New Mimic Ransomware Abuses Everything APIs for its Encryption Process

trendmicro.com/en_us/research/23/a/new-mimic-ransomware-abuses-everything-apis-for-its-encryption-p.html

January 26, 2023

Ransomware

Trend Micro researchers discovered a new ransomware that abuses the APIs of a legitimate tool called Everything, a Windows filename search engine developed by Voidtools that offers quick searching and real-time updates for minimal resource usage.

By: Nathaniel Morales, Earle Maui Earnshaw, Don Ovid Ladores, Nick Dai, Nathaniel Gregory Ragasa January 26, 2023 Read time: (words)

Trend Micro researchers discovered a new ransomware that abuses the APIs of a legitimate tool called Everything, a Windows filename search engine developed by Voidtools that offers quick searching and real-time updates for minimal resource usage. This ransomware (which we named Mimic based on a string we found in its binaries), was first observed in the wild in June 2022 and targets Russian and English-speaking users. It is equipped with multiple capabilities such as deleting shadow copies, terminating multiple applications and services, and abusing Everything32.dll functions to query target files that are to be encrypted.

In this blog entry, we will take a closer look at the Mimic ransomware, its components and functions, and its connection to the Conti builder that was leaked in early 2022.

Arrival and components

Mimic arrives as an executable that drops multiple binaries and a password-protected archive (disguised as Everything64.dll) which when extracted, contains the ransomware payload. It also includes tools that are used for turning off Windows defender and legitimate sdel binaries.

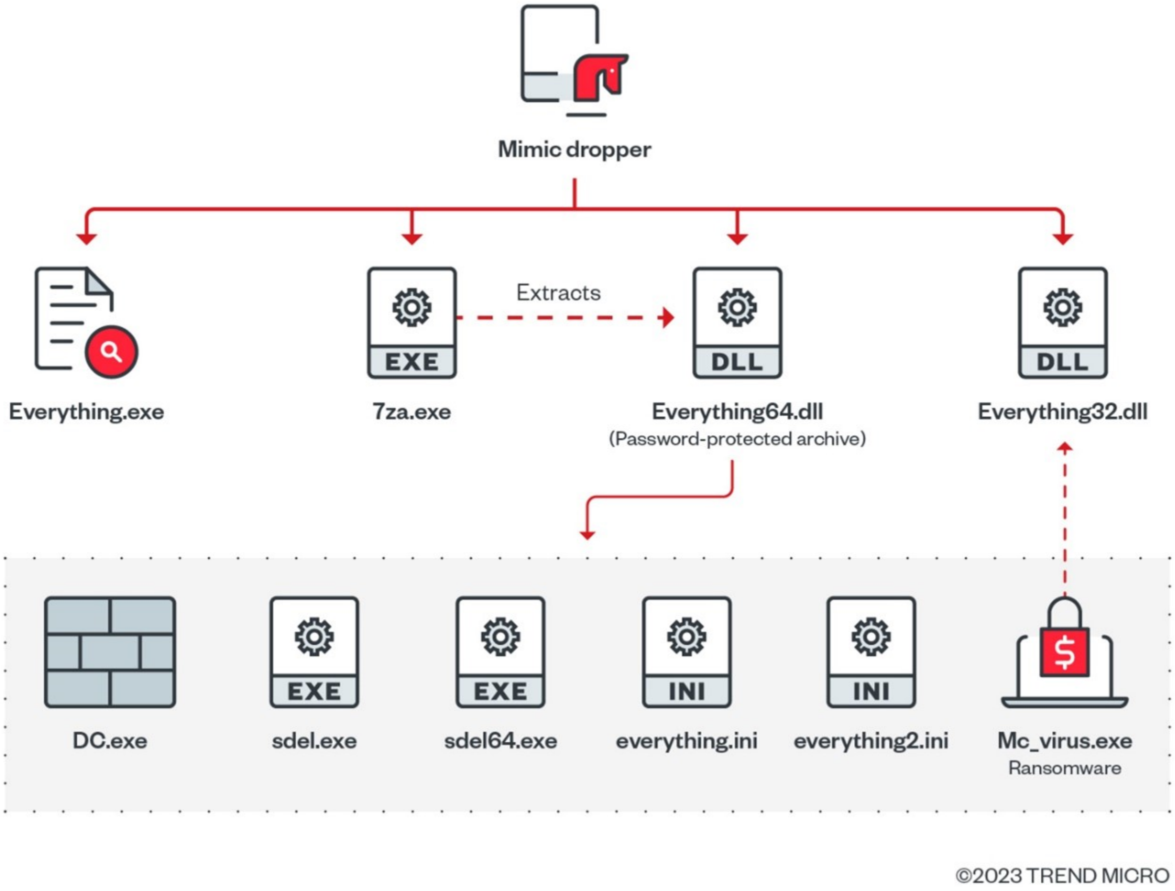


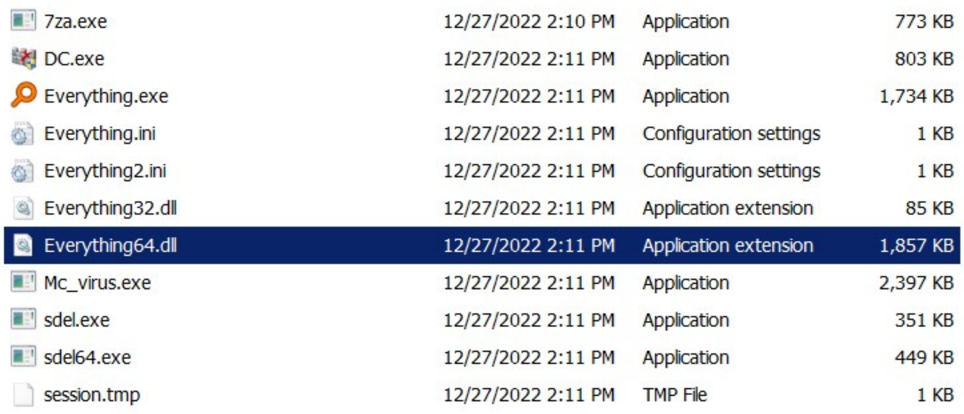
Figure 1. The Mimic ransomware components

Filename	Description
7za.exe	**Legitimate 7zip file that is used to extract the payload**
Everything.exe	**Legitimate Everything application**
Everything32.dll	**Legitimate Everything application**
Everything64.dll	**Password protected archive that contains the malicious payloads**

Table 1. Details of the Mimic ransomware components

When executed, it will first drop its components to the %Temp%/7zipSfx folder. It will then extract the password protected Everything64.dll to the same directory using the dropped 7za.exe via the following command:

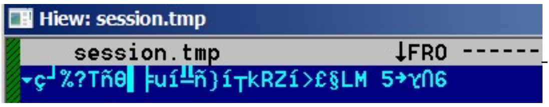
```
%Temp%\7ZipSfx.000\7za.exe" x -y -p20475326413135730160 Everything64.dll
```



7za.exe	12/27/2022 2:10 PM	Application	773 KB
DC.exe	12/27/2022 2:11 PM	Application	803 KB
Everything.exe	12/27/2022 2:11 PM	Application	1,734 KB
Everything.ini	12/27/2022 2:11 PM	Configuration settings	1 KB
Everything2.ini	12/27/2022 2:11 PM	Configuration settings	1 KB
Everything32.dll	12/27/2022 2:11 PM	Application extension	85 KB
Everything64.dll	12/27/2022 2:11 PM	Application extension	1,857 KB
Mc_virus.exe	12/27/2022 2:11 PM	Application	2,397 KB
sdel.exe	12/27/2022 2:11 PM	Application	351 KB
sdel64.exe	12/27/2022 2:11 PM	Application	449 KB
session.tmp	12/27/2022 2:11 PM	TMP File	1 KB

_Figure 2. Mimic ransomware’s dropped components

It will also drop the session key file session.tmp to the same directory, which will be used for continuing the encryption in case the process is interrupted.



```
Hiew: session.tmp
   session.tmp                ↓FRO ------
▾ç┘%?Tñθ▌╞uí╨ñ}í┬kRZí>£§LM 5→γ∩6
```

_Figure 3. The content of session.tmp

It will then copy the dropped files to “%LocalAppData%\{Random GUID}\”, after which the ransomware will be renamed to bestplacetolive.exe and the original files deleted from the %Temp% directory.

Based on our analysis, Mimic supports other command line arguments as shown in table 2.

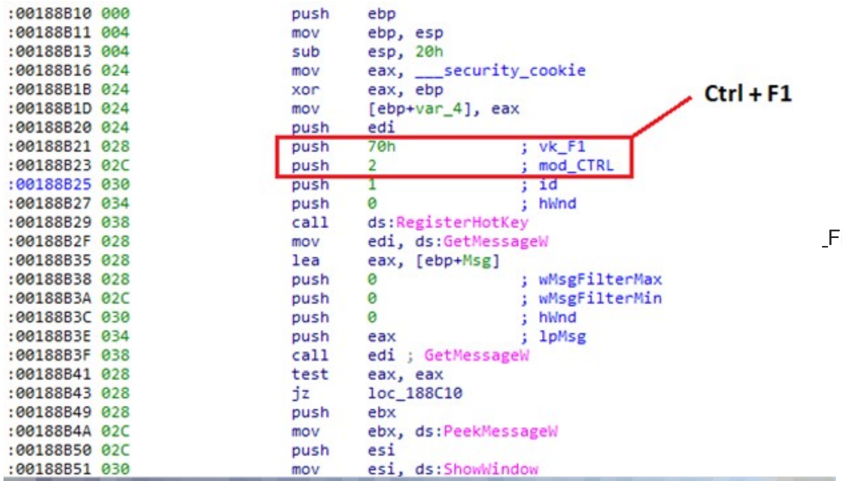
Cmdline option	Acceptable values	Description
-dir	**File path to be encrypted**	**Directory for encryption**
-e	all local net watch ul1 ul2	Encrypt all (Default) Encrypt Local files Encrypt files on Network shares ul:unlocker Creates a thread with interprocess communication and tries to unlock certain memory addresses from another process
-prot		**Protects the ransomware from being killed**
-pid	**<integer>**	**The process identifier (PID) of the previously-running ransomware.**

Table 2. Arguments accepted by Mimic ransomware

Mimic ransomware analysis

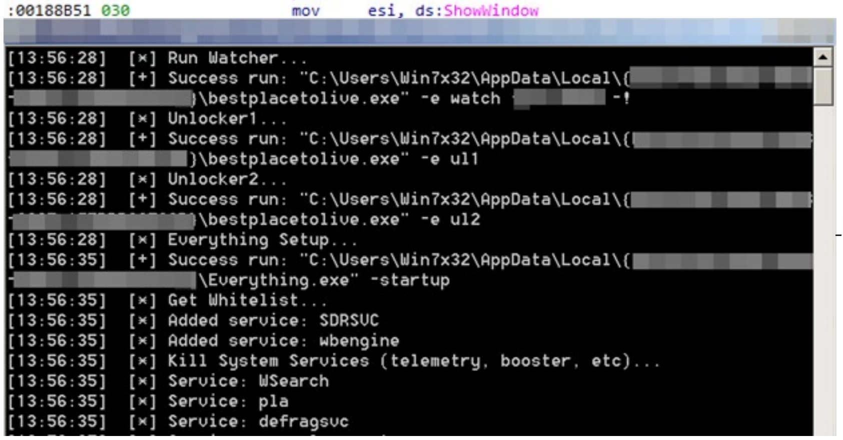
Mimic ransomware consists of multiple threads that employ the CreateThread function for faster encryption and render analysis more challenging for security researchers.

