Collecting Registry Information with Least Privilege Access Research Notes

Registry Keys and Values

There are several registry key values that are important to BloodHound, and thus collected by SharpHound:

- Domain Controllers
    - SYSTEM\CurrentControlSet\Services\Kdc
        - § StrongCertificateBindingEnforcement
    - SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL
        - § CertificateMappingMethods
- Certificate Authorities
    - SYSTEM\CurrentControlSet\Services\CertSvc\Configuration\{caName}
        - § Security
        - § EnrollmentAgentRights
        - § RoleSeparationEnabled
    - SYSTEM\CurrentControlSet\Services\CertSvc\Configuration\{caName}\PolicyModules\CertificateAuthority_MicrosoftDefault.Policy
        - § EditFlags
- All Hosts (NTLM Relay)
    - SYSTEM\CurrentControlSet\Control\Lsa\MSV1_0
        - § ClientAllowedNTLMServers
        - § NtlmMinClientSec

        - § NtlmMinServerSec
        - § RestrictReceivingNTLMTraffic
        - § RestrictSendingNTLMTraffic
    - SYSTEM\CurrentControlSet\Control\Lsa
        - § LMCompatibilityLevel
        - § UseMachineId
    - SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters
        - § EnableSecuritySignature
        - § RequireSecuritySignature

Requirements

In order to collect registry data the following must be in effect:

1. CollectionMethod includes DCRegistry, CARegistry, and/or NTLMRegistry

2. A network connection between the SharpHound collector and each target computer must be possible using the SMB protocol over port 445.

3. RemoteRegistry must be enabled on all target computers where registry collection is desired

4. The SharpHound collector account must have rights to remotely read the registry

5. The SharpHound collector account must have rights to read the registry keys

Group Policy

There are a handful of configuration options for registry access via GPO:

Network access: Remotely accessible registry paths and subpaths

https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-10/security/threat-protection/security-policy-settings/network-access-remotely-accessible-registry-paths-and-subpaths

Policy Path: Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options

Help Text:

> Network access: Remotely accessible registry paths and subpaths
>
> This security setting determines which registry paths and subpaths can be accessed over the network, regardless of the users or groups listed in the access control list (ACL) of the winreg registry key.
>
> Default:
>
> System\\CurrentControlSet\\Control\\Print\\Printers
> System\\CurrentControlSet\\Services\\Eventlog
> Software\\Microsoft\\OLAP Server
> Software\\Microsoft\\Windows NT\\CurrentVersion\\Print
> Software\\Microsoft\\Windows NT\\CurrentVersion\\Windows
> System\\CurrentControlSet\\Control\\ContentIndex
> System\\CurrentControlSet\\Control\\Terminal Server
> System\\CurrentControlSet\\Control\\Terminal Server\\UserConfig
> System\\CurrentControlSet\\Control\\Terminal Server\\DefaultUserConfiguration
> Software\\Microsoft\\Windows NT\\CurrentVersion\\Perflib
> System\\CurrentControlSet\\Services\\SysmonLog
> System\\CurrentControlSet\\Services\\CertSvc
> System\\CurrentControlSet\\Services\\Wins
>
> Caution
>
> Incorrectly editing the registry may severely damage your system. Before making changes to the registry, you should back up any valued data on the computer.
>
> Note: On Windows XP, this security setting was called "Network access: Remotely accessible registry paths." If you configure this setting on a member of the Windows Server 2003 family that is joined to a domain, this setting is inherited by computers running Windows XP, but will appear as the "Network access: Remotely accessible registry paths" security option. For more information, see Network access: Remotely accessible registry paths and subpaths.

Microsoft recommended setting: Enabled with a NULL value

When the policy is not configured or is configured with a NULL value the following registry keys will be accessible:

1. System\CurrentControlSet\Control\Print\Printers
2. System\CurrentControlSet\Services\Eventlog
3. Software\Microsoft\OLAP Server
4. Software\Microsoft\Windows NT\CurrentVersion\Print
5. Software\Microsoft\Windows NT\CurrentVersion\Windows
6. System\CurrentControlSet\Control\ContentIndex

7. System\CurrentControlSet\Control\Terminal Server
8. System\CurrentControlSet\Control\Terminal Server\UserConfig
9. System\CurrentControlSet\Control\Terminal Server\DefaultUserConfiguration
10. Software\Microsoft\Windows NT\CurrentVersion\Perflib
11. System\CurrentControlSet\Services\SysmonLog

To add additional keys without affecting default Windows behavior for remote management tools such as Microsoft Baseline Security Analyzer or Microsoft Configuration Manager, these keys will need to be included in the configuration with any additional keys.

Network access: Remotely accessible registry paths

https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-10/security/threat-protection/security-policy-settings/network-access-remotely-accessible-registry-paths

Policy Path: Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options

Help Text:

> Network access: Remotely accessible registry paths
>
> This security setting determines which registry keys can be accessed over the network, regardless of the users or groups listed in the access control list (ACL) of the winreg registry key.
>
> Default:
>
> System\\CurrentControlSet\\Control\\ProductOptions
> System\\CurrentControlSet\\Control\\Server Applications
> Software\\Microsoft\\Windows NT\\CurrentVersion
>
> Caution
>
> Incorrectly editing the registry may severely damage your system. Before making changes to the registry, you should back up any valued data on the computer.
> Note: This security setting is not available on earlier versions of Windows. The security setting that appears on computers running Windows XP, "Network access: Remotely accessible registry paths" corresponds to the "Network access: Remotely accessible registry paths and subpaths" security option on members of the Windows Server 2003 family. For more information, see Network access: Remotely accessible registry paths and subpaths.
> Default:
>
> System\CurrentControlSet\Control\ProductOptions
> System\CurrentControlSet\Control\Server Applications
> Software\Microsoft\Windows NT\CurrentVersion

Microsoft recommended setting: Enabled with a NULL value

When the policy is not configured or is configured with a NULL value the following registry keys will be accessible:

1. System\CurrentControlSet\Control\ProductOptions
2. System\CurrentControlSet\Control\Server Applications
3. Software\Microsoft\Windows NT\CurrentVersion

To add additional keys without affecting default Windows behavior for remote management tools such as Microsoft Baseline Security Analyzer or Microsoft Configuration Manager, these keys will need to be included in the configuration with any additional keys.

Registry Key Security

<website>

Key: Computer Configuration\Policies\Windows Settings\Security Settings\Registry

System Services

< website>

Service Name: Remote Registry

Methodology

Code and Documentation Review

Reviewed SharpHound and SharpHoundCommon code to determine registry collection method used: [Microsoft.Win32.RegistryKey]

Reviewed Microsoft documentation on remote registry access and GPOs related to remote registry access.

Lab Testing

Enabled Remote Registry on TellerDC01.magic.lab.lan in my homelab environment.

Created Test-RemoteRegistry.ps1 script to run in lab environment, which will test access to SharpHound required registry keys, and keys which are default enabled for remote access by policy defaults that are in place when no GPO setting is applied for Network Access: Remotely access registry paths (and subpaths).

In my lab I’m utilizing a standard user account with no privileged groups:

```
PS C:\Users\jsykora> whoami -all

USER INFORMATION

----------------

User Name        SID

=============== ==============================================

magic\jsykora S-1-5-21-3520149094-1197618848-1674470492-1273

GROUP INFORMATION

-----------------

Group Name                                  Type             SID                                           Attributes

========================================== ================
=============================================
=================================================================

MAGIC\Creator_Group_Test                    Group
S-1-5-21-3520149094-1197618848-1674470492-1554 Mandatory group, Enabled by default, Enabled
group

Everyone                                    Well-known group S-1-1-0                                       Mandatory group,
Enabled by default, Enabled group

BUILTIN\Users                               Alias            S-1-5-32-545                                  Mandatory
group, Enabled by default, Enabled group

BUILTIN\Remote Desktop Users                Alias            S-1-5-32-555
Mandatory group, Enabled by default, Enabled group

NT AUTHORITY\REMOTE INTERACTIVE LOGON       Well-known group S-1-5-14
Mandatory group, Enabled by default, Enabled group
```

NT AUTHORITY\INTERACTIVE Well-known group S-1-5-4
Mandatory group, Enabled by default, Enabled group

NT AUTHORITY\Authenticated Users Well-known group S-1-5-11
Mandatory group, Enabled by default, Enabled group

NT AUTHORITY\This Organization Well-known group S-1-5-15
Mandatory group, Enabled by default, Enabled group

LOCAL Well-known group S-1-2-0 Mandatory group,
Enabled by default, Enabled group

Authentication authority asserted identity Well-known group S-1-18-1 Mandatory
group, Enabled by default, Enabled group

MAGIC\GGP-DenyCreateChildObject Alias
S-1-5-21-3520149094-1197618848-1674470492-1548 Mandatory group, Enabled by default, Enabled
group, Local Group

MAGIC\P-AllowCreateChildObject Alias
S-1-5-21-3520149094-1197618848-1674470492-1550 Mandatory group, Enabled by default, Enabled
group, Local Group

MAGIC\GP-AllowCreateChildObject Alias
S-1-5-21-3520149094-1197618848-1674470492-1549 Mandatory group, Enabled by default, Enabled
group, Local Group

Mandatory Label\Medium Mandatory Level Label S-1-16-8192

PRIVILEGES INFORMATION

Privilege Name Description State

============================== ================================== ========

SeShutdownPrivilege Shut down the system Disabled

SeChangeNotifyPrivilege Bypass traverse checking Enabled

SeUndockPrivilege Remove computer from docking station Disabled

SeIncreaseWorkingSetPrivilege Increase a process working set Disabled

SeTimeZonePrivilege Change the time zone Disabled

USER CLAIMS INFORMATION

User claims unknown.

Kerberos support for Dynamic Access Control on this device has been disabled.

PS C:\Users\jsykora>

Remote Registry Permissions Test

Using the jsykora standard user account, I determined that without any additional modifications, a standard user account (domain users) was able to collect data from the DC's registry in the following areas:

Path: HKLM\System\CurrentControlSet\Control\ProductOptions\ProductSuite Data: Terminal Server

Path: HKLM\Software\Microsoft\Windows NT\CurrentVersion\Print\DoNotInstallCompatibleDriverFromWindowsUpdate Data: 1

Path: HKLM\Software\Microsoft\Windows NT\CurrentVersion\Windows\Spooler Data: yes

Path: HKLM\Software\Microsoft\Windows NT\CurrentVersion\ProductName Data: Windows Server 2019 Standard

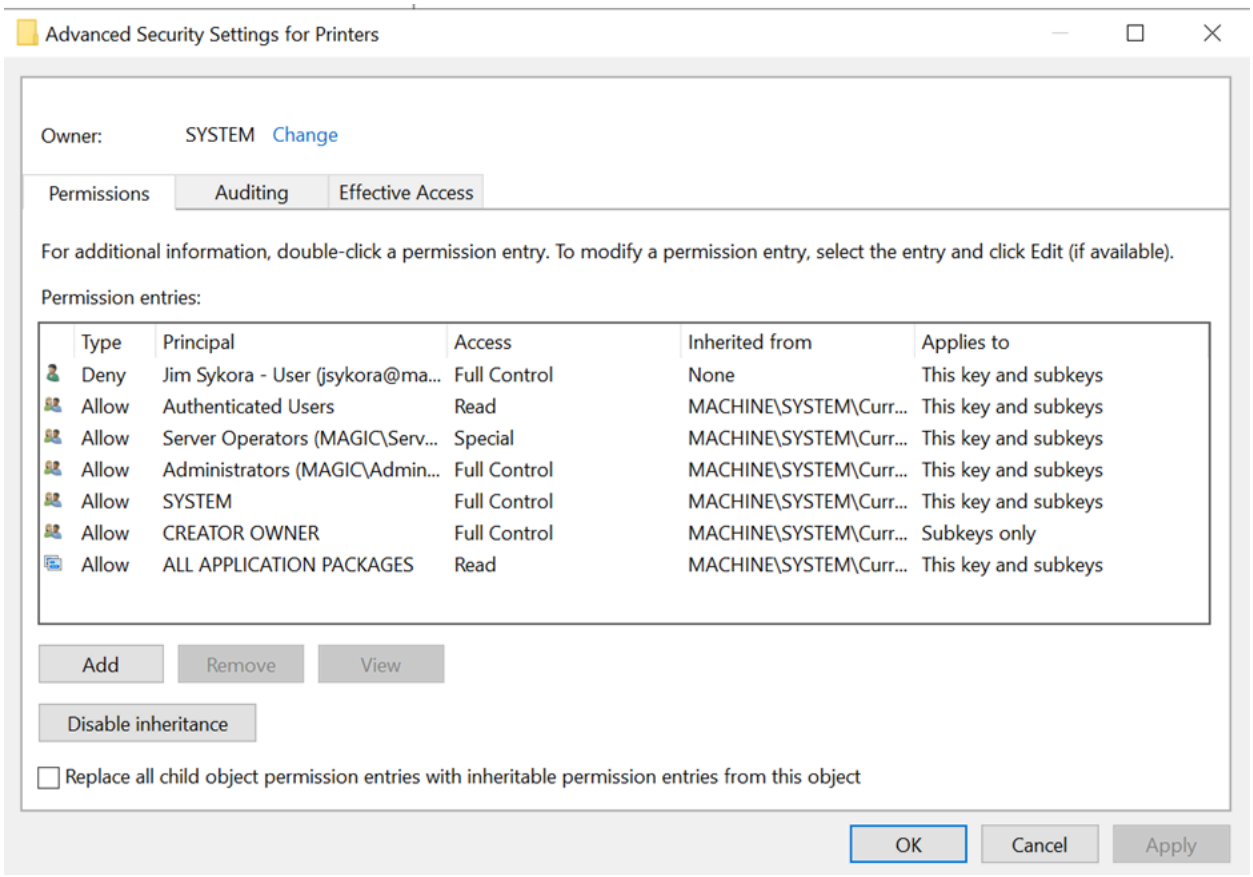
Path: HKLM\System\CurrentControlSet\Control\Print\Printers\DefaultSpoolDirectory Data: C:\Windows\system32\spool\PRINTERS

Path: HKLM\System\CurrentControlSet\Services\Eventlog\RequiredPrivileges Data: SeChangeNotifyPrivilege SeImpersonatePrivilege SeAuditPrivilege

This is due to the default policies in place as described in the Network access: Remote accessible registry paths (and subpaths) policy settings.

Deny ACE Test

I then chose a registry path which would have minimal impact: HKLM\System\CurrentControlSet\Control\Print\Printers\DefaultSpoolDirectory and created a Deny ACE for my user account ‘jsykora’ as a test to determine if the GP setting ‘Network access: Remotely accessible registry paths and subpaths’ overrides the default SD on this registry key, as by default Authenticated Users is granted Read rights via inheritance.



Here is script output from before making the change to the registry subkey DACL:

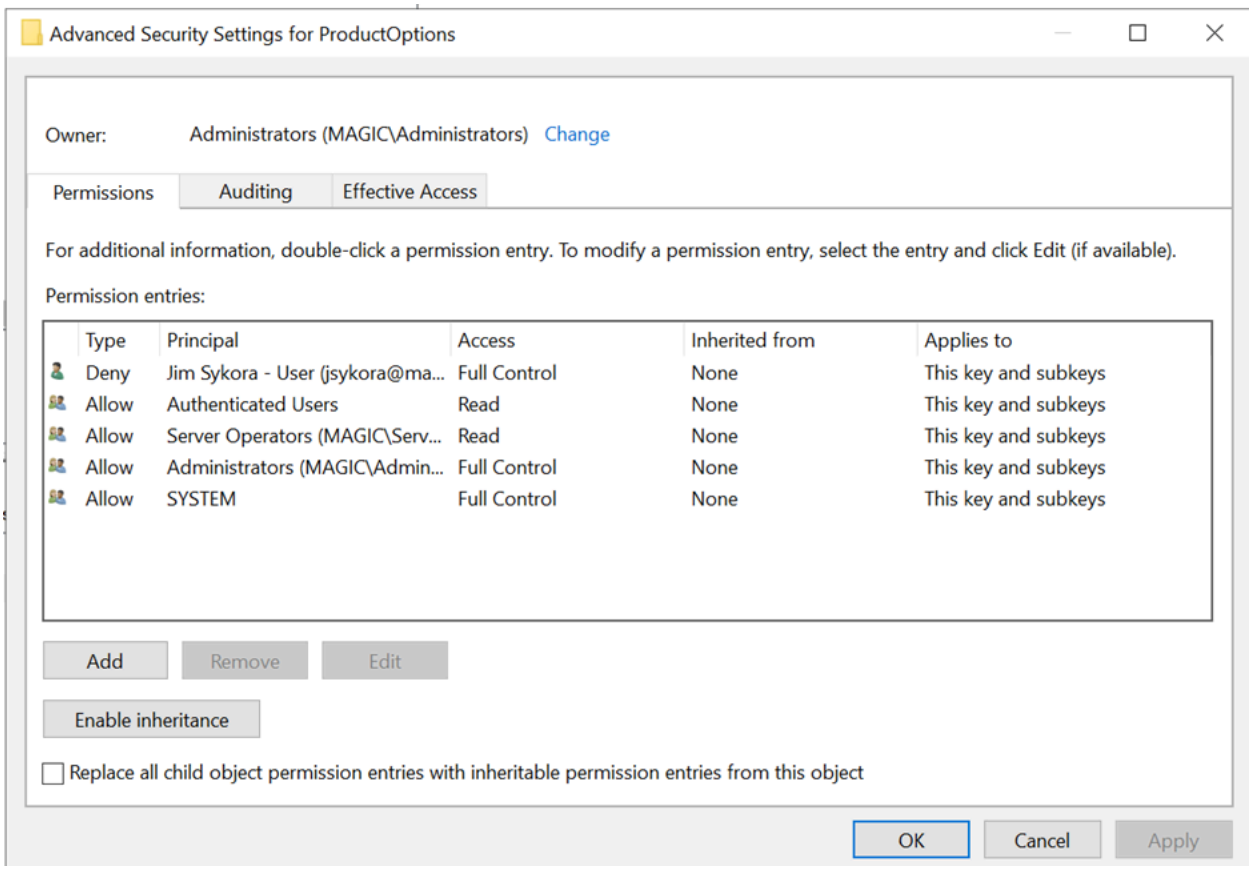
```
Path: HKLM\SYSTEM\CurrentControlSet\Services\Kdc\StrongCertificateBindingEnforcement - OpenSubKey Error
Path: HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\CertificateMappingMethods - OpenSubKey Error
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\MSV1_0\ClientAllowedNTLMServers - OpenSubKey Error
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\MSV1_0\NtlmMinClientSec - OpenSubKey Error
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\MSV1_0\NtlmMinServerSec - OpenSubKey Error
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\MSV1_0\RestrictReceivingNTLMTraffic - OpenSubKey Error
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\MSV1_0\RestrictSendingNTLMTraffic - OpenSubKey Error
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\LMCompatibilityLevel - OpenSubKey Error
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\UseMachineId - OpenSubKey Error
Path: HKLM\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters\EnableSecuritySignature - OpenSubKey Error
Path: HKLM\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters\RequireSecuritySignature - OpenSubKey Error
Path: HKLM\System\CurrentControlSet\Control\ProductOptions\ProductSuite  Data: Terminal Server
Path: HKLM\Software\Microsoft\Windows NT\CurrentVersion\ProductName  Data: Windows Server 2019 Standard
Path: HKLM\System\CurrentControlSet\Control\Print\Printers\DefaultSpoolDirectory  Data: C:\Windows\system32\spool\PRINTERS
Path: HKLM\System\CurrentControlSet\Services\Eventlog\RequiredPrivileges  Data: SeChangeNotifyPrivilege SeImpersonatePrivilege SeAuditPrivilege
Path: HKLM\Software\Microsoft\Windows NT\CurrentVersion\Print\DoNotInstallCompatibleDriverFromWindowsUpdate  Data: 1
Path: HKLM\Software\Microsoft\Windows NT\CurrentVersion\Windows\Spooler  Data: yes

PS C:\Users\jsykora>
```

Here is the script output from after adding the Deny jsykora ACE on the registry key DACL:

```
Path: HKLM\SYSTEM\CurrentControlSet\Services\Kdc\StrongCertificateBindingEnforcement - OpenSubKey Error
Path: HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\CertificateMappingMethods - OpenSubKey Error
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\MSV1_0\ClientAllowedNTLMServers - OpenSubKey Error
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\MSV1_0\NtlmMinClientSec - OpenSubKey Error
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\MSV1_0\NtlmMinServerSec - OpenSubKey Error
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\MSV1_0\RestrictReceivingNTLMTraffic - OpenSubKey Error
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\MSV1_0\RestrictSendingNTLMTraffic - OpenSubKey Error
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\LMCompatibilityLevel - OpenSubKey Error
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\UseMachineId - OpenSubKey Error
Path: HKLM\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters\EnableSecuritySignature - OpenSubKey Error
Path: HKLM\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters\RequireSecuritySignature - OpenSubKey Error
Path: HKLM\System\CurrentControlSet\Control\ProductOptions\ProductSuite  Data: Terminal Server
Path: HKLM\Software\Microsoft\Windows NT\CurrentVersion\ProductName  Data: Windows Server 2019 Standard
Path: HKLM\System\CurrentControlSet\Control\Print\Printers\DefaultSpoolDirectory - OpenSubKey Error
Path: HKLM\System\CurrentControlSet\Services\Eventlog\RequiredPrivileges  Data: SeChangeNotifyPrivilege SeImpersonatePrivilege SeAuditPrivilege
Path: HKLM\Software\Microsoft\Windows NT\CurrentVersion\Print\DoNotInstallCompatibleDriverFromWindowsUpdate  Data: 1
Path: HKLM\Software\Microsoft\Windows NT\CurrentVersion\Windows\Spooler  Data: yes

PS C:\Users\jsykora> |
```

Note that my user account jsykora is no longer able to open the System\CurrentControlSet\Control\Print\Printers subkey. This demonstrates that the ‘Network access: Remotely accessible registry paths and subpaths’ policy setting does not override the registry subkey permissions.

