Global Information Assurance Certification Paper

Designing and Operating a Secure Active Directory Infrastructure
Version 1.4b

Submitted by
Dermot Murphy

SANS Institute GSEC
Option 1

Table of Contents

1. Abstract 3
2. Introduction to Active Directory 4
    2.1 What is Active Directory?
    2.2 Active Directory Features
    2.3 Active Directory Concepts
3. Risk Management in Active Directory 17
    3.1 Risks
    3.2 Threats
    3.3 Vulnerabilities
4. Secure Active Directory Design 23
    4.1 Forest Design
    4.2 Domain Design
    4.3 OU Design
    4.4 Site Design
5. Secure Active Directory Operation 43
    5.1 Auditing
    5.2 Monitoring
    5.3 Recovering from Attacks
6. Summary 48
7. References 49
    7.1 Book References
    7.2 Magazine References
    7.3 Internet References
    7.4 Related Links
Appendix A – SRV Record Format
Appendix B – The Active Directory Database
Appendix C – Group Types

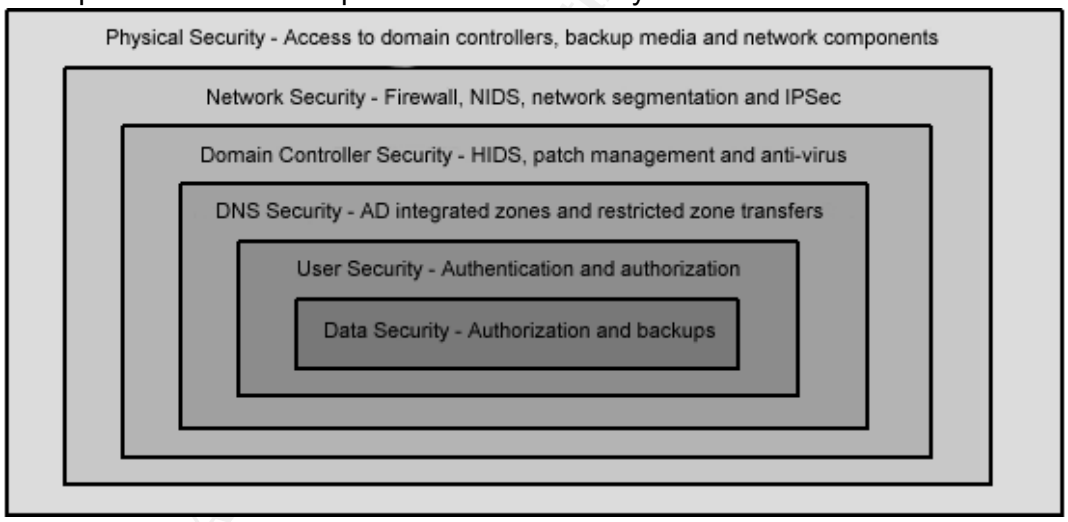
1 - Abstract

Microsoft designed Active Directory in response to Novell's Directory Service (NDS) as used with Netware and to replace the old flat domain model of Windows NT4 with a hierarchical one capable of meeting the needs of enterprise environments. They also needed to create a network operating system capable of meeting enterprise requirements – a market previously dominated by more scalable UNIX systems.

In doing so, Microsoft included an impressive set of security features capable of implementing identification, authentication, authorisation, delegation, public & private keys, security policies and auditing. It provides ways to implement key security principles such as "separation of privilege" through delegation (using Organizational Units) and "principle of least privilege" through access control lists, group membership and other authorization techniques.

This paper examines those features and investigates how considering them during the design phase of Active Directory can greatly improve reliability and reduce the cost of maintaining an Active Directory infrastructure. Although business structure cost & performance are key Active Directory design considerations, information security should be a design consideration especially if attempting to implement defence in depth (multiple layers of security) as shown in the diagram below.

Example of defence in depth for Active Directory…



We'll look at the security considerations throughout the various design stages and also how to maintain security in the daily operation of Active Directory. As long as the confidentiality, integrity and availability of data are considered during the design process then we can be confident that maintaining these goals on a day-to-day basis will be easier and cheaper.

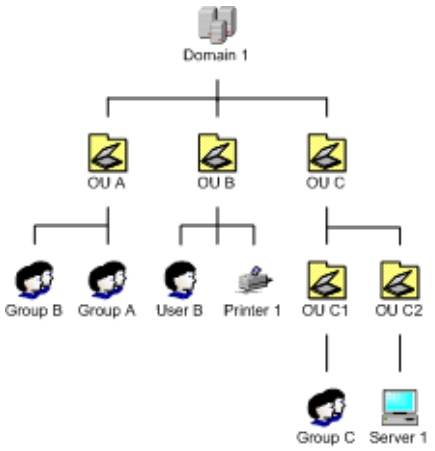
This paper does not attempt to cover preliminary design steps such as analyzing the existing business and technical environments. Nor does it attempt to provide a methodology for migrating NT4 domains to Windows 2000 Active Directory domains.

2. Introduction to Active Directory

2.1 What is Active Directory?

Before going any further, let's briefly talk about what Active Directory is and why we should be using it before finishing with a quick look at some of the key concepts behind Active Directory.

Here's an example of the logical hierarchical structure of Active Directory…



Active Directory domains can contain Organizational Units (OUs) which in turn can contain users, groups, computer, printers or any other resource. This hierarchical structure is often used to reflect organizational structure where OUs represent departments.

There are many ways to define Active Directory although to do so in one or two sentences is to offer a very abstract definition.

For example, we might describe Active Directory as a directory service which provides information about network resources to administrators, users and applications. This directory service is also responsible for making these resources easily available through authentication and authorization.

We could say that it is a hierarchical namespace which contains various objects representing users, computers, printers and file shares. This namespace is implemented using standard DNS namespace.

We could also say that Active Directory is a secure, distributed object-oriented database which stores user credentials and access control information required to utilize network resources. It replaces the old SAM databases used on NT4 domain controllers.

These definitions are all OK, but let's look at what Active Directory does from a security perspective.

Let's take confidentiality, integrity & availability of data as our three key information security requirements. To help meet these requirements we must have mechanisms which provide three main functions…

- Identification – allows users to uniquely identify themselves usually using a username.
- Authentication – allows users to prove their identity using a password, smart card or some form of biometric measurement.
- Authorization – allows or denies access to a resource based on privilege.

Active Directory services provide ways to implement these functions but it's ultimately up to the Administrative & Design staff to ensure that they're implemented effectively.

- Identification – unique security identifiers (SIDs) are provided to identify users, groups and computers to resources.
- Authentication – protocols such as Kerberos, LAN Manager (LM), NTLM and NTLMv2 are implemented to authenticate users.
- Authorization – groups, access control lists and user rights are used for determining the extent of access to network resources.

Beyond meeting these basic security requirements, Active Directory also helps implement key security principles such as "separation of privilege" through delegation (using Organizational Units) and "principle of least privilege" through access control lists, group membership and other authorization techniques.

We'll talk more about authentication in the section on domain design. We'll talk more about authorization in the section on OU design.

2.2 Active Directory Features

Now that we know what Active Directory is, why should we use it? Active Directory offers a simple and secure way for users and applications to access network resources. Users can find objects across the enterprise by searching for the object's attributes if the object name is not known. Administrators can use it do enforce security policies and control access to resources. Let's take a look at its features.

2.2.1 Security

Active Directory offers much in terms of security. Microsoft opted to use Kerberos for authentication in Active Directory. Kerberos is a system originally developed at MIT (http://web.mit.edu/kerberos) in the 1980s which uses keys as the basis for authentication. Although Microsoft added some if its own functionality to its implementation of Kerberos it is still compatible with any other system using Kerberos version 5 (RFC 1510 - http://www.ietf.org/rfc/rfc1510.txt).

Kerberos uses strongly encrypted keys so that users can authenticate across insecure network connections.

Although Kerberos is infinitely more secure than the NTLM or NTLM v2 authentication protocols, weak passwords are still a potential vulnerability. Applications such as KerbCrack or L0phtcrack version 4 can intercept Kerberos packets although this particular threat (and network sniffing in general) can be mitigated by using a fully switched network where every packet will be sent only to the destination computer.

Kerberos also offers mutual authentication where both client and server must authenticate to each other. This is useful when a client needs to authorize a service to perform some function on the client's behalf.

We'll cover Kerberos in some more detail in the section on domain design.

Active Directory uses public key infrastructure (PKI) technology. This allows the use of digital certificates which can be mapped to user accounts to further improve security during authentication. This technology also enables smart cards to be used as an alternative to the more traditional password based authentication method.

Digital certificates can be used for a number of other purposes including validating the authenticity of information replicated using SMTP and for handling LDAP requests over SSL.

Using discretionary access control lists (DACL), access can be defined not only on each object in the directory but also on each property of each object. For example, not only can you control who can create user accounts but you can also further control who can update personal user fields such as phone number, address, etc. It's discretionary in the sense that users with the appropriate permissions can also change the access control themselves. This is a key requirement (along with Organizational Units) when it comes to "separation of privilege" or delegation; you may want your help desk to be able to reset passwords but not look at specific user details.

Active Directory also stores security policies. A security policy can include account policies lockout policies, password policies and Kerberos policies. These security policies are implemented through Group Policy settings.

Active Directory supports auditing through the use of a second type of access control list called Security Access Control List (SACL). If an audit policy is being used then this list will control who gets audited. Audit policies can also be implemented through Group Policy.

2.2.2 Simplified Administration

Active Directory eases administrative tasks by providing ways (using Group Policy) to remotely deploy software and manage client & server configurations.

The idea of permissions inheritance also serves to further reduce administrative overhead. If access control is changed at a certain Organizational Unit or file system folder, that change will propagate throughout the OU or file structure automatically.

2.2.3 Scalability

One of the big problems with the flat Windows NT 4.0 domains was the limitation of being able to hold only 40,000 objects (21,000 users) in the SAM database. With Active Directory there is no such limitation.

According to Microsoft, there are no practical limits to the number of objects stored in Active Directory. The Active Directory database has been tested for up to 40 million objects. Performance tests show logon performance for a single LDAP client to be the same with 10,000 objects, 100,000 objects, and 1 million objects — that is, the directory service does not slow measurably when the size of the database increases.

2.2.4 Extensibility

Active Directory is a database which stores objects. Like any other database, Active Directory has a schema and this schema can be extended to include new objects which have not already been defined. This allows new applications to be integrated into Active Directory such as Microsoft Exchange.

2.2.5 Support for Open Standards

Apart from Kerberos and Public Key Infrastructure standards mentioned previously, Microsoft have included support for other well established industry standards in its implementation of Active Directory.

DNS is used by Active Directory as its domain naming service. It's also used by clients to locate Active Objects. For example, DNS SRV records are used to locate domain controllers. Active Directory will not function without DNS so DNS security is just as critical as Active Directory security especially when DNS servers are likely to be domain controllers as well.

Active Directory supports LDAP (Lightweight Directory Access Protocol) (RFC - http://www.ietf.org/rfc/rfc1777.txt) which is based on the X.500 directory access protocol and is used to provide low-level access from applications to Active Directory objects.

2.2.6 Support for standard name formats

Active Directory supports a number of naming standards for referencing its objects including…

- User Principle Names (UPN) which appear as username@domain,
- Universal Naming Convention (UNC) names which appear as \\domain\object
- HTTP URL (Uniform Resource Locator) names which appear as http://domain/object
- LDAP URL names which appear as ldap://domain/CN=common name,OU=organizational unit,DC=domain controller.

2.3 Active Directory Concepts

Compared with previous Microsoft operating systems, Active Directory is quite complex and introduces some new ideas which we'll cover here and briefly discuss any security implications.

2.3.1 FSMO Roles (Flexible Single Master Operations)

Active Directory has two modes of operation, "mixed mode" and "native mode". A mixed mode domain means that NT4 backup domain controllers are still being used in the domain. We would need NT4 domain controllers if there are clients or servers which require NTLM authentication (NT4 or earlier). An Active Directory domain operates in mixed mode by default. A native mode domain simply means that all domain controllers are Windows 2000. Once you change your mode of operation to native mode, you can not change back to mixed mode at a later stage.

As with NT4, Active Directory uses domain controllers to store its account information. There are some differences though. With NT4, there could only be one primary domain controller (PDC) which contained the only writeable accounts database. All of the other backup domain controllers (BDC) contained a read only copy of the database. While a BDC could handle authentication requests, they could not make any changes to the database. That role was reserved exclusively for the PDC. With Active Directory, every domain controller can handle updates to the Active Directory database. This is what is known as multi-master architecture. You'll sometimes hear that Active Directory eliminates the need for primary and backup domain controllers. This is only true in the sense that any domain controller can handle database updates.