When executed, it will first register a hotkey (Ctrl + F1, using the RegisterHotKey API) that displays the status logs being performed by the ransomware.



```
:00188B10 000                 push    ebp
:00188B11 004                 mov     ebp, esp
:00188B13 004                 sub     esp, 20h
:00188B16 024                 mov     eax, ___security_cookie
:00188B1B 024                 xor     eax, ebp
:00188B1D 024                 mov     [ebp+var_4], eax
:00188B20 024                 push    edi
:00188B21 028                 push    70h             ; vk_F1
:00188B23 02C                 push    2               ; mod_CTRL
:00188B25 030                 push    1               ; id
:00188B27 034                 push    0               ; hWnd
:00188B29 038                 call    ds:RegisterHotKey
:00188B2F 028                 mov     edi, ds:GetMessageW
:00188B35 028                 lea     eax, [ebp+Msg]
:00188B38 028                 push    0               ; wMsgFilterMax
:00188B3A 02C                 push    0               ; wMsgFilterMin
:00188B3C 030                 push    0               ; hWnd
:00188B3E 034                 push    eax             ; lpMsg
:00188B3F 038                 call    edi ; GetMessageW
:00188B41 028                 test    eax, eax
:00188B43 028                 jz      loc_188C10
:00188B49 028                 push    ebx
:00188B4A 02C                 mov     ebx, ds:PeekMessageW
:00188B50 02C                 push    esi
:00188B51 030                 mov     esi, ds:ShowWindow
```

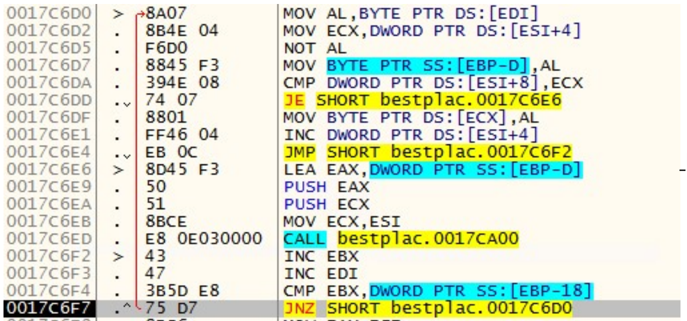
Figure 4. The function used for registering the hotkey



```
[13:56:28]  [×] Run Watcher...
[13:56:28]  [+] Success run: "C:\Users\Win7x32\AppData\Local\{[redacted]
-[redacted]}\bestplacetolive.exe" -e watch [redacted] -!
[13:56:28]  [×] Unlocker1...
[13:56:28]  [+] Success run: "C:\Users\Win7x32\AppData\Local\{[redacted]
[redacted]}\bestplacetolive.exe" -e ul1
[13:56:28]  [×] Unlocker2...
[13:56:28]  [+] Success run: "C:\Users\Win7x32\AppData\Local\{[redacted]
-[redacted]}\bestplacetolive.exe" -e ul2
[13:56:28]  [×] Everything Setup...
[13:56:35]  [+] Success run: "C:\Users\Win7x32\AppData\Local\{[redacted]
-[redacted]}\Everything.exe" -startup
[13:56:35]  [×] Get Whitelist...
[13:56:35]  [×] Added service: SDRSVC
[13:56:35]  [×] Added service: wbengine
[13:56:35]  [×] Kill System Services (telemetry, booster, etc)...
[13:56:35]  [×] Service: WSearch
[13:56:35]  [×] Service: pla
[13:56:35]  [×] Service: defragsvc
```

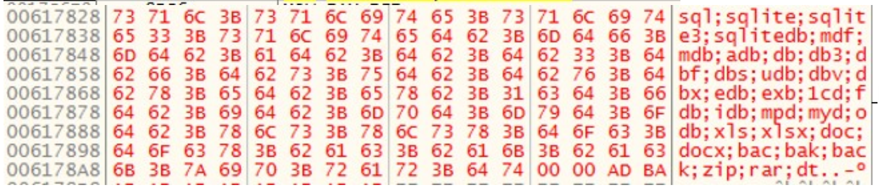
Figure 5. Sample logs that are shown when Ctrl +F1 is pressed

The ransomware's config is located at its overlay and is decrypted using the NOT Operation.



```
0017C6D0  > ┌8A07          MOV AL,BYTE PTR DS:[EDI]
0017C6D2  . │8B4E 04       MOV ECX,DWORD PTR DS:[ESI+4]
0017C6D5  . │F6D0          NOT AL
0017C6D7  . │8845 F3       MOV BYTE PTR SS:[EBP-D],AL
0017C6DA  . │394E 08       CMP DWORD PTR DS:[ESI+8],ECX
0017C6DD  .v│74 07         JE SHORT bestplac.0017C6E6
0017C6DF  . │8801          MOV BYTE PTR DS:[ECX],AL
0017C6E1  . │FF46 04       INC DWORD PTR DS:[ESI+4]
0017C6E4  .v│EB 0C         JMP SHORT bestplac.0017C6F2
0017C6E6  > │8D45 F3       LEA EAX,DWORD PTR SS:[EBP-D]
0017C6E9  . │50            PUSH EAX
0017C6EA  . │51            PUSH ECX
0017C6EB  . │8BCE          MOV ECX,ESI
0017C6ED  . │E8 0E030000   CALL bestplac.0017CA00
0017C6F2  > │43            INC EBX
0017C6F3  . │47            INC EDI
0017C6F4  . │3B5D E8       CMP EBX,DWORD PTR SS:[EBP-18]
0017C6F7  .^└75 D7         JNZ SHORT bestplac.0017C6D0
```

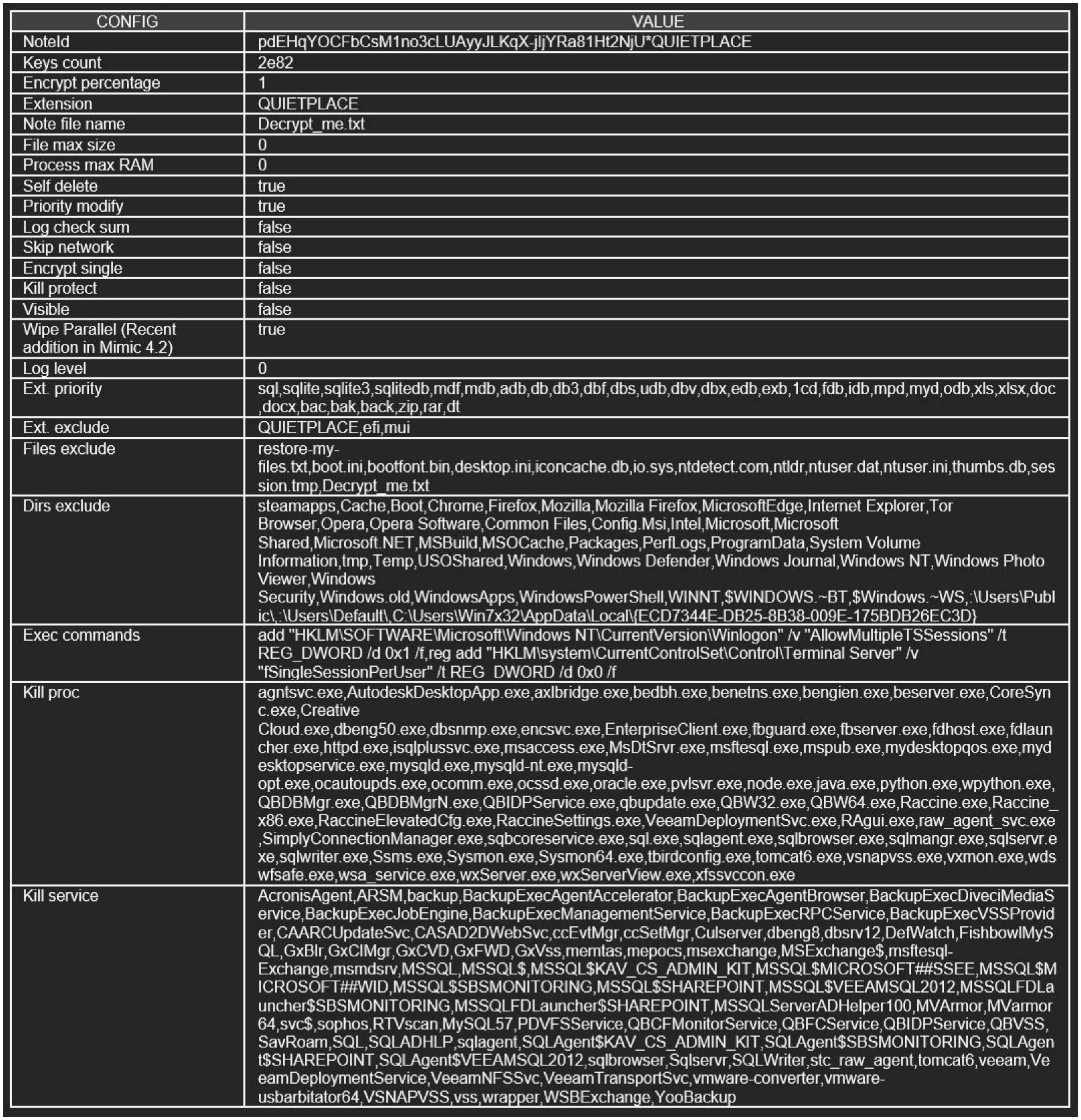
Figure 6. Decryption function for the config



```
00617828 73 71 6C 3B 73 71 6C 69 74 65 3B 73 71 6C 69 74 sql;sqlite;sqlit
00617838 65 33 3B 73 71 6C 69 74 65 64 62 3B 6D 64 66 3B e3;sqlitedb;mdf;
00617848 6D 64 62 3B 61 64 62 3B 64 62 3B 64 62 33 3B 64 mdb;adb;db;db3;d
00617858 62 66 3B 64 62 73 3B 75 64 62 3B 64 62 76 3B 64 bf;dbs;udb;dbv;d
00617868 62 78 3B 65 64 62 3B 65 78 62 3B 31 63 64 3B 66 bx;edb;exb;1cd;f
00617878 64 62 3B 69 64 62 3B 6D 70 64 3B 6D 79 64 3B 6F db;idb;mpd;myd;o
00617888 64 62 3B 78 6C 73 3B 78 6C 73 78 3B 64 6F 63 3B db;xls;xlsx;doc;
00617898 64 6F 63 78 3B 62 61 63 3B 62 61 6B 3B 62 61 63 docx;bac;bak;bac
006178A8 6B 3B 7A 69 70 3B 72 61 72 3B 64 74 00 00 AD BA k;zip;rar;dt..-°
```

Figure 7. Snippet from a decrypted config

Figure 8 shows a more thorough look at the config and its values.



