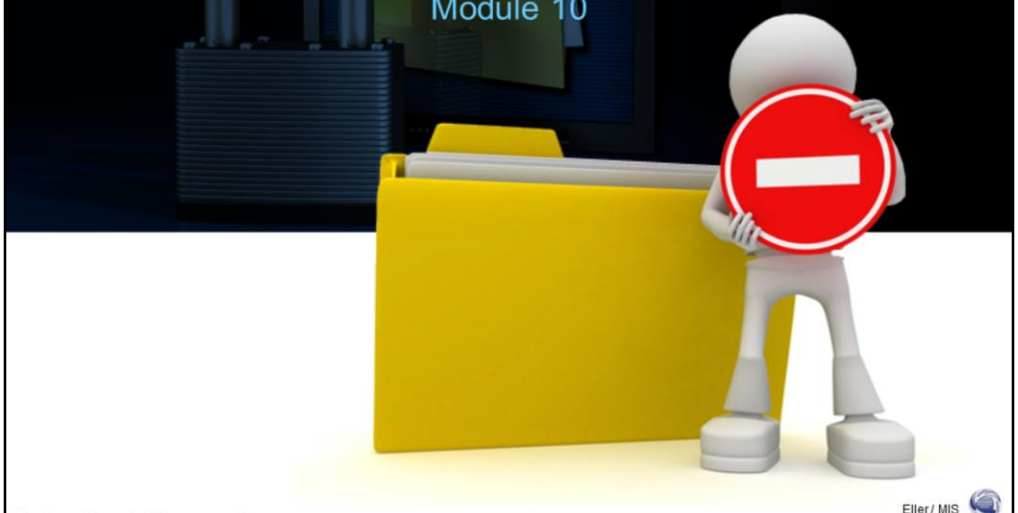
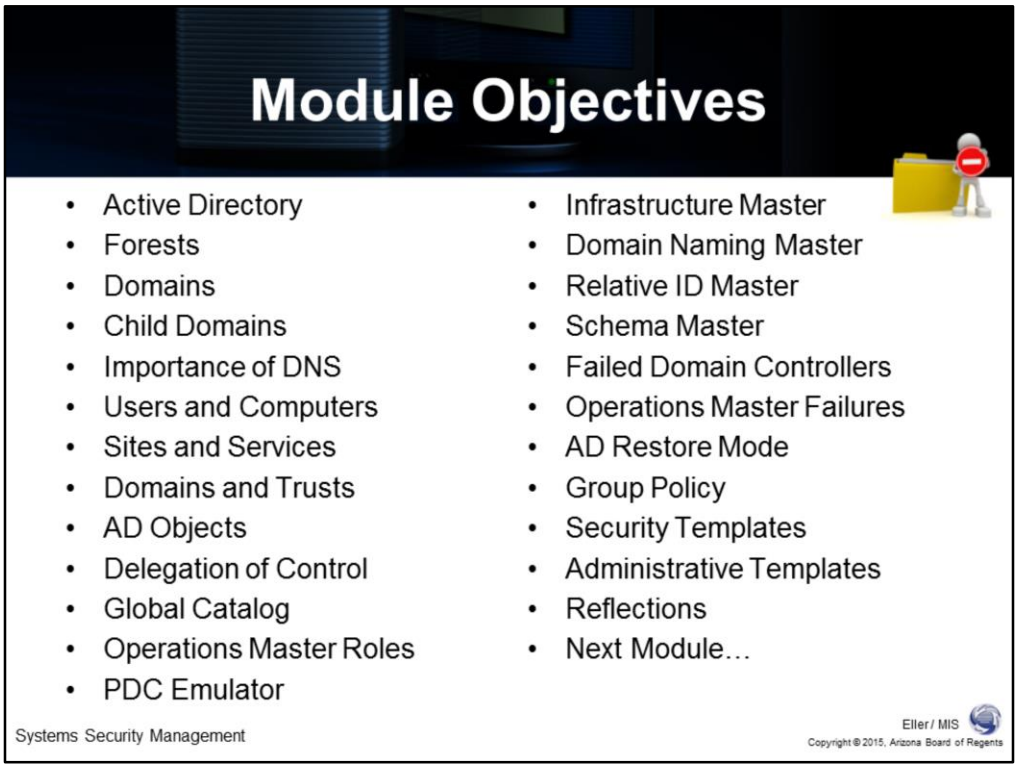
Microsoft Active Directory Services

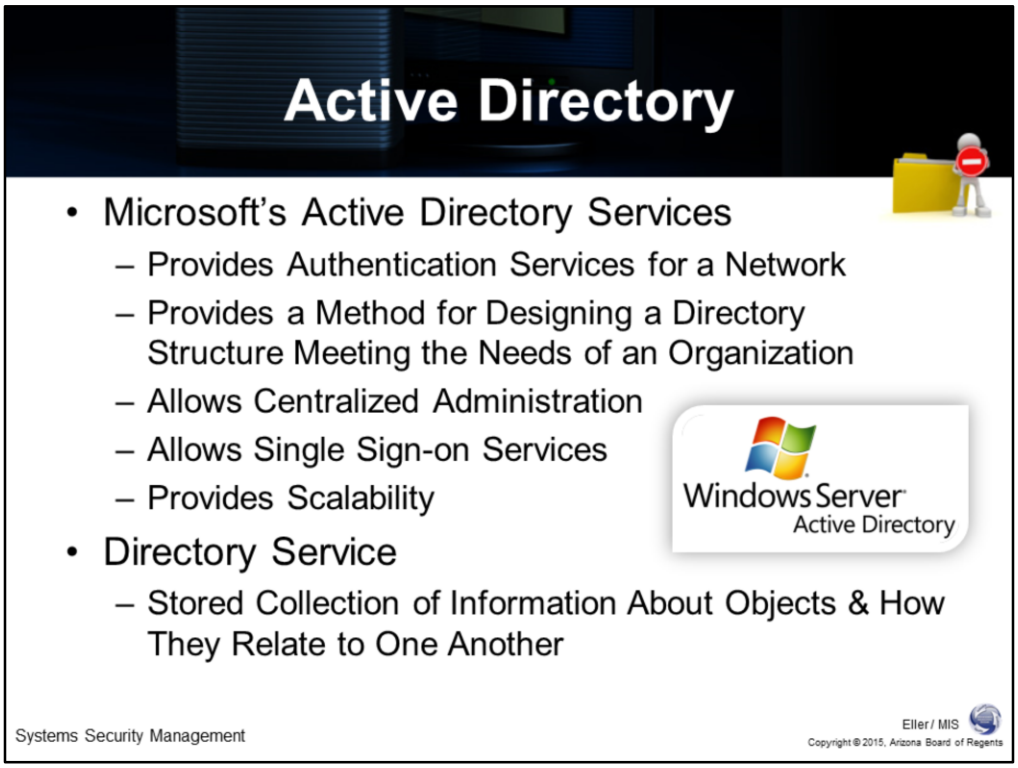
Module 10





At the end of the module, you should be able to:

- Describe what Active Directory is, why it is used, and how it works.
- Describe what a Forest, Domain, and Child Domain are and what their role in Active Directory is.
- Understand what the importance of Domain Naming Services is with Active Directory.
- Describe each Active Directory Tool and what they are used for.
- Explain each of the different object types available in AD and what they represent.
- Describe what a Global Catalog is and why it is necessary.
- Understand each of the five roles that play a major part of how AD works.
- Understand what is involved when an AD Domain Controller or Operations Master fails.
- Understand what is involved when restoring an Active Directory.
- Describe the various Security and Administrative Templates available for use with Active Directory.



Active Directory

Microsoft's Active Directory Services for Windows Server provides authentication services for a Windows-based network infrastructure. Ever since the days of Windows NT, Microsoft has provided server operating systems with the ability to create a network with a centralized mechanism for authentication. Active Directory was introduced with the Windows 2000 Server OS as a replacement for the Windows NT Security Accounts Manager (SAM) authentication structure.

In addition to providing authentication services, Microsoft has provided a variety of tools used to allow system administrators to design the structure of each Active Directory in order to meet the unique needs of the organization. This is accomplished through the creation and placement of various Active Directory objects, which we will discuss later in this lecture.

In terms of security, Active Directory allows an organization to adopt a solution for centralized administration of user accounts, passwords, and computers, among other options. Active Directory also provides an organization with the ability to utilize single sign-on services, making it easier for users to accomplish their daily work tasks. Finally, Active Directory provides scalability, allowing organizations to add Active Directory servers, objects, domains, and forests as necessary. This allows system administrators the ability to expand and contract the Active Directory as the organization changes over time.

In general, Active Directory is a Directory Service. Take a moment and think about what that might mean. What does a directory do? Well, in the simplest terms, a directory stores information, usually contact information. In actuality, this is very similar to how Active Directory works as a directory service, the main difference here is Active Directory not only stores contact information, it also stores a variety of information about various objects in the directory and how they relate to one another. In more technical terms, Active Directory is a complex relational database system.

Forests

Within Active Directory lies an overarching hierarchy that defines how systems and one or more AD domains communicate and work seamlessly together to accomplish goals. Within an AD forest can exist one or more independent domain trees.

Now, think about an actual forest in nature. Within a forest, you will find a wide variety of trees and other foliage which may or may not relate directly to one another, but each plays a distinct role within the forest itself. Active Directory forests have been designed in a manner mimicking nature in this regard. Here are some of the common characteristics of an AD forest:

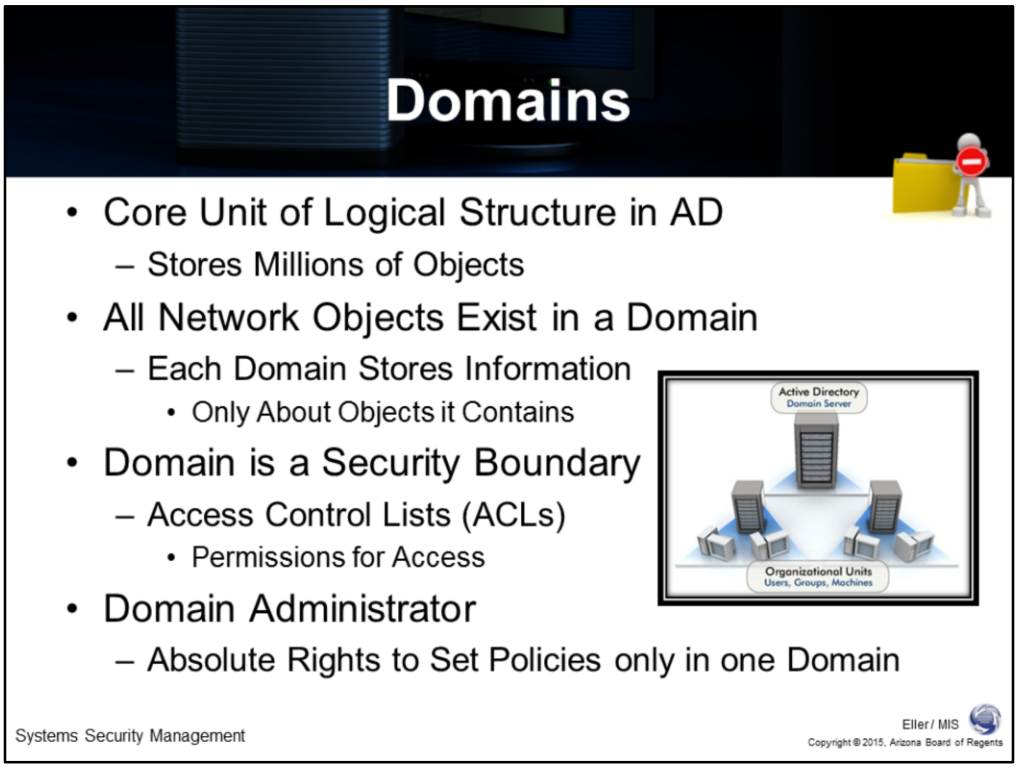
All trees share a common schema. In other words, domain trees will all share a common set of underlying rules. These rules will define how trees within the forest are structured. So, going back to the actual forest metaphor, if the common terrain is rocky, then you will only see trees capable of growing within that specific environment.

Trees have different naming structures according to their domains. As with a natural forest, domain trees can not only share the same naming structure as other trees, they do not have to follow this precisely. Other domain trees with different naming structures can exist within the same forest, much like different species of trees can exist in a real forest.

All domains share a common global catalog. In general, every domain tree within a forest is catalogued in order to allow computers and servers to locate and share resources with one another. Think of a global catalog as a map of the forest, without which you might become lost.

