Implementing Least-Privilege Administrative Models

By robinharwood

Archived: 2026-04-06 02:05:40 UTC

The following excerpt is from The Administrator Accounts Security Planning Guide, first published on April 1, 1999:

> "Most security-related training courses and documentation discuss the implementation of a principle of least privilege, yet organizations rarely follow it. The principle is simple, and the impact of applying it correctly greatly increases your security and reduces your risk. The principle states that all users should log on with a user account that has the absolute minimum permissions necessary to complete the current task and nothing more. Doing so provides protection against malicious code, among other attacks. This principle applies to computers and the users of those computers. "One reason this principle works so well is that it forces you to do some internal research. For example, you must determine the access privileges that a computer or user really needs, and then implement them. For many organizations, this task might initially seem like a great deal of work; however, it is an essential step to successfully secure your network environment. "You should grant all domain administrator users their domain privileges under the concept of least privilege. For example, if an administrator logs on with a privileged account and inadvertently runs a virus program, the virus has administrative access to the local computer and to the entire domain. If the administrator had instead logged on with a nonprivileged (nonadministrative) account, the virus's scope of damage would only be the local computer because it runs as a local computer user. "In another example, accounts to which you grant domain-level administrator rights must not have elevated rights in another forest, even if there is a trust relationship between the forests. This tactic helps prevent widespread damage if an attacker manages to compromise one managed forest. Organizations should regularly audit their network to protect against unauthorized escalation of privilege."

The following excerpt is from the Microsoft Windows Security Resource Kit, first published in 2005:

> "Always think of security in terms of granting the least amount of privileges required to carry out the task. If an application that has too many privileges should be compromised, the attacker might be able to expand the attack beyond what it would if the application had been under the least amount of privileges possible. For example, examine the consequences of a network administrator unwittingly opening an email attachment that launches a virus. If the administrator is logged on using the domain Administrator account, the virus will have Administrator privileges on all computers in the domain and thus unrestricted access to nearly all data on the network. If the administrator is logged on using a local Administrator account, the virus will have Administrator privileges on the local computer and thus would be able to access any data on the computer and install malicious software such as key-stroke logging software on the computer. If the administrator is logged on using a normal user account, the virus will have access only to the administrator's data and will not be able to install malicious software.

> By using the least privileges necessary to read email, in this example, the potential scope of the compromise is greatly reduced."

The principles described in the preceding excerpts have not changed, but in assessing Active Directory installations, we invariably find excessive numbers of accounts that have been granted rights and permissions far beyond those required to perform day-to-day work. The size of the environment affects the raw numbers of overly privileged accounts, but not the proportion-midsized directories may have dozens of accounts in the most highly privileged groups, while large installations may have hundreds or even thousands. With few exceptions, regardless of the sophistication of an attacker's skills and arsenal, attackers typically follow the path of least resistance. They increase the complexity of their tooling and approach only if and when simpler mechanisms fail or are thwarted by defenders.

Unfortunately, the path of least resistance in many environments has proven to be the overuse of accounts with broad and deep privilege. Broad privileges are rights and permissions that allow an account to perform specific activities across a large cross-section of the environment- for example, Help Desk staff may be granted permissions that allow them to reset the passwords on many user accounts.

Deep privileges are powerful privileges that are applied to a narrow segment of the population, such as giving an engineer Administrator rights on a server so that they can perform repairs. Neither broad privilege nor deep privilege is necessarily dangerous, but when many accounts in the domain are permanently granted broad and deep privilege, if only one of the accounts is compromised, it can quickly be used to reconfigure the environment to the attacker's purposes or even to destroy large segments of the infrastructure.

Pass-the-hash attacks, which are a type of credential theft attack, are ubiquitous because the tooling to perform them is freely available and easy-to-use, and because many environments are vulnerable to the attacks. Pass-the-hash attacks, however, are not the real problem. The crux of the problem is twofold:

1. It is usually easy for an attacker to obtain deep privilege on a single computer and then propagate that privilege broadly to other computers.
2. There are usually too many permanent accounts with high levels of privilege across the computing landscape.

Even if pass-the-hash attacks are eliminated, attackers would simply use different tactics, not a different strategy. Rather than planting malware that contains credential theft tooling, they might plant malware that logs keystrokes, or leverage any number of other approaches to capture credentials that are powerful across the environment. Regardless of the tactics, the targets remain the same: accounts with broad and deep privilege.

Granting of excessive privilege isn't only found in Active Directory in compromised environments. When an organization has developed the habit of granting more privilege than is required, it is typically found throughout the infrastructure as discussed in the following sections.

