *)v21 + FileW) )
      {
        v22 = 1;
        goto LABEL_38;
      }
      if ( (unsigned int)++FileW >= 2 )
        goto LABEL_38;
    }
    v22 = -1;

    if ( v22 )
    {

      flOldProtect = 0;
      v33 = 0;
      FileW = VirtualProtect(v21, 0x40u, 0x40u, &flOldProtect);
      if ( !FileW )
        return FileW;
      *v21 = *(_QWORD *)v20;
      *((_WORD *)v21 + 4) = *((_WORD *)v20 + 4);
      FileW = VirtualProtect(v21, 0x40u, flOldProtect, &v33);
```

Figure 6: Remove hooking

- DragonForce creates a mutex named “hsfjuukjzloqu28oajh727190”T to ensure that only one instance of ransomware is running at a single time.
- When the ransomware starts, it will attempt to create the mutex with the specified name. If the mutex already exists (i.e., another instance of the ransomware is already running), the process will fail or terminate, ensuring that only one copy of the malware is active.
- If the mutex doesn’t exist, the ransomware creates it and continues its execution. This effectively locks the system into only one active ransomware process.

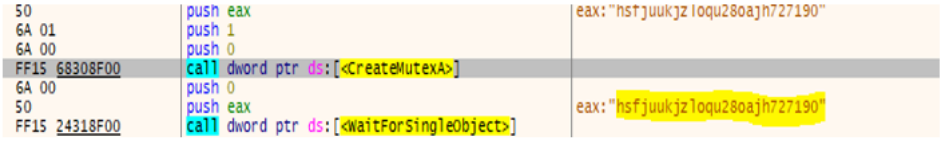


Figure 7: Mutex creation.

- DragonForce ransomware extracts the command-line parameters and compares them with the parameters list: -p, -m, -log, -size, and -nomutex.

  -p :-Encrypt Mode – path

  -m :-Encrypt Mode – all, local, net

  -log :-Specify log file

  -size :-Specify file encryption percentage

  -nomutex :-Do not create mutex

```
result = (int)CommandLineToArgvW(lpCmdLine, &pNumArgs);
v2 = result;
if ( result )
{
  v54 = 0;
  String2 = 11018;
  qmemcpy(v56, "6+++", sizeof(v56));
  for ( i = 0; i < 6; ++i )
    v56[i - 2] = (36 * (43 - (unsigned __int8)v56[i - 2]) % 127 + 127) % 127;
  v59 = (void *)pNumArgs;
  if ( pNumArgs <= 1 )
    goto LABEL_8;
  v4 = 1;
  while ( 1 )
  {
    v5 = (LPCWSTR *)(v2 + 4 * v4++);
    Src = v5;
    if ( !lstrcmpiW(*v5, &String2) )
      break;
    if ( v4 >= (int)v59 )
      goto LABEL_8;
```

Figure 8: Command line parameter

- Through GetLogicalDriveStringW API checking each drive in the victim’s machine and taking the handle of files present in drive. The use of the GetLogicalDriveStringsW API to enumerate and access each drive on the victim’s machine is a common tactic for ransomware and other malware types that need to operate across multiple drives, especially when their goal is to encrypt files or collect data.
- It provides a string of drive letters, which represent the different storage devices on the victim’s machine, such as C:\, D:\, E:\, etc.

- Using WMI’s query “SELECT * FROM Win32_ShadowCopy” to extract the shadow copies and delete them using the Delete method. SELECT * FROM Win32_ShadowCopy

cmd.exe /c C:\Windows\System32\wbem\WMIC.exe shadowcopy where “ID=’%s’” delete.

Encryption Process:

- DragonForce ransomware uses ChaCha8 encryption algorithms for encrypting files with targeted extensions.
- According to file size and file extension the dragonforce will encrypt files.
- File size < 3MB: the entire file is encrypted and 0x24 is added to the encrypted file's footer (Full encryption).
- File size > 3MB the first 3MB are encrypted (0x26 in the footer). In the case of targeted extensions, the entire file is encrypted. (Partly encryption)
- Files with Virtual machine extensions are encrypted by 20%.
- The calculation for the encryption of files with a size of 3 MB is illustrated in the below screenshot.

