netwrix

What is Active Directory?

Brian Svidergol

MCITP, MCSE, RHEL3, VCP, NCIE-SAN, MCT, MCSA,

Microsoft Certified Solutions Expert

Table of Contents

	Introduction	3
Chapter I	Introduction to Directory Services Technologies	4
Chapter II	Administration Tools	5
Chapter III	Inside AD	6
	Domain Controllers	6
	SYSVOL	8
Chapter IV	Forests, Domains, Trusts	10
	Forests	10
	Domains	11
	Trusts	13
Chapter V	Group Policy	16
Chapter VI	Inside the AD DS Database	18
Chapter VII	Replication	20
Chapter VIII	DNS and DHCP	23
Chapter IX	Security	26
Chapter X	Auditing	28
Chapter XI	User Unfriendly Log Data	31
	Useful References	32
	About the Author	34
	About Netwrix	34

netwrix

Introduction

IT administrators have been working with and around Active Directory since the introduction of the technology in Windows 2000 Server. Windows 2000 Server was released on February 17, 2000 but many administrators began working with Active Directory in late 1999 when it was released to manufacturing (RTM) on December 15, 1999.

This e-book is intended not only for beginner SysAdmins who want learn about Active Directory structure, key terminology and configurations. More experienced administrators will find a few best practices of AD management and discover the areas that are often lesser known to IT pros.

netwrix

I. Introduction to Directory Services Technologies

Like many other areas of IT, directory services has rapidly expanded with new features and functionality along with additional complexity. Instead of a single directory product such as AD DS, there are quite a few other services that make up the directory services category. In addition to the Microsoft solutions, many third-party vendors are creating products that standalone on their own or enhance and expand the Microsoft offerings. Today, directory services technologies from Microsoft includes the following products:

- Active Directory Domain Services (AD DS). AD DS is the core focus of this e-book so it doesn't require an introduction. But, how about an interesting fact instead? According to Microsoft Corporate Vice President Takeshi Numoto, Active Directory is used by 93% of the Fortune 1000.
- Active Directory Lightweight Directory Services (AD LDS). AD LDS is the lightweight, developer- friendly, directory that can be deployed on a client computer and client operating system as well as on a server. It isn't as full featured as AD DS (for example, Group Policy isn't part of it) but it can be useful as a decentralized directory for developers and testers.
- Active Directory Federation Services (AD FS). AD FS is a claims-based identity solution that helps independent organizations connect their directory services technologies together to facilitate single sign-on and cross-organizational resource access. Today, it has become a fairly common solution because it helps organizations connect to cloud services such as Microsoft Azure.

Additionally, there are two other roles that you may be wondering about. Active Directory Certificate Services (AD CS) and Active Directory Rights Management Services (AD RMS) are often grouped in with the other technologies listed above to form the suite of technologies offered by Microsoft for on-premise Active Directory related deployments. Additionally, there are products outside of the immediate Active Directory family such as Microsoft Forefront Identity Manager (FIM). Beyond the on-premise technologies, there are also several cloud-based solutions that offer services in the cloud such as Azure Active Directory and Azure Multi-Factor Authentication. For this white paper, we are focusing on AD DS deployed on-premise.

netwrix

II. Administration Tools

There are numerous tools for Active Directory. The tool that we will cover today is Active Directory Users and Computers (ADUC), which was released with Windows 2000 Server. ADUC is an MMC snap-in that enables administrators to manage Active Directory objects, including users, computers, groups, organizational units (OUs), and attributes. While the features of ADUC and many other features have been added to a new tool named Active Directory Administrative Center, ADUC remains a popular tool that administrator's use to manage their environment.

In addition to managing objects, ADUC can also manage domain operations. For example, you can raise the domain functional level from ADUC. You can also transfer the RID, PDC Emulator, and Infrastructure FSMO roles to a different domain controller by using ADUC.

Managing an object consists of some of the more obvious tasks such as resetting a user's password, adding users to security groups, and moving computer objects. However, the Advanced Features setting within ADUC can also allow you to manage the LostAndFound container, NTDS Quotas, Program Data, and System information. This view is not enabled by default but you can enable through the View menu. The Advanced Features option adds many tabs to the properties page of an object, including Published Certificates, Attribute Editor, Password Replication, and others.

The View menu also allows you to filter the view based on the object type, such as user, computer, printer and more. Individual columns can also be added or removed, to customize the view to include other attributes that have been assigned to the object, for example the last modify date, city, country, email address, and more. Finally, ADUC also enables you to delegate control of objects through the Delegation of Control wizard or by manually modifying permissions on an object.

III. Inside AD

There are many aspects to an AD DS deployment. In this white paper, we will look at domain controllers, the AD DS database, and SYSVOL. While we only focus on parts of these three components in this e-book, other parts and components are also important.

Domain Controllers

The domain controller is the backbone of Active Directory. Without a domain controller, you can't have a directory! You can use up to 1,200 domain controllers in a single domain. But, don't judge another administrator's environment by the size or scale of it! Let's look at the evolution of the domain controller:

- Windows NT 3.1 introduced the original Microsoft domain. Windows NT 3.1 (subsequently 3.5 and then 3.51) should not be confused with Windows 3.1 which was a 16-bit client operating system. The domain functionality included with Windows NT was not a multi-master model like AD DS. Thus, there was a primary domain controller (PDC) and backup domain controllers (BDCs). All changes were handled by the PDC. A BDC could be promoted to a PDC in a disaster recovery situation. Today, we have the PDC Emulator FSMO role which is directly related to the original PDC.

- Windows 2000 Server introduced Active Directory. With the release of Windows 2000 Server, Microsoft revamped a large amount of the traditional domain and marketed the service as Active Directory. A key feature of Active Directory was the multi-master model which allowed most of the Active Directory functionality, including changes, to take place on any DC in the domain.

- Windows Server 2003 introduced new features. With Windows Server 2003, Active Directory was updated with some administrative enhancements (such as multi-selecting objects in ADUC), added the ability to create forest trusts, and added the universal group membership caching feature. Other features were added or expanded too, especially around command-line administration.

- Windows Server 2003 R2 introduced AD FS and Active Directory Application Mode (ADAM). AD FS and ADAM were big enhancements, especially looking at them today in 2015. Back then, they weren't used much though. ADAM later became AD LDS while AD FS was updated along the way for cloud integration.

- Windows Server 2008 introduced read-only domain controllers (RODCs) and fine-grained password policies. With Windows Server 2008, RODCs became an option which allowed administrators to deploy DCs in insecure computer closets at branch offices, among other uses. In addition, fine-grained password policies were introduced, albeit with some administrative challenges such as not having a graphical user interface to manage the policies.

- Windows Server 2008 R2 introduced the recycle bin and the PowerShell module. Windows Server 2008 R2 continued refining some of the features introduced in Windows Server 2008 and offered the Recycle Bin and a PowerShell module which was paramount for administrators to be able to effectively manage AD DS from PowerShell.

- Windows Server 2012 introduced simplified management and enhanced virtualization support. The long awaited graphical user interface tools to manage the Recycle Bin and fine-grained password policies were introduced. Additionally, virtualization was enhanced and support for virtualizing DCs became mainstream. See https://technet.microsoft.com/en-us/library/hh831477.aspx for a complete guide on the changes.

