10 Methods for Identifying and Protecting Privileged AD Users

blog.netwrix.com/2023/10/03/privileged-user

Joe Dibley

Compromising privileged accounts is the penultimate objective of most cyberattacks — once attackers gain privileged access, they can then accomplish their final goal, whether that's to steal or encrypt information assets or disrupt business operations. Typically, cybercriminals gain a foothold in a network by compromising of a low-level account on a local machine. Using a variety of techniques, they then try to seize control of the more privileged accounts they need to access the organization's sensitive data and systems. In Active Directory (AD) environments, the real prize is a Domain Administrator account, but any privileged accounts will often suffice.

Key Elements of Privilege Management

An attacker with privileged access can pretty much move freely through your AD domain and access your most valuable IT resources. Therefore, to maintain security and compliance, it is imperative to establish strong identity governance and access management over all privileged accounts in your AD environment.

Specifically, ensuring the security of your systems and data involves two key components:

- Identifying which users have what privileges for which IT assets
- Detecting when any account obtains privileged access for the first time

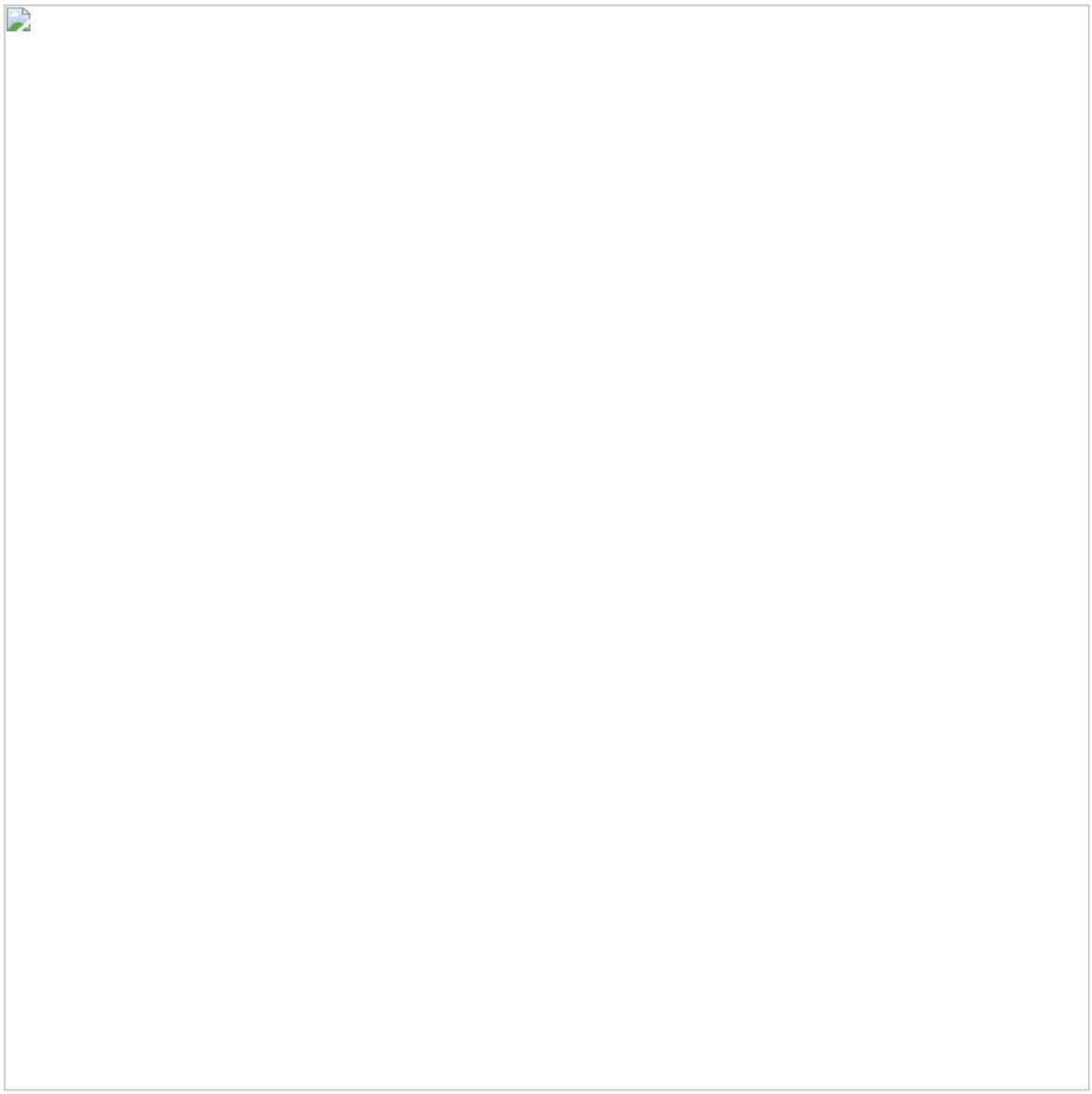
It's vital to understand that privileged access management is not a one-time endeavor. As users are onboarded and shift roles and the IT environment evolves, group entitlements and memberships change, and security teams need to spot any changes that could be malicious or otherwise improper.

