LA-UR-21-28036
Approved for public release; distribution is unlimited.

Title: Malware Analysis Report for SolarWinds.Orion.Core.BusinessLayer.dll

Author(s): Pearce, Lauren

Intended for: Training Event

Issued: 2021-08-11

Disclaimer:
Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by Triad National Security, LLC for the National Nuclear Security Administration of U.S. Department of Energy under contract 89233218CNA000001. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

Malware Analysis Report for SolarWinds.Orion.Core.BusinessLayer.dll

Lauren Pearce

Los Alamos National Laboratory

Computer Security Incident Response Team

Summary

SolarWinds Orion is a software suite designed to provide “single pane of glass” monitoring of network devices and applications. On December 13 2020, FireEye revealed that SolarWinds Orion was the victim of a supply chain attack and that the SolarWinds signed file SolarWinds.Orion.Core.BusinessLayer.dll contained a backdoor. SolarWinds deployed the back-doored patch to customers beginning in March 2020. FireEye first identified this backdoor and labeled it SUNBURST. The backdoor is composed of roughly 4,000 lines[1] of lightly obfuscated .NET within the SolarWinds.Orion.Core.BusinessLayer.OrionImprovementBusinessLayer class of the malicious dll. The backdoor’s capabilities include:

- Ability to detect and potentially disable antivirus and forensics tools
- Basic command and control via DNS
- More advanced command and control via HTTP

Conditions of Execution

SUNBURST performs the following five checks before making any changes to the victim host or attempting to contact a command and control server:

1. Obtains its own process name, hashes it, and compares it to an embedded hash of the name “solarwinds.businesslayerhost.exe”. “Solarwinds.businesslayerhost.exe” is the name of SolarWinds.Orion.Core.BusinessLayer.dll’s host process in normal conditions. If the hashes do not match, the backdoor does not execute any further.

2. Checks if its own file has a write time of over twelve days ago. SUNBURST does not execute if the write time of its file is not at least twelve days prior.

3. Obtains the name of the active directory domain in which it is executing. If the domain name contains the words “test” or “SolarWinds”, matches a list of specific domain names common to test labs, matches SolarWinds’ own development domain, or if the machine is off domain, it does not execute any further.

4. Checks a configuration file to confirm that it was not previously killed by its own DNS command and control. If the configuration file indicates it was previously killed, the malware will not execute any further.

[1] Microsoft. Analyzing Solorigate, the compromised DLL file that started a sophisticated cyberattack, and how Microsoft Defender helps protect customers. https://www.microsoft.com/security/blog/2020/12/18/analyzing-solorigate-the-compromised-dll-file-that-started-a-sophisticated-cyberattack-and-how-microsoft-defender-helps-protect/

    a. SUNBURST uses a standard SolarWinds configuration file to store this information. If a patch overwrites this configuration file, the malware may attempt to reach out again.

5. Tests its network connection by making a DNS request to api.solarwinds.com. If the request is not successful, it does not execute any further.

There is a legitimate background job associated with SolarWinds Orion that periodically calls the backdoor code. Consequently, the backdoor will intermittently re-attempt these checks. See Appendix 1 for a visual map of this control flow.

Many of the above features were likely included for protection against detection by SolarWinds developers. They also make detection in a lab environment more difficult. Further, the network connection test prevents the malware from performing potentially alerting DNS requests in an air-gapped environment.

Immediately after passing the five tests above, the malware creates the named pipe 583da945–62af-10e8–4902-a8f205c72b2e.

Antivirus and Forensic Tool Detection:

After completing the initial checks but prior to attempting to establish command and control, SUNBURST enumerates the processes, services, and drivers running on its victim machine, and hashes each string. It compares these hashes to a hard coded list of hashes. Multiple computer security companies brute forced the hashes and identified them as the names of processes, services, and drivers associated with various security products. Appendix 2 contains a full list of broken hashes.

If SUNBURST detects a hash match, it will do one of the following two things:

1. For some processes and drivers, SUNBURST will simply exit in order to avoid exposure. Processes that lead to an exit include common reverse engineering tools that can be found in any malware analyst’s sandbox, but also include a subset of security monitoring tools such as Sysmon, Tanium, RedCloak, and Avast.