Even with the multi-master architecture of Active Directory, there are still five key roles which must be exclusive to only one domain controller per domain or per forest depending on that role…

- Schema Master (one per forest) : this domain controller controls all modifications to the Active Directory schema. This is a critical role and only a special group of users called Schema Admins can make these changes. An incorrect or malicious change to the schema could corrupt the entire Active Directory. For this reason, the Schema Master is sometimes removed from the network entirely until a schema change needs to be made.
- Domain Naming Master (one per forest) : this domain controller controls the addition or removal of domains in the forest.
- Relative ID (RID) Master (one per domain) : this domain controller allocates RIDs to each domain. A RID is a unique identifier which is given to each new user, group or computer created in the domain. This relative ID combined with the domain's security ID makes up the SID of the object.
- PDC Emulator (one per domain) : this domain controller performs a number of key functions in a domain. It ensures that system time is synchronized with other domain controllers (a requirement for Kerberos authentication). It acts as a PDC if you are operating your Active Directory domain in mixed mode. It handles password changes even in native mode so if a domain controllers refuses a logon attempt due to password failure, it will forward the

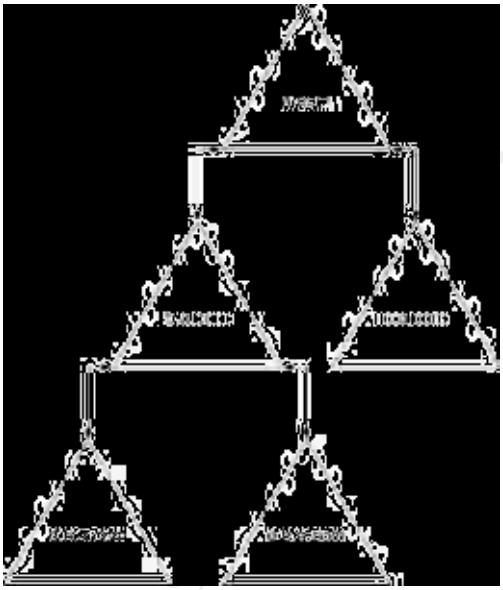
authentication request to the PDC Emulator in case replication of the password change has not yet spread to all domain controllers. It also makes changes to Group Policy Objects. If you want to edit a GPO, you must have a PDC Emulator.

- Infrastructure Master (one per domain) : this domain controller ensures that domain groups are updated whenever a member of that group is modified or renamed. For example, User A from Domain A is a member of Group B in Domain B. If User A is renamed to User C then the Infrastructure Master in Domain B must ensure that its Group B is also updated with this change.

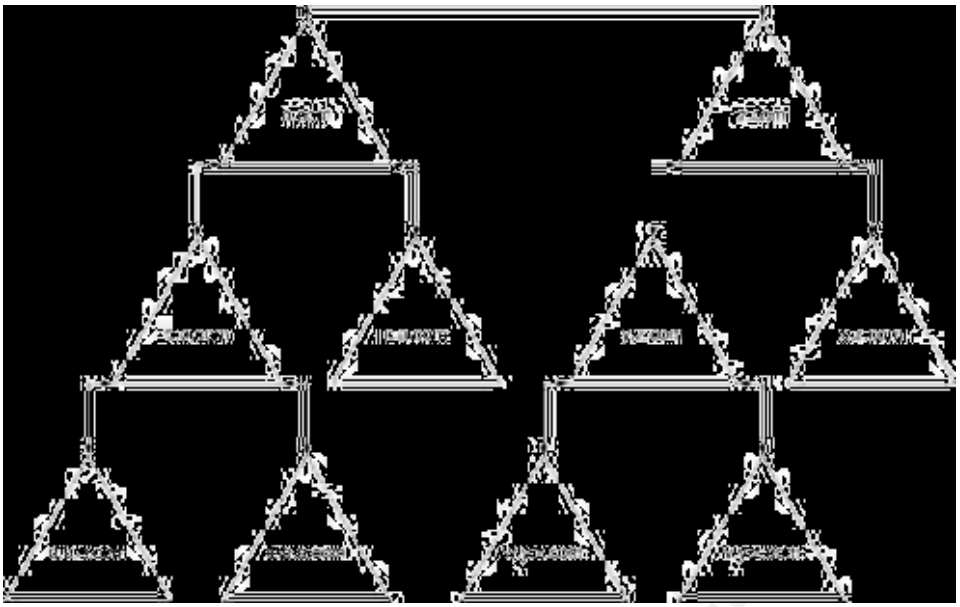
2.3.2 Forests, Domains and Trees

With Active Directory, domains can be group together in trees and forests. A forest is a group of one or more domains. Every domain in a forest automatically trusts every other domain in the forest. A domain in one forest will not automatically trust a domain in another forest.

A tree is a hierarchical, contiguous group of domains. In the example below, a forest consists of one tree. The root domain is a.com which has two child domains called b.a.com and c.a.com respectively. The b.a.com domain also has two child domains called d.b.a.com and e.b.a.com and all of the domains are contiguous.



There can be one or more trees in a forest. The tree structure is reflected in DNS domain names and it is usually DNS requirements that will dictate whether or not you have more than one tree in a forest. In the example below there are two trees in the forest. Each tree has its own contiguous namespace. The fact that there are two separate name spaces (a.com and z.com) is the only reason for there to be separate trees. It might be necessary to maintain separate name spaces because of politics, geographical reasons or perhaps as the result of a merger with a different company.



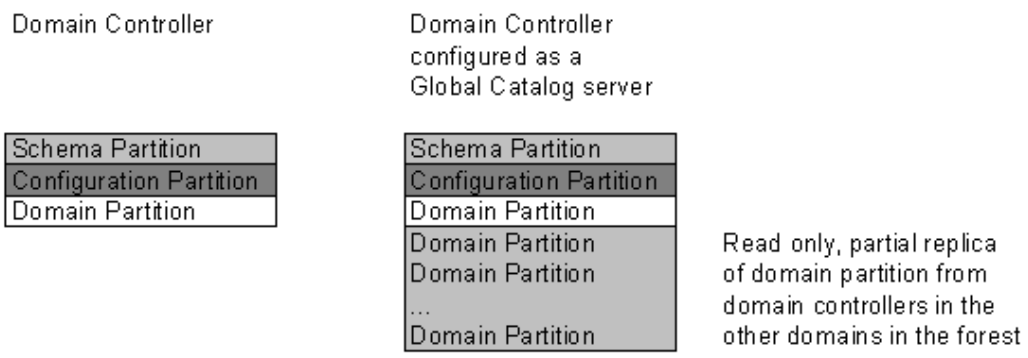
2.3.3 Global Catalogue

Active Directory also requires some of its domain controllers to be configured as global catalogue servers. The global catalog is what makes it so easy for users to find network resources regardless of whether they are in the same domain as the user. The global catalog, as the name suggests, contains information about every object in the entire Active Directory forest - it's global. It contains some key information about each object (a subset of each object's attributes defined in the schema) in the entire forest. It does, however, store all of the information for each object in its own domain.

The Active Directory database is divided in to three logical partitions. These partitions allow the database to be distributed to domain controllers throughout the forest.

- Schema Partition : stores class and attribute definitions for all Active Directory objects. Every domain controller in the forest has a full replica of this partition.
- Configuration Partition : stores configuration objects related to sites and services. Every domain controller in the forest has a full replica of this partition.
- Domain Partition : stores data about all objects in the domain. Every domain controller in the domain has a full replica of this partition.

A domain controller configured to be a global catalogue server will have additional domain partitions which are partial replicas of the domain partitions belonging to the other domains in the forest.



In addition to facilitating the search for network resources, the global catalog has a more important role to play. Any user or computer logging on to the domain must be able to access a global catalog. Part of the authentication process involves finding out which (if any) universal groups the user is a member of. The global catalog provides this information.

Kerberos also relies heavily on the global catalog when authenticating resource requests. Only a domain controller in the same domain as the resource/user can authenticate. Kerberos uses the global catalog to figure out which domain a resource is in. Once it knows this, it knows which domain controller to use for authentication.

2.3.4 Sites

Sites represent the physical structure of Active Directory. A site is really just a collection of IP Addresses. Active Directory uses sites to ensure that the domain controller closest to a user or computer is used to handle authentication requests.

Active Directory also uses sites to replicate partitions of the Active Directory database to other domain controllers. Domain controllers within the same site will replicate information faster than domain controllers in different sites.

So, sites are used to define well connected networks in order to control both authentication and replication traffic.

Sites generally have no direct security implications although they can be used to help improve recovery times in the aftermath of an impact by configuring delayed replication sites. We'll discuss this further in the section on site design.

2.3.5 Active Directory Database & Schema

Active Directory stores its objects in a database. The database is stored in a file called ntds.dit and is managed by an improved version of the Microsoft JET Engine (as used with early versions of Exchange) called the Extensible Storage Engine (ESE).

The schema defines the database. The schema contains a list of all of the classes from which you can create an Active Directory object (such as a user) and a list of all of the attributes which an object can have (such as logon name and password).

Changes to the schema could adversely affect the entire Active Directory forest and most schema changes can not be reversed. For this reason, only "Schema Admins" and only the domain controller with the "Schema Master" FSMO role have the necessary privilege to modify the schema.

There are a number of ways to control who can be a Schema Admin and we'll look at this further in the next section.

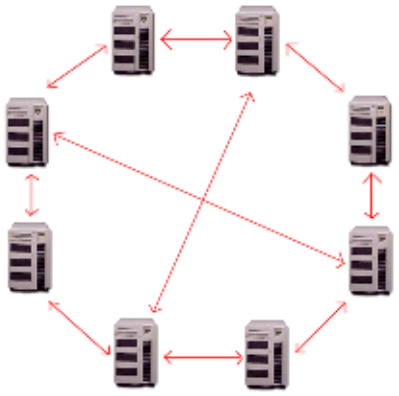
2.3.6 Replication

Replication ensures that a change made to one domain controller will also be made to all of the other domain controllers in the domain.

There are two general types of replication. Intra-site replication happens when domain controllers within the same site replicate information. Inter-site replication happens when domain controllers in different sites, perhaps separated by a dial-up link, replicate information.

- Intra-site Replication : Active Directory uses a mechanism called the Knowledge Consistency Checker (KCC) to maintain a replication ring topology between domain controllers in the same site. The KCC runs every 15 minutes and ensures that there are at least two replication paths leading to any particular domain controller. It also makes sure that there are no more than three hops between any two domain controllers in the site.

  Example of replication topology between 8 domain controllers – each domain controller has at least two paths and no more than three hops to get to any other domain controller (diagram courtesy of WindowsIT Pro magazine).



  When a change is made to information on a domain controller, that domain controller will wait five minutes before replicating the change to its replication partners. Urgent replication events, such as account lockouts, are replicated immediately.

Password changes will attempt to get replicated to the PDC Emulator before other domain controllers. This is because logon attempts which fail due to password problems get referred to the PDC Emulator as a point of authority.

- Inter-site Replication : Active Directory uses site links to replicate information between sites. A site link contains information such as the cost (a weighting which helps the KCC decide which link to use if there is more than one link available), network protocol to use (IP or SMTP), when and how often the link is available (for example, you may want to limit replication to twice during night time),

  SMTP could be selected as the replication protocol in cases where the network is not fully routed (your site link might use a dial-up connection) but can not be used to replicate full domain partition replicas. It can only be used to replicate schema, configuration and partial domain (global catalogue) partitions. Therefore, it can't be used to handle intra-domain replication.

  There are obvious security risks associated with having the SMTP service enabled on your domain controllers. The section on Risk Management in Active Directory explores these risks and explains how to mitigate them.

With multi-master replication architecture, there is always a risk that a change is made to the same object at the same time by different domain controllers. To overcome this problem, Active Directory uses Update Sequence Numbers (USNs) which act like version numbers. Every domain controller maintains a table of USNs for each object and its attributes. If a change is made to the object or one of its attributes, the USN for that object or attribute is incremented on the domain controller responsible for the change (this is called an originating change). The domain controllers regularly compare their USN values and if they find a USN with a higher value then they'll replicate that change (this is called a replicated change).

See the Microsoft Windows 2000 Server Security Center for more information on secure replication.

2.3.7 Trust Relationships

Trusts are special relationships between domains which allow the concept of the "single logon". A trust allows a user in domain A access resources in domain B without having to have a user account in domain B. If domain B trusts domain A then we can add the user from domain A to groups in domain B which in turn have access to resources in domain B.