Below are the top 10 methods you can use to protect your privileged user accounts from compromise. They are part of a comprehensive security strategy to prevent unauthorized users from obtaining privileged access to AD.

1.Closely manage built-in privileged groups like Domain Admins.

The built-in AD privileged groups like Administrators, Domain Admins, Enterprise Admins and Schema Admins are an obvious place to start. Next is the Local Administrators group on each of your Windows endpoints, which provide privileged access to those systems.

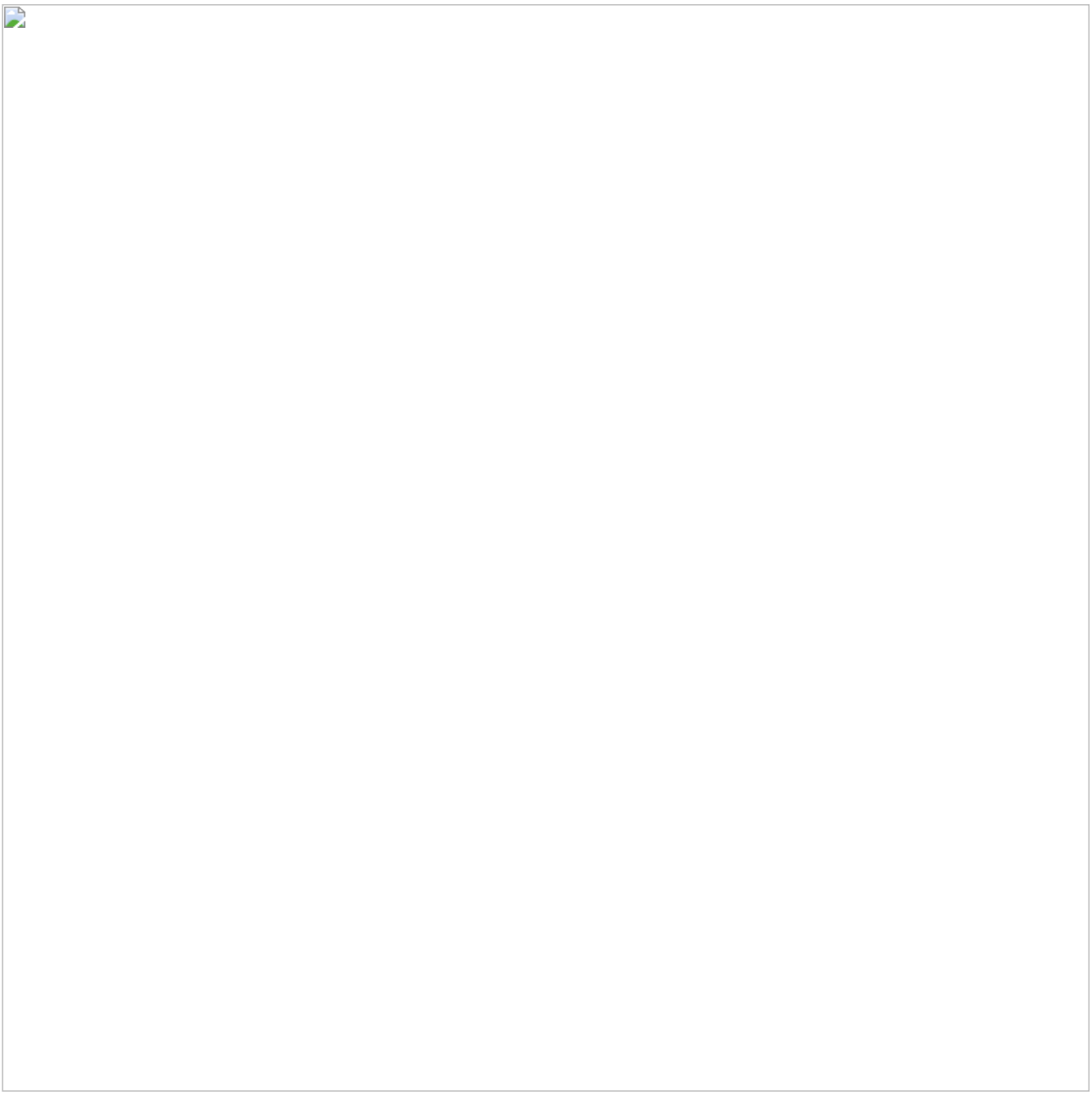
Identifying the members of each of these groups is as easy as viewing the Membership Property tab of the group in Active Directory Users & Computers (ADUC). Another way is to use the **Get-ADGroupMember** PowerShell cmdlet, as shown in the screenshot below. This is the preferred method when you need to analyze multiple groups in a quick fashion.