CONFIG	VALUE
NoteId	pdEHqYOCFbCsM1no3cLUAyyJLKqX-jIjYRa81Ht2NjU*QUIETPLACE
Keys count	2e82
Encrypt percentage	1
Extension	QUIETPLACE
Note file name	Decrypt_me.txt
File max size	0
Process max RAM	0
Self delete	true
Priority modify	true
Log check sum	false
Skip network	false
Encrypt single	false
Kill protect	false
Visible	false
Wipe Parallel (Recent addition in Mimic 4.2)	true
Log level	0
Ext. priority	sql,sqlite,sqlite3,sqlitedb,mdf,mdb,adb,db,db3,dbf,dbs,udb,dbv,dbx,edb,exb,1cd,fdb,idb,mpd,myd,odb,xls,xlsx,doc,docx,bac,bak,back,zip,rar,dt
Ext. exclude	QUIETPLACE,efi,mui
Files exclude	restore-my-files.txt,boot.ini,bootfont.bin,desktop.ini,iconcache.db,io.sys,ntdetect.com,ntldr,ntuser.dat,ntuser.ini,thumbs.db,session.tmp,Decrypt_me.txt
Dirs exclude	steamapps,Cache,Boot,Chrome,Firefox,Mozilla,Mozilla Firefox,MicrosoftEdge,Internet Explorer,Tor Browser,Opera,Opera Software,Common Files,Config.Msi,Intel,Microsoft,Microsoft Shared,Microsoft.NET,MSBuild,MSOCache,Packages,PerfLogs,ProgramData,System Volume Information,tmp,Temp,USOShared,Windows,Windows Defender,Windows Journal,Windows NT,Windows Photo Viewer,Windows Security,Windows.old,WindowsApps,WindowsPowerShell,WINNT,$WINDOWS.~BT,$Windows.~WS,:\Users\Public\,:\Users\Default\,C:\Users\Win7x32\AppData\Local\{ECD7344E-DB25-8B38-009E-175BDB26EC3D}
Exec commands	add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon" /v "AllowMultipleTSSessions" /t REG_DWORD /d 0x1 /f,reg add "HKLM\system\CurrentControlSet\Control\Terminal Server" /v "fSingleSessionPerUser" /t REG_DWORD /d 0x0 /f
Kill proc	agntsvc.exe,AutodeskDesktopApp.exe,axlbridge.exe,bedbh.exe,benetns.exe,bengien.exe,beserver.exe,CoreSync.exe,Creative Cloud.exe,dbeng50.exe,dbsnmp.exe,encsvc.exe,EnterpriseClient.exe,fbguard.exe,fbserver.exe,fdhost.exe,fdlauncher.exe,httpd.exe,isqlplussvc.exe,msaccess.exe,MsDtSrvr.exe,msftesql.exe,mspub.exe,mydesktopqos.exe,mydesktopservice.exe,mysqld.exe,mysqld-nt.exe,mysqld-opt.exe,ocautoupds.exe,ocomm.exe,ocssd.exe,oracle.exe,pvlsvr.exe,node.exe,java.exe,python.exe,wpython.exe,QBDBMgr.exe,QBDBMgrN.exe,QBIDPService.exe,qbupdate.exe,QBW32.exe,QBW64.exe,Raccine.exe,Raccine_x86.exe,RaccineElevatedCfg.exe,RaccineSettings.exe,VeeamDeploymentSvc.exe,RAgui.exe,raw_agent_svc.exe,SimplyConnectionManager.exe,sqbcoreservice.exe,sql.exe,sqlagent.exe,sqlbrowser.exe,sqlmangr.exe,sqlservr.exe,sqlwriter.exe,Ssms.exe,Sysmon.exe,Sysmon64.exe,tbirdconfig.exe,tomcat6.exe,vsnapvss.exe,vxmon.exe,wdswfsafe.exe,wsa_service.exe,wxServer.exe,wxServerView.exe,xfssvccon.exe
Kill service	AcronisAgent,ARSM,backup,BackupExecAgentAccelerator,BackupExecAgentBrowser,BackupExecDiveciMediaService,BackupExecJobEngine,BackupExecManagementService,BackupExecRPCService,BackupExecVSSProvider,CAARCUpdateSvc,CASAD2DWebSvc,ccEvtMgr,ccSetMgr,Culserver,dbeng8,dbsrv12,DefWatch,FishbowlMySQL,GxBlr,GxCIMgr,GxCVD,GxFWD,GxVss,memtas,mepocs,msexchange,MSExchange$,msftesql-Exchange,msmdsrv,MSSQL,MSSQL$,MSSQL$KAV_CS_ADMIN_KIT,MSSQL$MICROSOFT##SSEE,MSSQL$MICROSOFT##WID,MSSQL$SBSMONITORING,MSSQL$SHAREPOINT,MSSQL$VEEAMSQL2012,MSSQLFDLauncher$SBSMONITORING,MSSQLFDLauncher$SHAREPOINT,MSSQLServerADHelper100,MVArmor,MVarmor64,svc$,sophos,RTVscan,MySQL57,PDVFSService,QBCFMonitorService,QBFCService,QBIDPService,QBVSS,SavRoam,SQL,SQLADHLP,sqlagent,SQLAgent$KAV_CS_ADMIN_KIT,SQLAgent$SBSMONITORING,SQLAgent$SHAREPOINT,SQLAgent$VEEAMSQL2012,sqlbrowser,Sqlservr,SQLWriter,stc_raw_agent,tomcat6,veeam,VeeamDeploymentService,VeeamNFSSvc,VeeamTransportSvc,vmware-converter,vmware-usbarbitator64,VSNAPVSS,vss,wrapper,WSBExchange,YooBackup

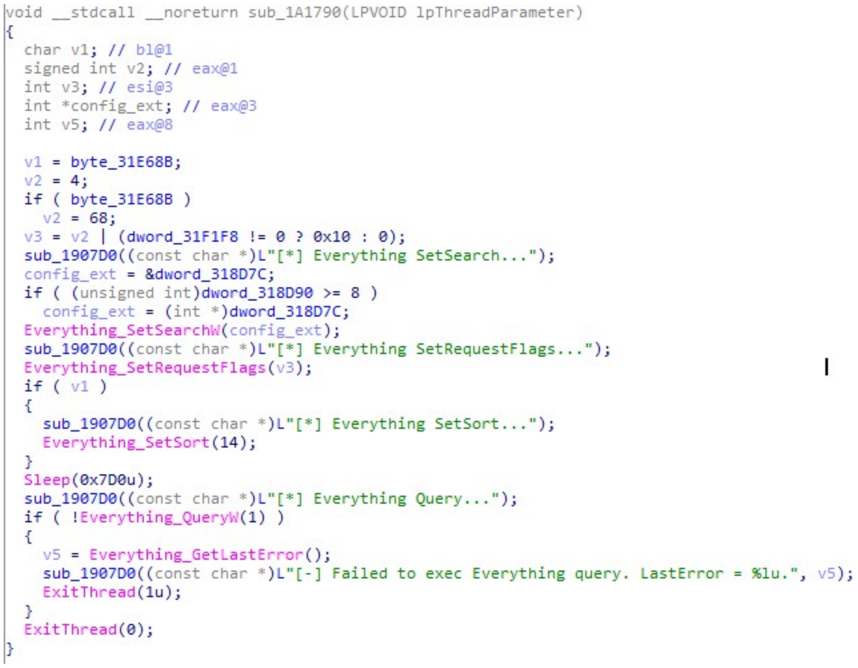
Figure 8. Mimic ransomware config details

Mimic ransomware possesses a plethora of capabilities, including the following:

- Collecting system information
- Creating persistence via the RUN key
- Bypassing User Account Control (UAC)
- Disabling Windows Defender
- Disabling Windows telemetry
- Activating anti-shutdown measures
- Activating anti-kill measures
- Unmounting Virtual Drives
- Terminating processes and services
- Disabling sleep mode and shutdown of the system
- Removing indicators
- Inhibiting System Recovery

Abusing Everything32 APIs for encryption

Mimic uses *Everything32.dll*, a legitimate Windows filename search engine that can return real time results for queries, in its routine. It abuses the tool by querying certain file extensions and filenames using Everything's APIs to retrieve the file's path for encryption.



```
void __stdcall __noreturn sub_1A1790(LPVOID lpThreadParameter)
{
  char v1; // bl@1
  signed int v2; // eax@1
  int v3; // esi@3
  int *config_ext; // eax@3
  int v5; // eax@8

  v1 = byte_31E68B;
  v2 = 4;
  if ( byte_31E68B )
    v2 = 68;
  v3 = v2 | (dword_31F1F8 != 0 ? 0x10 : 0);
  sub_1907D0((const char *)L"[*] Everything SetSearch...");
  config_ext = &dword_318D7C;
  if ( (unsigned int)dword_318D90 >= 8 )
    config_ext = (int *)dword_318D7C;
  Everything_SetSearchW(config_ext);
  sub_1907D0((const char *)L"[*] Everything SetRequestFlags...");
  Everything_SetRequestFlags(v3);
  if ( v1 )
  {
    sub_1907D0((const char *)L"[*] Everything SetSort...");
    Everything_SetSort(14);
  }
  Sleep(0x7D0u);
  sub_1907D0((const char *)L"[*] Everything Query...");
  if ( !Everything_QueryW(1) )
  {
    v5 = Everything_GetLastError();
    sub_1907D0((const char *)L"[-] Failed to exec Everything query. LastError = %lu.", v5);
    ExitThread(1u);
  }
  ExitThread(0);
}
```

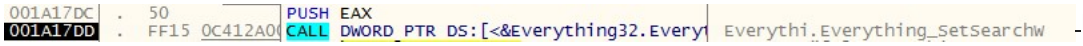
Figure 9. Overview of the function that utilizes Everything API

It uses the Everything_SetSearchW function to search for files to be encrypted or avoided using the following search format:

> *file:<ext:{list of extension}>file:<!endwith:{list of files/directory to avoid}>wholefilename<!{list of files to avoid}>*

The following query is used by Mimic to search for files to be encrypted or avoided:

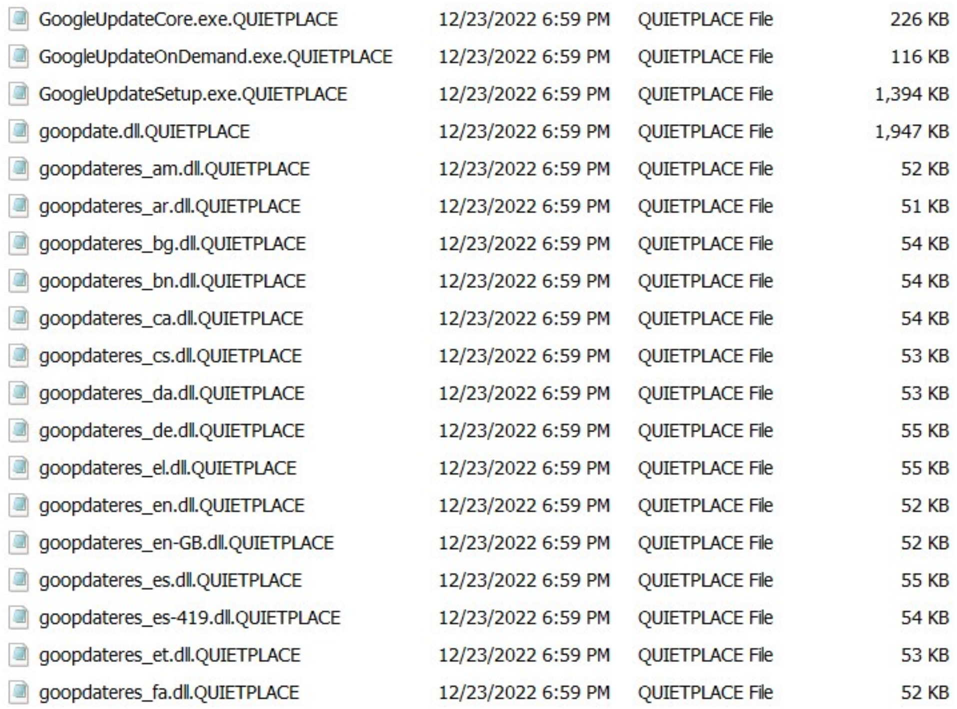
> file:
> <ext:;sql;sqlite;sqlite3;sqlitedb;mdf;mdb;adb;db;db3;dbf;dbs;udb;dbv;dbx;edb;exb;1cd;fdb;idb;mpd;myd;odb;xls;xlsx;doc;docx;bac;bak;back;zip;rar
> file:<!endwith:QUIETPLACE> <!"\steamapps\" !"\Cache\" !"\Boot\" !"\Chrome\" !"\Firefox\" !"\Mozilla\" !"\Mozilla Firefox\" !"\MicrosoftEdge\" !"\Interr
> Explorer\" !"\Tor Browser\" !"\Opera\" !"\Opera Software\" !"\Common Files\" !"\Config.Msi\" !"\Intel\" !"\Microsoft\" !"\Microsoft Shared\"
> !"\Microsoft.NET\" !"\MSBuild\" !"\MSOCache\" !"\Packages\" !"\PerfLogs\" !"\ProgramData\" !"\System Volume Information\" !"\tmp\" !"\Temp\"
> !"\USOShared\" !"\Windows\" !"\Windows Defender\" !"\Windows Journal\" !"\Windows NT\" !"\Windows Photo Viewer\" !"\Windows Security\"
> !"\Windows.old\" !"\WindowsApps\" !"\WindowsPowerShell\" !"\WINNT\" !"\$WINDOWS.~BT\" !"\$Windows.~WS\" !":\Users\Public\" !":\Users\Defa
> !"C:\Users\Win7x32\AppData\Local\{ECD7344E-DB25-8B38-009E-175BDB26EC3D}" !"NTUSER.DAT"> wholefilename:<!"restore-my-files.txt"
> !"boot.ini" !"bootfont.bin" !"desktop.ini" !"iconcache.db" !"io.sys" !"ntdetect.com" !"ntldr" !"ntuser.dat" !"ntuser.ini" !"thumbs.db" !"session.tmp"
> !"Decrypt_me.txt"> <!size:0>



```
001A17DC  .  50             PUSH EAX
001A17DD  .  FF15 0C412A0(  CALL DWORD PTR DS:[<&Everything32.Every   Everythi.Everything_SetSearchW
```

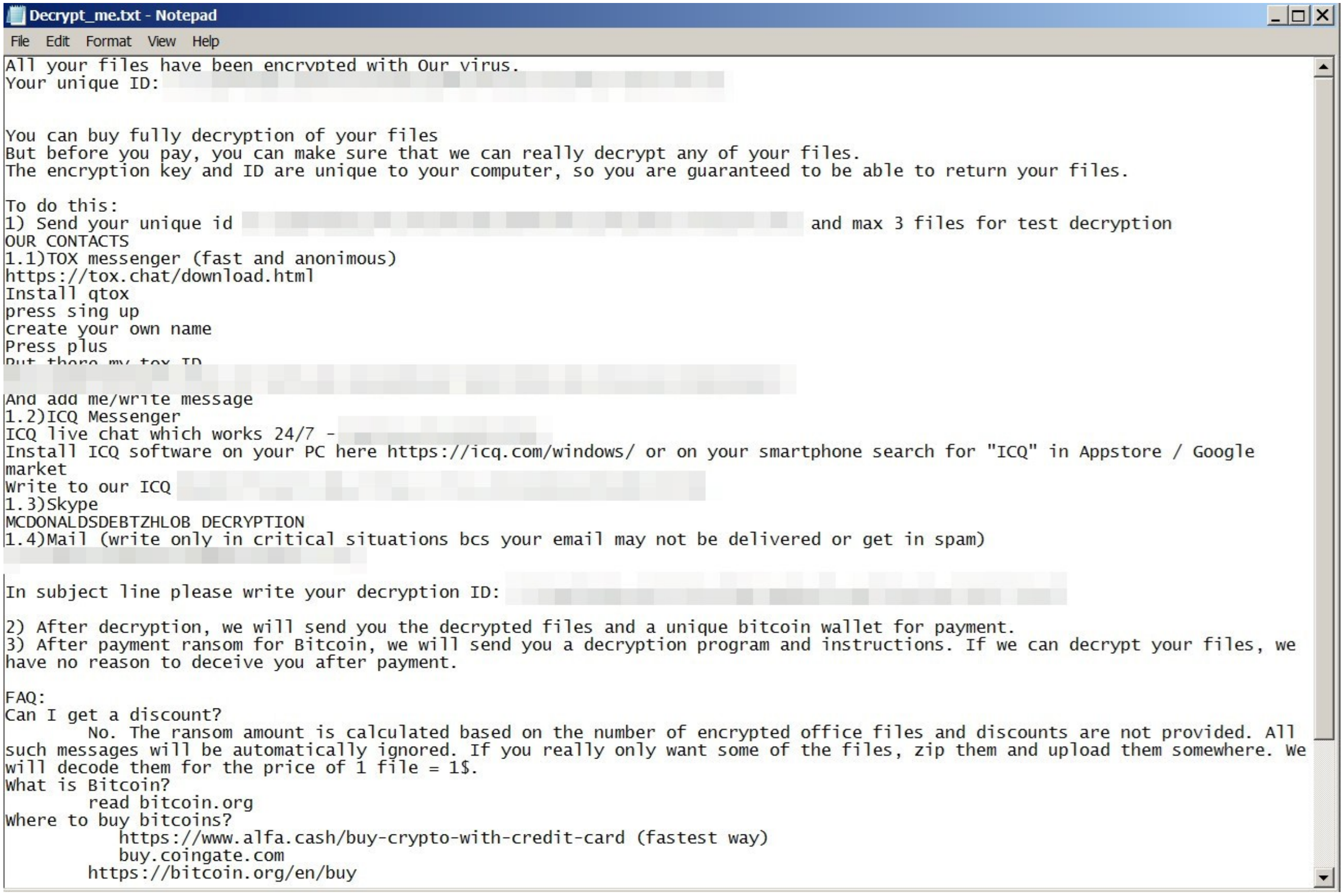
Figure 10. The Everything_SetSearchW API used by Mimic ransomware

It then appends the .QUIETPLACE file extension to the encrypted files and, finally, displays the ransom note.



GoogleUpdateCore.exe.QUIETPLACE	12/23/2022 6:59 PM	QUIETPLACE File	226 KB
GoogleUpdateOnDemand.exe.QUIETPLACE	12/23/2022 6:59 PM	QUIETPLACE File	116 KB
GoogleUpdateSetup.exe.QUIETPLACE	12/23/2022 6:59 PM	QUIETPLACE File	1,394 KB
goopdate.dll.QUIETPLACE	12/23/2022 6:59 PM	QUIETPLACE File	1,947 KB
goopdateres_am.dll.QUIETPLACE	12/23/2022 6:59 PM	QUIETPLACE File	52 KB
goopdateres_ar.dll.QUIETPLACE	12/23/2022 6:59 PM	QUIETPLACE File	51 KB
goopdateres_bg.dll.QUIETPLACE	12/23/2022 6:59 PM	QUIETPLACE File	54 KB
goopdateres_bn.dll.QUIETPLACE	12/23/2022 6:59 PM	QUIETPLACE File	54 KB
goopdateres_ca.dll.QUIETPLACE	12/23/2022 6:59 PM	QUIETPLACE File	54 KB
goopdateres_cs.dll.QUIETPLACE	12/23/2022 6:59 PM	QUIETPLACE File	53 KB
goopdateres_da.dll.QUIETPLACE	12/23/2022 6:59 PM	QUIETPLACE File	53 KB
goopdateres_de.dll.QUIETPLACE	12/23/2022 6:59 PM	QUIETPLACE File	55 KB
goopdateres_el.dll.QUIETPLACE	12/23/2022 6:59 PM	QUIETPLACE File	55 KB
goopdateres_en.dll.QUIETPLACE	12/23/2022 6:59 PM	QUIETPLACE File	52 KB
goopdateres_en-GB.dll.QUIETPLACE	12/23/2022 6:59 PM	QUIETPLACE File	52 KB
goopdateres_es.dll.QUIETPLACE	12/23/2022 6:59 PM	QUIETPLACE File	55 KB
goopdateres_es-419.dll.QUIETPLACE	12/23/2022 6:59 PM	QUIETPLACE File	54 KB
goopdateres_et.dll.QUIETPLACE	12/23/2022 6:59 PM	QUIETPLACE File	53 KB
goopdateres_fa.dll.QUIETPLACE	12/23/2022 6:59 PM	QUIETPLACE File	52 KB

