securelist.com /gopuram-backdoor-deployed-through-3cx-supply-chain-attack/109344/

Not just an infostealer: Gopuram backdoor deployed through 3CX supply chain attack

Authors

- Georgy Kucherin
- Vasily Berdnikov
- Vilen Kamalov

On March 29, Crowdstrike published a report about a supply chain attack conducted via 3CXDesktopApp, a popular VoIP program. Since then, the security community has started analyzing the attack and sharing their findings. The following has been discovered so far:

- The infection is spread via 3CXDesktopApp MSI installers. An installer for macOS has also been trojanized.
- The malicious installation package contains an infected dll library that decrypts a shellcode from the d3dcompiler_47.dll library's overlay and executes it.
- The decrypted payload extracts C2 server URLs from icons stored in a GitHub repository (the repository is removed).
- The payload connects to one of the C2 servers, downloads an infostealer and starts it.
- The infostealer collects system information and browser history, then sends it to the C2 server.

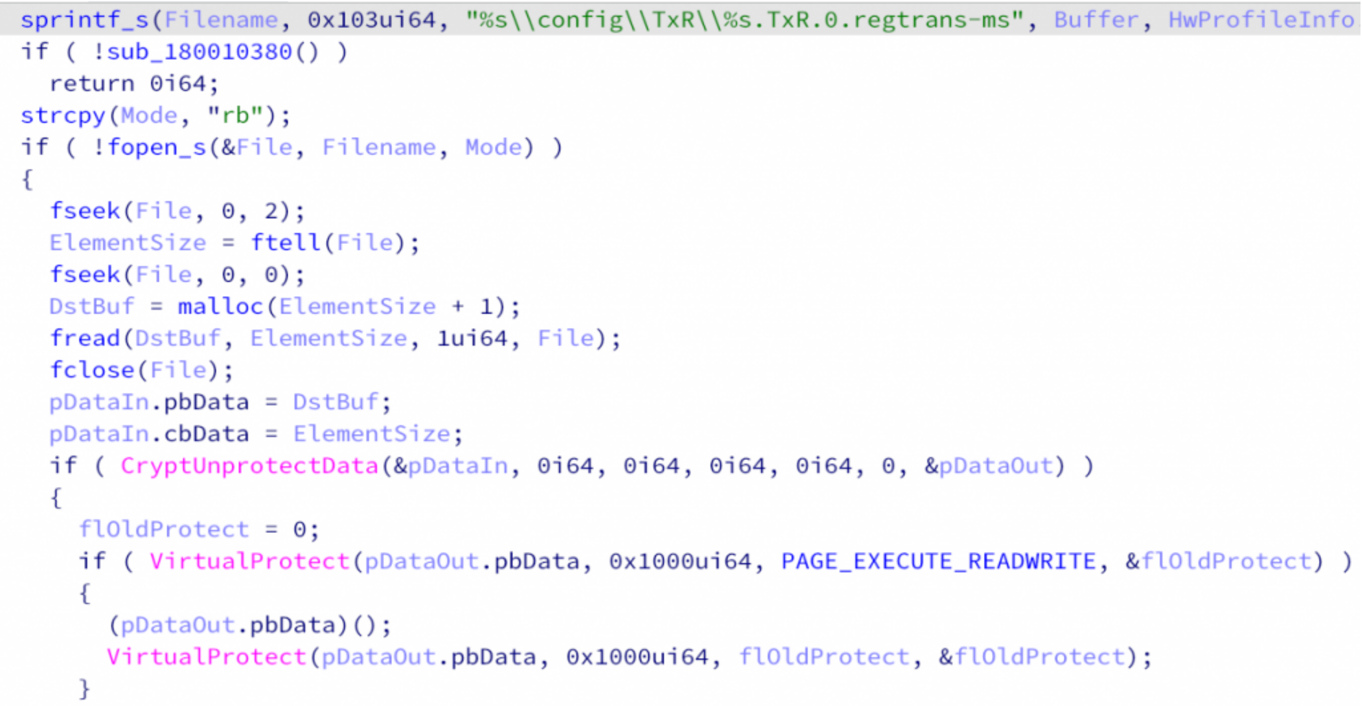
As we reviewed available reports on the 3CX attack, we began wondering if the compromise concluded with the infostealer or further implants followed. To answer that question, we decided to review the telemetry we had on the campaign. On one of the machines, we observed a DLL named guard64.dll, which was loaded into the infected 3CXDesktopApp.exe process. Interestingly enough, we opened an investigation into a case linked to that DLL on March 21, about a week before the supply chain attack was discovered. A DLL with that name was used in recent deployments of a backdoor that we dubbed “Gopuram” and had been tracking internally since 2020. Three years ago, we were investigating an infection of a cryptocurrency company located in Southeast Asia. During the investigation, we found that Gopuram coexisted on victim machines with AppleJeus, a backdoor attributed to the Korean-speaking threat actor Lazarus.