Keep in mind that periodic reviews of group members provide insight into the state of the group at that moment only. Still, it is a good place to begin.

2. Check for groups nested inside privileged groups.

It's vital to check whether privileged groups have any groups as members, since these nested groups grant their members all the access rights of the privileged group. For example, in the screenshot below, you can see that one of the members of the Domain Admins group is a group called LabPowerUsers — which means any member of the LabPowerUsers group has administrative privileges across the domain.



Whenever possible, you should avoid group nesting, especially inside privileged groups, because it provides attackers with a way to gain access to critical resources and data with less risk of triggering alarms — they add an account they control to a nested group, rather than directly to the parent privileged group, which may be more closely monitored.

As shown earlier, the **Get-ADGroupMember** cmdlet reveals only the immediate members of the queried group, so it will list any nested groups but not their members. One way to also show the members of any nested groups is to use the LDAP filter shown in red below:

```
param([string]$groupDn )
$s = new-object system.directoryservices.directorysearcher
$s.searchroot = new-object system.directoryservices.directoryentry
$s.filter = “(&(memberOf:1.2.840.113556.1.4.1941:=$groupDn))”
$s.propertiestoload.add(“name”)
$s.propertiestoload.add(“objectclass”)
$r = $s.FindAll()
foreach ($e in $r)
{Write-Host
$e.Properties.objectclass[$e.Properties.objectclass.Count-1]:
$e.properties.name
```

If this code is saved as a script (.GetNestedMembers.ps1), it can be executed as follows, specifying the Distinguished Name of the desired group:

```
.GetNestedMembers.ps1 -groupDn “CN=Administrators,CN=Builtin,DC=lab,DC=local”
```