Another way of looking at a trust is that it allows authentication across domains. It is not responsible for authorization. "Domain A trusts domain B" is another way of saying that domain A will allow authentication of users from domain B. No more, no less.

There are different types of trust relationship. A trust might be one way in which case domain A trusts domain B but domain B does not trust domain A. A trust might

be two way where domain A trusts domain B and domain B trusts domain A (this could also be accomplished with two separate one way trusts).

Trust relationships could also be transitive. Imagine the case where domain A trusts domain B and domain B trusts domain C. If these trusts are transitive then domain A will also trust domain C.

Finally, trusts can be implicit or explicit. An implicit trust is created and maintained automatically. An explicit trust must be created manually.

Active Directory supports two different types of trust relationships. It uses Kerberos to automatically establish and maintain implicit, two-way, transitive trusts between parent and child domains. It also uses NTLM to create explicit, one-way, non-transitive trusts between domains that are not part of the same tree. This type of trust has to be created manually. This is like the trust used with NT4 and is the only way to create trust between a Windows 2000 domain and a Windows NT domain or Windows 2000 domains in different forests.

There is another special type of trust used in Active Directory. This is called a "Shortcut Trust" and is used to speed up the logon process by shortening the trust path that would normally have to be followed. For example, let's say domain A trusts domain B which, in turn, trusts domain C. If there is a shortcut trust between domain A and domain C then authentication can be achieved (i.e. a Kerberos ticket received) directly from domain C without having to go through domain B.

With trusts come vulnerabilities, especially when the trust is between two domains in different forests. Malicious users could take advantage of SID histories to elevate their privilege. We'll cover these vulnerabilities in more detail in the next section.

2.3.8 DNS and Active Directory

For a primer on DNS, please visit the Microsoft DNS Center and refer to the Windows 2000 DNS whitepaper there.

Microsoft say that Active Directory, like any other directory service, is primarily a namespace. In order for Active Directory to operate with Internet technologies, Active Directory uses DNS to implement its namespace.

Active Directory has three key uses for DNS…

- DNS was originally designed to provide name resolution for the Internet. It converts DNS names to IP addresses and converts IP addresses to DNS names.

- Active Directory uses DNS to locate resources. For example, clients and servers use DNS to locate domain controllers and global catalogue servers when they first start up or when a user logs on. Domain controllers are located using special records called SRV records (see appendix A for a description of an SRV record).

- Active Directory also uses DNS for naming domains. An Active Directory domain hierarchy is reflected in the DNS name of a domain. For example, if domain b is a child domain of domain a.com then it will be identified as b.a.com and so forth.

Active Directory will work away with any version of DNS as long as that version provides support for SRV records. As mentioned earlier, these records provide a means of locating services such as domain controllers or global catalogue servers as opposed to normal host name resolution using A or PTR records. It is also desirable that the DNS solution include the following two features…

- Incremental Zone Transfers : only the change to the zone file is replicated as opposed to replicating the entire zone file.
- Dynamic Updates : clients and servers with dynamically allocated IP addresses (DHCP) can automatically register with DNS. An entry does not have to be manually created.

Microsoft offers two ways to implement DNS. You can use the traditional primary and secondary zones as used with BIND versions of DNS or you can integrate DNS with Active Directory.

Integrating DNS with Active Directory is offers much in terms of security. Every DNS record is an Active Directory object therefore it has its own access control list. Active Directory allows for secure dynamic updates to DNS which means the DNS server will only accept dynamic updates from computers that have permission to update the DNS record in the zone. Computers attempting to update DNS have to authenticate first.

DNS was designed to be a distributed database due to the scale of the Internet and also due to the volume of DNS query traffic generated every day. Therefore it requires some replication mechanism. Standard primary and secondary DNS zones implement these DNS databases as simple text files which get replicated in “zone transfers”. Zone transfers can generate a lot of network traffic between DNS servers. Active Directory integrated zones include zone transfers as part of replication which is encrypted by RPC thus providing more security.

As with the Internet, if DNS fails then Active Directory will also fail. So, it is critical that DNS is will protected. We’ll talk more about DNS in the section on Domain Design.

2.3.9 LDAP

LDAP is an Internet protocol designed at the University of Michigan. It is a stripped-down version of the X.500 protocol intended for use on high-speed TCP/IP networks. It is commonly used by e-mail clients to retrieve e-mail addresses from directory servers such as Exchange. Any LDAP-enabled client can access Active Directory.

LDAP can be used to issue a denial of service attack in the form of a flood of LDAP requests.

Active Directory is a complex directory service which provides many features to both users and administrators. The next section takes a look at how best to manage and mitigate risk for Active Directory to ensure confidentiality, integrity and availability of its information.

3. Risk Management in Active Directory

In the last section, we talked about what Active Directory does and how it does it. We know that Active Directory provides identification, authentication and authorization. We know that it uses a variety of components in its implementation such as DNS, forests, domains, trusts, a database & schema, replication, FSMO roles, etc.

Risks to Active Directory come from vulnerabilities within these components and from within the environments in which they operate (server hardware, operating system, etc.) but they can also arise from the manner in which Active Directory is designed and maintained. Such examples might include weak password policies, insecure access control mechanisms or relaxed controls regarding privileged group membership.

For this reason it's important not just to address known vulnerabilities but also to ensure that Active Directory is securely implemented and maintained.

Let's now explore these vulnerabilities, what they mean in terms of risk and what we can to do manage the risk.

There are four general categories of risk management…

- Risk Mitigation – this is the most often used strategy – many defence in depth functions are aimed at mitigating risk. For an Active Directory implementation, this might include firewall implementation, patch management, anti-virus software, strong authentication and access control policies, etc.
- Risk Avoidance – this is often used early on in the design stage when decisions might be made to use a tried and trusted operating system or other mature product. It also includes the practice of disabling functionality if not required. For example, many viruses are 16-bit applications and expect filenames to be in "xxxxxxxx.yyy" format. NTFS uses automatic 8.3 name generation to provide backward compatibility. This can be disabled using a registry key to avoid the risk of viruses targeting files in this way.
- Risk Acceptance – used as a last resort if there are no reasonable alternatives or perhaps chosen if the risk is deemed low with minimal potential damage. With this strategy, the ability to recover quickly from attacks is crucial. Redundancy and backup/data retention policies are very important.
- Risk Transference – this method allows a third party (insurance company, internet service provider or some other specialist vendor) to accept a particular risk.

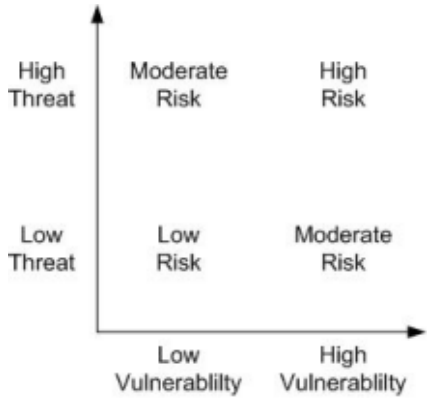
Just as with Defence In-Depth, the most effective and realistic strategy is to combine these methods as opposed to opting for one method in particular. For example, to secure a domain controller, we might use Risk Mitigation to patch an RPC locator vulnerability, we might use Risk Avoidance to use Windows 2000 instead of Windows 2003 and to disable the SMTP service if we're not using it for replication, we might use Risk Acceptance by keeping regular system state backups and we might use Risk Transference by using an Internet Service Provider to host DNS.

3.1 Risks, Threats and Vulnerabilities

A Threat is any activity which could potentially damage the confidentiality, integrity or availability of your data. A Vulnerability is a weakness in a component or policy that allows a threat to happen. A risk is a combination of threat and vulnerability.

Risk = Threat x Vulnerability

The "multiplication" sums up the relationship between threat and vulnerability nicely. High threat combined with a high vulnerability results in high risk



A vulnerability without a threat is not a risk. The problem with vulnerabilities is that they could be discovered first by someone interested in exploiting them. Even if the vulnerability is first discovered by the vendor responsible, the time taken for a threat to be developed for a newly discovered vulnerability is being reduced constantly. Companies like Microsoft will only announce a vulnerability once a patch (or other solution) has been developed to fix that vulnerability. However, these patches don't always get applied in a timely fashion and so the risk grows because now there's a threat to the vulnerability.

3.2 Threats to Active Directory

Now we'll talk about what type of threats exist to Active Directory and where these threats might come from.

3.2.1 Type of threats to Active Directory

Here's how Microsoft categorize the type of threats to Active Directory. See the Best Practice Guide for Securing Active Directory Installations and Day-to-Day Operations for further details. Let's examine them and see how they relate to our three key security requirements of confidentiality, integrity and availability.

S-T-R-I-D-E

- Spoofing – the goal of this type of attack is to pretend to be someone or something you're not. It generally involves using false credentials to access resources.

- Tampering with Data – this type of threat targets data confidentiality, integrity and availability by modifying user or system data.

- Repudiation – this type of threat is intended to perform some action which can not be traced back to a particular individual. This usually involves avoiding or tampering with security logs.

- Information Disclosure – this type of threat targets confidentiality. Anyone who accesses data which is considered to be private or confidential compromises confidentiality. Properly configured access control mechanisms are essential in helping to manage this type of threat.

- Denial of Service – this type of threat targets only availability by attempting to consume server resources such as CPU, disk space or memory in order to block legitimate use of that resource.

- Elevation of Privilege – like Data Tampering, this type of attack could target confidentiality, integrity or availability of data. The goal of this type of attack is to obtain higher privileges than intended in order to access resources or perform tasks.

3.2.2 Source of threats to Active Directory

Note – for more information about Windows 2000 and Active Directory groups, please refer to section 4.3.1 and appendix C below.

Now that we know the type of threats to Active Directory, where might they come from? Just as with vulnerabilities, threats can come from users, software or the environment. Here's a brief explanation of the different user threats as defined by Microsoft at TechEd 203.

- Rogue Anonymous User – has no user account. Such a user could use social engineering or guesswork to obtain access to Active Directory objects. Risks from anonymous users can be reduced by enforcing complex password policies and staying up to date with security patches.

- Rogue Authenticated User – has a user account and limited access to Active Directory objects. Risks from authenticated users can be reduced by using the principle of least privilege in the form of access control.

- Rogue Data Administrator – can create, modify or delete certain Active Directory objects. Could create or delete Active Directory objects. Data Administrators must be trusted but the risk of having a Data Administrator with malicious intent can be mitigated through well designed Delegation. However, a rogue data administrator could take advantage of an inherent vulnerability in Windows 2000 Active Directory by creating a universal group with over 5,000 users. This would impact replication, as every extra user added to the group will require replication of the entire group to all global catalogue servers. They could create multiple accounts with the same user principle name which could impact user logons if UPNs are used for logon ID. They could also create

objects until physical domain controller disk space is used up potentially causing the domain controller to crash.

- Rogue Service Administrator – can modify an operating system and has access to domain controllers. A service administrator usually requires full control over Active Directory objects and domain controllers. So, a rogue service administrator can do just about anything to impact the confidentiality, integrity and availability of Active Directory data. Even a trusted service administrator could, in an effort to mitigate risk, be the source of a denial-of-service. For example, a firewall which has been configured to block RPC could affect replication if domain controllers are attempting to replicate across the firewall; a likely scenario if you have a DMZ which contains a domain controller. It is key for service administrators to be trusted personnel.

- Someone with physical access – if someone has physical access to a domain controller then it doesn't matter what other security measures are in place. That person could target confidentiality by stealing data (such as password or user data) on disks, target integrity by booting into an alternate operating system to modify the existing one or target availability by destroying hardware. SYSKEY can be used to reduce the risk of stolen passwords being cracked.

Apart from these types of users, threats can also come from other areas…

- Software – malware in the form of a virus, worm or trojan designed to exploit a known vulnerability. Software is probably the source of most threats today. A quick look at the Internet Storm Centre is enough to convince anyone of the constant threat from malware.

- Environment – not all threats are malicious in nature. Environmental threats come in the form of fire, water, dust, temperature, vibrations, etc. The distributed nature and multi-master architecture of Active Directory helps reduce this risk by providing redundancy. How do you mitigate the risk of an earthquake? You can't avoid it or mitigate against it. You must use Risk Acceptance to provide a good recovery/redundancy plan and use Risk Transference to let an insurance company cover the cost of recovery.

The ability to detect a threat early is key to damage limitation. Threat Detection methods include regular audits, network and host intrusion detection systems as well as resource monitors such as Microsoft Operations Manager which could be used to alert on unusual CPU utilization, disk space usage or memory utilization. Microsoft Operations Manager could actually be used to implement a host intrusion detection system by alerting on resource utilization and event log entries for a domain controller.

