

RESOLVED: StackFull Software Splunk Log Access Issue

Incident Reported: *May 24, 2023*

Incident Resolved: *May 24, 2023*

SOC Analyst Team Manager:

Jamar

Training Supervisor:

Alice, Level 2 SOC Analyst

Analyst Solving the Problem:

Will Schmidt, Level 1 SOC Analyst

Report written by Will Schmidt. Please reach out to him directly if you have any questions, or want further explanation on anything mentioned in this report.

wschmidt1988@gmail.com / @wschmidt on Slack / 314-610-2954

Executive Summary

As part of her training regimen, Alice grants new SOC analysts on the team access to Splunk, where we're able to view many different logs like firewall logs, Windows Event logs, Jira logs, and software engineering logs.

After Alice granted me access to Splunk, we discovered that there were configuration issues preventing us from searching these various log files. The issue was with a configuration file (config.conf) that had been inadvertently changed to prevent us from viewing the logs.

In this report, you'll see how this issue was resolved. Additionally, we'll discuss ways that StackFull Software can tighten their overall security protocols and prevent future issues like this by modifying user permissions on specific files.

Identifying the Location of the config.conf File

Step one was for us to locate the proper config.conf file. Since there are a few different options, we needed to be sure that the one we were going to modify was the one that had been previously changed inadvertently.

To do this, we began with `< locate config.conf >`, which pulled up the following list in our terminal:

A terminal window with a dark background and light text. The prompt is 'fstack@ubuntu: /'. The command 'locate config.conf' has been entered, and the output lists several files: /home/fstack/.config/neofetch/config.conf, /home/fstack/Documents/config.conf, /opt/splunk/etc/system/local/config.conf, /var/lib/dpkg/info/fontconfig-config.conf, /var/lib/dpkg/info/im-config.conf, /var/lib/dpkg/info/libsensors-config.conf, /var/lib/dpkg/info/motd-news-config.conf, and /var/lib/dpkg/info/pkg-config.conf. The prompt is now 'fstack@ubuntu: /\$' with a cursor.

```
fstack@ubuntu: /$ locate config.conf
/home/fstack/.config/neofetch/config.conf
/home/fstack/Documents/config.conf
/opt/splunk/etc/system/local/config.conf
/var/lib/dpkg/info/fontconfig-config.conf
/var/lib/dpkg/info/im-config.conf
/var/lib/dpkg/info/libsensors-config.conf
/var/lib/dpkg/info/motd-news-config.conf
/var/lib/dpkg/info/pkg-config.conf
fstack@ubuntu: /$
```

The relevant files our command returned in the terminal.

The list reads as follows:

- /home/fstack/.config/neofetch/config.conf
- /home/fstack/Documents/config.conf
- /opt/splunk/etc/system/local/config.conf
- /var/lib/dpkg/info/fontconfig-config.conf
- /var/lib/dpkg/info/im-config.conf
- /var/lib/dpkg/info/libsensors-config.conf
- /var/lib/dpkg/info/motd-news-config.conf
- /var/lib/dpkg/info/pkg-config.conf

We focused on the path highlighted above because we know that Splunk stores all of its files in the /opt/splunk directory. Given that knowledge, we knew that was where the proper config.conf file was located.

So, we changed to the /opt/splunk/etc/system/local/config.conf directory to investigate further:

A terminal window with a dark background and light text. The prompt is 'fstack@ubuntu: /opt/splunk/etc/system/local'. The command 'ls -l' has been entered, and the output shows a single file: '-rwxrwxrwx 1 root root 223 May 24 17:28 config.conf'. The prompt is now 'fstack@ubuntu: /opt/splunk/etc/system/local\$' with a cursor.

```
fstack@ubuntu: /opt/splunk/etc/system/local$ ls -l
total 4
-rwxrwxrwx 1 root root 223 May 24 17:28 config.conf
fstack@ubuntu: /opt/splunk/etc/system/local$
```

Navigating into the /opt/splunk/etc/system/local/config.conf directory.

Checking Permissions on the config.conf File

After navigating into the `/opt/splunk/etc/system/local/config.conf` directory, we immediately checked permissions on the `config.conf` file. We ran the `< ls -l >` command to list the directory with the permissions:

```
-rwxrwxrwx 1 root root 223 May 24 17:28 config.conf
```

What this image tells us is the following:

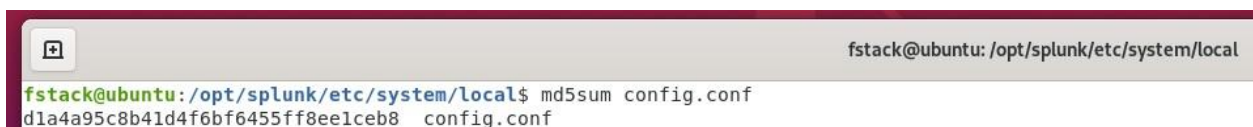
- **Permissions: Users, groups, and world all have read, write, and execute permissions**
- **Hardlinks: There is only 1 hard link to the file**
 - Meaning: There is 1 link that exists between the filename and the actual data stored on the filesystem
- **User: Root—this file is owned by the root user**
- **Group: Root—this file is owned by the root group**
- File Size: 185 bytes
- Date: Created on 9-29-2022
- Name: File name is `config.conf`

The key information here has been highlighted above. First, this file is owned by root. Second, anyone who runs this file can read, write, or execute. When you put those two things together, you understand that anyone in the world who accesses this file, whether they read, write, or execute, can do so as root.