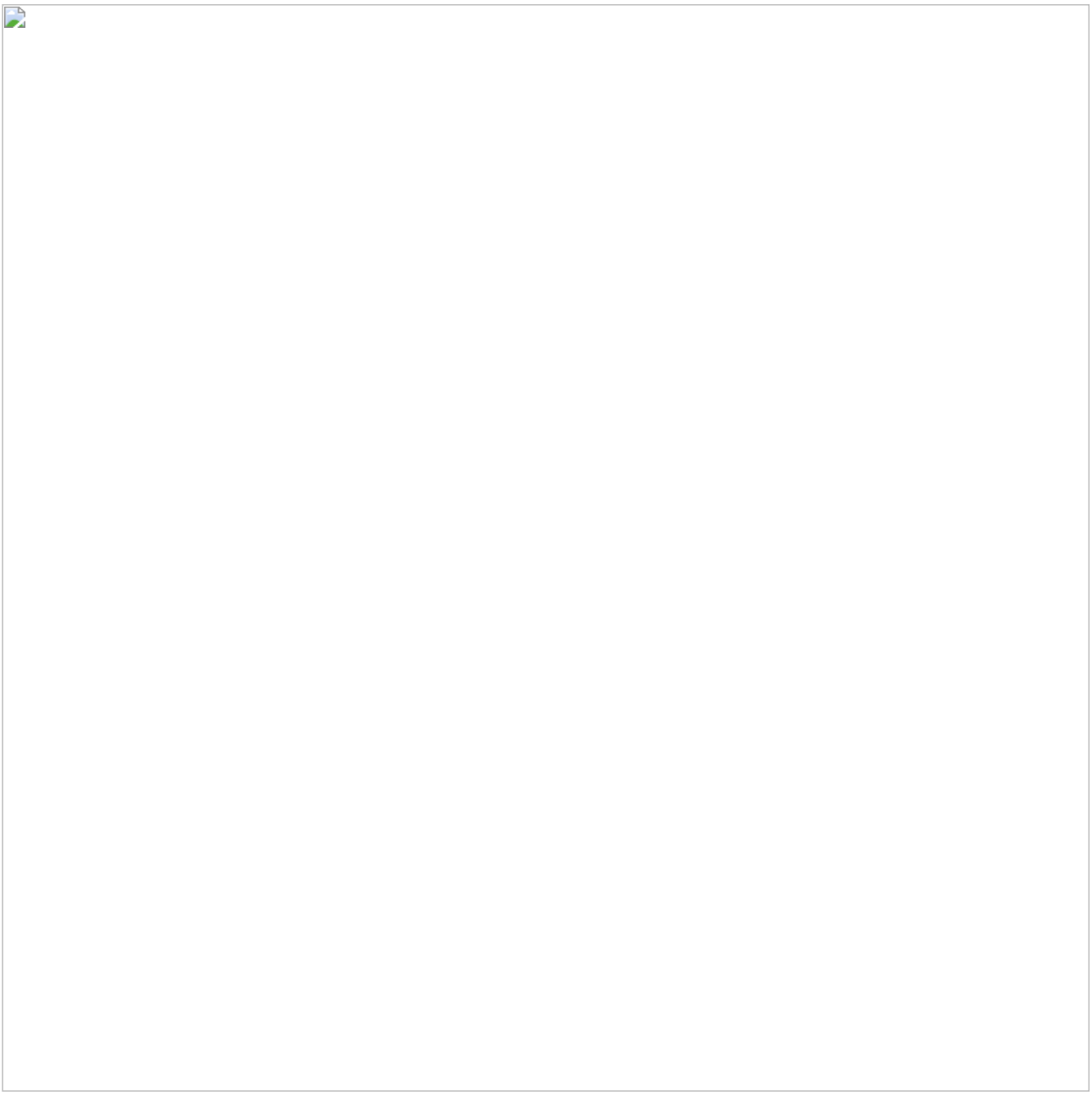
The output shows the group’s name and lists its members:

```
group : Domain Admins
user : bosshogg
user : azuresync
user : Barry Vista
user : Randy Smith
user : AdminService
```

To gain fast visibility into elevated privilege without all the hassle and expense of scripting, many organizations choose to invest in a third-party privileged access management (PAM) tool such as the Netwrix PAM solution.