In Active Directory, it is common to find that the EA, DA and BA groups contain excessive numbers of accounts. Most commonly, an organization's EA group contains the fewest members, DA groups usually contain a multiplier of the number of users in the EA group, and Administrators groups usually contain more members than the populations of the other groups combined. This is often due to a belief that Administrators are somehow "less

privileged" than DAs or EAs. While the rights and permissions granted to each of these groups differ, they should be effectively considered equally powerful groups because a member of one can make himself or herself a member of the other two.

When we retrieve the membership of local Administrators groups on member servers in many environments, we find membership ranging from a handful of local and domain accounts, to dozens of nested groups that, when expanded, reveal hundreds, even thousands, of accounts with local Administrator privilege on the servers. In many cases, domain groups with large memberships are nested in member servers' local Administrators groups, without consideration to the fact that any user who can modify the memberships of those groups in the domain can gain administrative control of all systems on which the group has been nested in a local Administrators group.

Although workstations typically have significantly fewer members in their local Administrators groups than member servers do, in many environments, users are granted membership in the local Administrators group on their personal computers. When this occurs, even if UAC is enabled, those users present an elevated risk to the integrity of their workstations.

Important

You should consider carefully whether users require administrative rights on their workstations, and if they do, a better approach may be to create a separate local account on the computer that is a member of the Administrators group. When users require elevation, they can present the credentials of that local account for elevation, but because the account is local, it cannot be used to compromise other computers or access domain resources. As with any local accounts, however, the credentials for the local privileged account should be unique; if you create a local account with the same credentials on multiple workstations, you expose the computers to pass-the-hash attacks.

In attacks in which the target is an organization's intellectual property, accounts that have been granted powerful privileges within applications can be targeted to allow exfiltration of data. Although the accounts that have access to sensitive data may have been granted no elevated privileges in the domain or the operating system, accounts that can manipulate the configuration of an application or access to the information the application provides present risk.

As is the case with other targets, attackers seeking access to intellectual property in the form of documents and other files can target the accounts that control access to the file stores, accounts that have direct access to the files, or even groups or roles that have access to the files. For example, if a file server is used to store contract documents and access is granted to the documents by the use of an Active Directory group, an attacker who can modify the membership of the group can add compromised accounts to the group and access the contract documents. In cases in which access to documents is provided by applications such as SharePoint, attackers can target the applications as described earlier.

The larger and more complex an environment, the more difficult it is to manage and secure. In small organizations, reviewing and reducing privilege may be a relatively simple proposition, but each additional server, workstation, user account, and application in use in an organization adds another object that must be secured. Because it can be difficult or even impossible to properly secure every aspect of an organization's IT infrastructure, you should focus efforts first on the accounts whose privilege create the greatest risk, which are

typically the built-in privileged accounts and groups in Active Directory, and privileged local accounts on workstations and member servers.

Although this document focuses on securing Active Directory, as has been previously discussed, most attacks against the directory begin as attacks against individual hosts. Full guidelines for securing local groups on member systems cannot be provided, but the following recommendations can be used to help you secure the local Administrator accounts on workstations and member servers.

On all versions of Windows currently in mainstream support, the local Administrator account is disabled by default, which makes the account unusable for pass-the-hash and other credential theft attacks. However, in domains containing legacy operating systems or in which local Administrator accounts have been enabled, these accounts can be used as previously described to propagate compromise across member servers and workstations. For this reason, the following controls are recommended for all local Administrator accounts on domain-joined systems.

Detailed instructions for implementing these controls are provided in Appendix H: Securing Local Administrator Accounts and Groups. Before implementing these settings, however, ensure that local Administrator accounts are not currently used in the environment to run services on computers or perform other activities for which these accounts should not be used. Test these settings thoroughly before implementing them in a production environment.

Built-in Administrator accounts should never be used as service accounts on member servers, nor should they be used to log on to local computers (except in Safe Mode, which is permitted even if the account is disabled). The goal of implementing the settings described here is to prevent each computer's local Administrator account from being usable unless protective controls are first reversed. By implementing these controls and monitoring Administrator accounts for changes, you can significantly reduce the likelihood of success of an attack that targets local Administrator accounts.

In one or more GPOs that you create and link to workstation and member server OUs in each domain, add the Administrator account to the following user rights in **Computer Configuration\Policies\Windows Settings\Security Settings\Local Policies\User Rights Assignments**:

- Deny access to this computer from the network
- Deny log on as a batch job
- Deny log on as a service
- Deny log on through Remote Desktop Services

When you add Administrator accounts to these user rights, specify whether you are adding the local Administrator account or the domain's Administrator account by the way that you label the account. For example, to add the NWTRADERS domain's Administrator account to these deny rights, you would type the account as **NWTRADERS\Administrator**, or browse to the Administrator account for the NWTRADERS domain. To ensure that you restrict the local Administrator account, type **Administrator** in these user rights settings in the Group Policy Object Editor.

