Ensuring the Confidentiality and Integrity of the Splunk Configuration File

[Sajjad Sheykhi], Course 2 Project

December 19, 2024

Executive Summary

StackFull Software recently experienced a configuration issue within its Splunk environment that prevented our security team from searching and reviewing critical log data. This report details the problem encountered, the steps taken to resolve it, and recommendations for strengthening the file's confidentiality and integrity. By implementing these measures, StackFull Software will better safeguard its critical Splunk configurations and maintain consistent, secure access to vital security logs.

The Problem

Splunk's log search functionality became unavailable due to an improperly modified configuration file (`config.conf`). The file had been inadvertently altered, resulting in corrupted settings that restricted our ability to analyze system logs and identify potential security incidents. Additionally, the file's permissions were too broad, increasing the risk that unauthorized personnel could modify it.

Impact

- Inability to perform timely and effective log analysis.
- Heightened security risk due to possible unauthorized changes.
- Interruption of normal security monitoring operations.

How the Problem Was Solved

1. **Identification of the File Location:** Using file search commands under `/opt/splunk`, we located the `config.conf` file in `/opt/splunk/etc/system/local/`.

2. **Permission and Integrity Assessment:** We reviewed the file's permissions and found them too permissive. Before making changes, we calculated the file's MD5 hash to establish a baseline for integrity monitoring.

3. **Correcting the Configuration:** We edited `config.conf` to restore required entries. Specifically, we appended:

   ```
   [admin]
   AliceAdmin1
   [YourName]Admin2
   ```

   After saving these changes, we recalculated the MD5 hash to confirm the file was intentionally modified. and it changed into : **MD5 Before Modification:** c70754d9c7bab08a8c441f90c37f27eb config.conf
   MD5 After Modification: 8db8821a81f9b8075cf0f385c38a8517

4. **Creating a Secure Backup:** We copied the updated configuration file to `/home/fstack`, ensuring a known-good version of the file is securely stored for future reference.

Upon completing these steps, Splunk functionality was fully restored, allowing our security team to resume normal log searches and analyses.

Improving Confidentiality of the Configuration File

To ensure that only authorized individuals can modify the Splunk configuration file, we recommend the following measures:

Restrictive File Permissions

Limit write access to the file's owner (such as the `splunk` user) and remove write permissions from the group and others. For example:

```
chmod 640 /opt/splunk/etc/system/local/config.conf
chown splunk:splunk /opt/splunk/etc/system/local/config.conf
```

This ensures that only the designated user has the ability to change critical Splunk configurations.

Role-Based Access Control (RBAC)

Implement RBAC to strictly define and enforce who can access and modify configuration files. Assign roles so that only certain administrators or security staff can adjust Splunk settings.

Access Control Lists (ACLs) and Auditing

Consider using ACLs for finer-grained permission control and maintain audit logs of any changes made to critical configuration files, providing an additional layer of accountability.

Using `md5sum` to Monitor File Integrity

The `md5sum` command generates a unique hash value for a file based on its contents. By regularly recording and comparing the MD5 hash of `config.conf`, unauthorized modifications can be detected quickly.

Recommended Approach

- **Baseline Hash Recording:** Take an MD5 hash of `config.conf` after verifying its correctness and record this baseline hash.
- **Regular Integrity Checks:** Periodically run `md5sum config.conf` and compare the output to the baseline. Any discrepancy prompts a review to ensure changes are authorized.
- **Automation and Alerts:** Integrate regular hash checks into automated scripts or security tools, and configure alerts to notify the security team if the hash does not match the baseline.

Conclusion

By resolving the `config.conf` issue and adjusting its permissions, StackFull Software can now securely and efficiently monitor Splunk logs. Moving forward, implementing tighter access controls and utilizing MD5 hash checks will strengthen the confidentiality and integrity of this critical configuration file. These proactive measures will help ensure the consistent and secure functioning of our cybersecurity operations.