3.3 Vulnerabilities in Active Directory

As mentioned earlier, vulnerabilities can manifest themselves in a number of ways. There can be vulnerabilities inherent in components used by Active Directory, there can be vulnerabilities inherent in the environment in which Active Directory is

implemented and there can be vulnerabilities introduced in the way Active Directory is configured.

Here's a list of vulnerabilities specific (though not all exclusive) to Active Directory…

- Vulnerabilities in AD components
  - FSMO Roles (Domain Controllers) – anyone with the privilege to shut down a computer could impact a FSMO role. For example, if the PDC Emulator is unavailable then user logon requests with new passwords will fail because the password change would not be able to replicate.
  - Global Catalogue – remember, this is required for user logons to succeed. For every user attempting to log on, the global catalogue is checked to see what universal group membership they might have. No global catalogue means no logons. This problem can be bypassed with a registry setting (IgnoreGCFailures) but this poses another problem – if a resource denies access to a certain universal group but you don't check to see if the user is a member of that group then you have an access control vulnerability.
  - Sites – it's not a good idea to apply group policy objects to sites because site policies will override domain and domain controller policies. Password policies are domain-level policies and should be left in the control of individual domains.
  - Database & Schema – schema modifications could bring down the entire Active Directory. All default object permissions are stored in the schema; even a non-malicious schema change could potentially affect security in the entire forest.
  - Replication – replication relies on replication protocols such as RPC and SMTP which have their own vulnerabilities such as the MS Blaster worm which impacted RPC. Changes to universal groups can also affect replication as the entire group needs to be replicated when a change is made.
  - Trust relationships – explicit trusts between domains in different forests are vulnerable to SID spoofing. Active Directory users and groups can have more than one SID. If a user or group migrates from one domain to another it will have a new SID but should still be able to access resources using its original SID, otherwise access control lists and groups would have to be updated to reflect the SID change. So users and groups maintain a list of their current and previous SIDs. This is called a SID history. SID spoofing occurs when a rogue administrator adds a SID to the SID history of a user account to gain unauthorized access to Active Directory resources. SID filtering is used to mitigate this risk by ensuring that only SIDs from the trusted domains are allowed. If SID filtering is applied to a trust then it will apply to all domains in that forest so any domains that you want to quarantine in this way should be placed in that forest.
  - DNS – a target for many threats including denial-of-service attacks, spoofing where the attacker impersonates a DNS server in order to misdirect a client, or DNS cache poisoning where the attacker makes a valid DNS server misdirect a client.

    - LDAP – as a query protocol like DNS, it is vulnerable to denial-of-service attacks using a flood of LDAP requests.
- Vulnerabilities in AD environment
    - Network – a network operating system without a network is not much use. An attack on a network could quickly render Active Directory useless. Network vulnerabilities might also allow packet sniffing. If your network is using hubs instead of switches, there's a greater chance of a computer using "promiscuous mode" whereby it can listen to all packets including those not addressed to it. Packet sniffing can lead to captured authentication packets which can lead to password cracking.
    - Hardware – vulnerability in vendor hardware used to implement domain controllers. This could be a security vulnerability where a server has hot-swappable mirrored disks allowing a disk to be easily removed without impacting the server. It could also be a reliability vulnerability.
    - Software – other software running on your domain controller may present a vulnerability. What if your anti-virus software itself was subject to an attack and fooled into thinking that any .dit file contained a virus?
    - Computer Room – physical security, local climate (temperature, humidity, vibration, electro-magnetic interference, etc.)
- Vulnerabilities in AD configuration
    - Authentication – Kerberos is a secure authentication protocol, however the NTLM or NTLMv2 protocols introduce vulnerabilities due to the way passwords are encrypted. If Active Directory is configured to allow this type of authentication then the vulnerability will be present.
    - Access Control – weak access control introduces vulnerabilities which makes it easier for threats to succeed.
    - Group Policies – weak policies are vulnerable to threats too. A weak password policy which doesn't enforce a minimum password length of 14 makes it easier for a brute force attack to succeed.
    - DNS – DNS configured as Active Directory integrated is more secure than DNS implemented as standard primary & secondary zones. Poorly designed DNS could result in internal or private DNS records being made available to the public and more vulnerable to attacks.
    - Forests and domains – it's critical, from a design perspective, to understand when to use forests, domains or organizational units in order to meet data isolation or autonomy requirements. Failure to do so could result in breaches of confidentiality of Active Directory data.

OK, now we have an idea as to what vulnerabilities exist and where the threats might come from. The next two sections discuss how best to mitigate or avoid the subsequent risks in the configuration of Active Directory through good design and on an ongoing basis once Active Directory is implemented.

4. Secure Active Directory Design

With Windows NT4 it was pretty straight forward, a domain is a security boundary. With the hierarchical model of Active Directory the lines aren't so clearly drawn. The true security boundary for Active Directory is the forest rather than the domain. So security needs to be designed at the forest level as well as the domain level.

For this reason, it's important to understand what levels of security can be achieved at the forest level, domain level and organizational unit level. Domains can no longer be considered the autonomous, isolated entities they were with NT4.

Autonomy means that administrators can independently manage the resources over which they have authority. Isolation means that administrators not authorized to manage resources can't manage them.

The process for designing your Active Directory structure should be a top down approach beginning with the forest design, moving on to the domain and DNS design and then moving on to the organizational unit design before finishing with a site topology or physical design. Often, each of these steps will have to take into account the existing business requirements, infrastructure and equipment but these are beyond the scope of this paper. We will focus on security requirements and how they are taken into account at each step of the design stage.

Two key objectives when designing your Active Directory structure should be simplicity and security. Often a single forest/single domain with organizational units is best/most frequently used model. In a recent Windows IT Pro poll, 73% of Active Directory administrators were happy with their single forest/single domain model. As long as the principle of least privilege is properly applied throughout the design stage, the potential vulnerabilities in Active Directory configuration can be avoided.

We'll now take a look at the four stages of Active Directory design and examine how to ensure confidentiality, integrity and availability by adhering to the principle of least privilege.

4.1 Forest Design

Forest design naturally plays a vital role in ensuring the confidentiality of Active Directory data. The options for forest design are simple; do you want a single forest or multiple forests?

In order to answer this question we must carefully consider what levels of autonomy and isolation are required. It is important that these requirements be completely addressed at the design stage since forests can not be easily merged at a later stage. Merging two forests or migrating a domain from one forest to another are both complex operations best avoided if at all possible; a domain can only be removed from a forest if it has no child domains.

Although cost is also a factor and each additional forest in the design comes with its own cost. Every forest created needs at least one domain and every domain has its own costs for creation and maintenance.

The Microsoft Windows Security Resource Kit recommends the following factors to take into account when assessing your autonomy and isolation requirements.

- Enterprise Administrators have ultimate control. The Enterprise Admins group has authority over every object in every domain in the forest. Enterprise Administrators can not be isolated from any domain in the forest. If domains require strict isolation (you don't trust each others administrators) then multiple forests are needed to meet this requirement.
- There can only be one schema per forest. If domains require a different schema to other domains then multiple forests are required. A schema contains the default permissions for every Active Directory object. Active Directory's default security settings may be sufficient for small companies but for larger companies, additional security settings and design may have to be considered. If domains require different default permissions then multiple schemas and, therefore, multiple forests are required. A forest change affects all domains in the forest so some sort of change control policy needs to exist. If a change control policy can not be agreed upon by all domain stakeholders then multiple forests will be required.
- The Global Catalogue contains a subset of attributes belonging to every Active Directory object in the forest. These attributes are used to make it easy to find objects anywhere in the forest. If any Active Directory objects should not be searchable then a separate forest is required.
- If there is a security requirement that a domain should only have a one-way trust relationship with another domain then a separate forest for each domain is required. It's like saying "You trust me but I don't trust you". Since all domains in a forest have two-way Kerberos trusts the only way to implement this requirement is two have an explicit one-way trust between the domains while in different forests.
- If complete domain isolation is required, including isolation from other domain administrators, then multiple forests will be required.

As mentioned in section 2.3.7 and section 3.3, trusts between domains in different forests are vulnerable to SID spoofing. In order to mitigate the risk of SID spoofing, it is recommended that SID filters be applied to the trust so that only SIDs from the trusted domain are allowed.

Every forest created will have a root forest domain which is the first domain created in a forest. This domain will provide the initial membership for the Enterprise Admins and Schema Admins groups. The Built-in Administrator account is added to these groups by default. This account along with any other accounts subsequently placed in these groups must be protected as they have control over the schema and the entire forest. There are a number of ways to protect these users and groups…

- Limit the number of Enterprise & Schema Admins to around five users and use those account only when absolutely necessary. Sometimes an empty root forest domain (with no user or computer accounts) is used to ensure isolation of these groups. You could have an even more secure policy of ensuring that there are no permanent members of the Schema Admins group.

Just add users as necessary and remove them again as soon as the task is complete.
- Use a feature of Group Policy Objects called Restricted Groups which ensures that group membership can not be changed. We'll talk more about group policy in section 4.3, Organizational Unit Design.
- Use strong authentication methods, for example, smart cards or strong passwords.
- Ensure physical security of forest root domain controllers particularly if all domain controllers are physically located near each other. As with any domain, if the domain controllers are not functional then neither is Active Directory.

4.2 Domain Design

As mentioned before, a Windows 2000 Active Directory domain is no longer a complete security boundary. Consider what happens when a computer joins a domain. There are a number of security implications arising from the fact that the Domain Admins & Enterprise Admins groups get added to local Administrators group on the machine. This reflects the fact that there is implicit trust between all parent and child domains in the same forest.

Rather than being seen as a security boundary, a Windows 2000 Active Directory domain should be seen as an administrative and security policy boundary. The following factors should be taken into account when determining autonomy and isolation requirements at a domain level.

- Administrative boundary – the Domain Admins group for any domain, with the exception of the forest root domain, has absolute control within that domain and this control does not apply to other domains in the forest. Administrators in the forest root domain, by virtue of the fact that they are also members of the Enterprise Admins group are not isolated from controlling other domains in the forest. An empty forest root domain could be used to ensure that this is not the case. So, domain administrators have autonomy over their own domain but they are isolated from controlling other domains. If you have domain administrators which do not require and should not have control over the entire forest then multiple domains will be required.
- Account policy boundary – account policies applied to one domain will not affect other domains in the forest. Account policies consist of password policies such as minimum length and password complexity, account lockout policies such as the number of incorrect logon attempts before an account is locked out and the duration for which an account should be locked out. Account policies also consist of Kerberos policies such as how long a Kerberos ticket should last. Account policies applied to one domain can not affect account policies applied to other domains in the forest.
- Preserve an old NT domain structure – if migrating from NT4, there may be political requirements that existing domains are preserved. Migration from NT4 to Windows 2000 is beyond the scope of this paper.

Again, simplicity and security should be the key design principles when assessing domain requirements. Start with the minimum number of domains, i.e. one, and add additional domains if there are valid administrative or policy-related requirements.

There is another important security-related factor which can influence the number of domains used in an Active Directory structure. This is DNS and is covered in some detail in the section 4.2.1 below.

4.2.1 Designing DNS for Active Directory security

Active Directory is not a flat domain model, it uses DNS namespace to reflect the domain hierarchy. The domain naming strategy used by an organization is an important task that must be properly addressed during the Active Directory design stage. There are a number of ways to integrate DNS with Active Directory in a secure way.

A tree consists of one or more domains with a contiguous namespace, e.g. example.com, sales.example.com, marketing.example.com, etc. As mentioned in section 2.3.2, there can be one or more trees in a forest although our rule of thumb applies here too; the fewer trees the better.

One of the big problems around DNS design is that most organizations will already have some public DNS presence, e.g. a publicly available website. In this case, the first question we must ask is whether or not to share private (Active Directory) and public namespace.

To have the same internal and external namespace, internal users must be able to access internal and external DNS records while public or external users should only be able to access external DNS records. This sort of configuration would involve having a firewall between internal and external DNS as well as duplicating the external DNS internally. This is known as segmented or split-brain DNS and if this is not configured properly then you run the risk of having your internal or private DNS records exposed to an attacker.

The simplest option of having internal and external DNS records in a single namespace poses an unacceptable security risk in terms of exposing private DNS records to an attacker.

It is definitely good practice to avoid presenting internal DNS records to the general public. An alternative is to differentiate between internal and external resources quickly and easily. To accomplish this separate internal and external namespaces are required. This would involve having a delegated namespace where a section of your external DNS reserved for internal DNS records. For example, if you already have example.com as a public namespace then you can create a zone called internal.example.com to host your private DNS records (including SRV records). Having a separate zone also means that you can delegate control of your internal DNS to Active Directory administrators.