Note

Even if local Administrator accounts are renamed, the policies will still apply.

These settings will ensure that a computer's Administrator account cannot be used to connect to the other computers, even if it is inadvertently or maliciously enabled. Local logons using the local Administrator account cannot be completely disabled, nor should you attempt to do so, because a computer's local Administrator account is designed to be used in disaster recovery scenarios.

Should a member server or workstation become disjoined from the domain with no other local accounts granted administrative privileges, the computer can be booted into safe mode, the Administrator account can be enabled, and the account can then be used to effect repairs on the computer. When repairs are completed, the Administrator account should again be disabled.

Law Number Six: A computer is only as secure as the administrator is trustworthy. - Ten Immutable Laws of Security (Version 2.0)

The information provided here is intended to give general guidelines for securing the highest privilege built-in accounts and groups in Active Directory. Detailed step-by-step instructions are also provided in Appendix D: Securing Built-In Administrator Accounts in Active Directory, Appendix E: Securing Enterprise Admins Groups in Active Directory, Appendix F: Securing Domain Admins Groups in Active Directory, and in Appendix G: Securing Administrators Groups in Active Directory.

Before you implement any of these settings, you should also test all settings thoroughly to determine if they are appropriate for your environment. Not all organizations will be able to implement these settings.

In each domain in Active Directory, an Administrator account is created as part of the creation of the domain. This account is by default a member of the Domain Admins and Administrator groups in the domain, and if the domain is the forest root domain, the account is also a member of the Enterprise Admins group. Use of a domain's local Administrator account should be reserved only for initial build activities and, possibly, disaster-recovery scenarios. To ensure that a built-in Administrator account can be used to effect repairs in the event that no other accounts can be used, you should not change the default membership of the Administrator account in any domain in the forest. Instead, you should following guidelines to help secure the Administrator account in each domain in the forest. Detailed instructions for implementing these controls are provided in Appendix D: Securing Built-In Administrator Accounts in Active Directory.

The goal of implementing the settings described here is to prevent each domain's Administrator account (not a group) from being usable unless a number of controls are reversed. By implementing these controls and monitoring the Administrator accounts for changes, you can significantly reduce the likelihood of a successful attack by leveraging a domain's Administrator account. For the Administrator account in each domain in your forest, you should configure the following settings.

By default, all accounts in Active Directory can be delegated. Delegation allows a computer or service to present the credentials for an account that has authenticated to the computer or service to other computers to obtain services on behalf of the account. When you enable the **Account is sensitive and cannot be delegated** attribute on a domain-based account, the account's credentials cannot be presented to other computers or services on the network, which limits attacks that leverage delegation to use the account's credentials on other systems.

When you enable the **Smart card is required for interactive logon** attribute on an account, Windows resets the account's password to a 120-character random value. By setting this flag on built-in Administrator accounts, you ensure that the password for the account is not only long and complex, but is not known to any user. It is not technically necessary to create smart cards for the accounts before enabling this attribute, but if possible, smart cards should be created for each Administrator account prior to configuring the account restrictions and the smart cards should be stored in secure locations.

Although setting the **Smart card is required for interactive logon** flag resets the account's password, it does not prevent a user with rights to reset the account's password from setting the account to a known value and using the account's name and new password to access resources on the network. Because of this, you should implement the following additional controls on the account.

Although disabling the Administrator account in a domain makes the account effectively unusable, you should implement additional restrictions on the account in case the account is inadvertently or maliciously enabled. Although these controls can ultimately be reversed by the Administrator account, the goal is to create controls that slow an attacker's progress and limit the damage the account can inflict.

In one or more GPOs that you create and link to workstation and member server OUs in each domain, add each domain's Administrator account to the following user rights in **Computer Configuration\Policies\Windows Settings\Security Settings\Local Policies\User Rights Assignments**:

- Deny access to this computer from the network
- Deny log on as a batch job
- Deny log on as a service
- Deny log on through Remote Desktop Services

Note

When you add local Administrator accounts to this setting, you must specify whether you are configuring local Administrator accounts or domain Administrator accounts. For example, to add the NWTRADERS domain's local Administrator account to these deny rights, you must either type the account as **NWTRADERS\Administrator**, or browse to the local Administrator account for the NWTRADERS domain. If you type **Administrator** in these user rights settings in the Group Policy Object Editor, you will restrict the local Administrator account on each computer to which the GPO is applied.