For the sake of thoroughness, I will delete the deny ACE from the previous Printers subkey and test a key from the policy ‘Network access: Remotely accessible registry paths’:



And here also, we can see that the GP setting ‘Network access: Remotely accessible registry paths’ does not override the permissions on the subkey:

```
Path: HKLM\SYSTEM\CurrentControlSet\Services\Kdc\StrongCertificateBindingEnforcement - OpenSubKey Error
Path: HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\CertificateMappingMethods - OpenSubKey Error
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\MSV1_0\ClientAllowedNTLMServers - OpenSubKey Error
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\MSV1_0\NtlmMinClientSec - OpenSubKey Error
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\MSV1_0\NtlmMinServerSec - OpenSubKey Error
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\MSV1_0\RestrictReceivingNTLMTraffic - OpenSubKey Error
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\MSV1_0\RestrictSendingNTLMTraffic - OpenSubKey Error
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\LMCompatibilityLevel - OpenSubKey Error
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\UseMachineId - OpenSubKey Error
Path: HKLM\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters\EnableSecuritySignature - OpenSubKey Error
Path: HKLM\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters\RequireSecuritySignature - OpenSubKey Error
Path: HKLM\System\CurrentControlSet\Control\ProductOptions\ProductSuite - OpenSubKey Error
Path: HKLM\Software\Microsoft\Windows NT\CurrentVersion\ProductName  Data: Windows Server 2019 Standard
Path: HKLM\System\CurrentControlSet\Control\Print\Printers\DefaultSpoolDirectory  Data: C:\Windows\system32\spool\PRINTERS
Path: HKLM\System\CurrentControlSet\Services\Eventlog\RequiredPrivileges  Data: SeChangeNotifyPrivilege SeImpersonatePrivilege SeAuditPrivilege
Path: HKLM\Software\Microsoft\Windows NT\CurrentVersion\Print\DoNotInstallCompatibleDriverFromWindowsUpdate  Data: 1
Path: HKLM\Software\Microsoft\Windows NT\CurrentVersion\Windows\Spooler  Data: yes

PS C:\Users\jsykora>
```

Privileged User Remote Registry Permissions Test

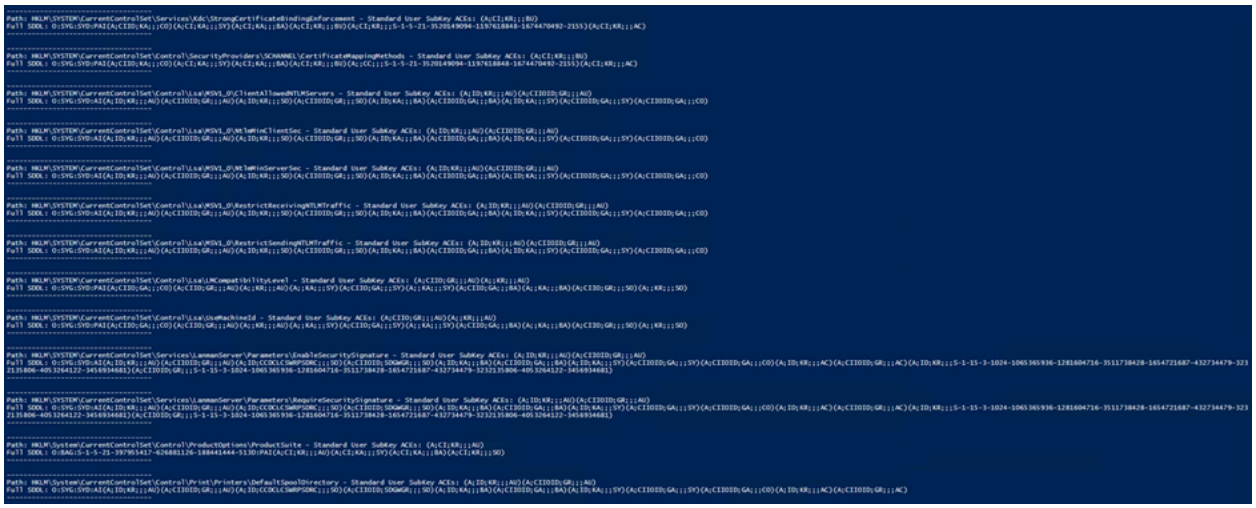
I also ran this same script from another host under the security context of a member of Domain Admins. Of note is that in the environment I’m testing, not all these registry values are populated:

```
Path: HKLM\SYSTEM\CurrentControlSet\Services\Kdc\StrongCertificateBindingEnforcement - NullValue
Path: HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\CertificateMappingMethods - NullValue
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\MSV1_0\ClientAllowedNTLMServers - NullValue
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\MSV1_0\NtlmMinClientSec  Data: 536870912
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\MSV1_0\NtlmMinServerSec  Data: 536870912
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\MSV1_0\RestrictReceivingNTLMTraffic - NullValue
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\MSV1_0\RestrictSendingNTLMTraffic - NullValue
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\LMCompatibilityLevel  Data: 0
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\UseMachineId - NullValue
Path: HKLM\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters\EnableSecuritySignature  Data: 1
Path: HKLM\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters\RequireSecuritySignature  Data: 1
Path: HKLM\System\CurrentControlSet\Control\ProductOptions\ProductSuite  Data: Terminal Server
Path: HKLM\Software\Microsoft\Windows NT\CurrentVersion\ProductName  Data: Windows Server 2019 Standard
Path: HKLM\System\CurrentControlSet\Control\Print\Printers\DefaultSpoolDirectory  Data: C:\Windows\system32\spool\PRINTERS
Path: HKLM\System\CurrentControlSet\Services\Eventlog\RequiredPrivileges  Data: SeChangeNotifyPrivilege SeImpersonatePrivilege SeAuditPrivilege
Path: HKLM\Software\Microsoft\Windows NT\CurrentVersion\Print\DoNotInstallCompatibleDriverFromWindowsUpdate  Data: 1
Path: HKLM\Software\Microsoft\Windows NT\CurrentVersion\Windows\Spooler  Data: yes

PS C:\Windows\system32>
```

Registry Permissions

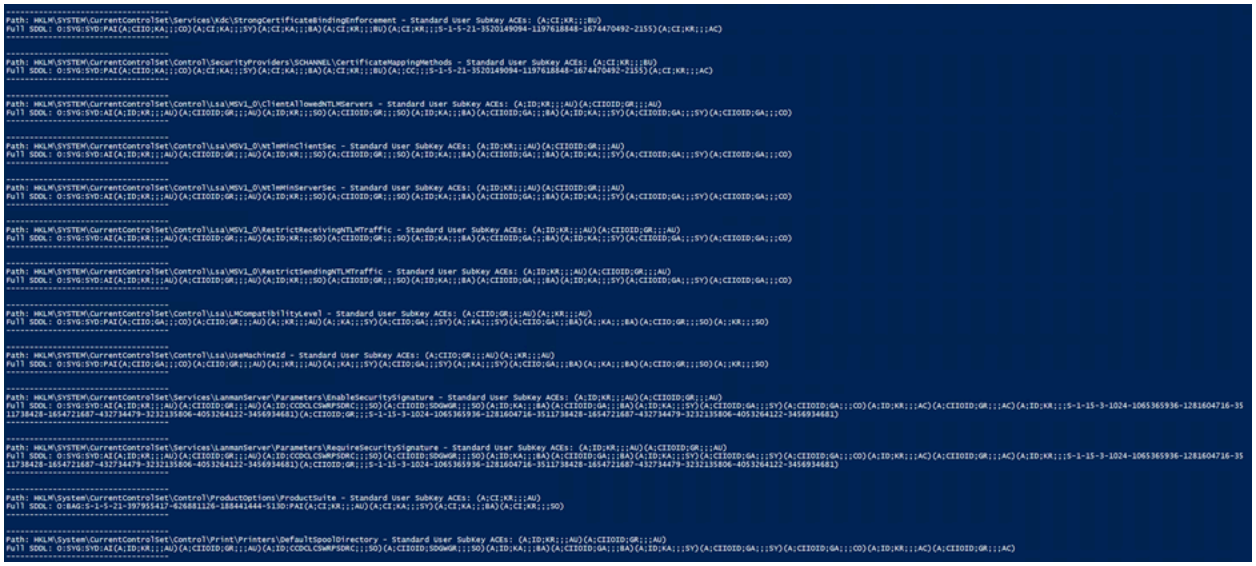
Based on this information, my next step will be to script out the collection of the security descriptor on all subkeys required by SharpHound.

Using Get-SharpHoundLocalRegistryPermissions.ps1 and in the security context of a member of Domain Admins, I get this output on ‘TellerDC01’ in my lab:



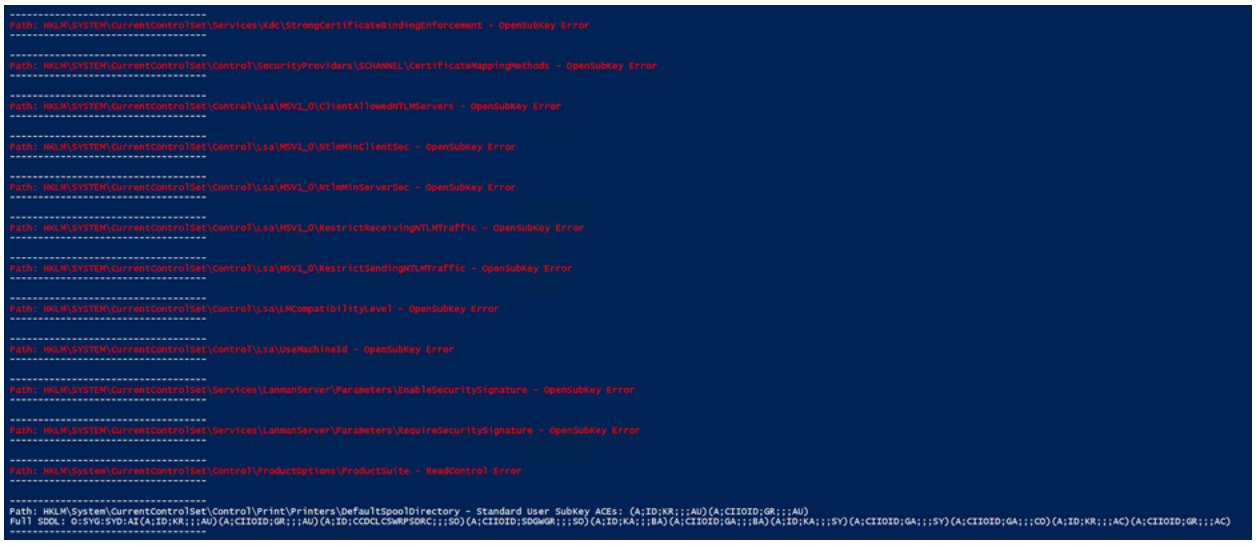
Text of this can be found at:
https://github.com/JimSycurity/LeastPrivilegeSharpHound/blob/main/RemoteRegistry/Data/TellerDC01-Get-SharpHoundLocalRegistryPermissions.txt

Using Get-SharpHoundRemoteRegistryPermissions.ps1 scriptlet from in the security context of a Domain Admin account I get this output, which matches the local gather:



https://github.com/JimSycurity/LeastPrivilegeSharpHound/blob/main/RemoteRegistry/Data/TellerDC01-Get-SharpHoundRemoteRegistryPermissions01.txt

Using the same Get-SharpHoundRemoteRegistryPermissions.ps1 scriptlet on a Windows 10 client with standard user permissions (jsykora account) I get this output:



https://github.com/JimSycurity/LeastPrivilegeSharpHound/blob/main/RemoteRegistry/Data/TellerDC01-Get-SharpHoundRemoteRegistryPermissions02.txt

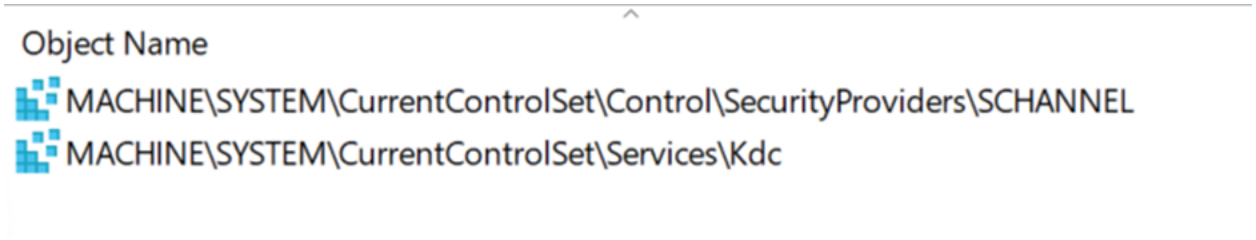
This data demonstrates that by default on Windows Server 2019, which is the OS for TellerDC01, either Authenticated Users or Builtin Users are granted KeyRead rights on all registry subkeys currently collected by SharpHound.

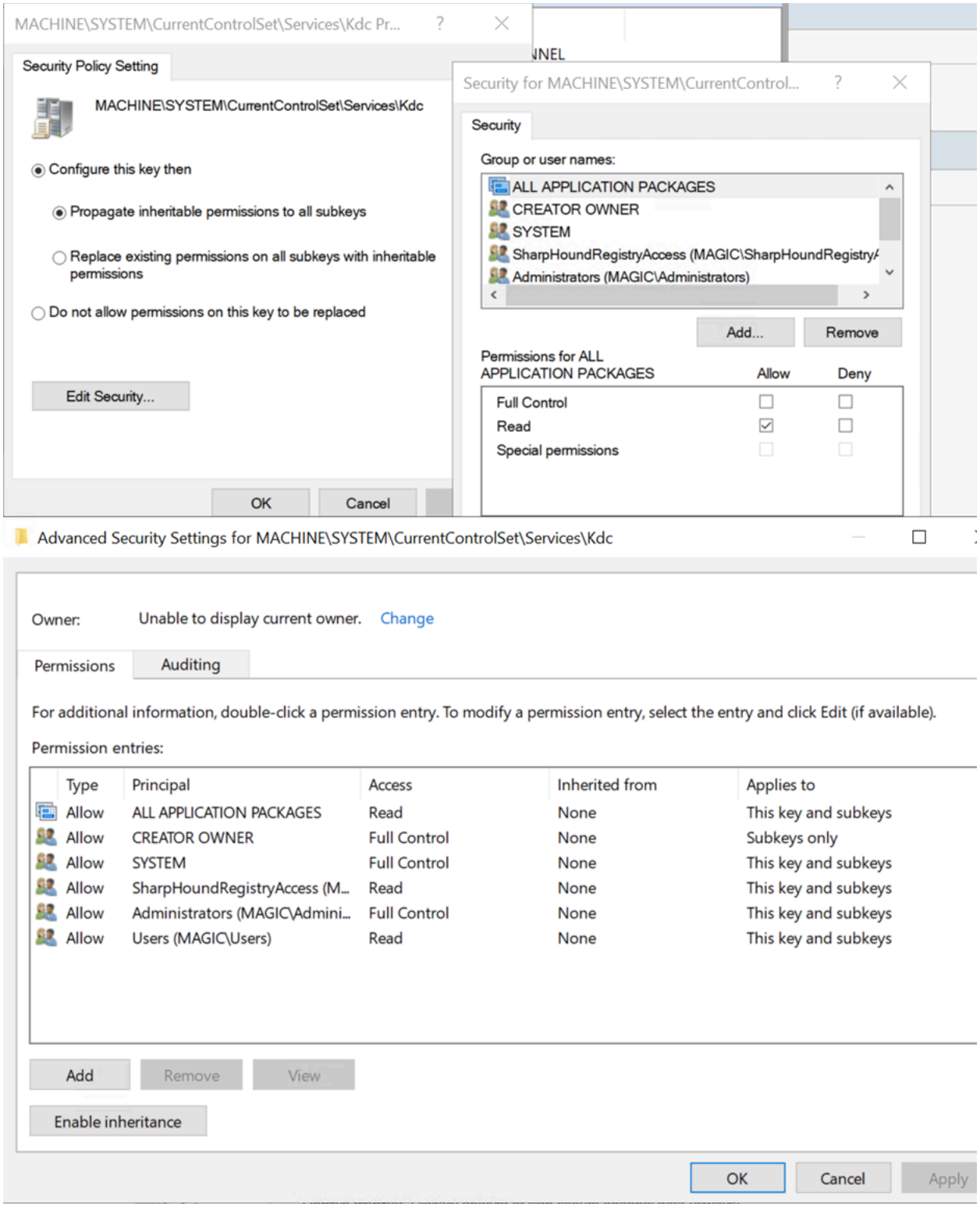
The last result collecting permissions via remote registry from a standard user account makes sense as the jsykora account does not have rights to open the subkey for all of the SharpHound values and is only granted KeyRead rights on the ProductSuite subkey. The DefaultSpoolDirectory subkey grants GenericRead rights to Authenticated Users, which includes the ReadControl access mask.

Collections on Windows Server 2012R2 and Windows Server 2025 domain controllers also indicate that either Builtin Users or Authenticated Users are granted rights to read key values in all SharpHound-related registry keys.

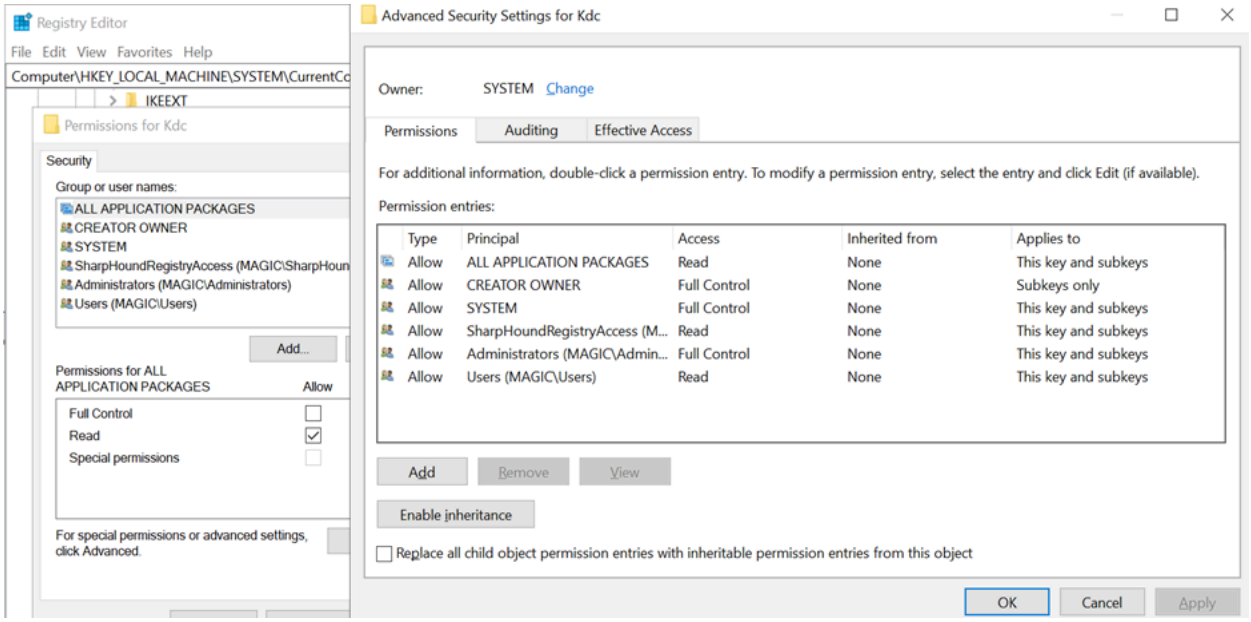
GPO Tests and Aha! Moment

As I prepared to configure a new GPO and link it to the Domain Controllers OU for the purposes of testing the ‘Network access:’ settings, I recalled that prior to testing this methodically, I had added 2 settings to a GPO which modify security descriptors on registry subkeys:





This change had the intended effect on the registry subkey:



I did remove these settings from the policy applied to the Domain Controllers OU, but it's unlikely that these changes will be removed from the registry as nearly every GPO changes a registry setting and often removing the GP setting does not revert the registry change.