- Windows Server 2012 R2 focused on security enhancements. New features included multi-factor authentication, single sign-on from connected devices, and multi-factor access control. See https://technet.microsoft.com/en-us/library/dn268294.aspx for a complete guide on the changes.

There are some good practices to adhere to when deploying DCs. Many of these practices are documented. But not many organizations are implementing these practices. We will skip over the well-known good practices such as maintaining the Active Directory database on one set of disk spindles, the log files on separate disk spindles, and the operating system on its own set of disk spindles. Some of the lesser implemented good practices for domain controllers are:

Run the Server Core installation of the operating system

Many administrators avoid change, especially for systems such as AD DS that are incredibly stable. So when a new administrator proposes switching over to the Server Core installation, he is often met with icy stares. But the reality is that most administrators administrate AD DS remotely by launching ADUC or PowerShell on their client or administrative computer. All of the core management tools including the Active Directory Administrative Center (ADAC) and Windows PowerShell work almost identically when used locally on a DC or remotely from a client computer or an administrative computer. Thus, by moving to the Server Core installation, the administrative experience isn't degraded. And, you gain security enhancements and some small performance enhancements.

Do not run other software or services on a DC

Back in the old days, like 10 years ago, most organizations used physical servers because virtualization was in its infancy. So, when it was time to provision a new file server, DHCP server, or print server, administrators often just tapped an existing server. A DC was often used too. Fast forward to 2015 when virtualization is the de facto standard and automated provisioning helps deliver a new VM in minutes and the old way of doing things isn't nearly as compelling.

Now, when you need a place for a file server, DHCP server, print server, or some other application server, you can provision a new VM. Or, better yet, you can provision a new VM as a utility server. A utility server is a server that hosts all of the applications and services that are too small to warrant a dedicated server. This allows your DCs to stick with a dedicated service which brings more stability.

Adjust the startup order and set a BIOS password

While all of your read-write DCs should be in a secure data center, there are plenty of IT and non-IT people that have access to the data center. For example, the contracted electricians that works on the cooling system have data center access. In addition, there are likely network guys, cabling guys, and IT management with data center access. Anybody that has physical access to a DC can gain access to a physical DC in only a couple of minutes at a console in the data center. There are specialized freeware boot images available that you can use to boot into and reset passwords, install malware, or gain access to the disk data, assuming that the disk isn't encrypted. To avoid this, perform the following configurations:

- Ensure that all removable media is not part of the BIOS boot order. Instead, only the hard disk where the operating system installed should be part of the boot order. This is true for your virtualization host servers too, if you have virtual DCs.
- Set a strong BIOS password. If you don't set a BIOS password, somebody can update the boot order, boot to the Windows Server installation media or many freeware toolkits, perform a repair to get to a command prompt. Once at the command prompt, they can wreak some havoc and quickly reset passwords for domain accounts.
- Keep the DCs in a locked cabinet. While a BIOS password is one layer of security, if the attacker is semi-capable, he or she will likely know how to reset the BIOS so that the configuration resets and password is removed. Often, this requires gaining access to the motherboard. You can reduce the risk of such an attack by keeping DCs in a locked cabinet. Some servers also allow for chassis locks. In high security environments, you should opt for both.

Standardize the configuration of all domain controllers.

You should try to match the configuration settings for each DC. You can accomplish some of this by using build automation through deployment tools such as System Center Configuration Manager. Items of interest for DCs are the event log size settings to ensure that you have large sizes to capture auditing and security related information, boot settings such as the timeout waiting for OS selection on physical servers, firmware and BIOS versions and settings, and hardware configuration. Of course, there are many other configuration items to standardize by using Group Policy. The primary goal is to configure the DCs identically.

SYSVOL

The system volume (SYSVOL) is a special directory on each DC. It is made up of several folders with one being shared and referred to as the SYSVOL share. The default location is %SYSTEMROOT%\SYSVOL\sysvol for the shared folder, although you can change that during the DC promotion process or anytime thereafter. SYSVOL is made up of:

- Folders. The folders are used to store:
  - Group Policy templates (GPTs), which are replicated via SYSVOL container (GPC) is replicated via Active Directory replication
  - Scripts, such as startup scripts that are referenced in a GPO.
- Junction points. Junction points work like a shortcut. One directory can point to a different directory. In File Explorer, a junction point and a directory look and feel the same. You can view junction points by running the *dir /AL /S* command.

SYSVOL replication occurs over DFSR. Initially with Windows 2000 Server, Windows Server 2003, and Windows Server 2003 R2, replication was handled by File Replication Service (FRS). Starting with domains created in Windows Server 2008, DFSR is the default SYSVOL replication method. FRS wasn't very efficient. Any time that a file in SYSVOL changed, FRS replicated the entire file to all domain controllers. With DFSR, only the changed part of the file is replicated, although only for files over 64KB. DFSR uses Remote Differential Compression (RDC). RDC is what enables the replication of only changed data. Some admins may remember migrating from FRS to DFSR when Windows Server 2008 was released. Without reliable and timely replication, one side effect that users may experience is inconsistent GPO application since the SYSVOL data may not be in sync across all of the DCs.

IV. Forests, Domains, Trusts

Forests and domains are fairly well understood by administrators. Trusts aren't though. In this section, we'll take a look at some portions of forests, domains, and trusts and discuss some good practices around them.

Forests

A forest is the top most logical container in an AD DS environment. It was first introduced with Active Directory in Windows Server 2000. A forest is made up of one or more domains and all of the objects in the domains. In the database, a forest is a just a container, similar to many of the objects below it such as domains and OUs. Importantly, the forest is the defined security boundary for an AD DS environment. In the early days of Active Directory, the domain was originally defined as the security boundary. Unlikely many of the other components that we discuss in this white paper, there aren't any direct limitations on the number of forests that you can deploy. Since they are the top most object, you can create as many as you want, assuming that you have enough physical servers or VMs (don't take this as a recommendation though!). There are three forest-wide directory partitions in a forest:

- Schema. The schema partition defines all of the classes, objects, and attributes that can be used. The schema is shared among all of the domains in the forest. Objects such as users, groups, and OUs are defined in the schema.

- Configuration. The configuration partition is responsible for managing the forest topology, forest settings, and domain settings. You can find a list of all of the domains, DCs, and GCs in the configuration partition. You can view the configuration partition in a domain named contoso.com by viewing *cn=configuration,dc=contoso,dc=com in ADSIEdit*.

- Application. The application partition is used to store application data. A common example of data in the application partition is DNS.

Of the 5 FSMO roles, 2 of the roles are specific to the forest:

- Schema Master. This role is used for schema updates. As such, the role holder must be online and available to perform a schema update.

- Domain Naming Master. This role is used to add and remove domains for the forest. As such, the role holder must be online and available to perform domain additions and removals.

There is a good amount of guidance around Active Directory forests published on the internet. Below are some of the recommended practices surrounding forests:

- Always start with a single forest. Then, if you have requirements that cannot be met with a single forest implementation, begin adding forests as necessary. Better yet, go back and validate the requirements first. Using multiple forests in a production environment is often unnecessary and adds management overheard and unneeded complexity. With a backend technology that everybody expects to be always running, you should opt for a simple implementation that is implemented and maintained based on good practices, as opposed to a multi-forest implementation with a large number of domain controllers. For many environments, a single production forest will meet or exceed requirements. Additionally, it is a good idea to have a second non-production forest to use for development, testing, and quality assurance.