2. For some hashes mapping to specific service names, SUNBURST will attempt to disable the service rather than simply exiting. Services falling into this category include those created by CarbonBlack, Windows Defender, and F-Secure’s Anti-Virus, among others. SUNBURST attempts to disable these services by modifying their start values in the Windows registry. This registry value will not affect the service until the next power cycle. After making this change, SUNBURST notes it in its configuration file, and then exits. When relaunched later, SUNBURST checks the setting in the configuration file and simply assumes that the services are dead. It does not re-check if they are running. Consequently, if the service was resilient to the registry setting change, SUNBURST may still execute while a blacklisted service is running.

SUNBURST proceeds in execution only if it does not detect any blacklisted processes or drivers and only if it has attempted to disable all blacklisted services. In addition to broken hashes, Appendix 2 contains a full list of processes, services, drivers, and their corresponding actions.

Command and Control over DNS

Assuming the victim machine passes the above checks, SUNBURST enters a function that is capable of performing very limited command and control over DNS. The malware uses its own domain generation algorithm to generate a semi-random domain. The first part of the string generated is a random identifier. In limited cases, the second part of the generated string is an encoding of the victim site's domain name. The malware then prepends this semi-random string to one of the following four domains:

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

A complete request will resemble the following:
04spiistorug1jq5o6o0.appsync-api.us-west-2.avsvmcloud[.]com

The adversary controlled the DNS server for avsvmcloud[.]com and consequently controlled what IP address was returned to the victim machine. The returned IP address dictates program flow and results in one of the following four outcomes:

1. SUNBURST updates a configuration file and terminates. The configuration file change prevents the backdoor from attempting any further communication in the future.
    a. SUNBURST uses a standard SolarWinds configuration file to store this information. If a patch overwrites this configuration file, the victim machine may attempt to reach out again.
2. SUNBURST transitions to passive mode[2]. In this state, the backdoor is still viable for future use, but is not being actively exploited.
3. SUNBURST transitions to active mode. The backdoor prepares to move to HTTP command and control. SUNBURST will continue beaconing in this mode.
4. SUNBURST has all of the configuration information it needs to move to HTTP command and control and beacons, just waiting for the signal to move to HTTP.

[2] This report uses the vocabulary established by FireEye: FireEye Threat Research. “SUNBURST Additional Technical Details”. https://www.fireeye.com/blog/threat-research/2020/12/sunburst-additional-technical-details.html

If the controller returns a CNAME with one of the responding IPs, the backdoor will move to HTTP command and control.

The table contained in Appendix 3 details the specific subnets and their corresponding outcomes.

Command and Control over HTTP

Command and control over DNS gives the malware controller very few options, but one of them is to move to command and control over HTTP. The options available over HTTP are still comparatively limited, but include functionality to perform system reconnaissance and drop and execute a follow on stage of malware. FireEye's initial post on the SolarWinds compromise[3] examined the HTTP command and control routines in depth and provided a table of commands, available in Appendix 4. Once the adversary leverages command and control over HTTP to drop and execute a follow on stage, the adversary is no longer dependent on the SolarWinds backdoor for command and control or actions on objective.

Conclusion

The SUNBURST backdoor clearly prioritizes stealth. It is common for malware to attempt to detect reverse engineering tools, but rarely does malware check for such an exhaustive list of them. Additionally, while it is common for a malware sample to quit upon detection of reverse engineering or forensics tools, this sample's choice to quit upon detecting certain security tools such as Tanium or Redcloak is relatively unique. The degree of effort this adversary put into avoiding reverse engineering tools and their willingness to walk away in the face of certain security applications suggests that this adversary placed a greater than average emphasis on stealth. This emphasis on stealth itself suggests that they hoped to preserve this backdoor for an extended period.

SUNBURST is only one component of a complicated and protracted attack chain. The limited features available over command and control indicate that this backdoor was designed to perform system reconnaissance, and then either terminate command and control activity or drop additional files to host and execute them. Open source reports indicate that SUNBURST has dropped at least four different families of follow on malware, the most common being a memory dropper FireEye dubbed TEARDROP. SUNBURST is a sophisticated and stealthy backdoor designed to remain hidden, gather system information, and drop and execute additional malware.

