Finding SUNBURST victims and targets by using passive DNS, OSINT

vrieshd.medium.com/finding-sunburst-victims-and-targets-by-using-passivedns-osint-68f5704a3cdc

VriesHD

January 23, 2021

VriesHD

Jan 23, 2021

·

8 min read

For over a month now, the hack of SolarWinds' Orion IT management platform has been present in the news regularly with plenty of interesting discoveries on the modus operandi of the attackers and the effects of the hack on several targeted companies and government branches. However, there's been little information about some of the connections that SUNBURST has shared 'in public' and the stories of the affected organisations, while there have also been some stories that tried to grasp these connections, but ended up in providing the opposite effect; a false sense of security.

A quick summary for those who've not been aware of this recent hack yet; On December 13, 2020, FireEye put out a post sharing that they "discovered a supply chain attack trojanizing SolarWinds Orion business software updates in order to distribute malware". After some research, it turned out that up to 18.000 SolarWinds customers could've potentially received the trojanized updates for the Orion software. These customers should be considered 'victims' of the attack. Only 'high value' organisations of interest to the attackers received additional malicious software intended for further exploitation. These customers should be considered 'targets' of the attack.

Decrypting SUNBURST domains

There have been plenty of posts and tools on how to decrypt SUNBURST domains so I'll try to keep this as short as possible:

In general, the SUNBURST backdoor collects several kinds of information about the infected system, encrypts this information into a combination of strings, adds these together, and sends this information back to the attackers through the use of DNS requests for subdomains of the avsvmcloud[.]com domain. To be specific, the subdomains are always similar to the following patterns:

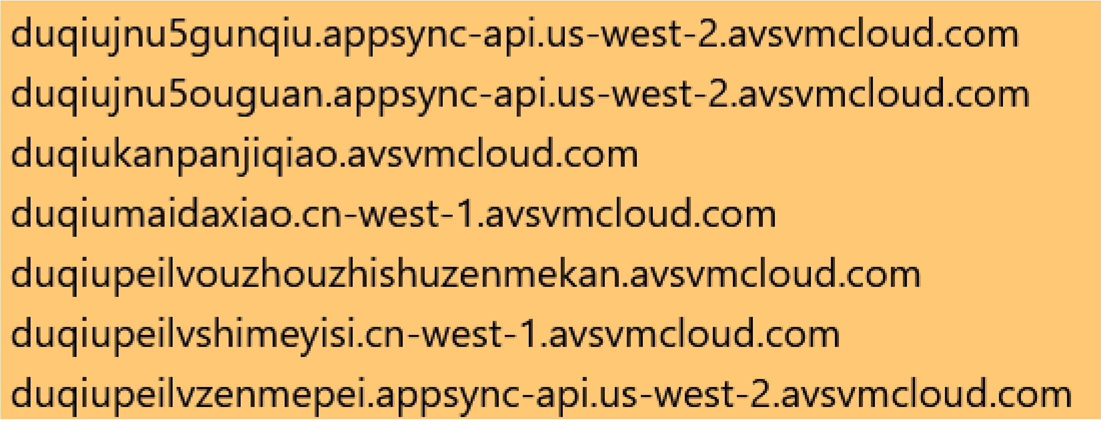
> [subdomain].appsync-api.eu-west-1[.]avsvmcloud[.]com
>
> [subdomain].appsync-api.us-west-2[.]avsvmcloud[.]com
>
> [subdomain].appsync-api.us-east-1[.]avsvmcloud[.]com
>
> [subdomain].appsync-api.us-east-2[.]avsvmcloud[.]com

Even though there are four possible options for the first-subdomain, being eu-west-1/us-west-2/us-east-1/us-east-2, these do not seem to relate to any specific geographical targeting, nor does changing these domains change anything on the encoded data that's been submitted in the third-subdomain. Their only intention so far seems to be to mimic services like AmazonAWS to give the made connections some form of legitimacy. Occasionally I've seen several variations on these four first-subdomains like cn-west-1, eu-west-2, and us-west-1 yet there is no indication that these subdomains have been in use by the backdoor itself.

As for the third-subdomain, this is where the transferred data comes into play. I don't want to get too much into the actual encryption/decryption as others like RedDrip Team from QiAnXin Technology, Prevasio, Cloudflare and NETRESEC have already written detailed reports on this. In summary, these subdomains consist of the following parts: an encoded GUID, a byte that functions as the XORkey for the GUID and the hostname of the local network of the infected system or other additional information like encoded timestamps or active Antivirus-products or the confirmation to become a target instead of a victim. These are the important bits that supply both the attackers as well as the community important information about the infected systems.