Figure 11. Files that were encrypted by the Mimic ransomware

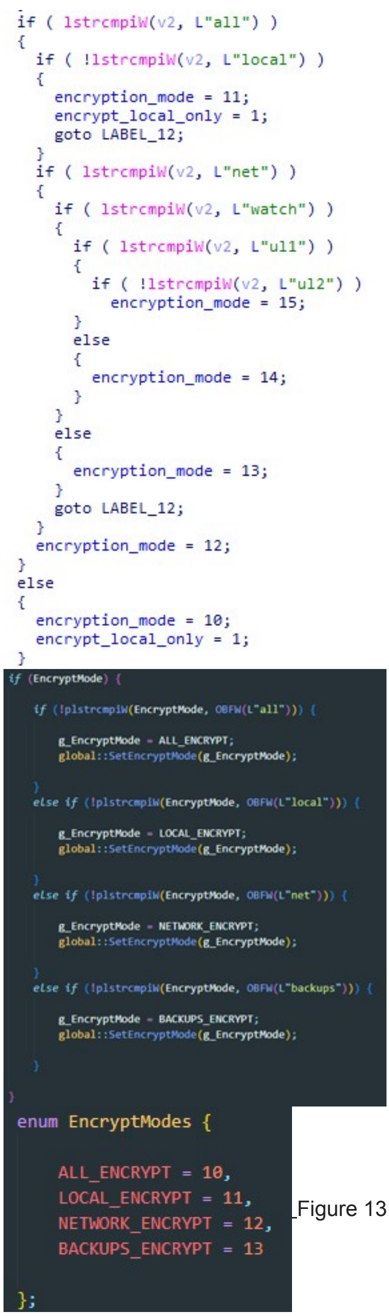


```
All your files have been encrypted with Our virus.
Your unique ID: [redacted]


You can buy fully decryption of your files
But before you pay, you can make sure that we can really decrypt any of your files.
The encryption key and ID are unique to your computer, so you are guaranteed to be able to return your files.

To do this:
1) Send your unique id [redacted] and max 3 files for test decryption
OUR CONTACTS
1.1)TOX messenger (fast and anonimous)
https://tox.chat/download.html
Install qtox
press sing up
create your own name
Press plus
Put there my tox ID
[redacted]
And add me/write message
1.2)ICQ Messenger
ICQ live chat which works 24/7 - [redacted]
Install ICQ software on your PC here https://icq.com/windows/ or on your smartphone search for "ICQ" in Appstore / Google
market
Write to our ICQ [redacted]
1.3)Skype
MCDONALDSDEBTZHLOB DECRYPTION
1.4)Mail (write only in critical situations bcs your email may not be delivered or get in spam)
[redacted]

In subject line please write your decryption ID: [redacted]

2) After decryption, we will send you the decrypted files and a unique bitcoin wallet for payment.
3) After payment ransom for Bitcoin, we will send you a decryption program and instructions. If we can decrypt your files, we
have no reason to deceive you after payment.

FAQ:
Can I get a discount?
        No. The ransom amount is calculated based on the number of encrypted office files and discounts are not provided. All
such messages will be automatically ignored. If you really only want some of the files, zip them and upload them somewhere. We
will decode them for the price of 1 file = 1$.
What is Bitcoin?
        read bitcoin.org
Where to buy bitcoins?
           https://www.alfa.cash/buy-crypto-with-credit-card (fastest way)
           buy.coingate.com
        https://bitcoin.org/en/buy
```

Figure 12. The Mimic ransom note

Code from leaked Conti builder

From our analysis, some parts of the code seemed to be based on, and share several similarities with the Conti ransomware builder that was leaked in March 2022. For example, the enumeration of the encryption modes shares the same integer for both Mimic and Conti.



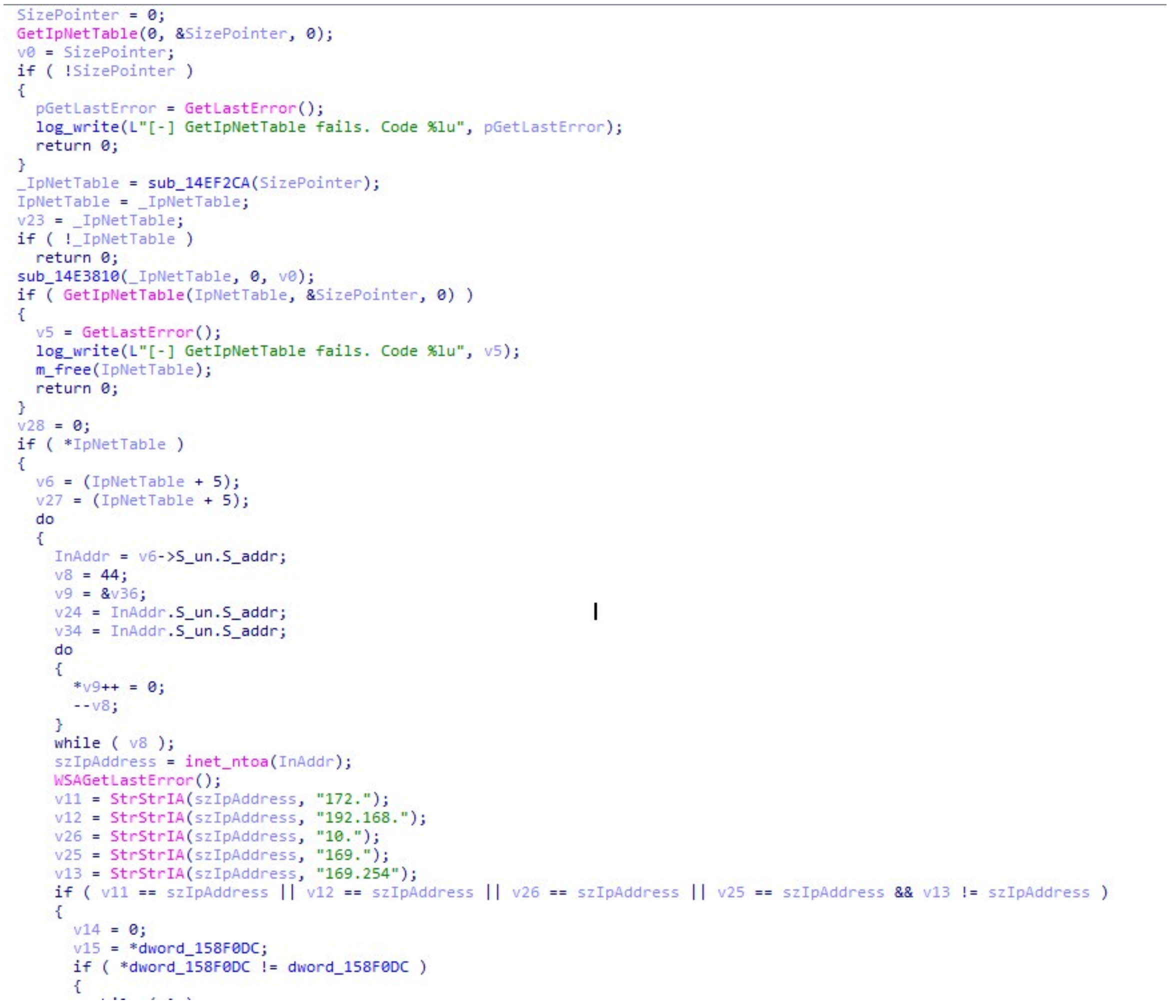
```
if ( lstrcmpiW(v2, L"all") )
{
  if ( !lstrcmpiW(v2, L"local") )
  {
    encryption_mode = 11;
    encrypt_local_only = 1;
    goto LABEL_12;
  }
  if ( lstrcmpiW(v2, L"net") )
  {
    if ( lstrcmpiW(v2, L"watch") )
    {
      if ( lstrcmpiW(v2, L"ul1") )
      {
        if ( !lstrcmpiW(v2, L"ul2") )
          encryption_mode = 15;
      }
      else
      {
        encryption_mode = 14;
      }
    }
    else
    {
      encryption_mode = 13;
    }
    goto LABEL_12;
  }
  encryption_mode = 12;
}
else
{
  encryption_mode = 10;
  encrypt_local_only = 1;
}
```

```
if (EncryptMode) {

    if (!plstrcmpiW(EncryptMode, OBFW(L"all"))) {

        g_EncryptMode = ALL_ENCRYPT;
        global::SetEncryptMode(g_EncryptMode);

    }
    else if (!plstrcmpiW(EncryptMode, OBFW(L"local"))) {

        g_EncryptMode = LOCAL_ENCRYPT;
        global::SetEncryptMode(g_EncryptMode);

    }
    else if (!plstrcmpiW(EncryptMode, OBFW(L"net"))) {

        g_EncryptMode = NETWORK_ENCRYPT;
        global::SetEncryptMode(g_EncryptMode);

    }
    else if (!plstrcmpiW(EncryptMode, OBFW(L"backups"))) {

        g_EncryptMode = BACKUPS_ENCRYPT;
        global::SetEncryptMode(g_EncryptMode);

    }

}
```

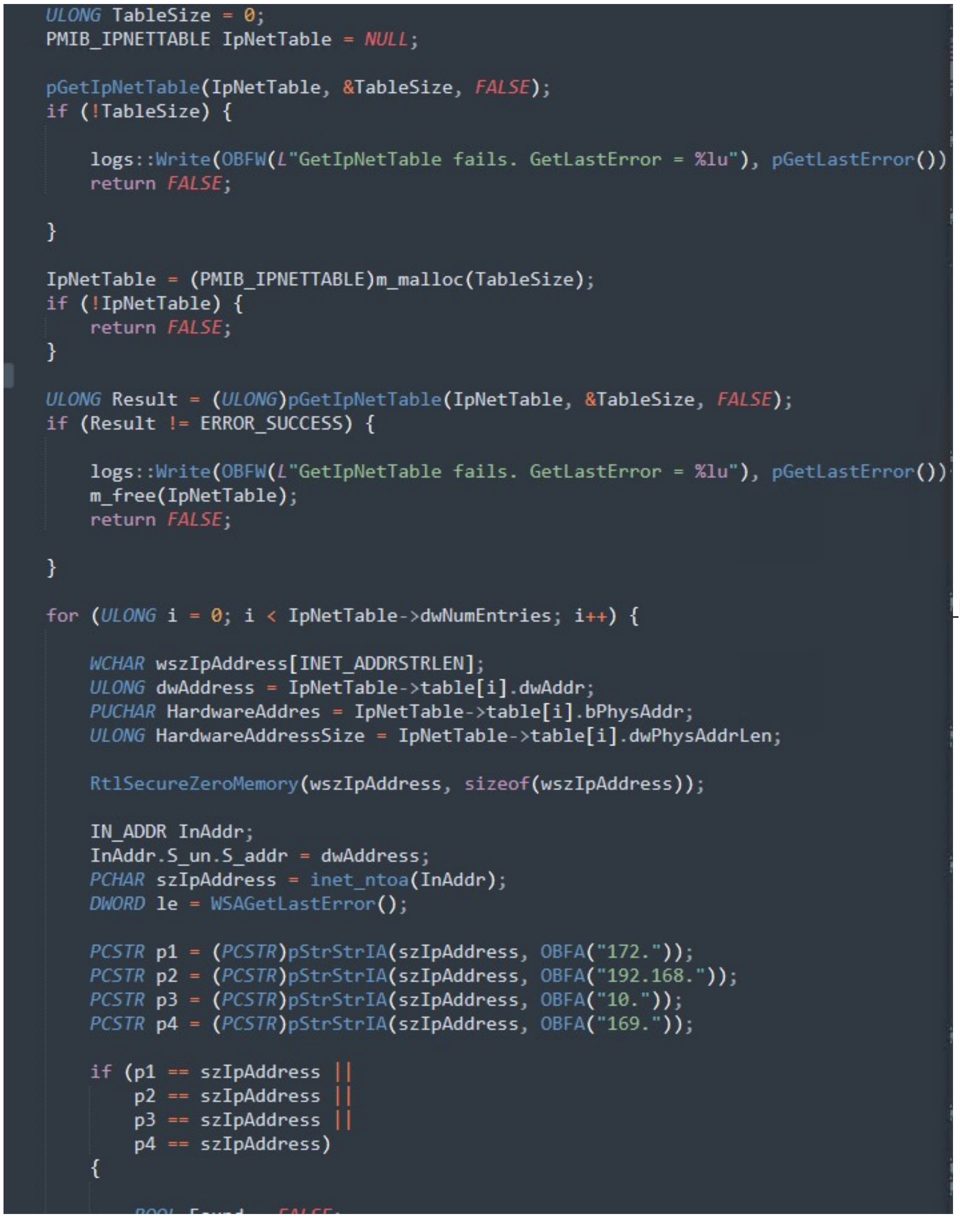
```
enum EncryptModes {

    ALL_ENCRYPT = 10,
    LOCAL_ENCRYPT = 11,
    NETWORK_ENCRYPT = 12,
    BACKUPS_ENCRYPT = 13

};
```

