# Identifying and Resolving Group Policy Settings Problems

## Were You Authenticated by a DC?

Since access to the Active Directory objects is dependent on a domain controller for that domain, if you are unable to authenticate with a DC in the domain Group Policy will fail to apply.

An easy way to determine if a logged on user has been authenticated by an Active Directory DC is to check the **LOGONSERVER** environment variable.   
(Type **Set** at a command prompt to obtain a list of environment variables.)

If the LOGONSERVER value indicates the local computer name, rather than the name of a domain controller, the user has been logged on with a local account, or with cached domain credentials. They have not contacted a valid Domain Controller, and Group Policy will fail to apply.

Checking the value of the LOGONSERVER environment variable is not a valid test on a Windows XP client. Due to Windows XP **Fast Logon Optimization**, Windows XP will display the last DC to authenticate the user even when logged on with cached credentials.

## Can You Access Sysvol?

Recall that every GPO consists of an **active directory portion** (the GPC) and a **file system portion** (the **GPT**). Since the file system portion of each GPO is stored beneath the SYSVOL share on each domain controller, both user and computer accounts must have adequate access to the SYSVOL share to successfully process Group Policy.

Therefore, there are a number of things to check with regard to SYSVOL access if Group Policy is failing.

**First, verify that the user, computer, and System accounts all have adequate permissions to access the SYSVOL share on the authenticating DC.**

One way to check whether the computer has access is as follows:

1. Use the **PsExec** command to launch a **cmd** prompt interactively: **psexec.exe -sid cmd.exe**
2. Then, type at the command prompt that is launched: **dir \\<FQDN of the domain>\sysvol**
3. If contents are displayed, walk the contents of the failing policy to make sure that you have proper access to all relevant files.

If you are unable to access the SYSVOL share, and if you fail with **access denied**, check the both share-level and NTFS-level permissions on SYSVOL, along with the following user rights:

* Access this computer from the network
* Bypass traverse checking

If the problem does not seem to be access-related, that is if all relevant accounts have proper access to SYSVOL, then you need to make sure that the required policy files, the GPT files, are there, and that there are no **file-level permissions** on the GPTs blocking access for the USER, COMPUTER, or SYSTEM accounts. These accounts should have at least Read permission to these files.

If you fail to access the SYSVOL with non-security related errors (examples would include system errors 53, 67, 51, etc.) you could be having **network connectivity** problems. The following items should be checked.

* Can you connect to \\<FQDN of Domain>\sysvol?
* If **No** -> Can you connect to %LOGONSERVER%\sysvol?
  + If **Yes** -> Check to make sure that the DFS client isn’t disabled
    - Check to make sure that the TCP/IP NetBIOS Helper service is started
    - Check to make sure that the client computer is pointing to the proper DNS server and that the DC has registered its records with the DNS server
  + If **No** -> Can you connect to **\\<IP address of the logonserver>\sysvol**?
    - If **No** ->Check to make sure that the Sysvol is shared out on that DC
    - If **Yes** -> Check DNS.
  + If **Sometimes** -> You could be seeing **sporadic results** based on the DC that is hosting the **\\<FQDN of Domain>\sysvol** share. Try selecting a different DC to host the SYSVOL share and refresh policy.

## Check ACLs on GPO

In order for a GPO to apply to a specific user account, that user, or a group including the user, must be in the ACL for the GPO itself, with at least **Read** and **Apply Group Policy permissions**. These permissions are assigned via GPEdit, in the Properties of each GPO.

When checking ACLs for any purpose, you should also make sure that no permissions have been denied to any of the accounts in question. Remember that a **Deny** for any permission will **override** a **Grant** for the same permission. For example, if the user is granted the permission, but a group containing the user is denied the permission, the deny overrules the grant, and the user is not granted the permission he was denied.

It is also required that the account have permissions to read the gPLink attribute on the linking OU.

## Check GPO and OU Properties

There are a number of options that can be set in the Properties of each GPO and OU that should be checked in event of trouble.

It may also be necessary to check the Properties of all OUs between the location of the User or Computer account and the location of the GPO, to ensure that a value set at one level is not preventing the GPO from applying further down the tree.