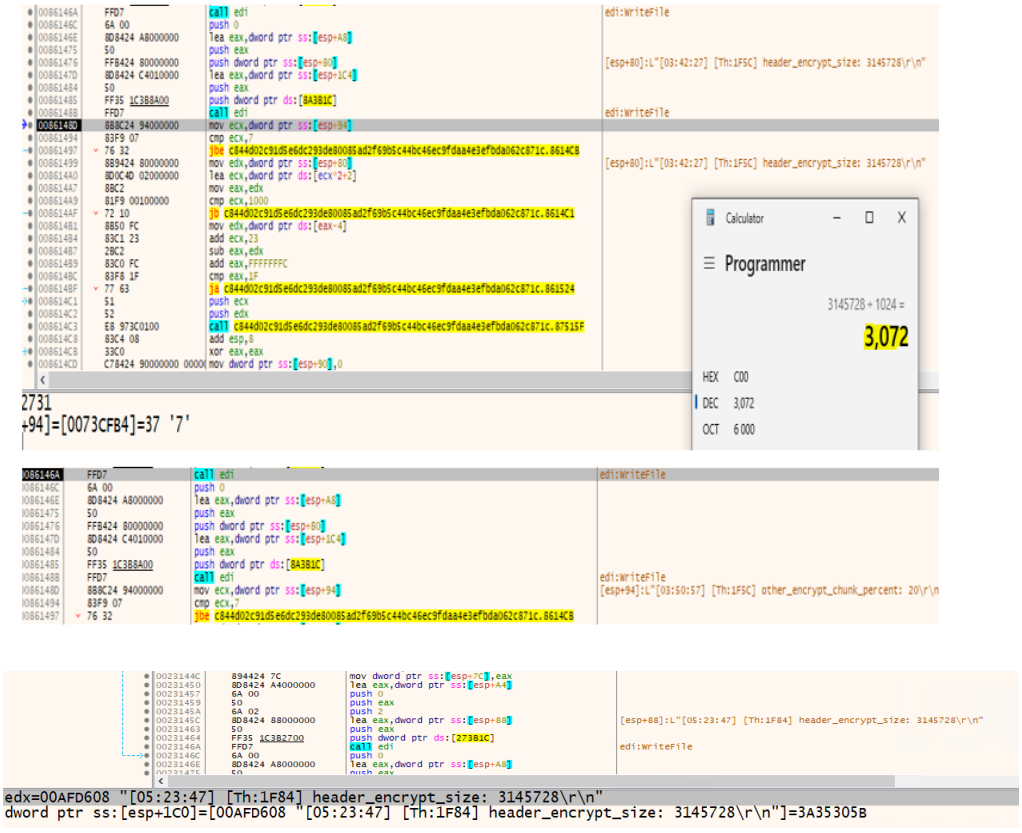


Figure 9: Encryption process

- The specific list of folders ,directory and file extensions that will be excluded from encryption routines. Below screenshot mentioned list of files and folder.
- Excluded folders: "temp, winnt, tmp, thumb, $Recycle.Bin, $RECYCLE.BIN, System Volume Information, Boot, Windows, perflogs, Public"
- Excluded file extensions: ".exe, .dll, .lnk, .sys, .msi, .bat"

Figure 10: Excluded files and folders.

- It creates the ransom note called readme.txt in every traversed directory:

```
6A 00       push 0
6A 02       push 2
6A 00       push 0
6A 00       push 0
68 00000040 push 40000000
50          push eax                                  eax:L"C:\\readme.txt"
FF15 60317700 call dword ptr ds:[<CreateFileW>]
```

- Ransomware generates 32 random bytes and then 8 random bytes representing the ChaCha8 key and nonce respectively.
- Every file will be encrypted with a different key and nonce.
- By using the API sequence of CryptImportKey,CryptGenRandom and CryptEncrypt encrypting the files on the victim's machine.

```
v5 = (int *)(a3 + 600);
if ( !CryptGenRandom(hProv, 8u, (BYTE *)(a3 + 600)) )
  return 0;
for ( j = a3 + 6483322; !(j % 4); ++j )
  ;
v6 = 64;
v7 = (_BYTE *)(a3 + 536);
do
{
  *v7++ = 0;
  --v6;
}
```

Figure 11: Random key and nonce generation

- To speed up the file system encryption operations, the executable creates multiple threads equal to the number of processors.

```
do
{
  v55 += 4;
  ++v54;
  *(_DWORD *)(v55 + dword_465798 - 4) = CreateThread(0, 0, StartAddress, &dword_465798, 0, 0);
}
while ( v54 < dword_46579C );
```

Figure 12: Encryption threads

- Ransomware extensions are added to the infected files, as shown in the screenshot below.