While thinking about this, I thought to review the registry paths which the Network access: GP settings modify:

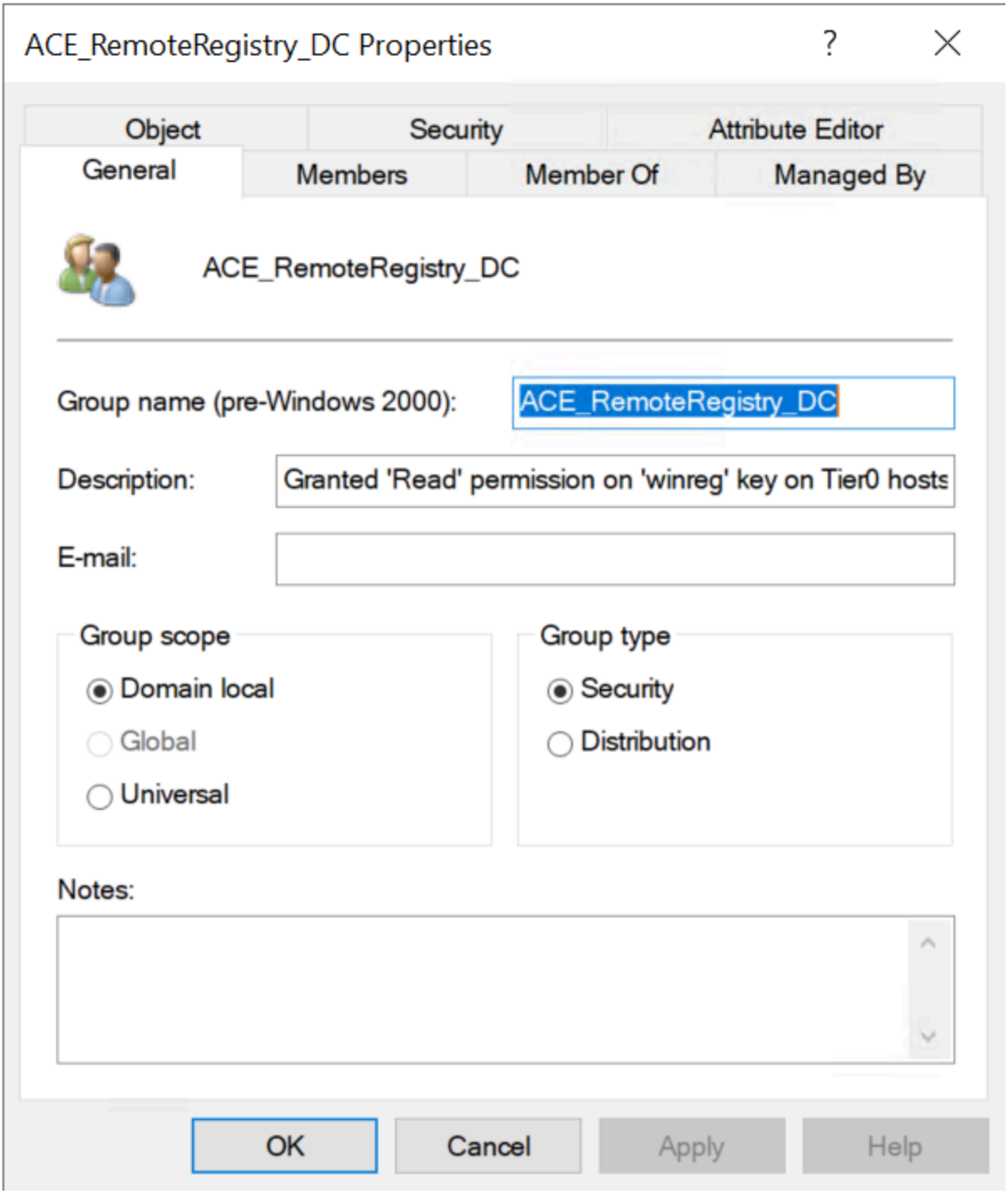
'Network access: Remotely accessible registry paths and subpaths' – 'Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurePipeServers\winreg\AllowedPaths'

'Network access: Remotely accessible registry paths' – 'HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurePipeServers\winreg\AllowedExactPaths'

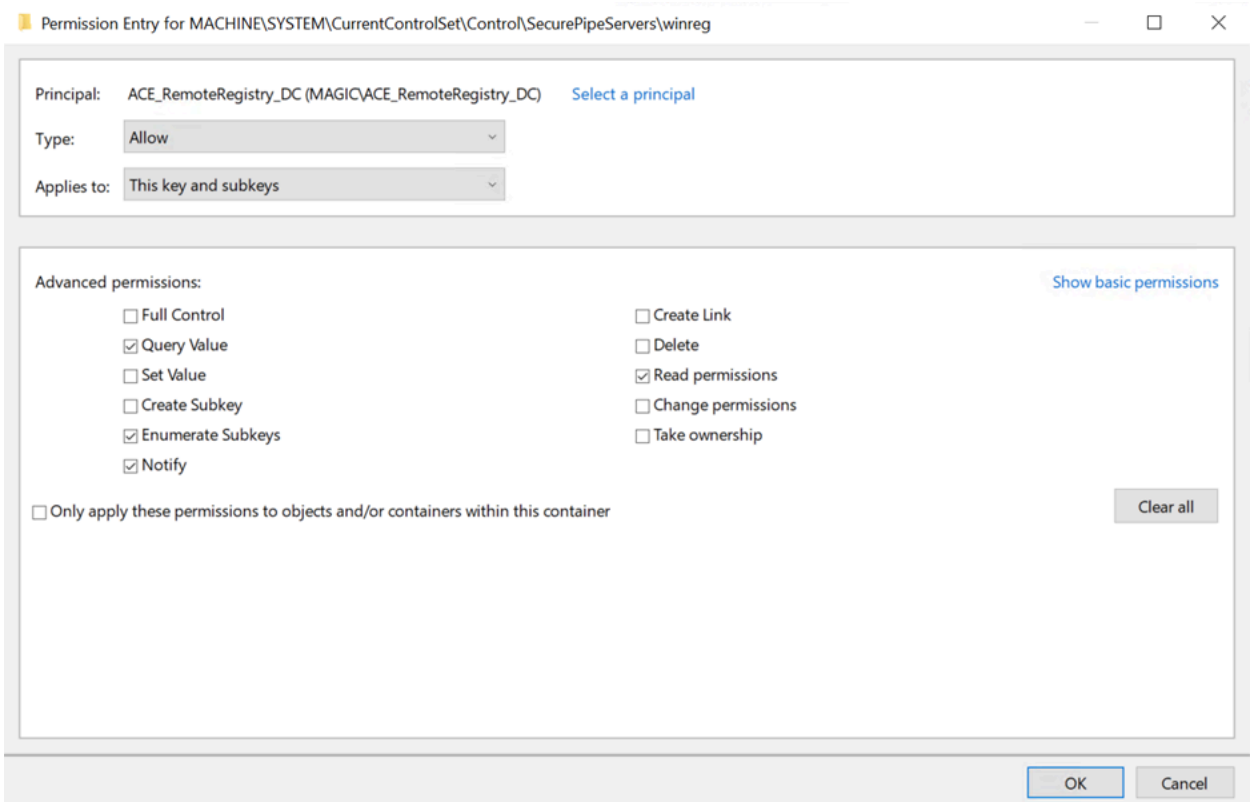
The Aha! Moment is the realization that these two GP settings modify registry keys which control access to the winreg Named Pipe by overriding the DACL on the winreg key.

Winreg Named Pipe & GPO

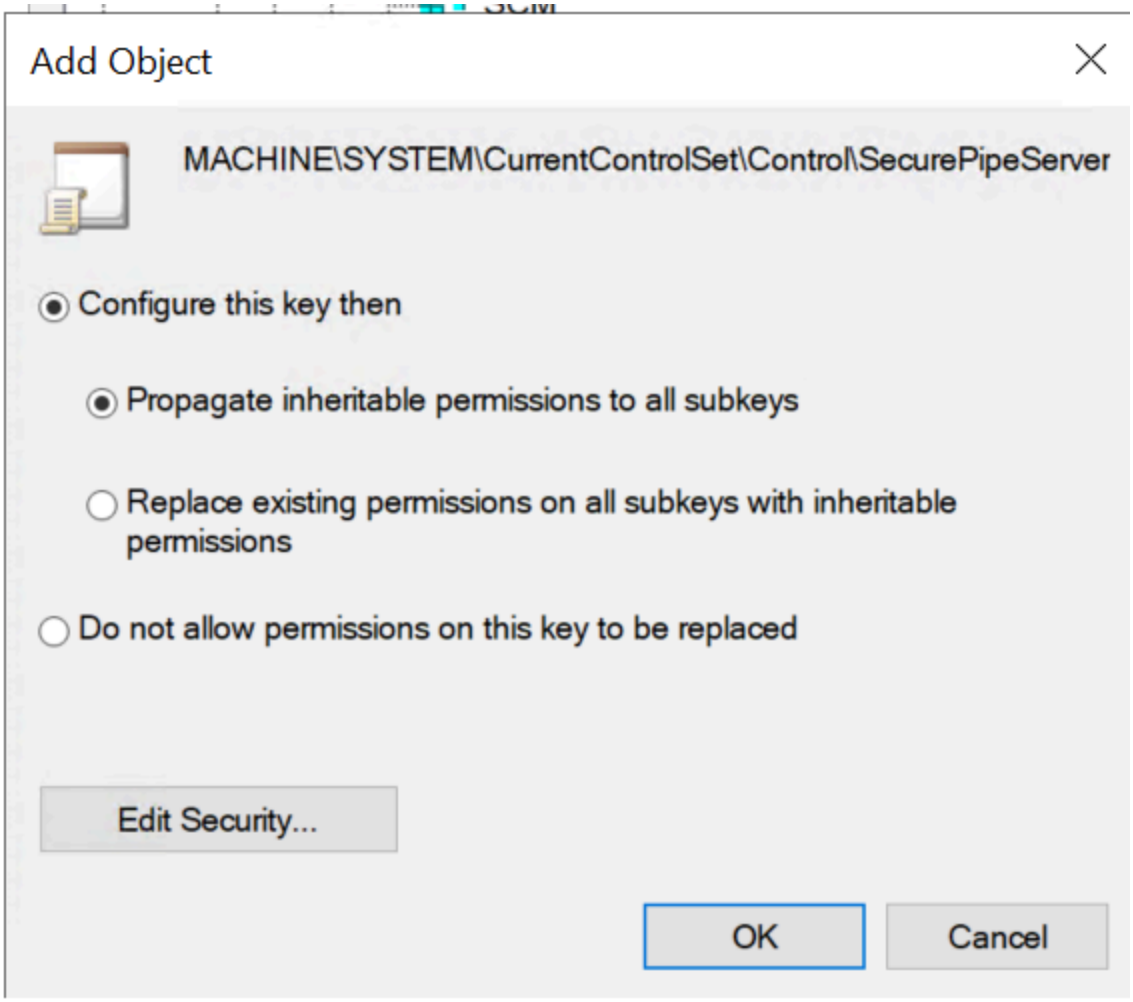
I created a new local security group 'ACE_RemoteRegistry_DC' in the OU: OU=Groups,OU=Tier0,DC=magic,DC=lab,DC=lan and added my "SharpHound Service Account" as a member:

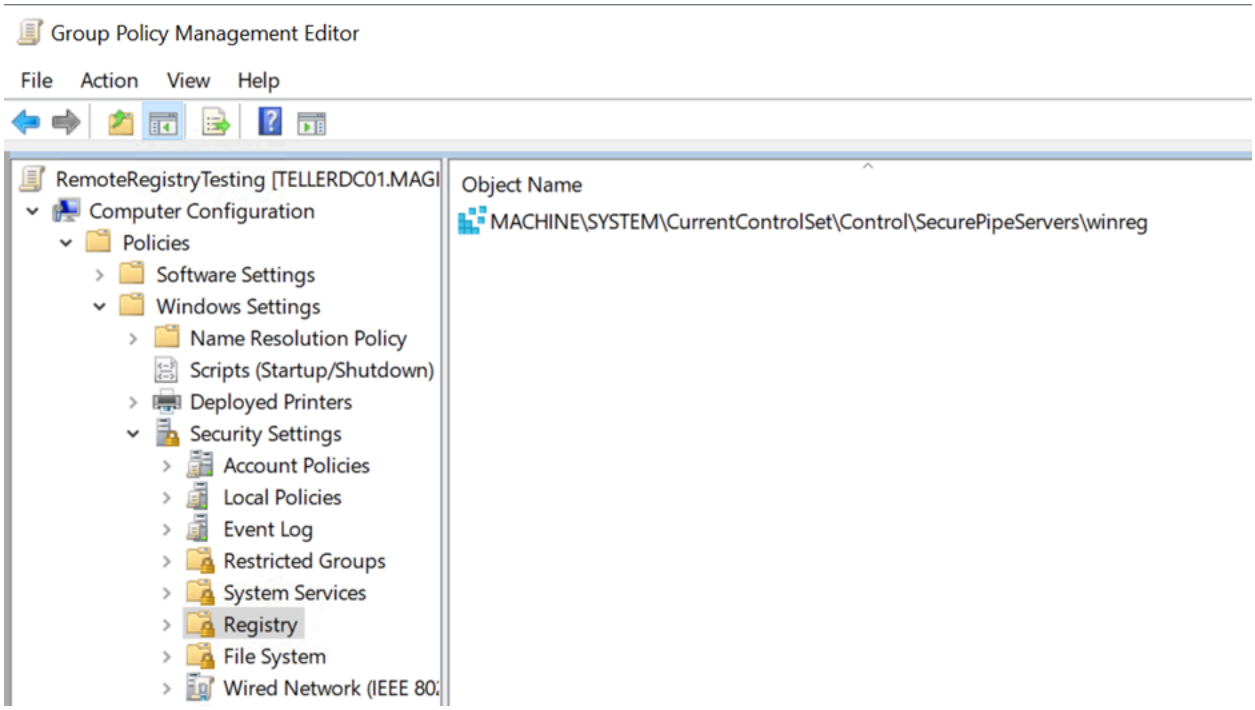


Then in the GPO I linked to the Domain Controllers OU I created a Key setting for 'HKLM\SYSTEM\CurrentControlSet\Control\SecurePipeServers\winreg' and added an ACE with the trustee/principal being ACE_RemoteRegistry_DC and the Allow access granted being Read:

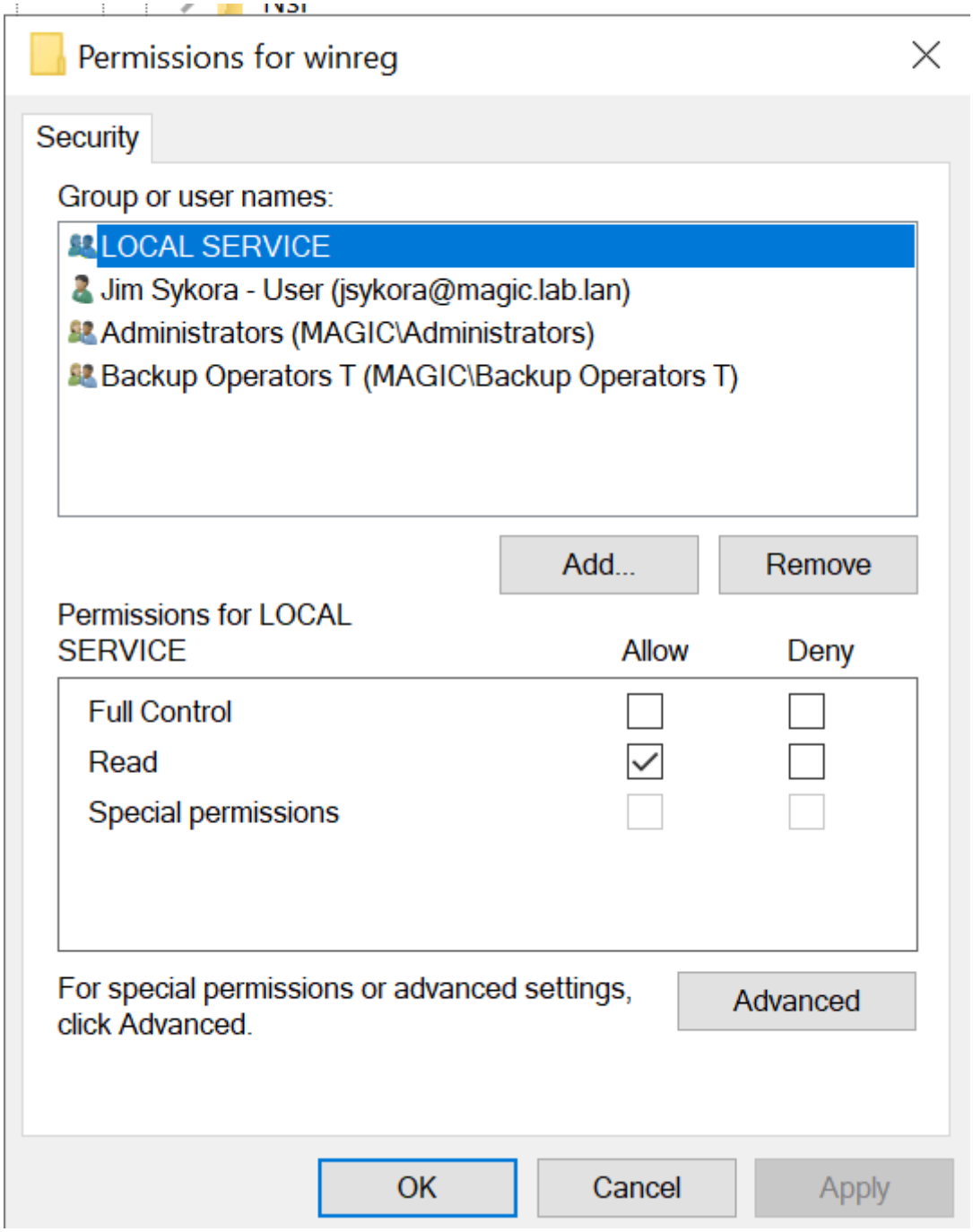


Add object settings used:


Add Object
MACHINE\SYSTEM\CurrentControlSet\Control\SecurePipeServer
Configure this key then
Propagate inheritable permissions to all subkeys
Replace existing permissions on all subkeys with inheritable permissions
Do not allow permissions on this key to be replaced
Edit Security...
OK
Cancel