Besides ensuring that the DNS design prevents exposure of private DNS records to public DNS requests, we can further secure DNS servers by implementing the

following security measures as described in the Microsoft Windows Security Resource Kit.

- Use Active Directory Integrated DNS zones – dynamic updates can be secure, DNS records have access control lists, zone transfers are part of Active Directory RPC replication. Zones are hosted on domain controllers which are more likely to be more physically secure than other member servers.
- Implement separate internal and external DNS servers – even if you've decided to use the same internal and external namespace, you should separate DNS servers in order to protect private DNS records from the public.
- Restrict zone transfers – it is possible to restrict zone transfers only to specific DNS servers or IP addresses using the DNS console. This mitigates the risk of having an attacker perform an unauthorised zone transfer.
- Implement IPSec between DNS clients and servers – With IPSec, a client must authenticate with a DNS server before performing any DNS queries.
- Restrict DNS traffic at the firewall – a firewall is used to block network traffic in to and out of your private network. It can be used to block incoming and outgoing DNS queries. See section 4.4, Site Design for more information on firewalls.
- Limit management of DNS – Windows 2000 provides the DNSAdmins group to facilitate delegation of DNS management. A member of the DNSAdmins group can manage DNS without having access to other Active Directory objects.
- Protect the DNS cache – DNS servers maintain a DNS cache which is used to store the results of recent queries. This cache can speed up the process of resolving names or IP addresses but it is also vulnerable to DNS cache poisoning whereby an attacker adds false information to the cache. Windows 2000 DNS has an option to "Secure Cache Against Pollution" whereby the DNS server will not cache a DNS response which doesn't appear to match the request.

An additional recommendation would be to avoid implementing all DNS servers on the same network segment. An attack targeting specific ranges of IP addresses could disable all of your DNS servers at once. In January 2001, Microsoft was subject to such an attack which resulted in all of their web servers being unavailable for 24 hours.

4.3 OU Design

The need for autonomy and isolation drives the need for multiple forests and domains. The need for administrative delegation drives the need for multiple organizational units. This delegation is a key function not just for ease of administration but in that it allows separation of privilege which is another key principle when it comes to implementing security.

Organizational Units are often used to reflect a business structure and the criteria may be departmental, functional or project related. But because organizational units are used for delegation it makes sense to use them to reflect administrative structure.

An organizational unit is used to implement access control and delegation by having an access control list. Organizational units are also used with Group Policy. Let's discuss access control, delegation and group policy in more detail to see how they can be used to implement security in Active Directory.

4.3.1 Access Control

Before talking about access control, let's briefly discuss groups and what type of groups are available in Active Directory. A group is simply a collection of users.

There are two basic types of group in Active Directory.

- Security Group – used to control access to resources.
- Distribution Group – used for other purposes not related to security such as e-mail distribution lists.

Regardless of the type of group, each group also has a scope which determines where the group can be used. Active Directory provides three different group scopes.

- Global Groups can only contain users in the same domain as it, however it can access resources in other trusted domains.
- Domain Local Groups can contain users from any domain but can only access resources in the same domain as it.
- Universal Groups are a combination of Global and Domain Local groups in the sense that they can contain users from any trusted domain and can also access resources from any trusted domain but they are only available if Active Directory is operating in native mode (see section 2.3.1). Information about Universal Groups are included in the global catalogue.

Groups can contain other groups but there are some rules about group membership which vary depending on whether or not Active Directory is operating in native mode.

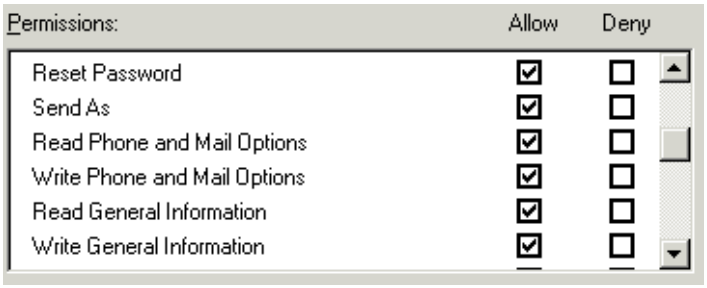
- In native mode, global groups can contain other global groups from the same domain. This can not be done in mixed mode.
- In native mode, domain local groups can contain other domain local groups from the same domain. This can not be done in mixed mode.
- Universal groups are not available in mixed mode.

Please refer to Appendix C for more information on groups.

We'll talk more about groups at the end of this section when discussing how best to implement access control.

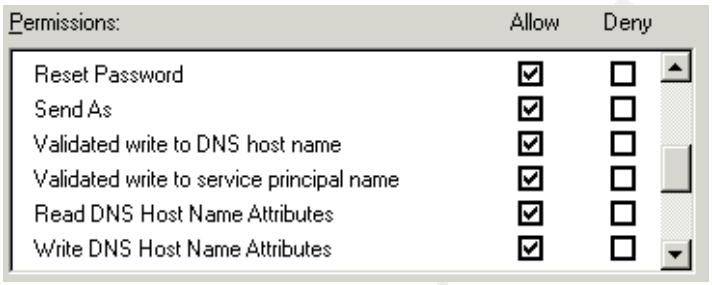
Access control is used to implement the principle of least privilege. Every object in Active Directory has an access control list (ACL). An ACL consists of a number of access control entries (ACE). These entries determine who can access the object and also the extent of that access. Different objects have different permissions.

For example, a user object has permissions such as "reset password" and "Read Phone and Mail Options"…



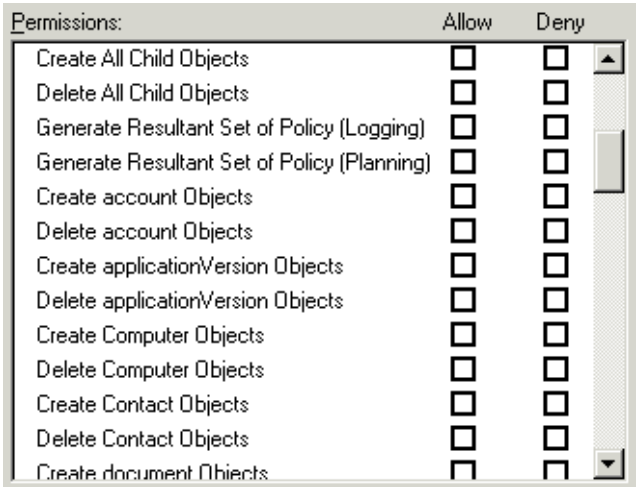
Note that the ability to explicitly deny access is something new to Windows 2000 and Active Directory. A deny permission will always take priority over an allow permission. For example, suppose a user is a member of two groups. Both groups have access control entries in an object's access control list. One group has read access to the object while the other group has read access explicitly set to deny. The user will be denied access to the object despite having permission to read the object in one of the groups.

A computer object has different permissions such as "Validated write to DNS host name" or "Read DNS Host Name Attributes"…



Every Active Directory object has a default permission which is defined in the Active Directory schema. Security principles, i.e. users, groups or computers can be assigned these permissions. This is what makes up an ACE.

Organizational Units are no different in that they too have access control lists and their own set of permissions such as "Create All Child Objects" and "Delete Computer Objects"…



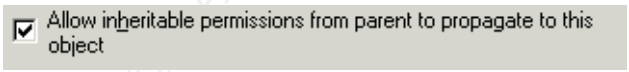
The difference is that an organizational unit is a special type of Active Directory object because it is also a container. It can contain other Active Directory objects such as users and computers.

This allows for delegation. For example, if an organizational unit contains all the users and computers in the sales department, some administrators could be given limited access to manage just those users and computers while isolating them from managing the HR users and computers in human resources OU.

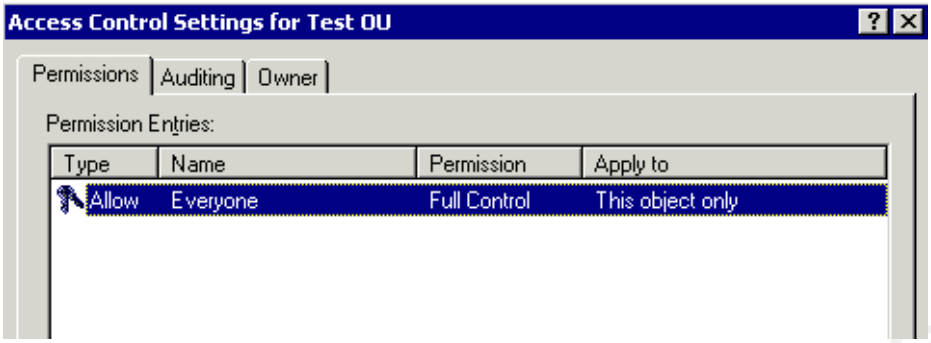
Up until now, we've only talked about permission explicitly given to an object. However, there's a second implicit type of permission which objects can inherit from their parent. This is called inherited permission.

Organizational Units can be hierarchical just like domains. An organizational unit can contain other organizational units. For this reason, additional degrees of access control are made possible by the concept of inheritance. An organizational unit can inherit the access control list used by its parent organizational unit. This greatly simplifies the task of delegation.

The scope of inheritance can be controlled in two ways. First, an object has the option to prevent inheritance although this is most useful for organizational units in order to implement a degree of isolation if required…



Second, each entry in the object's access control list can be configured to apply only to that object or to apply to any child objects if that object is a container such as an organizational unit…



Access control need to be carefully managed if it is to work as intended. There are some basic rules for implementing access control in a standard and consistent way.

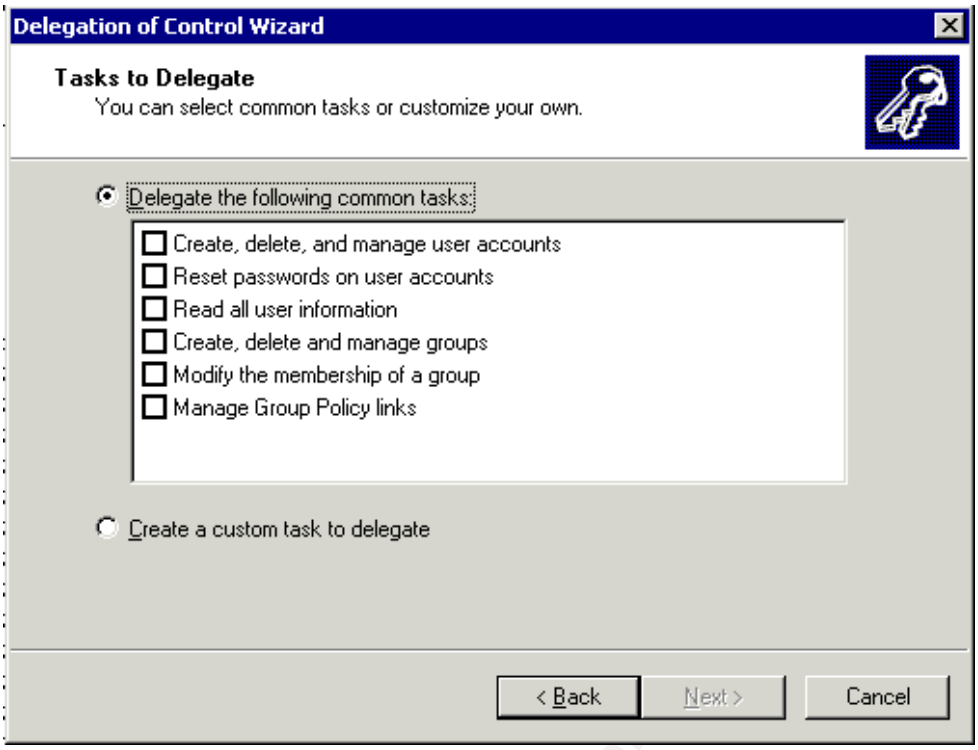
- Permissions should only be assigned to domain local groups.
- Domain local groups should have global groups as their members.
- Global groups should contain the actual users requiring access.

So, the user is assigned to a global group. The global group is assigned to a domain local group (this can have members from any trusted domain) which resides in the same domain as the resource being accessed. The domain local group is given the required permission to access the resource. In other words, the domain local group is given an access control entry in the object's access control list.

4.3.2 Delegation of Administration

Delegation is just another form of access control. Delegation is implemented by applying access control lists to organizational units.

