Reassembling Victim Domain Fragments from SUNBURST DNS

netresec.com/

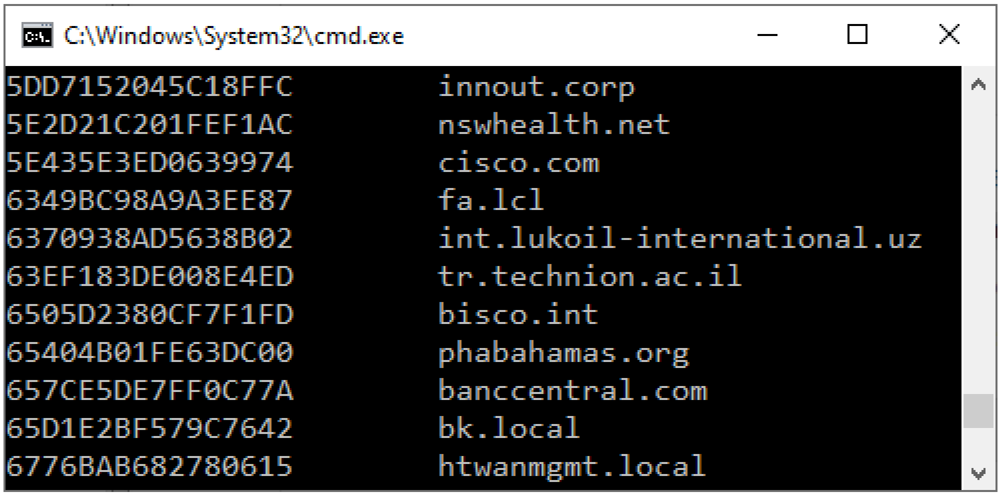
December 17, 2020

Erik Hjelmvik

,

Thursday, 17 December 2020 22:30:00 (UTC/GMT)

We are releasing a free tool called SunburstDomainDecoder today, which is created in order to help CERT organizations identify victims of the trojanized SolarWinds software update, known as SUNBURST or Solorigate.



```
5DD7152045C18FFC        innout.corp
5E2D21C201FEF1AC        nswhealth.net
5E435E3ED0639974        cisco.com
6349BC98A9A3EE87        fa.lcl
6370938AD5638B02        int.lukoil-international.uz
63EF183DE008E4ED        tr.technion.ac.il
6505D2380CF7F1FD        bisco.int
65404B01FE63DC00        phabahamas.org
657CE5DE7FF0C77A        banccentral.com
65D1E2BF579C7642        bk.local
6776BAB682780615        htwanmgmt.local
```

SunburstDomainDecoder can be fed with DNS queries to avsvmcloud.com in order to reveal the full internal domain names of infected companies and organizations.

UPDATE December 18, 2020 (v1.1)

SunburstDomainDecoder has now been updated to automatically reassemble fragmented domain name segments in order to show the full domain in the output.

UPDATE December 19, 2020 (v1.2)

Domain names that have been base32 encoded, such as domain names with uppercase letters, can now be extracted with SunburstDomainDecoder. The queried SUNBURST subdomains are now also included in the output.

UPDATE December 21, 2020 (v1.6)

Improved parsing of base32 encoded domain names. SUNBURST victim domains like "LKDataCenter.com", "Sunkistgrowers.com" and "BrokenArrow.Local" can now be extracted.

UPDATE December 27, 2020 (v1.7)

Improved reassembly of long domain names, like "CIMBMY.CIMBDomain.com" and "BE.AJINOMOTO-OMNICHEM.AD", that get segmented into multiple parts. Extraction of time stamps and security applications, including "Windows Defender", "Carbon Black", "CrowdStrike", "FireEye", "ESET" and "F-Secure". See Sergei Shevchenko's blog post Sunburst Backdoor, Part III: DGA & Security Software for more details.

UPDATE January 4, 2021 (v1.8)

Security products (WinDefend, ESET etc.) are now included in the summary output at the end. SUNBURST stage2 victims, which accept C2 domains in CNAME responses, are indicated with a "STAGE2" tag. The previous release marked stage2 queries with a "DNSSEC" tag. Improved extraction of truncated base32 domains, such as "*TED.com".