- After encryption of each traversed directory files renamed with new names as filename.dragonforce_encrypted extension.

```
83C4 0C     add esp,C
8D4D D8     lea ecx,dword ptr ss:[ebp-28]
BA 2C6B7800 mov edx,c844d02c91d5e6dc293de80085ad2f69b5c44bc46ec9fdaa4e3efbda062c871c.78682786B2C:".dragonforce_encrypted"
03F7        add esi,edi
E8 FFF6FFFF call c844d02c91d5e6dc293de80085ad2f69b5c44bc46ec9fdaa4e3efbda062c871c.7478C0
50          push eax
```

```
8D45 CC     lea eax,dword ptr ss:[ebp-34]             [ebp-34]:L".dragonforce_encrypted"
0F4745 CC   cmova eax,dword ptr ss:[ebp-34]           [ebp-34]:L".dragonforce_encrypted"
50          push eax                                  eax:L"C:\\!-\\kivrkp4v3mr3fmdv.dragonforce_encrypted"
53          push ebx                                  ebx:L"C:\\!-\\kivrkp4v3mr3fmdv.dragonforce_encrypted"
FF15 10316D00 call dword ptr ds:[<lstrcatw>]
8B4D E0     mov ecx,dword ptr ss:[ebp-20]
```

2kmxeewi34vx6s4q247x5znl3msn5snl2si3fpipbgsx5zd326sfmsx4zi77man4zv.dragonforce_encrypted	8/4/2025 1:23 PM	DRAGONFORCE_E...	23 KB
2kmxeewi34vx6s4q247x5znl3msn5snl2si3fpipbgsx5zd326sfmsx4zi77mdxwzt.dragonforce_encrypted	8/4/2025 1:23 PM	DRAGONFORCE_E...	33 KB
2kmxeewi34vx6s4q247x5znl3msn5snl2si3fpipbgsx5zd326sfmsx4zi77muo63g.dragonforce_encrypted	8/4/2025 1:23 PM	DRAGONFORCE_E...	5,225 KB
2kmxeewi34vx6s4q247x5znl3msn5snl2si3fpipbgsx5zd326sfmsx4zi77muo63k.dragonforce_encrypted	8/4/2025 1:23 PM	DRAGONFORCE_E...	6 KB
2kmxeewi34vx6s4q247x5znl3msn5snl2si3fpipbgsx5zd326sfmsx4zi77muo636.dragonforce_encrypted	8/4/2025 1:23 PM	DRAGONFORCE_E...	1,601 KB
3ksxg24i3ksnv2wyb6rbtuov.dragonforce_encrypted	8/4/2025 1:23 PM	DRAGONFORCE_E...	2,063 KB
3ksxgzwp36un7ewv3grbtuov.dragonforce_encrypted	8/4/2025 1:23 PM	DRAGONFORCE_E...	2,490 KB
b97812a2e6be54e725defbab	7/20/2025 10:20 PM	Application	419 KB

Figure 13: Ransomware Extension added

- Renaming infected files with ransomware extension as shown in the below screenshot.
- The ransomware uses Base32 encoding for filenames and appends a specific extension to the encrypted files.

```
68 94EA6D00 push b97812a2e6be54e725defbab88357fa2.6DEA94   6DEA94:L"Renaming: %s , %s"
E8 96460100 call b97812a2e6be54e725defbab88357fa2.6A1530
83C4 0C     add esp,C
8D55 9C     lea edx,dword ptr ss:[ebp-64]                 [ebp-64]:L"Data07.asp"
8D4D CC     lea ecx,dword ptr ss:[ebp-34]                 [ebp-34]:L"kivrkp4v3mr3fmdv"
E8 A87F0100 call b97812a2e6be54e725defbab88357fa2.6A4E50
837D E0 07  cmp dword ptr ss:[ebp-20],7
8D45 CC     lea eax,dword ptr ss:[ebp-34]                 [ebp-34]:L"kivrkp4v3mr3fmdv"
0F4745 CC   cmova eax,dword ptr ss:[ebp-34]               [ebp-34]:L"kivrkp4v3mr3fmdv"
68 B8EA6D00 push b97812a2e6be54e725defbab88357fa2.6DEAB8   6DEAB8:L"New name: %s"
E8 72460100 call b97812a2e6be54e725defbab88357fa2.6A1530
837D E0 07  cmp dword ptr ss:[ebp-20],7
8D4D CC     lea ecx,dword ptr ss:[ebp-34]                 [ebp-34]:L"kivrkp4v3mr3fmdv"
8D45 B4     lea eax,dword ptr ss:[ebp-4C]
0F474D CC   cmova ecx,dword ptr ss:[ebp-34]               [ebp-34]:L"kivrkp4v3mr3fmdv"
837D C8 07  cmp dword ptr ss:[ebp-38],7
51          push ecx
0F4745 B4   cmova eax,dword ptr ss:[ebp-4C]
50          push eax
68 D4EA6D00 push b97812a2e6be54e725defbab88357fa2.6DEAD4   6DEAD4:L"%s\\%s"
53          push ebx                                      ebx:L"C:\\!-\\kivrkp4v3mr3fmdv"
FF15 A4326D00 call dword ptr ds:[<wsprintfW>]
53          push ebx                                      ebx:L"C:\\!-\\kivrkp4v3mr3fmdv"
68 E0EA6D00 push b97812a2e6be54e725defbab88357fa2.6DEAE0   6DEAE0:L"Resulting name: %s"
E8 43460100 call b97812a2e6be54e725defbab88357fa2.6A1530
8B4D E0     mov ecx,dword ptr ss:[ebp-20]
```

Figure 14: Renaming files

- DragonForce ransomware creates a Persistence through registry key folder name as dragonforce_encrypted under HKEY_CLASSES_ROOT using RegCreateKeyExW.