First, make sure the GPO in question has not been disabled. This is set via the Properties of the GPO.

Does an OU **closer** to the user or computer object have **Block inheritance** set? If so, that would prevent a GPO linked to a higher-level container from applying to an account at a lower level of the tree. This can be overridden with the **No Override** option on the GP Link.

## What Is the Replication Status of the GPO?

If you determine that a GPO or GPOs are not applying in the expected manner, **Active Directory** and/or **SYSVOL** replication could be a factor, and should be checked.

Recall that each GPO consists of two pieces – the **GPC** and the **GPT**. The GPC is maintained in Active Directory, the GPT is maintained in SYSVOL on the local file system of each DC. In addition, as previously noted, each of these relies on a different replication mechanism to maintain **consistency** of these objects across all DCs. GPC data is replicated via Active Directory replication, GPT data is replicated via FRS / DFSR replication. Therefore, it may be necessary to check the replication status of these objects.

Therefore, it is important to make sure that the version number is synchronized across all of the DCs for both the **AD version** (GPC) and the **Sysvol version** (GPT) of policy. It is also important to make sure that the AD version and the Sysvol version of the policy is in sync. GPOTool provides this service and information for you.

* A **sysvol mismatch** error typically indicates that the Sysvol version of policy is not in sync across all of the DCs in the domain.
* A **DS mismatch** error indicates that AD replication has not yet successfully replicated the DS version across all DCs in the domain.
* A **version mismatch** error typically indicates that the DS version and Sysvol version do not match on a particular DC.

This is usually caused by an authoritative **restore** of either the GPC or GPT.

On the other hand, it may be caused by **FRS / DFSR issues** such as anti-virus or a security policy changing the security descriptors of the contents of the Sysvol.

## Enable Group Policy Debug Logging

### **Enable userenv.log:**

HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Winlogon

Entry: UserEnvDebugLevel

Type: REG\_DWORD

Value data: 10002 (Hexadecimal)

Log File location: %Systemroot%\Debug\UserMode\Userenv.log

### **Enable gpsvc.log: (for windows server 2008 or later)**

Key: HKLM\Software\Microsoft\Windows NT\CurrentVersion\Diagnostics

Value Name GPSvcDebugLevel

Value Type: REG\_DWORD

Value data: 30002 (Hexadecimal)

Log File location: %Systemroot%\Debug\UserMode\gpsvc.log

# Troubleshooting Client Side Extension Problems

Remember that the Core Group Policy engine has somewhat limited responsibility when it comes to the specifics of Group Policy for any given part of the system, and that the Client-Side Extensions (CSE) are actually responsible for application of policy in their specific areas.

With that in mind, it makes sense to include a status check of any applicable CSEs to the overall troubleshooting approach.

While details about the CSEs are provided in another section of this course, in the present context of troubleshooting overall Group Policy functions, keep these two things in mind:

* Make sure that relevant **CSEs** are not disabled.
* If Core GP functions seem to be okay, it may be necessary to drill down at CSE-level to complete troubleshooting.

## Troubleshooting the Registry Client-Side Extension

The **Registry Client-side Extension** is invoked when the **Administrative Templates nodes** are used in **GPEDIT**. The Registry CSE processes GPOs, which contain settings specified via the Administrative Templates.

Windows provides an improved mechanism for applying such registry-based policy settings, which makes it easier to determine what registry keys have been affected by Policy, and if necessary, undo or adjust the settings.

This improved mechanism relies on the use of the **REGISTRY.POL** and **NTUSER.POL** files. A REGISTRY.POL file exists as part of each GPO for which registry-based settings have been specified, i.e. via the Administrative Templates. When Policy is applied, changes are applied from the REGISTRY.POL files to the registry of the target computer, per the GPO order of precedence. As the REGISTRY.POL files are processed, the **NTUSER.POL** files are created, which reflect all of the changes that have been written to the registry as a result of the applied GPOs.

There are two **NTUSER.POL** files on each system – one containing User settings, and one containing Computer settings. The user copy is stored in the root of the User Profile. The computer copy is stored in the root of the **All Users** profile.