- Avoid the empty forest root domain. Upon initial release of Active Directory, Microsoft recommended using an empty forest root domain which would form a security boundary for enterprise objects stored in the root domain such as the Enterprise Admins group. However, not long thereafter, the guidance changed and the empty forest root was no longer recommended by default. Administrators found that maintaining the empty forest root domain added to the administrative overhead of their environment without returning much value. Today, the latest thinking is forest reduction. Minimize the total number of forests.

- If using two-way forests trusts, consolidate forests. Each forest that you maintain requires administrative overhead. In addition, each forest increases the complexity of your environment which also makes it harder to secure, maintain, and recover. If you are using two-way trusts between forests, you should strongly consider consolidating forests because a two-way trust between forests is effectively a single forest with extra overhead.

Domains

A domain is the logical container that sits directly below the forest container. Previous to Active Directory, there was a Windows domain that was part of previous Windows Server versions. It had similar core functionality as an Active Directory domain but without a forest above it. Historically, the beginning of the domain as we know it goes back to X.400 which is a telecommunications standard first recommended in 1984! Each domain is contained in a single forest container. A domain houses other containers and objects below it. In the early days of Active Directory, the domain was originally defined as the security boundary. However, that definition has been updated and now the forest is defined as the security boundary. That was a key change that went unnoticed by some administrators.

From a scalability perspective, you can have a very large number of domains in a single forest, as follows:

- Windows 2000 Server. Upon initial release, Active Directory supported up to 800 domains in a single forest.
- Windows Server 2003 and later. Once you use the Windows Server 2003 forest functional level or a higher level, a single forest can support up to 1,200 domains.

Several components work together in a domain. A domain includes the following components:

- Schema
- Global catalog
- Replication service
- Operations master roles

The schema, defined earlier in the Forest section, defines objects that are used in a domain. These can be both physical and logical objects. For example, a physical computer is represented by a computer account object, while a subnet is represented by a subnet object. Objects have many attributes. Object attributes define the properties, limits, and format of the objects. Attributes can be multi-valued, strings, integers, Boolean (true or false), or many other types. The specific attributes that an object has is defined by the schema. A global catalog server stores information about every object within a domain. Administrators and users query a global catalog server to find information about objects. For example, if an administrator needs to look up information about a user account, including address, phone number, and office location, he would query the global catalog server to retrieve the information.

The operations master roles, also known as flexible single master operations (FSMO) roles, perform specific tasks within a domain. The five FSMO roles are:

- Schema Master
- Domain naming Master
- Infrastructure Master
- Relative ID (RID) Master
- PDC Emulator

In every forest, there is a single Schema and Domain naming Master which are discussed in the Forest section of this e-book. In each domain, there is 1 Infrastructure Master, 1 RID Master, and 1 PDC Emulator. At any given time, there can only be one DC performing the functions of each role. Therefore, a single DC could be running all five FSMO roles, however, there can be no more than five servers in a single-domain environment that run the roles. For additional domains, each domain will contain its own Infrastructure Master, RID Master, and PDC Emulator.

The RID Master provisions RIDs to each DC in a domain. New objects in a domain, such as a user or computer object, receive a unique security identifier (SID). The SID includes a domain identifier, which is unique to each domain, and a specific RID for each object. Combining the two ensures that every object in the domain has a unique identifier, but contains both the domain SID and the RID. The PDC Emulator controls authentication within a domain, whether Kerberos v5 or NTLM. When a user changes their password, the change is processed by the PDC Emulator. Finally, the Infrastructure Master synchronizes objects with the global catalog servers. The infrastructure Master will compare its data to a global catalog server's data and receive the data not found in its database from the global catalog server. If all DCs in a domain are also global catalog servers, then all DCs will have up-to-date information, assuming that replication is functional. In such a scenario, the location of the Infrastructure Master role is irrelevant since it doesn't have any real work to do.

Trusts

A trust is a relationship between forest and/or domains. In a forest, all of the domains trust each because a two-way transitive trust is created when each domain is added. This allows authentication to pass through from one domain to any other domain in the same forest. You can create trusts outside of the forest too with other AD DS forests and domains or Kerberos v5 realms. Back in the days of Windows NT 4.0, there wasn't a forest or a hierarchical structure. If you had multiple domains, you had to manually create trusts between them. With Active Directory, you automatically have two-way transitive trusts between domains in the same forest. Back with Windows NT 4.0, you had to use NetBIOS to establish trusts too! Luckily, things have come a long way and now we've got additional trust functionality, especially around securing trusts with selective authentication and SID filtering. Each trust in a domain is stored as a *trustedDomain* object (TDO) in the System container. Thus, to find and list all of the trusts and trust types in a domain named contoso.com, run the Windows PowerShell command:

```
Get-ADObject -SearchBase "cn=system,dc=contoso,dc=com" -
Filter * -Properties trustType | where {$_.objectClass -eq
"trustedDomain"} | select Name,trustType
```

There are 4 valid values for the *trustType* attribute. However, only the value 1 (indicating a trust with an NT domain) and the value 2 (indicating a trust with an Active Directory domain) are common. There is a lot of other good information about trusts stored in the *trustedDomain* object. In a domain named contoso.com, run the Windows PowerShell command to look at all of the trust properties:

```
Get-ADObject -SearchBase "cn=system,dc=contoso,dc=com" -Filter
* -Properties * | where {$_.objectClass -eq "trustedDomain"} |
FL
```

You can also view many of the core properties of a trust by running the *Get-ADTrust –Filter ** command. The table below shows the trust properties and a description of each property.

Trust property	Property description
Direction	Valid values are bidirectional, inbound, or outbound. Note that the direction is relative to the domain in which you are running the query.
DisallowTransivity	I think this is a Microsoft typo as it really should be *"DisallowTransitivity"*. This can be set to True or False based on whether the trust disallows tranitivity.
DistinguishedName	The DN of the trusted domain object.
ForestTransitive	This is set to True when a forest trust is transitive and False when a forest trust is non-transitive.
IntraForest	This is set to True when a trust is between domains in the same forest or set to False when a trust is between domains in different forests.
IsTreeParent	Valid values are True and False.
IsTreeRoot	Valid values are True and False.
Name	The name of the domain that is part of the trust, not the domain where the query is run.
ObjectClass	This is set to trustedDomain for trusts.
ObjectGUID	Globally unique identifier for the trust. An example is *de207451-51ed-44cd-4248- 85ad9fcb2d50*.
SelectiveAuthentication	Set to True if the trust is configured for selective authentication or False if it isn't.
SIDFilteringForestAware	Set to True if a forest trust is configured for selective authentication.
SIDFilteringQuarantined	Set to True when SID filtering with quarantining is used for a trust. Used for external trusts only.