Domains operate independently, but forests allow communication across the entire organization. Each domain tree within a forest is able to expand and contract as necessary without affecting other trees within the same forest; however, the forest itself will allow seamless communication between domains and trees when necessary.

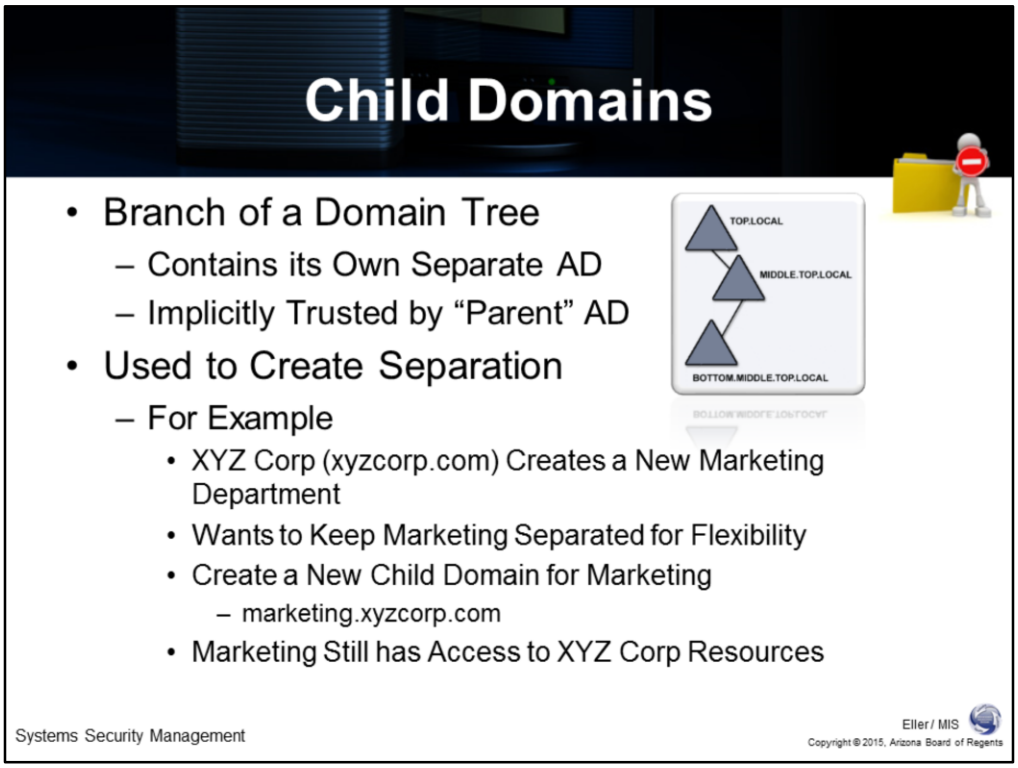
An implicit two-way trust exists between domains and domain trees. When working within an Active Directory forest, in order to have seamless communication between different domains and trees a trust relationship must exist. In other words, with multiple forests, one tree will not trust another automatically. In this situation a trust relationship must be created manually; however, in the single forest example, different domains and trees will automatically trust one another. It is these trust relationships which allows the resources in one domain to be made available to users of another domain within the same organization.



Domains

Domains were mentioned throughout the discussion of forests, for without a domain tree, a forest cannot exist. In general, a domain is a core unit of logical structure within an Active Directory. Think of a single domain as a warehouse capable of storing millions of different objects that define specific attributes of the domain. So, for example, all network objects exist within a domain, and the domain will only store specific information about the objects it contains. So, in a two-domain forest, each domain will have network objects defining how the network for each domain is structured; however, neither domain will contain information about the network objects of the other domain, even though both are members of the same forest.

A domain also acts as a security boundary for a network. Access control lists, which we will discuss more in-depth in Module 10, will be used within domains to control specific permissions for accessing network resources. In addition, a domain administrator will have absolute rights to set policies and permissions within a single domain. Each domain will have a different domain administrator, and one domain's administrator cannot modify policies and permissions within another domain without being given explicit permission by that domain's administrator.

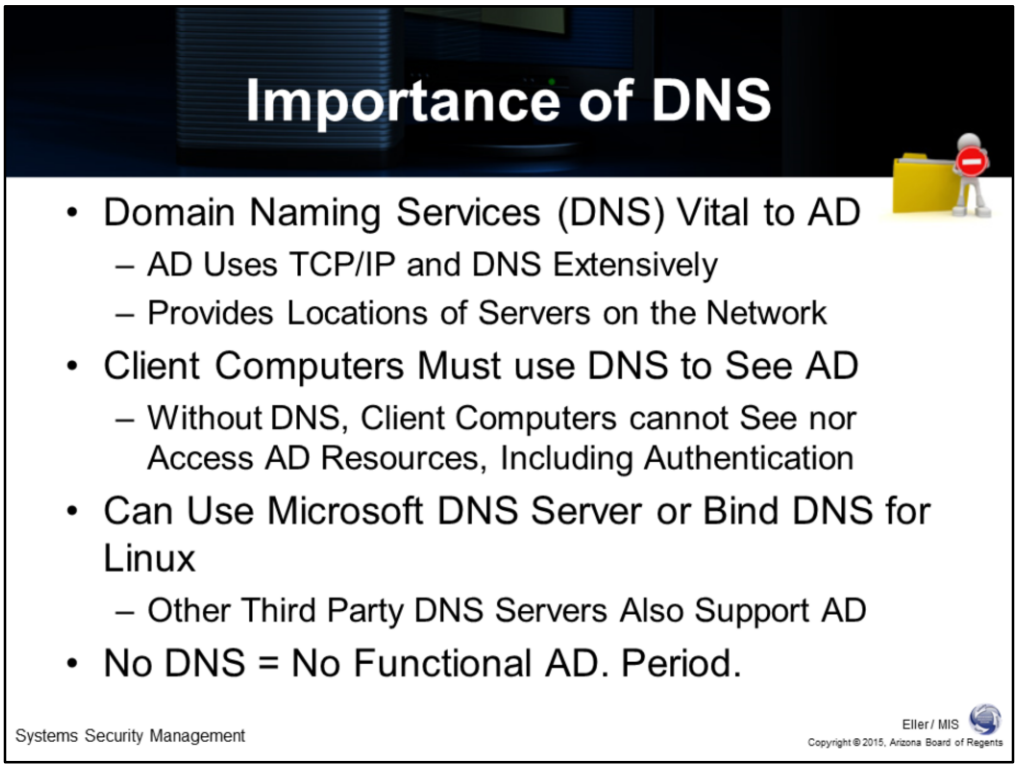


Child Domains

Child domains are very similar to a regular domain. Consider a real tree that has multiple branches. A child domain would exist within a domain tree as a branch of the existing domain. Child domains cannot exist within a domain tree or forest unless a "parent" or "top-level" domain already exists. When a child domain is created, it will have its own separate Active Directory, and since it can only be created within an existing domain tree, the "parent" domain will implicitly trust the child domain when sharing resources.

Child domains are usually created in order to provide a logical separation on a network. Let's take a quick look at one possible example of the need to create a child domain.

The XYZ Corporation, whose chosen domain is xyzcorp.com, has created a new Marketing Department. The company wants to keep Marketing separated from the rest of the company for potential flexibility in the future. In order to accomplish this on the network, the System Administrator creates a child domain for the Marketing Department. This would make the marketing department domain named marketing.xyzcorp.com. Using this structure, Marketing is its own separate entity on the network; however, it will still have access to other XYZ Corp network resources.



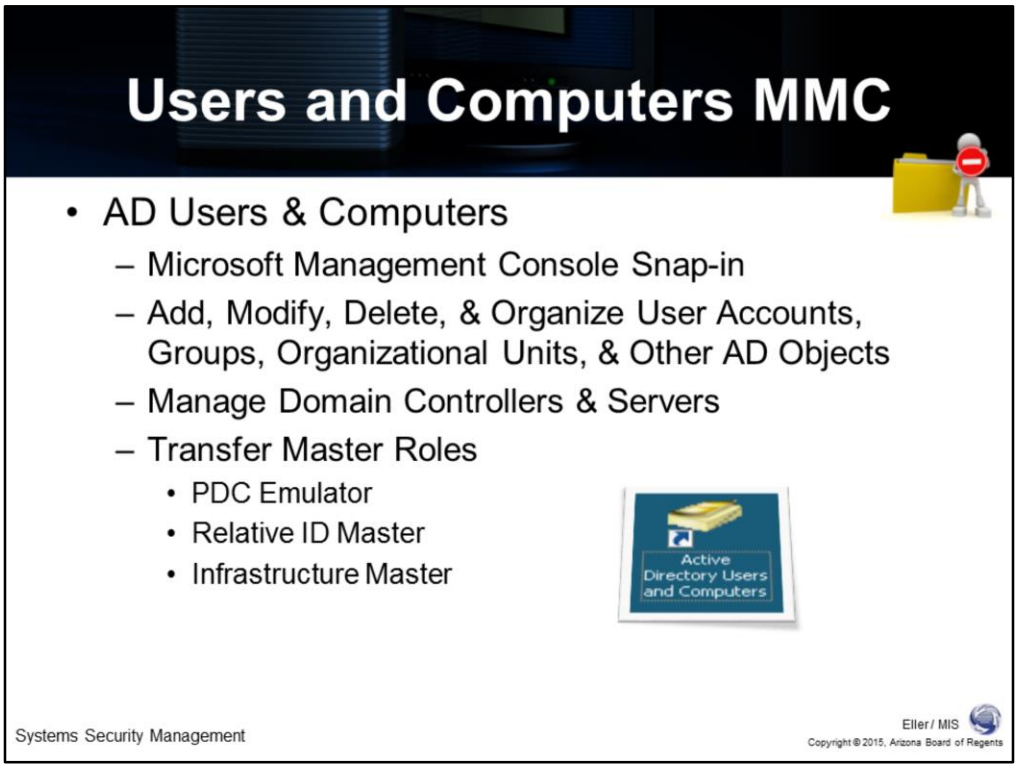
The Importance of DNS

Domain Naming Services, or DNS, is vital to Active Directory, as an Active Directory uses the TCP/IP protocol and DNS extensively for network communications. DNS servers are designed to provide a means for translating domain names (such as xyzcorp.com) from a friendly name into an IP Address. As such, Active Directory will use DNS to provide the locations of servers on the network to client computers and other servers.

Client computers absolutely must use DNS in order to see and communicate with Active Directory servers on the network (in general, client computers will need to use DNS in order to do anything on both a local network and the Internet). Without a defined DNS server on the network, client computers will not be able to access Active Directory resources, including authentication services. So, client computers would, in essence, be unusable in an Active Directory environment without the use of DNS.

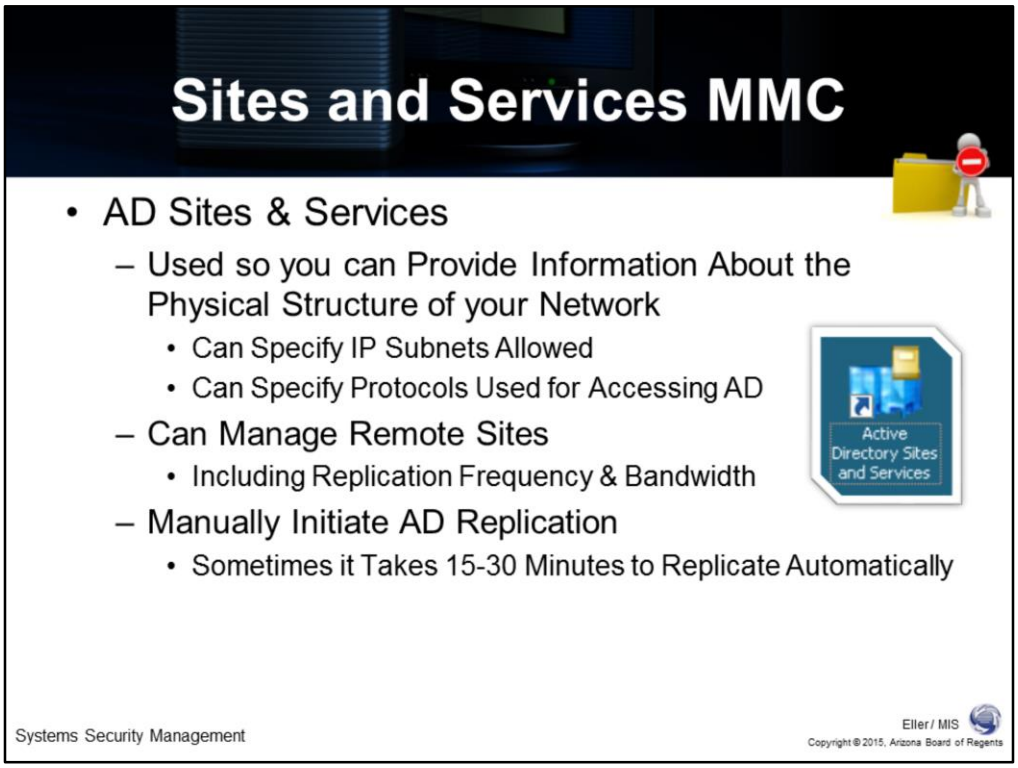
Active Directory has been designed to make use of a variety of DNS server options. Microsoft offers a DNS server component within Windows Server that is designed specifically to work with Active Directory. Another alternative DNS server option is Bind DNS for Linux servers. While these are the two most popular options for use with an Active Directory, other third party solutions may also support AD.