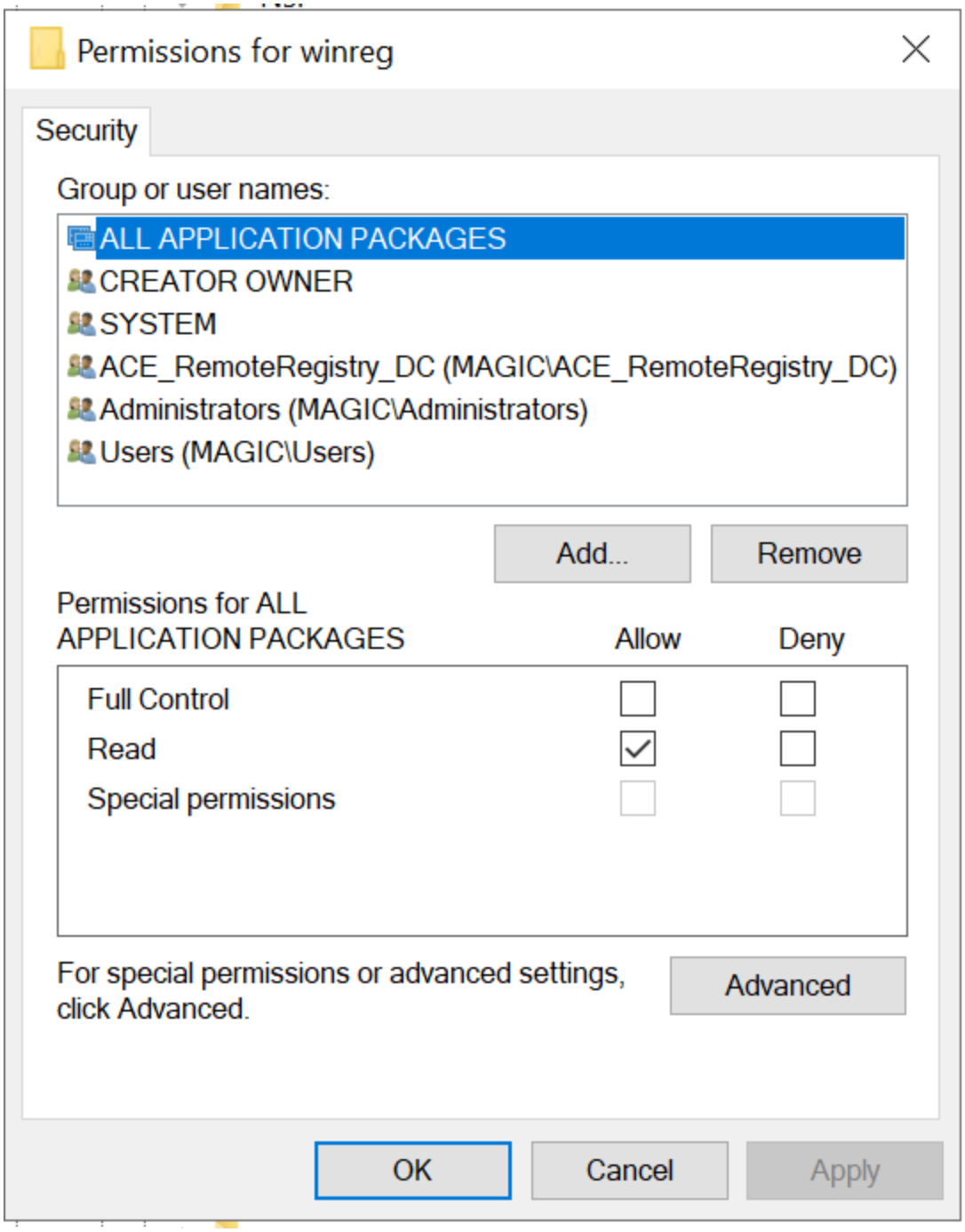


Here is the DACL on the winreg key prior to applying this new GPO:

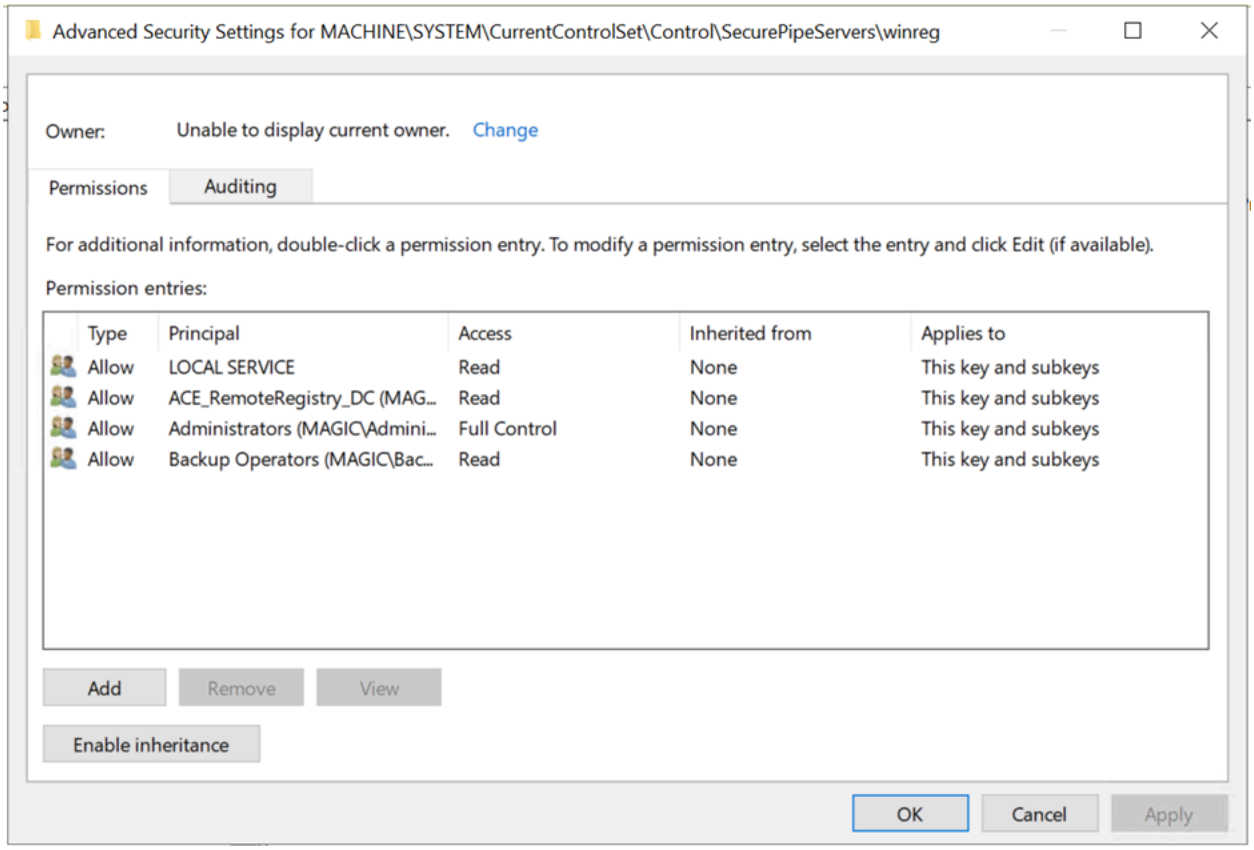


Note: I manually added my jsykora account on TellerDC01

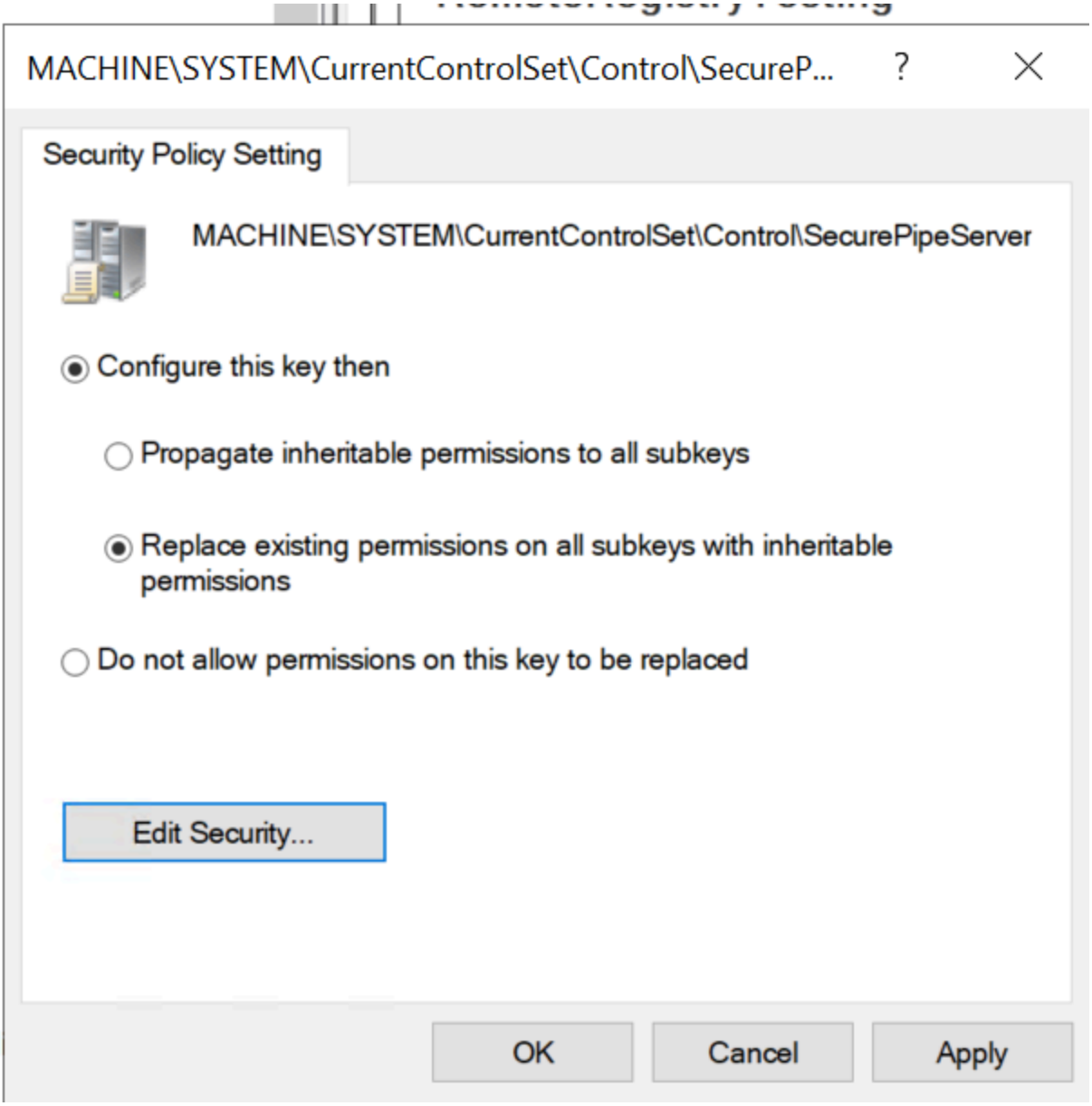
This did not have the desired effect:



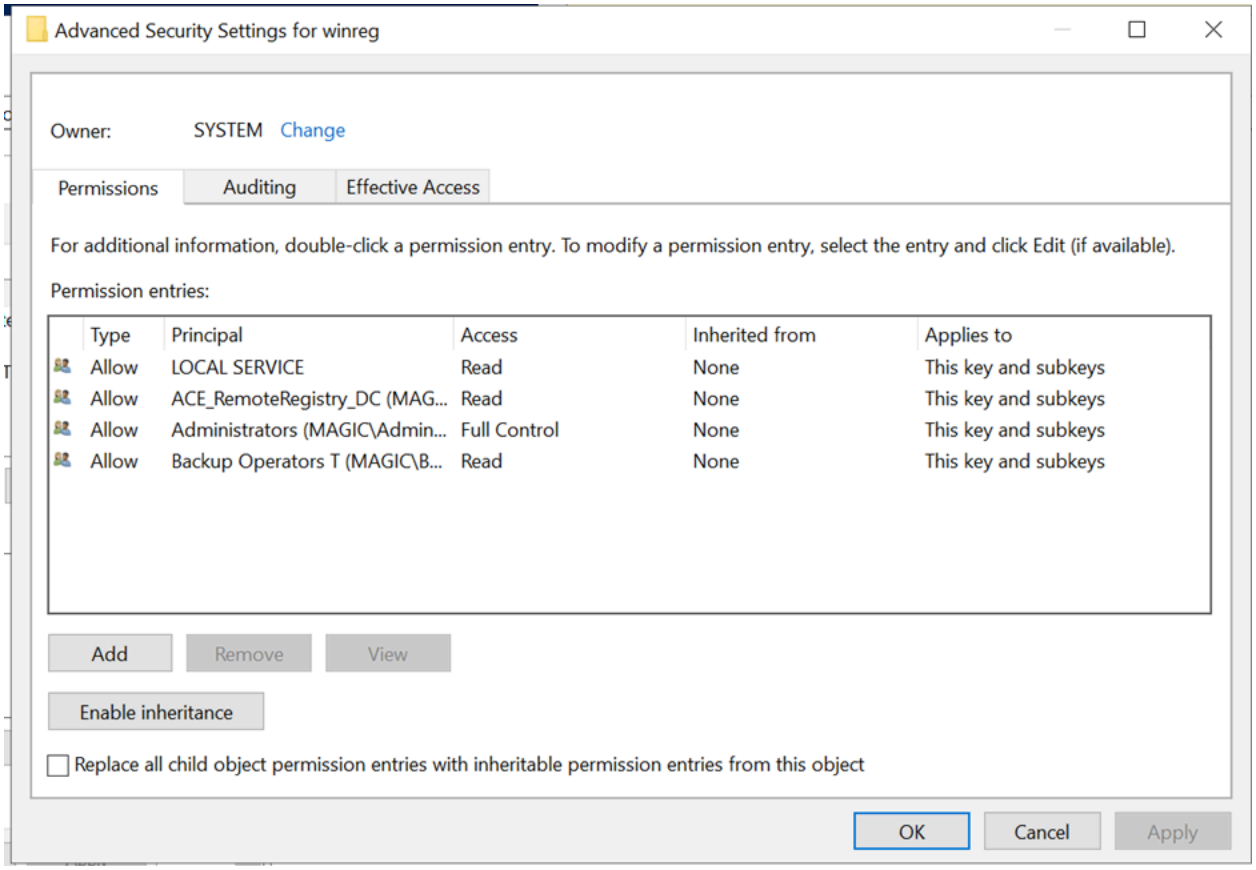
I edited the GPO to set the DACL like this:



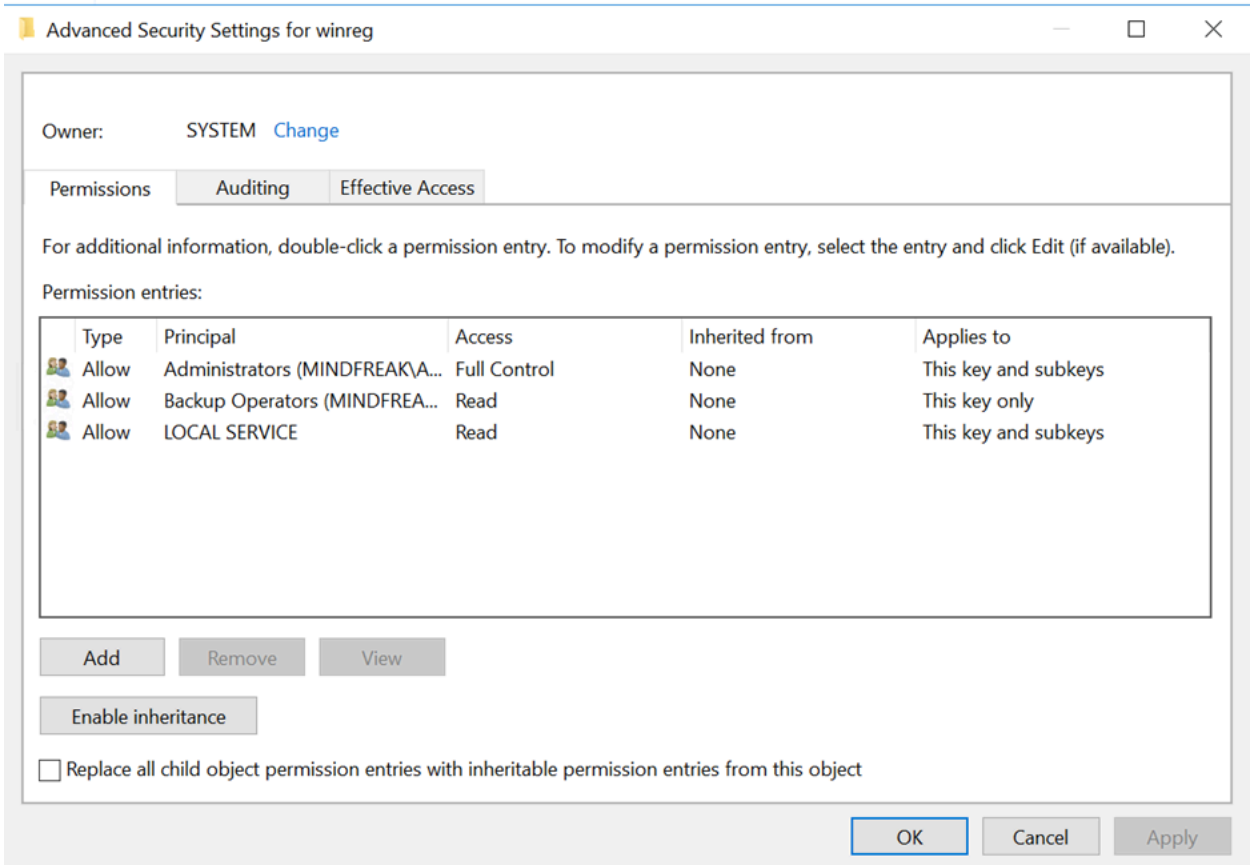
And configured the winreg key settings like this:



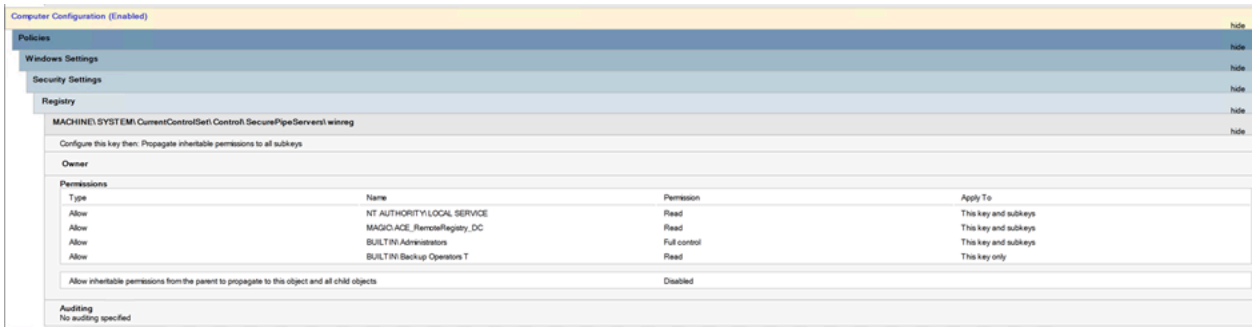
This appears to have the desired effect:



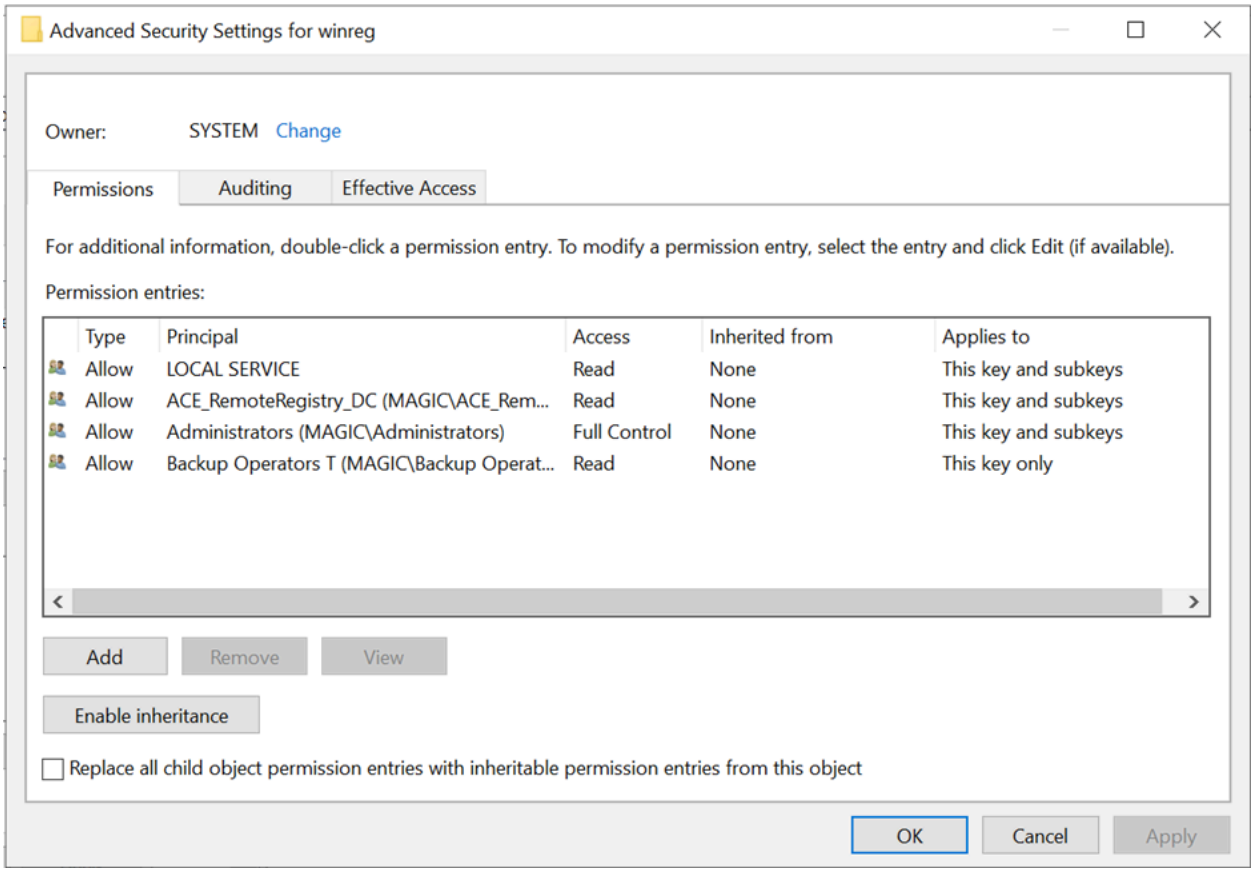
The default permissions on the winreg key are:



After further revision, I have a GPO setting configured like this:



This GPO setting results in a winreg security descriptor like this:



In SDDL, the winreg security descriptor looks like this:

O:SYG:SYD:PAI(A;CI;KR;;;LS)(A;CI;KA;;;BA)(A;;KR;;;BO)(A;CI;KR;;;S-1-5-21-3520149094-1197618848-1674470492-3101)

S-1-5-21-3520149094-1197618848-1674470492-3101 is the SID of the ACE_RemoteRegistry_DC group.