Source	Set to the DN of the trust root. In a forest trust, the DN of the root domain of the forest is the source.
Target	Set to the domain name of the other side of the trust.
TGTDelegation	Set to True if Kerberos full delegation is enabled on outbound forest trusts. Default is False.
TrustAttributes	Set to a numerical value indicating the trust configuration.
TrustedPolicy	Undocumented.
TrustingPolicy	Undocumented.
TrustType	Set to Uplevel for trusts with Active Directory forests and domains, DownLevel for trusts pre-Active Directory domains such as NT 4 domains, Kerberos realm for trusts with Unix/Linux realms.
UplevelOnly	Set to True if only Windows 2000 and later operating systems can use the trust link.
UsesAESKeys	Set to True for realm trusts that use AES encryption keys.
UsesRC4Encryption	Set to True for realm trusts that use RC4 encryption keys.

From a scalability perspective, there are a couple of things about trusts that you should be aware of:

- Maximum number of trusts for Kerberos authentication. If a client in a trusted domain attempts to access a resource in a trusting domain, the client can't authenticate if the truth path has more than 10 trust links. In environments with a large number of trusts and long trust paths, you should implement shortcut trusts to improve performance and ensure Kerberos authentication functionality.
- Performance deteriorates after 2,400 trusts. In really large and complex environments, you may have an enormous number of trusts. After you reach 2,400 trusts, any additional trusts added to your environment could significantly impact performance over the trusts, especially related to authentication.

V. Group Policy

Group Policy provides a method of centralizing configuration settings and management of operating systems, computer settings, and user settings in an environment. While these settings are managed by using Group Policy Objects (GPOs), GPOs cannot be applied directly to user or computer objects. A GPO must be applied to a domain, site, or organizational unit. By default, all objects within the container that the GPO has been applied to will receive the GPO settings. Child objects and containers will also receive the configured settings through inheritance, unless inheritance blocking is configured. Blocking inheritance complicates configurations and can cause unexpected results. Additionally, GPOs may be set to be enforced which ensures that GPOs will always be applied, regardless of inheritance settings. The enforced setting should also be used with caution.

Group Policy processes policy settings in the following order:

1. Local Group Policy
2. Site-linked policies
3. Domain-linked policies
4. OU-linked policies

Any policy that is configured by two or more GPOs will be overwritten or modified by the last GPO that is processed. For example, if a site policy is applied that modifies system settings, and an OU policy modifies the same system settings, then the OU policy will take precedent because it is processed last. Additionally, if a policy is enforced, the settings that are defined by that specific policy cannot be overwritten by a subsequent GPO, even if the other GPO is processed last.

GPOs that modify computer settings are applied at computer startup. Settings that are for users are applied at the time that the user logs on. By default, GPOs are processed synchronously. Synchronous processing ensures that all settings are applied before the user completes the log on process. Alternatively, asynchronous processing can be configured, to allow multiple operations to be performed. However, asynchronous processing can cause undesired effects. For example, if a policy is configured to remove Start Menu options for a user, the user could log on (and have access to the start menu) before the policy is applied. By default, GPOs are also reapplied every 90 minutes, with a randomized offset of up to 30 minutes. For domain controllers, the policies are refreshed every 5 minutes. Both of these refresh settings can be configured by using Group Policy.

If you do not wish to wait for a policy to refresh automatically, the *gpupdate.exe* command can be used locally, or with switches, remotely. The gpupdate.exe command processes the policies and applies only the settings that have been changed since the last refresh. Additionally, other command switches can be used to force applying all settings, specifying only user or computer settings, logging off, or restarting a computer after applying the settings.

To troubleshoot or view the applied policies, the *gpresult.exe* command can be used. The *gpresult.exe* command also has switches, allowing you to view the final applied policy in HTML format. The gpresult.exe command can also be ran remotely against target computers by name or by IP address. You can also specify specific user accounts to see the settings that would be applied if that user were to log on.

VI. Inside the AD DS Database

The Active Directory database is made up of a single file named *ntds.dit*. By default, it is stored in the *%SYSTEMROOT%\NTDS* folder. The folder also contains the following related files:

- Edb.chk. This file is a checkpoint file. Checkpoint files are commonly used in a transactional database system to keep track of which log file entries have been committed to the database. This is useful during a system crash to avoid data loss.
- Edb.log. There are typically multiple log files starting with *"edb"* such as *edb0013A.log* and *edb0013B.log*. Additionally, there is the *edb.log* file which is the active log file. These logs are the transaction logs used to record changes made in AD DS. All changes are first written to a transaction log and eventually make their way into the database a short time later.
- Temp.edb. As the name implies, this file is a temporary file used to track transactions that are taking place. It is also used when you run a database compaction job.
- Res1.log and res2.log or edbres00001.jrs and edbres00002.jrs. These log files are each 10MB in space and used in a situation where you are critically low on disk space on the system volume. In older versions of Windows Server, the *res1.log* and *res2.log* files are used. Since Windows Server 2008, the *"edbres"* naming is used, along with a new file extension of *.jrs*.

The Active Directory database is based on Microsoft's Joint Engine Technology (JET) which is a database engine that was developed in 1992. Microsoft Access is also based on the JET technology. Over the years, there have been rumors that Active Directory's database would be moved over to SQL Server (similar to rumors for Microsoft Exchange) but so far, that doesn't seem likely. I've heard third-hand that SQL was tested as the AD DS database engine but that performance issues prevented it from becoming the database standard. Because AD DS is a single use database, it can effectively run on JET technology (whereas JET technology may not be a good fit for the majority of transactional database needs which often have multiple uses). Microsoft chose to use the Indexed Sequential Access Method (ISAM) model for indexing data in the AD DS database. To work with the data, including transferring data in and out of the database, the Extensible Storage Engine (ESE) is used. ESE helps to maintain a consistent, and therefore optimal, database, especially in the event of a system crash. ESE is sometimes called JET Blue and is used by other technologies besides Active Directory including Microsoft Exchange, Windows Server's BranchCache, and Microsoft's Desktop Search. The database technologies for Active Directory have been around a long time. Each technology, by itself, could account for several pages of text to dive into how they work.

If you are interested in learning more, have a look at the following articles:

- Extensible Storage Engine Architecture at https://technet.microsoft.com/en- us/library/aa998171(v=exchg.65).aspx
- How the Active Directory data store really works (inside NTDS.dit) – a blog by Christoffer Andersson at https://technet.microsoft.com/en- us/library/aa998171(v=exchg.65).aspx

VII. Replication

Active Directory replication is the method of transferring and updating Active Directory objects from one DC to another DC. The connections between DCs are built based on their locations within a forest and site. Each site in Active Directory contains one or more subnets, which identify the range of IP addresses associated with the site. By mapping the IP address of a DC to a subnet, Active Directory knows which DCs are in which site. Connections are configured between sites to ensure that Active Directory objects are replicated between sites. Active Directory replication relies on the following technologies to operate successfully:

- DNS
- Remote procedure call (RPC)
- SMTP (optional)
- Kerberos
- LDAP

There are four main components of replication in Active Directory:

- Multimaster replication. Multimaster replication, compared to single-master replication as used in Windows NT 4.0, ensures that each domain controller can receive updates for objects for which it is authoritative. This provides fault tolerance within an Active Directory environment.
- Pull replication. Pull replication ensures that DCs request object changes instead of changes being pushed (especially unnecessarily). Pulling slightly reduces replication traffic between DCs.
- Store-and-forward replication. Store-and-forward replication ensures that every DC communicates with a subset of DCs to transfer the object changes that have occurred. With store- and-forward, every DC would communicate with every other DC, which is inefficient. Store-and- forward replication balances the replication load among the DCs within an Active Directory environment.
- State-based replication. State-based replication ensures that each DC tracks the state of replication updates which eliminates conflicts and unnecessary replication.