Figure 13. Similarities between Mimic (top) and the leaked Conti builder (bottom)

The code related to argument **net** is also based on Conti. It will use the GetIpNetTable function to read the Address Resolution Protocol (ARP) cache and check if IP addresses contain “172.”, “192.168”, “10.”, or “169.” Mimic added a filter to exclude IP addresses that contain “169.254”, which is the IP range of Automatic Private IP Addressing (APIPA).



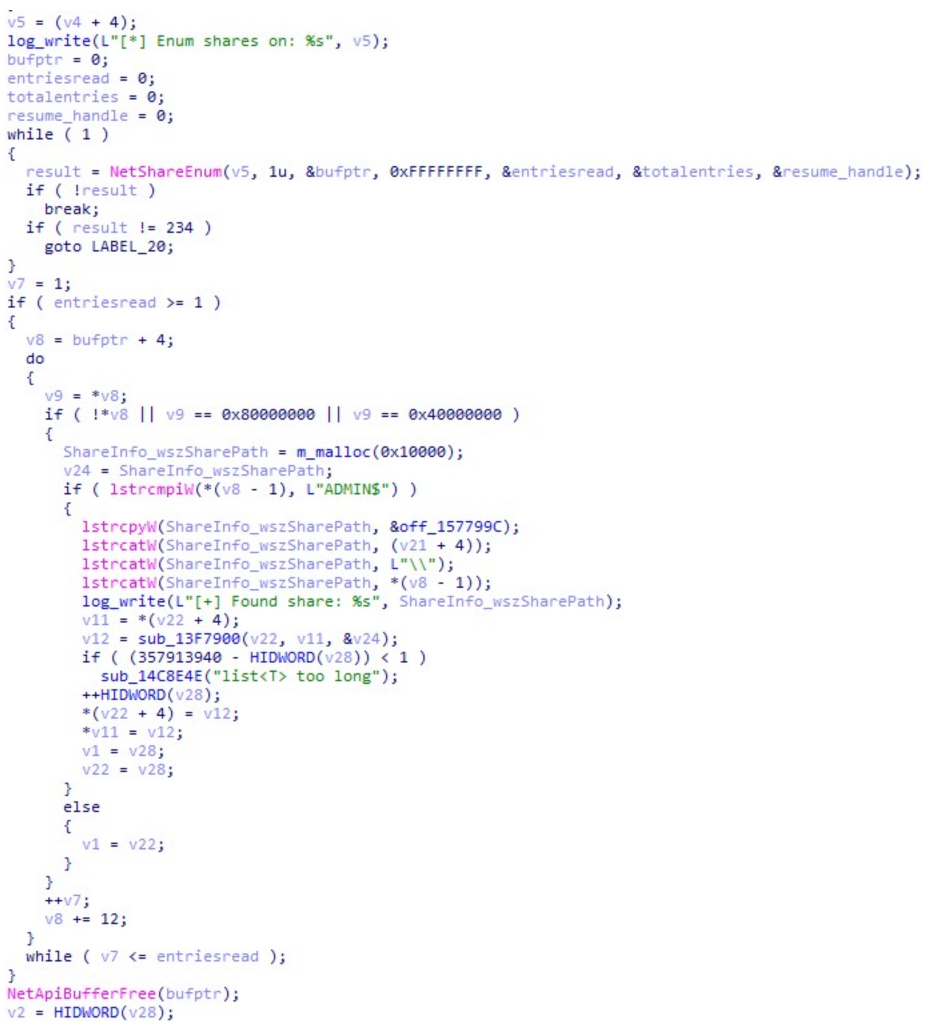
```
SizePointer = 0;
GetIpNetTable(0, &SizePointer, 0);
v0 = SizePointer;
if ( !SizePointer )
{
  pGetLastError = GetLastError();
  log_write(L"[-] GetIpNetTable fails. Code %lu", pGetLastError);
  return 0;
}
_IpNetTable = sub_14EF2CA(SizePointer);
IpNetTable = _IpNetTable;
v23 = _IpNetTable;
if ( !_IpNetTable )
  return 0;
sub_14E3810(_IpNetTable, 0, v0);
if ( GetIpNetTable(IpNetTable, &SizePointer, 0) )
{
  v5 = GetLastError();
  log_write(L"[-] GetIpNetTable fails. Code %lu", v5);
  m_free(IpNetTable);
  return 0;
}
v28 = 0;
if ( *IpNetTable )
{
  v6 = (IpNetTable + 5);
  v27 = (IpNetTable + 5);
  do
  {
    InAddr = v6->S_un.S_addr;
    v8 = 44;
    v9 = &v36;
    v24 = InAddr.S_un.S_addr;
    v34 = InAddr.S_un.S_addr;
    do
    {
      *v9++ = 0;
      --v8;
    }
    while ( v8 );
    szIpAddress = inet_ntoa(InAddr);
    WSAGetLastError();
    v11 = StrStrIA(szIpAddress, "172.");
    v12 = StrStrIA(szIpAddress, "192.168.");
    v26 = StrStrIA(szIpAddress, "10.");
    v25 = StrStrIA(szIpAddress, "169.");
    v13 = StrStrIA(szIpAddress, "169.254");
    if ( v11 == szIpAddress || v12 == szIpAddress || v26 == szIpAddress || v25 == szIpAddress && v13 != szIpAddress )
    {
      v14 = 0;
      v15 = *dword_158F0DC;
      if ( *dword_158F0DC != dword_158F0DC )
      {
```



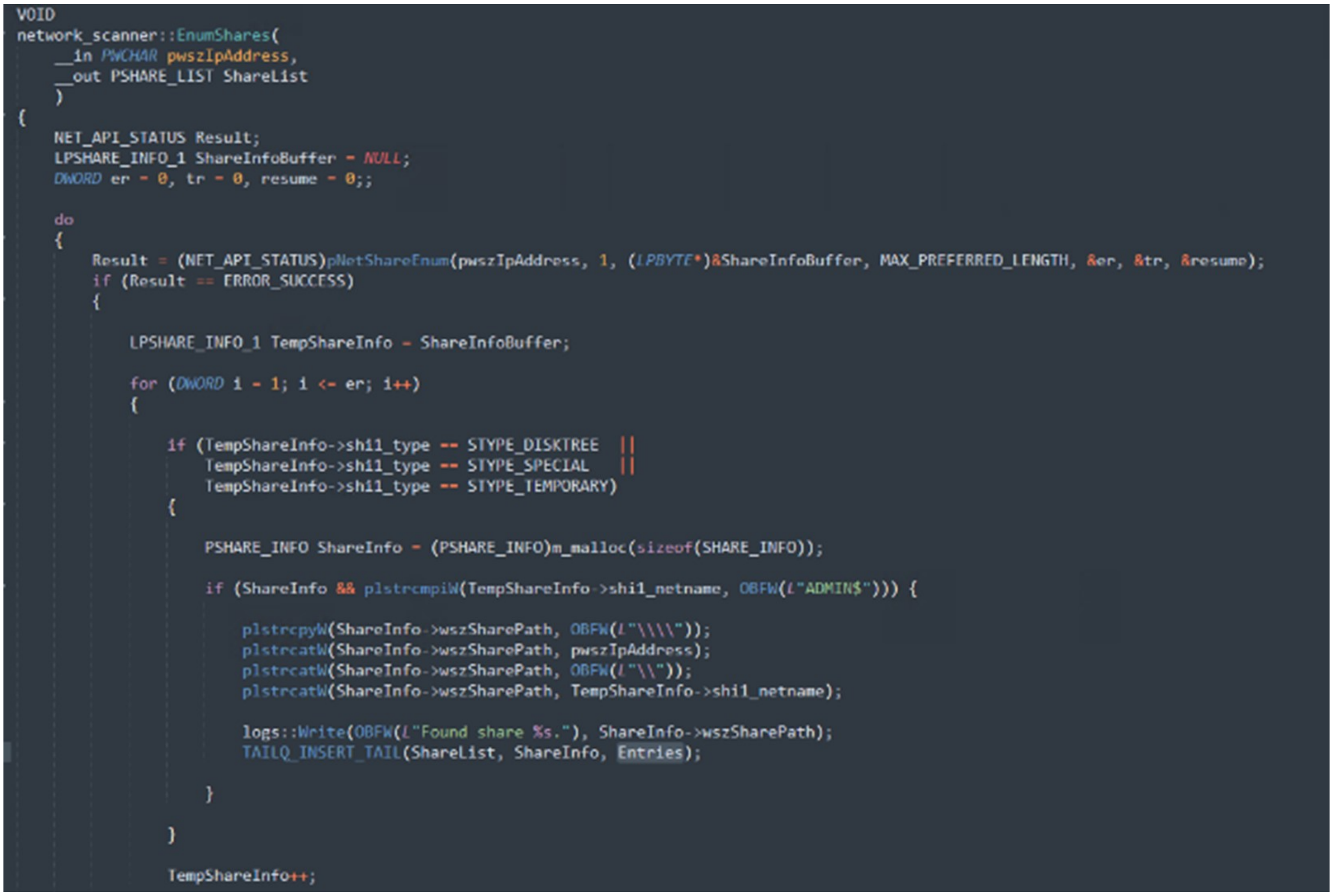
```
ULONG TableSize = 0;
PMIB_IPNETTABLE IpNetTable = NULL;

pGetIpNetTable(IpNetTable, &TableSize, FALSE);
if (!TableSize) {

    logs::Write(OBFW(L"GetIpNetTable fails. GetLastError = %lu"), pGetLastError())
    return FALSE;

}

IpNetTable = (PMIB_IPNETTABLE)m_malloc(TableSize);
if (!IpNetTable) {
    return FALSE;
}

ULONG Result = (ULONG)pGetIpNetTable(IpNetTable, &TableSize, FALSE);
if (Result != ERROR_SUCCESS) {

    logs::Write(OBFW(L"GetIpNetTable fails. GetLastError = %lu"), pGetLastError())
    m_free(IpNetTable);
    return FALSE;

}

for (ULONG i = 0; i < IpNetTable->dwNumEntries; i++) {

    WCHAR wszIpAddress[INET_ADDRSTRLEN];
    ULONG dwAddress = IpNetTable->table[i].dwAddr;
    PUCHAR HardwareAddres = IpNetTable->table[i].bPhysAddr;
    ULONG HardwareAddressSize = IpNetTable->table[i].dwPhysAddrLen;

    RtlSecureZeroMemory(wszIpAddress, sizeof(wszIpAddress));

    IN_ADDR InAddr;
    InAddr.S_un.S_addr = dwAddress;
    PCHAR szIpAddress = inet_ntoa(InAddr);
    DWORD le = WSAGetLastError();

    PCSTR p1 = (PCSTR)pStrStrIA(szIpAddress, OBFA("172."));
    PCSTR p2 = (PCSTR)pStrStrIA(szIpAddress, OBFA("192.168."));
    PCSTR p3 = (PCSTR)pStrStrIA(szIpAddress, OBFA("10."));
    PCSTR p4 = (PCSTR)pStrStrIA(szIpAddress, OBFA("169."));

    if (p1 == szIpAddress ||
        p2 == szIpAddress ||
        p3 == szIpAddress ||
        p4 == szIpAddress)
    {

        BOOL Found = FALSE;
```