UPDATE January 12, 2021 (v1.9)

DNS queries with encoded timestamps are tagged with either "AVProducts" or "Ping", depending on if they include an update of the installed/running security products and services or not. The summary data at the end has been modified to also show partial domain names, such as "paloaltonetworks*".

UPDATE February 16, 2021 (v2.0)

Slightly faster and even more accurate than previous versions.

Download SunburstDomainDecoder.zip

SUNBURST DNS Traffic

SUNBURST victims, who have installed one of the trojanized SolarWinds Orion software updates, will query for domain names formatted like this:

<SUBDOMAIN>.appsync-api.eu-west-1.avsvmcloud.com
<SUBDOMAIN>.appsync-api.us-west-2.avsvmcloud.com
<SUBDOMAIN>.appsync-api.us-east-1.avsvmcloud.com
<SUBDOMAIN>.appsync-api.us-east-2.avsvmcloud.com

The "SUBDOMAIN" string has different values for each victim and the second half of this string actually contains an encoded domain name (encrypted with a simple substitution cipher).

RedDrip's decode.py

The RedDrip Team published a SUNBURST DGA decoding script yesterday, which can be used to identify SUNBURST victim organizations like CISCO and Belkin by decoding the domain names encoded in the outgoing DNS queries for subdomains of avsvmcloud.com.

This is what it looks like when RedDrip's decode.py script is fed with domain names from John Bambenek's uniq-hostnames.txt file.

```
cat uniq-hostnames.txt | python decode.py
02m6hcopd17p6h450gt3.appsync-api.us-west-2.avsvmcloud.com .gh
039n5tnndkhrfn5cun0y0sz02hij0b12.appsync-api.us-west-2.avsvmcloud.com ad001.mtk.lo
04spiistorug1jq5o6o0.appsync-api.us-west-2.avsvmcloud.com isi
060mpkprgdk087ebcr1jov0te2h.appsync-api.us-east-1.avsvmcloud.com belkin.com
06o0865eliou4t0btvef0b12eu1.appsync-api.us-east-1.avsvmcloud.com gncu.local
07605jn8l36uranbtvef0b12eu1.appsync-api.us-east-1.avsvmcloud.com gncu.local
07q2aghbohp4bncce6vi0odsovertr2s.appsync-api.us-east-1.avsvmcloud.com csnt.princegeor
07ttndaugjrj4pcbtvef0b12eu1.appsync-api.us-east-1.avsvmcloud.com gncu.local
08amtsejd02kobtb6h07ts2fd0b12eu1.appsync-api.eu-west-1.avsvmcloud.com sm-group.local
0b0fbhp20mdsv4scwo11r0oirssrc2vv.appsync-api.us-east-2.avsvmcloud.com ville.terrebonn
[...]
```

The beauty of this approach is that passive DNS data can be used in order to reliably identify the victims. This is great news for national CERTs, because they typically have readily access to passive DNS data and can use the decoded domain names in order to identify and reach out to victims in their country.

After using the python script provided by ReadDrip Team I noticed two things:

1. The leaked domain names were internal domain names used on the victim organizations' corporate networks. Many of the domains were using the ".local" suffix.
2. Most of the extracted domains were truncated to around 15 bytes, which make it difficult to identify the victim organization.

Truncated Domains Fragmented Domains

I later learned that what seemed to be truncated domains were actually fragmented domains, where long domain names would be split into multiple queries. This revelation turns the output from RedDrip's python tool into an interesting domain name puzzle. At this point I decided to take a closer look at the malicious SolarWinds update I had downloaded from SolarWind's website a few days ago -- yes, that's right the malicious software update "SolarWinds-Core-v2019.4.5220-Hotfix5.msp" (MD5: 02af7cec58b9a5da1c542b5a32151ba1) was actually available for download from SolarWinds' website long after they had been notified about their software being backdoored!

