My concern with this advice from TM is that it does not address the issue of why this happened in the first place or how to stop it occurring in the future.

The spp\_farm account is being used as the app pool identity for the SecurityTokenServiceApplicationPool and the SharePoint Central Administration v4 app pool on all SharePoint WFEs, so this would mean that all servers are potentially vulnerable to the HTTP 500 error as referenced by <https://support.microsoft.com/en-gb/help/3114011/800703fa-illegal-operation-attempted-on-a-registry-key-that-has-been-m>

**But**, this is the correct configuration for these app pool identities according to Microsoft:

From <https://docs.microsoft.com/en-us/sharepoint/install/account-permissions-and-security-settings-in-sharepoint-server-2016>

The SharePoint farm service account should only run the SharePoint Timer service, SharePoint Inights (if applicable), the IIS Application Pools for Central Administration, SharePoint Web Services System (used for the topology service), and SecurityTokenServiceApplicationPool (used for the Security Token Service).

I have had a trawl through the security audit log on WFE-22 it appears that the spp\_farm account is being used to logon (and logoff) to the server at 06:11:01 and 13:45:42 each day (the log only holds entries back to 24/10)

This looks like some kind of system monitoring service “Microsoft-Windows-Security-Auditing”.

This \***could**\* be a cause of the issue.

There’s a couple of things to note:

1. Service accounts should NOT have the logon locally right. It appears at least SPP\_Farm has this right. We should get this checked for all SharePoint service accounts except the “Claims to Windows Token Service” account – I don’t believe we are using this service.

Service accounts, with the exception of the account running the Claims to Windows Token Service, should have Deny logon locally and Deny logon through Remote Desktop Services in the Local Security Policy\User Rights Assignment. This is set via secpol.msc.

From <https://docs.microsoft.com/en-us/sharepoint/install/account-permissions-and-security-settings-in-sharepoint-server-2016>

1. Disabling the logon rights for the service accounts may fix the issue, however the comments by the Microsoft Support engineer below are worrying – this article is referenced at the bottom of <https://support.microsoft.com/en-gb/help/3114011/800703fa-illegal-operation-attempted-on-a-registry-key-that-has-been-m>

Indeed there are many logon/logoff scenarios. Besides interactive logon and remote desktop, starting/stopping a Windows service and an IIS app pool also causes the identity user logon/logoff along with the user profile load/unload. The key is to not to share the user identity of a COM+ application with any Windows services or IIS app pools and not to use the same identity to logon to the machine interactively or remotely. The issue is not limited to COM+ applications. It can happen for non-COM+ inproc COM dlls in IIS if loading user profile is not enabled for the app pool.

<https://blogs.msdn.microsoft.com/distributedservices/2009/11/06/a-com-application-may-stop-working-on-windows-server-2008-when-the-identity-user-logs-off/>

This means that disabling the logon rights may not be sufficient to prevent the issue happening.

1. In the same support article: Question to MS Support engineer:
2. We are facing the same error in our Production environment. Will chnaging the group policy cause any harm?

Will it cause any performance issues? Please let me know if there is any alternate solution other than Group Policy edit.

His response:

I haven't seen any issues from customers with changing the group policy. There are other workarounds. If your issue is with a COM+ server app (dllhost.exe), make sure you use an unique identity for it (don't not share the identity with any other app or use it to logon). If the issue is with an IIS app, ensure "load user profile" is enabled for the app pool or use an unique identity.

The MS support engineer also points out:

Note that it is an intermittent issue. You can run into the issue later.