**Windows Privileges Overview**

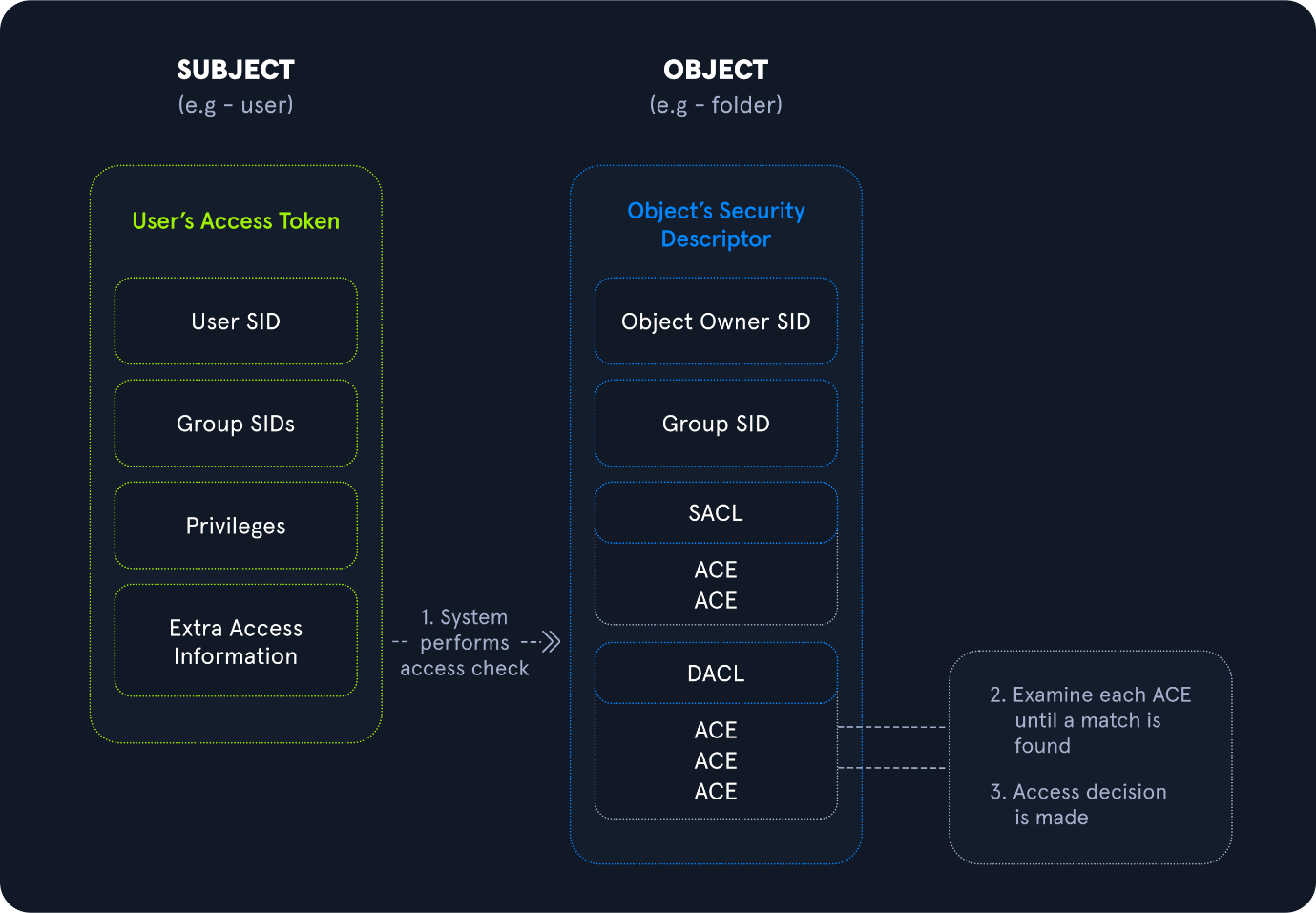
[Privileges](https://docs.microsoft.com/en-us/windows/win32/secauthz/privileges) in Windows are rights that an account can be granted to perform a variety of operations on the local system such as managing services, loading drivers, shutting down the system, debugging an application, and more. Privileges are different from access rights, which a system uses to grant or deny access to securable objects. User and group privileges are stored in a database and granted via an access token when a user logs on to a system. An account can have local privileges on a specific computer and different privileges on different systems if the account belongs to an Active Directory domain. Each time a user attempts to perform a privileged action, the system reviews the user's access token to see if the account has the required privileges, and if so, checks to see if they are enabled. Most privileges are disabled by default. Some can be enabled by opening an administrative cmd.exe or PowerShell console, while others can be enabled manually.

