LocalSystem Account

The LocalSystem account is a predefined local account used by the service control manager. This account is not recognized by the security subsystem, so you cannot specify its name in a call to the **LookupAccountName** function. It has extensive privileges on the local computer, and acts as the computer on the network. Its token includes the NT AUTHORITY\SYSTEM and BUILTIN\Administrators SIDs; these accounts have access to most system objects. The name of the account in all locales is .\LocalSystem. The name, LocalSystem or *ComputerName*\LocalSystem can also be used. This account does not have a password. If you specify the LocalSystem account in a call to the **CreateService** or**ChangeServiceConfig** function, any password information you provide is ignored.

A service that runs in the context of the LocalSystem account inherits the security context of the SCM. The user SID is created from the **SECURITY_LOCAL_SYSTEM_RID** value. The account is not associated with any logged-on user account. This has several implications:

- The registry key **HKEY_CURRENT_USER** is associated with the default user, not the current user. To access another user's profile, impersonate the user, then access **HKEY_CURRENT_USER**.
- The service can open the registry key **HKEY_LOCAL_MACHINE\SECURITY**.
- The service presents the computer's credentials to remote servers.
- If the service opens a command window and runs a batch file, the user could hit CTRL+C to terminate the batch file and gain access to a command window with LocalSystem permissions.

The LocalSystem account has the following privileges:

- **SE_ASSIGNPRIMARYTOKEN_NAME** (disabled)
- **SE_AUDIT_NAME** (enabled)
- **SE_BACKUP_NAME** (disabled)
- **SE_CHANGE_NOTIFY_NAME** (enabled)
- **SE_CREATE_GLOBAL_NAME** (enabled)
- **SE_CREATE_PAGEFILE_NAME** (enabled)
- **SE_CREATE_PERMANENT_NAME** (enabled)
- **SE_CREATE_TOKEN_NAME** (disabled)
- **SE_DEBUG_NAME** (enabled)
- **SE_IMPERSONATE_NAME** (enabled)
- **SE_INC_BASE_PRIORITY_NAME** (enabled)
- **SE_INCREASE_QUOTA_NAME** (disabled)
- **SE_LOAD_DRIVER_NAME** (disabled)
- **SE_LOCK_MEMORY_NAME** (enabled)
- **SE_MANAGE_VOLUME_NAME** (disabled)
- **SE_PROF_SINGLE_PROCESS_NAME** (enabled)
- **SE_RESTORE_NAME** (disabled)

- **SE_SECURITY_NAME** (disabled)
- **SE_SHUTDOWN_NAME** (disabled)
- **SE_SYSTEM_ENVIRONMENT_NAME** (disabled)
- **SE_SYSTEMTIME_NAME** (disabled)
- **SE_TAKE_OWNERSHIP_NAME** (disabled)
- **SE_TCB_NAME** (enabled)
- **SE_UNDOCK_NAME** (disabled)

Most services do not need such a high privilege level. If your service does not need these privileges, and it is not an interactive service, consider using the **LocalService account** or the **NetworkService account**. For more information, see**Service Security and Access Rights**.

Community Additions ADD

Local System Account adding a user to local administrator account on a DC

I what circumstances does a local system account add a user to the administrator account. Any information on this would be greatly appreciated. Thanks.

Thomas TItus

7/1/2015

Oh, ehll yeah you sure can and ehullva lot more !

Query AD objects using LocalSystem

Can this account query the AD objects? From the description here, it does not seems it will, but want to confirm.

Prashant Thakwani

3/18/2011

Long, long ago, in a thread about a year and half away, it was asked whether or not the LocalSystem could query AD Domain Objects, and the answer is an EMPHATIC yes !

Even though is called the "LocalSystem" account, it still belongs in the security context of the local hosting computer so the LocalSystem account of DomainMember PC/Server can definitely query AD objects.

The local System account has its own uniquely self managing password and via registry edits or domain policies the length of time the password remains the same or even set to never change, the password strength is natively strong usually of mixed case, numbers and at least one special character but will also comply with your account policies such as a domain / local account policy password of 24 characters and requiring complexity, the locally managed system password will abide accordingly.

You don't have to know the password to logon as the LocalSystem, if you can execute a cmd prompt from an AT/Scheduler job or just use Microsoft Sysinternals psexec (download it from Microsoft if you don't already have it and you should be using it anyway http://technet.microsoft.com/en-us/sysinternals) to spawn a command prompt or any other application as the LocalSystem, e.g.

psexec -i -s -h %systemroot%\system32\cmd.exe (this works on Vista, XP, Windows 7, all the servers and Windows8)

Execute a WhoAmI.exe or echo %userprofile% to confirm your the local System, i.e.

