Technical Reference Guide

CVE-2020-11651
and
CVE-2020-11652

May 2020

Introduction

This document provides technical guidelines for applying CVE-2020-11651 / CVE-2020-11652 patches and performing exploitation assessment and check up analysis.

Please note: The information shared within this document is intended to support client efforts in performing the patch process, verifying application of the remediation and providing supporting insights as to how to determine if an exploitation of your environment had occurred.

We answer the following questions:

- Do I need to patch?
- Where do I find the patch?
- How do I apply the patch?
- How can I ensure the patch worked?
- Am I still vulnerable after applying updates or patches?
- How do I know if I was compromised before patching?

Do I need to patch?

All current versions of Salt and versions dating back to version 0.15 of Salt are affected. This means everyone running Salt masters needs to either upgrade or patch. If choosing to upgrade, you must upgrade to the 2019.2.4 or 3000.2 versions of Salt.

Attention:
Immediately STOP the service on all unpatched, exposed masters, and then update and restart all Salt masters.

Where do I find the patch?

It is recommended that you upgrade your Salt masters to a currently supported version which already contains the patch. The patched versions are: 2019.2.4 or 3000.2. If you plan to upgrade, follow the instructions at: https://repo.saltstack.com

If you are unable to upgrade your Salt master, you will need to obtain a patch file for your version. The following command will tell you the version of your Salt master:

```
salt --version
```

Request a patch for your version at: https://www.saltstack.com/lp/request-patch-april-2020/

How do I apply the patch?

Once you have the patch file for your version of Salt, you can use the patch utility to apply the patch against your Salt installation.

> **Note**
> The following section assumes that all commands are being **run as the root user.**

If you're not sure where your Salt installation lives, run the following command:

```
salt-run salt.cmd grains.get saltpath
```

This command returns the path where Salt is installed. For example:

```
/usr/lib/python2.7/dist-packages/salt
```

In this example, the command to apply your patch would be the following:

To verify with a trial run:

```
# patch -p2 -d  /usr/lib/python2.7/dist-packages/salt/ --dry-run <
Fix-CVE-2020-11651-and-Fix-CVE-2020-11652-XXXX.patch
```

To apply the patch:

```
patch -p2 -d  /usr/lib/python2.7/dist-packages/salt/ <
Fix-CVE-2020-11651-and-Fix-CVE-2020-11652-XXXX.patch
```

To check the return code of the patch command:

```
 echo $?
```

> **Note**
> The echo command should return the number 0.

To restart the master:

```
 systemctl restart salt-master
```

How can I ensure the patch worked?

You can verify the patches are effective using any of the following methods:

Check the output and return code from the patch command:

```
# patch -p2 -d  /usr/lib/python2.7/dist-packages/salt/ <
Fix-CVE-2020-11651-and-Fix-CVE-2020-11652-XXXX.patch

checking file master.py
Hunk #1 succeeded at 1053 (offset 3 lines).
Hunk #2 succeeded at 1077 (offset 3 lines).
Hunk #3 succeeded at 1102 (offset 3 lines).
Hunk #4 succeeded at 1853 (offset 3 lines).
checking file tokens/localfs.py
checking file utils/verify.py
checking file wheel/config.py
checking file wheel/file_roots.py

 # echo $?

 0
```

> **Note**
> As long as the echo command returns the number 0, the patch was applied successfully.

Grep for 'expose_methods' in the Salt installation location.

```
# grep expose_methods /usr/lib/python3/dist-packages/salt/master.py > /dev/null; echo $?
0
```

> **Note**
> As long as this command returns 0, the patch was applied successfully.

Am I still vulnerable after applying updates or patches?

If you have updated your Salt master to a recent supported version or applied patches to your master, you are no longer vulnerable to attacks using the exploits found in CVE-2020-11651 and CVE-2020-11652. However, you should ensure that you were not compromised before patching.

How do I know if I was compromised before patching?

You can check for compromises using the following methods:

Check for vulnerabilities using rootkit checkers. The following are some recommended rootkit checkers:

https://www.tecmint.com/scan-linux-for-malware-and-rootkits/
https://www.computerworld.com/article/3412285/best-anti-rootkit-tools.html
http://www.chkrootkit.org/

Checking for known CVE-2020-11651 and CVE-2020-11652 attacks. The earliest widespread attack in the wild is a crypto miner. If you were exploited with this attack you have likely seen all of your services shut down and your syslogs turned off.