The core Group Policy engine invokes the Registry CSE at startup, logon, or the Policy Refresh Interval, and passes the extension the list of Group Policy Objects to be applied. As the GPOs are processed, settings are applied from the **REGISTRY.POL** files to the appropriate registry keys on the target machine, and the **NTUSER.POL** files are created. The **NTUSER.POL** files record all of the registry-based settings applied. At Policy refresh, the Registry CSE parses the existing NTUSER.POL files on the target computer to clear any previously-specified registry settings.

## Troubleshooting Scripts CSE

Windows provides a mechanism for specifying and executing scripts via Group Policy. This capability relies on the Scripts Client Side Extension.

By default, Group Policy may be configured to run the following types of scripts:

* Start Up
* Log On
* Log Off
* Shutdown

The role of the Scripts CSE in processing scripts is fairly straightforward. It simply locates the specified script based on data provided by individual GPOs, and then passes the data to a local USERINIT process, which executes the specified script. The Scripts CSE is not responsible for errors that occur within the execution of the script.

### **Script Storage - SYSVOL**

Each GPO that specifies a script to be run includes a **SCRIPTS.INI** file as part of the GPO data. SCRIPTS.INI contains the path and parameters for each script to be run. SCRIPTS.INI is a hidden file. On the SYSVOL, SCRIPTS.INI is stored in the following directories:

* <domain>\Policies\<policy>\User\scripts\Logon
* <domain>\Policies\<policy>\User\scripts\Logoff
* <domain>\Policies\<policy>\Machine\scripts\Shutdown
* <domain>\Policies\<policy>\Machine\scripts\Startup

### **Script Storage - Local GPO**

If a script is specified via a Local GPO, the SCRIPTS.INI is located in:

* **%SYSTEMROOT%\System32\GroupPolicy\User\Scripts** for logon and logoff scripts.
* **%SYSTEMROOT%\System32\GroupPolicy\Machine\Scripts f**or startup and shutdown scripts.

### **Typical Scripts CSE Process Errors**

As noted in the previous section, the two most frequent problems encountered with the Scripts CSE are:

* Bad Script Path in SCRIPTS.INI
* Hung Script

Options for gathering troubleshooting information include:

* Enabling USERENV logging, with verbose switch.
* Process Monitor

### **Hung Scripts**

The default time allocated for scripts to run is 10 minutes…but note that this is the time allocated for all scripts to run. This means that if a script hangs, fails, or stops executing for any reason, other scripts will be processed (only if scripts are running asynchronously), but after 10 minutes any unprocessed scripts will be shut down.

This timeout value can be modified via the Computer policy setting **Maximum wait time** for Group Policy Scripts.

### **Reasons Scripts Hang**

Often the startup or shutdown scripts will fail is because they run under the context of the system. Some operations do not behave the same when run under the context of the system. To test your script with system credentials launch the **command prompt** with PsExec tool.

## Troubleshooting Security CSE Issues

When troubleshooting issues related to Security Policy, it is useful to have some familiarity with the Security Client Side Extension, and with some of the specific common problems encountered by customers.

For troubleshooting, the best idea is to check the log file in the following location:

**%SYSTEMROOT%\security\logs\winlogon.log**

This is one of the most important tools in troubleshooting the Security CSE. If you are having a problem with the Security CSE one of your first steps should be enabling this debug logging and collecting the **winlogon.log** file.

### **Security CSE Process**

When the Security CSE is notified by the Core GP engine, it is provided a list of GPOs to apply. The Security CSE then copies the **gpttmpl.inf** file from the folder structure of each policy in the **Sysvol**. It copies that file locally to the hidden folder **%SYSTEMROOT%\Security\Templates\Policies**. The settings are read from the **gpttmpl.inf** in the Sysvol and written to an intermediary file named **tmpgptfl.inf**. Once the copy has completed successfully the file is copied off and is named incrementally starting from gpt00000.inf. If the GPO is linked to the domain then the cached template will be named with the **.dom** extension otherwise, it will be named with the .inf extension. This is done because some settings are only applied if they are linked to the domain. The templates are then applied from the cached location in order from least to greatest. This means that the gpt00000.inf will be applied before gpt00001.inf and that gpt00001.inf will have a higher precedence in the case of a conflict.