3. Pay attention to organizational unit (OU) permissions.

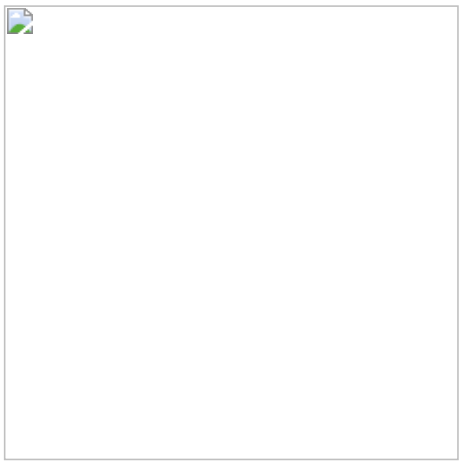
In addition to gaining privileges through group membership, users can be delegated control at the organizational unit level. The screenshot below shows a group being delegated privileges to the Sales OU; every member of that group will be able to accomplish the selected tasks, including modifying the membership of any group in the Sales OU.



Keeping track of OU permissions is complicated by inheritance. As in NTFS security, the permissions of parent OUs propagate down to child OUs and leaf objects. This can result in a lot of users have elevated privileges that are hard to discover, as illustrated in the following screenshot that shows all the child OUs and leaf objects of the Sales OU:

While you can break inheritance from a parent OU, doing so can cause problems, such as managers being unable to manage objects they are responsible for.

To generate audit events whenever a directory object is accessed , you can use AD Group Policy. Go to **Computer Configuration** > **Windows Settings** > **Security Settings** > **Advanced Audit Policy Configuration** > **Audit Policies** > **DS Access** and configure **Audit Directory Service Changes** as shown below.



Keep in mind that turning on this auditing will generate a huge volume of log entries, so manually reviewing the audit logs takes a great deal of time. Moreover, the log entries will not spell out which permissions are being assigned; they simply reveal that the permissions for a particular OU need to be reviewed.