Some additional checks:

- Your minions have probably been affected and turned into crypto-mining rigs. Depending on which versions of the exploit you've been attacked with, there may be other backdoors (RATs) installed.
- In order to make itself persistent the virus that is utilizing this exploit has been writing an entry into the crontab for the root user which attempts to download and update the virus. To ensure that machines are not re-infected please check the output from `crontab -l` for any unfamiliar entries. Some versions of the attack will infect other crontabs, not just the root user. You can search across all cron with `cd /var/spool/cron/ && grep -r . *`
- `ls /tmp` - Look for directories under the "/tmp" directory with names like "salt-store".
- `less /var/log/salt/master` - Look in `/var/log/salt/master` for unrecognized jobs, especially those with errors may indicate probing attempts
- Check the output of the jobs runner for any recent Salt jobs that fall outside of normal Salt usage with the command, `salt-run jobs.list_jobs`.
- The virus is changing a number of system settings, stopping all system services, and running a cryptocurrency miner. Check the output of `ps aux` as well as `top` for unrecognized processes, especially with high CPU usage.

Audit the commands that were run. We recommend auditing the commands run on both the master and minion to investigate if there are any unexpected or unrecognized commands. There are a couple of ways to determine what commands have been run from your master:

1. *Audit Master Log Files:* Audit the Salt Master log file (/var/log/salt/master) for any jobs run. For example:

```
2020-05-04 18:24:12,466 [salt.client.mixins                :421 ][INFO    ][841653] Runner completed: 20200504182412456691
2020-05-04 18:24:12,472 [salt.master                       :2346][INFO    ][841649] User root Published command test.ping with jid 20200504182412468729
```

   If you have debug logs enabled you will be able to see more detailed information about the jobs being run.

```
2020-05-04 18:24:12,455 [salt.utils.event                  :737 ][DEBUG   ][841653] Sending event: tag = salt/wheel/20200504182412454895/new; data = {'fun': 'wheel.key.finger', 'jid': '20200504182412454895', 'tag': 'salt/wheel/20200504182412454895', 'user': 'root', '_stamp': '2020-05-04T22:24:12.455122'}
```

2. *Audit Minion Log Files:* Audit the Salt Minion Log File (/var/log/salt/minion) for any jobs run. For example:

```
2020-05-04 17:51:13,643 [salt.minion                       :1482][INFO    ][846500] User root Executing command test.version with jid 202003100000000000
```

   If you have debug logs enabled you will be able to see more detailed information about the job being run.

```
2020-05-04 18:28:44,841 [salt.minion                       :1489][DEBUG   ][846500] Command details {'arg': [], 'cmd': '_send_pub', 'fun': 'test.version', 'jid': '20200504175309735193', 'kwargs': {'show_jid': False, 'show_timeout': False}, 'ret': '', 'tgt': '*', 'tgt_type': 'glob', 'user': 'root'}
```

3. *Get a view of jobs executed.* Use the following command on the Salt master:

```
salt-run jobs.list_jobs
```

If you're running a Salt Master with a version earlier than 2019.2.4, 3000.2 or have not applied any of the patches available for 2015.8.10, 2016.3.x 2016.11, 2017.7, 2018.3 releases, you may not have been compromised, but we advise upgrading as soon as possible.

Now what?

Turn off any unpatched exposed masters. If you have a mitigation plan in place, we advise you to start the process of implementing the patch.

At minimum, we advise:

1. Turn off any masters exposed to the Internet. If there's an indication of a possible compromise, we advise powering down all affected machines.
2. Take snapshots of the filesystem for forensics.
3. If feasible, build a new master with an up-to-date OS and patched Salt.

If it is not feasible to build a new Salt master:

1. Ensure that the Salt master has been updated and restarted, including applying any and all outstanding operating system updates and patches.
2. Ensure that the Salt master is not exposed to the Internet at large.
3. Ensure that known artifacts mentioned above are no longer present, such as crontab entry and salt-store directories.
4. Continue monitoring your infrastructure for unexpected applications or behaviors.