The bottom line with Active Directory is without a DNS server, the AD will not be functional. Period.



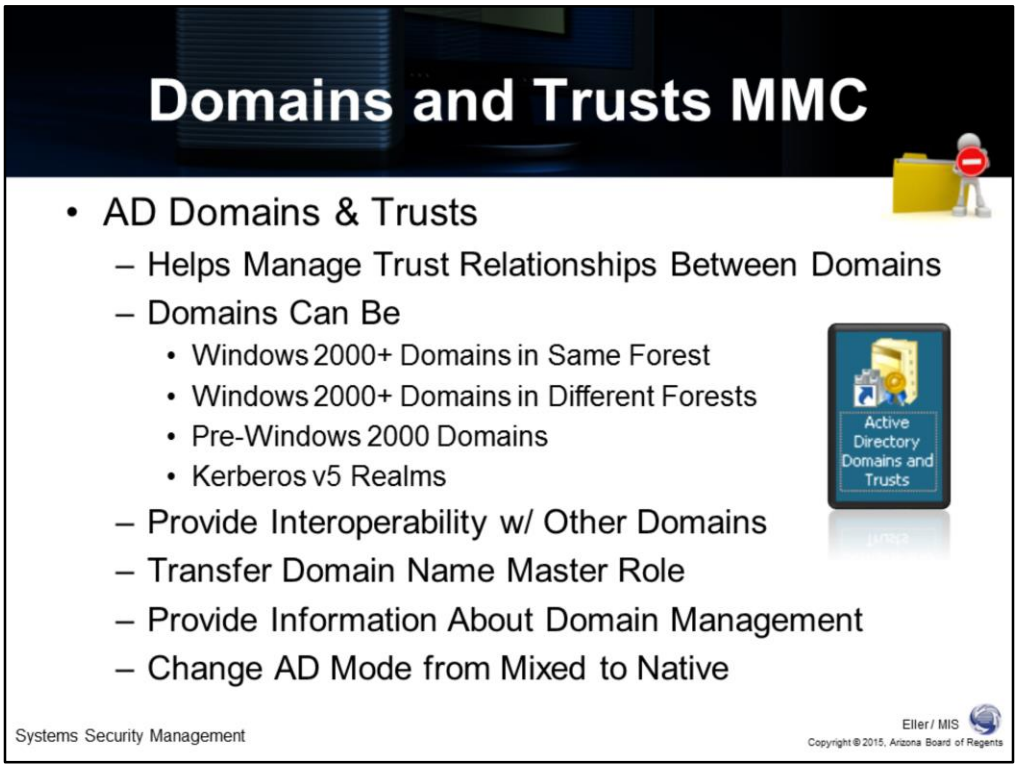
Active Directory Users and Computers

In order to interface with an Active Directory server, Microsoft has created three different tools which use the Microsoft Management Console, or MMC. The first of these tools is the one a System Administrator will spend most of their time working with: Active Directory Users and Computers. This particular tool is designed to allow a SysAdmin to add, modify, delete and organize user accounts, security groups, organizational units, and other AD objects. SysAdmins will also use this tool to manage domain controllers and other Microsoft servers. The last component of this tool involves the ability to transfer specific master roles within AD, including: the PDC Emulator, the Relative ID Master, and the Infrastructure Master. We will discuss each of these roles later in this module.



Active Directory Sites and Services

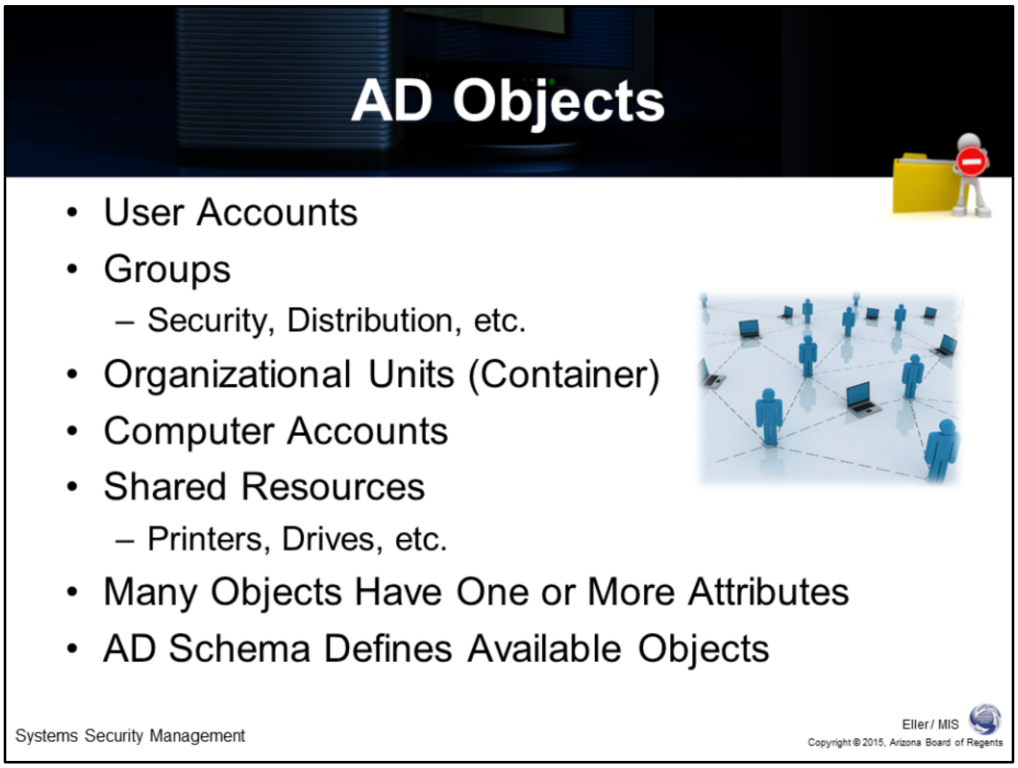
The second most common AD tool used by SysAdmins is the Active Directory Sites and Services tool. This tool is used by SysAdmins in order to provide information about the physical structure of the network, including the specific IP subnets and network protocols allowed to access the AD. Sites and Services also allows for the management of remote sites, including the frequency by which an Active Directory is replicated to other AD domain controllers and how much bandwidth the system can use for replication. Finally, the most important aspect of this tool involves the ability to manually initiate AD replication. Whenever you have an environment with multiple domain controllers, it can take anywhere between 15 and 30 minutes for changes to be replicated to other controllers. So, for example, if you create a user account for someone and they try to authenticate against a different domain controller that has not yet received the updated AD information, that user will be denied access. Sites and Services allows a SysAdmin to manually initiate an AD replication, which will immediately allow the user to login with their new user account information.



Active Directory Domains and Trusts

The third tool is Active Directory Domains and Trusts. This tool's primary use is the management of trust relationships between domains. Now we have spoken already about the implicit two-way trusts that are created between domains in a forest. While this tool can be used to manage these relationships, it is also used to manage manual trust relationships between domains in multiple forests, without which the domains would be unable to share resources. An example of when this might be used would be after one company merges or acquires another company and they need to create a network environment where resources are shared between the two previously separated companies. Depending on the needed trust relationship, Domains and Trusts can be used to manage relationships with Windows 2000 (and beyond) domains in the same or different forests, pre-Windows 2000 domains (such as Windows NT domains), and Kerberos v5 Realms.

Other uses for Domains and Trusts involve providing interoperability with other domains, the ability to transfer the Domain Naming Master Role, and to provide information about domain management. It can also be used to change the Active Directory mode from mixed to native. What this means is in a mixed mode Active Directory, you can have domain controllers that use Windows Server 2000, 2003, and 2008 without any problems. By changing to native mode, you are required to use only the highest version of Windows Server you have. So, if you have a Windows Server 2008 domain controller, then in native mode, all domain controllers must run Windows Server 2008.



Active Directory Objects

We have briefly touched on the use of objects within Active Directory. The term object comes from the relational database nature of Active Directory itself. Virtually everything within an Active Directory is an object. Having said that there are specific objects which can be created and used to design and make use of an AD. These include the following:

User Accounts – Used for authentication purposes.
Groups – There are two general types of groups: security and distribution. Security groups are used to assign permissions to specific resources, while distribution groups are used for mailing lists.
Organizational Units – These are containers within an AD and used to provide structure. They are depicted as folders within the Active Directory Users and Computers tool.
Computer Accounts – These are similar to user accounts. When you add a computer to an Active Directory domain, a computer account is created and used as a second layer of authentication.
Shared Resources – Anything that can be shared on a network can be added to the Active Directory. This includes network printers and shared network drives.

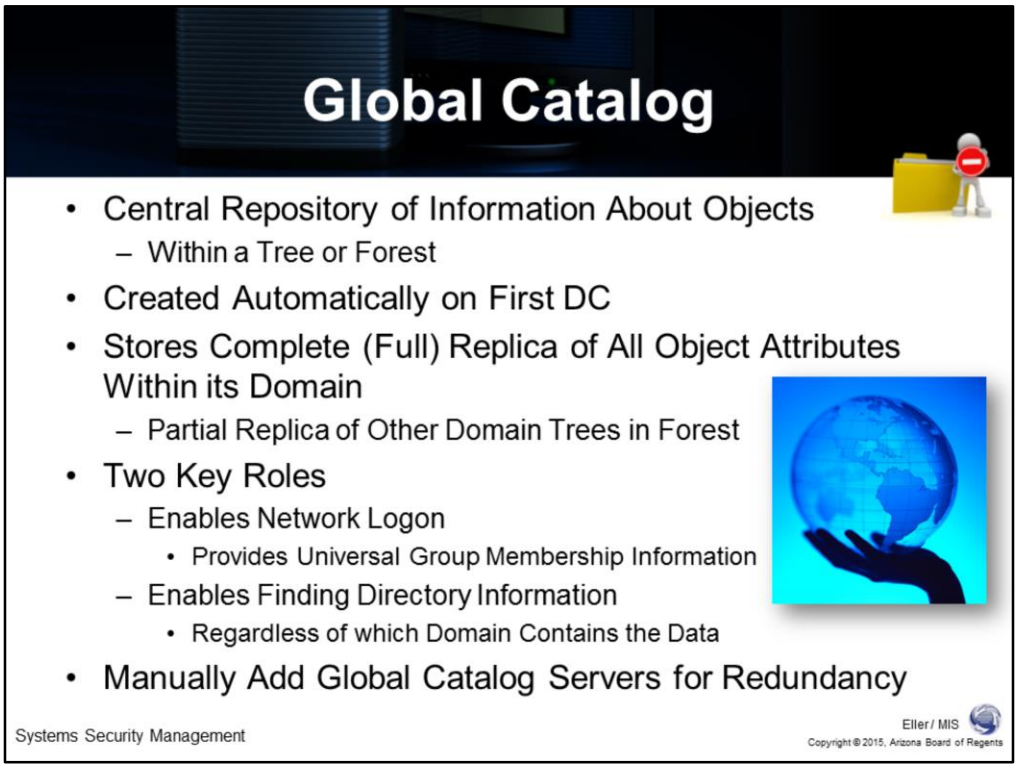
While there are a number of objects that can be created in AD, each object has one or more attributes which can be used for various purposes. For example, when creating a user account you can specify an e-mail address for that user, which can then be accessed by distribution groups in AD, Microsoft Exchange, or SharePoint servers for mailing purposes. Which attributes are available is defined specifically by the Active Directory Schema.

Delegation of Control

How does an organization deal with remote management of other company sites? Perhaps even sites that are located in other cities, states, or even countries? The answer is delegation of control within the Active Directory.