Replication is managed by the Knowledge Consistency Checker (KCC). The KCC manages replication between DCs in a single site by using automatically created connections. The KCC reads configuration data and reads and writes connection objects for DCs. The KCC only uses RPC to communicate with the directory service.

Intrasite replication does not use compression and changes are sent to DCs immediately. However, intersite replication relies on user-defined links that must be created. The KCC uses these links to create a topology so that replication is managed across the site-to-site links. Site connections can be controlled on a schedule and the replication data is compressed to minimize bandwidth usage. The default replication schedule for site-to-site connections is 180 minutes which is usually way too long for the vast majority of organization. This can be configured to as low as 15 minutes in the GUI, and even faster by modifying the registry. A replication packet size is calculated based on the amount of RAM in the DC. By default, the packet size limits are 1/100th the size of RAM, with a minimum of 1 MB and a maximum of 10 MB. Additionally, the maximum number of objects in a packet is 1/1,000,000th the size of the system RAM, with a minimum of 100 objects, and a maximum of 1,000 objects. Therefore, in modern servers that have more than 1 GB or RAM, replication packet sizes will either contain up to 10 MB of data or up to 1,000 objects. .The maximum packet size and object limit can be configured by modifying the registry in the following location:

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\NTDS\Parameters

The following are components the primary replication components:

- Knowledge Consistency Checker (KCC). The KCC is a process that runs on each DC and communicates directly with Ntdsa.dll to read and write replication objects.
- Directory System Agent (DSA). The DSA is a directory service component that runs as Ntdsa.dll on each DC. It provides an interface for services and processes to read the directory database.
- Extensible Storage Engine (ESE). The ESE manages directory database records, which may contain one or more columns.
- Remote Procedure Call (RPC). Directory replication is communicated by using the RPC protocol. RPC is a communication protocol that allows developers to execute code on a local or remote system without having to develop specific code for remote execution. The KCC also uses RPC to communicate with DCs to request information when building a replication topology.
- Intersite Topology Generator (ISTG). The ISTG manages the intersite inbound replication connection objects for a specific site. There is one ISTG server in each site. By default, the first DC in each site is the ISTG. To find the ISTG in a site named HQ in a domain named tailspintoys.com, you can run the Windows PowerShell command:

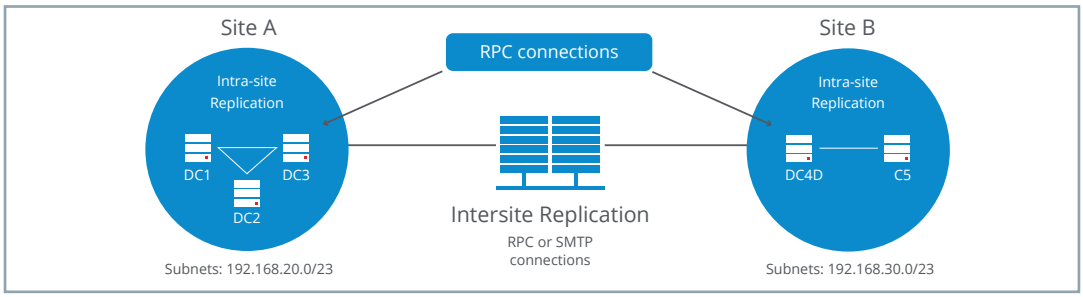
```
Get- ADObject -Identity "cn=NTDS Site Settings,
cn=HQ,cn=sites,cn=configuration,dc=tailspintoys,dc=com" -Properties
interSiteTopologyGenerator | Select interSiteTopologyGenerator
```

The Active Directory objects that are used by the KCC and its components include:

- Sites. Sites are Active Directory objects in the site class, which correspond to the subnets in a given site.
- Subnets. Subnet objects are in the subnet class, and define the network IP subnet that is corresponded with a site.
- Servers. A server object, in the server class, represents server computers, including DCs. Server objects are treated as security principals which are stored in a separate directory partition and have separate globally unique identifiers (GUIDs).
- NTDS Settings. NTDS Setting objects are in the nTDSDSA class, and represent an instance of Active Directory on a specific DC.
- Connections. Connection objects are in the nTDSConnection class, and define a one-way, inbound route from a source DC to the DC that is storing the connection object.
- Site Links. Site Link objects are in the siteLink class, and identify the protocol and schedule to replicate data between two or more sites.
- NTDS Site Settings. NTDS Site Setting objects are in the nTDSSiteSettings class, and identify site-wide settings for Active Directory. There is only one NTDS Site Settings object per site in the Sites container.
- Cross-reference. Cross-reference objects are in the crossRef class, and store the location of Active Directory partitions in the Partitions container.

The diagram below shows a typical two-site Active Directory environment with some of the replication components.

Figure 7.1
A typical two-site
Active Directory environment



Beginning with Windows PowerShell in Windows Server 2012, there are 25 cmdlets to specifically manage Active Directory replication. These cmdlets offer functionality such as viewing replication information, configuring sites, managing site links, and forcing replication to occur. The RepAdmin.exe command line tool is also available to provide information and configure Active Directory replication. Another replication tool is the Active Directory Replication Status Tool. It is available at http://www.microsoft.com/en-us/download/details.aspx?id=30005. You can use it to analyze and troubleshoot Active Directory replication issues.

VIII. DNS and DHCP

AD DS provides a built-in method of storing and replicating DNS records by using Active Directory-integrated DNS zones. All of the records and zone data stored within the zone are replicated to other DNS servers by using the native AD DS replication service. Each DC stores a writable copy of the DNS zone data for namespaces for which they are authoritative. Active Directory-integrated zones also provide the ability to use secure dynamic updates, which supports controlling which computers may make updates and prevents unauthorized changes from being made.

DNS zone data is stored in an application directory partition. A forest-wide partition named *ForestDnsZones* is used for the zone data. For each AD DS domain, a domain partition is created named *DomainDnsZones*. Typically, DNS implementations are used with a contiguous namespace. For example, the Fully Qualified Domain Name (FQDN) of an AD DS domain might be corp.contoso.com, and the FQDN of a client in that domain would be client.corp.contoso.com. However, AD DS and Active Directory-integrated DNS zones support disjoint namespaces. In such a scenario, the FQDN of the AD DS domain might be na.corp.contoso.com, while a client FQDN could be client.corp.contoso.com. Notice that the *"na"* portion of the FQDN is not present in the client FQDN. There are several requirements and considerations when using a disjoint namespace. For more information, see https://technet.microsoft.com/en-us/library/cc731125%28v=ws.10%29.aspx.

AD DS requires DNS to function, and uses three specific components for the AD DS infrastructure:

- Domain controller locator. The Locator is implemented in the Net Logon service and provides the names of DCs in an AD DS environment. The Locator uses address (A) and service (SRV) DNS resource records to identify DCs in an AD DS environment.