As an example, lets' take a closer look at this DNS query from John Bambenek's passive DNS data:
r1qshoj05ji05ac6eoip02jovt6i2v0c.appsync-api.us-west-2.avsvmcloud.com

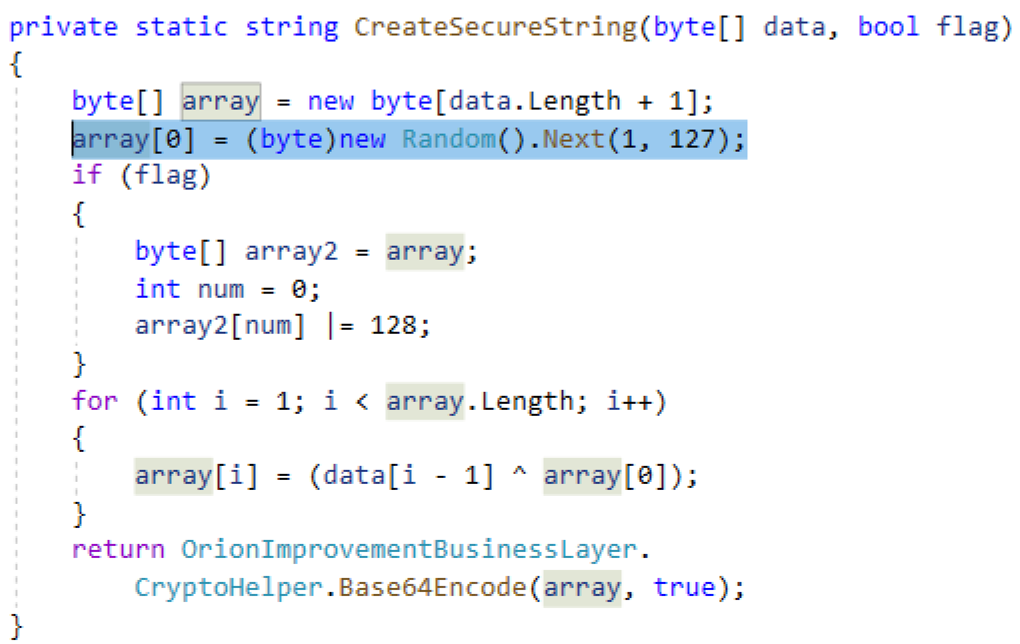
This query can be broken down into three parts:

1. **r1qshoj05ji05ac6** : What is encoded here???
2. **eoip02jovt6i2v0c** : Base32 encoded string "city.kingston."
3. **.appsync-api.us-west-2.avsvmcloud.com** : DNS trailer without encoded data

So, which "City of Kingston", or "Kingston City", should we contact to let them know that they have installed a trojanized SolarWinds update? Is it Kingston Jamaica, City of Kingston NY USA, City of Kingston Ontario Canada, Kingston City Tennessee USA or City of Kingston Australia?

After analyzing the "SolarWinds.Orion.Core.BusinessLayer.dll" file (MD5: b91ce2fa41029f6955bff20079468448) from the "SolarWinds-Core-v2019.4.5220-Hotfix5.msp" I learned that the initial "r1qshoj05ji05ac6" string is representing a unique "GUID" value for the infected machine. This GUID is generated by calculating an MD5 hash of the MAC address of the first active non-Loopback network interface, the domain name and the "MachineGuid" registry key value in "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography".

This MD5 hash is then squeezed into a tiny 8 byte array by XOR'ing overlapping bytes. The "CreateSecureString" function in the trojanized SolarWinds update then "encrypts" this hash using XOR with a random key, which is prepended to the data. The XOR key and the XOR'ed data is then finally base32 encoded into what makes up the first part of the subdomain to query for. Don't let the SUNBURST source code below fool you, it is actually using base32 encoding with a custom alphabet even though the function is called "Base64Encode";



```
private static string CreateSecureString(byte[] data, bool flag)
{
    byte[] array = new byte[data.Length + 1];
    array[0] = (byte)new Random().Next(1, 127);
    if (flag)
    {
        byte[] array2 = array;
        int num = 0;
        array2[num] |= 128;
    }
    for (int i = 1; i < array.Length; i++)
    {
        array[i] = (data[i - 1] ^ array[0]);
    }
    return OrionImprovementBusinessLayer.
        CryptoHelper.Base64Encode(array, true);
}
```

Image: SUNBURST source code generates a random value between 1 and 127 as XOR key

Each DNS lookup from an infected machine will query for a unique subdomain because a new XOR key will be generated for each request. Luckily for us, this XOR key is provided in each request, so we can use it in order to "decrypt" the subdomain and get the original 8 bytes derived from the MAC+domain+MachineGuid MD5 hash.