4. Look for admin-equivalent rights on domain controllers (DCs).

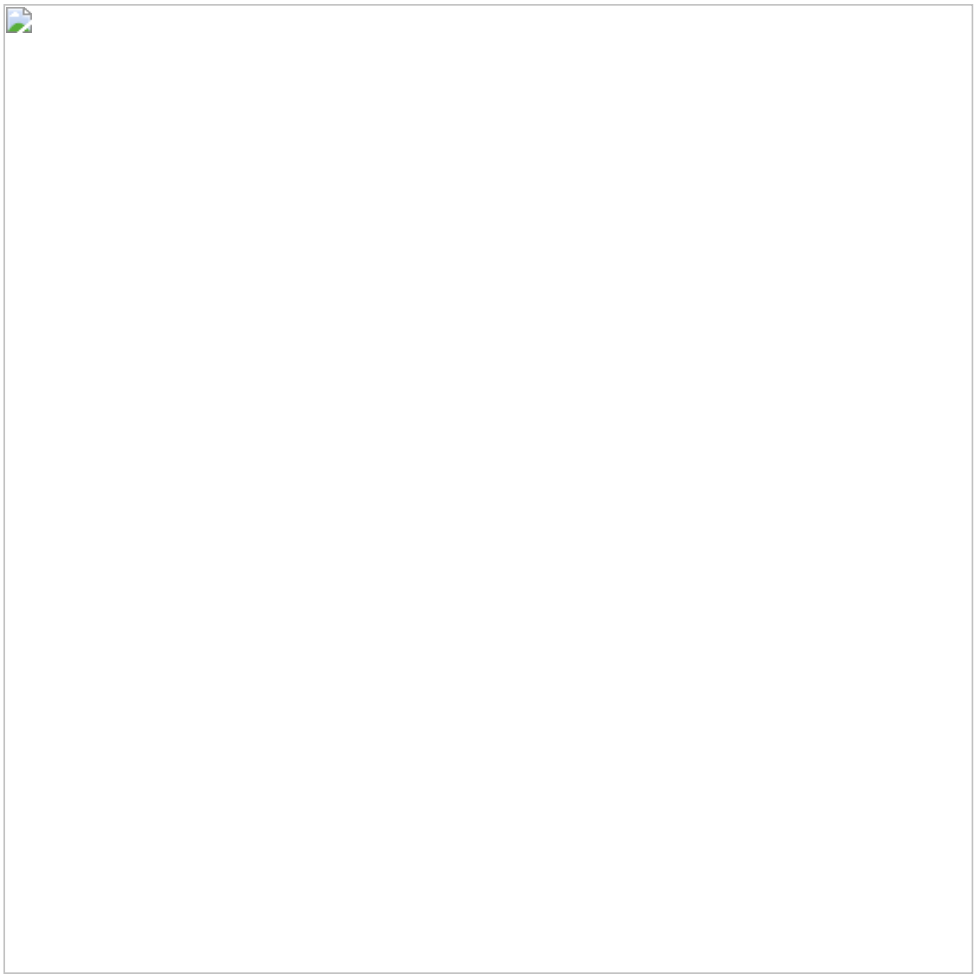
Domain controllers are special machines that run the Windows Server operating system and provide vital authentication and authorization services. Therefore, it's important to look for accounts with admin-level privileges on your DCs. For example, a service account might be granted the right to back up or restore files on DCs. Attacks can compromise these accounts by using open-source tools like Mimikatz to discover their password hashes and then cracking their plaintext passwords.

You can identify which accounts have these rights based on:

- Reviewing the Default Domain Controllers policy
- Using a PowerShell script
- Using PAM tools such as Netwrix PAM solutions

5. Uncover users with password reset authority over other accounts.

The ability to reset the password of a user account is highly sought after by threat actors, because resetting the password of an account enables them to gain control of the account. This right is included in the Full Control permission. You can also delegate the specific permission to reset passwords for other users using the Delegation of Control wizard, as shown below:



In the example below, members of the PC Support group have permission to reset the passwords of domain administrators. Therefore, by compromising the account of any member of the PC support group, an attacker can easily gain elevated domain-wide privileges.

Manually scouring every OU and user object to see who has this privilege is time consuming and error prone, and trying to manually audit all password reset activity is similarly impractical. Netwrix AD security products make it easy to see which accounts have password reset rights. The screenshot below shows an environment in which 1,000 users in the prod.net domain can have their password reset by 15 different individuals:

6. Monitor use of privileged service accounts.

An earlier tip mentioned the need to watch for service accounts that has access rights on a DC that are equivalent to those of administrators. But the risks associated with service accounts are not limited to DCs. Workloads like Exchange, SQL Server, backup solutions and other business applications run under service accounts that often have administrative privileges, and anyone who knows the credentials of such an account can use it for malicious purposes.

Service account passwords are often not changed on a regular basis, or even ever, so it's often unclear exactly who knows them. Even if service account passwords are regularly rotated, the logon activity of these accounts should be audited to ensure they are used only for their assigned services and not for illicit activity, such as interactive logon to a domain controller.

Using the Security log on a DC to determine whether a service account is being used improperly requires assembling multiple pieces of a puzzle. For example, Event 4768 shows that a Kerberos authentication ticket was requested and 4672 indicates special privileges were assigned to new logon, but they need to be correlated with Event 4624 (an account was successfully logged on).

7. Uncover users with write access to GPOs that are applied to DCs or servers running applications with domain privileged access.

Group Policy is a powerful tool for managing both users and machines like DCs, application servers and Windows endpoints. That is why attackers try to gain control of it and why it is imperative for you to monitor who has writable access to any Group Policy object (GPO).