Active Directory provides the Delegation of Control Wizard to assist in setting up basic delegation. This wizard modifies the access control list on an organizational unit based on the answers you provide to its questions. You select the group that you want control delegated to and the wizard will prompt you to assign permissions to the group…

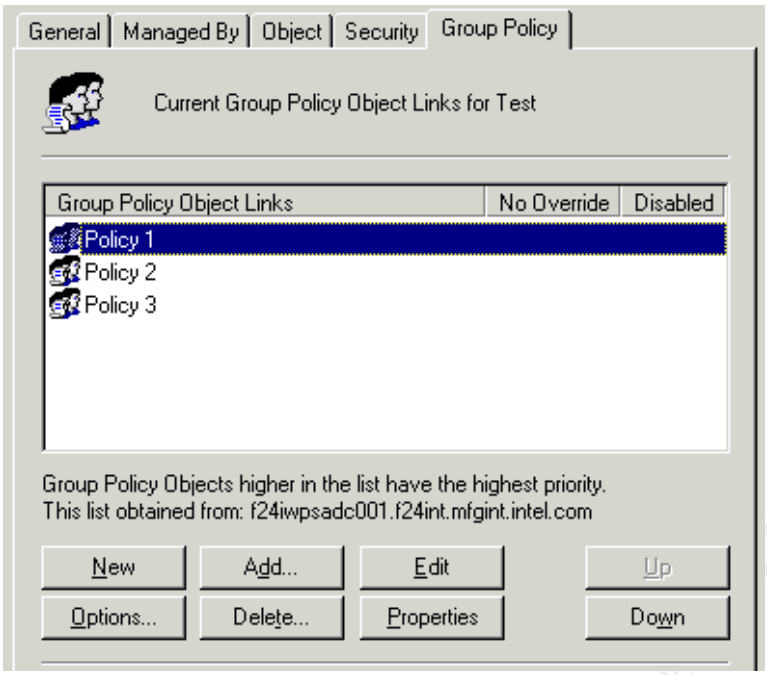


The end product is that the organizational unit's access control list is updated to reflect these new permissions.

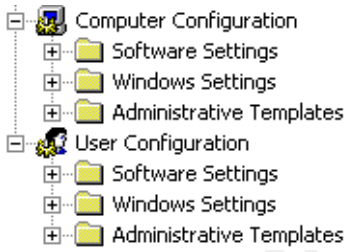
4.3.3 Group Policy Objects

Group Policy Objects are used to centrally manage the desktop, application and security configurations of client and severs. Organizational Units are also used to implement policy-based security in the form of group policy. Generally, any user or computer object in an organizational unit linked to a group policy will receive that policy. The exception to this rule is when policy filtering is used which we'll discuss later.

Group Policy Objects can be managed from the Active Directory Users and Computers console. By right-clicking on a domain or organizational unit and selecting it's properties. Under the "Group Policy" tab, group policy objects can be added, removed or edited…



Group policy can be applied to users or computers. In other words, a group policy applied to a user will follow that user around regardless of what computer they use. Likewise, a group policy applied to a computer will be applied to the computer regardless of who (if anyone) is logged on. Every group policy object has a computer configuration which gets applied to computer objects and a user configuration which gets applied to user objects…



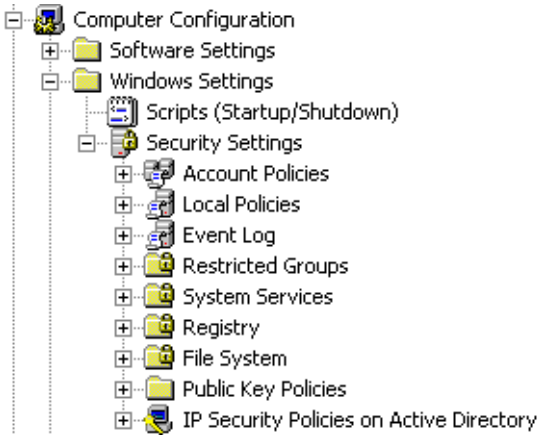
Here's a list of what group policy can manage…

- Computer Configuration
  - Software Settings – allows applications to be assigned to computers so that the application will always be installed on the computer.
  - Windows Settings – contain startup & shutdown scripts as well as security settings.
  - Administrative Templates – changes that can be applied to the registry such as DNS connection suffixes, how to run startup & shutdown scripts, etc.
- User Configuration
  - Software Settings – allows applications to be assigned or made available (published) to users.
  - Windows Settings – contain logon & logoff scripts as well as security settings.

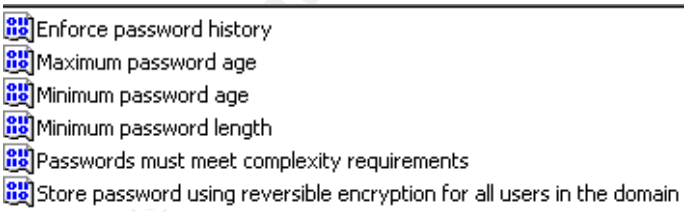
- Administrative Templates –changes that can be applied to the registry such as desktop environment configuration.

The Windows Settings part of both computer and user parts of a group policy object contain the security settings which combined with Administrative templates allows us to implement policy-based security.

Computer Security settings include account policies which contain password policies and account lockout policies. Account policies are applied to the domain OU so that there is a standard account lockout and password policy for all machines.



Password policy settings are shown below. They include settings to enforce password history to prevent the same password being used twice in a shot period of time, settings to specify how long a password should be used for before requiring a change, minimum length which should be 15 or should contain special characters (meets complexity requirements) if you want to avoid storing an LanManager hash for the password. LanManager hashes are very vulnerable to password cracking by tools such as L0phtCrack v4.

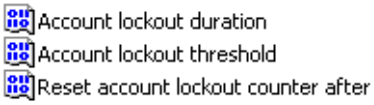


Here are Microsoft's recommended password policy settings compared with their default settings (Best Practice Guide for Securing Active Directory)…

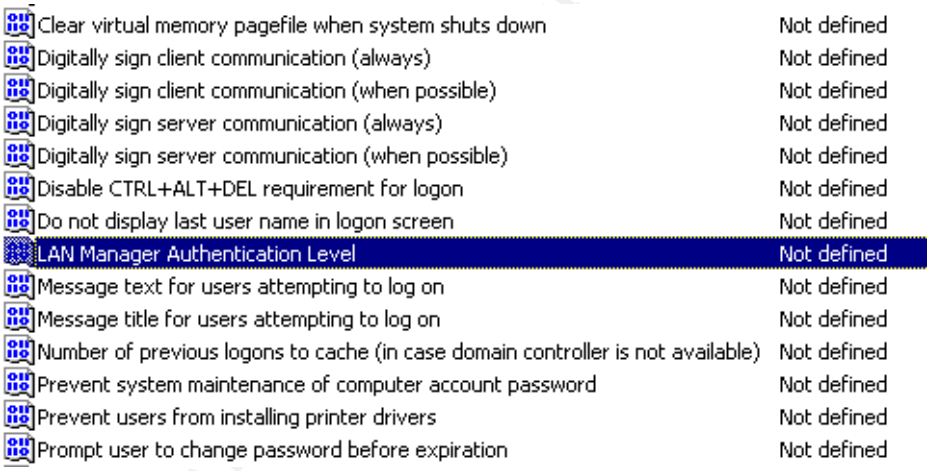
Policy	Default	Recommended	Comments
Enforce password history	1 passwords	24 passwords	Prevents users from reusing passwords.
Maximum password age	42 days	(No change)	

Minimum password age	0 days	2 days	Prevents users from cycling through their password history to reuse passwords.
Minimum password length	0 characters	8 characters	Ensures minimum password strength.
Password must meet complexity requirements	Disabled	Enable	For the definition of a complex password, see "Creating a Strong Administrator Password" earlier in this guide.
Store password using reverse encryption for all users in domain	Disabled	(No change)	

Account Lockout settings are shown below. These settings determine how many unsuccessful logon attempts are allowed before an account is locked out and the amount of time that an account should be locked for.



Local Policy settings include more security settings, some of which are shown below. The level of Lan Manager authentication can be specified, the number of logons to cache in the event of domain controllers being unavailable, etc.



As mentioned in section 4.1, Restricted Groups are configured here too. This allows Administrators to control group membership by ensuring that if a member gets deleted, that member will be added again when the policy is applied or refreshed.

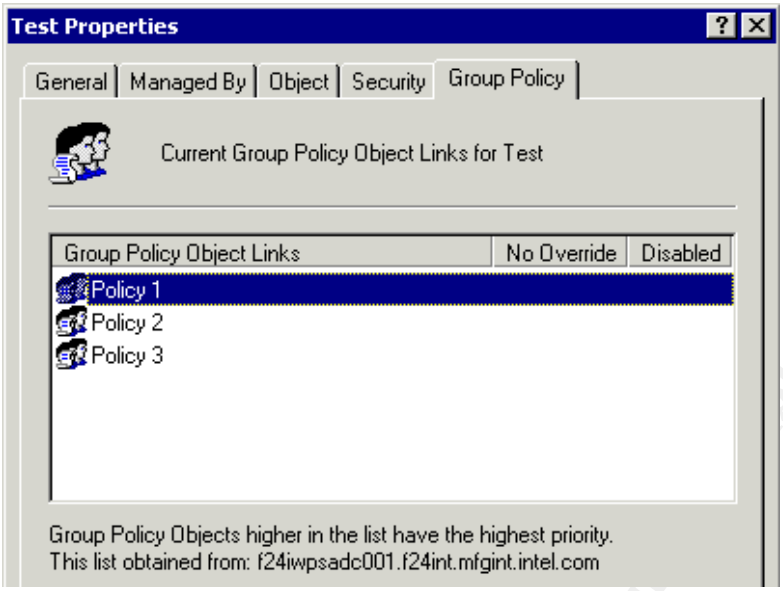
The vast majority of security related group policy settings apply to the computer. User related settings generally involve using Administrative Templates to restrict access to components such as Explorer settings, desktop settings, network settings, etc. An example of desktop settings is shown below.

Please refer to the Microsoft Windows 2000 Group Policy Whitepaper for a complete list of group policy settings.

Group policy can be applied to sites, domains and organizational units so a user or computer could have more than one group policy affecting it. Consequently, there must be an order to which group policy gets applied. This order is as follows…

- Local Policy – every Windows 20000 computer has a local group policy object which gets applied first.
- Site Policy – any site policy will get applied next. These should only be used if you need to apply a network related configuration change.
- Domain Policy – any domain policy such as account policy (password, account lockout, Kerberos) will be applied next.
- OU Policy – finally, policies link to organizational units get applied. If there are multiple organizational units, the group policy will be applied starting at the top OU in the hierarchy and working down to the OU containing the object. It's also possible for one OU to have multiple policies linked to it. In this case, the polices are applied according to their order in the GPO list. For example, this Test OU has three policies linked to it. Group Policies higher in the list are applied later so they have higher priority.

Computer group policy objects are applied when a computer starts up while user group policy objects are applied when a user logs in. Thereafter group policy objects are refreshed every 90 minutes by default.



The fact that multiple group policy objects can get applied to users or computers means that there is a possibility of one group policy object overwriting a value set in another group policy object. Each group policy object value can either be set or left unconfigured. An unconfigured value will not change the value if it has already been set by another group policy.

There are a number of ways to affect the way in which group policy gets applied…

Group Policy Filtering involves setting the ACL of the group policy object. Every group policy object has a permission called "Apply To". If you set this permission to DENY for any user or computer then that policy will not be applied.

The "No Override" setting ensures that a group policy will not get overwritten by another group policy object below it in the hierarchy.

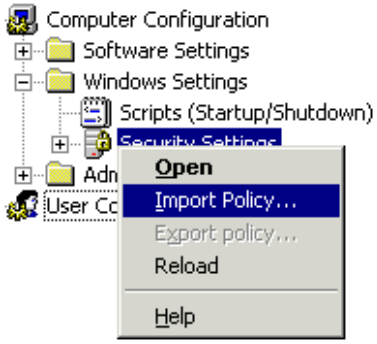
The "Block Inheritance" setting ensures that a group policy will not automatically inherit settings from group policy objects above it in the hierarchy.

Group Policy can also be used to apply security templates to computers. Security templates are pre-defined settings designed to provide different levels of security. They are similar to group policy objects in the sense that they were designed to make it easy to manage security for a large number of machines. There are a number of templates available with varying levels of security implemented in each.

The list includes Default Workstation, Default Server, Default Domain Controller, Secure Workstation or Server, Highly Secure Workstation or Server, Secure Domain Controller, Highly Secure Domain Controller, Highly Secure Web Server.

An Administrator could also make a custom security template using the Security Templates management console.