The output from my "SunburstDomainDecoder.exe" tool will print the "decrypted" 8 byte GUID in the first column, the decoded victim domain segment or timestamp in the second column and the queried SUNBURST subdomain in the last column. Each DNS query line read from standard input will generate a "GUID DecodedHostname SunburstSubdomain" line on standard output.

```
SunburstDomainDecoder.exe < uniq-hostnames.txt
F18613981DEC4D1A 2020-10-02T21:00:00.0000000Z 02m6hcopd17p6h450gt3
BD6DEFBBE9FEA3A9 ad001.mtk.lo 039n5tnndkhrfn5cun0y0sz02hij0b12
2BF8DE15406EA780 2020-08-25T03:00:00.0000000Z 043o9vacvthf0v95t81l
573DEB889FC54130 2020-08-13T21:00:00.0000000Z,
WindowsDefender_RUNNING,CrowdStrike_RUNNING 04jrge684mgk4eq8m8adfg7
518092C8FD571806 2020-06-09T22:30:00.0000000Z 04r0rndp6aom5fq5g6p1
F18613981DEC4D1A 2020-07-06T08:30:00.0000000Z 04spiistorug1jq5o6o0
BC1CB013239B4B92 2020-04-25T10:00:00.0000000Z 05q2sp0v4b5ramdf71l7
3ED2E979D53B2523 belkin.com 060mpkprgdk087ebcr1jov0te2h
4225A5C345C1FC8E gncu.local 06o0865eliou4t0btvef0b12eu1
[...]
```

The tool then finishes off by outputting the domains that are complete or at least have the last part of their domain intact. Some of these domains are complete because they were short enough to fit in one single SUNBURST DNS query, while others have been pieced together by SunburstDomainDecoder from domain fragments arriving in separate SUNBURST DNS queries.

```
[...]
F59BBAACBA3493C0 dufferincounty.on.ca
F5D6AA262381B084 glu.com
F9024D5B1E9717C6 gyldendal.local
F90BDDB47E495629 central.pima.gov
F956B5EF56BCF666 coxnet.cox.com
F9A9387F7D252842 city.kingston.on.ca
FB0B50553BC00DED gloucesterva.net
FBB6164BC2B0DFAD ARYZTA.COM
FD04AC52C95A1B0A bmrn.com
FDFCAB8E4C0AB3EE ansc.gob.pe
FE7FF8C9104A0508 thoughtspot.int
FF6760F36DB3D7DC smes.org
```

We can now see that it was "city.kingston.on.ca", (City of Kingston, Ontario, Canada) who had installed a trojanized SolarWinds update.

Download SunburstDomainDecoder

The C# source code and a compiled Windows binary for SunburstDomainDecoder is available here:
https://www.netresec.com/files/SunburstDomainDecoder.zip

The source code and Windows binary is shared under a Creative Commons CC-BY license, which means that you are free to:

- Share : copy and redistribute the material in any medium or format
- Adapt : remix, transform, and build upon the material for any purpose, even commercially.

Provided that you give appropriate credit, provide a link to the license, and indicate if changes were made.

Running SunburstDomainDecoder on Linux/MacOS

Wanna run SunburstDomainDecoder.exe but not in Windows? No problems, the tool runs perfectly fine in Mono. Another option is to build SunburstDomainDecoder.cs as a .NET core project in Linux.

.NET Reversing

Would you like to verify my findings or learn more about .NET reverse engineering? Cool, then I'd recommend that you download dnSpy in order to reverse engineer the SUNBURST .NET DLL (which can be extracted from the msp installer with 7zip). Or you can have a look at the already extracted OrionImprovementBusinessLayer.cs on GitHub.

Posted by Erik Hjelmvik on Thursday, 17 December 2020 22:30:00 (UTC/GMT)

Tags: #SunburstDomainDecoder #SUNBURST #SolarWinds #Solorigate #domain #DNS #pDNS #Windows Defender #Carbon Black #FireEye #ESET #F-Secure #Trojan #avsvmcloud