SecureIQlab™

Bridging the enterprise cloud security gap

Detection Science: Understanding Sunburst Command & Control server Implementation.

Language:	English
Last Revision:	14th Jan, 2021

www.secureiqlab.com

Content:

REVERSING SUNBURST'S C&C SERVER IMPLEMENTATION

Last month, the world learned about Sunburst, the backdoor implanted into SolarWinds' Orion product via a supply chain compromise. Both high-level and low-level details on Sunburst can be found all over the internet, including in a previous post. In this paper, SecureIQLab will reverse enough details of Sunburst's Command & Control (C&C) to construct a rough C&C server implementation. The technical reader will be able to use this information to supplement their own threat research and improve their organization's security posture.

COMMAND AND CONTROL OVERVIEW

Command & Control is split in two parts: DNS and HTTP. The first part, DNS, is responsible for selecting targets while the second part, HTTP, implements the communication channel for the backdoor.

There are 4 phases to the C&C, which the following diagram lists:

Phase	Description
Kill (DNS)	No more DNS queries are made. Infection effectively ends.
Beacon (DNS)	DNS queries will continue until instructed otherwise. If more domain fragments need to be sent, then the subsequent DNS queries will follow.
Preactivation (DNS)	Target may be of interest, contingent on the state of some services. DNS queries encode service states instead of domain name fragments.
Activation (HTTP)	Target is of interest. Transition to HTTP based C&C.

DNS

The developer of Sunburst knew that comprising the SolarWinds supply chain would result in a wide reach. As a result, a mechanism to pick out high-value targets was necessary. To select the crème de la crème of targets, a C&C system was devised to exfiltrate identifying information over DNS queries. Prior to Command & Control, Sunburst uses a Domain Generation Algorithm (DGA) to construct a fully qualified domain name (FQDN) that follows the schema (dubbed "Type 1"):

subdomain`[.]appsync-api[.]`***region***`[.]avsvmcloud[.]com`

Where *region* is one of the following:

eu-west-1	us-west-2	us-east-1	us-east-2

And where *subdomain* is an encoded string that is derived from a *userId* and the name of the domain the victim's computer is connected to. The *userId* identifier is derived from the victim's network interface MAC address, domain name, and MachineGuid. So, the *userId* value is static and unique.

A decoding routine for *subdomain* for Type 1 FQDNs was posted by RedDrip Team shortly after Sunburst came to light. The *subdomain* string is capped to 32 bytes. The first 16 bytes encode: an XOR byte, *userId*, and a fragment index. The rest of the bytes encode a fragment of the victim's domain name. If the domain name is too long, then the domain name will be fragmented and split across multiple Type 1 FQDNs. If the domain name is fragmented, then the *userId* and the fragment index must be used to reconstruct the domain. If the fragment index is 35, then the fragment is the entire domain name or the last fragment.

To begin Command & Control, Sunburst will perform a DNS query for the Type 1 FQDN from above. The C&C DNS server will then decode the FQDN's *subdomain* to obtain the victim's *userId* and domain name fragment. If the fragment is the entire domain name, then the C&C operator would make one of 3 choices:

1. The domain is not of interest (Kill phase)
2. The domain is of unknown interest / defer the interest verdict (Beacon phase)
3. The domain is of interest (Preactivation phase)

Kill Phase

To select choice 1, the C&C DNS server would begin the "Kill phase" by replying to the DNS query with a DNS A record with an address that falls into one of the following subnets:

10.0.0.0/8
172.16.0.0/12
192.168.0.0/16
224.0.0.0/3
20.140.0.0/15
96.31.172.0/24
131.228.12.0/22
144.86.226.0/24

Beacon Phase

To select choice 2, the C&C DNS server would continue the "Beacon phase" by replying to the DNS query with a DNS A record with an address that falls into one of the following subnets:

8.18.144.0/23
87.238.80.0/21
87.238.80.0/21
71.152.53.0/24

If the fragment was not the entire domain name then the C&C operator would need to transition to the Beacon phase to receive subsequent fragments and reconstruct the domain name. Choices 1 and 2 from above are also on the table in this case but are unlikely since those choices would be the result of an uninformed decision–something that would be abnormal for this sophisticated attack.