We recommend restricting local Administrator accounts on member servers and workstations in the same manner as domain-based Administrator accounts. Therefore, you should generally add the Administrator account for each domain in the forest and the Administrator account for the local computers to these user rights settings.

In each domain in the forest, the Default Domain Controllers policy or a policy linked to the Domain Controllers OU should be modified to add each domain's Administrator account to the following user rights in **Computer Configuration\Policies\Windows Settings\Security Settings\Local Policies\User Rights Assignments**:

- Deny access to this computer from the network
- Deny log on as a batch job
- Deny log on as a service

- Deny log on through Remote Desktop Services

Note

These settings will ensure that the local Administrator account cannot be used to connect to a domain controller, although the account, if enabled, can log on locally to domain controllers. Because this account should only be enabled and used in disaster-recovery scenarios, it is anticipated that physical access to at least one domain controller will be available, or that other accounts with permissions to access domain controllers remotely can be used.

When you have secured each domain's Administrator account and disabled it, you should configure auditing to monitor for changes to the account. If the account is enabled, its password is reset, or any other modifications are made to the account, alerts should be sent to the users or teams responsible for administration of AD DS, in addition to incident response teams in your organization.

The Enterprise Admins group, which is housed in the forest root domain, should contain no users on a day-to-day basis, with the possible exception of the domain's local Administrator account, provided it is secured as described earlier and in Appendix D: Securing Built-In Administrator Accounts in Active Directory.

When EA access is required, the users whose accounts require EA rights and permissions should be temporarily placed into the Enterprise Admins group. Although users are using the highly privileged accounts, their activities should be audited and preferably performed with one user performing the changes and another user observing the changes to minimize the likelihood of inadvertent misuse or misconfiguration. When the activities have been completed, the accounts should be removed from the EA group. This can be achieved via manual procedures and documented processes, third-party privileged identity/access management (PIM/PAM) software, or a combination of both. Guidelines for creating accounts that can be used to control the membership of privileged groups in Active Directory are provided in Attractive Accounts for Credential Theft and detailed instructions are provided in Appendix I: Creating Management Accounts for Protected Accounts and Groups in Active Directory.

Enterprise Admins are, by default, members of the built-in Administrators group in each domain in the forest. Removing the Enterprise Admins group from the Administrators groups in each domain is an inappropriate modification because in the event of a forest disaster-recovery scenario, EA rights will likely be required. If the Enterprise Admins group has been removed from Administrators groups in a forest, it should be added to the Administrators group in each domain and the following additional controls should be implemented:

- As described earlier, the Enterprise Admins group should contain no users on a day-to-day basis, with the possible exception of the forest root domain's Administrator account, which should be secured as described in Appendix D: Securing Built-In Administrator Accounts in Active Directory.
- In GPOs linked to OUs containing member servers and workstations in each domain, the EA group should be added to the following user rights:
    - Deny access to this computer from the network
    - Deny log on as a batch job
    - Deny log on as a service
    - Deny log on locally
    - Deny log on through Remote Desktop Services.

This will prevent members of the EA group from logging on to member servers and workstations. If jump servers are used to administer domain controllers and Active Directory, ensure that jump servers are located in an OU to which the restrictive GPOs are not linked.

- Auditing should be configured to send alerts if any modifications are made to the properties or membership of the EA group. These alerts should be sent, at a minimum, to users or teams responsible for Active Directory administration and incident response. You should also define processes and procedures for temporarily populating the EA group, including notification procedures when legitimate population of the group is performed.

As is the case with the Enterprise Admins group, membership in Domain Admins groups should be required only in build or disaster-recovery scenarios. There should be no day-to-day user accounts in the DA group with the exception of the local Administrator account for the domain, if it has been secured as described in Appendix D: Securing Built-In Administrator Accounts in Active Directory.

When DA access is required, the accounts needing this level of access should be temporarily placed in the DA group for the domain in question. Although the users are using the highly privileged accounts, activities should be audited and preferably performed with one user performing the changes and another user observing the changes to minimize the likelihood of inadvertent misuse or misconfiguration. When the activities have been completed, the accounts should be removed from the Domain Admins group. This can be achieved via manual procedures and documented processes, via third-party privileged identity/access management (PIM/PAM) software, or a combination of both. Guidelines for creating accounts that can be used to control the membership of privileged groups in Active Directory are provided in Appendix I: Creating Management Accounts for Protected Accounts and Groups in Active Directory.