You are the System Administrator for XYZ Corporation and as such, you are in control of the AD. The company decides the time is right to expand into Europe, so they invest in opening a new location there. As the SysAdmin, you can create an Organizational Unit for the new Europe location, within which you store user accounts and other AD objects and settings for the remote site.

Now, in order to provide adequate technical support for the Europe location, XYZ Corp hires a SysAdmin for that location. You can delegate control of the Europe OU to the new SysAdmin. This allows the new SysAdmin to do everything you can with, or without, restrictions, but ONLY within the Europe OU. This allows the Europe SysAdmin to perform his or her job function and prevents modifications to the XYZ Corp main AD information.

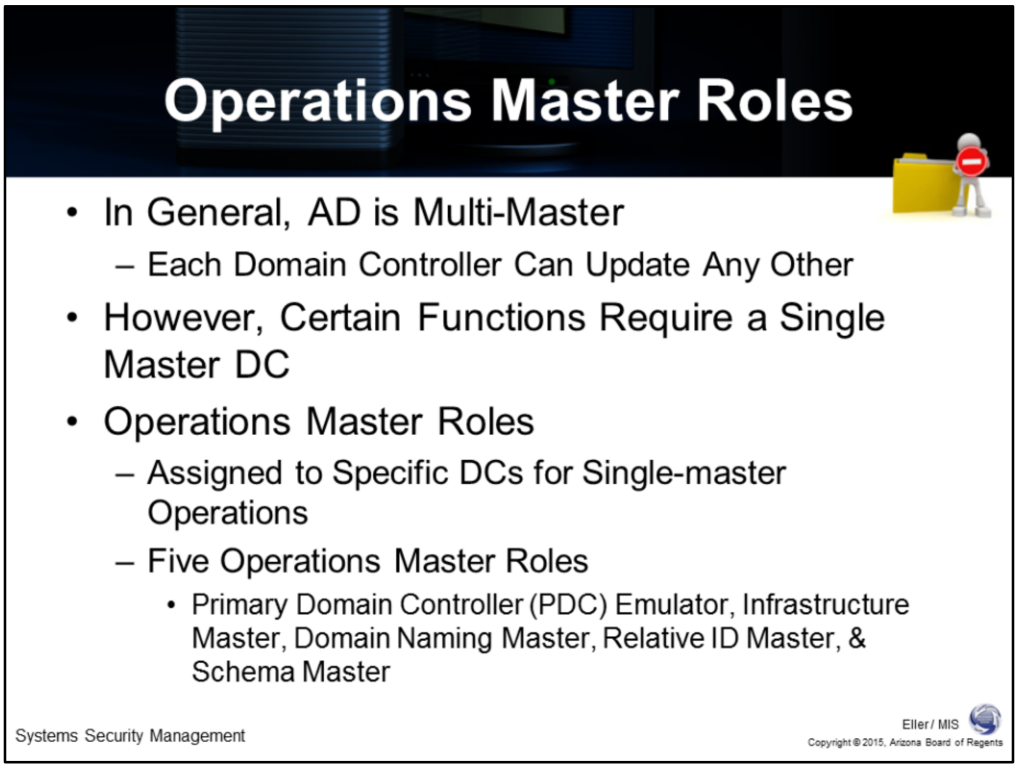


Global Catalogs

A global catalog is a central repository of information about objects stored within an Active Directory forest. As mentioned previously, think of the global catalog as a map of locations within a natural forest. Global catalogs are created automatically on the first domain controller within an AD forest and stores a complete (or full) replica of all object attributes within its domain. If other domain trees exist in the same forest, then the global catalog will also contain a partial replica of the object attributes contained within other domain trees in the forest.

There are two key roles a global catalog server is responsible for: enabling network logons and finding directory information. By providing network logons, the global catalog will allow for the creation and use of Universal Groups, security groups used to allow resources to be accessed between domains. The global catalog also allows for finding directory information regardless of which domain contains the data.

It is very important to understand that by default, only the first domain controller is configured as a global catalog. Additional domain controllers can be manually specified as global catalog servers through the use of the Sites and Services tool. It is strongly recommended that SysAdmins have at least two global catalog servers in each domain in order to provide redundancy. If the only global catalog server were to fail, none of the computers within the AD domain would be able to communicate effectively on the network. It is possible for the AD to reconstruct this information by specifying a second global catalog after the original fails; however, this process takes anywhere from 24-48 hours to complete, during which time there will be network communication issues. The best course of action is to ensure redundancy before a potential failure might occur.

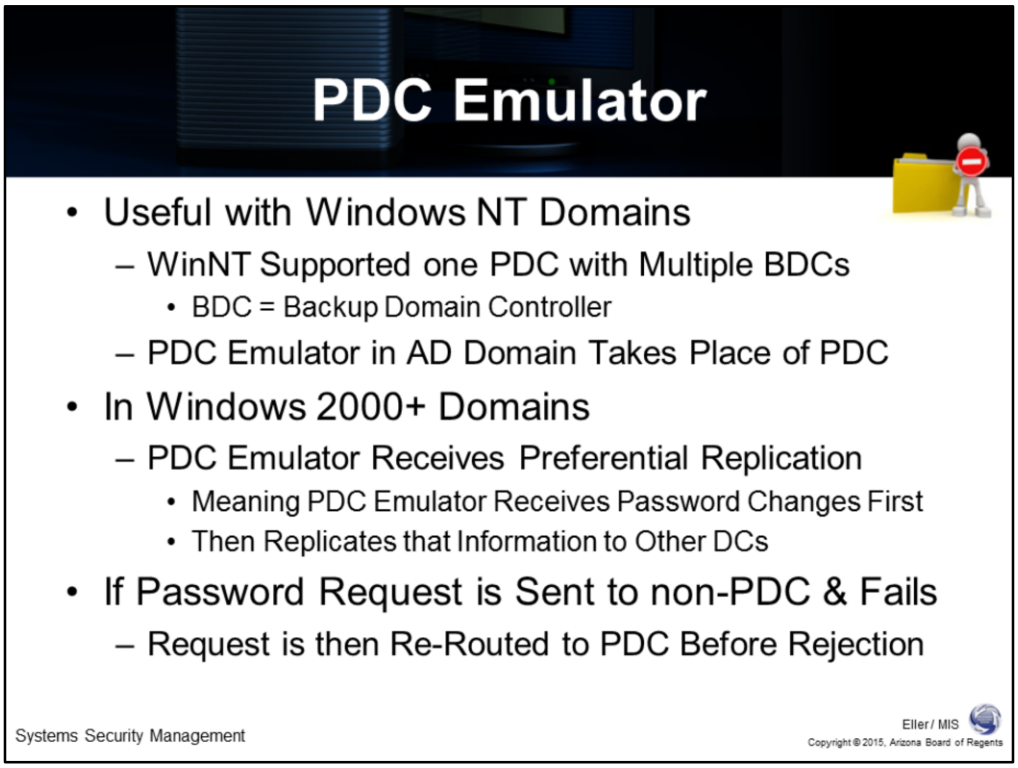


Operations Master Roles

In general, Active Directory is multi-master. This means each and every domain controller has the ability to update any other. However, there are certain functions within AD which require a single Master Domain Controller. This is where the Operations Master Roles come into play.

Operations Masters can be assigned to specific domain controllers for single-master operations; however, by default, the first domain controller within a domain tree will control all of the operations master roles. These roles should be spread out among domain controllers in order to ensure minimal issues should one domain controller fail.

There are five operations master roles, each of which have been previously mentioned, and will be discussed further. These roles include: the primary domain controller (PDC) emulator, the infrastructure master, the domain naming master, the relative ID master, and the schema master.

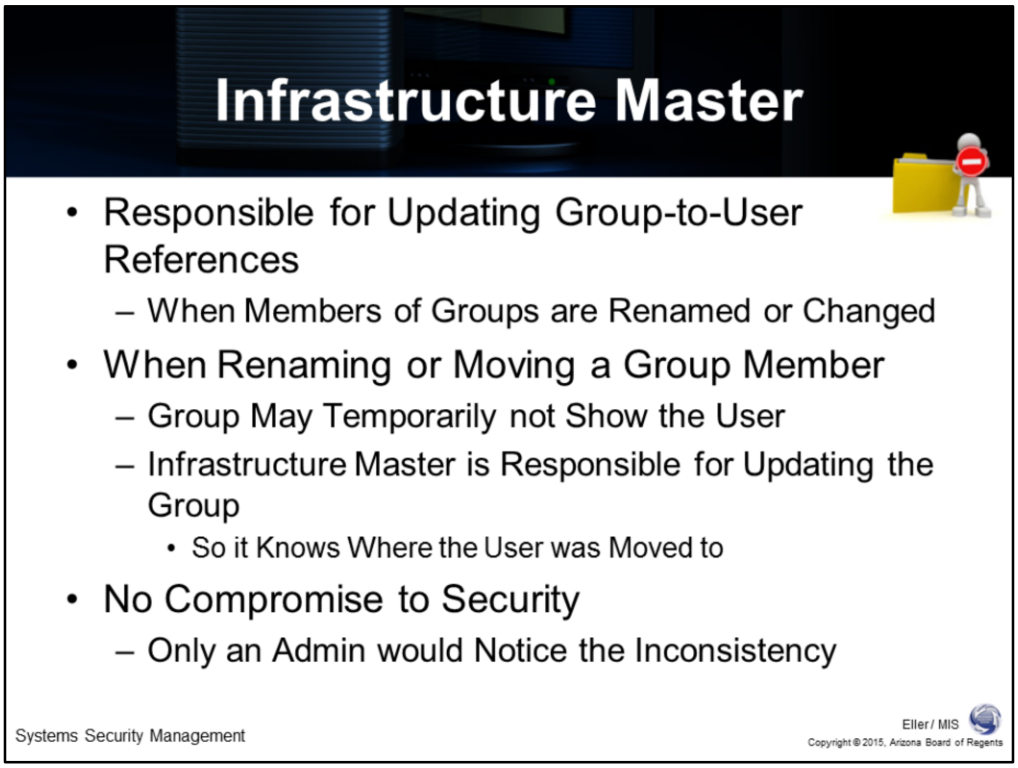


Primary Domain Controller (PDC) Emulator

The PDC emulator has ties back in the original domain structure of Windows NT domains. With Windows NT networks, a domain supported multiple domain controllers for redundancy; however, it only supported a single primary domain controllers. All other domain controllers were called Backup Domain Controllers (or BDCs). The idea with this setup was that you could make a change to any one BDC and that server would send the update to the PDC. The PDC would then disseminate the information to the other BDCs.

In Windows 2000 and beyond, the PDC Emulator was created in order to replace the PDC in the domain structure. The PDC emulator receives preferential replication of information within the domain, meaning this particular domain controller receives password changes first then replicates that information to the other domain controllers. So, is this all the PDC emulator does? The answer is no. Another example occurs when someone attempts to login after changing their password. If the password is sent to be authenticated against a domain controller that is not the PDC emulator master and it fails, that domain controller will forward the authentication request to the PDC emulator master for verification. The password sent is then either authenticated or rejected, then the user is notified either by successful login or an incorrect password message.

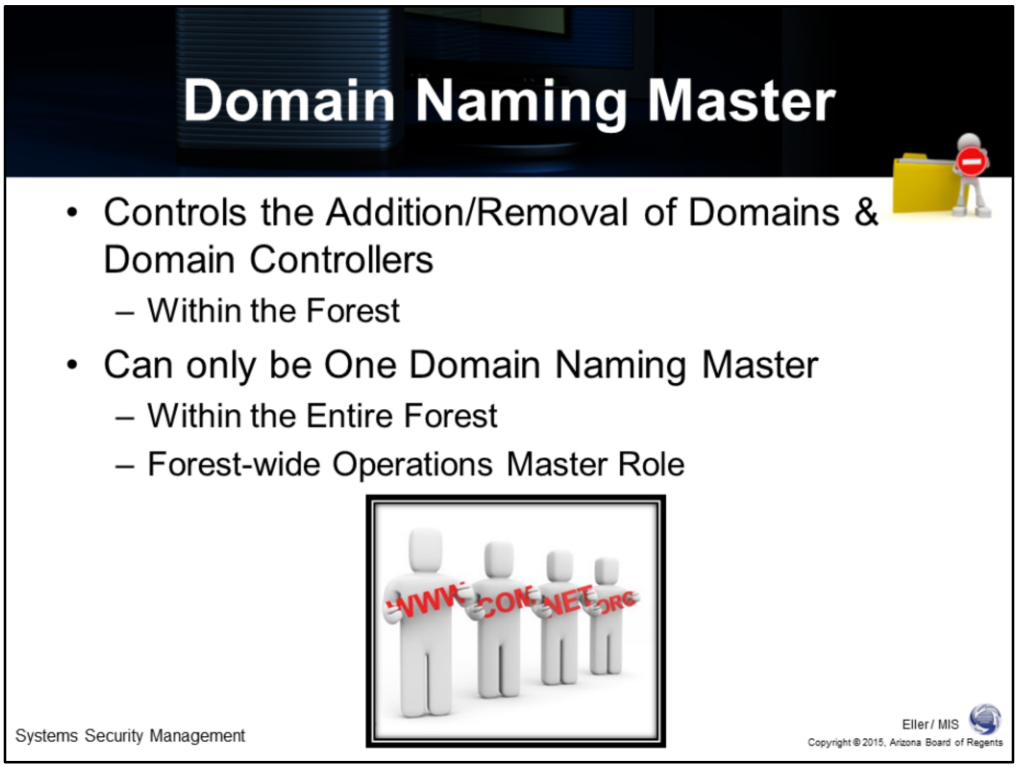
The PDC emulator master role is a domain specific role, meaning there will be a PDC emulator master server in every domain within a forest.