To be clear, this is a nightmare scenario in the security world. Thankfully, it's easily fixed and we can implement some new security protocols to ensure it doesn't happen in the future (more on that later).

Checking the md5 Hash

After examining permissions, we checked the md5 hash on `config.conf` with the `< md5sum >` command. This returned the output: `d1a4a95c8b41d4f6bf6455ff8ee1ceb8`



```
fstack@ubuntu: /opt/splunk/etc/system/local$ md5sum config.conf
d1a4a95c8b41d4f6bf6455ff8ee1ceb8  config.conf
```

The md5 hash on our initial config.conf file.

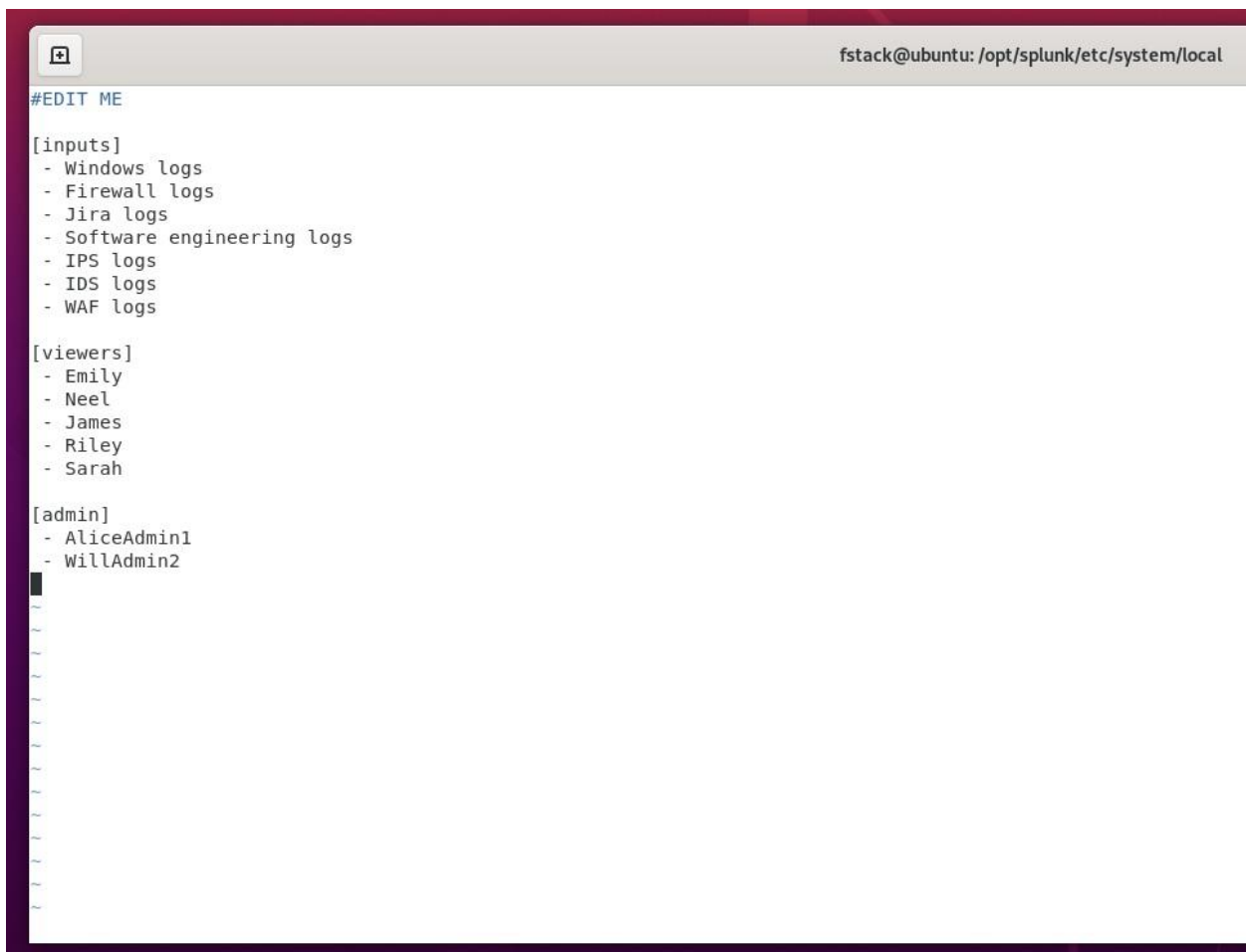
For additional context, finding the md5 hash is important because it helps us verify the integrity of a file. When we run `< md5sum >` we get a specific output, like we did with the `config.conf` file above.