Domain Admins are, by default, members of the local Administrators groups on all member servers and workstations in their respective domains. This default nesting should not be modified because it affects supportability and disaster recovery options. If Domain Admins groups have been removed from the local Administrators groups on the member servers, they should be added to the Administrators group on each member server and workstation in the domain via restricted group settings in linked GPOs. The following general controls, which are described in depth in Appendix F: Securing Domain Admins Groups in Active Directory should also be implemented.

For the Domain Admins group in each domain in the forest:

1. Remove all members from the DA group, with the possible exception of the built-in Administrator account for the domain, provided it has been secured as described in Appendix D: Securing Built-In Administrator Accounts in Active Directory.

2. In GPOs linked to OUs containing member servers and workstations in each domain, the DA group should be added to the following user rights:

    - Deny access to this computer from the network
    - Deny log on as a batch job
    - Deny log on as a service

- Deny log on locally
- Deny log on through Remote Desktop Services

This will prevent members of the DA group from logging on to member servers and workstations. If jump servers are used to administer domain controllers and Active Directory, ensure that jump servers are located in an OU to which the restrictive GPOs are not linked.

3. Auditing should be configured to send alerts if any modifications are made to the properties or membership of the DA group. These alerts should be sent, at a minimum, to users or teams responsible for AD DS administration and incident response. You should also define processes and procedures for temporarily populating the DA group, including notification procedures when legitimate population of the group is performed.

As is the case with the EA and DA groups, membership in the Administrators (BA) group should be required only in build or disaster-recovery scenarios. There should be no day-to-day user accounts in the Administrators group with the exception of the local Administrator account for the domain, if it has been secured as described in Appendix D: Securing Built-In Administrator Accounts in Active Directory.

When Administrators access is required, the accounts needing this level of access should be temporarily placed in the Administrators group for the domain in question. Although the users are using the highly privileged accounts, activities should be audited and, preferably, performed with a user performing the changes and another user observing the changes to minimize the likelihood of inadvertent misuse or misconfiguration. When the activities have been completed, the accounts should immediately be removed from the Administrators group. This can be achieved via manual procedures and documented processes, via third-party privileged identity/access management (PIM/PAM) software, or a combination of both.

Administrators are, by default, the owners of most of the AD DS objects in their respective domains. Membership in this group may be required in build and disaster recovery scenarios in which ownership or the ability to take ownership of objects is required. Additionally, DAs and EAs inherit a number of their rights and permissions by virtue of their default membership in the Administrators group. Default group nesting for privileged groups in Active Directory should not be modified, and each domain's Administrators group should be secured as described in Appendix G: Securing Administrators Groups in Active Directory, and in the general instructions below.

1. Remove all members from the Administrators group, with the possible exception of the local Administrator account for the domain, provided it has been secured as described in Appendix D: Securing Built-In Administrator Accounts in Active Directory.

2. Members of the domain's Administrators group should never need to log on to member servers or workstations. In one or more GPOs linked to workstation and member server OUs in each domain, the Administrators group should be added to the following user rights:

   - Deny access to this computer from the network
   - Deny log on as a batch job,
   - Deny log on as a service

- This will prevent members of the Administrators group from being used to log on or connect to member servers or workstations (unless multiple controls are first breached), where their credentials could be cached and thereby compromised. A privileged account should never be used to log on to a less-privileged system, and enforcing these controls affords protection against a number of attacks.

3. At the domain controllers OU in each domain in the forest, the Administrators group should be granted the following user rights (if they do not already have these rights), which will allow the members of the Administrators group to perform functions necessary for a forest-wide disaster recovery scenario:

    - Access this computer from the network
    - Allow log on locally
    - Allow log on through Remote Desktop Services

4. Auditing should be configured to send alerts if any modifications are made to the properties or membership of the Administrators group. These alerts should be sent, at a minimum, to members of the team responsible for AD DS administration. Alerts should also be sent to members of the security team, and procedures should be defined for modifying the membership of the Administrators group. Specifically, these processes should include a procedure by which the security team is notified when the Administrators group is going to be modified so that when alerts are sent, they are expected and an alarm is not raised. Additionally, processes to notify the security team when the use of the Administrators group has been completed and the accounts used have been removed from the group should be implemented.