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\windows\system32>whoami
NT AUTHORITY\SYSTEM
C:\windows\system32>echo %userprofile%
C:\Documents and Settings\LocalService
```

If using Vista or higher, kill the existing explorer process that was running in your Administrator context and launch explorer.exe from your system shell and Explorer will launch and create your profile as if you were a newly created user logging on for the first time. The System account always had a profile but it wasn't an interactive one that included an Internet Explorer profile or other user specific items such as Flash Player and AdobeAcrobat.

You should be comfortable with the concept of running your system in Admin context since LocalSystem is flying without a net, you have the ability to break things harder as the LocalSystem and interactive System sessions aren't supported, if you called anyone for tech support they might not even understand what you're describing and just stop trying to assist you.

This is about as risky as smoking an already lit cigarette while pumping gasoline or using your iPhone during take off and

landing, even though everyone says its dangerous it really isn't but people like their misconceptions.

CunfStucker
12/28/2012

List is incorrect (v2)

The list above is inaccurate. A list of the privileges I got from a service on Windows 7.

SeAssignPrimaryTokenPrivilege Disabled
SeAuditPrivilege Enabled
SeBackupPrivilege Disabled
SeChangeNotifyPrivilege Enabled
SeCreateGlobalPrivilege Enabled
SeCreatePagefilePrivilege Enabled
SeCreatePermanentPrivilege Enabled
SeCreateSymbolicLinkPrivilege Enabled
SeDebugPrivilege Enabled
SeImpersonatePrivilege Enabled
SeIncreaseBasePriorityPrivilege Enabled
SeIncreaseQuotaPrivilege Disabled
SeIncreaseWorkingSetPrivilege Enabled
SeLoadDriverPrivilege Disabled
SeLockMemoryPrivilege Enabled
SeManageVolumePrivilege Disabled
SeProfileSingleProcessPrivilege Enabled
SeRestorePrivilege Disabled
SeSecurityPrivilege Disabled
SeShutdownPrivilege Disabled
SeSystemEnvironmentPrivilege Disabled
SeSystemProfilePrivilege Enabled
SeSystemtimePrivilege Disabled
SeTakeOwnershipPrivilege Disabled
SeTcbPrivilege Enabled
SeTimeZonePrivilege Enabled
SeUndockPrivilege Disabled

Please note for instance that there is the SeCreateSymbolicLink privilege in that list. From the LocalSystem account on my system, I usually get 4 additional privileges but there is also no SeCreateTokenPrivilege. Oh, and please note, this isn't an attempt at passing off my list as a "definitive list" it is just to point out that the list given in this topic is missing some privileges that this account normally has when the token hasn't had privileges stripped on Windows 7.

Re: the comment by GeekWench.

The comment by GeekWench was originally aimed at this comment, but since I wanted to edit this to make it less like "this is a definitive list" and more like "this topic isn't quite right, see what I get" it is now above.

Anyway, about the registry key that you pointed out. Yes it is true that if the RequiredPrivileges key is set then the SCM removes

privileges from the user token, this is assuming that I wasn't in control of the service. The service being used was written by me and completely under my control. In doing this, I guaranteed that the RequiredPrivileges registry value wasn't set. When this isn't set, it is documented to give all privileges available to that user. See **http://msdn.microsoft.com/en-us/library/windows/desktop/ms685976(v=vs.85).aspx** for this.

crescens2k
5/11/2012

None

None

crescens2k
5/10/2012

None

None

crescens2k
5/10/2012

Actually, you can't just grab a service and enumerate its privileges

While the first responder below may have enumerated the privileges assigned to a service running as LocalSystem on his/her Windows 7 machine, that doesn't mean that those are the default privileges. Using the RequiredPrivileges registry value, you can remove privileges granted to a specific service account running as Local System, Network Service or Local Service. You cannot add privileges to an account using the RequiredPrivileges registry value, however; you can only remove privileges provided by the default configurations.

Geekwench
4/30/2012

Query AD objects using LocalSystem

Can this account query the AD objects? From the description here, it does not seems it will, but want to confirm.

Prashant Thakwani
3/18/2011

LocalSystem privileges

This page contradicts the privileges listed on the other page (SeExports > SeLocalSystemSid):
http://msdn.microsoft.com/en-us/library/aa489494.aspx