The goal of an assessment is often to gain administrative access to a system or multiple systems. Suppose we can log in to a system as a user with a specific set of privileges. In that case, we may be able to leverage this built-in functionality to escalate privileges directly or use the target account's assigned privileges to further our access in pursuit of our ultimate goal.

**Windows Authorization Process**

Security principals are anything that can be authenticated by the Windows operating system, including user and computer accounts, processes that run in the security context or another user/computer account, or the security groups that these accounts belong to. Security principals are the primary way of controlling access to resources on Windows hosts. Every single security principal is identified by a unique [Security Identifier (SID)](https://docs.microsoft.com/en-us/troubleshoot/windows-server/identity/security-identifiers-in-windows). When a security principal is created, it is assigned a SID which remains assigned to that principal for its lifetime.

The below diagram walks through the Windows authorization and access control process at a high level, showing, for example, the process started when a user attempts to access a securable object such as a folder on a file share. During this process, the user's access token (including their user SID, SIDs for any groups they are members of, privilege list, and other access information) is compared against [Access Control Entries (ACEs)](https://docs.microsoft.com/en-us/windows/win32/secauthz/access-control-entries) within the object's [security descriptor](https://docs.microsoft.com/en-us/windows/win32/secauthz/security-descriptors) (which contains security information about a securable object such as access rights (discussed below) granted to users or groups). Once this comparison is complete, a decision is made to either grant or deny access. This entire process happens almost instantaneously whenever a user tries to access a resource on a Windows host. As part of our enumeration and privilege escalation activities, we attempt to use and abuse access rights and leverage or insert ourselves into this authorization process to further our access towards our goal.



[Image source](https://docs.microsoft.com/en-us/windows/security/identity-protection/access-control/security-principals)

**Rights and Privileges in Windows**

Windows contains many groups that grant their members powerful rights and privileges. Many of these can be abused to escalate privileges on both a standalone Windows host and within an Active Directory domain environment. Ultimately, these may be used to gain Domain Admin, local administrator, or SYSTEM privileges on a Windows workstation, server, or Domain Controller (DC). Some of these groups are listed below.

| **Group** | **Description** |
| --- | --- |
| Default Administrators | Domain Admins and Enterprise Admins are "super" groups. |
| Server Operators | Members can modify services, access SMB shares, and backup files. |
| Backup Operators | Members are allowed to log onto DCs locally and should be considered Domain Admins. They can make shadow copies of the SAM/NTDS database, read the registry remotely, and access the file system on the DC via SMB. This group is sometimes added to the local Backup Operators group on non-DCs. |
| Print Operators | Members can log on to DCs locally and "trick" Windows into loading a malicious driver. |
| Hyper-V Administrators | If there are virtual DCs, any virtualization admins, such as members of Hyper-V Administrators, should be considered Domain Admins. |
| Account Operators | Members can modify non-protected accounts and groups in the domain. |
| Remote Desktop Users | Members are not given any useful permissions by default but are often granted additional rights such as Allow Login Through Remote Desktop Services and can move laterally using the RDP protocol. |
| Remote Management Users | Members can log on to DCs with PSRemoting (This group is sometimes added to the local remote management group on non-DCs). |
| Group Policy Creator Owners | Members can create new GPOs but would need to be delegated additional permissions to link GPOs to a container such as a domain or OU. |
| Schema Admins | Members can modify the Active Directory schema structure and backdoor any to-be-created Group/GPO by adding a compromised account to the default object ACL. |
| DNS Admins | Members can load a DLL on a DC, but do not have the necessary permissions to restart the DNS server. They can load a malicious DLL and wait for a reboot as a persistence mechanism. Loading a DLL will often result in the service crashing. A more reliable way to exploit this group is to [create a WPAD record](https://cube0x0.github.io/Pocing-Beyond-DA/). |