However, if we were to change anything in the file, even just one letter in a line of strings, it would completely alter the entire md5 hash. The output is 100% different.

So, we can compare hashes and know, for a certainty, if the file has been altered in any way. It could be someone on our team, or it could be a bad actor trying to exploit our systems.

Case in point, Alice and I had to append the `config.conf` file with a few additional lines that would allow us access to look at the log files:

- [admin]
 - AliceAdmin1
 - WillAdmin2



```
fstack@ubuntu: /opt/splunk/etc/system/local
#EDIT ME

[inputs]
- Windows logs
- Firewall logs
- Jira logs
- Software engineering logs
- IPS logs
- IDS logs
- WAF logs

[viewers]
- Emily
- Neel
- James
- Riley
- Sarah

[admin]
- AliceAdmin1
- WillAdmin2
```

The updates we added to the `config.conf` file in vim.

After adding that text, we saved the new file and ran the `< md5sum >` command again to get the new hash for the file: `8a51d7b04ba0b82b5308e4f4e6968a2b`

```
fstack@ubuntu: /opt/splunk/etc/system/local
fstack@ubuntu: /opt/splunk/etc/system/local$ md5sum config.conf
8a51d7b04ba0b82b5308e4f4e6968a2b  config.conf_
```

The md5 hash on our updated config.conf file.

Do you see how the two hashes are vastly different from each other, despite only small changes?

- **Initial hash:** `d1a4a95c8b41d4f6bf6455ff8ee1ceb8`
- **Updated hash:** `8a51d7b04ba0b82b5308e4f4e6968a2b`

Backing Up the config.conf File

The last thing for us to do was to backup the updated config.conf file to the `/home/fstack` directory. We ran the `< cp >` command to copy it there:

```
fstack@ubuntu: /opt/splunk/etc/system/local
fstack@ubuntu: /opt/splunk/etc/system/local$ sudo cp config.conf /home/fstack
```

Copying the config.conf file to our /home/stack directory.

Then, we verified the file had made it to the `/home/fstack` directory:

```
fstack@ubuntu: ~
fstack@ubuntu: /opt/splunk/etc/system/local$ cd /home/fstack
fstack@ubuntu: ~$ ls -l
total 56
drwxrwxr-x 2 fstack fstack 4096 Aug 31 2022 Desktop
drwxr-xr-x 2 fstack fstack 4096 Sep 29 2022 Documents
drwxr-xr-x 3 fstack fstack 4096 Sep 26 2022 Downloads
drwxr-xr-x 2 fstack fstack 4096 Aug 31 2022 Music
drwxr-xr-x 2 fstack fstack 4096 Aug 31 2022 Pictures
drwxr-xr-x 2 fstack fstack 4096 Aug 31 2022 Public
drwxr-xr-x 2 fstack fstack 4096 Aug 31 2022 Templates
drwxr-xr-x 2 fstack fstack 4096 Aug 31 2022 Videos
-rwxr-xr-x 1 root root 224 May 24 18:38 config.conf
drwxr-xr-x 2 fstack fstack 4096 Aug 31 2022 demo1
drwxr-xr-x 2 fstack fstack 4096 Aug 30 2022 demo2
drwxr-xr-x 32 fstack fstack 4096 May 23 23:43 practice
-rwxr-xr-x 1 fstack fstack 272 Aug 31 2022 sample.sh
drwxr-xr-x 17 fstack fstack 4096 Aug 31 2022 ubuntu
```

Showing the `ls -l` contents of the `/home/fstack` directory confirms config.conf is there.

Security Improvements

The biggest improvement we have to offer is around file permissions. As mentioned above, the `config.conf` file had read, write, and execute privileges for user, group, and world. In other words, that file was saying:

“Hey everyone, feel free to read my contents, write new contents, or execute my script.”

Consider if a bad actor were to access the system and have permissions to read sensitive files? Write malicious scripts? Run executable commands?

Thankfully, we can work to avoid this by ensuring only authorized users are allowed to access files and modify them. In the industry, this is what’s known as the Principle of Least Privilege.

We only give access to a resource to those who actually need it. Users should not be given root or administrator access, and we shouldn’t run everything as root or administrator.

To help, here’s a security implementation we can use. On Splunk configuration files, we can set permissions as follows:

```
< sudo chmod 744 config.conf >
```

With this command, we’re saying:

- Users can read, write, and execute
- Group can read
- World can read

```
└─rw-r--r-- 1 root  root  224 May 24 18:38 config.conf
```

The `config.conf` file is now safely and securely updated in our system, and proper access has been restored to all the members on our team who need access.