- Active Directory domain names in DNS. The AD DS domain names in DNS are the FQDN that we discussed earlier.

- Active Directory DNS objects. While DNS domains and AD DS domains typically have the same name, they are two separate objects with different roles. DNS stores zones and zone data required by AD DS and responds to DNS queries from clients. AD DS stores object names and object records and uses LDAP queries to retrieve or modify data. DNS zones that are stored in AD DS have a container object that is in the dnsZone class. The dnsZone object has a DNS node, which uses the dnsNode class. Each unique name in a DNS zone has a unique dnsNode object. For AD DS, this also includes individual functions. Therefore, one DC may have multiple roles, such as being a global catalog server, which is indicated in the dnsNode object.

As mentioned earlier, DCs are identified by the SRV records in a DNS zone. Components of AD DS are stored in DNS using the following format in the _msdcs subdomain: *_Service.Protocol.DcType._msdsc.DnsDomainName*. For example, the Lightweight Directory Access Protocol (LDAP) service of the Primary Domain Controller (PDC) in the contoso.com AD DS domain would be *_ldap._tcp.pdc.contoso.com*. The service and protocol strings use underscores (_) as a prefix to avoid potential collisions with existing resources or records in the namespace. The Net Logon service requires 17 different SRV records to perform lookups. A full list of SRV records can be found at https://technet.microsoft.com/en-us/library/cc759550%28v=ws.10%29.aspx.

In addition to the SRV records, the Net Logon service also requires two A records for clients that may not be SRV- aware. This includes a record for the DnsDomainName, and a record for *gc._msdsc.DnsForestName*. This enables non-SRV-aware clients to look up a domain controller or global catalog server by using an A record.

DNS is susceptible to security threats, such as foot printing, denial-of-service attacks, data modification, and redirection. To mitigate these threats, DNS zones can be secured by using secure dynamic updates, restricting zone transfers, plus implementing zone delegation and DNS Security Extensions (DNSSEC). By using secure dynamic updates, computers will be authenticated through Active Directory, and security settings will be applied when performing a zone transfer. Additionally, zone transfers can also be restricted to specific IP addresses within the network. Zone delegation can be approached by using two methods. First, is to limit DNS changes to a single team or entity, with all changes tracked and approved. This method limits the amount of people making changes, but allows for a single point of failure. Secondly, zones can be delegated to individuals who will be managing each component of a network or domain. While changes may still need to be approved and tracked, this spreads out risk among multiple people, and may limit damage if only one component becomes compromised. DNSSEC validates DNS responses by providing origin authority, data integrity, and authenticated denial of existence. Windows Server 2012's implementation of DNSSEC meets the standards for RFC 4033, 4034, and 4035.

There are six resource record types that are used specifically with DNSSEC:

- Resource record signature (RRSIG)
- Next Secure (NSEC)
- Next Secure 3 (NSEC3)
- Next Secure 3 Parameter (NSEC3PARAM)
- DNS Key (DNSKEY)
- Delegation Signer (DS)

For more information on each of the record types and their use, see https://technet.microsoft.com/en- us/library/jj200221.aspx.

DHCP is another network service that is used by Windows Server. In an AD DS environment, DHCP servers must be authorized before they can lease IP addresses to clients on a network. DHCP servers are authorized by their IP addresses, and will be checked against AD DS to verify that it is authorized to lease IP addresses. If an unauthorized DHCP server detects an authorized DHCP server, the unauthorized DHCP server will stop leasing addresses to clients. In an AD DS environment, the DHCP service must be installed on a server that is a member of the domain, or it cannot be authorized. Installing and running the DHCP service on a stand-alone server is supported, but must be on a separate network or VLAN than any authorized DHCP server. To authorize a DHCP server, the administrator must be a member of the Enterprise Admins built-in security group. However, the right to authorize DHCP server may be delegated to other administrators within the domain. To authorize a DHCP by using it's FQDN, the FQDN must not exceed 64 characters. If the FQDN is more than 64 characters, it must be authorized by using an IP address.

DHCP can be integrated with DNS to provide dynamic updates to pointer (PTR) and A records in a DNS zone. This ability enables a DHCP server to be a proxy for any DHCP client running an operating system that does not automatically update their DNS registration.

In Windows Server 2012, DHCP can be configured with DHCP failover. DHCP failover enables the DHCP server to be configured in hot standby mode, which provides redundancy, or load balance mode, which allocates client leases across two DHCP servers. The mode can be changed at any time, but a DHCP scope only supports using one mode at a time. IPv4 addresses that have been leased or reserved, including the options and settings for each scope, are shared by two DHCP servers. A single DHCP server supports up to 31 failover relationships. Failover relationships can be reused for additional scopes to avoid exceeding the limit.

When using DHCP hot standby mode, two servers operate the DHCP service, however one server provides and responds to all DHCP requests. The secondary server will only provide leases if the primary server is unreachable. To provide leases, a percentage of the IP address pool must be reserved for use by the secondary server. By default, this is set to 5%. If the secondary server leases all of the IP addresses in the reserved space, it will not issue additional IP addresses from the primary server's scope. Existing leases will be renewed if requested by a DHCP client. Additionally, when the secondary server leases an IP address, the lease time is the maximum client lead time (MCLT) duration, not the full scope lease time. After the MCLT time has expired, the secondary server will use the entire address pool in the scope, assuming that the primary server has resumed.

Using DHCP in load balancing mode is the default method of deployment. In this method, two servers provide the DHCP services simultaneously for a DHCP scope. The load balancing method is defined by a percentage of IP addresses on each server, and by default is split 50:50. This ratio or percentage can be configured to any amount between the two servers. The DHCP servers load balance based on a hash of the requesting client's MAC address. The MAC address thus determines which DHCP server will respond to a client's DHCP request. Similar to hot standby mode, if the partner server is unavailable, the remaining server will lease and renew IP addresses for the MCLT duration. After the MCLT time has expired, if the partner server is not online, the remaining server will lease addresses from the entire IP address pool for the scope.

IX. Security

Security is a huge topic because it encompasses so many areas. Even in a specific technology like AD DS, security is a huge topic. For this e-book, we will focus on 3 specific areas of security in AD DS:

- Securing LDAP traffic with SSL/TLS
- Modifying the access control list (ACL) on administrative accounts
- Enabling strong authentication in a domain

The first step to securing LDAP traffic with SSL/TLS is to install AD CS. It is a good practice is to install the role on a member server. Before you begin implementing a PKI, you should spend ample time gathering information, designing the PKI, and planning. Once you have an operational PKI, you can issue a certificate to each DC.

It is also possible to use a third-party Certification Authority (CA) to secure LDAP communication. The certificate must not have strong private key protection enabled. Additionally, the DC FQDN must appear in either the Common Name of the Subject field, or as a DNS entry in the Subject Alternative Name extension. They Enhanced Key Usage must also include Server Authentication as an option, object identifier (OID) 1.3.6.1.5.5.7.3.1 By enabling LDAP over SSL, LDAP communication can occur on both ports 389 (LDAP) and 636 (LDAP/SSL). Global catalog requests performed with SSL used port 3269.