Testing the Winreg Allow ACE

I re-login to MagicPC01 as jsykora to refresh the group membership on my security context and validate that jsykora is now a member of ACE_RemoteRegistry_DC:

```
PS C:\Windows\system32> whoami /user /groups

USER INFORMATION
----------------

User Name     SID
============ =============================================
magic\jsykora S-1-5-21-3520149094-1197618848-1674470492-1273


GROUP INFORMATION
-----------------

Group Name                                 Type             SID                                            Attributes
========================================== ================ ============================================== ===============================================================
MAGIC\Creator_Group_Test                   Group            S-1-5-21-3520149094-1197618848-1674470492-1554 Mandatory group, Enabled by default, Enabled group
Everyone                                   Well-known group S-1-1-0                                        Mandatory group, Enabled by default, Enabled group
BUILTIN\Users                              Alias            S-1-5-32-545                                   Mandatory group, Enabled by default, Enabled group
BUILTIN\Remote Desktop Users               Alias            S-1-5-32-555                                   Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\REMOTE INTERACTIVE LOGON      Well-known group S-1-5-14                                       Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\INTERACTIVE                   Well-known group S-1-5-4                                        Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Authenticated Users           Well-known group S-1-5-11                                       Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\This Organization             Well-known group S-1-5-15                                       Mandatory group, Enabled by default, Enabled group
LOCAL                                      Well-known group S-1-2-0                                        Mandatory group, Enabled by default, Enabled group
Authentication authority asserted identity Well-known group S-1-18-1                                       Mandatory group, Enabled by default, Enabled group
MAGIC\SharpHoundRegistryAccess             Alias            S-1-5-21-3520149094-1197618848-1674470492-2155 Mandatory group, Enabled by default, Enabled group, Local Group
MAGIC\GGP-DenyCreateChildObject            Alias            S-1-5-21-3520149094-1197618848-1674470492-1548 Mandatory group, Enabled by default, Enabled group, Local Group
MAGIC\P-AllowCreateChildObject             Alias            S-1-5-21-3520149094-1197618848-1674470492-1550 Mandatory group, Enabled by default, Enabled group, Local Group
MAGIC\ACE_RemoteRegistry_DC                Alias            S-1-5-21-3520149094-1197618848-1674470492-3101 Mandatory group, Enabled by default, Enabled group, Local Group
MAGIC\GP-AllowCreateChildObject            Alias            S-1-5-21-3520149094-1197618848-1674470492-1549 Mandatory group, Enabled by default, Enabled group, Local Group
Mandatory Label\Medium Mandatory Level     Label            S-1-16-8192
```

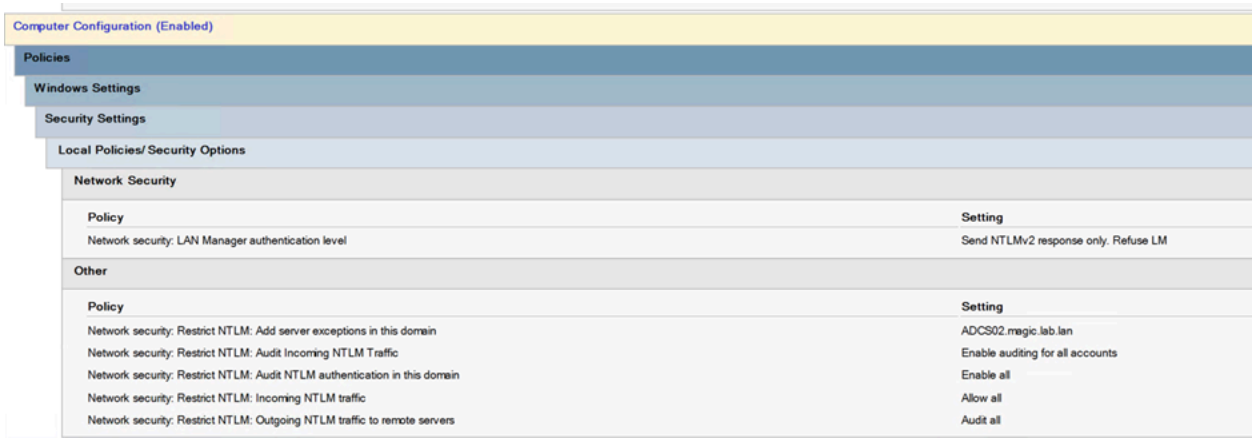
And now with this new group membership, running Test-RemoteRegistry.ps1 I get this output:

```
-----------------------------------
Path: HKLM\SYSTEM\CurrentControlSet\Services\Kdc\StrongCertificateBindingEnforcement -
-----------------------------------

-----------------------------------
Path: HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\CertificateMappingMethods -
-----------------------------------

-----------------------------------
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\MSV1_0\ClientAllowedNTLMServers -
-----------------------------------

-----------------------------------
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\MSV1_0\NtlmMinClientSec  Data: 536870912
-----------------------------------

-----------------------------------
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\MSV1_0\NtlmMinServerSec  Data: 536870912
-----------------------------------

-----------------------------------
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\MSV1_0\RestrictReceivingNTLMTraffic -
-----------------------------------

-----------------------------------
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\MSV1_0\RestrictSendingNTLMTraffic -
-----------------------------------

-----------------------------------
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\LMCompatibilityLevel -
-----------------------------------

-----------------------------------
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\UseMachineId -
-----------------------------------

-----------------------------------
Path: HKLM\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters\EnableSecuritySignature  Data: 1
-----------------------------------

-----------------------------------
Path: HKLM\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters\RequireSecuritySignature  Data: 1
-----------------------------------

-----------------------------------
Path: HKLM\System\CurrentControlSet\Control\ProductOptions\ProductSuite  Data: Terminal Server
-----------------------------------

-----------------------------------
Path: HKLM\Software\Microsoft\Windows NT\CurrentVersion\ProductName  Data: Windows Server 2019 Standard
-----------------------------------

-----------------------------------
Path: HKLM\System\CurrentControlSet\Control\Print\Printers\DefaultSpoolDirectory  Data: C:\Windows\system32\spool\PRINTERS
-----------------------------------

-----------------------------------
Path: HKLM\System\CurrentControlSet\Services\Eventlog\RequiredPrivileges  Data: SeChangeNotifyPrivilege SeImpersonatePrivilege SeAuditPrivilege
-----------------------------------

-----------------------------------
Path: HKLM\Software\Microsoft\Windows NT\CurrentVersion\Print\DoNotInstallCompatibleDriverFromWindowsUpdate  Data: 1
-----------------------------------

-----------------------------------
Path: HKLM\Software\Microsoft\Windows NT\CurrentVersion\Windows\Spooler  Data: yes
-----------------------------------

PS C:\Windows\system32>
```

https://github.com/JimSycurity/LeastPrivilegeSharpHound/blob/main/RemoteRegistry/Data/TellerDC01-Test-RemoteRegistry02.txt

Several keys show with no value, which is because those values are not defined. However, we can now retrieve the values for NtlmMinClientSec & NtlmMinServerSec.

And if we create and link a GPO to the Domain Controllers OU with settings like this:



Computer Configuration (Enabled)

Policies

Windows Settings

Security Settings

Local Policies/ Security Options

Network Security

Policy	Setting
Network security: LAN Manager authentication level	Send NTLMv2 response only. Refuse LM

Other

Policy	Setting
Network security: Restrict NTLM: Add server exceptions in this domain	ADCS02.magic.lab.lan
Network security: Restrict NTLM: Audit Incoming NTLM Traffic	Enable auditing for all accounts
Network security: Restrict NTLM: Audit NTLM authentication in this domain	Enable all
Network security: Restrict NTLM: Incoming NTLM traffic	Allow all
Network security: Restrict NTLM: Outgoing NTLM traffic to remote servers	Audit all

Test-RemoteRegistry returns this:

```
----------
Path: HKLM\SYSTEM\CurrentControlSet\Services\Kdc\StrongCertificateBindingEnforcement -
----------

----------
Path: HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\CertificateMappingMethods -
----------

----------
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\MSV1_0\ClientAllowedNTLMServers -
----------

----------
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\MSV1_0\NtlmMinClientSec  Data: 536870912
----------

----------
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\MSV1_0\NtlmMinServerSec  Data: 536870912
----------

----------
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\MSV1_0\RestrictReceivingNTLMTraffic -
----------

----------
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\MSV1_0\RestrictSendingNTLMTraffic  Data: 1
----------

----------
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\LMCompatibilityLevel  Data: 4
----------

----------
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\UseMachineId -
----------

----------
Path: HKLM\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters\EnableSecuritySignature  Data: 1
----------

----------
Path: HKLM\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters\RequireSecuritySignature  Data: 1
----------

----------
Path: HKLM\System\CurrentControlSet\Control\ProductOptions\ProductSuite  Data: Terminal Server
----------

----------
Path: HKLM\Software\Microsoft\Windows NT\CurrentVersion\ProductName  Data: Windows Server 2019 Standard
----------

----------
Path: HKLM\System\CurrentControlSet\Control\Print\Printers\DefaultSpoolDirectory  Data: C:\Windows\system32\spool\PRINTERS
----------

----------
Path: HKLM\System\CurrentControlSet\Services\Eventlog\RequiredPrivileges  Data: SeChangeNotifyPrivilege SeImpersonatePrivilege SeAuditPrivilege
----------

----------
Path: HKLM\Software\Microsoft\Windows NT\CurrentVersion\Print\DoNotInstallCompatibleDriverFromWindowsUpdate  Data: 1
----------

----------
Path: HKLM\Software\Microsoft\Windows NT\CurrentVersion\Windows\Spooler  Data: yes
----------
```

Adjusting the winreg permissions, whether manually, or via the GPOs which modify the AllowPaths and/or AllowedExactPaths provides access to the winreg named pipe. This is similar to Share permissions when thinking of an NTFS file share. The underlying registry permissions are still applicable, regardless of how winreg is modified. This is no different than the NTFS permissions on a shared file or directory still being effective at controlling access even if the Share Permissions allow Everyone Full Control.

I added some additional registry path entries to Test-RemoteRegistry.ps1:

```
    # Other sensitive or potentially sensitive paths:
    Secrets                                 = 'SECURITY\Policy\Secrets'
    AEPolicy                                = 'SOFTWARE\Policies\Microsoft\Cryptography\AutoEnrollment'
    Sam                                     = 'SAM\SAM\Domains\Account\Users'
    C                                       = 'SAM\SAM\Domains\Builtin\Aliases\00000220'
    Winlogon                                = 'SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon'
    Security                                = 'SYSTEM\CurrentControlSet\Services\Kdc\Security'
    SocketAddressList                       = 'SYSTEM\CurrentControlSet\Services\Netlogon\Private'
    Blob                                    = 'SOFTWARE\Policies\Microsoft\SystemCertificates\Root\Certificates\F0796D513217181A3C5A9372E56952634A16B6E7'
    OSManagedAuthLevel                      = 'SOFTWARE\Policies\Microsoft\TPM'
    FriendlyTypeName                        = 'SOFTWARE\Classes\.symlink'
)
```

And then ran this under the context of jsykora from MagicPC01 and received this output:

https://github.com/JimSycurity/LeastPrivilegeSharpHound/blob/main/RemoteRegistry/Data/TellerDC01-Test-RemoteRegistry03.txt

An entry like this means that no value is present:

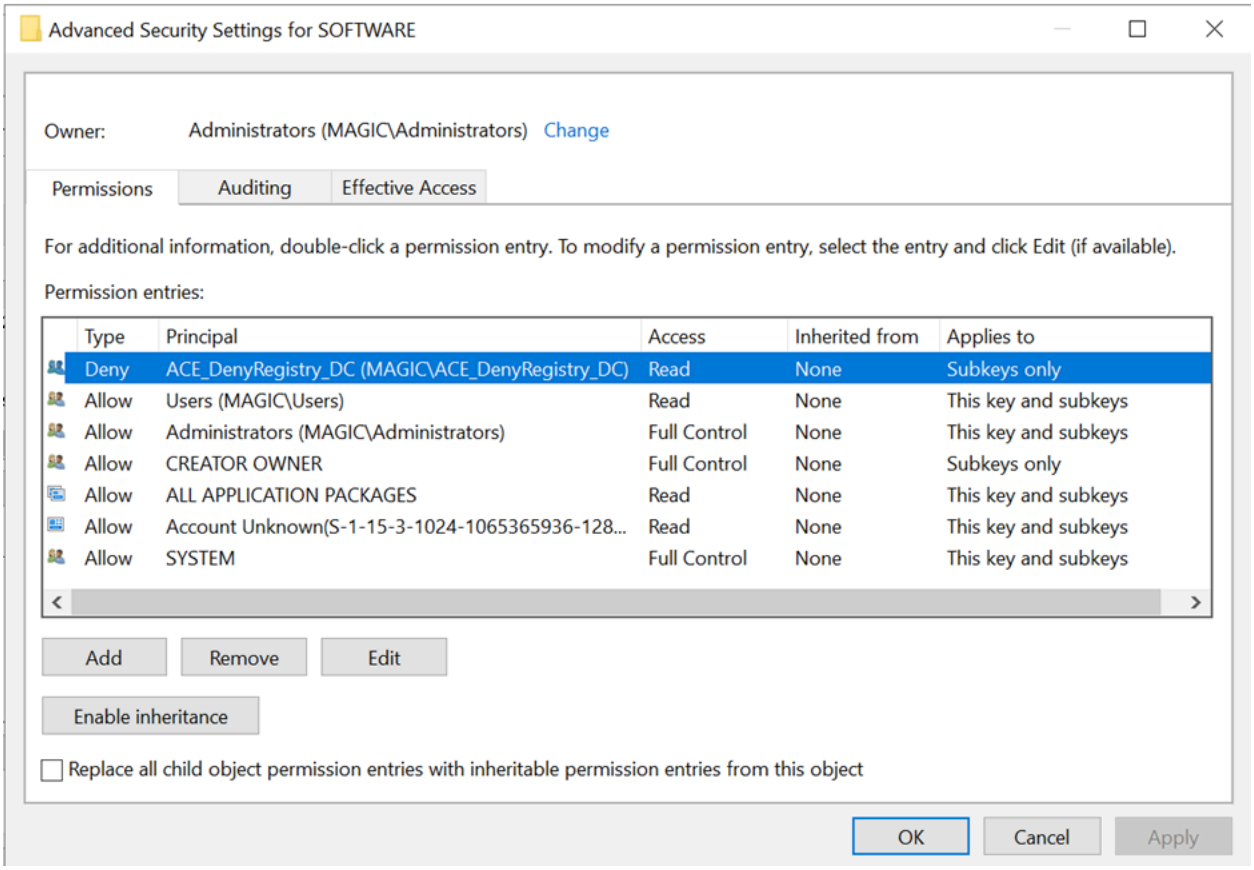
Path: HKLM\SYSTEM\CurrentControlSet\Control\Lsa\UseMachineId -

Whereas a value like this means that we have no ability to open the subkey specified:

Path: HKLM\SECURITY\Policy\Secrets\Secrets - OpenSubKey Error

Testing Winreg Allow ACE with HKLM/Software Deny ACE

In the magic.lab.lan domain I created another domain local security group ‘ACE_DenyRegistry_DC’, added my test account jsykora to it, and then manually created a Deny ACE at HKLM\SOFTWARE which will inherit down to any child key which does not have the DACL_Protected flag set:



I relogged on to MagicPC01 as jsykora to update group membership.