Note

When you implement restrictions on the Administrators group in GPOs, Windows applies the settings to members of a computer's local Administrators group in addition to the domain's Administrators group. Therefore, you should use caution when implementing restrictions on the Administrators group. Although prohibiting network, batch and service logons for members of the Administrators group is advised wherever it is feasible to implement, do not restrict local logons or logons through Remote Desktop Services. Blocking these logon types can block legitimate administration of a computer by members of the local Administrators group. The following screenshot shows configuration settings that block misuse of built-in local and domain Administrator accounts, in addition to misuse of built-in local or domain Administrators groups. Note that the **Deny log on through Remote Desktop Services** user right does not include the Administrators group, because including it in this setting would also block these logons for accounts that are members of the local computer's Administrators group. If services on computers are configured to run in the context of any of the privileged groups described in this section, implementing these settings can cause services and applications to fail. Therefore, as with all of the recommendations in this section, you should thoroughly test settings for applicability in your environment.

 least privilege admin models

Generally speaking, role-based access controls (RBAC) are a mechanism for grouping users and providing access to resources based on business rules. In the case of Active Directory, implementing RBAC for AD DS is the process of creating roles to which rights and permissions are delegated to allow members of the role to perform day-to-day administrative tasks without granting them excessive privilege. RBAC for Active Directory can be designed and implemented via native tooling and interfaces, by leveraging software you may already own, by

purchasing third-party products, or any combination of these approaches. This section does not provide step-by-step instructions to implement RBAC for Active Directory, but instead discusses factors you should consider in choosing an approach to implementing RBAC in your AD DS installations.

In the simplest RBAC implementation, you can implement roles as AD DS groups and delegate rights and permissions to the groups that allow them to perform daily administration within the designated scope of the role.

In some cases, existing security groups in Active Directory can be used to grant rights and permissions appropriate to a job function. For example, if specific employees in your IT organization are responsible for the management and maintenance of DNS zones and records, delegating those responsibilities can be as simple as creating an account for each DNS administrator and adding it to the DNS Admins group in Active Directory. The DNS Admins group, unlike more highly privileged groups, has few powerful rights across Active Directory, although members of this group have been delegated permissions that allow them to administer DNS and is still subject to compromise and abuse could result in elevation of privilege.

In other cases, you may need to create security groups and delegate rights and permissions to Active Directory objects, file system objects, and registry objects to allow members of the groups to perform designated administrative tasks. For example, if your Help Desk operators are responsible for resetting forgotten passwords, assisting users with connectivity problems, and troubleshooting application settings, you may need to combine delegation settings on user objects in Active Directory with privileges that allow Help Desk users to connect remotely to users' computers to view or modify the users' configuration settings. For each role you define, you should identify:

1. Which tasks members of the role perform on a day-to-day basis and which tasks are less frequently performed.
2. On which systems and in which applications members of a role should be granted rights and permissions.
3. Which users should be granted membership in a role.
4. How management of role memberships will be performed.

In many environments, manually creating role-based access controls for administration of an Active Directory environment can be challenging to implement and maintain. If you have clearly defined roles and responsibilities for administration of your IT infrastructure, you may want to leverage additional tooling to assist you in creating a manageable native RBAC deployment. For example, if Forefront Identity Manager (FIM) is in use in your environment, you can use FIM to automate the creation and population of administrative roles, which can ease ongoing administration. If you use Microsoft Endpoint Configuration Manager and System Center Operations Manager (SCOM), you can use application-specific roles to delegate management and monitoring functions, and also enforce consistent configuration and auditing across systems in the domain. If you have implemented a public key infrastructure (PKI), you can issue and require smart cards for IT staff responsible for administering the environment. With FIM Credential Management (FIM CM), you can even combine management of roles and credentials for your administrative staff.

In other cases, it may be preferable for an organization to consider deploying third-party RBAC software that provides "out-of-box" functionality. Commercial, off-the-shelf (COTS) solutions for RBAC for Active Directory,

Windows, and non-Windows directories and operating systems are offered by a number of vendors. When choosing between native solutions and third-party products, you should consider the following factors:

1. Budget: By investing in development of RBAC using software and tools you may already own, you can reduce the software costs involved in deploying a solution. However, unless you have staff who are experienced in creating and deploying native RBAC solutions, you may need to engage consulting resources to develop your solution. You should carefully weigh the anticipated costs for a custom-developed solution with the costs to deploy an "out-of-box" solution, particularly if your budget is limited.
2. Composition of the IT environment: If your environment is comprised primarily of Windows systems, or if you are already leveraging Active Directory for management of non-Windows systems and accounts, custom native solutions may provide the optimal solution for your needs. If your infrastructure contains many systems that are not running Windows and are not managed by Active Directory, you may need to consider options for management of non-Windows systems separately from the Active Directory environment.
3. Privilege model in the solution: If a product relies on placement of its service accounts into highly privileged groups in Active Directory and does not offer options that do not require excessive privilege be granted to the RBAC software, you have not really reduced your Active Directory attack surface you've only changed the composition of the most privileged groups in the directory. Unless an application vendor can provide controls for service accounts that minimize the probability of the accounts being compromised and maliciously used, you may want to consider other options.

