## The Problem

An issue was identified with the Splunk config file known as config.conf. This file prevented the proper viewing of logs. Investigation revealed that a Level 1 SOC analyst, James, (not to throw anyone under the bus) had unintentionally modified the config.conf file, causing a configuration problem. This resulted in authorized users unable to search and analyze crucial log data.

## How I solved the problem

***Documented course of action below with snapshots of evidence***

1. Access to Unit 2 VM established using SSH (predefined in scenario)
2. Located the config.conf file within /opt/splunk directory
3. config.conf file permissions were checked. This is where vulnerability was found.
4. Hash was verified to track changes
5. File was edited to show admins in script per direction: Added admins.
6. File permissions were also edited in accordance with industry practice of role-based access control.
7. New hash created to confirm the integrity of the modified file
8. A back up config.conf was created in the /home/fstack/ directory.

## How Stackfull Software can Improve confidentiality of the Splunk configuration file so only authorized users can modify it

To improve confidentiality and ensure only authorized users can modify it, Stackfull Software can implement the following:

1. Implement Role based access control to define roles and assign permissions to suers based on their responsibilities.
2. Consider encrypting sensitive configuration files to add an extra layer of protection and making it harder for unauthorized users to make changes.

## How the md5sum command can be used for file integrity monitoring

1. Periodically run the command on critical files and compare with a baseline hash to identify any discrepancies .
2. Integrate automated script or monitoring tools to perform regular checks for integrity and provide prompt notice.
3. Configure an alerting and logging system when these changes are made to MD5 hash.

## Course of Action

1. find /opt/splunk/ -name config.conf
   located

2. CD'd into path of config.conf and cat'd

```
fstack@ip-172-31-47-29:~$ find /opt/splunk/ -name config.conf
/opt/splunk/etc/system/local/config.conf
fstack@ip-172-31-47-29:~$ cd /opt/splunk/etc/system/local/
fstack@ip-172-31-47-29:/opt/splunk/etc/system/local$ ls
config.conf
fstack@ip-172-31-47-29:/opt/splunk/etc/system/local$
```

```
fstack@ip-172-31-47-29:/opt/splunk/etc/system/local$ cat config.conf
#EDIT ME


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

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

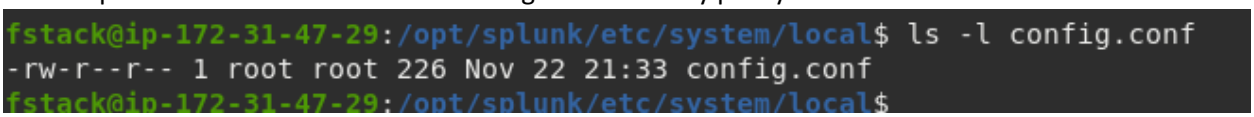
3. ls -l on config.conf. Found the permissions fully accessible to groups and users with rwx. Owned by root

```
fstack@ip-172-31-47-29:/opt/splunk/etc/system/local$ ls -l config.conf
-rwxrwxrwx 1 root root 226 Nov 22 21:33 config.conf
```

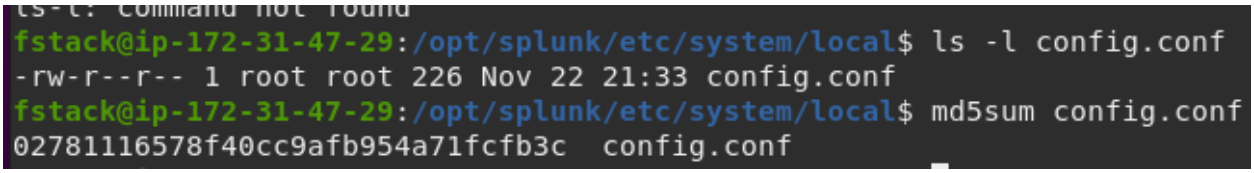
4. Created initial hash on config.conf using md5sum config.conf
   c70754d9c7bab08a8c441f90c37f27eb  config.conf
5. Used vi to edit config.conf-
   added: [admin]
   - AliceAdmin1
   - ChrisAdmin2

```
#EDIT ME


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

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

[admins]
 - AliceAdmin1
 - ChrisAdmin2
~
"config.conf" [readonly] 22L, 226C
```

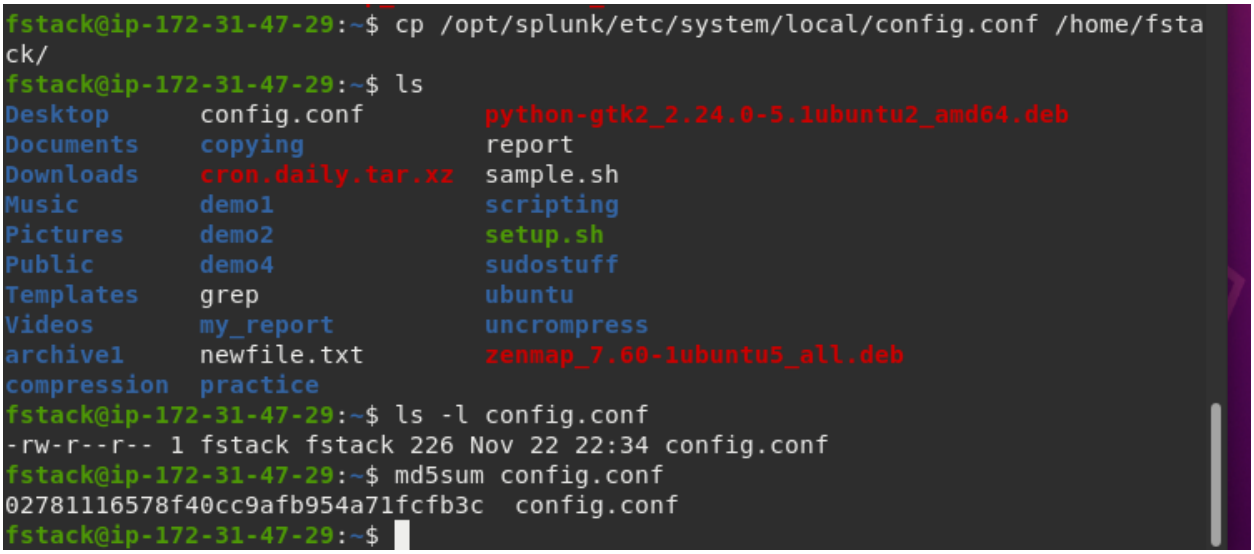
6. Chmod 644 config. File to restrict permissions based on config.conf note to 644 to ensure groups and users have read permissions and only admins can write. This is established in order to prevent future incidences and to align with security policy. Cd




7. Mdfsum config.conf and established new hash. Confirmed hashes are different 02781116578f40cc9afb954a71fcfb3c config.conf




8. Cp file into home directory to create back up and checked hash as well as permissions to ensure it was the same




Sincerely,

Chris Dayao