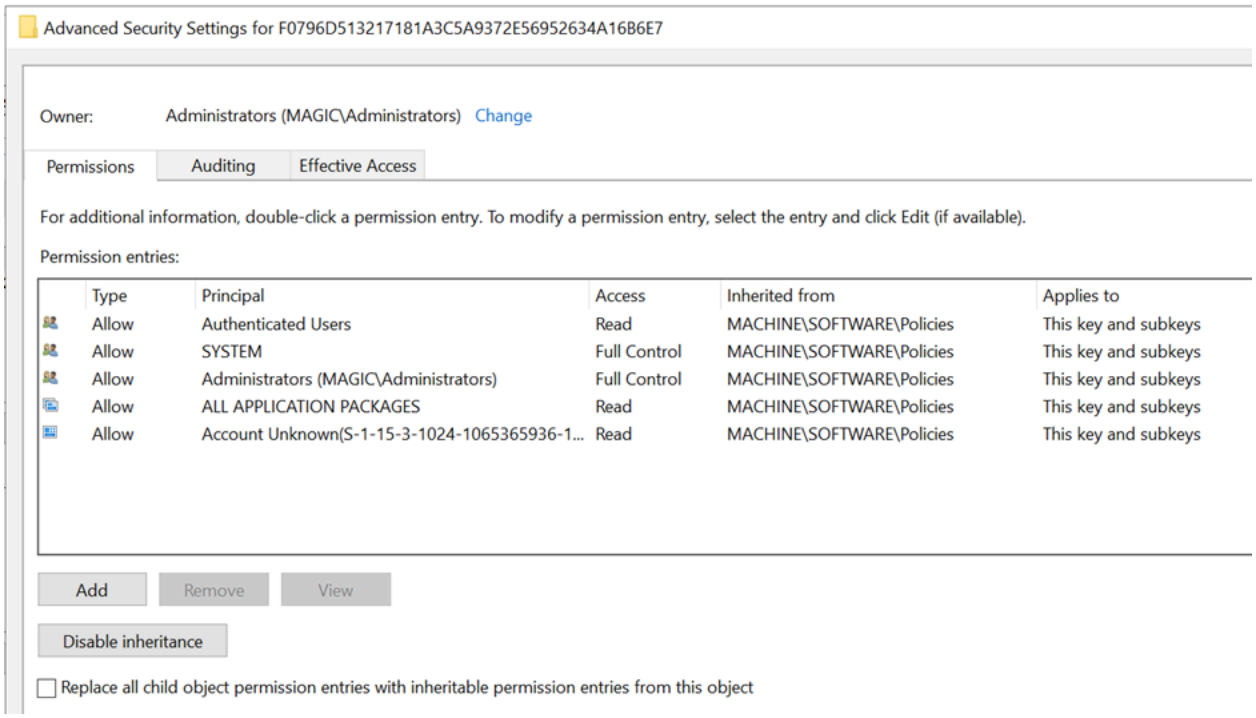
Honestly, this inherited Deny ACE was a fool's errand as most keys worth protecting will have their own explicit DACLs with the DACL_Protected flag enabled so that inheritance is disabled. So this attempt at utilizing an inherited Deny ACE predicably didn't have much effect, except for this path:

```
----------------------------------
Path: HKLM\SOFTWARE\Classes\.symlink\FriendlyTypeName - OpenSubKey Error
----------------------------------
```

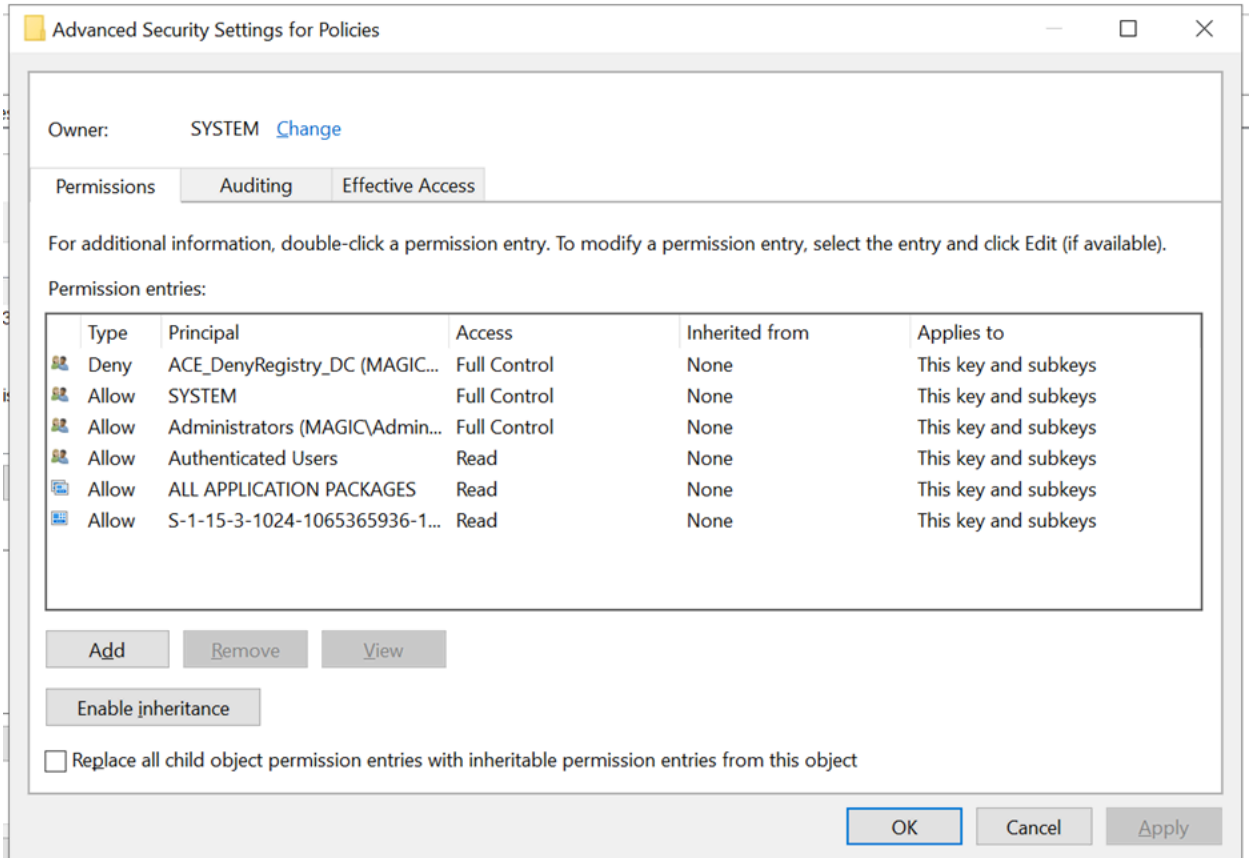
https://github.com/JimSycurity/LeastPrivilegeSharpHound/blob/main/RemoteRegistry/Data/TellerDC01-Test-RemoteRegistry04.txt

However, if there are certain registry key paths which we wish to deny access to we can target specific locations in the registry. For example, if I wanted to block access to 'HKLM\SOFTWARE\Policies\Microsoft\SystemCertificates\Root\Certificates\F0796D513217181A3C5A9372E56952634A16B6E7\Blob' by the jsykora account, I could review the DACL for the

F0796D513217181A3C5A9372E56952634A16B6E7 subkey as such, noting that its permissions propagate from HKLM\SOFTWARE\Policies:



And apply a deny ACE at HKLM\SOFTWARE\Policies:



However, this Deny is realistically only overriding the Allow Authenticated Users Read ACE which also propagates down.

This configuration does block access to the Path: HKLM\SOFTWARE\Policies\Microsoft\SystemCertificates\Root\Certificates\F0796D513217181A3C5A9372E56952634A16B6E7 key, however.

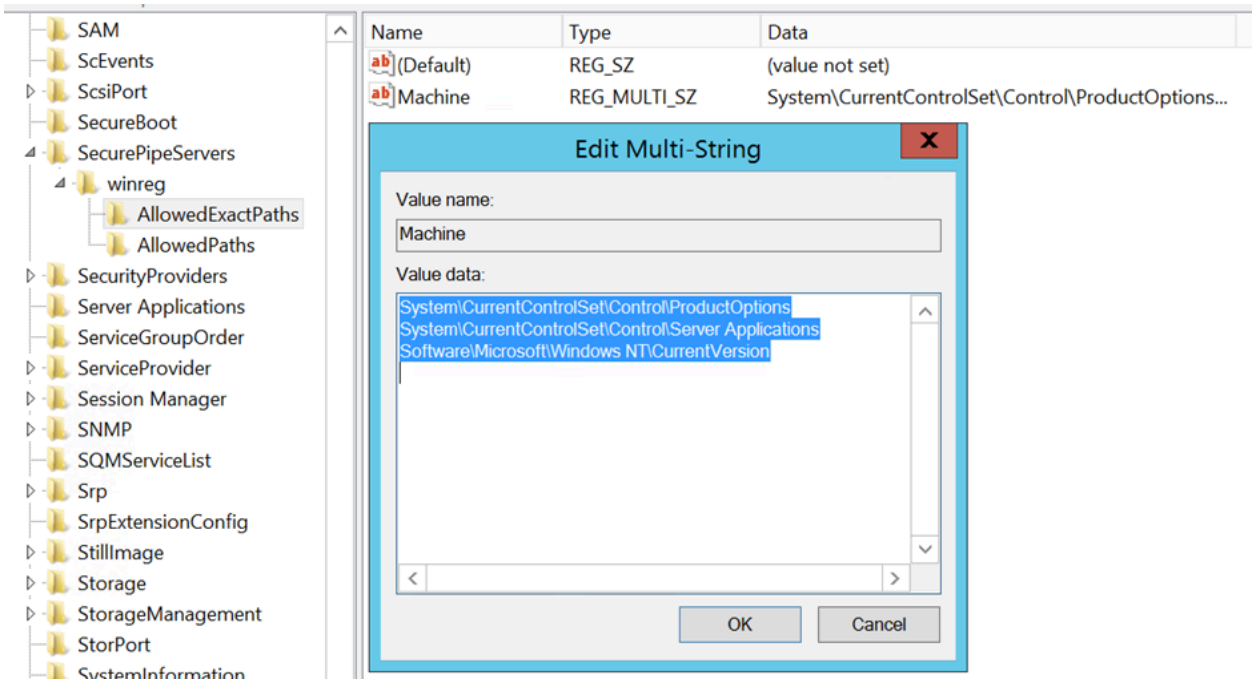
```
----------------------------------
Path: HKLM\SOFTWARE\Policies\Microsoft\SystemCertificates\Root\Certificates\F0796D513217181A3C5A9372E56952634A16B6E7\Blob - OpenSubKey Error
----------------------------------
```

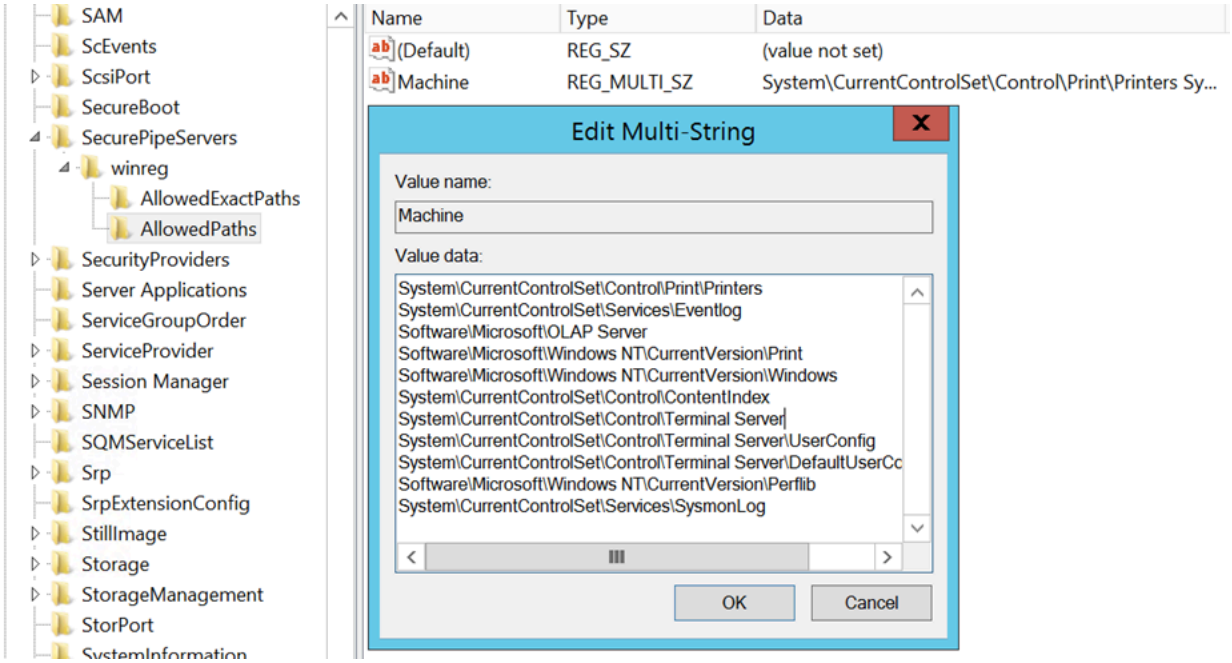
https://github.com/JimSycurity/LeastPrivilegeSharpHound/blob/main/RemoteRegistry/Data/TellerDC01-Test-RemoteRegistry05.txt

Documenting Changes to Network access: Remotely accessible registry…

At this point, I still haven't configured any GPOs with the Network access: Remotely accessible registry paths (and subpaths) settings. In a brief view of the NT5.1 source code I found several instances of subsystems adding their own exemptions to AllowedExactPaths or AllowedPaths subkeys in the winreg subkey. So first I'll document the native values of these before I modify any GPO settings.

TellerDC02 – Windows Server 2012R2:





System\CurrentControlSet\Control\ProductOptions

System\CurrentControlSet\Control\Server Applications

Software\Microsoft\Windows NT\CurrentVersion

System\CurrentControlSet\Control\Print\Printers

System\CurrentControlSet\Services\Eventlog

Software\Microsoft\OLAP Server

Software\Microsoft\Windows NT\CurrentVersion\Print

Software\Microsoft\Windows NT\CurrentVersion\Windows

System\CurrentControlSet\Control\ContentIndex

System\CurrentControlSet\Control\Terminal Server

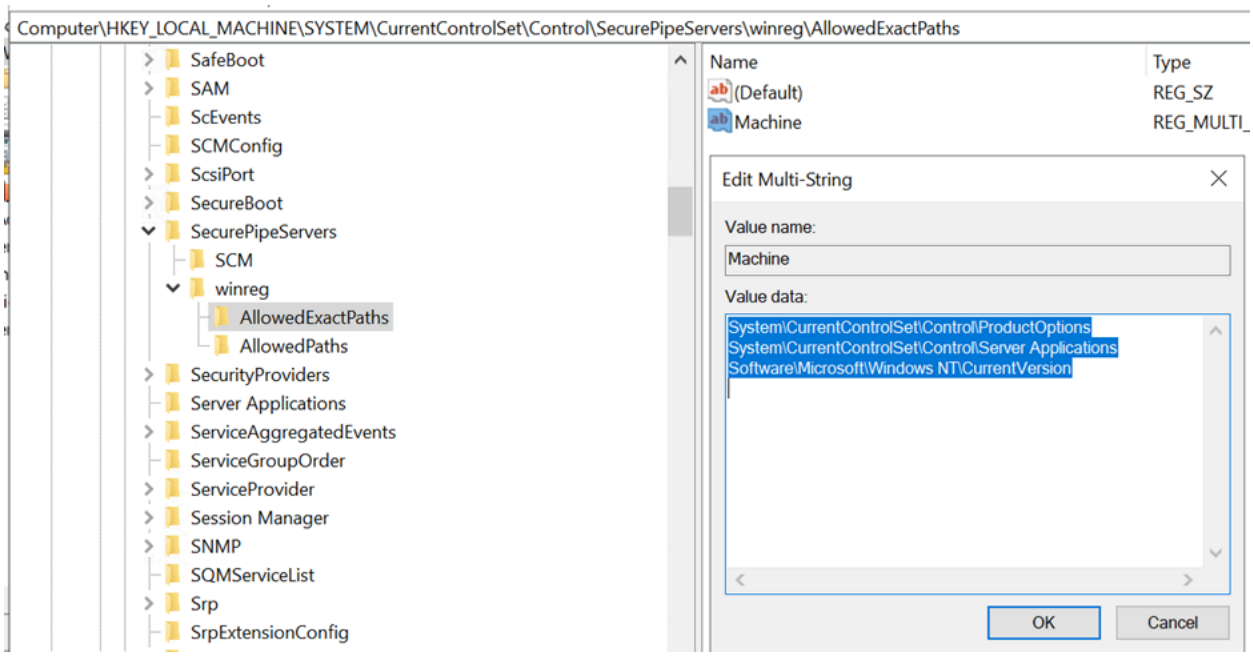
System\CurrentControlSet\Control\Terminal Server\UserConfig

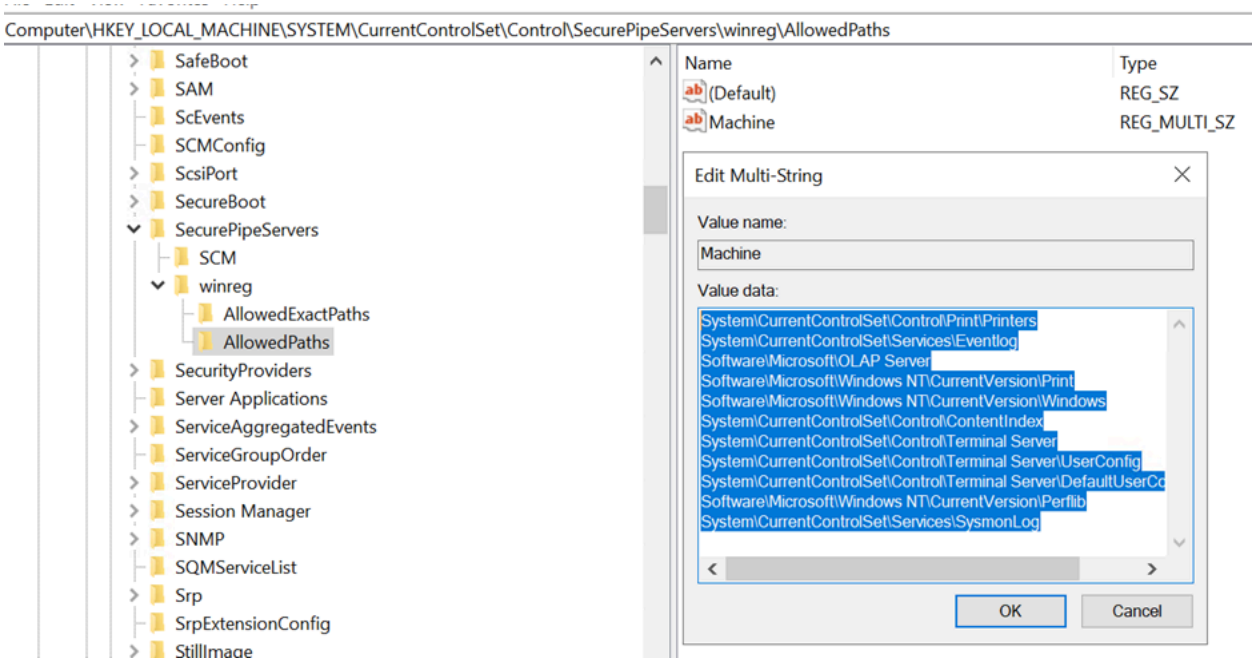
System\CurrentControlSet\Control\Terminal Server\DefaultUserConfiguration

Software\Microsoft\Windows NT\CurrentVersion\Perflib

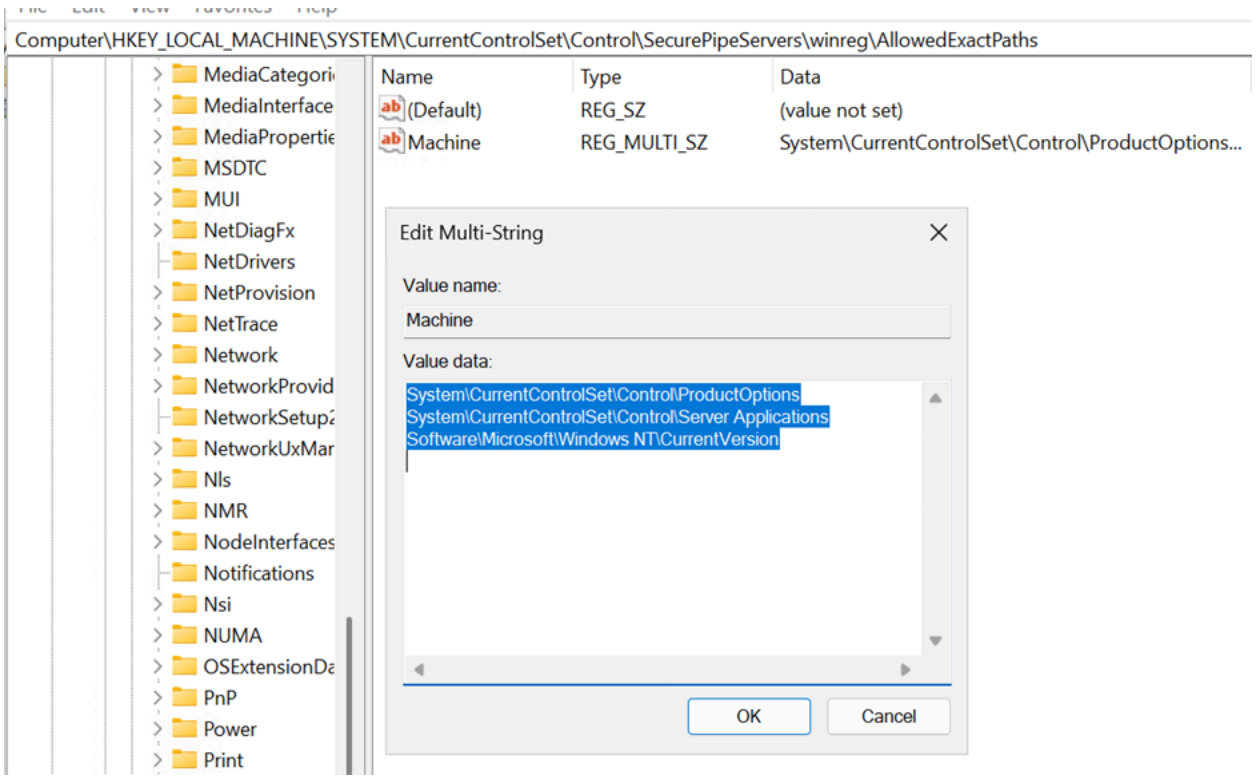
System\CurrentControlSet\Services\SysmonLog