[3] FireEye Threat Research. "Highly Evasive Attacker Leverages SolarWinds Supply Chain to Compromise Multiple Global Victims with SUNBURST Backdoor". https://www.fireeye.com/blog/threat-research/2020/12/evasive-attacker-leverages-solarwinds-supply-chain-compromises-with-sunburst-backdoor.html

References:

FireEye Threat Research. “Highly Evasive Attacker Leverages SolarWinds Supply Chain to Compromise Multiple Global Victims with SUNBURST Backdoor”. https://www.fireeye.com/blog/threat-research/2020/12/evasive-attacker-leverages-solarwinds-supply-chain-compromises-with-sunburst-backdoor.html

FireEye Threat Research. “SUNBURST Additional Technical Details”. https://www.fireeye.com/blog/threat-research/2020/12/sunburst-additional-technical-details.html

Sentinel Labs. “SolarWinds SUNBURST Backdoor: Inside the APT Campaign”. https://labs.sentinelone.com/solarwinds-sunburst-backdoor-inside-the-stealthy-apt-campaign/.https://research.checkpoint.com/2020/sunburst-teardrop-and-the-netsec-new-normal/

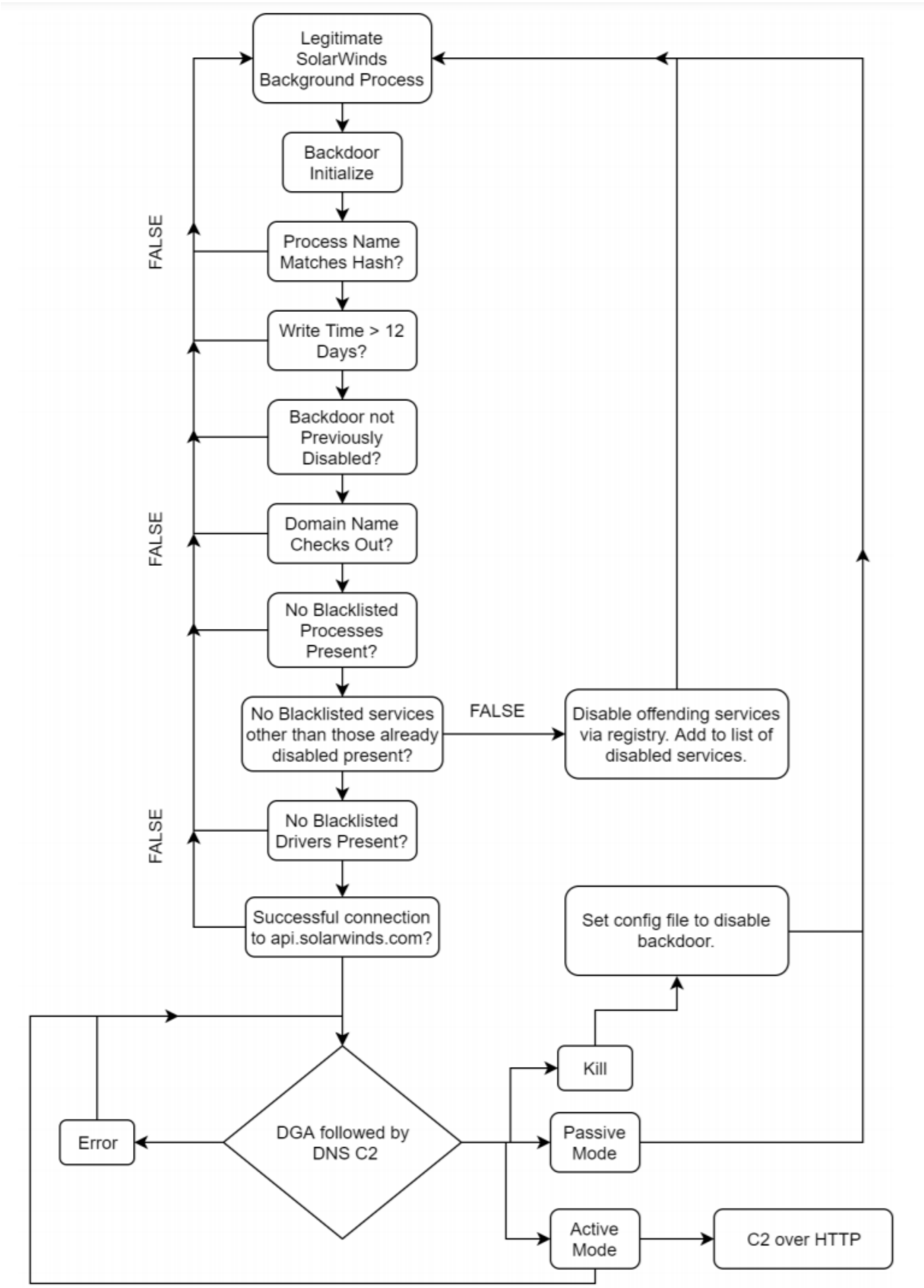
Symantec Enterprise Blogs. “SolarWinds Attacks: Stealthy Attackers Attempted to Evade Detection”. https://symantec-enterprise-blogs.security.com/blogs/threat-intelligence/solarwinds-attacks-stealthy-attackers-attempted-evade-detection

