January 5th, 2020

Industry Security Advisory

SolarWinds Orion API

Vulnerability Advisory

Cyberint

EXECUTIVE SUMMARY

Supplementing the SolarWinds Security Bulletin released in mid-December 2020, detailing a suspected nation-state threat actor introducing a backdoor into SolarWinds Orion versions 2019.4 HF5, 2020.2 and 2020.2 HF1, this bulletin provides an update based on recent observations in late December 2020 and early January 2021.

In the first instance, the public release of a proof-of-concept (PoC) local file disclosure/inclusion (LFD/LFI) exploit on 28 December 2020 allows configuration and credentials to be stolen and as such has lead to multiple threat actors conducting widespread scanning activity in order to identify and target vulnerable SolarWinds Orion installations.

Given the increase in scanning activity along with widespread press and social media coverage, organizations are again reminded to follow earlier recommendations and isolate vulnerable hosts before updating SolarWinds Orion to the latest version as soon as possible. Furthermore, those that find themselves with vulnerable installations should take steps to investigate a potential breach.

PROOF-OF-CONCEPT LOCAL FILE DISCLOSURE

Published to GitHub 28 December 2020 as a `Gist` [1] by a known security researcher named `0xsha`, the proof-of-concept (PoC) local file disclosure/inclusion (LFD/LFI), written in Python, allows a vulnerable installation to be determined before attempting to gather both configuration data and credentials (Figure 1).

```
                          python CVE-2020-10148.py http://
[*] Trying to leak valid file version
[+] Got location header
[+] Version seems valid
[*] Trying to leak web.config file
[+] Target is vulnerable Got the web.config file
[+] web.config written to :                web.config
[*] Trying to leak SWNetPerfMon.db file (works only on older versions of orion)
[+] Target is vulnerable Got the SWNetPerfMon.db file
[+] SWNetPerfMon.db written to :                SWNetPerfMon.db
```

Figure 1 - PoC Python script execution

IMPACT

The impact arising from the initial SolarWinds Orion vulnerability, tracked as CVE-2020-10148 and exploited in a campaign dubbed 'SUNBURST' remains **severe** and is compounded by the subsequent use of the web shell threat 'SUPERNOVA' along with the public release of proof-of-concept code to gather configuration and credential data from vulnerable installations.

As such, the combination of both the vulnerability and the exploit could allow a malicious threat actor to obtain the SolarWinds Orion password database, chaining the authentication bypass vulnerability together with arbitrary file read resulting in local file disclosure.

In addition to vulnerable hosts being easily identifiable from open source searches, such as the query `http.title:solarwinds http.favicon.hash:-1776962843` on Shodan [5] (Figure 2), increased internet-wide scanning activity has been observed.

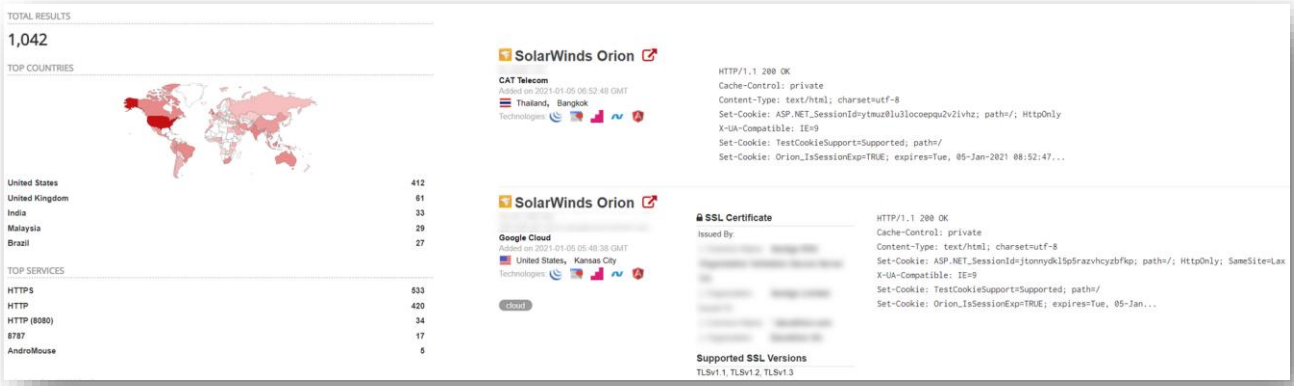

Figure 2 - Shodan search results

Affected versions of SolarWinds Orion remain the same as the initial vulnerability announcement [6]:

- 2019.4 HF 5
- 2020.2 (No hotfix)
- 2020.2 HF 1

The initial vulnerability is considered **critical**, potentially leading to full server compromise, and was assigned a CVSS v3 Base Score of 9.8.