Passive DNS and the post-December noise

As mentioned above, the SUNBURST backdoor reports back to the avsmcloud[.]com domain with the collected data in the shape of DNS requests for a specific subdomain. So collecting as much as these requests as possible is important as in a lot of cases the collected data is transferred from the backdoor to the attackers in several batches (e.g. local hostnames and timestamps are never sent together, nor do long hostnames get sent in one request but are fragmented in multiple queries based on their length as the subdomains are limited to a max length of 32 characters). There are many ways to get passive DNS on avsmcloud[.]com, there are several pastebins with lists of passive DNS and there are several parties like RiskIQ, FarSight DNSDB, VirusTotal, and others that have big lists of records for the domain. However, after the first reports came out about the hack, the passive DNS results for avsmcloud[.]com subdomains have kind of gotten out of hand as the domain accepted any request due to the lack of knowing what kind of systems were running the Solarwinds software and malicious updates, both before and after the Microsoft takeover. Combined with the messy nature of passive DNS on its own, it turned out into a bit of a mess… And that's an understatement with close to 200k recorded subdomains.. and probably way more as this is only based on my findings…



```
duqiujnu5gunqiu.appsync-api.us-west-2.avsvmcloud.com
duqiujnu5ouguan.appsync-api.us-west-2.avsvmcloud.com
duqiukanpanjiqiao.avsvmcloud.com
duqiumaidaxiao.cn-west-1.avsvmcloud.com
duqiupeilvouzhouzhishuzenmekan.avsvmcloud.com
duqiupeilvshimeyisi.cn-west-1.avsvmcloud.com
duqiupeilvzenmepei.appsync-api.us-west-2.avsvmcloud.com
```

a small portion of passive DNS data on avsmcloud[.]com

Fortunately, there are a few clues that helping sorting through this noise:

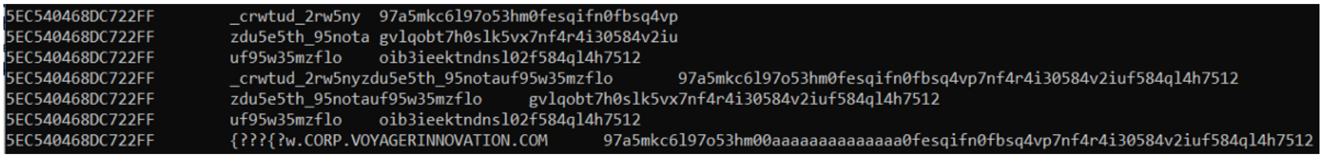
First of all, we know the backdoor communicates in the mentioned patterns as mentioned above so that sorts out a big part of the noise (set the odd cn-west-1, etc. subdomains aside for a bit, as their third-subdomains could still contain actual information). Second, we know the GUID and XORkey make up 16 characters and the backdoor has a 32 character limit, so the third-subdomain should be between 17 and 32 characters long. Furthermore, you can discard any subdomains that contain any symbols in the third-subdomain.

The SUNBURST Puzzles

Now that you've got a decent bunch of DNS requests, you can start decoding the subdomains with tools such as the ones provided by RedDrip, FireEye, or NETRESEC. Their tools will do a lot of work for you, and sometimes even do all the work, depending on the amount of data you supply to the tools. The GUID's come into play to help with connecting the separate queries, as that specific GUID stays unique to the infected system regardless of the XOR'ing of the GUID. This way you're also able to match encoded timestamps to hostnames and the other way around. The XORkey, however, is also an indicator for longer split domains on which part is based on the decoded value of the byte, ranging from 0 to 35. The first part of the payload will have a byte value of 0 if the domain is long enough to require multiple requests. The last part of the payload will always have a byte value of 35. Infected systems with short domain names will have only one request with a byte value of 35. This is kinda tricky as it's not always clear whether a domain is the last part of a fragmented domain, or just very short.

Most of this will work out just fine, however, sometimes you will find yourself ending up missing a piece or two for a full domain. In the case of only lowercase alphanumeric domains, this ain't too much of a problem as you will often be able to find the remaining bit by using the same passive DNS to look for similar domains with additional characters, or you can find them while simply googling for the bit you have. Do however keep some caution while doing this, as not every first result will be the one you're looking for. E.g. uo8igvgkvslrh9b9e6vi0edsovertr2s[.]appsync-api[.]us-east-1[.]avsvmcloud.com decodes to 'csnt.princegeor'. When searching for princegeorge one of the first results ends up as princegeorge[.]ca, which seems plausible, however, with some proper research you will be able to find that princegeorge[.]com, even though seemingly unimportant at first sight, has had an actual subdomain involving 'csnt'. If you look at who owns princegeorge[.]com, it's fairly obvious which of the two is way more likely to use Solarwinds software.