Malware.News. “Sunburst Backdoor: A Deeper Look into the SolarWinds Supply Chain Malware”. https://malware.news/t/sunburst-backdoor-a-deeper-look-into-the-solarwinds-supply-chain-malware/45556

Santosh Kumar. “Deep Dive into SolarWinds Sunburst Backdoor”. https://santoshjram.medium.com/deep-dive-into-solarwinds-sunburst-backdoor-e24c0e9042c6

Sophos News. “How SunBurst Malware does Defense Evasion”. https://news.sophos.com/en-us/2020/12/21/how-sunburst-malware-does-defense-evasion/

Appendix 1: Path to HTTP C2


Legitimate SolarWinds Background Process
Backdoor Initialize
Process Name Matches Hash?
FALSE
Write Time > 12 Days?
Backdoor not Previously Disabled?
Domain Name Checks Out?
FALSE
No Blacklisted Processes Present?
No Blacklisted services other than those already disabled present?
FALSE
Disable offending services via registry. Add to list of disabled services.
No Blacklisted Drivers Present?
FALSE
Successful connection to api.solarwinds.com?
Set config file to disable backdoor.
Kill
Error
DGA followed by DNS C2
Passive Mode
Active Mode
C2 over HTTP

[4]

[4] The “passive” and “active” terminology comes from FireEye’s “SUNBURST Additional Technical Details”, previously cited. This report also contains a differently detailed flow chart.

Appendix 2: Broken Hashes

Multiple security companies broke the hashes embedded in SUNBURST[5][6][7]. This list comes from a Check Point Research report titled “SUNBURST, TEARDROP, and the NetSec New Normal”[8].

Hashes and their mapped processes that resulted in the malware exiting:
2597124982561782591 = apimonitor-x64
2600364143812063535 = apimonitor-x86
13464308873961738403 = autopsy64
4821863173800309721 = autopsy
12969190449276002545 = autoruns64
3320026265773918739 = autoruns
12094027092655598256 = autorunsc64
10657751674541025650 = autorunsc
11913842725949116895 = binaryninja
5449730069165757263 = blacklight
292198192373389586 = cff explorer
12790084614253405985 = cutter
5219431737322569038 = de4dot
15535773470978271326 = debugview
7810436520414958497 = diskmon
13316211011159594063 = dnsd
13825071784440082496 = dnspy
14480775929210717493 = dotpeek32
14482658293117931546 = dotpeek64
8473756179280619170 = dumpcap
3778500091710709090 = evidence center
8799118153397725683 = exeinfope
12027963942392743532 = fakedns
576626207276463000 = fakenet
7412338704062093516 = ffdec

[5] Sentinel Labs. “SolarWinds SUNBURST Backdoor: Inside the APT Campaign”. https://labs.sentinelone.com/solarwinds-sunburst-backdoor-inside-the-stealthy-apt-campaign/.

[6] Symantec Enterprise Blogs. “SolarWinds Attacks: Stealthy Attackers Attempted to Evade Detection”. https://symantec-enterprise-blogs.security.com/blogs/threat-intelligence/solarwinds-attacks-stealthy-attackers-attempted-evade-detection.