Over the years, we observed few victims compromised with Gopuram, but the number of infections began to increase in March 2023. As it turned out, the increase was directly related to the 3CX supply chain attack. We found out that the threat actor specifically targeted cryptocurrency companies, dropping the following files on infected machines:

- C:\Windows\system32\wlbsctrl.dll, a malicious library (MD5: 9f85a07d4b4abff82ca18d990f062a84);
- C:\Windows\System32\config\TxR\<machine hardware profile GUID>.TxR.0.regtrans-ms, an encrypted shellcode payload.

Once dropped, wlbsctrl.dll becomes loaded on every startup by the IKEEXT service via DLL hijacking. We further saw DLLs with the names ualapi.dll and ncobjapi.dll being sideloaded into spoolsv.exe and WmiPrvSE.exe, respectively.

The wlbsctrl.dll library is responsible for decrypting and executing the shellcode stored in the C:\Windows\System32\config\TxR directory. The decryption is notably performed through the CryptUnprotectData API function that uses a different encryption key internally on every machine. This makes it difficult for researchers to decrypt the payload from the file without physical access to the victim machines.



```
sprintf_s(Filename, 0x103ui64, "%s\\config\\TxR\\%s.TxR.0.regtrans-ms", Buffer, HwProfileInfo
if ( !sub_180010380() )
  return 0i64;
strcpy(Mode, "rb");
if ( !fopen_s(&File, Filename, Mode) )
{
  fseek(File, 0, 2);
  ElementSize = ftell(File);
  fseek(File, 0, 0);
  DstBuf = malloc(ElementSize + 1);
  fread(DstBuf, ElementSize, 1ui64, File);
  fclose(File);
  pDataIn.pbData = DstBuf;
  pDataIn.cbData = ElementSize;
  if ( CryptUnprotectData(&pDataIn, 0i64, 0i64, 0i64, 0i64, 0, &pDataOut) )
  {
    flOldProtect = 0;
    if ( VirtualProtect(pDataOut.pbData, 0x1000ui64, PAGE_EXECUTE_READWRITE, &flOldProtect) )
    {
      (pDataOut.pbData)();
      VirtualProtect(pDataOut.pbData, 0x1000ui64, flOldProtect, &flOldProtect);
    }
```