Preactivation Phase

To select choice 3, the C&C DNS server would begin the “Preactivation phase” by replying to the DNS query with a DNS A record with an address that falls into one of the following subnets:

18.130.0.0/16
99.79.0.0/16
184.72.0.0/15

Once in the preactivation phase, queried FQDNs will now follow the schema (dubbed “Type 2”):

subdomain`[.]appsync-api[.]`***region***`[.]avsvmcloud[.]com`

Where *subdomain* this time encodes the victim’s *userId*, a timestamp, and data. The victim’s domain name is no longer included. The encoding (and decoding) routine for subdomain for Type 2 FQDNs is different from that from Type 1 FQDNs. The new decoding routine has been posted online by sysopfb. The data that was encoded is a 2-byte bitfield that represents the state of 8 services of the below security products:

Security Product Service
Windows Defender
Windows Defender ATP
Defender for Identity
Carbon Black
CrowdStrike Falcon
FireEye
ESET
F-Secure

When the C&C DNS server receives a DNS query for a Type 2 FQDN, the C&C operator will use the product service states to make a more informed decision on whether or not to continue the infection. If the infected system is still of interest, the operator will begin the “activation” phase by replying to a DNS query, sent while in the preactivation phase, with a DNS CNAME record with a hostname value for the C&C HTTP server, optionally prepended with a URI schema (http[s]://). If no schema is specified, the https schema will be used.

HTTP

Once in the Activation phase, Command & Control moves from using DNS to using HTTP.

Activation Phase

Sunburst uses HTTP(S) as the channel to communicate backdoor job commands and results.

Job Type	Job ID
Idle	0
Exit	1
SetTime	2
CollectSystemDescription	3
UploadSystemDescription	4
RunTask	5
GetProcessByDescription	6
KillTask	7
GetFileSystemEntries	8
WriteFile	9
FileExists	10
DeleteFile	11
GetFileHash	12
ReadRegistryValue	13
SetRegistryValue	14
DeleteRegistryValue	15
GetRegistrySubKeyAndValueNames	16
Reboot	17
None	18

To commence the HTTP C&C activity, a GET or HEAD request is sent to the C&C HTTP server. In the request, the header "`If-None-Match:` *`encodedUserId`*" `is added`, where *encodedUserId* encodes the victim's *userId*. The *encodedUserId* value is decoded using the algorithm:

1. Split *encodedUserId* into 2 hex strings of equal length.
2. Convert both hex strings into byte arrays.
3. XOR each byte of the first array with the 3rd, 4th, 5th, 6th byte of the second array in a cyclical fashion.

The C&C HTTP server will use steganography in responses to encode backdoor jobs in benign-looking content. The encoding algorithm is the following:

1. Compress and encode the string "*job_id job_parameters*" using *`OrionImprovementBusinessLayer.HttpHelper.Inflate`*, where *job_id* values are in the job table above and where *job_parameters* is the job's parameters.
2. Calculate the size of the inflated bytes from step 1 and prepended it as a DWORD to the inflated byte array.
3. Convert the byte array to a hex string and distribute, in order, across GUIDs, 32-length hex strings, and 16-length hex strings.

There are infinitely many different templates that can be used to construct the output of step 3. According to FireEye, one template used in the wild was one that resembles an Assembly Manifest.

The backdoor will periodically check for any queued jobs by sending a request to the C&C server. To keep the backdoor channel alive, the C&C HTTP server must send a job other than Idle, None, Exit, Reboot at least once every 3 requests. The SetTime job can be used to effectively increase the maximum number requests between jobs temporarily.

The result of some job commands is sent back to the C&C HTTP server in the next request using the POST or PUT method. If the compressed result is greater than 10000 bytes, then the job result is sent back encoded and compressed in a POST request. Otherwise, the job result is sent back encoded within JSON in a PUT request.

Once the two-way backdoor channel has been established, the C&C operator can move from the Command & Control phase of the kill chain to the Objective phase.

SERVER IMPLEMENTATION

Using what was covered here, we were able to construct a basic implementation of Sunburst's C&C DNS and HTTP servers. The code can be found here. In the repo there are 3 things:

- A Visual Studio solution for SolarWinds.BusinessLayerHost.exe (the Sunburst loader) and SolarWinds.Orion.Core.BusinessLayer.dll (the Sunburst backdoor). Modifications, such as shortening delays, were made to the backdoor for testing convenience.
- A Python3 implementation of the C&C DNS server.
- A Python3 implementation of the C&C HTTP server.

Sunburst initially uses DNS-based C&C. Knowing these details about Sunburst's C&C allows us to improve our security by incorporating this information into security products or by creating our own honeypots to detect infected machines and stop future attacks.

COPYRIGHT AND DISCLAIMER

For more information about SecureIQLab and the testing methodologies, please visit our website.

SecureIQLab (January 2021)