[7] Sophos Labs IOCs. https://github.com/sophoslabs/IoCs/blob/master/Sunburst_blocklists.csv

[8] Check Point Research. “SUNBURST, TEARDROP, and the NetSec New Normal”. https://research.checkpoint.com/2020/sunburst-teardrop-and-the-netsec-new-normal/.

```
682250828679635420 = fiddler
13014156621614176974 = fileinsight
18150909006539876521 = floss
10336842116636872171 = gdb
12785322942775634499 = hiew32demo
13260224381505715848 = hiew32
17956969551821596225 = hollows_hunter
8709004393777297355 = idaq64
14256853800858727521 = idaq
8129411991672431889 = idr
15997665423159927228 = ildasm
10829648878147112121 = ilspy
9149947745824492274 = jd-gui
3656637464651387014 = lordpe
3575761800716667678 = officemalscanner
4501656691368064027 = ollydbg
10296494671777307979 = pdfstreamdumper
14630721578341374856 = pe-bear
4088976323439621041 = pebrowse64
9531326785919727076 = peid
6461429591783621719 = pe-sieve32
6508141243778577344 = pe-sieve64
10235971842993272939 = pestudio
2478231962306073784 = peview
9903758755917170407 = peview
14710585101020280896 = ppee
13611814135072561278 = procdump64
2810460305047003196 = procdump
2032008861530788751 = processhacker
27407921587843457 = procexp64
6491986958834001955 = procexp
2128122064571842954 = procmon
10484659978517092504 = prodiscoverbasic
8478833628889826985 = py2exedecompiler
10463926208560207521 = r2agent
7080175711202577138 = rabin2
8697424601205169055 = radare2
7775177810774851294 = ramcapture64
16130138450758310172 = ramcapture
506634811745884560 = reflector
18294908219222222902 = regmon
3588624367609827560 = resourcehacker
9555688264681862794 = retdec-ar-extractor
5415426428750045503 = retdec-bin2llvmir
3642525650883269872 = retdec-bin2pat
13135068273077306806 = retdec-config
```

```
3769837838875367802 = retdec-fileinfo
191060519014405309 = retdec-getsig
1682585410644922036 = retdec-idr2pat
7878537243757499832 = retdec-llvmir2hll
13799353263187722717 = retdec-macho-extractor
1367627386496056834 = retdec-pat2yara
12574535824074203265 = retdec-stacofin
16990567851129491937 = retdec-unpacker
8994091295115840290 = retdec-yarac
13876356431472225791 = rundotnetdll
14968320160131875803 = sbiesvc
14868920869169964081 = scdbg
106672141413120087 = scylla_x64
79089792725215063 = scylla_x86
5614586596107908838 = shellcode_launcher
3869935012404164040 = solarwindsdiagnostics
3538022140597504361 = sysmon64
14111374107076822891 = sysmon64
7982848972385914508 = task explorer
8760312338504300643 = task explorer-x64
17351543633914244545 = tcpdump
7516148236133302073 = tcpvcon
15114163911481793350 = tcpview
15457732070353984570 = vboxservice
16292685861617888592 = win32_remote
10374841591685794123 = win64_remotex64
3045986759481489935 = windbg
17109238199226571972 = windump
6827032273910657891 = winhex64
5945487981219695001 = winhex
8052533790968282297 = winobj
17574002783607647274 = wireshark
3341747963119755850 = x32dbg
14193859431895170587 = x64dbg
17439059603042731363 = xwforensics64
17683972236092287897 = xwforensics
700598796416086955 = redcloak
3660705254426876796 = avgsvc
12709986806548166638 = avgui
3890794756780010537 = avgsvca
2797129108883749491 = avgidsagent
3890769468012566366 = avgsvcx
14095938998438966337 = avgwdsvcx
11109294216876344399 = avgadminclientservice
1368907909245890092 = afwserv
11818825521849580123 = avastui
```