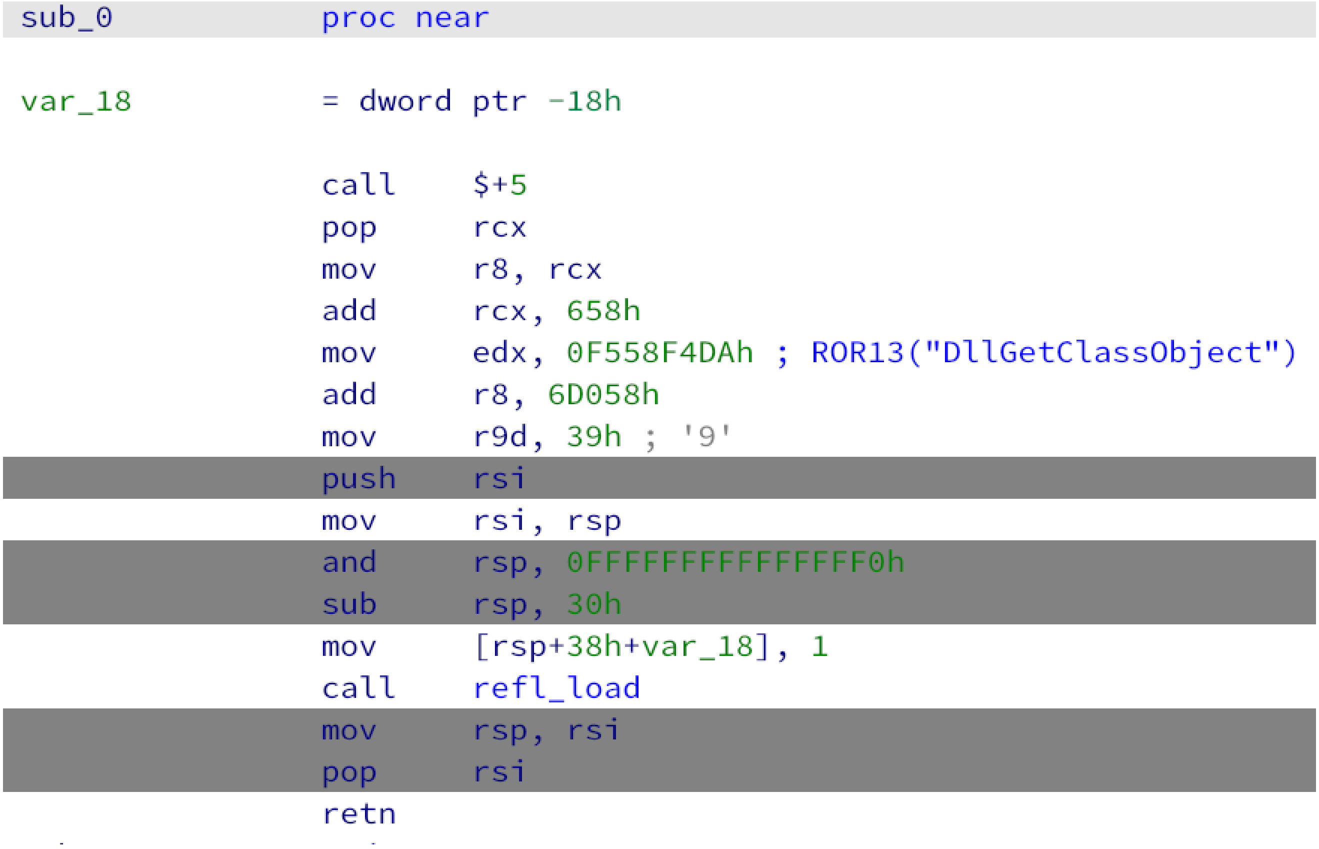
Snippet of the loading function using CryptUnprotectData

The component loaded by the library is Gopuram's main module. As mentioned above, its name in the export directory is guard64.dll. The job of the main module is to connect to a C2 server and request commands. The backdoor implements commands that allow the attackers to interact with the victim's file system and create processes on the infected machine. Gopuram was additionally observed to launch in-memory modules. Just like the implants used in the 3CX campaign, Gopuram's modules are DLL files that include an export function named DllGetClassObject. We have observed eight modules so far:

Module name	**Description**
Ping	Pings a host specified in the argument.
Connect	Connects to a given host via a socket and waits for the server to send data.
Registry	Manipulates registry (lists, adds, deletes and exports keys).
Service	Manipulates (creates, lists, starts, stops and deletes) services.
Timestomp	Performs timestomping on files.
Inject	Performs payload injections through syscalls via mapping a shellcode to a remote process and creating a remote thread.
KDU	Kernel Driver Utility that allows an attacker to bypass driver signature enforcement. The utility is used to load an unsigned driver (MD5: F684E10FF1FFCDD32C62E73A11382896). The driver collects information about installed AV filters and writes it to the C:\Windows\System32\catroot2\edb.chk.log file.
Update	Encrypts a provided payload and writes it to the C:\Windows\System32\config\TxR\<machine hardware profile GUID>.TxR.0.regtrans-ms file.
Net	Partially implements features of the net command: management of users, groups, sessions and network shares.