Privileged identity management (PIM), sometimes referred to as privileged account management (PAM) or privileged credential management (PCM) is the design, construction, and implementation of approaches to managing privileged accounts in your infrastructure. Generally speaking, PIM provides mechanisms by which accounts are granted temporary rights and permissions required to perform build-or-break fix functions, rather than leaving privileges permanently attached to accounts. Whether PIM functionality is manually created or is implemented via the deployment of third-party software one or more of the following features may be available:

- Credential "vaults," where passwords for privileged accounts are "checked out" and assigned an initial password, then "checked in" when activities have been completed, at which time passwords are again reset on the accounts.
- Time-bound restrictions on the use of privileged credentials
- One-time-use credentials
- Workflow-generated granting of privilege with monitoring and reporting of activities performed and automatic removal of privilege when activities are completed or allotted time has expired
- Replacement of hard-coded credentials such as user names and passwords in scripts with application programming interfaces (APIs) that allow credentials to be retrieved from vaults as needed
- Automatic management of service account credentials

One of the challenges in managing privileged accounts is that, by default, the accounts that can manage privileged and protected accounts and groups are privileged and protected accounts. If you implement appropriate RBAC and PIM solutions for your Active Directory installation, the solutions may include approaches that allow you to effectively depopulate the membership of the most privileged groups in the directory, populating the groups only temporarily and when needed.

If you implement native RBAC and PIM, however, you should consider creating accounts that have no privilege and with the only function of populating and depopulating privileged groups in Active Directory when needed. Appendix I: Creating Management Accounts for Protected Accounts and Groups in Active Directory provides step-by-step instructions that you can use to create accounts for this purpose.

Law Number Six: There really is someone out there trying to guess your passwords. - 10 Immutable Laws of Security Administration

Pass-the-hash and other credential theft attacks are not specific to Windows operating systems, nor are they new. The first pass-the-hash attack was created in 1997. Historically, however, these attacks required customized tools, were hit-or-miss in their success, and required attackers to have a relatively high degree of skill. The introduction of freely available, easy-to-use tooling that natively extracts credentials has resulted in an exponential increase in the number and success of credential theft attacks in recent years. However, credential theft attacks are by no means the only mechanisms by which credentials are targeted and compromised.

Although you should implement controls to help protect you against credential theft attacks, you should also identify the accounts in your environment that are most likely to be targeted by attackers, and implement robust authentication controls for those accounts. If your most privileged accounts are using single factor authentication such as user names and passwords (both are "something you know," which is one authentication factor), those accounts are weakly protected. All that an attacker needs is knowledge of the user name and knowledge of the password associated with the account, and pass-the-hash attacks are not required the attacker can authenticate as the user to any systems that accept single factor credentials.

Although implementing multi-factor authentication does not protect you against pass-the-hash attacks, implementing multi-factor authentication in combination with protected systems can. More information about implementing protected systems is provided in Implementing Secure Administrative Hosts, and authentication options are discussed in the following sections.

If you have not already implemented multi-factor authentication such as smart cards, consider doing so. Smart cards implement hardware-enforced protection of private keys in a public-private key pair, preventing a user's private key from being accessed or used unless the user presents the proper PIN, passcode, or biometric identifier to the smart card. Even if a user's PIN or passcode is intercepted by a keystroke logger on a compromised computer, for an attacker to reuse the PIN or passcode, the card must also be physically present.

In cases in which long, complex passwords have proven difficult to implement because of user resistance, smart cards provide a mechanism by which users may implement relatively simple PINs or passcodes without the credentials being susceptible to brute force or rainbow table attacks. Smart card PINs are not stored in Active Directory or in local SAM databases, although credential hashes may still be stored in LSASS protected memory on computers on which smart cards have been used for authentication.

Another benefit of implementing smart cards or other certificate-based authentication mechanisms is the ability to leverage Authentication Mechanism Assurance to protect sensitive data that is accessible to VIP users. Authentication Mechanism Assurance is available in domains in which the functional level is set to Windows Server 2012 or Windows Server 2008 R2. When it is enabled, Authentication Mechanism Assurance adds an

administrator-designated global group membership to a user's Kerberos token when the user's credentials are authenticated during logon using a certificate-based logon method.