Infrastructure Master

The Infrastructure Master is responsible for updating group-to-user references in the Active Directory. So, if you were to change the name of a group, or add or remove users from a group, the Infrastructure master would be responsible for updating the other domain controllers with this information. If you were to rename or move a group or group member, during the time the domain controller takes to replicate this information, the other servers may temporarily not show the change. It is the infrastructure master that is responsible for updating the group membership so the domain controllers know where the user was moved to in the Active Directory. It is important to note that this is not a compromise to the security of the system as only a SysAdmin actively working on the domain controllers would notice the inconsistency.

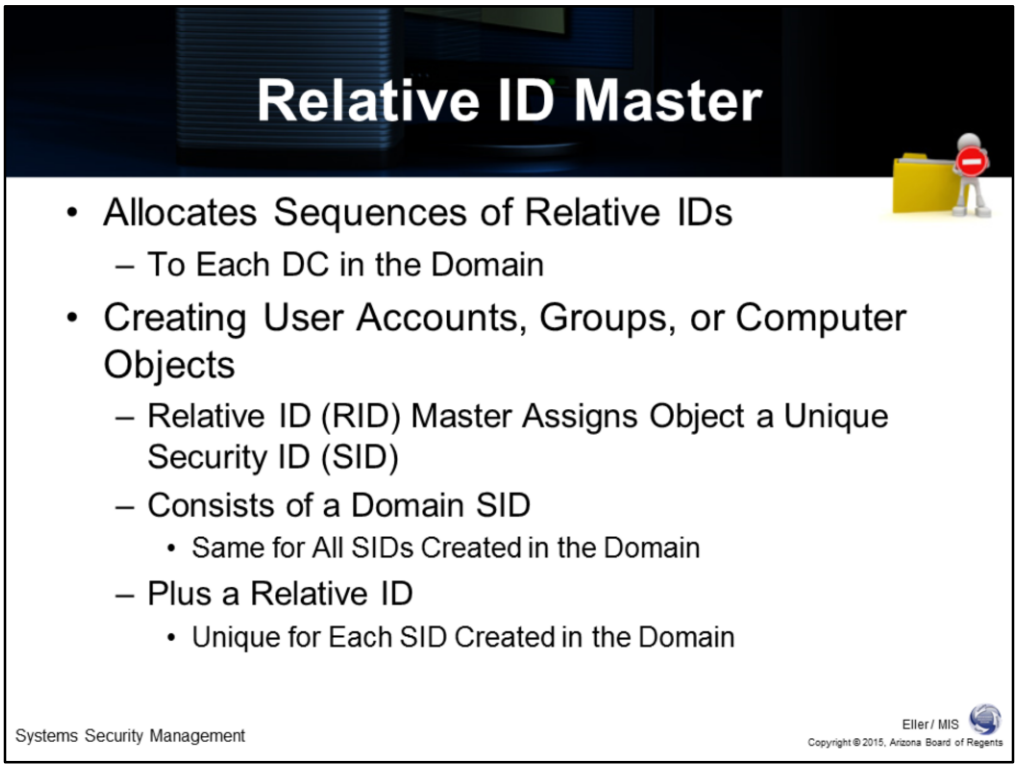
The infrastructure master role is a domain specific role, meaning there will be an infrastructure master server in every domain within a forest.



Domain Naming Master

The Domain Naming Master controls the addition or removal of domains and domain controllers within a forest. In other words, after you create an Active Directory domain, the domain naming master is responsible for allowing the creation or removal of additional domain controllers, child domains, and other domain trees all within the same forest.

It is important to note that there will only be a single domain naming master throughout the entire forest. Unlike the PDC emulator master and the infrastructure master, which can one server each within every domain within a forest.

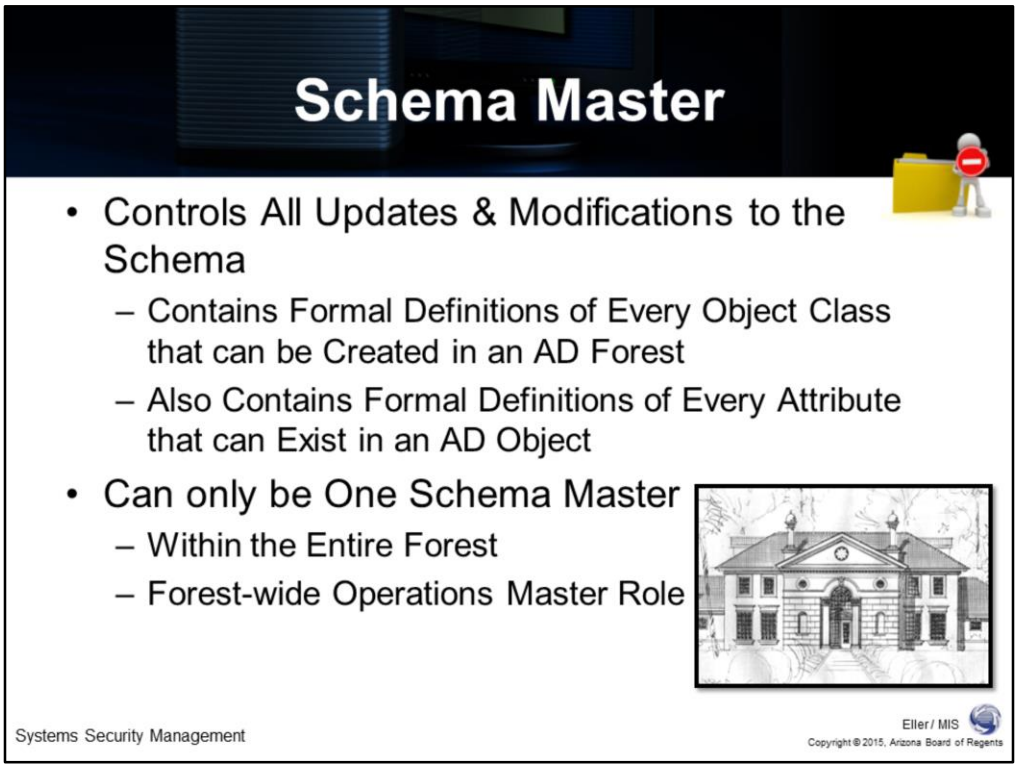


Relative ID Master

The relative ID master is responsible for allocating sequences of relative IDs to each domain controller in a domain. Relative IDs are a string of numbers and letters which are used to uniquely identify objects within an Active Directory. Every object in Active Directory is assigned a security ID that is connected to its common name. Without a collection of Relative IDs to assign, the Active Directory would not allow any additional objects to be created.

So, when creating a user account, group, or computer object, the relative ID (RID) master will assign the object a unique security identifier (SID). The entire security identifier will consist of two parts. The first part is a domain security identifier, which is the same for all SIDs created within the domain. The second part is a relative identifier, which is unique for each SID created within the domain.

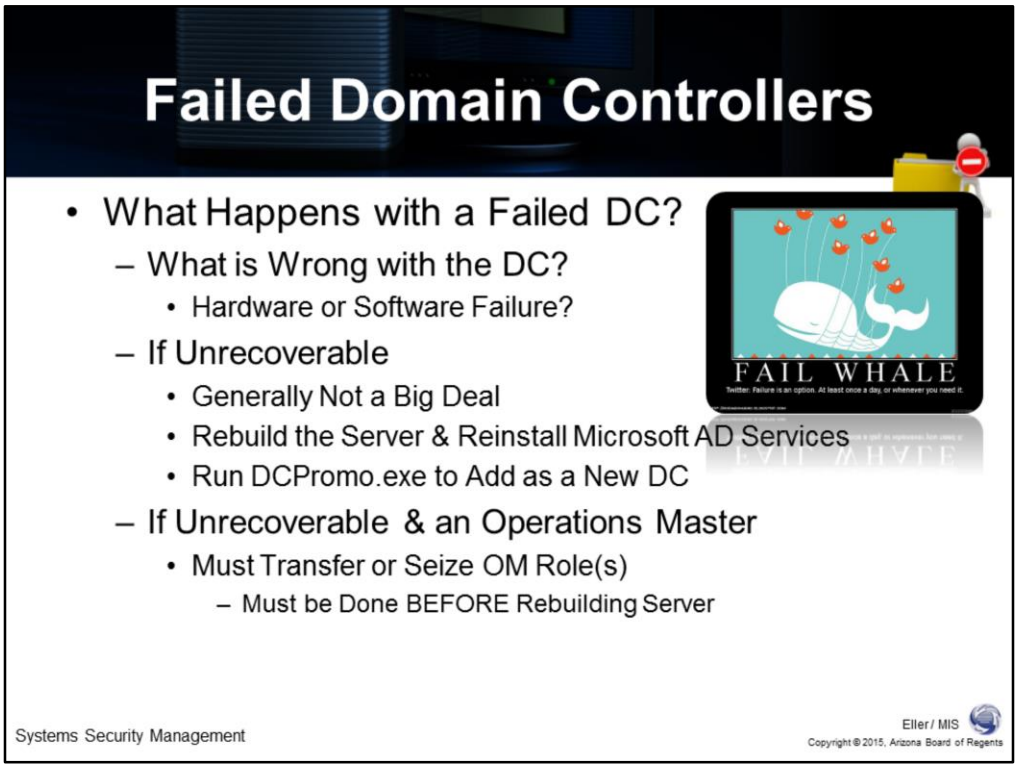
The relative ID master role is a domain specific role, meaning there will be a relative ID master server in every domain within a forest.



Schema Master

The schema master role is responsible for controlling all updates and modifications to the underlying Active Directory schema. The schema contains formal definitions of every object class that can be created in an AD forest and formal definitions of every attribute that can exist in an AD object. By default, the AD Schema tool is hidden by default on an Active Directory server, and this is because it is not recommended that anyone modify the schema manually as it can cause irreparable damage to the Active Directory. The only reason to use the Schema tool is to transfer the Schema Master role to another server if deemed necessary.

It is important to note that there will only be a single schema master throughout the entire forest. Unlike the PDC emulator master, infrastructure master, and relative ID master, which exist as one server each within every domain in a forest.

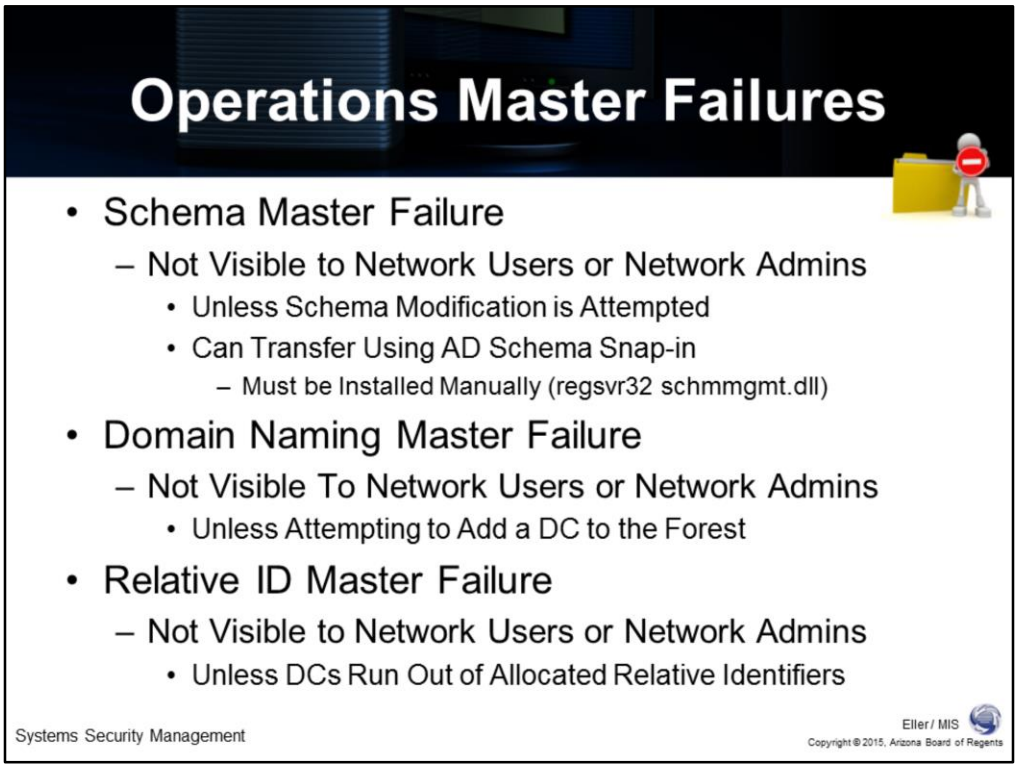


Failed Domain Controllers

Now, what happens when a domain controller fails? Did the server's hardware fail? Or is there a software problem? The SysAdmin must determine the cause of the problem, its severity, and how likely it will be that the server can be recovered.