A good starting point is to review the permissions on all GPOs that are linked to either the Domain root or the Domain Controllers OU. This is easily done by reviewing the Delegation tab of the Default Domain Controllers policy, as shown below:

8. Identify all user accounts with access to any AD management solution.

Many organizations use third-party solutions to simplify AD management tasks. These solutions can sometimes enable two methods of attack for threat actors looking to gain privileged access.

- The first involves the use of a service or proxy account that is granted privileged access to all or a subset of AD to enable the solution's management functions.

- The second method uses any account with permissions assigned *within* the management solution. Here, a low-level user might be granted the ability to perform privileged tasks (e.g., reset another user's password) over a subset of AD accounts. Depending on the level of delegation, gaining control over an account like this one can be just as good as being a Domain Admin.

To gain visibility into these risks, you must inventory of all management applications in use. Note that they often do not reside on a domain controller. Then you need to identify all service and proxy accounts that have privileged access, and monitor what these accounts are doing. Most management solutions provide some means of establishing an audit trail to monitor for inappropriate behavior. Any alerts generated by these applications can also be piped to a SIEM solution if one is available.

Alternatively, Netwrix Auditor can send daily activity summaries detailing all changes to hardware and software, scheduled tasks, applications, network settings and more on your Windows servers. Below is a screenshot showing all installed applications:

9. Don't forget virtualization infrastructure admins.

To secure your virtual AD environment, you must know which accounts have privileged access to your virtual infrastructure. Remember that anyone who manages the virtual environment that hosts DCs or member servers has the equivalent of administrative access to a physical machine.

For example, if you're running Hyper-V, members of the Hyper-V Administrators local group have admin-level access on the guest operating system. Similarly, in VMware environments, the root user on an ESXi system has the same level of access. And while member servers may not host AD, any management applications or service accounts with AD privileges present the same risk.

10. Prevent admins from leaving credential artifacts.

When a user logs on to a system, pieces of credential information often remain after the user logs off. This information can include cleartext passwords, password hashes, NTLM hashes and Kerberos tickets. Using tools such as Mimikatz, attackers can readily exploit these artifacts left behind in the memory of servers and workstations to authenticate themselves as privilege accounts. For example, cleartext passwords can simply be reused, and hashes can be passed as part of an authentication request, using additional hacking tools to gain access to other systems.

Unfortunately, identifying which machines have privileged credential artifacts on them is nearly impossible. Therefore, your focus needs to be on determining where privileged users are logging on.

As previously mentioned, you can monitor event ID 4627 (Special privileges assigned to new logon) on any current Windows system for groups that are known to be privileged. And you can monitor event ID 4769 (A Kerberos service ticket was requested) on your DCs to identify privileged logons.

After you have an idea of how privileged accounts are being used, ensure that WDigest settings prevent cleartext passwords from being stored in memory. You should also disallow privileged accounts from being used on user workstations to keep their credentials secure, and use an account with Local Admin privileges to address end-user issues.

Keeping a Tight Grip on Privileged Accounts

Privileged user accounts — from accounts with local Admin rights to one machine to those with full Domain Admin privileges — are regularly targeted by adversaries. To prevent breaches and downtime, you need to know who your privileged users are and monitor for changes that could affect privileged access. However, if you rely on native tools, assessment and monitoring are time-consuming and error-prone processes. For the reliable 24/7 monitoring you need to protect your privileged accounts from compromise and misuse, you need automated third-party tools. The Netwrix suite of solutions gives you the visibility you require to detect and thwart privilege escalation attempts and privilege abuse.

Joe Dibley
Security Researcher at Netwrix and member of the Netwrix Security Research Team. Joe is an expert in Active Directory, Windows, and a wide variety of enterprise software platforms and technologies, Joe researches new security risks, complex attack techniques, and associated mitigations and detections.