The *certreq.exe* command can be used to create a new certificate request. To create a request, run the following command:

```
certreq -new request.inf request.req
```

The *request.inf* file must contain the DC FQDN, and also provides information on the key length and request time. For a full example of the *request.inf* file, see http://support.microsoft.com/kb/321051. The *request.req* file will be created, and must be submitted to the CA. After receiving the certificate, process the request with the *certreq.exe* command.

```
certreq -accept certnew.cer
```

To verify that SSL has been enabled successful, the Active Directory Administration Tool (*ldp.exe*) can be used to manually connect to the AD DS environment. Launch the *ldp.exe* tool, and perform a new connection. In the port field, specify 636. If the RootDSE information is displayed, LDAP over SSL has been configured successfully.

Another way to secure AD DS is to modify the ACL of user accounts that have administrative privileges. User accounts in administrative groups have their ACLs replaced every hour by a process on the PDC Emulator. This process compares the ACLs on each account in the administrative groups and the group itself to the *AdminSDHolder* object. In a domain named contoso.com, the object is located at *CN=AdminSDHolder,CN=System, DC=Contoso,DC=Com*. Any difference between the account and the *AdminSDHolder* object is then replaced on the account. The following groups and members of these groups are checked by the *AdminSDHolder* process:

- Administrators
- Backup Operators
- Print Operators
- Account Operators
- Domain Admins
- Schema Admins
- Cert Publishers
- Enterprise Admins
- Server Operators

Additionally, the *Administrator* and *krbtgt* user accounts are also checked independently regardless of group membership. Accounts that are protected by the *AdminSDHolder* process can be identified by the *adminCount* attribute of the account. If the account is being protected, this attribute will be set to 1.

To modify the ACL of an account being protected by the *AdminSDHolder* process, three steps must be taken to verify that the ACL is not replaced.

1. Remove the account object from all protected groups.
2. Set the adminCount attribute to 0.
3. Enable inheritance on the account object.

Note that simply changing the *adminCount* to 0 without removing the object from the protected groups will not stop the process from replacing the ACLs. Removing the object from the groups without modifying the *adminCount* attribute doesn't fix the situation.

Another solution is to modify the *AdminSDHolder* object to include the ACLs that you want to apply to the user accounts. By including the additional access control entries in the *AdminSDHolder* object, they will also be included when the ACLs of each account object is compared. This is useful for modifying the permissions of all administrative user account objects.

Another area of AD DS security is to enable strong domain authentication in your environment. This will ensure that users can authenticate to Active Directory by using only strong authentication protocols. By default, Windows Server 2012 accepts authentication requests that use NTLM and NTLMv2, although it responds by replying with NTLMv2 only. This setting can be restricted by modifying the *"Network Security: LAM Manager Authentication Level"* setting in group policy. This setting is applied to DCs, and can be set to *"Send NTLMv2 response only. Refuse LM and NTLM."* By configuring this setting, DCs will not respond unless they receive an NTLMv2 request.

X. Auditing

Prior to Windows Server 2008 R2 and Windows 7, auditing in Windows was a fairly simple topic. You navigated to the auditing policies in a GPO and enabled auditing and chose Success, Failure, or both. There were a number of articles on the internet describing each of the auditing policies and many administrators quickly avoided anything that didn't provide much value. Below is a screen capture showing the available audit policy settings.

Figure 10.1
Audit policy settings

Audit account logon events	Not Defined
Audit account management	Not Defined
Audit directory service access	Not Defined
Audit logon events	Not Defined
Audit object access	Not Defined
Audit policy change	Not Defined
Audit privilege use	Not Defined
Audit process tracking	Not Defined
Audit system events	Not Defined

In Windows Server 2008 R2, a new feature was introduced to allow for advanced auditing policies in Group Policy. Officially, 53 new settings were made available to replace the original 9 policy settings shown above. A little known fact is that these 53 new settings were actually available in Windows Server 2008. However, you had to use logon scripts and *auditpol.exe* to take advantage of the new settings. Thus, most administrators didn't.

One common area of confusion is the apparent overlap of the original 9 policy settings (hereafter called the basic audit policy settings) and the advanced audit policy settings. There actually isn't any overlap though. Let's examine why by looking at the account management auditing. With basic audit policy settings, you can enable the *"Audit account management"* policy for Success and Failure. With advanced audit policy, you can enable auditing for application group management, computer account management, distribution group management, account management events, security group management, and user account management. Enabling the *"Audit account management"* basic audit policy setting is the same as enabling auditing in the 6 subcategories available in an advanced audit policy. Neither provide more data. But, as many administrators have realized, generating too much audit data can be worse than not generating any audit data because of the massive volume of audit data that can be generated. Administrators have been struggling with audit data for a long time. Some of the common struggles are:

- Windows event logs fill up. Windows event logs can be configured in a number of different ways. You can set a maximum log size and delete old event as needed. You can archive a log when it fills up and then start a new log. Or, you can configure the logs not to overwrite events and require manual intervention. You can even shut the server down if you can't write to the Security event log. Administrators often can't afford for new events not to be written or for servers to shut down when a log fills up. Thus, overwriting events or archiving are the most common settings. But this leads to administrative overhead: monitor event log sizes, monitor disk space, move archived logs off of server, manage archived logs, and figuring out a way to search through all of the data.

- Disk volumes run out of space. I still find it amusing that in 2015, disk space is still the major source of server downtime in many companies. Log files are a common problem whether from auditing or from applications such as IIS. I have heard from several organizations that experienced an outage and the root cause was the system volume running out of space due to Windows event log archiving.

- Inability to locate specific audit data. When you generate a massive amount of data, every data management task, even normally simple tasks, become complex and time consuming. Tasks such as compressing the files, copying the files to another location over the network, or searching through the files for a specific key term become problematic and incredibly time consuming. Administrators are turning toward third-party solutions to help.

- Inability to use audit data in a timely fashion. Imagine a call from the security team about an employee that has potentially viewed confidential HR data. They ask you to pull up audit data for the user over the last few weeks. Not a big deal if you have 1GB worth of audit data. But when you have 500GB worth of audit data, it suddenly becomes your full time job for a few weeks.

The advanced audit policy settings can help. By offering more granular auditing options, you can greatly reduce the amount of data gathered. This minimizes the struggles mentioned above. But, there is a big investment in time to switch over to the advanced audit policy settings. For some organizations, that investment will pay for itself and more.

Let's take a quick look at how this impacts the number of events captured. In this first example, in a Windows Server 2003 R2 domain named adatum.com, I set basic audit settings so that only account management success events are being audited, as shown below. There isn't any significance to the operating system version because the basic audit settings shown below are available in every version of Windows Server since Windows 2000 Server.

Figure 10.2
Basic audit settings

Policy	Policy Setting
Audit account logon events	Not Defined
Audit account management	Not Defined
Audit directory service access	Not Defined
Audit logon events	Not Defined
Audit object access	Not Defined
Audit policy change	Success, Failure
Audit privilege use	Not Defined
Audit process tracking	Not Defined
Audit system events	Not Defined

Then, I created a new computer object and refreshed the Security event log. Below are the entries related to the new computer object creation.