The discovery of the new Gopuram infections allowed us to attribute the 3CX campaign to the Lazarus threat actor with medium to high confidence. Our attribution is based on the following facts:

- While investigating an attack on a Southeast Asian cryptocurrency company in 2020, we found Gopuram coexisting on the same machine with the AppleJeus backdoor, which is attributed to Lazarus.
- The Gopuram backdoor has been observed in attacks on cryptocurrency companies, which is aligned with the interests of the Lazarus threat actor.
- While looking for additional implants that used the same loader shellcode as the 3CX implants, we discovered a sample on a multiscanner service (MD5: 933508a9832da1150fcfdbc1ca9bc84c) loading a payload that uses the wirexpro[.]com C2 server. The same server is listed as an IoC for an AppleJeus campaign by Malwarebytes.



```
sub_0           proc near

var_18          = dword ptr -18h

                call    $+5
                pop     rcx
                mov     r8, rcx
                add     rcx, 658h
                mov     edx, 0F558F4DAh ; ROR13("DllGetClassObject")
                add     r8, 6D058h
                mov     r9d, 39h ; '9'
                push    rsi
                mov     rsi, rsp
                and     rsp, 0FFFFFFFFFFFFFFF0h
                sub     rsp, 30h
                mov     [rsp+38h+var_18], 1
                call    refl_load
                mov     rsp, rsi
                pop     rsi
                retn
```

First bytes of the loader shellcode used in 3CX and AppleJeus

Note, though, that the shellcode is based on open-source code that has been used by other threat actors, for example, SilentBreak. Still, the use of that shellcode along with the 0xF558F4DA constant (which is the ROR13 hash for the string DllGetClassObject) is a more unique pattern.

- While investigating a malicious MSI file (MD5: ec3f99dd7d9dbce8d704d407b086e84f) that has been uploaded to a multiscanner service, we observed the following two events:
    - The dll library dropped from the MSI was observed to launch an in-memory payload that contacts the oilycargo[.]com domain. This domain name has previously been attributed to Lazarus by multiple researchers.
    - In our telemetry, we observed AvBugReport.exe, the executable hosting dll, to contain Gopuram's main module payload, guard64.dll.

These four facts allow us to conclude that Lazarus is likely the threat actor deploying the Gopuram backdoor.

As for the victims in our telemetry, installations of the infected 3CX software are located all over the world, with the highest infection figures observed in Brazil, Germany, Italy and France.

As the Gopuram backdoor has been deployed to less than ten infected machines, it indicates that attackers used Gopuram with surgical precision. We additionally observed that the attackers have a specific interest in cryptocurrency companies.

As it turns out, the infostealer is not the only malicious payload deployed during the 3CX supply chain attack. The threat actor behind Gopuram additionally infects target machines with the full-fledged modular

Gopuram backdoor. We believe that Gopuram is the main implant and the final payload in the attack chain. Our investigation of the 3CX campaign is still far from complete. We will continue analyzing the deployed implants to find out more details about the toolset used in the supply chain attack.

Gopuram indicators of compromise

MD5 hashes

9f85a07d4b4abff82ca18d990f062a84
96d3bbf4d2cf6bc452b53c67b3f2516a

File paths

C:\Windows\System32\config\TxR\<machine hardware profile GUID>.TxR.0.regtrans-ms
C:\Windows\system32\catroot2\edb.chk.log

More indicators of compromise and YARA rules for detecting Gopuram components are available for TIP subscribers. Contact intelreports@kaspersky.com for more details.