8146185202538899243 = avastsvc
2934149816356927366 = aswidsagent
13029357933491444455 = aswidsagenta
6195833633417633900 = aswengsrv
2760663353550280147 = avastavwrapper
16423314183614230717 = bccavsvc
2532538262737333146 = psanhost
4454255944391929578 = psuaservice
6088115528707848728 = psuamain
13611051401579634621 = avp
18147627057830191163 = avpui
17633734304611248415 = ksde
13581776705111912829 = ksdeui
7175363135479931834 = tanium
3178468437029279937 = taniumclient
13599785766252827703 = taniumdetectengine
6180361713414290679 = taniumendpointindex
8612208440357175863 = taniumtracecli
8408095252303317471 = taniumtracewebsocketclient64

Hashes and the mapped services that the malware attempts to disable:

Windows Defender:
   5183687599225757871 = msmpeng
   917638920165491138 = windefend
Windows Sense:
   10063651499895178962 = mssense
   16335643316870329598 = sense
Windows Sensor:
   10501212300031893463 = microsoft.tri.sensor
   155978580751494388 = microsoft.tri.sensor.updater
NIST:
   17204844226884380288 = cavp
Carbon Black:
   5984963105389676759 = cb
   11385275378891906608 = carbonblack
   13693525876560827283 = carbonblackk
   17849680105131524334 = cbcomms
   18246404330670877335 = cbstream
CrowdStrike:
   8698326794961817906 = csfalconservice
   9061219083560670602 = csfalconcontainer
   11771945869106552231 = csagent
   9234894663364701749 = csdevicecontrol
   8698326794961817906 = csfalconservice
FireEye:

```
    15695338751700748390 = xagt
    640589622539783622 = xagtnotif
    9384605490088500348 = fe_avk
    6274014997237900919 = fekern
    15092207615430402812 = feelam
    3320767229281015341 = fewscservice
ESET:
    3200333496547938354 = ekrn
    14513577387099045298 = eguiproxy
    607197993339007484 = egui
    15587050164583443069 = eamonm
    9559632696372799208 = eelam
    4931721628717906635 = ehdrv
    2589926981877829912 = ekrnepfw
    17997967489723066537 = epfwwfp
    14079676299181301772 = ekbdflt
    17939405613729073960 = epfw
F-SECURE:
    521157249538507889 = fsgk32st
    14971809093655817917 = fswebuid
    10545868833523019926 = fsgk32
    15039834196857999838 = fsma32
    14055243717250701608 = fssm32
    5587557070429522647 = fnrb32
    12445177985737237804 = fsaua
    17978774977754553159 = fsorsp
    17017923349298346219 = fsav32
    17624147599670377042 = f-secure gatekeeper handler starter
    16066651430762394116 = f-secure network request broker
    13655261125244647696 = f-secure webui daemon
    3421213182954201407 = fsma
    14243671177281069512 = fsorspclient
    16112751343173365533 = f-secure gatekeeper
    3425260965299690882 = f-secure hips
    9333057603143916814 = fsbts
    3413886037471417852 = fsni
    7315838824213522000 = fsvista
    13783346438774742614 = f-secure filter
    2380224015317016190 = f-secure recognizer
    3413052607651207697 = fses
    3407972863931386250 = fsfw
    10393903804869831898 = fsdfw
    3421197789791424393 = fsms
    541172992193764396 = fsdevcon
```

Drivers and their mapped processes that resulted in the malware exiting:

17097380490166623672 = cybkerneltracker.sys
15194901817027173566 = atrsdfw.sys
12718416789200275332 = eaw.sys
18392881921099771407 = rvsavd.sys
3626142665768487764 = dgdmk.sys
12343334044036541897 = sentinelmonitor.sys
397780960855462669 = hexisfsmonitor.sys
6943102301517884811 = groundling32.sys
13544031715334011032 = groundling64.sys
11801746708619571308 = safe-agent.sys
18159703063075866524 = crexecprev.sys
835151375515278827 = psepfilter.sys
16570804352575357627 = cve.sys
1614465773938842903 = brfilter.sys
12679195163651834776 = brcow_x_x_x_x.sys
2717025511528702475 = lragentmf.sys
17984632978012874803 = libwamf.sys