VULNERABLE FILE VERSION

Having requested the `/Orion/invalid.aspx.js` file, the location header is queried to determine if a valid version can be found (Figure 3).

```
77  leakVersion = requests.get(target+"/Orion/invalid.aspx.js" ,verify=False)
78
79  if(leakVersion.headers["location"]):
80      print("[+] Got location header")
81      index = leakVersion.headers["location"].index(".i18n.ashx")
82      leakedVersion = (leakVersion.headers["location"][index:])
```

Figure 3 - Leak File Version

WEB CONFIG

Once confirmed as vulnerable, the PoC exploit will attempt to retrieve the web.config file used in IIS and SolarWinds Orion potentially leading to the exposure of sensitive server configuration data (Figure 4):

```
94  leakedConfig = requests.get(target+"/web.config"+leakedVersion, verify=False)
```

Figure 4 - Leak Web Config

PASSWORD DATA

The `SWNetPerfMon.db` file contains, according to SolarWinds, database connection information including credentials and is stored locally on the server in the following files:

- `C:\\inetpub\\SolarWinds\\SWNetPerfMon.db`
- `C:\\Program Files (x86)\\SolarWinds\\Orion\\SWNetPerfMon.db`

Utilizing the PoC, this file can be accessed remotely on a vulnerable installation (Figure 5) and would subsequently allow the file to be downloaded.

```
114  leakedDB = requests.get(target+"/SWNetPerfMon.db"+leakedVersion, verify=False)
```

Figure 5 - Leak Password Data

Once downloaded, a threat actor would then be able to decrypt the `SWNetPerfMon.db` file using publicly available tools, such as a 'Credential Dumping Tool' shared by a researcher named Rob 'Mubix' Fuller [2] and as documented in various articles [3] [4], leading to plain text output (Figure 6) containing both the username and password of SolarWinds Orion users.

```
| Type: SolarWinds.Orion.Core.SharedCredentials.Credentials.UsernamePasswordCredential
| Name: DomainAdmin
|       Desc:
|       Owner: Orion
|               Password:
|               Username:
```

Figure 6 - Mubix Credential Dumping Tool output

DETECTION

As part of the exploitation process, threat actors will attempt to retrieve the following files using GET requests and as such HTTP access attempts for the targeted files should be monitored and reviewed:

- `/web.config.i18n.ashx?l=en-US&v=[leakedVersion]`
- `/SWNetPerfMon.db.i18n.ashx?l=en-US&v=[leakedVersion]`

Recommendations

- As previously detailed, organizations with vulnerable SolarWinds Orion installations should update as soon as possible:
  - `2018.2, 2018.4 & 2019.2 SUPERNOVA Patch`
  - `2019.4 HF6`
  - `2020.2.1 HF2`
- Aside from checking the software installed, a HTTP GET request to `/Orion/invalid.aspx.js` on a SolarWinds Orion server will identify the version to determine if it is vulnerable.
- Access logs should be monitored and reviewed for unexpected or suspicious access attempts to the `SWNetPerfMon.db` and `web.config` files.
- Consideration should be given to explicitly blocking access to vulnerable pages and the named files.

REFERENCES

[1] https://gist.github.com/0xsha/75616ef6f24067c4fb5b320c5dfa4965

[2] https://github.com/mubix/solarflare

[3] https://www.atredis.com/blog/2018/10/24/fun-with-the-solarwinds-orion-platform

[4] https://malicious.link/post/2020/solarflare-release-password-dumper-for-solarwinds-orion/

[5] https://www.shodan.io/search?query=http.title%3Asolarwinds+http.favicon.hash%3A-1776962843

[6] https://www.solarwinds.com/securityadvisory