Figure 14. Comparison of the Mimic (top) and the leaked Conti builder (bottom) “net” argument

Mimic also uses the Conti code in Windows Share Enumeration, where it employs the NetShareEnum function to enumerate all shares on the gathered IP addresses.



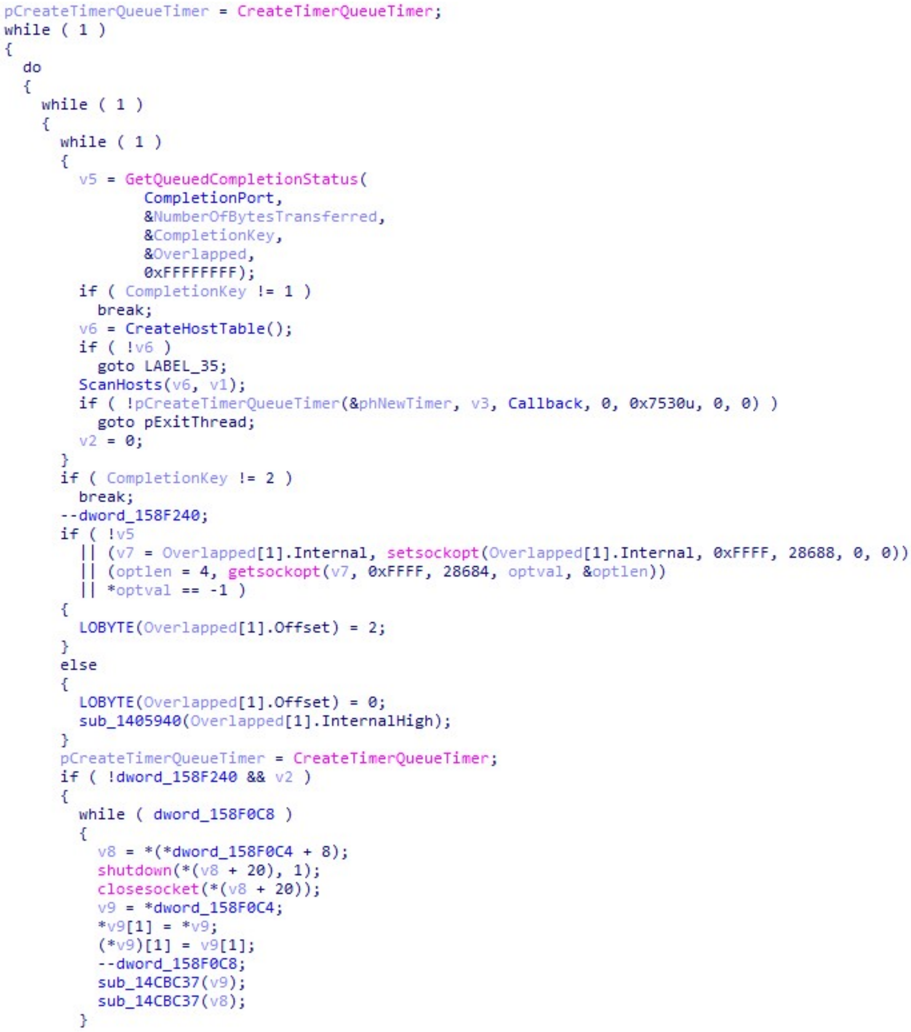
```
v5 = (v4 + 4);
log_write(L"[*] Enum shares on: %s", v5);
bufptr = 0;
entriesread = 0;
totalentries = 0;
resume_handle = 0;
while ( 1 )
{
  result = NetShareEnum(v5, 1u, &bufptr, 0xFFFFFFFF, &entriesread, &totalentries, &resume_handle);
  if ( !result )
    break;
  if ( result != 234 )
    goto LABEL_20;
}
v7 = 1;
if ( entriesread >= 1 )
{
  v8 = bufptr + 4;
  do
  {
    v9 = *v8;
    if ( !*v8 || v9 == 0x80000000 || v9 == 0x40000000 )
    {
      ShareInfo_wszSharePath = m_malloc(0x10000);
      v24 = ShareInfo_wszSharePath;
      if ( lstrcmpiW(*(v8 - 1), L"ADMIN$") )
      {
        lstrcpyW(ShareInfo_wszSharePath, &off_157799C);
        lstrcatW(ShareInfo_wszSharePath, (v21 + 4));
        lstrcatW(ShareInfo_wszSharePath, L"\\");
        lstrcatW(ShareInfo_wszSharePath, *(v8 - 1));
        log_write(L"[+] Found share: %s", ShareInfo_wszSharePath);
        v11 = *(v22 + 4);
        v12 = sub_13F7900(v22, v11, &v24);
        if ( (357913940 - HIDWORD(v28)) < 1 )
          sub_14C8E4E("list<T> too long");
        ++HIDWORD(v28);
        *(v22 + 4) = v12;
        *v11 = v12;
        v1 = v28;
        v22 = v28;
      }
      else
      {
        v1 = v22;
      }
    }
    ++v7;
    v8 += 12;
  }
  while ( v7 <= entriesread );
}
NetApiBufferFree(bufptr);
v2 = HIDWORD(v28);
```



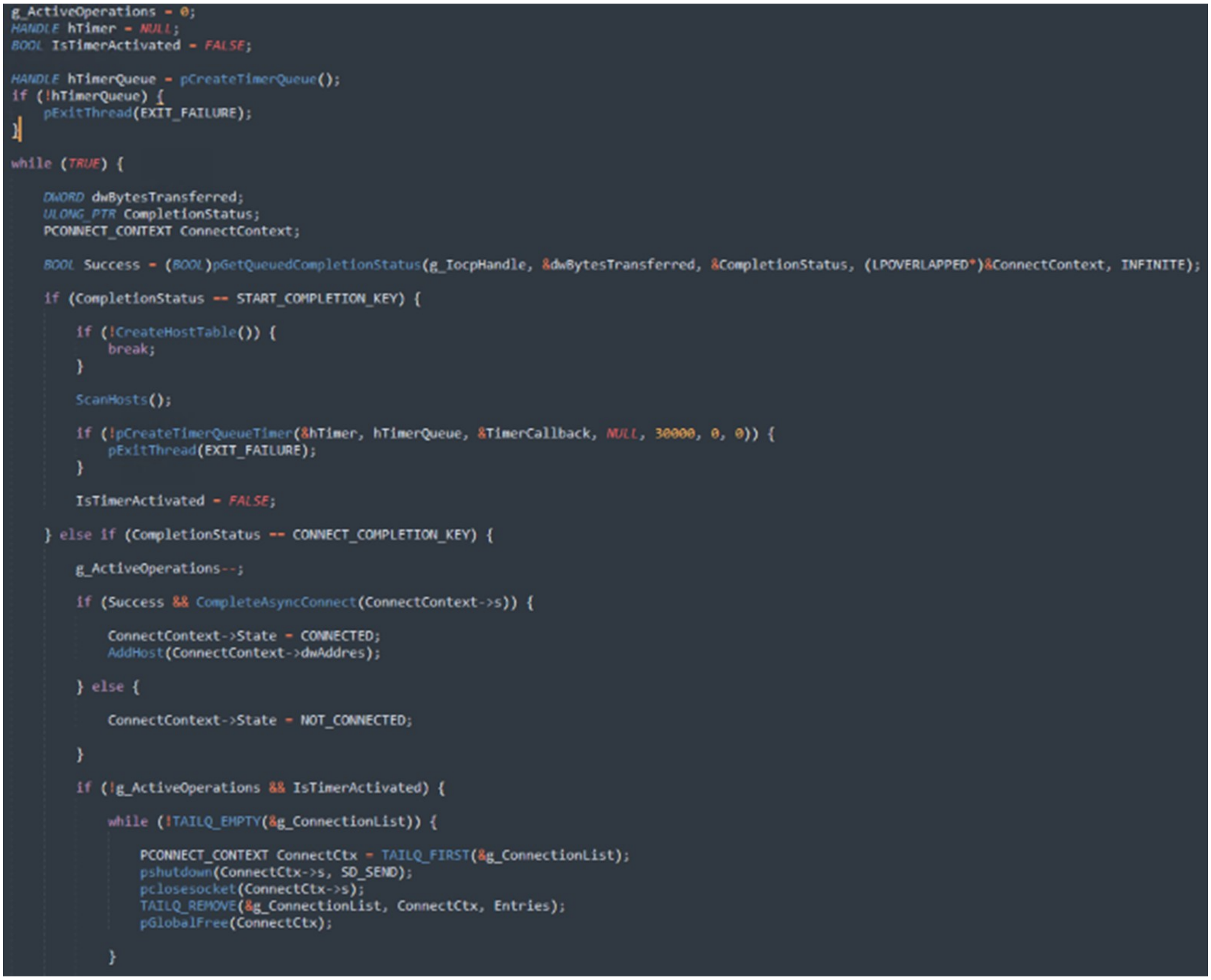
```
VOID
network_scanner::EnumShares(
    __in PWCHAR pwszIpAddress,
    __out PSHARE_LIST ShareList
    )
{
    NET_API_STATUS Result;
    LPSHARE_INFO_1 ShareInfoBuffer = NULL;
    DWORD er = 0, tr = 0, resume = 0;;

    do
    {
        Result = (NET_API_STATUS)pNetShareEnum(pwszIpAddress, 1, (LPBYTE*)&ShareInfoBuffer, MAX_PREFERRED_LENGTH, &er, &tr, &resume);
        if (Result == ERROR_SUCCESS)
        {

            LPSHARE_INFO_1 TempShareInfo = ShareInfoBuffer;

            for (DWORD i = 1; i <= er; i++)
            {

                if (TempShareInfo->shi1_type == STYPE_DISKTREE  ||
                    TempShareInfo->shi1_type == STYPE_SPECIAL   ||
                    TempShareInfo->shi1_type == STYPE_TEMPORARY)
                {

                    PSHARE_INFO ShareInfo = (PSHARE_INFO)m_malloc(sizeof(SHARE_INFO));

                    if (ShareInfo && plstrcmpiW(TempShareInfo->shi1_netname, OBFW(L"ADMIN$"))) {

                        plstrcpyW(ShareInfo->wszSharePath, OBFW(L"\\\\"));
                        plstrcatW(ShareInfo->wszSharePath, pwszIpAddress);
                        plstrcatW(ShareInfo->wszSharePath, OBFW(L"\\"));
                        plstrcatW(ShareInfo->wszSharePath, TempShareInfo->shi1_netname);

                        logs::Write(OBFW(L"Found share %s."), ShareInfo->wszSharePath);
                        TAILQ_INSERT_TAIL(ShareList, ShareInfo, Entries);

                    }

                }

                TempShareInfo++;
```