In the event the server is unrecoverable this is generally not a big deal. The SysAdmin will need to rebuild the server by reinstalling Windows Server and adding the server as an additional domain controller in the Active Directory. Adding a domain controller is as simple as running the DCPromo.exe command.

In the event a server is unrecoverable and IS an operations master, the operations master role must be transferred or seized BEFORE rebuilding the server. This is necessary in order to keep the domain operational and to allow another domain controller to recreate the missing operations master information. If the server is rebuilt with the same name as the failed operations master server and the role was not transferred or seized, this can cause the active directory to not work properly. The key here is to know which servers are responsible for which operations master roles and to transfer or seize roles in the event of a failure before rebuilding the server.



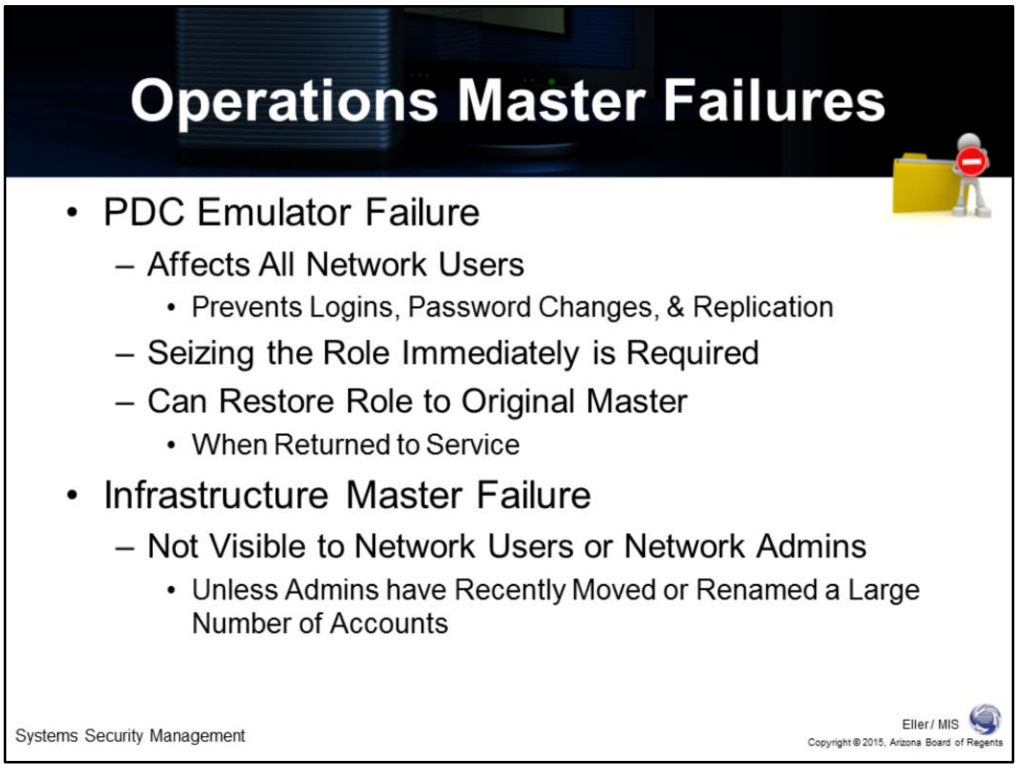
Operations Master Failures

When an operations master server fails, there are varying degrees to which the failure is noticeable on the network.

When a schema master server fails, this is not visible to network users or network admins unless a schema modification is attempted. Schema modifications usually occur when an Active Directory server is upgraded from mixed to native mode or the forest functionality is raised from an older version of Windows Server to a newer version, for example, from Windows Server 2003 to Windows Server 2008. In order to transfer this role to another server a SysAdmin must transfer using the AD Schema MMC snap-in, which must be installed manually using the command regsvr32 schmmgmt.dll.

In the event of a domain naming master failure, this is also not visible to network users or network admins unless a SysAdmin is attempting to add a domain controller to the forest.

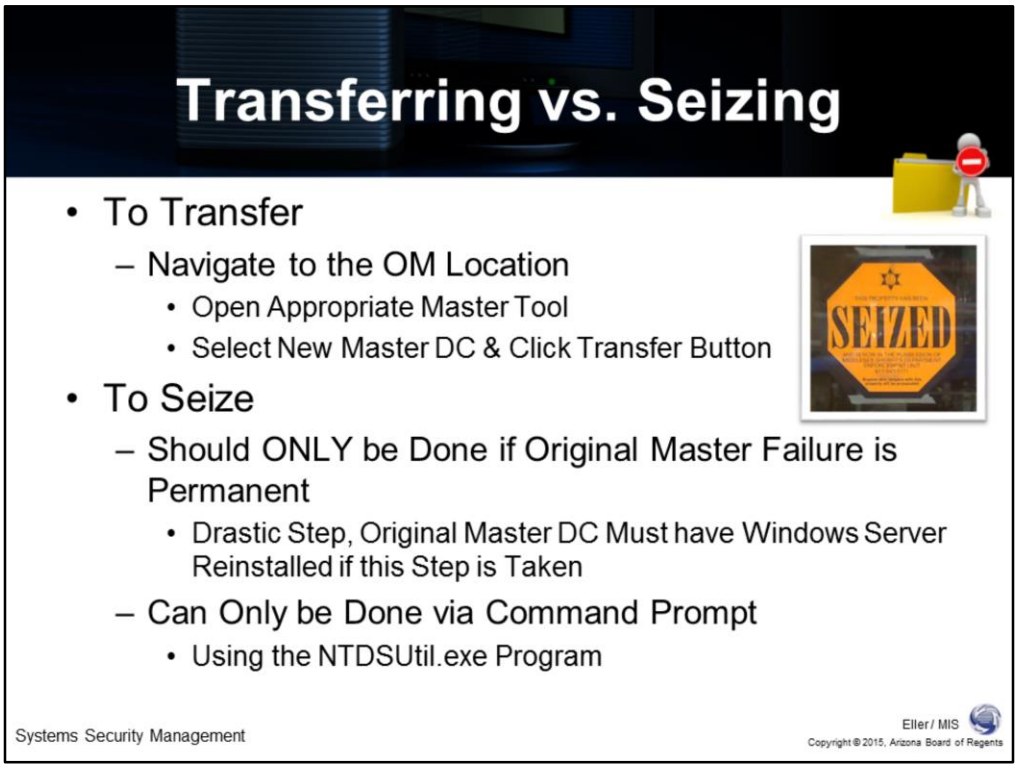
With a relative ID master failure, the failure is not visible to network users or network admins unless the domain controllers run out of allocated relative identifiers. A RID master server allocates a random number of RIDs to domain controllers when requested, so this failure may take some time to notice.



Operations Master Failures (continued)

A PDC emulator failure will affect all network users as this will prevent user logins, password changes, and active directory replication. If the server with this role fails, the immediate seizure of the role is required. When seized the failed server cannot be repaired, it MUST be rebuilt in order to prevent future problems. Once the server is returned to service, the role can be transferred back to the original master.

With an infrastructure master failure, the failure will not be visible to network users or network admins. The only time this kind of failure will be noticed by a SysAdmin is when they attempt to move or rename a large number of accounts. The changes will not be replicated if the infrastructure master has failed.



Transferring vs. Seizing Roles

We have spoken about transferring and seizing roles. But how do you know when to transfer, when to seize, and how to do either?

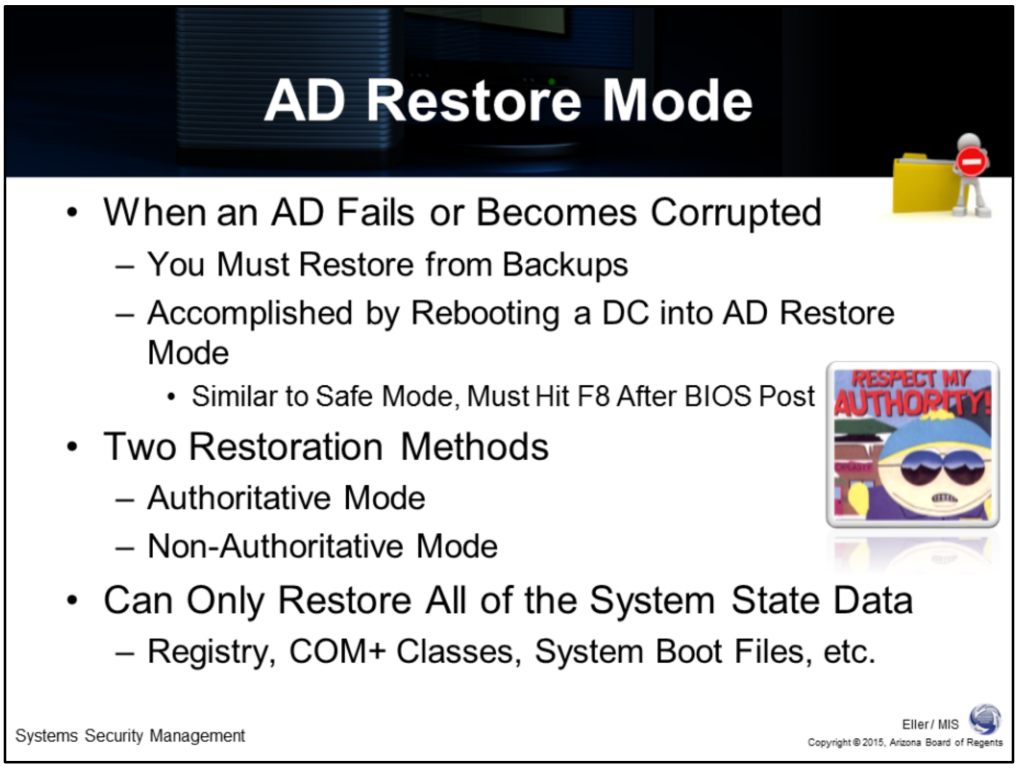
To Transfer Roles

In order to transfer a role, the SysAdmin must navigate to the specific operations master location by opening the appropriate master tool on the server the role should be transferred to. The PDC Emulator, RID Master, and Infrastructure Master roles can be transferred using the AD Users and Computers tool. The Domain Naming Master role can be transferred using the AD Domains and Trusts tool, and the Schema Master role can be transferred using the AD Schema Tool. Once the tool has been selected, the SysAdmin can transfer the role by opening the tool and clicking the Transfer button.

To Seize Roles

Seizing a Master role should ONLY be done if the original master failure is permanent. If there is any chance of repairing the failed master server, seizure should not be considered unless there will be an extended delay and the role to seize is not the PDC emulator role. Remember, seizing a role is a drastic step and the original master domain controller must have Windows Server reinstalled if this step is taken.