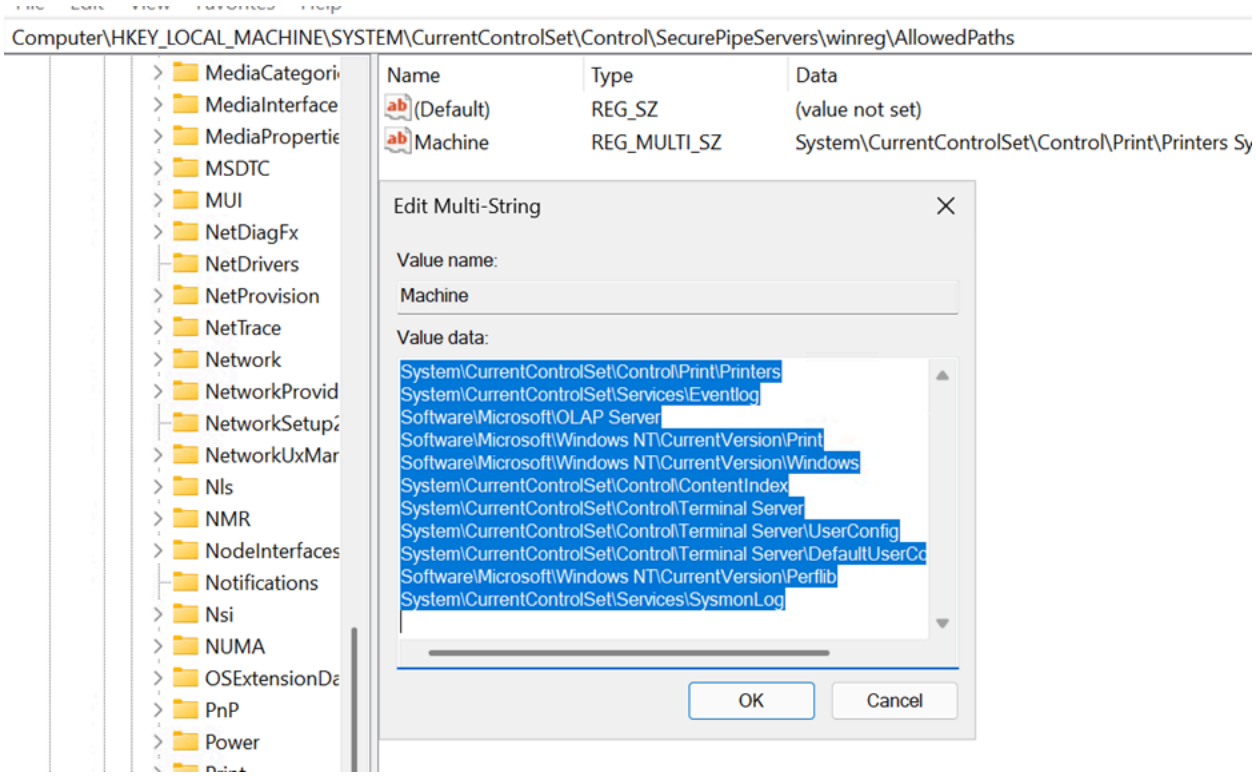
TellerDC01 – Windows Server 2019:





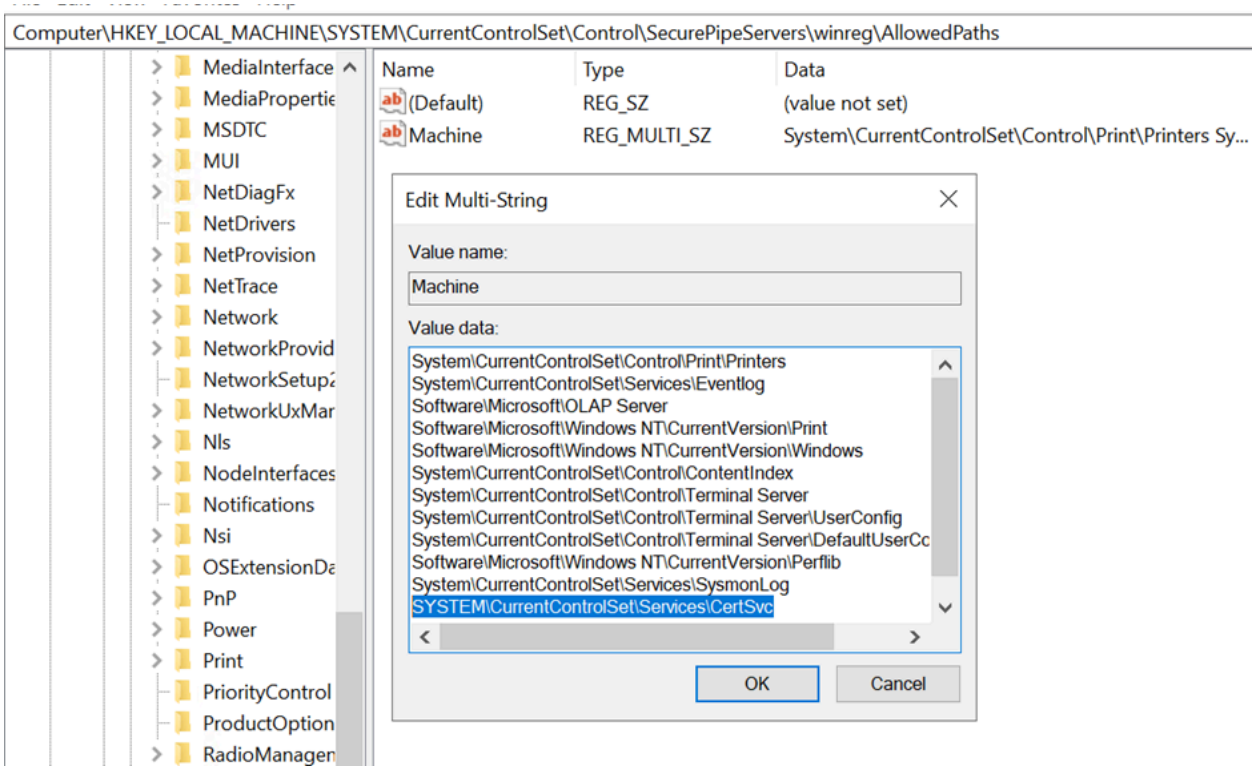
InheritanceII – Windows Server 2025:





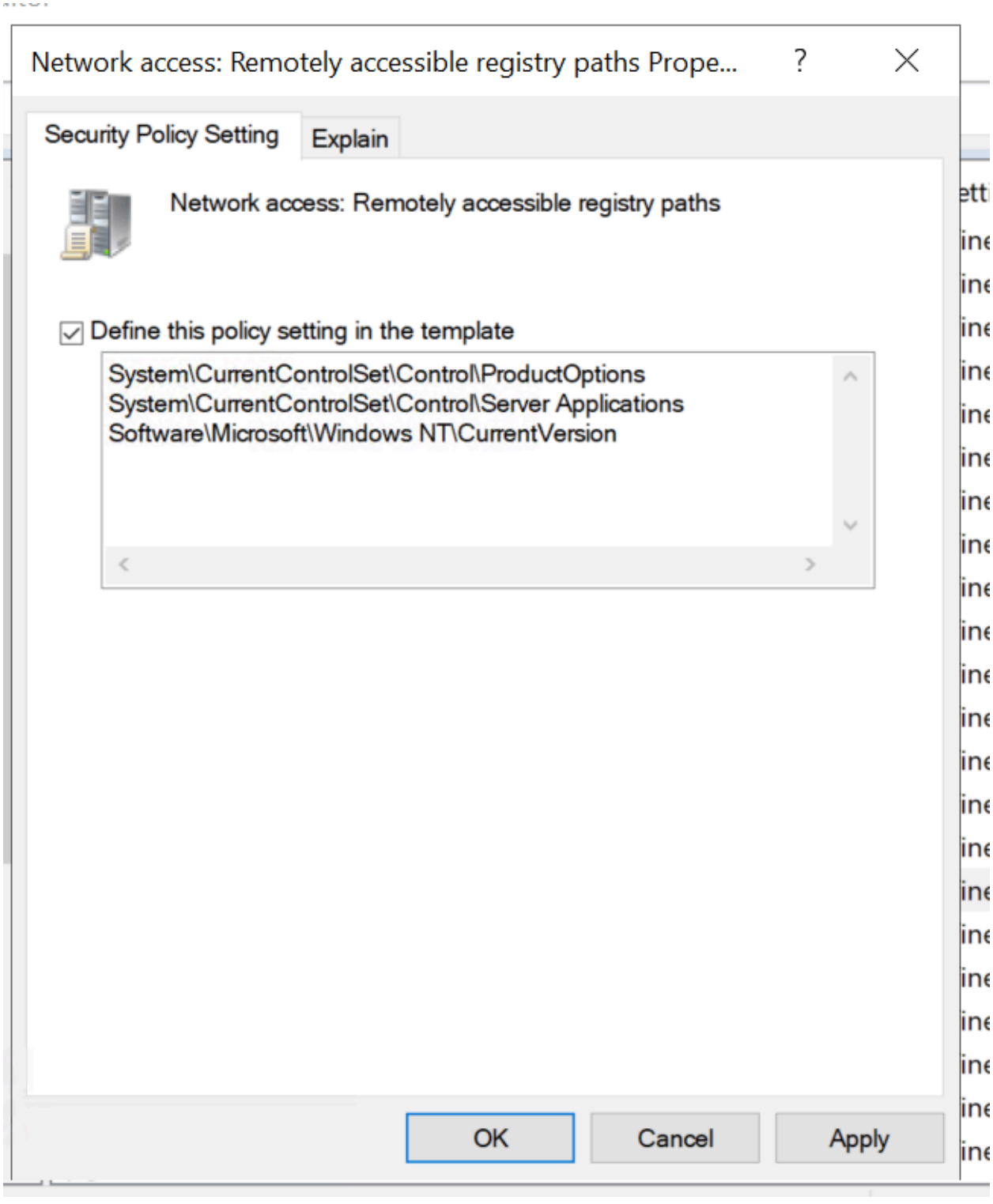
And because we don’t only care about Domain Controllers, here is from an issuing CA:

ADCS02 – Windows Server 2019 CA:

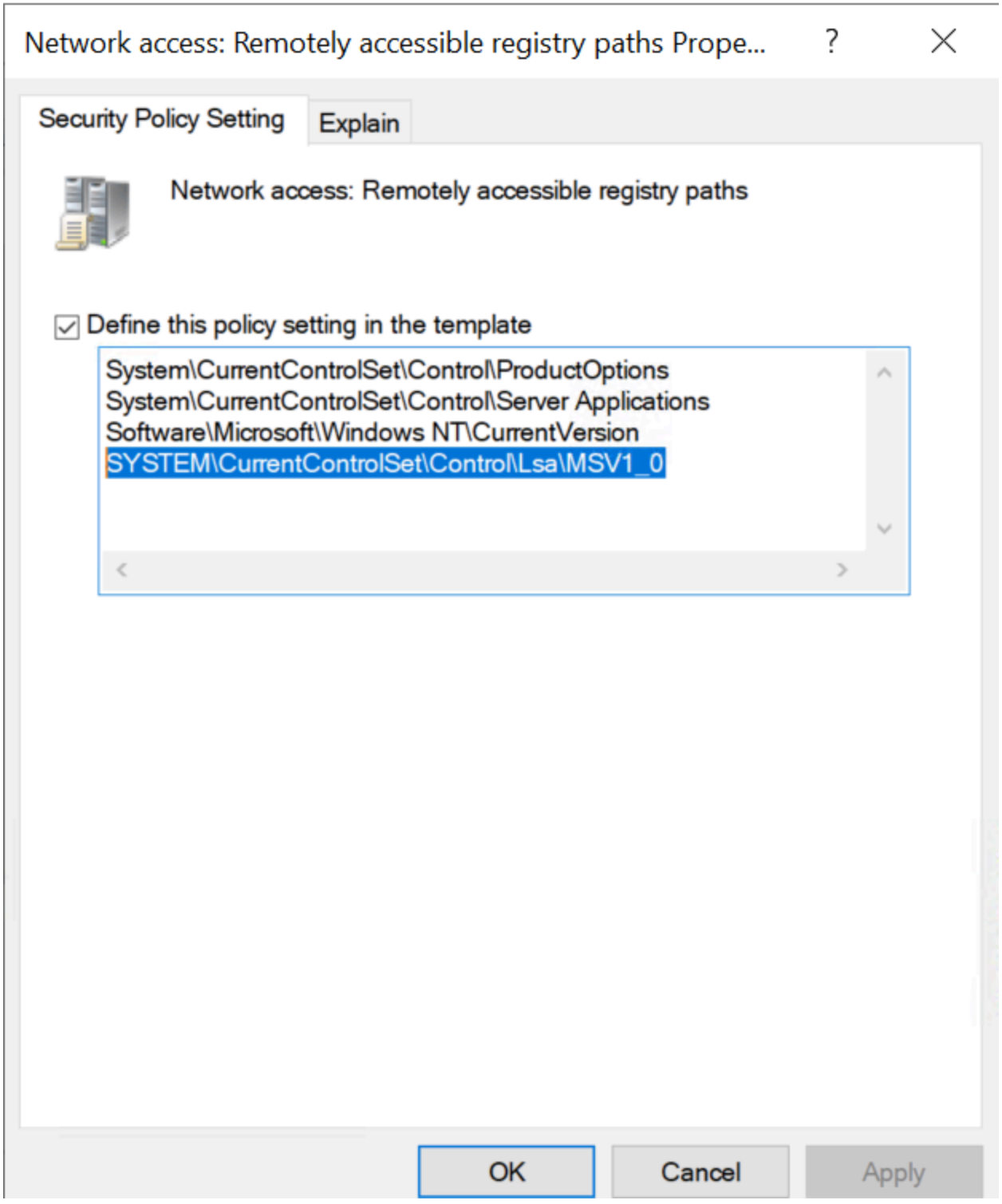


Here I'm just highlighting the difference. Installing the AD CS role (or perhaps activating it) adds the CertSvc subkey to the AllowedPaths exception.

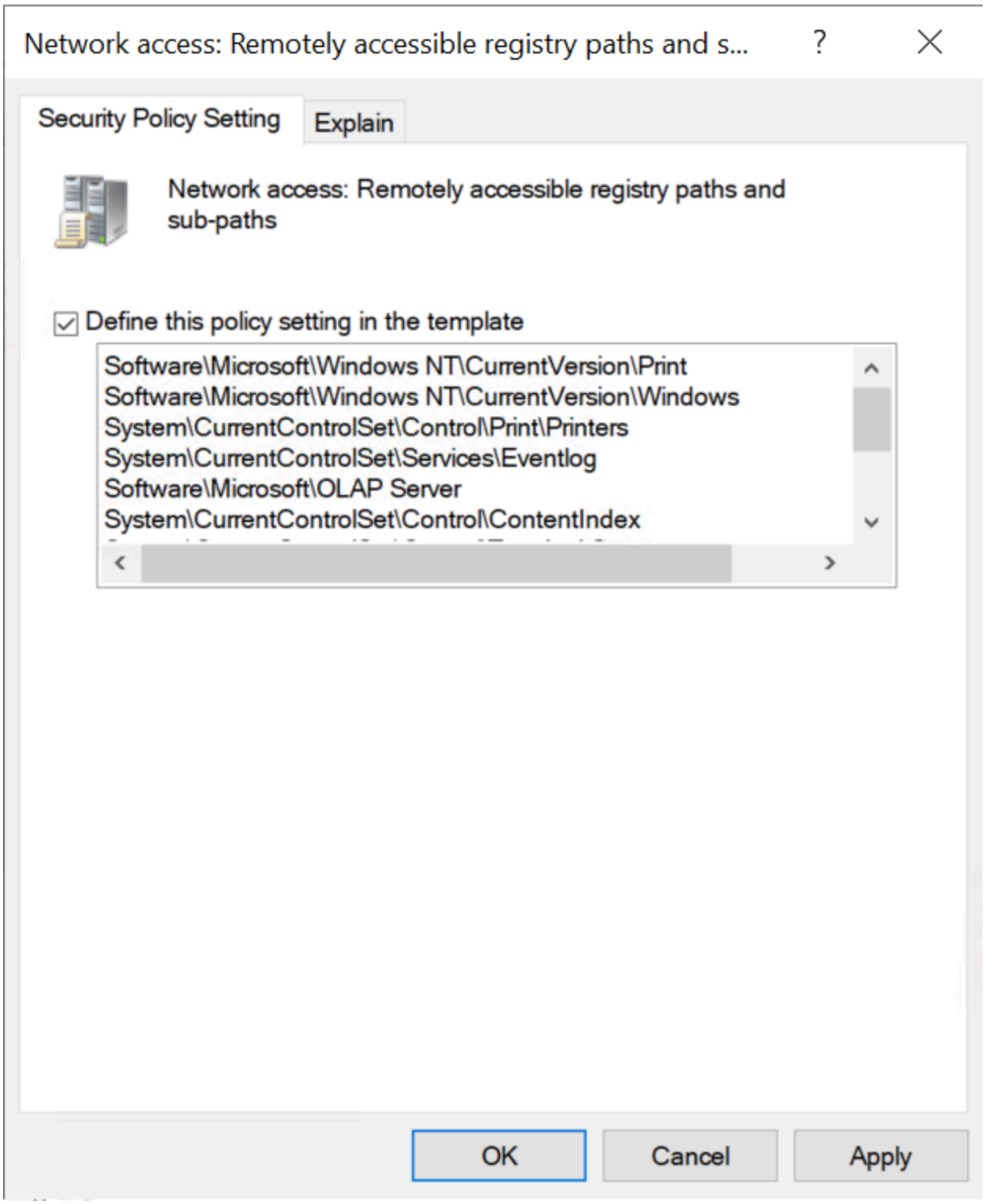
For the GPO settings, when I attempt to define the setting for Network access: Remotely accessible registry paths, the predefined paths populate automatically:



As such, if I were to just add a key it doesn't appear this would overwrite anything. To test I added this SharpHound required path:



Network access: Remotely accessible registry paths and sub-paths also prepopulates the default paths when defined:



Software\Microsoft\Windows NT\CurrentVersion\Print

Software\Microsoft\Windows NT\CurrentVersion\Windows

System\CurrentControlSet\Control\Print\Printers

System\CurrentControlSet\Services\Eventlog

Software\Microsoft\OLAP Server

System\CurrentControlSet\Control\ContentIndex

System\CurrentControlSet\Control\Terminal Server

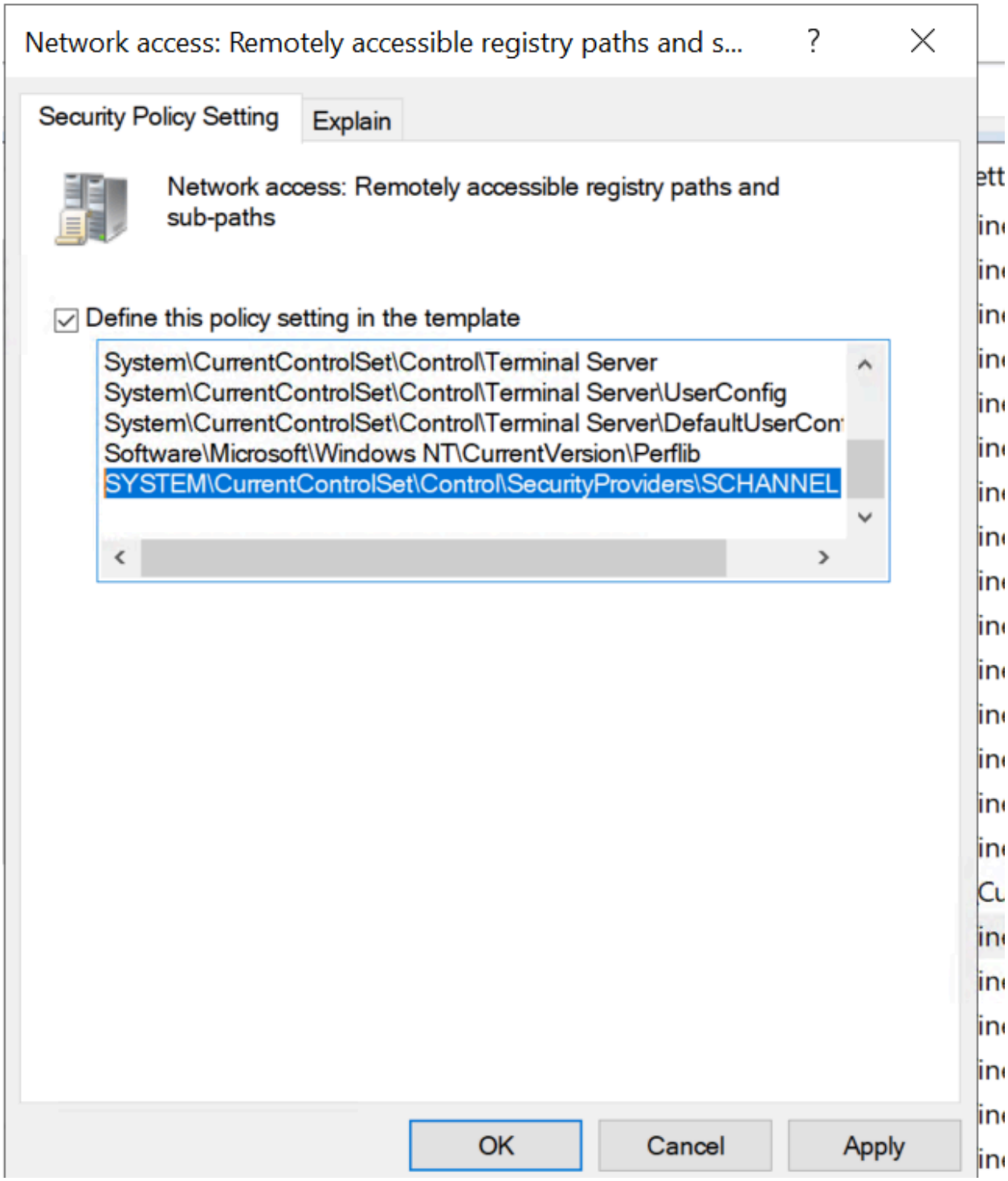
System\CurrentControlSet\Control\Terminal Server\UserConfig

System\CurrentControlSet\Control\Terminal Server\DefaultUserConfiguration

Software\Microsoft\Windows NT\CurrentVersion\Perflib

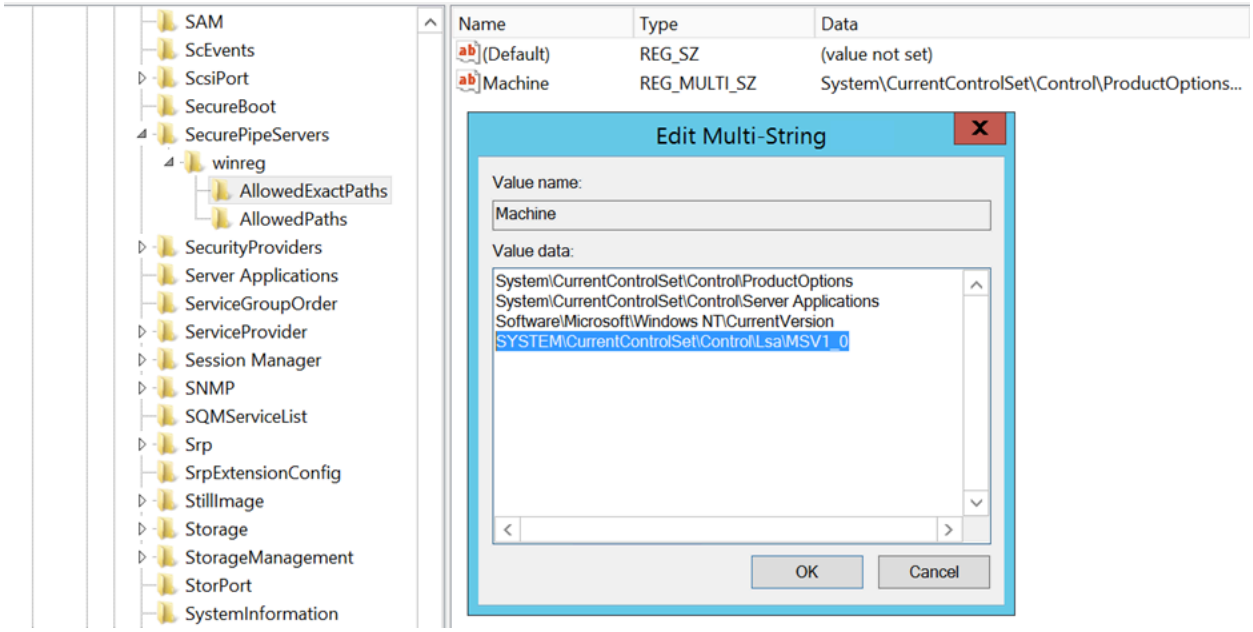
System\CurrentControlSet\Services\SysmonLog

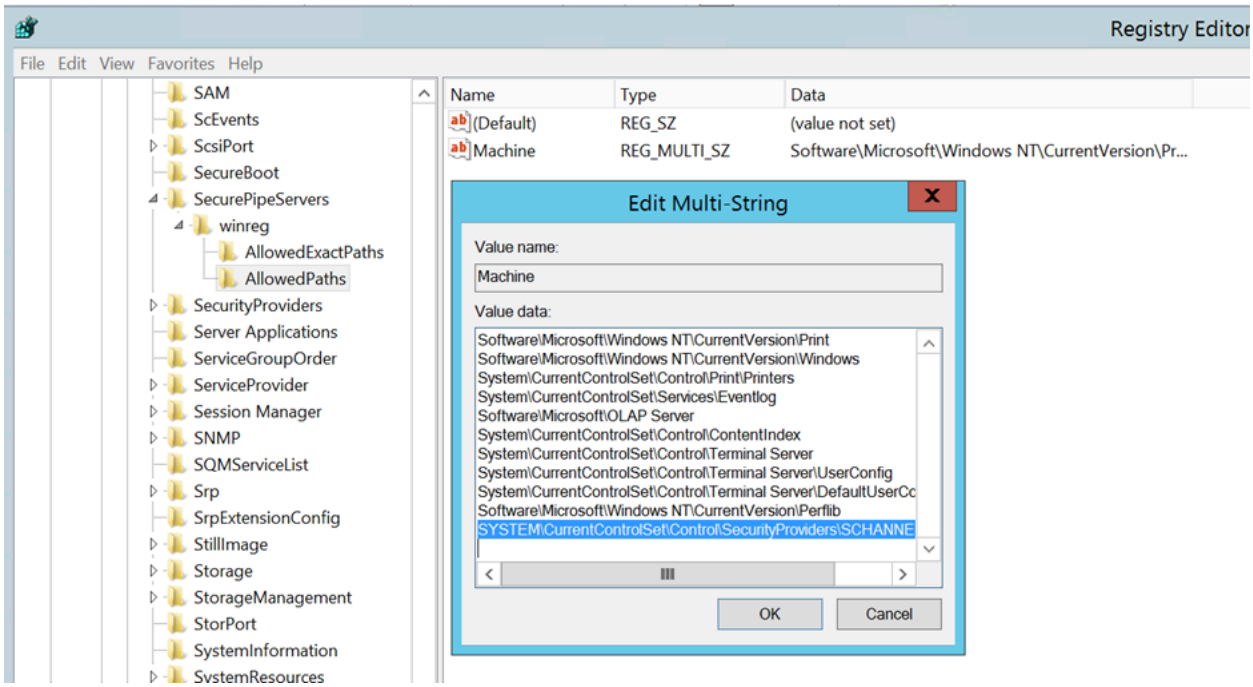
I suspect that if this policy were defined and applied to a scope that includes an ADCS host it would override the certsvc path. To test this, I will remove the SysmonLog path and add a SharpHound path:



I've configured this GPO to only apply to the scope of the Domain Controllers OU in magic.lab.lan

Results on TellerDC02:





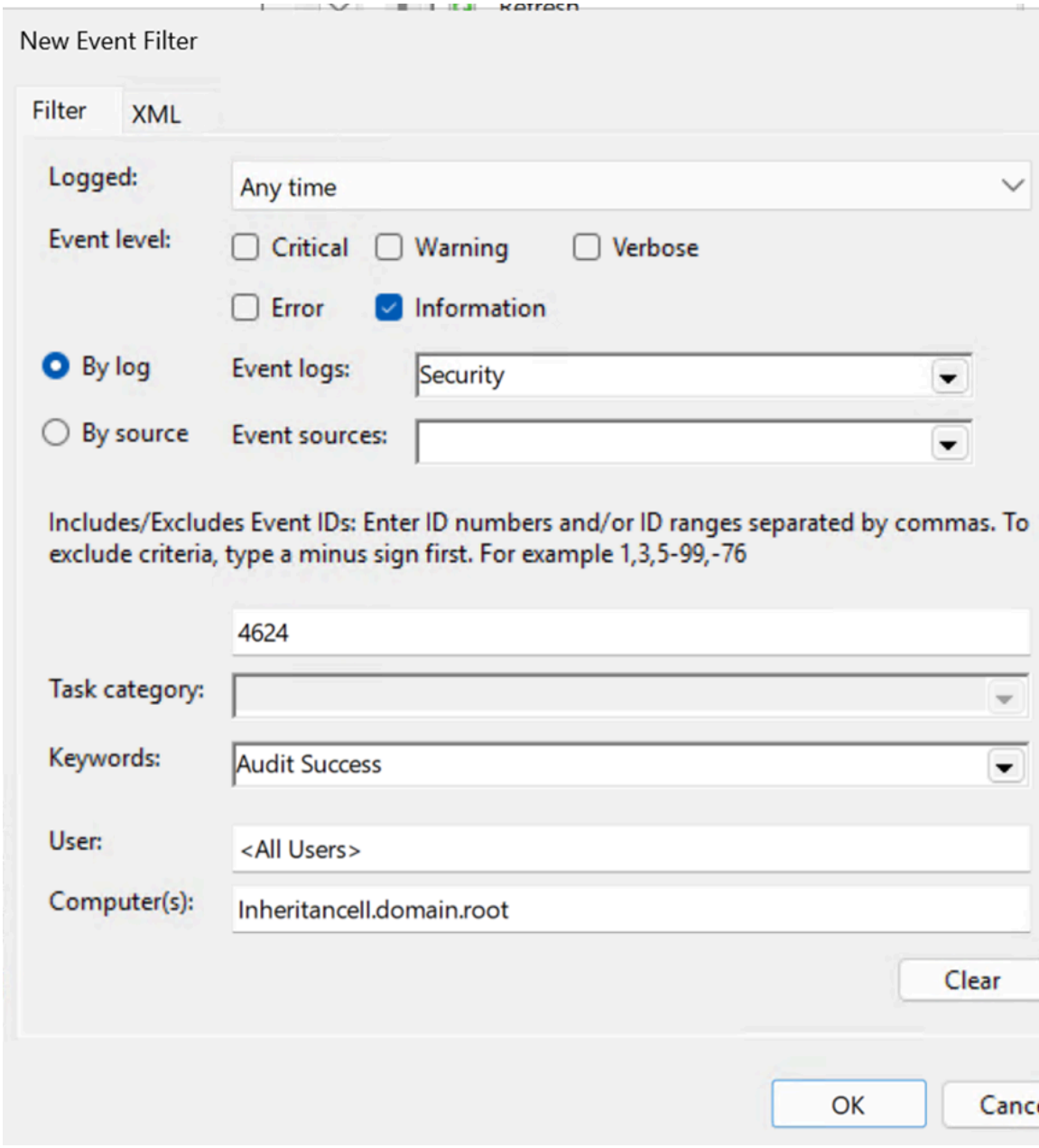
Note that removing the Sysmon subkey from the subpaths setting in GPO caused it to be removed from the Machine registry value. Applying this GP setting blanket across a domain could be impactful to exemptions that are created by installed roles or software that are not part of the default state.

It may be prudent to script a capture of these settings across a fleet prior to making mass changes via GPO.

Additional Least Privilege Controls

- Can Remote Registry be scheduled to be enabled only when SharpHound collection occurring?
    - Elad had an idea to trigger a scheduled task to start the Remote Registry service on an EventID of the SharpHound service account successfully logging on
        - § There are likely better ways to do this, however, filtering on a successful 4624 from the computer host where SharpHound collection is being performed is a good first step:

§



- Would it be feasible (and secure) to grant the SharpHound collector account rights to start and stop the Remote Registry service?

Other Notes:

- When Remote Registry is running, do Authenticated Users have rights to read permissions on all registry subkeys? What are some “sensitive” registry keys?

- No. Remote Registry does not modify or change permissions or access checks on the registry or registry keys.

- When Remote Registry is running, only the keys that are specified in the winreg\AllowedExactPaths and winreg\AllowedPaths bypass the DACL on the winreg key, which controls Named Pipe access to winreg, which is how Remote Registry functions.

- Only registry keys which grant Authenticated Users GenericRead permissions will grant Authenticated Users the ability to read permissions on registry keys.

- Can we utilize inherited DENY ACEs for a trustee like Network logon with an implicit Allow ACE for the SharpHound service account?

  - Perhaps, but deny ACEs can get messy quickly. With the understanding that the winreg key controls access to the Remote Registry 'winreg' named pipe

- In the NT5.1 source, setuput.cpp includes a function AddCARegKeyToRegConnectExemptions(), which adds remote registry exemptions for CA keys:

```
4927    HRESULT
4928    AddCARegKeyToRegConnectExemptions()
4929    {
4930        // add ourselves to list of people that take ACLs seriously
4931        // and should be allowed to reveal our key to outsiders.
```

There are several places in the NT5.1 source which create exemptions in the winreg subkeys for specific software, any of which could be

- In the NT5.1 source, registry.cxx implements the functions for dealing with the windows registry:

  - OpenKey() – Opens a handle to a key

    § Calls BuildCompleteName()

    § If CompleteName is empty assumes a predefined key in which case it checks to make sure it's not a RemoteRegistry with SACL access

    · Predefined keys are:

      - PREDEFINED_KEY_CLASSES_ROOT,
      - PREDEFINED_KEY_CURRENT_USER,
      - PREDEFINED_KEY_LOCAL_MACHINE,

        - PREDEFINED_KEY_USERS,
        - PREDEFINED_KEY_CURRENT_CONFIG
    - § Calls RegOpenKeyEx()
- QueryKeyInfo() – Retrieve the information of a key.
    - § If OpenKey() succeeds, the value can be read if it exists as the access check already occurred.
- QueryKeySecurity() – Retrieve security information of a particular key
    - § Calls OpenKey() with Access_System_Security and Read_Control
    - § Calls RegGetKeySecurity()
- QueryValue() –
    - § Calls OpenKey(), if this succeeds the value can be read
- IsAccessAllowed() – Determine if a key allows a particular access
    - § Calls OpenKey with the desired access to test if access is allowed

- NT5.1 Source, winreg.h:
    - RegGetKeySecurity()
    - RegOpenKeyEx()
- NT5.1 Source, registry.hxx & registry.cxx
    - IsRemoteRegistry() – Returns true if _RemoteRegistry, which can be set to true by the InitializeMachineName() function in registry.cxx
- Nothing from the NT5.1 Source was particularly fruitful as everything funnels back to the RegOpenKeyEx() API for access and handling the remote registry connection. I don't currently have access to the source for RegOpenKeyEx().

Conclusions:

SharpHound currently utilizes Remote Registry to capture key registry data from Domain Controllers, AD CS Certificate Authorities, and domain-joined workstations and servers.

Remote Registry access for the purposes of collecting this data requires:

- Network access from client to server via SMB (tcp445)
- Remote Registry service to be running on both the client and the server
- Permissions to open a handle to the \PIPE\winreg named pipe with Read access rights
- Permission granted to open the registry Key with Read access
- Permissions granted on the specific registry SubKeys with Read access

There are two methods to provide remote registry access without Administrator privileges on the remote host:

1. Configure DACL on the HKLM\SYSTEM\CurrentControlSet\Control\SecurePipeServers\winreg subkey
2. Create an exception to the winreg subkey DACL using either:
    a. HKLM\SYSTEM\CurrentControlSet\Control\SecurePipeServers\winreg\AllowedExactPaths
    b. HKLM\SYSTEM\CurrentControlSet\Control\SecurePipeServers\winreg\AllowedPaths

Both method 1 and 2 can be configured manually, with scripting, or via GPO. Neither of these methods will override DACLs configured on the registry key or subkey level.

All registry subkeys that SharpHound can currently collect have DACLs in the registry which permit Authenticated Users or Builtin Users to read the subkey and its data. An unprivileged principal on host is a different scenario than any principal remotely from a security aspect.

Across data gathered in my lab from Windows Server 2012 R2, Windows Server 2019, and Windows Server 2025 there are between 61 and 69 total subkeys in the registry paths that SharpHound desires to collect. Of those, only 4 or 5 of those subkeys do not grant unprivileged principals KeyRead rights.

For remotely collecting registry data over the network via SharpHound there are two known feasible approaches:

1. Create WinReg named pipe remote connection exceptions for specific registry paths. This will allow any Authenticated User to connect to the named pipe for those specific key paths. Security descriptors on the registry keys and subkeys provide granular control. Of the 2 exception options, AllowExactPaths is more secure as it does not allow access to subkeys over the winreg connection. If configuration via GPO is desired, this would be the 'Network access: Remotely accessible registry paths' setting.

2. Modify the DACL of the HKLM\SYSTEM\CurrentControlSet\Control\SecurePipeServers\winreg subkey by adding an Allow Read ACE with the trustee principal being a tier-appropriate domain local security group which is placed in a secure OU for that tier. Add the SharpHound collection service account for that tier into the corresponding security group. Security descriptors on the registry keys and subkeys provide granular control.

There is no perfect solution here. Each is a tradeoff between granting some trust across the entire winreg named pipe to the SharpHound collector for that tier vs granting trust across explicit registry paths in the winreg named pipe to any Authenticated User.

Note: Microsoft AD CS role automatically creates a remote registry path exception in the HKLM\SYSTEM\CurrentControlSet\Control\SecurePipeServers\winreg\AllowedPaths\Machine value. If making modifications to AllowedPaths via GPO be mindful of this when scoping the policy.

Additional Remote Registry attack surface reduction considerations:

- For all but Domain Controllers and File servers, restrict inbound SMB (tcp445) traffic from all but SharpHound collector host(s) and other management or administrative hosts at the network or host's network layer.

- Consider whether Remote Registry service needs to be always running. Is it feasible for this service to be started only during times when it is necessary? Service start and stop can be orchestrated. A simple option could be utilizing Scheduled Tasks, perhaps pushed out via GPO. The event trigger for the task to start Remote Registry could be a timeframe. It could also be an event, such as a successful logon from the SharpHound collector host. This may not be feasible if Remote Registry is required for other administrative, management, or posture activities.

- Adding the SharpHound collector, via group membership as a trustee granted Allow Read access on the winreg subkey to enable registry-wide access for the collection account

can be offset with targeted Deny ACEs in the registry hierarchy. Well-known highly-sensitive registry paths do not grant allow access to non-privileged principals by default.

- Deny ACEs could also be utilized in the registry hierarchy in instances where the AllowedPaths or AllowedExactPaths winreg exemptions are used. However, it is more challenging to deny access to Authenticated Users and allow access to the SharpHound collector along with the other required principals such as SYSTEM, Administrators, Backup Operators, All Application Packages, etc. Without carefully crafted combinations of inherited deny ACEs with explicit allow ACEs this strategy is impossible. Even with perfectly crafted ACEs and a very targeted set of registry paths this method would be troublesome.

Additional data on this topic is available in the form of 47 pages of notes with screenshots and a (currently private) GitHub repository with PowerShell scriptlets and data captures.

References:

- https://learn.microsoft.com/en-us/windows/win32/sysinfo/registry-key-security-and-access-rights

- https://attack.mitre.org/mitigations/M1024/

- https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-rrp/0fa3191d-bb79-490a-81bd-54c2601b7a78