**User Rights Assignment**

Depending on group membership, and other factors such as privileges assigned via domain and local Group Policy, users can have various rights assigned to their account. This Microsoft article on [User Rights Assignment](https://docs.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/user-rights-assignment) provides a detailed explanation of each of the user rights that can be set in Windows as well as security considerations applicable to each right. Below are some of the key user rights assignments, which are settings applied to the localhost. These rights allow users to perform tasks on the system such as logon locally or remotely, access the host from the network, shut down the server, etc.

| **Setting** [**Constant**](https://docs.microsoft.com/en-us/windows/win32/secauthz/privilege-constants) | **Setting Name** | **Standard Assignment** | **Description** |
| --- | --- | --- | --- |
| SeNetworkLogonRight | [Access this computer from the network](https://docs.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/access-this-computer-from-the-network) | Administrators, Authenticated Users | Determines which users can connect to the device from the network. This is required by network protocols such as SMB, NetBIOS, CIFS, and COM+. |
| SeRemoteInteractiveLogonRight | [Allow log on through Remote Desktop Services](https://docs.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/allow-log-on-through-remote-desktop-services) | Administrators, Remote Desktop Users | This policy setting determines which users or groups can access the login screen of a remote device through a Remote Desktop Services connection. A user can establish a Remote Desktop Services connection to a particular server but not be able to log on to the console of that same server. |
| SeBackupPrivilege | [Back up files and directories](https://docs.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/back-up-files-and-directories) | Administrators | This user right determines which users can bypass file and directory, registry, and other persistent object permissions for the purposes of backing up the system. |
| SeSecurityPrivilege | [Manage auditing and security log](https://docs.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/manage-auditing-and-security-log) | Administrators | This policy setting determines which users can specify object access audit options for individual resources such as files, Active Directory objects, and registry keys. These objects specify their system access control lists (SACL). A user assigned this user right can also view and clear the Security log in Event Viewer. |
| SeTakeOwnershipPrivilege | [Take ownership of files or other objects](https://docs.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/take-ownership-of-files-or-other-objects) | Administrators | This policy setting determines which users can take ownership of any securable object in the device, including Active Directory objects, NTFS files and folders, printers, registry keys, services, processes, and threads. |
| SeDebugPrivilege | [Debug programs](https://docs.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/debug-programs) | Administrators | This policy setting determines which users can attach to or open any process, even a process they do not own. Developers who are debugging their applications do not need this user right. Developers who are debugging new system components need this user right. This user right provides access to sensitive and critical operating system components. |
| SeImpersonatePrivilege | [Impersonate a client after authentication](https://docs.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/impersonate-a-client-after-authentication) | Administrators, Local Service, Network Service, Service | This policy setting determines which programs are allowed to impersonate a user or another specified account and act on behalf of the user. |
| SeLoadDriverPrivilege | [Load and unload device drivers](https://docs.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/load-and-unload-device-drivers) | Administrators | This policy setting determines which users can dynamically load and unload device drivers. This user right is not required if a signed driver for the new hardware already exists in the driver.cab file on the device. Device drivers run as highly privileged code. |
| SeRestorePrivilege | [Restore files and directories](https://docs.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/restore-files-and-directories) | Administrators | This security setting determines which users can bypass file, directory, registry, and other persistent object permissions when they restore backed up files and directories. It determines which users can set valid security principals as the owner of an object. |

Further information can be found [here](https://4sysops.com/archives/user-rights-assignment-in-windows-server-2016/).

Typing the command whoami /priv will give you a listing of all user rights assigned to your current user. Some rights are only available to administrative users and can only be listed/leveraged when running an elevated cmd or PowerShell session. These concepts of elevated rights and [User Account Control (UAC)](https://docs.microsoft.com/en-us/windows/security/identity-protection/user-account-control/how-user-account-control-works) are security features introduced with Windows Vista to default to restricting applications from running with full permissions unless necessary. If we compare and contrast the rights available to us as an admin in a non-elevated console vs. an elevated console, we will see that they differ drastically.

Below are the rights available to a local administrator account on a Windows system.