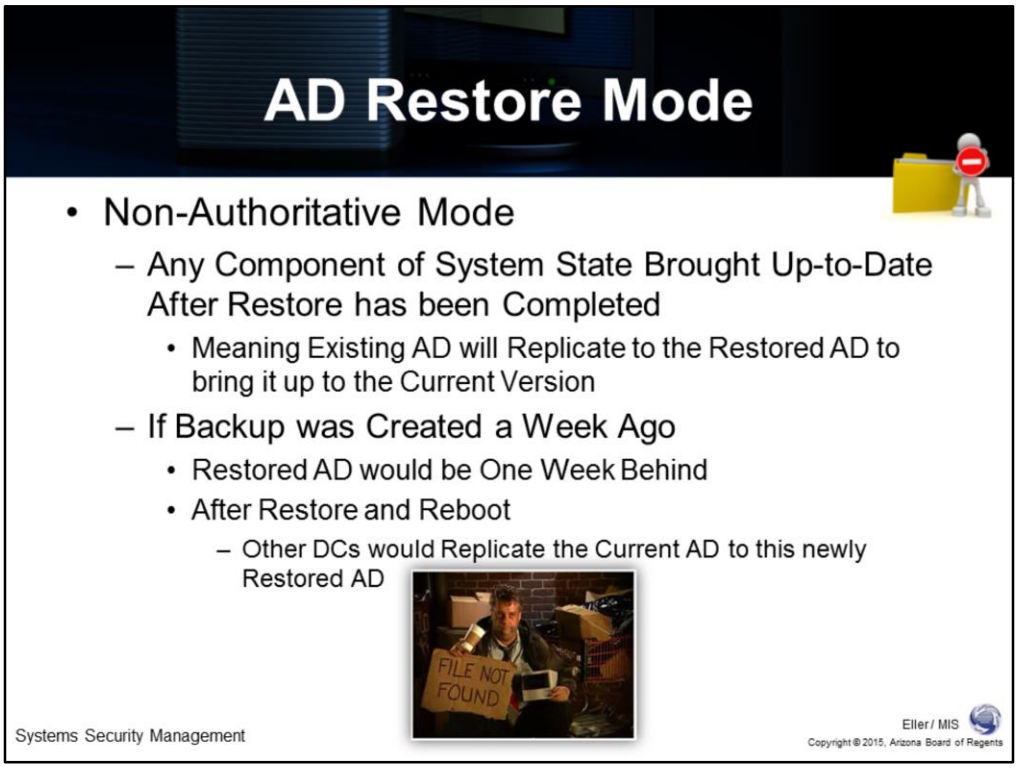
Seizure of a master role can only be accomplished through the command prompt using the NTDSUtil.exe program. Details on the exact commands to seize roles can be readily found via Internet search. Again, this should only be done as a last resort.



Active Directory Restore Mode

In the event an entire Active Directory fails or becomes corrupted, the SysAdmin will need to restore the AD from backups. Restoration of an Active Directory can only be accomplished by rebooting the domain controller into Active Directory Restore Mode. This is similar to Windows safe mode and is started in the same manner. When you reboot the server, after the BIOS post and before Windows starts to load, hit F8 and choose AD Restore Mode.

Once booted into restore mode, there are two restoration methods: authoritative and non-authoritative. Regardless of which method selected, the SysAdmin must restore all System State data from the backup. This will include restoring the registry, COM+ classes, and system boot files, among others.

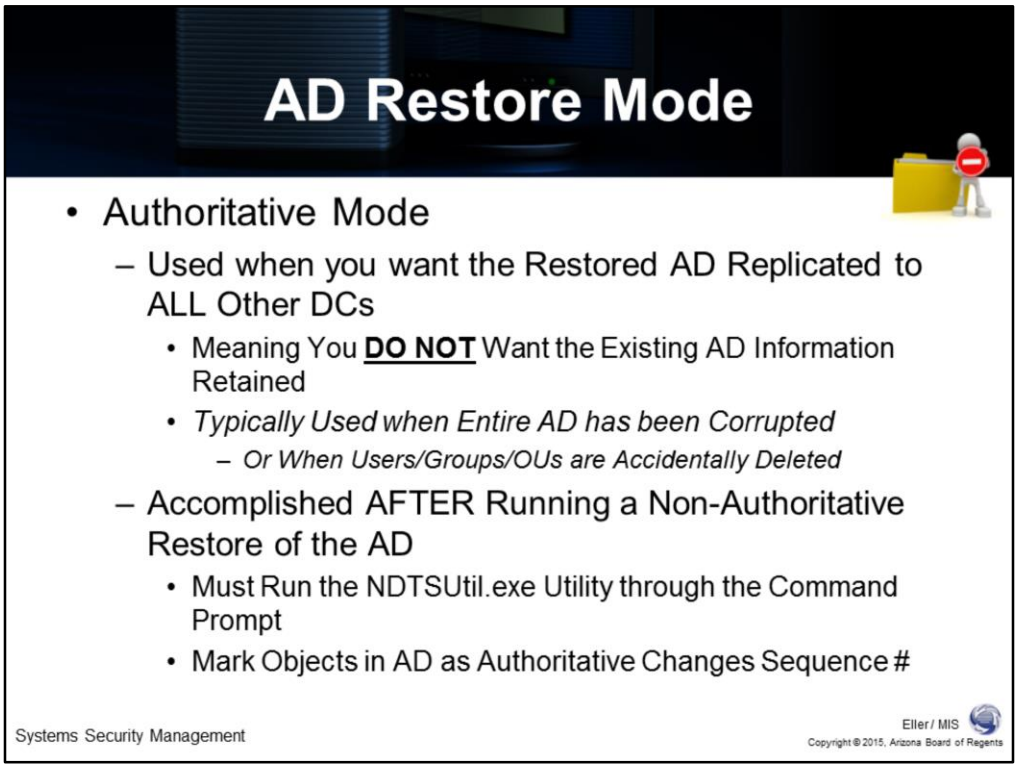


Active Directory Restore Mode (continued)

Non-Authoritative Restore

The non-authoritative restore method will restore the Active Directory to the same state it was in when the backup occurred, then once the system is rebooted and communicates with other AD domain controllers, the domain controller will be updated to the most recent version stored on the other domain controllers.

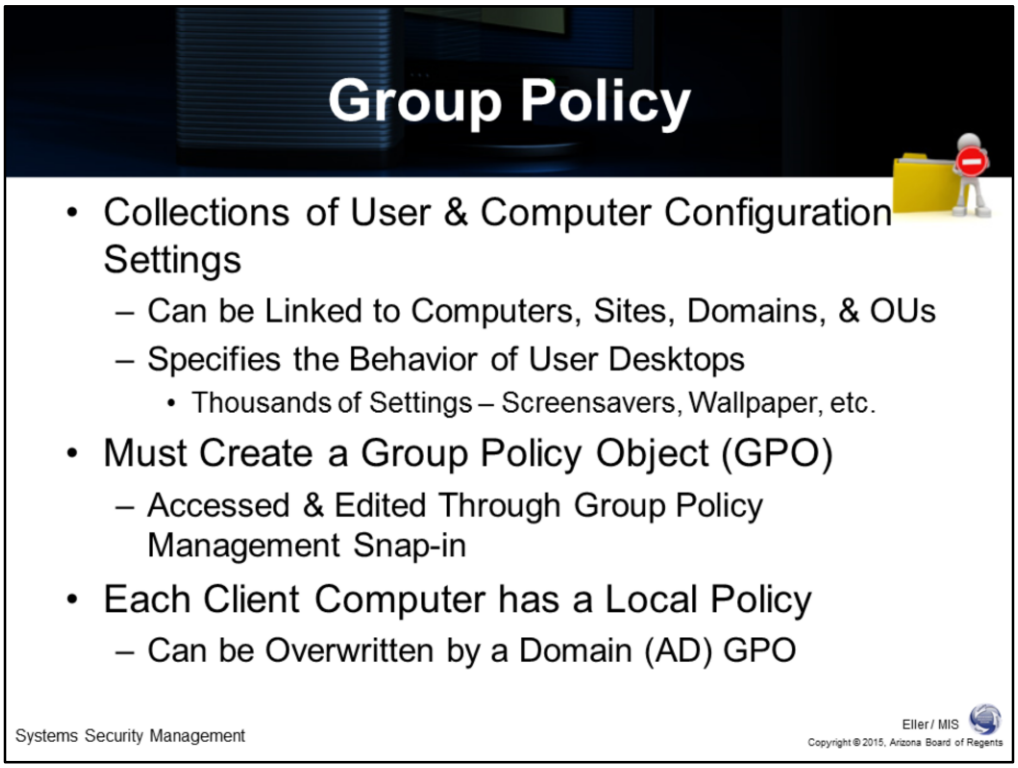
So, if the backup was created one week ago, the restored domain controller would be one week behind. After a restore and reboot, the other domain controllers would update the restored domain controller to the most up-to-date version. This is the default Active Directory restore mode behavior.



Active Directory Restore Mode (continued)

Authoritative Restore

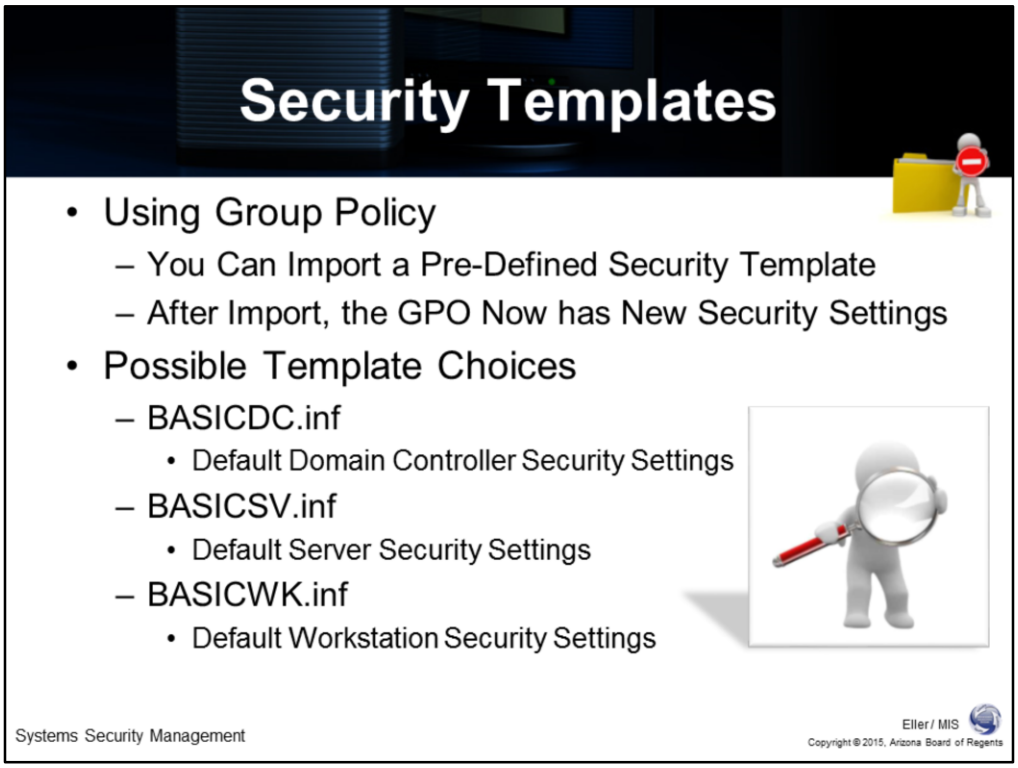
The authoritative restore method is used when you want the restored AD information replicated to all other domain controllers. This means you DO NOT want the existing active directory information retained after a restore. This is typically used when an entire active directory has been corrupted or when users, groups, or OUs have been accidentally deleted. After running a non-authoritative restore of the Active Directory, the SysAdmin must open a command prompt and run the NTDSUtil.exe utility. By using this utility to mark the objects in AD as authoritative will change the Active Directory sequence number. This sequence number is what tells the other domain controllers on the network which domain controller has the most up-to-date information for replication. Once the authoritative restore has completed, rebooting the domain controller will initiate a replication of the restored information to the existing domain controllers.



Group Policy

Group Policy is a collection of user and computer configuration settings which are defined by the SysAdmin. These group policies can be linked to specific computers, sites, domains, and organizational units. In general these settings will specify the specific behavior of user computers on the network. There are thousands of possible settings controlling everything from user password policies to screensavers and wallpapers.

In order to use Group Policy, the SysAdmin must create a Group Policy Object (or GPO). These objects are accessed and edited through the Group Policy Management MMC snap-in. Each client computer on the network has a local security policy which can be overwritten by a domain GPO. In order to assist with setting up GPOs, Microsoft has provided a number of Security and Administrative templates which can be used to define certain settings automatically.