- File name icon.ico contains default registry key in public directory called as Persistence mechanism of ransomware.

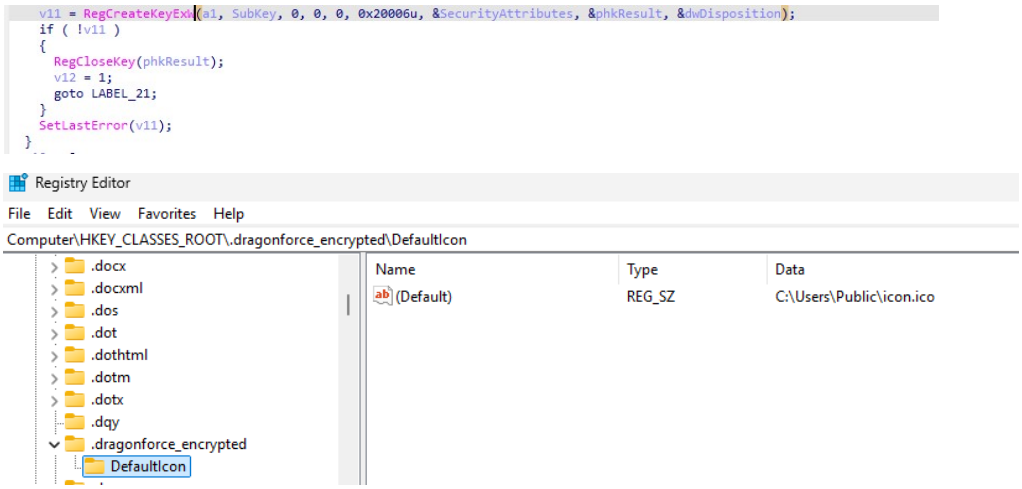


Figure 15: Registry entry

- Dropping and writing ransom note content in text file named as readme.txt for decryption of files or data on victim's machine. After paying ransom from the user, the attacker will decrypt the victim's machine files.

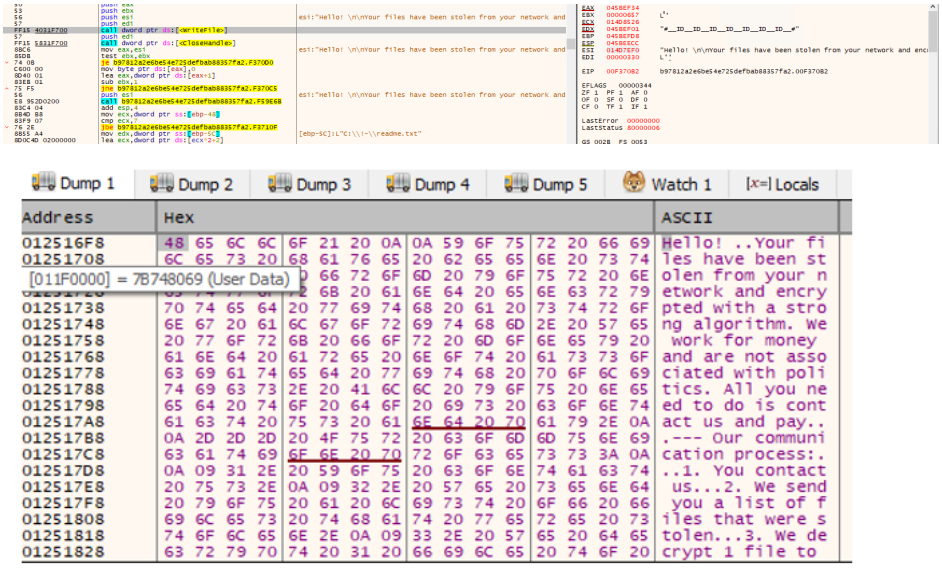


Figure 16: Decryption of the readme content

- The ransom note file is shown in the image below.