Another bit that the tools seem to have problems with, are the domains that contain characters beyond lower alphanumeric characters and ".-_". As the SUNBURST backdoor uses a different method of encryption for these domains (base32 encoding with a custom alphabet). There are a few options, often consisting of missing pieces or misplaced single/double 0's when joining parts. When you do know the order of the pieces is correct based on the byte values, check for any potential overlapping/connecting 0's. Often that solves the issue. As for the missing pieces, you can be a bit cheeky with those.

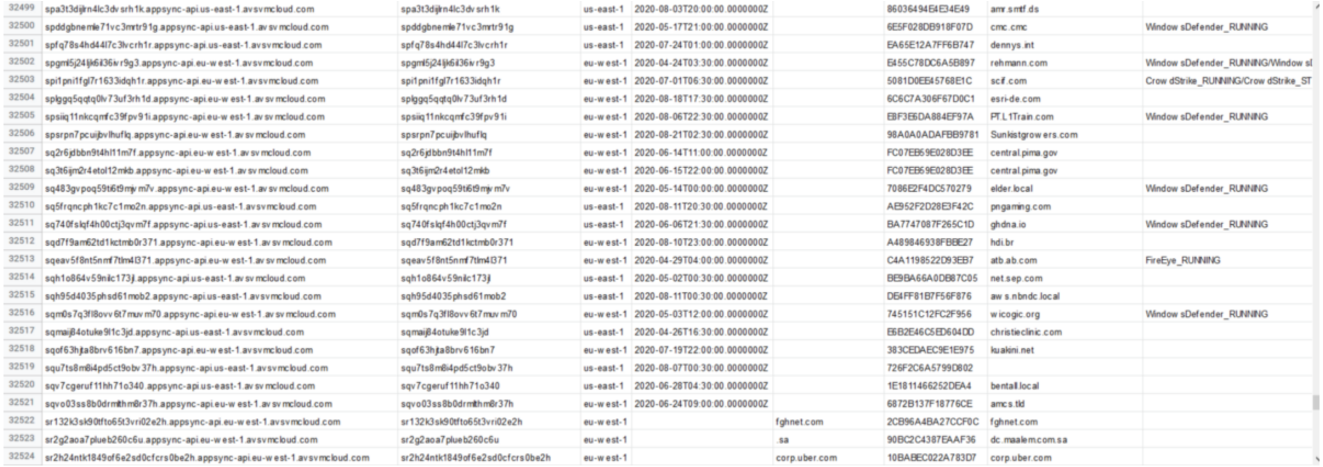


```
5EC540468DC722FF        _crwtud_2rw5ny  97a5mkc6l97o53hm0fesqifn0fbsq4vp
5EC540468DC722FF        zdu5e5th_95nota gvlqobt7h0slk5vx7nf4r4i30584v2iu
5EC540468DC722FF        uf95w35mzflo    oib3ieektndnsl02f584ql4h7512
5EC540468DC722FF        _crwtud_2rw5nyzdu5e5th_95notauf95w35mzflo       97a5mkc6l97o53hm0fesqifn0fbsq4vp7nf4r4i30584v2iuf584ql4h7512
5EC540468DC722FF        zdu5e5th_95notauf95w35mzflo     gvlqobt7h0slk5vx7nf4r4i30584v2iuf584ql4h7512
5EC540468DC722FF        uf95w35mzflo    oib3ieektndnsl02f584ql4h7512
5EC540468DC722FF        {???{?w.CORP.VOYAGERINNOVATION.COM      97a5mkc6l97o53hm00aaaaaaaaaaaaaa0fesqifn0fbsq4vp7nf4r4i30584v2iuf584ql4h7512
```

Example of tweaking for GUID '5EC540468DC722FF'

Sometimes manually joining the parts allows the tool to better understand the given input. If this fails, I prefer to just add a 'donor' piece. As we know the backdoor limits it's pieces to 32 characters max, we know that when we miss the first part out of four parts, that the first part has to be 16 characters starting with 00. Add in the donor and you will get a view of what the other pieces are. Sure, you don't have the full domain at this point, but knowing what 3/4 pieces make up for is way more information than having none. It also gives you additional options to find the potential missing part with more passive DNS/OSINT work. You could even bruteforce the connecting bit of the missing first and second piece by comparing the results to the first part of the second piece. Will it resolve your entire domain? No, but keep in mind that knowing a single extra character could mean so much more for further passive DNS/OSINT work and potential informing victims/targets.