Appendix 3: DNS Command and Control

Subnet	Function	Summary
10.0.0.0/8 172.16.0.0/12 192.168.0.0/16 224.0.0.0/4 fc00:: - fe00:: fec0:: - ffc0:: ff00:: - ff00:: 20.140.0.0/15 96.31.172.0/24 131.228.12.0/22 144.86.226.0/24	Atm or ImpLink	Terminates the malware and updates a configuration file to prevent the malware from attempting any further communication in the future. Note that The malware uses a standard SolarWinds configuration file to store this information. If a patch overwrites this configuration file, the malware may attempt to reach out again.
41.84.159.0/24 74.114.24.0/21 154.118.140.0/24 217.163.7.0/24	Ipx	Transitions the backdoor to passive mode[9], indicating that the backdoor is still viable, but not being actively exploited. The backdoor will have to be reinitialized prior to further user.
18.130.0.0/16 99.79.0.0/16 184.72.0.0/15	NetBios	Transitions the backdoor to active mode. Provides configuration information that prepares the backdoor to move to HTTP command and control. Indicates that the malware should continue beaconing.
71.152.53.0/24 87.238.80.0/21 199.201.117.0/24	NetBios	Indicates that the backdoor should continue beaconing. If a CNAME is provided with one of these IPs, the backdoor will move to HTTP command and control.

[9] This report uses the vocabulary established by FireEye here: FireEye Threat Research. “SUNBURST Additional Technical Details”. https://www.fireeye.com/blog/threat-research/2020/12/sunburst-additional-technical-details.html

Appendix 4: HTTP Command and Control Commands

Table of commands available through HTTP command and control. FireEye provided this table in the blog post “Highly Evasive Attacker Leverages SolarWinds Supply Chain to Compromise Multiple Global Victims with SUNBURST Backdoor”, previously cited.

Command	Value	Operation
Idle	0	No operation
Exit	1	Terminate the current thread.
SetTime	2	Sets the delay time between main event loop executions Delay is in seconds, and varies random between [.9 * <delay>, 1.1 * <delay>]. If the delay is < 300 it is doubled on the next execution through the loop, this means it should settle onto an interval of around [5, 10] minutes. There is a second, unrelated delay routine that delays for a random interval between [16hrs, 83hrs]
CollectSystemDescription	3	Profile the local system including hostname, username, OS version, MAC addresses, IP address, DHCP configuration, and domain information.
UploadSystemDescription	4	Perform a HTTP request to the specified URL, parse the results and compare components against unknown hashed values. Format a report and send to the C2 server.
RunTask	5	Starts a new process with the given file path and arguments
GetProcessByDescription	6	Returns a process listing. If no arguments are provided returns just the PID and process name. If an argument is provided it also returns the parent PID and username and domain for the process owner.
KillTask	7	Terminate the given process, by PID.

GetFileSystemEntries	8	Given a path and an optional match pattern recursively list files and directories
WriteFile	9	Given a file path and a Base64 encoded string write the contents of the Base64 decoded string to the given file path. Write using append mode. Delay for [1s, 2s] after writing is done.
FileExists	10	Tests whether the given file path exists.
DeleteFile	11	Deletes the specified file path.
GetFileHash	12	Compute the MD5 of a file at a given path and return result as a HEX string. If an argument is provided, it is the expected MD5 hash of the file and returns an error if the calculated MD5 differs.
ReadRegistryValue	13	Arbitrary registry read from one of the supported hives
SetRegistryValue	14	Arbitrary registry write from one of the supported hives.
DeleteRegistryValue	15	Arbitrary registry delete from one of the supported hives
GetRegistrySubKeyAndValueNames	16	Returns listing of subkeys and value names beneath the given registry path
Reboot	17	Attempts to immediately trigger a system reboot.

Appendix 4: Indicators

Host:

Named Pipe: 583da945-62af-10e8-4902-a8f205c72b2e

SolarWinds.Orion.Core.BusinessLayer.dll SHA256:
d0d626deb3f9484e649294a8dfa814c5568f846d5aa02d4cdad5d041a29d5600

Network:
appsync-api.eu-west-1[.]avsvmcloud[.]com
appsync-api.us-west-2[.]avsvmcloud[.]com
appsync-api.us-east-1[.]avsvmcloud[.]com
appsync-api.us-east-2[.]avsvmcloud[.]com