Security Templates can get applied to a client or server using the Security Configuration & Analysis console on that machine. However, it makes sense to use Group Policy objects to deploy a security template to entire groups of computers. To do this, right-click on the "Security Settings" and select "Import Policy"…



As with organizational units, group policy objects can use their access control lists to delegate control. This is key in enabling autonomy. There's little point in delegating control of an organizational unit to an administrator if control of a group policy object linked to that organizational unit is not also delegated.

Group Policy is an extremely powerful tool which allows administrators to centrally standardise security settings across entire domains or organizational units.

4.4 Site Design

Site design is largely influenced by replication requirements but there are security implications. For example, group policy can be applied to sites. Sites are independent to the logical structure of Active Directory so the ability to apply group policy to a site has major security implications. A site could contain more than one domain so the application of a group policy to a site could potentially violate isolation requirements already incorporated into the forest and domain plan. For this reason, only enterprise administrators can create group policy objects at the site level.

4.4.1 What is a site?

Site design differs from forest, domain and OU design because it reflects a physical structure as opposed to a logical structure. A site is a physical representation of the network structure and has no direct relationship with forests, domains or organisational units. A site could span more than one domain as easily as a domain could span more than one site.

A site is simply a collection of subnets. The only requirement is that these subnets be "well connected" which generally means that they are all part of the same LAN. There remains some debate surrounding the definition of "well connected". It's generally accepted that a network connection speed of 512Kbps qualifies as well connected but values of 256Kbps have also been suggested. There is a Group Policy setting which detects slow links when applying group policy. If left unconfigured, the default value is 500Kbps and this is probably as good a figure as any in defining well connected.

4.4.2 Why are sites important?

Active Directory has two important uses for sites.

Firstly, because Active Directory domain controllers are multi-master any domain controller has the ability to service client requests. It is preferable for a client to use the domain controller nearest to it in order to minimise network traffic and maximise efficiency in dealing with the request. This request is most likely to be an authentication request on behalf of a user.

Secondly, Active Directory uses sites to enable replication of information between domain controllers. Domain controllers within the same site will replicate information faster and more frequently than with domain controllers in different sites.

These are the main criteria in determining where to place domain controllers but where and how we deploy domain controllers has security implications which we'll cover in the section 4.4.3 below.

4.4.3 Domain Controller Placement

Domain controllers contain information that is critical to Active Directory functioning properly and therefore should be a secure build. This is important if a domain controller. Microsoft's general recommendation is to have at least one domain controller in every site and at least two domain controllers in every domain. It is important that every site have at least one domain controller in order to avoid having authentication traffic going across a slow network connection. If it's the case that your sites are geographically dispersed and not all of the sites have service administrators then it is even more critical to have a secure domain controller.

Microsoft's Best Practice Guide for Securing Active Directory Installations and Day-to-Day Operations makes a number of recommendations for ensuring secure domain controller deployment.

- Always install the latest service pack and hotfixes – this is the only way to be sure that all known vulnerabilities are addresses.
- Disable Automatic 8.3 name generation – as mentioned previously, many forms of malware target the old DOS-style named files. An easy way to mitigate this risk is to disable those names on NTFS.
- Install anti-virus software and keep the definition files updated – apart from keeping anti-virus software up to date, you should exclude the Active Directory database & log files, File Replication Service database & log files and the entire SYSVOL folder from scans to avoid conflicts between the anti-virus software and Active Directory when trying to open these files.
- Prohibit the user of cached credentials when unlocking the console – when cached credentials are used to unlock a console, no domain controller is checked to see if that account has since been disabled.
- Create a reserve file to allow for recovery from disk space attacks – create a large file which resides on the same volume as the ntds.dit file. If disk space

is consumed in an attack, this file can be deleted to allow the Active Directory database sufficient disk space to operate normally.

- Enable only essential services – table 9 of the best practice guide provides a list of services and recommends whether or not they should be disabled or configured not to start up automatically.
- Remove NTFS permissions which give "Everyone" access – strangely, the default permission on NTFS partitions is to grant Everyone Full Control. This should be changed to allow only Administrators have Full Control and limit Everyone to Read and Execute access.
- If possible, disable "Pre-Windows 2000 Compatibility" – this ensures that the Active directory can not be queried using anonymous access.
- Place the ntds.dit file and SYSVOL on a separate volume to the system volume since the system volume is most often targeted in malware attacks.
- Maintain physical security
    - make it difficult for a domain controller to be booted into an alternative operating system such as Linux by removing floppy and CD-ROM drives.
    - If access to a domain controller can not be controlled, then consider using SYSKEY to prevent domain controller reboots. SYSKEY is used to provide additional protection for Active Directory passwords. It works by encrypting passwords with a system key which can be derived from a password or stored on a floppy disk. If SYSKEY is used, then either the password or floppy disk will be required anytime the domain controller reboots.
    - Store backup tapes off-site in a secure location.

When considering where to place FSMO roles on domain controllers, it's worth noting that the two forest-wide roles, the schema master and domain naming master roles, could be placed on the same domain controller as they are rarely used and should both be tightly controlled. Apart from keeping tight control on Schema Admin members, it might be a good alternative to keep the schema master completely disconnected from the network, only connecting it when schema changes are required.

It is likely that your defence in-depth solution includes one or more firewalls to control incoming and outgoing network traffic. It may be that you have a DMZ (semi-secure area) set up for web servers and you place a domain controller in the DMZ to provide authentication.

If the firewall is not configured properly any number of issues could arise including problems with user and computer authentication, trust issues, file access issues, DNS issues and replication issues. Because of the number of RPC vulnerabilities, it may be tempting to block RPC traffic at the firewall level. However, this can cause many issues for domain controllers attempting to communicate over the firewall. For Active Directory to function correctly through a firewall, a number of protocols must be allowed to function normally. Here is a list of protocols (as described in Microsoft's whitepaper on Active Directory in Networks Segmented by Firewalls) which must be taken into consideration when domain controllers are placed either side of a firewall…

1. User Authentication - A user network logon across a firewall uses the following protocols :

- Microsoft-DS traffic (445/tcp, 445/udp)
- Kerberos authentication protocol (88/tcp, 88/udp)
- Lightweight Directory Access Protocol (LDAP) ping (389/udp)
- Domain Name System (DNS) (53/tcp, 53/udp)

2. Computer Authentication - A computer logon to a domain controller uses the following protocols :

- Microsoft-DS traffic (445/tcp, 445/udp)
- Kerberos authentication protocol (88/tcp, 88/udp)
- LDAP ping (389/udp)
- DNS (53/tcp, 53/udp)

3. Establishing an Explicit Trust Between Domains - when establishing a trust between domain controllers in different domains, the domain controllers communicate with each other by means of the following protocols :

- Microsoft-DS traffic (445/tcp, 445/udp)
- LDAP (389/tcp) or 636/tcp if using Secure Sockets Layer (SSL))
- LDAP ping (389/udp)
- Kerberos authentication protocol (88/tcp, 88/udp)
- DNS (53/tcp, 53/udp)

4. Validating and Authenticating a Trust - trust validation between two domain controllers in different domains uses the following protocols :

- Microsoft-DS traffic (445/tcp, 445/udp)
- LDAP (389/tcp or 636/tcp if using SSL)
- LDAP ping (389/udp)
- Kerberos (88/tcp, 88/udp)
- DNS (53/tcp, 53/udp)

5. File Access – uses the SMB protocol :

- File access uses SMB over IP (445/tcp, 445/udp).

6. DNS Lookups – use the DNS protocol :

- DNS (53/tcp, 53/udp)

7. Active Directory Replication - replication requires the following protocols :

- RPC Dynamic Assignment (1024-65535/tcp)
- LDAP (389/tcp or 636/tcp if using SSL)
- LDAP ping (389/udp)
- Kerberos (88/tcp, 88/udp)

- DNS (53/tcp, 53/udp)
- SMB over IP traffic (445/tcp, 445/udp)

Also, the Internet Control Message Protocol (ICMP) protocol must be allowed through the firewall from the clients to the domain controllers so that the clients can receive Group Policy information.

Most of these functions can be implemented using the same five ports…

- Port 53
- Port 88
- Port 389
- Port 445
- Port 636

The firewall can be easily configured to accommodate this traffic. However, the biggest challenge lies in allowing RPC traffic, which is required for replication, through the firewall. This is because RPC assigns ports dynamically to RPC services when they start up. A client requesting an RPC service must first go to another service called the RPC Endpoint Mapper on port 135 in order to be directed to the correct port.

Configuring the firewall to allow all possible RPC ports would render it useless since any port from 1024 to 65535 could be used. The solution in this case is to limit the ports used by RPC to a static port number which can more easily be configured and controlled by the firewall. A registry setting can be added to the domain controller's registry to achieve this…

By adding a value called TCP/IP Port to the registry key below…

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS\Parameters

…and assigning a fixed port number to it you can limit RPC replication to one static port.

Now, you just have to add the RPC locator port and the new static port to your firewall configuration and replication can function normally across the firewall.

5. Secure Active Directory Operation

If Active Directory is well designed then it should be easy enough to maintain on a daily basis. There are still a number of tasks which need to be performed regularly and contingency plans to be used in the event of a compromise.

5.1 Auditing

Regular security audits are a good way to keep track of events in Active Directory. Active Directory provides Audit Policies as an easy way to implement auditing. An audit policy lets you define what events should be captured in the domain controller's security event log.

There are some guidelines (recommended in Microsoft's book "Windows 2000 Active Directory Services") which should be adhered to when creating audit policies.

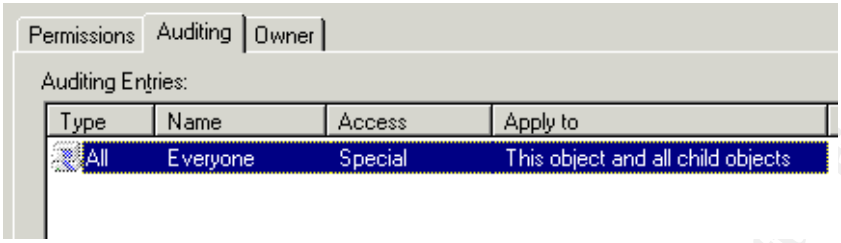
- If you need to track trends then logs should be archived and not overwritten.
- When creating an audit policy, you should audit for access by the Everyone group to ensure that everyone and not just authenticated users are tracked.
- An audit policy is no good if the results are never reviewed – review the security event logs frequently. Of course, auditing should be automated to be of any practical use so it is desirable to have your Active Directory monitor can audit the logs for you.
- Naming conventions make auditing easier because it is easier to identify potential rogue machines.
- Audit administrative tasks in order to capture any changes made by administrators.

There are two steps to implementing auditing in Active Directory. First, you must create an audit policy in a group policy object by choosing what you want to audit. Active Directory allows for auditing for success or failure of the following events (again, courtesy of "Windows 2000 Active Directory Services")…

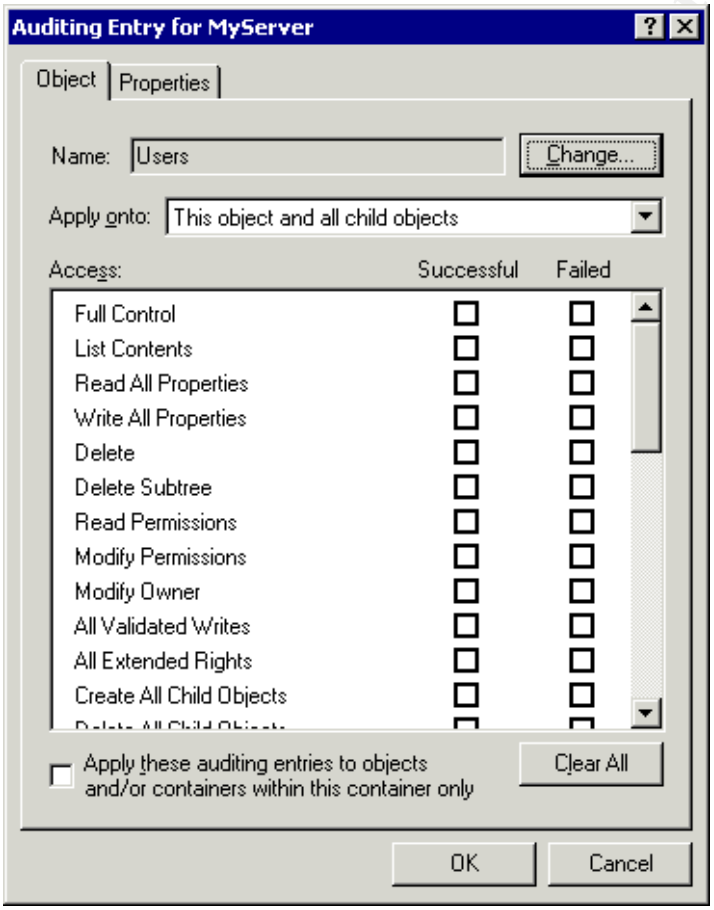
- Account Logon – a domain controller has received a request to authenticate a user account
- Account Management – an administrator has created, changed or deleted a user account or a group
- Directory Service Access – a user has accessed an Active Directory object
- Logon Events – a user has logged on or logged off
- Object Access – a user has accessed a file, a folder or a printer object
- Policy Change – a change was made to user security options, user rights or audit policies
- Privilege Use – a user has performed a privileged task such as changing the system time
- Process Tracking – a program performed an action
- System Events – a user shutdown or started up a computer

Once you have identified the activities you want to audit, the second step is to enable auditing of specific resources. This involves configuring a security access control list (SACL) for an Active Directory object.