**Local Admin User Rights - Elevated**

If we run an elevated command window, we can see the complete listing of rights available to us:

Local Admin User Rights - Elevated

PS C:\htb> whoami

winlpe-srv01\administrator

PS C:\htb> whoami /priv

PRIVILEGES INFORMATION

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

Privilege Name Description State

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

SeIncreaseQuotaPrivilege Adjust memory quotas for a process Disabled

SeSecurityPrivilege Manage auditing and security log Disabled

SeTakeOwnershipPrivilege Take ownership of files or other objects Disabled

SeLoadDriverPrivilege Load and unload device drivers Disabled

SeSystemProfilePrivilege Profile system performance Disabled

SeSystemtimePrivilege Change the system time Disabled

SeProfileSingleProcessPrivilege Profile single process Disabled

SeIncreaseBasePriorityPrivilege Increase scheduling priority Disabled

SeCreatePagefilePrivilege Create a pagefile Disabled

SeBackupPrivilege Back up files and directories Disabled

SeRestorePrivilege Restore files and directories Disabled

SeShutdownPrivilege Shut down the system Disabled

SeDebugPrivilege Debug programs Disabled

SeSystemEnvironmentPrivilege Modify firmware environment values Disabled

SeChangeNotifyPrivilege Bypass traverse checking Enabled

SeRemoteShutdownPrivilege Force shutdown from a remote system Disabled

SeUndockPrivilege Remove computer from docking station Disabled

SeManageVolumePrivilege Perform volume maintenance tasks Disabled

SeImpersonatePrivilege Impersonate a client after authentication Enabled

SeCreateGlobalPrivilege Create global objects Enabled

SeIncreaseWorkingSetPrivilege Increase a process working set Disabled

SeTimeZonePrivilege Change the time zone Disabled

SeCreateSymbolicLinkPrivilege Create symbolic links Disabled

SeDelegateSessionUserImpersonatePrivilege Obtain an impersonation token for another user in the same session Disabled

When a privilege is listed for our account in the Disabled state, it means that our account has the specific privilege assigned. Still, it cannot be used in an access token to perform the associated actions until it is enabled. Windows does not provide a built-in command or PowerShell cmdlet to enable privileges, so we need some scripting to help us out. We will see ways to abuse various privileges throughout this module and various ways to enable specific privileges within our current process. One example is this PowerShell [script](https://www.powershellgallery.com/packages/PoshPrivilege/0.3.0.0/Content/Scripts%5CEnable-Privilege.ps1) which can be used to enable certain privileges, or this [script](https://www.leeholmes.com/adjusting-token-privileges-in-powershell/) which can be used to adjust token privileges.

A standard user, in contrast, has drastically fewer rights.

**Standard User Rights**

Standard User Rights

PS C:\htb> whoami

winlpe-srv01\htb-student

PS C:\htb> whoami /priv

PRIVILEGES INFORMATION

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

Privilege Name Description State

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

SeChangeNotifyPrivilege Bypass traverse checking Enabled

SeIncreaseWorkingSetPrivilege Increase a process working set Disabled

User rights increase based on the groups they are placed in or their assigned privileges. Below is an example of the rights granted to users in the Backup Operators group. Users in this group do have other rights that UAC currently restricts. Still, we can see from this command that they have the [SeShutdownPrivilege](https://docs.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/shut-down-the-system), which means that they can shut down a domain controller that could cause a massive service interruption should they log onto a domain controller locally (not via RDP or WinRM).

**Backup Operators Rights**

Backup Operators Rights

PS C:\htb> whoami /priv

PRIVILEGES INFORMATION

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

Privilege Name Description State

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

SeShutdownPrivilege Shut down the system Disabled

SeChangeNotifyPrivilege Bypass traverse checking Enabled

SeIncreaseWorkingSetPrivilege Increase a process working set Disabled

**Detection**

This [post](https://blog.palantir.com/windows-privilege-abuse-auditing-detection-and-defense-3078a403d74e) is worth a read for more information on Windows privileges as well as detecting and preventing abuse, specifically by logging event [4672: Special privileges assigned to new logon](https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/event-4672) which will generate an event if certain sensitive privileges are assigned to a new logon session. This can be fine-tuned in many ways, such as by monitoring privileges that should *never* be assigned or those that should only ever be assigned to specific accounts.

**Moving On**

As attackers and defenders, we need to review the membership of these groups. It's not uncommon to find seemingly low privileged users added to one or more of these groups, which can be used to compromise a single host or further access within an Active Directory environment. We will discuss the implications of some of the most common rights and walk through exercises on how to escalate privileges if we obtain access to a user with some of these rights assigned to their account.