This makes it possible for resource administrators to control access to resources, such as files, folders, and printers, based on whether the user logs on using a certificate-based logon method, in addition to the type of certificate used. For example, when a user logs on by using a smart card, the user's access to resources on the network can be specified as different from what the access is when the user does not use a smart card (that is, when the user logs on by entering a user name and password). For more information about Authentication Mechanism Assurance, see the Authentication Mechanism Assurance for AD DS in Windows Server 2008 R2 Step-by-Step Guide.

In Active Directory for all administrative accounts, enable the **Require smart card for interactive logon** attribute, and audit for changes to (at a minimum), any of the attributes on the **Account** tab for the account (for example, cn, name, sAMAccountName, userPrincipalName, and userAccountControl) administrative user objects.

Although setting the **Require smart card for interactive logon** on accounts resets the account's password to a 120-character random value and requires smart cards for interactive logons, the attribute can still be overwritten by users with permissions that allow them to change passwords on the accounts, and the accounts can then be used to establish non-interactive logons with only user name and password.

In other cases, depending on the configuration of accounts in Active Directory and certificate settings in Active Directory Certificate Services (AD CS) or a third-party PKI, User Principal Name (UPN) attributes for administrative or VIP accounts can be targeted for a specific kind of attack, as described here.

Although a thorough discussion of attacks against public key infrastructures (PKIs) is outside the scope of this document, attacks against public and private PKIs have increased exponentially since 2008. Breaches of public PKIs have been broadly publicized, but attacks against an organization's internal PKI are perhaps even more prolific. One such attack leverages Active Directory and certificates to allow an attacker to spoof the credentials of other accounts in a manner that can be difficult to detect.

When a certificate is presented for authentication to a domain-joined system, the contents of the Subject or the Subject Alternative Name (SAN) attribute in the certificate are used to map the certificate to a user object in Active Directory. Depending on the type of certificate and how it is constructed, the Subject attribute in a certificate typically contains a user's common name (CN), as shown in the following screenshot.

 Screenshot that shows the Subject attribute in a certificate typically contains a user's common name.

By default, Active Directory constructs a user's CN by concatenating the account's first name + " "+ last name. However, CN components of user objects in Active Directory are not required or guaranteed to be unique, and moving a user account to a different location in the directory changes the account's distinguished name (DN), which is the full path to the object in the directory, as shown in the bottom pane of the previous screenshot.

Because certificate subject names are not guaranteed to be static or unique, the contents of the Subject Alternative Name are often used to locate the user object in Active Directory. The SAN attribute for certificates issued to users from enterprise certification authorities (Active Directory integrated CAs) typically contains the user's UPN or

email address. Because UPNs are guaranteed to be unique in an AD DS forest, locating a user object by UPN is commonly performed as part of authentication, with or without certificates involved in the authentication process.

The use of UPNs in SAN attributes in authentication certificates can be leveraged by attackers to obtain fraudulent certificates. If an attacker has compromised an account that has the ability to read and write UPNs on user objects, the attack is implemented as follows:

The UPN attribute on a user object (such as a VIP user) is temporarily changed to a different value. The SAM account name attribute and CN can also be changed at this time, although this is usually not necessary for the reasons described earlier.

When the UPN attribute on the target account has been changed, a stale, enabled user account or a freshly created user account's UPN attribute is changed to the value that was originally assigned to the target account. Stale, enabled user accounts are accounts that have not logged on for long periods of time, but have not been disabled. They are targeted by attackers who intend to "hide in plain sight" for the following reasons:

1. Because the account is enabled, but hasn't been used recently, using the account is unlikely to trigger alerts the way that enabling a disabled user account might.
2. Use of an existing account doesn't require the creation of a new user account that might be noticed by administrative staff.
3. Stale user accounts that are still enabled are usually members of various security groups and are granted access to resources on the network, simplifying access and "blending in" to an existing user population.

The user account on which the target UPN has now been configured is used to request one or more certificates from Active Directory Certificate Services.

When certificates have been obtained for the attacker's account, the UPNs on the "new" account and the target account are returned to their original values.

The attacker now has one or more certificates that can be presented for authentication to resources and applications as if the user is the VIP user whose account was temporarily modified. Although a full discussion of all of the ways in which certificates and PKI can be targeted by attackers is outside the scope of this document, this attack mechanism is provided to illustrate why you should monitor privileged and VIP accounts in AD DS for changes, particularly for changes to any of the attributes on the **Account** tab for the account (for example, cn, name, sAMAccountName, userPrincipalName, and userAccountControl). In addition to monitoring the accounts, you should restrict who can modify the accounts to as small a set of administrative users as possible. Likewise, the accounts of administrative users should be protected and monitored for unauthorized changes.