The security access control list for an Active Directory object can be found under the security tab of the object's properties. Click the "Advanced" button and select the Auditing tab…



From here, you can specify to whom you want the audit policy to apply and whether you want to audit for success of failure or both…



5.2 Monitoring

Without monitoring, there is no mechanism for detecting threats to Active Directory.

It is good practice to use a network intrusion detection system (NIDS) or a host intrusion detection systems (HIDS)

A number of factors would normally determine which option you go for…

- Budget - generally NIDS is more expensive to implement since it requires additional hardware.
- Network - since it operates in realtime, NIDS has trouble dealing with high-speed switched networks plus is confused by encrypted networks (IPSec).
- Platforms - HIDS might not support a particular O/S.
- Threat concerns - if you're more concerned about IP-based DOS attacks then go for NIDS.
- Monitoring for specific events - HIDS can detect specific events such as logon/logoff, it's much more difficult to interpret NIDS data and identify events.

A good defence in-depth solution would be to combine network intrusion detection systems (NIDS) with host intrusion detection systems (HIDS).

What should we look out for? Here's a comprehensive list of events or behaviours to watch out for as specified in the Windows IT Pro article, "Monitoring your AD-Enabled Network"

1. Domain Controllers
   - Low CPU or memory resources on domain controllers
   - Low disk space on volumes housing the SYSVOL folder, the AD database (ntds.dit) file, or the AD transactional log files
   - Slow or broken connections between domain controllers (within a site or across sites)
   - Slow or failed client network logon authentication requests
   - Slow or failed LDAP query responses
   - Slow or failed Key Distribution Centre (KDC) requests
   - Slow or failed AD synchronization requests
   - Netlogon service not functioning properly
   - Directory service agent (DSA) service not functioning properly
   - KCC not functioning properly
   - Excessive number of Server Message Block (SMB) connections
   - Insufficient RID allocation pool size on local server
   - Problems with transitive or external trusts to Win2K or downlevel NT domains
   - Low AD cache hit rate for name resolution queries

2. Replication
   - Failed replication
   - Slow replication
   - Replication topology invalid/incomplete (i.e., it lacks transitive closure/consistency)
   - Replication using excessive network bandwidth
   - Win2K dropping too many properties during replication
   - Update sequence number (USN) update failures
   - Other miscellaneous replication-related failure events

3. Global Catalogue
   - Slow or failed GC query responses
   - GC replication failures

4. DNS
   - Missing or incorrect SRV records for domain controllers
   - Slow or failed DNS query responses
   - DNS server zone-file update failures

5. FSMO Roles
   - Inaccessibility of one or more Operation Master servers
   - Forest or domain-centric Operation Master roles not consistent across domain controllers or within a domain or forest
   - Slow or failed Operation Master responses

6. Miscellaneous components
   - Low-level network connectivity problems
   - TCP/IP routing problems
   - DHCP IP address allocation pool shortages
   - WINS server query or replication failures (for legacy NetBIOS systems and applications)
   - Application or service failures or performance problems

5.3 Recovering from Attacks

Even after configuring Active Directory to be as secure as possible there will always be the risk of a new vulnerability being exploited by an attacker. For this reason, it is essential to be able to recover quickly from an attack.

Fortunately, there are a number of methods available to help us…

5.3.1 Backups & System State

Obviously you need to have a good backup policy in order to deal with disaster recovery. Without backups, the entire Active Directory is potentially at risk and a backup is only useful if it includes system state data.

System state backups include the registry of the computer being backed up, the active directory database and SYSVOL folder and system boot files.

Part 2 of the Best Practice Guide for Securing Active Directory Installations and Day-to-Day Operations provides a good procedure for recovering from catastrophic forest-wide corruption.

5.3.2 Non-authoritative Restores

A non-authoritative restore happens when your restore a system state while acknowledging that any changes made since the backup will still be replicated to other domain controllers.

To perform a non-authoritative restore on a domain controller, it must first be placed into “Directory Services Restore Mode” which is one of the boot options.

When the domain controller starts up, you can log on as Administrator and use the backup wizard to restore the system state data.

This type of restore is useful for getting a domain controller back on the network and part of the replication topology following a downtime local to that domain controller.

5.3.3 Authoritative Restores

An Authoritative restore happens when you restore Active Directory data and do not allow changes to that data since the backup to be replicated. In effect, you are saying that this copy of the data is the valid copy.

This is a useful option if you are recovering Active Directory data which has been deleted and the change has subsequently replicated to the other domain controllers.

To perform an authoritative restore on a domain controller, it must be disconnected from the network. You must start the domain controller in “Directory Services Restore Mode” and first perform a non-authoritative restore of the system state.

After re-starting the domain controller, go back into “Directory Services Restore Mode” and run the ntdsutil program. Using this utility, you can mark some or all of the Active Directory to be “authoritative”. In other words, you’re saying that this domain controller is now authoritative and it’s Active Directory data should be replicated to other domain controllers.

Finally, connect the domain controller to the network again and re-start the computer as normal. The data on this domain controller will then replicate to the other domain controllers.

5.3.4 Delayed replication sites

This is an interesting use of Active Directory sites where a domain controller is placed into a site which deliberately has replication configured to occur only once a day or once a week.

The idea is that any damaging change which gets replicated through the domain or forest will not reach this particular domain controller before you have had time to do an authoritative restore to undo the change.

5.3.5 Net IQ GPA – Group Policy Repository

Third party products such as Net IQ’s Group Policy Administrator allow for group policies to be backed up in a central location.

6. Summary

There is no one correct way to design your Active Directory structure but there are plenty of wrong ways!

Active Directory should be designed with autonomy and isolation in mind.

Designing and maintaining a secure Active Directory is by no means quick and easy but Active Directory does provide the means to implement key security principles of least privilege and separation of privilege. It also provides the means to ensure confidentiality, integrity and availability of information as long as it is designed with these goals in mind.

If Active Directory is designed with security in mind then the day-to-day operation will be far easier to maintain.

7. References

7.1 Book References

Smith & Komar. Microsoft Windows Security Resource Kit. Redmond, Washington: Microsoft Press 2003. 33-130.

Fuchs, Jim. Microsoft Windows 2000 Security Technical Reference. Redmond, Washington: Microsoft Press 2000. 89-174, 237-298.

Spealman, Jill. Designing a Microsoft Windows 2000 Directory Services Infrastructure. Redmond, Washington: Microsoft Press 2001.

Spealman, Jill. Microsoft Windows 2000 Active Directory Services. Redmond, Washington: Microsoft Press 2003.

Desai & Chellis. Windows 2000 Directory Services Administration. Alameda, California: Sybex 2001.

Steen, Doug. Designing a Microsoft Windows 2000 Network Infrastructure. Redmond, Washington: Microsoft Press 2001. 425-503.

Bragg, Roberta. Five Key Lessons to Securing Your Active Directory. Quest Software

7.2 Magazine References

Sutela, Jesse. “Delayed Replication AD Recovery” Windows & .NET Magazine. July 2004.

Holme, Dan. “Planning & Customizing AD Delegation” Windows & .NET Magazine. January 2004.

Mar-Elia, Darren. “GPO Security” Windows & .NET Magazine. October 2003.
“Planning for Active Directory” Windows & .NET Magazine. September 2000.

Smith, Randy. “Monitoring Important Security Events” Windows & .NET Magazine. October 2003.

Bennion, Jess. “AD Delegation : Beyond the Basics” Windows & .NET Magazine. August 2002.

McIntosh, Robert. “Active Directory Sites” Windows & .NET Magazine. December 2000.

Minasi, Mark. “DNS and Active Directory” Windows & .NET Magazine. July 2001.
“Active Directory Oddities” Windows & .NET Magazine. September 2000.

Toombs, Douglas. “Single Domain Migration” Windows & .NET Magazine. July 2001.

Daily, Sean. “Monitoring Your AD-Enabled Network” Windows 2000 & .NET Magazine. September 2000.

Sharick, Paula. “Active Directory Delegation of Control Wizard” Windows 2000 & .NET Magazine. September 2000.

7.3 Internet References

Tandon, Sanjay. “Active Directory Security : Planning & Operations” TechEd. 2003. URL: http://www.only4gurus.com/v2/download.asp?ID=3035

Only4Gurus - SplitBrain DNS Server Configuration for ISPs

Only4Gurus – Active Directory in Networks segmented by Firewalls

Only4Gurus – Active Directory Security : Planning & Operations

Only4Gurus – DNS in the Active Directory Tree

7.4 Other References & Related Links

Microsoft Technet

Microsoft Knowledge Base

Windows 2000 Server Domain Migration Cookbook

Active Directory Operations Overview

Windows 2000 Server Active Directory Technology Center

Microsoft Active Directory Security Center

SecurityFocus – An Audit of Active Directory Security (Part 1)

SecurityFocus – An Audit of Active Directory Security (Part 2)

WindowSecurity – Securing Windows 2000 Active Directory (Part 1)

FTP Online - Secure your Applications with Active Directory

Windows-Expert.net - Active Directory & DNS

Appendix A - General SRV Record Format

An SRV record takes the following format in DNS.

_Service._Protocol.DnsDomainName TTL Class SRV Priority Weight Port Target

- The _Service field specifies the name of the service, HTTP, LDAP, etc.
- The _Protocol field specifies the protocol, such as TCP or UDP.
- The DnsDomainName field specifies the domain name to which the resource record refers.
- The TTL field is a 32-bit value representing the number of seconds that the entry remains in DNS cache
- The Class field is IN for Internet or CH for Chaos (RFC 1034)
- The SRV field indicates that the DNS record is an SRV record
- The Priority field can be 0 to 100 and specifies the priority of the host. Clients attempt to contact the host with the lowest priority as a preferred host.
- The Weight field is a load balancing mechanism. When the priority field is the same for two or more records in the same domain, clients should try records with higher weights more often, unless the clients support some other load balancing mechanism.
- The Port field shows the port number used by the service on this host.
- The Target field shows the fully qualified domain name for the host supporting the service.

Appendix B – The Active Directory Database

Access to this database is accomplished almost exclusively through the *Directory SystemAgent* (ntdsa.dll)"

This database contains three tables.

Schema table - like any database, Active Directory has a schema which is a formal definition of the database structure and types of objects and attributes which can be contained in the database. The schema contains a list of all classes and attributes in the forest.

Link table contains linked attributes, which contain values referring to other objects in the Active Directory. For example, the MemberOf attribute on a user object contains values that reference groups to which the user belongs.

Data table represents users, groups, application-specific data, and any other data stored in the Active Directory. The data table can be thought of as having rows where each row represents an instance of an object such as a user, and columns where each column represents an attribute in the schema such as GivenName.

Active Directory consists of a directory and a location mechanism used to locate entries in the database. The database, stored in a file called *ntds.dit*, is managed by the *Extensible Storage Engine* (ESE), which is also used in Microsoft Exchange. That engine, in turn, is based on the *Microsoft Jet Engine*.

Appendix C – Group Types

Windows 2000 and Active Directory comes with a number of default groups. This list below is not an exhaustive list of all the default groups but it lists the groups which we are likely to use in a security context.

- Predefined Groups – these are global groups created automatically.
  - Enterprise Admins – automatically added to the Domain Admins group for each domain in the forest.
  - Domain Admins – automatically added to the Built-In Local Administrators Group of each member server.
- Built-In Groups – these are domain local groups used to manage domain controllers and Active Directory.
  - Adminstrators – automatically contains Domain Admins and Enterprise Admins as well as the Administrator account.
- Built-In Local Groups – these are domain local groups like Built-In Groups except that they exist on ordinary member servers to allow administration.
  - Administrators – automatically contains Domain Admins.
- Special Identity Groups – these are internally managed groups that can represent different users at different times.
  - Authenticated Users – any user with a valid user account.
  - Everyone – any users who accesses the computer