For those seeking additional passive DNS data or just want to check whether they are a victim/target, I've got a sheet with 35k known public subdomains and their transmitted data over here.



32499	spa3t3dijrn4lc3dvsrh1k.appsync-api.us-east-1.avsvmcloud.com	spa3t3dijrn4lc3dvsrh1k	us-east-1	2020-08-03T20:00:00.0000000Z		86036494E4E34E49	amr.smtf.ds	
32500	spddgbnemle71vc3mrtr91g.appsync-api.us-east-1.avsvmcloud.com	spddgbnemle71vc3mrtr91g	us-east-1	2020-05-17T21:00:00.0000000Z		6E5F028DB918F07D	cmc.cmc	WindowsDefender_RUNNING
32501	spfq78s4hd44l7c3lvcrh1r.appsync-api.us-east-1.avsvmcloud.com	spfq78s4hd44l7c3lvcrh1r	us-east-1	2020-07-24T01:00:00.0000000Z		EA65E12A7FF6B747	dennys.int	
32502	spgml5j24ljk6il36ivr9g3.appsync-api.eu-west-1.avsvmcloud.com	spgml5j24ljk6il36ivr9g3	eu-west-1	2020-04-24T03:30:00.0000000Z		E455C78DC6A5B897	rehmann.com	WindowsDefender_RUNNING/Windows
32503	spi1pni1fgl7r1633idqh1r.appsync-api.eu-west-1.avsvmcloud.com	spi1pni1fgl7r1633idqh1r	eu-west-1	2020-07-01T06:30:00.0000000Z		5081D0EE45768E1C	scif.com	CrowdStrike_RUNNING/CrowdStrike_ST
32504	splggq5qqtq0lv73uf3rh1d.appsync-api.eu-west-1.avsvmcloud.com	splggq5qqtq0lv73uf3rh1d	eu-west-1	2020-08-18T17:30:00.0000000Z		6C6C7A306F67D0C1	esri-de.com	
32505	spsiiq11nkcqmfc39fpv91i.appsync-api.eu-west-1.avsvmcloud.com	spsiiq11nkcqmfc39fpv91i	eu-west-1	2020-08-06T22:30:00.0000000Z		EBF3E6DA884EF97A	PT.L1Train.com	WindowsDefender_RUNNING
32506	spsrpn7pcuijbvlhuflq.appsync-api.eu-west-1.avsvmcloud.com	spsrpn7pcuijbvlhuflq	eu-west-1	2020-08-21T02:30:00.0000000Z		98A0A0ADAFBB9781	Sunkistgrowers.com	
32507	sq2r6jdbbn9t4hl11m7f.appsync-api.eu-west-1.avsvmcloud.com	sq2r6jdbbn9t4hl11m7f	eu-west-1	2020-06-14T11:00:00.0000000Z		FC07EB59E028D3EE	central.pima.gov	
32508	sq3t6ijm2r4etol12mkb.appsync-api.eu-west-1.avsvmcloud.com	sq3t6ijm2r4etol12mkb	eu-west-1	2020-06-15T22:00:00.0000000Z		FC07EB59E028D3EE	central.pima.gov	
32509	sq483gvpoq59ti6t9mjvm7v.appsync-api.eu-west-1.avsvmcloud.com	sq483gvpoq59ti6t9mjvm7v	eu-west-1	2020-05-14T00:00:00.0000000Z		7086E2F4DC570279	elder.local	WindowsDefender_RUNNING
32510	sq5frqncph1kc7c1mo2n.appsync-api.us-east-1.avsvmcloud.com	sq5frqncph1kc7c1mo2n	us-east-1	2020-08-11T20:30:00.0000000Z		AE952F2D28E3F42C	pngaming.com	
32511	sq740fslqf4h00ctj3qvm7f.appsync-api.us-east-1.avsvmcloud.com	sq740fslqf4h00ctj3qvm7f	us-east-1	2020-06-06T21:30:00.0000000Z		BA7747087F265C1D	ghdna.io	WindowsDefender_RUNNING
32512	sqd7f9am62td1kctmb0r371.appsync-api.eu-west-1.avsvmcloud.com	sqd7f9am62td1kctmb0r371	eu-west-1	2020-08-10T23:00:00.0000000Z		A489846938FBBE27	hdi.br	
32513	sqeav5f8nt5nmf7tlm4l371.appsync-api.eu-west-1.avsvmcloud.com	sqeav5f8nt5nmf7tlm4l371	eu-west-1	2020-04-29T04:00:00.0000000Z		C4A1198522D93EB7	atb.ab.com	FireEye_RUNNING
32514	sqh1o864v59nilc173jl.appsync-api.us-east-1.avsvmcloud.com	sqh1o864v59nilc173jl	us-east-1	2020-05-02T00:30:00.0000000Z		BE9BA66A0DB87C05	net.sep.com	