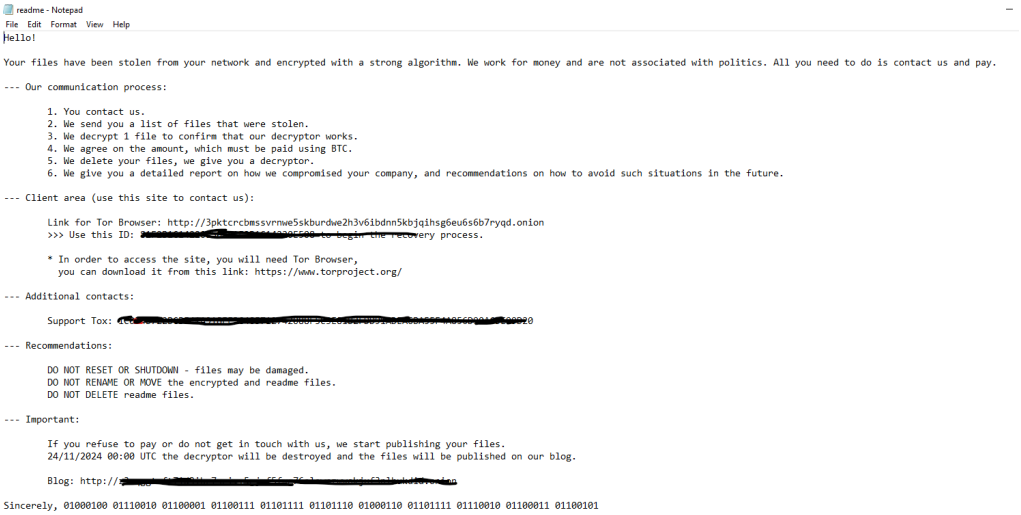


Figure 17: Ransom note

- After infection, the malware changes the screen wallpaper and drops a readme text file, as shown in the image below.

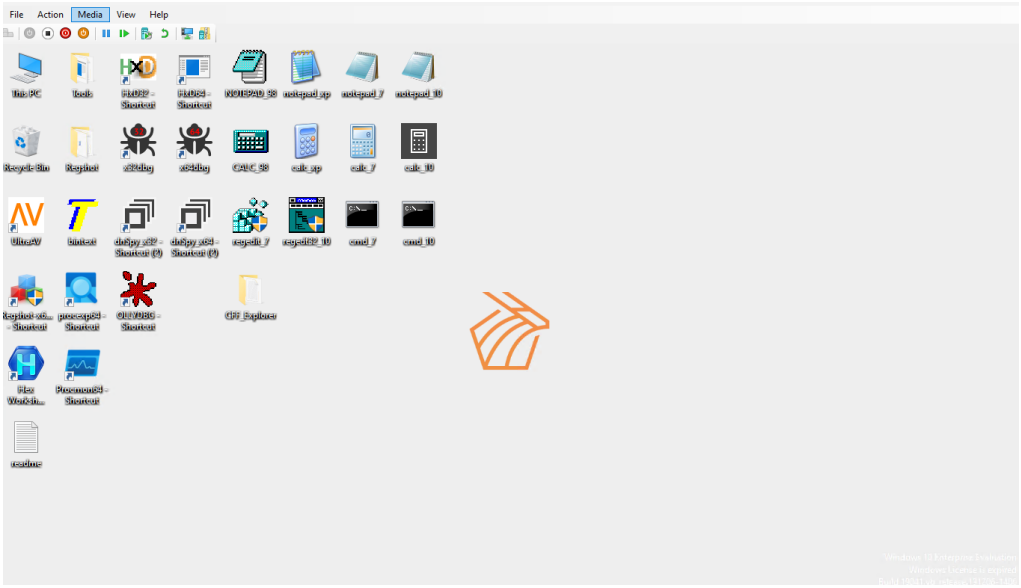


Figure 18: Infected machine desktop

- The following is a list of targeted file extensions:

.accdb, .accdc, .accde, .accdr, .accdt, .accft, .adb, .ade, .adf, .adp, .arc, .ora, .alf, .ask, .btr, .bdf, .cat, .cdb, .ckp, .cma, .cpd, .dacpac, .dad, .dadiagrams, .daschema, .db, .db-shm, .db-wal, .db3, .dbc, .dbf, .dbs, .dbt, .dbv, .dbx, .dcb, .dct, .dcx, .ddl, .dlis, .dp1, .dqy, .dsk, .dsn, .dtsx, .dxl, .eco, .ecx, .edb, .epim, .exb, .fcd, .fdb, .fic, .fmp, .fmp12, .fmpsl, .fol, .fp3, .fp4, .fp5, .fp7, .fpt, .frm, .gdb, .grdb, .gwi, .hdb, .his, .ib, .idb, .ihx, .itdb, .itw, .jet, .jtx, .kdb, .kexi, .kexic, .kexis, .lgc, .lwx, .maf, .maq, .mar, .mas, .mav, .mdb, .mdf, .mpd, .mrg, .mud, .mwb, .myd, .ndf, .nnt, .nrmlib, .ns2, .ns3, .ns4, .nsf, .nv, .nv2, .nwdb, .nyf, .odb, .oqy, .orx, .owc, .p96, .p97, .pan, .pdb, .pdm, .pnz, .qry, .qvd, .rbf, .rctd, .rod, .rodx, .rpd, .rsd, .sas7bdat, .sbf, .scx, .sdb, .sdc, .sdf, .sis, .spq, .sql, .sqlite, .sqlite3, .sqlitedb, .te, .temx, .tmd, .tps, .trc, .trm, .udb, .udl, .usr, .v12, .vis, .vpd, .vvv, .wdb, .wmdb, .wrk, .xdb, .xld, .xmlff, .abcddb, .abs, .abx, .accdw, .adn, .db2, .fm5, .hjt, .icg, .icr, .kdb, .lut, .maw, .mdn, .mdt.

- The following is a list of targeted virtual machine file extensions:

. vdi, .vhd, .vmdk, .pvm,.vmem,.vmsn, .vmsd, .nvram, .vmx, .raw, .qcow2, .subvol, .bin, .vsv, .avhd, .vmrs, .vhdx, .avdx, .vmcx, .iso.

MITRE ATT&CK:

MITRE Tactic	Technique	Description
Initial Access	Phishing	Delivers malware via spear-phishing emails
Execution	User Execution	Relies on victims opening malicious file attachments (scripts, or executables) delivered via phishing