Figure 15. Comparison of the Mimic (top) and the leaked Conti (bottom) Share Enumeration function

Finally, Mimic’s port scanning is also based on the Conti builder.



```
pCreateTimerQueueTimer = CreateTimerQueueTimer;
while ( 1 )
{
  do
  {
    while ( 1 )
    {
      while ( 1 )
      {
        v5 = GetQueuedCompletionStatus(
               CompletionPort,
               &NumberOfBytesTransferred,
               &CompletionKey,
               &Overlapped,
               0xFFFFFFFF);
        if ( CompletionKey != 1 )
          break;
        v6 = CreateHostTable();
        if ( !v6 )
          goto LABEL_35;
        ScanHosts(v6, v1);
        if ( !pCreateTimerQueueTimer(&phNewTimer, v3, Callback, 0, 0x7530u, 0, 0) )
          goto pExitThread;
        v2 = 0;
      }
      if ( CompletionKey != 2 )
        break;
      --dword_158F240;
      if ( !v5
        || (v7 = Overlapped[1].Internal, setsockopt(Overlapped[1].Internal, 0xFFFF, 28688, 0, 0))
        || (optlen = 4, getsockopt(v7, 0xFFFF, 28684, optval, &optlen))
        || *optval == -1 )
      {
        LOBYTE(Overlapped[1].Offset) = 2;
      }
      else
      {
        LOBYTE(Overlapped[1].Offset) = 0;
        sub_1405940(Overlapped[1].InternalHigh);
      }
      pCreateTimerQueueTimer = CreateTimerQueueTimer;
      if ( !dword_158F240 && v2 )
      {
        while ( dword_158F0C8 )
        {
          v8 = *(*dword_158F0C4 + 8);
          shutdown(*(v8 + 20), 1);
          closesocket(*(v8 + 20));
          v9 = *dword_158F0C4;
          *v9[1] = *v9;
          (*v9)[1] = v9[1];
          --dword_158F0C8;
          sub_14CBC37(v9);
          sub_14CBC37(v8);
        }
```



```
g_ActiveOperations = 0;
HANDLE hTimer = NULL;
BOOL IsTimerActivated = FALSE;

HANDLE hTimerQueue = pCreateTimerQueue();
if (!hTimerQueue) {
    pExitThread(EXIT_FAILURE);
}

while (TRUE) {

    DWORD dwBytesTransferred;
    ULONG_PTR CompletionStatus;
    PCONNECT_CONTEXT ConnectContext;

    BOOL Success = (BOOL)pGetQueuedCompletionStatus(g_IocpHandle, &dwBytesTransferred, &CompletionStatus, (LPOVERLAPPED*)&ConnectContext, INFINITE);

    if (CompletionStatus == START_COMPLETION_KEY) {

        if (!CreateHostTable()) {
            break;
        }

        ScanHosts();

        if (!pCreateTimerQueueTimer(&hTimer, hTimerQueue, &TimerCallback, NULL, 30000, 0, 0)) {
            pExitThread(EXIT_FAILURE);
        }

        IsTimerActivated = FALSE;

    } else if (CompletionStatus == CONNECT_COMPLETION_KEY) {

        g_ActiveOperations--;

        if (Success && CompleteAsyncConnect(ConnectContext->s)) {

            ConnectContext->State = CONNECTED;
            AddHost(ConnectContext->dwAddres);

        } else {

            ConnectContext->State = NOT_CONNECTED;

        }

        if (!g_ActiveOperations && IsTimerActivated) {

            while (!TAILQ_EMPTY(&g_ConnectionList)) {

                PCONNECT_CONTEXT ConnectCtx = TAILQ_FIRST(&g_ConnectionList);
                pshutdown(ConnectCtx->s, SD_SEND);
                pclosesocket(ConnectCtx->s);
                TAILQ_REMOVE(&g_ConnectionList, ConnectCtx, Entries);
                pGlobalFree(ConnectCtx);

            }
```

Figure 16. Comparison of the Mimic (top) and leaked Conti builder (bottom) port scanning function
More information about the behavior of Mimic ransomware can be found in this report.

Conclusion

Mimic ransomware, with its multiple bundled capabilities, seems to implement a new approach to speeding up its routine by combining multiple running threads and abusing Everything's APIs for its encryption (minimizing resource usage, therefore resulting in more efficient execution). Furthermore, the threat actor behind Mimic seems to be resourceful and technically adept, using a leaked ransomware builder to capitalize on its various features, and even improve on it for more effective attacks.

To protect systems from ransomware attacks, we recommend that both individual users and organizations implement best practices such as applying data protection, backup, and recovery measures to secure data from possible encryption or erasure. Conducting regular vulnerability assessments and patching systems in a timely manner can also minimize the damage dealt by ransomware that abuse exploits.

A multilayered approach can help organizations guard possible entry points into the system (endpoint, email, web, and network). The right security solutions can also detect malicious components and suspicious behavior to protect enterprises.

- Trend Micro Vision One™ provides multilayered protection and behavior detection, which helps block questionable behavior and tools early on before the ransomware can do irreversible damage to the system.
- Trend Micro Cloud One™ Workload Security protects systems against both known and unknown threats that exploit vulnerabilities. This protection is made possible through techniques such as virtual patching and machine learning.
- Trend Micro™ Deep Discovery™ Email Inspector employs custom sandboxing and advanced analysis techniques to effectively block malicious emails, including phishing emails that can serve as entry points for ransomware.
- Trend Micro Apex One™ offers next-level automated threat detection and response against advanced concerns such as fileless threats and ransomware, ensuring the protection of endpoints.

Indicators of Compromise

SHA-256	Version	Detection name
08f8ae7f25949a742c7896cb76e37fb88c6a7a32398693ec6c2b3d9b488114be	1.1	Ransom.Win32.MIMIC.SMZTJJ-A
9c16211296f88e12538792124b62eb00830d0961e9ab24b825edb61bda8f564f	1.13	Ransom.Win32.MIMIC.SMZTJJ-A
e67d3682910cf1e7ece356860179ada8e847637a86c1e5f6898c48c956f04590	1.14	Ransom.Win32.MIMIC.THLBGBB
c634378691a675acbf57e611b220e676eb19aa190f617c41a56f43ac48ae14c7	3	Ransom.Win32.MIMIC.THLBGBB
c71ce482cf50d59c92cfb1eae560711d47600541b2835182d6e46e0de302ca6c	3	Ransom.Win32.MIMIC.THLBGBB
7ae4c5caf6cda7fa8862f64a74bd7f821b50d855d6403bde7bcbd7398b2c7d99	3.3	Ransom.Win32.MIMIC.THHAABB
a1eeeeae0eb365ff9a00717846c4806785d55ed20f3f5cbf71cf6710d7913c51	3.3	Ransom.Win32.MIMIC.SMZTJJ-A
b0c75e92e1fe98715f90b29475de998d0c8c50ca80ce1c141fc09d10a7b8e7ee	3.3	Ransom.Win32.MIMIC.SMZTJJ-A
1dea642abe3e27fd91c3db4e0293fb1f7510e14aed73e4ea36bf7299fd8e6506	3.4	Ransom.Win32.MIMIC.SMZTJJ-A
4a6f8bf2b989fa60daa6c720b2d388651dd8e4c60d0be04aaed4de0c3c064c8f	3.4	Ransom.Win32.MIMIC.THLBGBB
b68f469ed8d9deea15af325efc1a56ca8cb5c2b42f2423837a51160456ce0db5	3.4	Ransom.Win32.MIMIC.SMZTJJ-A
bb28adc32ff1b9dcfaac6b7017b4896d2807b48080f9e6720afde3f89d69676c	3.4	Ransom.Win32.MIMIC.SMZTJJ-A
bf6fa9b06115a8a4ff3982427ddc12215bd1a3d759ac84895b5fb66eaa568bff	3.4	Ransom.Win32.MIMIC.SMZTJJ-A
ed6cf30ee11b169a65c2a27c4178c5a07ff3515daa339033bf83041faa6f49c1	3.4	Ransom.Win32.MIMIC.THLBGBB
480fb2f6bcb1f394dc171ecbce88b9fa64df1491ec65859ee108f2e787b26e03	3.7	Ransom.Win32.MIMIC.SMZTJJ-A
30f2fe10229863c57d9aab97ec8b7a157ad3ff9ab0b2110bbb4859694b56923f	3.9	Ransom.Win32.MIMIC.SMZTJJ-A
2e96b55980a827011a7e0784ab95dcee53958a1bb19f5397080a434041bbeeea	4	Ransom.Win32.MIMIC.SMZTJJ-A
136d05b5132adafc4c7616cd6902700de59f3f326c6931eb6b2f3b1f458c7457	4.2	Ransom.Win32.MIMIC.SMZTJJ-A
c576f7f55c4c0304b290b15e70a638b037df15c69577cd6263329c73416e490e		HackTool.Win32.DEFENDERCONTROL.Z