Figure 10.3
Entries related to the new computer object creation

Success Audit	3/6/2015	3:42:16 PM	Security	Account Management	626	Administrator	A-DC-01
Success Audit	3/6/2015	3:42:16 PM	Security	Account Management	646	Administrator	A-DC-01
Success Audit	3/6/2015	3:42:16 PM	Security	Account Management	646	Administrator	A-DC-01
Success Audit	3/6/2015	3:42:16 PM	Security	Account Management	628	Administrator	A-DC-01
Success Audit	3/6/2015	3:42:16 PM	Security	Account Management	645	Administrator	A-DC-01

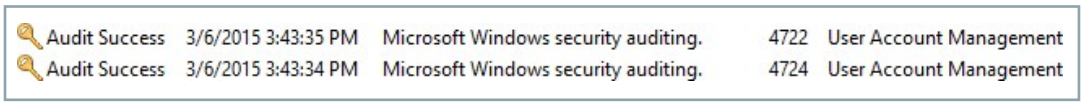
There are 5 events. Next, in a Windows Server 2012 R2 domain named contoso.com, I created an advanced audit policy based on wanting to audit only successful user account management events, as shown below.

Figure 10.4
An advanced audit policy

Audit Application Group Management	Not Configured
Audit Computer Account Management	Not Configured
Audit Distribution Group Management	Not Configured
Audit Other Account Management Events	Not Configured
Audit Security Group Management	Not Configured
Audit User Account Management	Success

Then, I created a new computer object and refreshed the Security event log. Below are the entries related to the new computer object creation.

Figure 10.5
Entries related to the new computer object creation



Audit Success	3/6/2015 3:43:35 PM	Microsoft Windows security auditing.	4722	User Account Management
Audit Success	3/6/2015 3:43:34 PM	Microsoft Windows security auditing.	4724	User Account Management

With the advanced auditing policy, only 2 events were logged. That's a big difference, especially when you think about how that would extrapolate once all auditing categories were configured. Note that not only would we save some time with the granular approach, but we will also save quite a bit more time later after the other auditing categories are configured. Now, you may be wondering why creating a computer account is showing up in user account auditing? That's because computer objects are part of the AD DS user class along with user objects! You may also be wondering why the event log IDs are different for each environment. This is unrelated to the auditing configuration and instead is because of a change in the event log IDs used in Windows Server 2008 and newer. Many of the event IDs changed between versions of Windows Server.

XI. User Unfriendly Log Data

As a side effect of auditing and logging, administrators have to deal with a ton of data and information. Often, not only is there an overload of information, but often the information isn't readily usable. Here are some of the pain points that administrators are dealing with when it comes to audit and log data:

- The information is unformatted. In this situation, information is captured but it is unformatted. In other words, everything looks the same and it can be a struggle to extract the key pieces of information that you need. In these situations, administrators struggle to work with the data – filtering, sorting, manipulating, and extracting are difficult without hours of administrative effort.

- The informatio is verbose. In this situation, there is too much information. For example, take an example of a user signing in to their computer in the morning. Instead of a single event log entry, you end up with several entries, each covering a part of the overall sign in process. Another example is a packet capture. You may be troubleshooting an authentication issue on a DC, but your packet capture contains tons of other irrelevant data. In some cases, administrators can turn to tools to create filters to minimize data capture. But that isn't nearly as effective for capturing Windows security related data on an ongoing basis.

- The information is uncorrelated. For example, imagine a scenario where you are troubleshooting authentication from a federated partner. You have information in the Windows event logs on the DC, Windows event logs on the AD FS servers, Windows event logs on the target web server, Windows event logs on the client computer, AD FS proxy logs, AD FS specific logs, and IIS logs. What's worse is that there isn't any correlation to the logs beyond a date and time stamp. The built-in tools in Windows don't provide a good event correlation solution for this problem so administrators are often using third-party tools to help.

- The information is ignored. After extended periods of time dealing with unformatted, verbose, and uncorrelated information, administrators often start ignoring logs, ignoring alerts, and ignoring important information that may be buried *"in there somewhere"*. This same phenomenon is often seen in monitoring projects when so many alerts are being generated that they go unanswered.

Part of the solution to this problem is to reduce the amount of information you capture. But that sometimes goes against what a security team wants a company to capture – everything. So there is a balancing act. You need to capture enough information to make it useful but without making it a burden. Administrators often turn to third- party tools to help come up with the right solution.

netwrix

Useful References

The following resources will help you improve Active Directory management and security:

Manage

Blog post | Top Seven Challenges with Active Directory

Blog post | Active Directory Management: Top 7 Common Mistakes

Blog post | Create AD Users in Bulk and Email Their Credentials

Blog post | How to Lock, Unlock, Enable and Disable AD accounts

Blog post | How to Create, Delete, Rename, Disable and Join Computers in AD

Blog post | How to Add and Remove AD Groups and Objects in Groups

Blog post | Managing OUs and Moving Their Objects

Best Practices | Active Directory Group Management Best Practices

Best Practices | Group Policy Best Practices

Monitor

How-to | How to Monitor User Logons in a Domain

How-to | How to Monitor Active Directory Group Membership Changes

How-to | How to Detect Changes to Organizational Units and Groups in Active Directory

How-to | How to Detect User Account Changes in Active Directory

How-to | How to Detect Password Changes in Active Directory

Secure

Podcast | Active Directory Security 101, Episode 1: Listen and Learn

Blog post | Privileged Access Management to Secure a Compromised AD Forest

Blog post | Securing Active Directory

Blog post | 3 Ways to Secure Service Accounts in Active Directory

Best Practices | Active Directory Delegated Permissions Best Practices

Get Complete Visibility into What's Going On in Your Active Directory and Group Policy

With Netwrix Auditor for Active Directory

See Netwrix Auditor in Action

Download Free Trial

netwrix

About the Author

Brian Svidergol is focused on Microsoft infrastructure and cloud-based solutions around Windows, Active Directory, Exchange, System Center, virtualization and MDOP. He holds numerous certifications, including MCITP, MCSE, RHEL3, VCP, NCIE-SAN, MCT, MCSA and Microsoft Certified Solutions Expert: Server Infrastructure. Brian is an author of Microsoft Official Curriculum (MOC) course 6426C — Configuring and Troubleshooting Identity and Access Solutions with Windows Server 2008 Active Directory. He has worked on Microsoft certification exam development and related training content for several years. He also co-authored the Active Directory Cookbook.

About Netwrix

Netwrix is a software company that enables information security and governance professionals to reclaim control over sensitive, regulated and business-critical data, regardless of where it resides. Over 10,000 organizations worldwide rely on Netwrix solutions to secure sensitive data, realize the full business value of enterprise content, pass compliance audits with less effort and expense, and increase the productivity of IT teams and knowledge workers.

Founded in 2006, Netwrix has earned more than 150 industry awards and been named to both the Inc. 5000 and Deloitte Technology Fast 500 lists of the fastest growing companies in the U.S.

For more information about Netwrix, visit https://www.netwrix.com/.

Contact Us

Corporate Headquarters

300 Spectrum Center

Drive Suite 200

Irvine, CA 92618

Phones

USA: 1-949-407-5125

Toll-free: 888-638-9749

EMEA: +44 (0) 203 588 3023

netwrix.com/social