32515	sqh95d4035phsd61mob2.appsync-api.us-east-1.avsvmcloud.com	sqh95d4035phsd61mob2	us-east-1	2020-08-11T00:30:00.0000000Z		DE4FF81B7F56F876	aws.nbndc.local	
32516	sqm0s7q3fl8ovv6t7muvm70.appsync-api.eu-west-1.avsvmcloud.com	sqm0s7q3fl8ovv6t7muvm70	eu-west-1	2020-05-03T12:00:00.0000000Z		745151C12FC2F956	wicogic.org	WindowsDefender_RUNNING
32517	sqmaij84otuke9l1c3jd.appsync-api.us-east-1.avsvmcloud.com	sqmaij84otuke9l1c3jd	us-east-1	2020-04-26T16:30:00.0000000Z		E6B2E46C5ED604DD	christieclinic.com	
32518	sqof63hjta8brv616bn7.appsync-api.eu-west-1.avsvmcloud.com	sqof63hjta8brv616bn7	eu-west-1	2020-07-19T22:00:00.0000000Z		383CEDAEC9E1E975	kuakini.net	
32519	squ7ts8m8i4pd5ct9obv37h.appsync-api.us-east-1.avsvmcloud.com	squ7ts8m8i4pd5ct9obv37h	us-east-1	2020-08-07T00:30:00.0000000Z		726F2C6A5799D802		
32520	sqv7cgeruf11hh71o340.appsync-api.us-east-1.avsvmcloud.com	sqv7cgeruf11hh71o340	us-east-1	2020-06-28T04:30:00.0000000Z		1E1811466252DEA4	bentall.local	
32521	sqvo03ss8b0drmlthm8r37h.appsync-api.eu-west-1.avsvmcloud.com	sqvo03ss8b0drmlthm8r37h	eu-west-1	2020-06-24T09:00:00.0000000Z		6872B137F18776CE	amcs.tld	
32522	sr132k3sk90tfto65t3vri02e2h.appsync-api.eu-west-1.avsvmcloud.com	sr132k3sk90tfto65t3vri02e2h	eu-west-1		fghnet.com	2CB96A4BA27CCF0C	fghnet.com	
32523	sr2g2aoa7plueb260c6u.appsync-api.eu-west-1.avsvmcloud.com	sr2g2aoa7plueb260c6u	eu-west-1		.sa	90BC2C4387EAAF36	dc.maalem.com.sa	
32524	sr2h24ntk1849of6e2sd0cfcrs0be2h.appsync-api.eu-west-1.avsvmcloud.com	sr2h24ntk1849of6e2sd0cfcrs0be2h	eu-west-1		corp.uber.com	10BABEC022A783D7	corp.uber.com	

Overview of data in the sheet mentioned above.

I do want to point out that if your domain/hostname is not in this sheet, that it does not mean you/your organisation are not affected. This is only information that is known publicly upon this point.

If anyone has additional subdomains that are not in this sheet, feel free to share them with me through Twitter(tweet/dm) or the comment section in the sheet. As I want to contradict a quote from a previous story on the SUNBURST subject;

> “the full extent of this breach will most likely never be communicated to the public, and instead will be restricted to trusted parts of the intelligence community.”

The only way the public will not be able to determine the full extent of this breach on its own is by hiding the information that we as a security community have on this attack. This is not your regular hacking/leaked database incident, based on both the sophistication of the campaign and the targeted organisations. I understand that networks need to get investigated and cleaned first, but I would like to ask every affected organisation to be open about their infection(s) and the steps taken afterwards. As for those having access to more DNS data, keep in mind that this is a joint effort and that we’re all missing pieces. Sharing is caring. Follow the example of FireEye. We need subdomains to match domains with pings, we need CNAMES to match with targets, etc. Security isn’t always a business model.