Persistence	Registry Run Keys	Creates or modifies autorun Registry keys for persistence.
Privilege Escalation	Access Token Manipulation	DragonForce may also abuse token impersonation to gain system-level access
Defense Evasion	Obfuscated Files or Information	DragonForce obfuscates its payload by packing or encrypting parts of the code to evade detection and hinder static analysis.
Discovery	File and Directory Discovery	After gaining access, they recursively enumerate files and directories on compromised systems for encryption.
Lateral Movement	Remote Desktop Protocol	DragonForce connects to multiple internal systems over RDP to expand their foothold. RDP is used for lateral movement within the network.
Command & Control	Application Layer Protocol: Web Protocols	C2 communication is established using HTTP.
Impact	Data /System Encrypted	Deploys ransomware to encrypt files on victim systems, Once encryption is complete, the ransomware drops ransom note files on disk and often changes the desktop wallpaper to a ransom message

Conclusion:

- DragonForce ransomware is a highly dangerous threat. It demonstrates typical ransomware behavior, such as file encryption, ransom demands, and communication with a C2 server. However, it also exhibits advanced evasion techniques like obfuscation, anti-analysis.
- DragonForce ransomware deploys payloads derived from leaked LockBit3.0 and Conti source code. The encryption percentage of a file is determined by both file size and whether the extension is included in the ransomware's targeted list.

IOC:

SHA256	c844d02c91d5e6dc293de80085ad2f69b5c44bc46ec9fdaa4e3efbda062c871cb9bba02d18bacc4bc8d9e4f70657d3815
File Created	C:\Users\Public\log.logC:\Users\Public\wallpaper_white.pngC:\Users\Public\icon.icoreadme.txt
Registry key	HKCR\.dragonforce_encrypted
Mutex	hsfjuukjzloqu28oajh727190