Security Templates

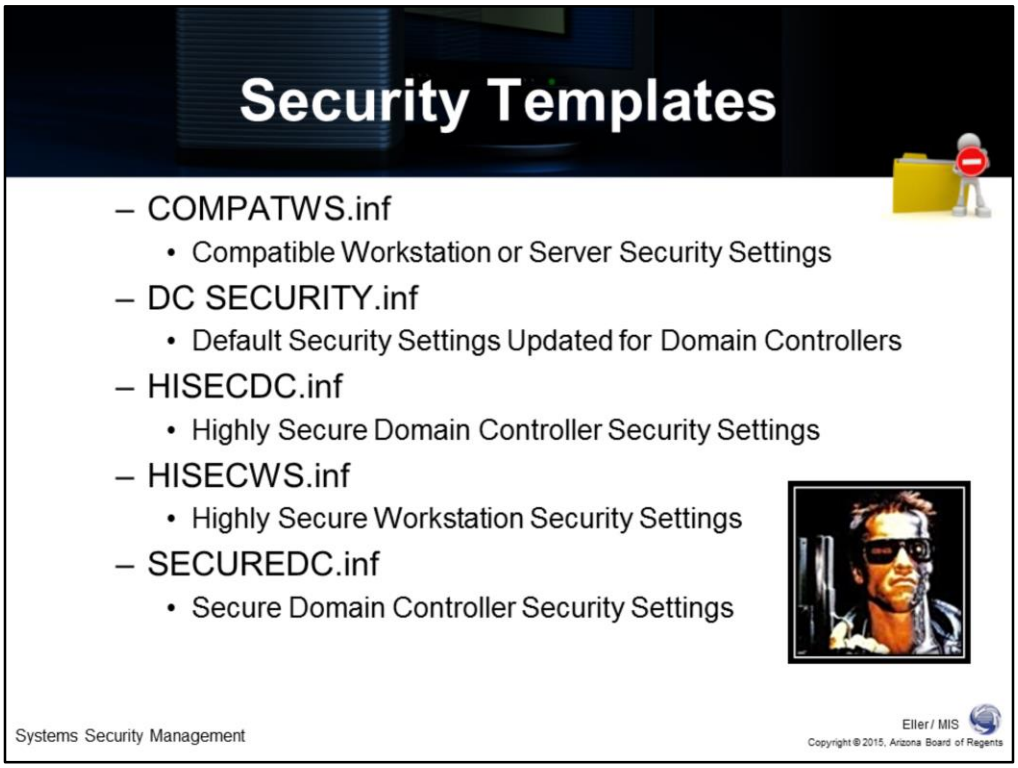
When creating a GPO, a SysAdmin can import a pre-defined security template. After the import, the GPO will have new default security settings. The following is a list of the possible security template choices and what kind of default security they will provide.

Possible Template Choices

BASICDC.inf – This is the default Domain Controller security settings template. In the event a SysAdmin makes a setting change on a domain controller that causes problems, BASICDC.inf can be imported in order to reset back to the default.

BASICSV.inf – This is the default server security settings template used to revert changes to a GPO for a Windows Server back to the default settings.

BASICWK.inf – This is the default workstation security settings template used to revert changes for a Windows client computer back to default settings.



Security Templates (continued)

Possible Template Choices (continued)

COMPATWS.inf – This is compatible workstation or server security settings and is used for backwards compatibility with older applications which cannot run using updated security settings.

DC SECURITY.inf – These are newer default security settings updated for domain controllers and is used for reverting domain controllers back to the original defaults.

HISECDC.inf – This is a highly secure security template designed to increase domain controller security to high security levels.

HISECWS.inf – This is a highly secure security template designed to increase workstation security to high security levels.

SECUREDC.inf – This is a security template designed to increase domain controller security to a level between default and high security settings.

Security Templates (continued)

Possible Template Choices (continued)

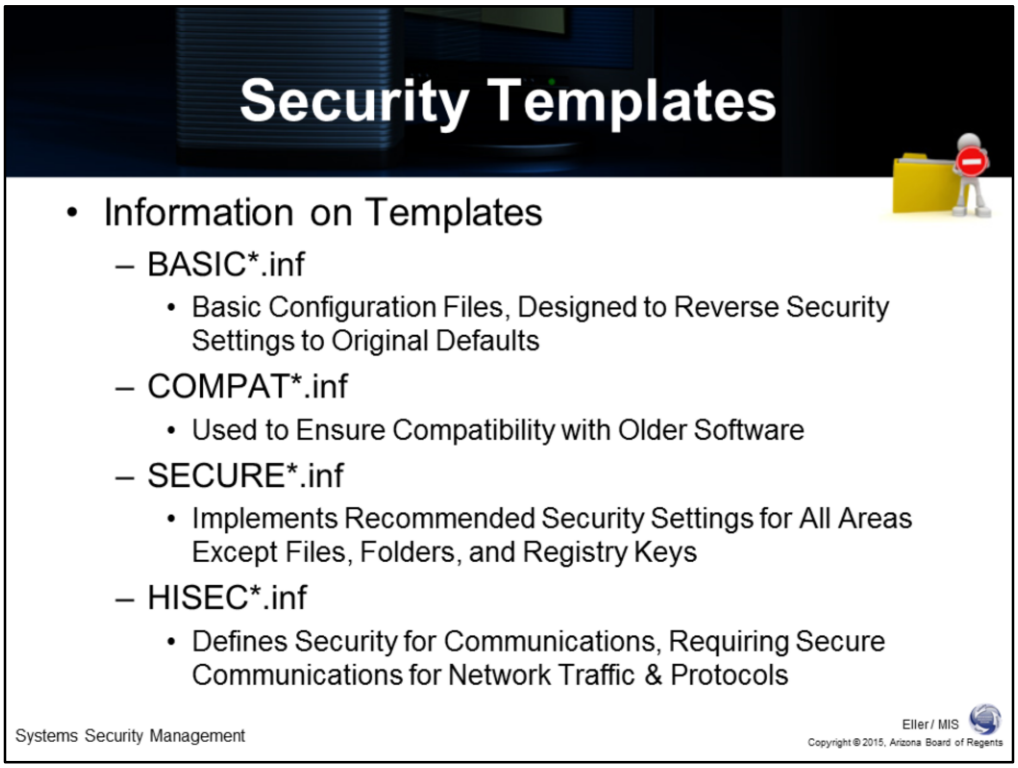
SECUREWS.inf – This template defines secure workstation or server security settings, setting security levels between the default settings and high security settings.

SETUP SECURITY.inf – This is the out-of-the-box default security settings for older versions of Windows Servers and Workstations.

NOTSSID.inf – This template reduces security to allow older software to run properly in terminal services (or remote desktop) modes.

OCFILESS.inf – This template provides optional component file security for servers. These optional settings would include security for optional components such as Internet Information Services.

OCFILESW.inf – This template will provide optional component file security for workstations.



Security Templates (continued)

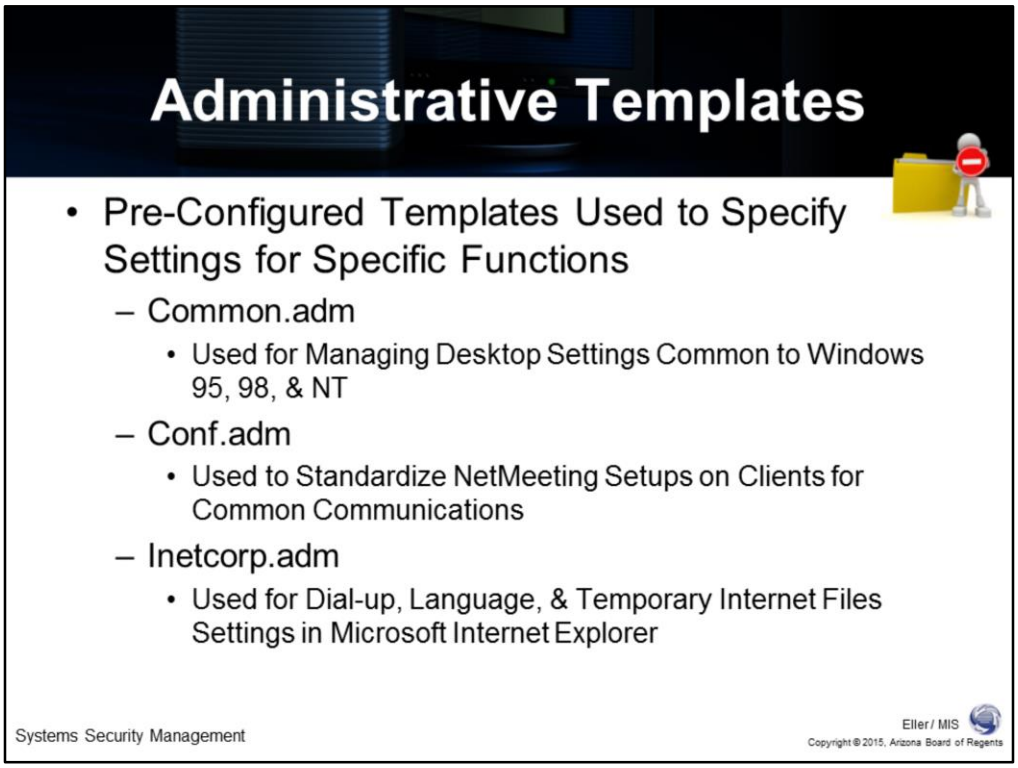
To recap, in general the following naming structure is given to GPO security templates:

BASIC*.inf – these are basic configuration files designed solely to reverse security settings back to the original default levels.

COMPAT*.inf – these are used to ensure compatibility with older software applications.

SECURE*.inf – these templates are used to implement recommended security settings for all areas except files, folders, and registry keys.

HISEC*.inf – these templates define security for communications, requiring secure communications for network traffic and protocols.



Administrative Templates

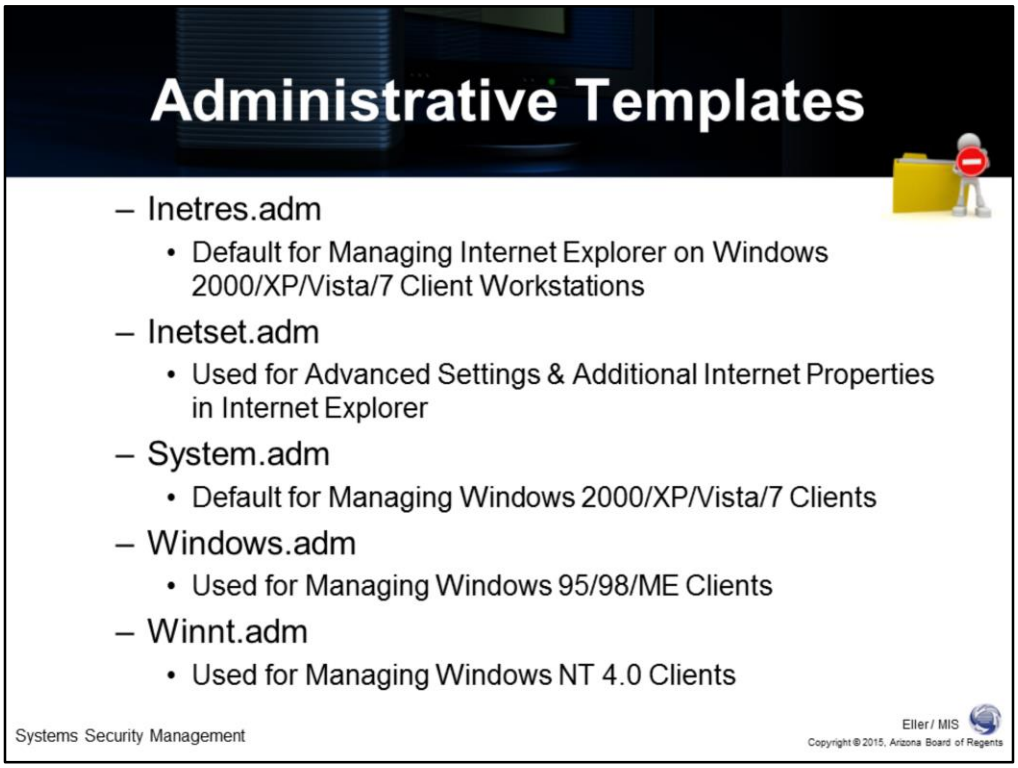
Administrative templates can also be used with GPOs in order to provide specific settings for specific functions. Here is a list of the most common administrative templates and what they are used for:

Possible Template Choices

COMMON.adm – This template is used for managing desktop settings common to Windows 95, 98, and NT.

CONF.adm – This template is used to standardize NetMeeting setups on client computers for common communications over the network.

INETCORP.adm – This template is used for dial-up, language, and Temporary Internet Files settings in Microsoft Internet Explorer.



Administrative Templates (continued)

Possible Template Choices (continued)

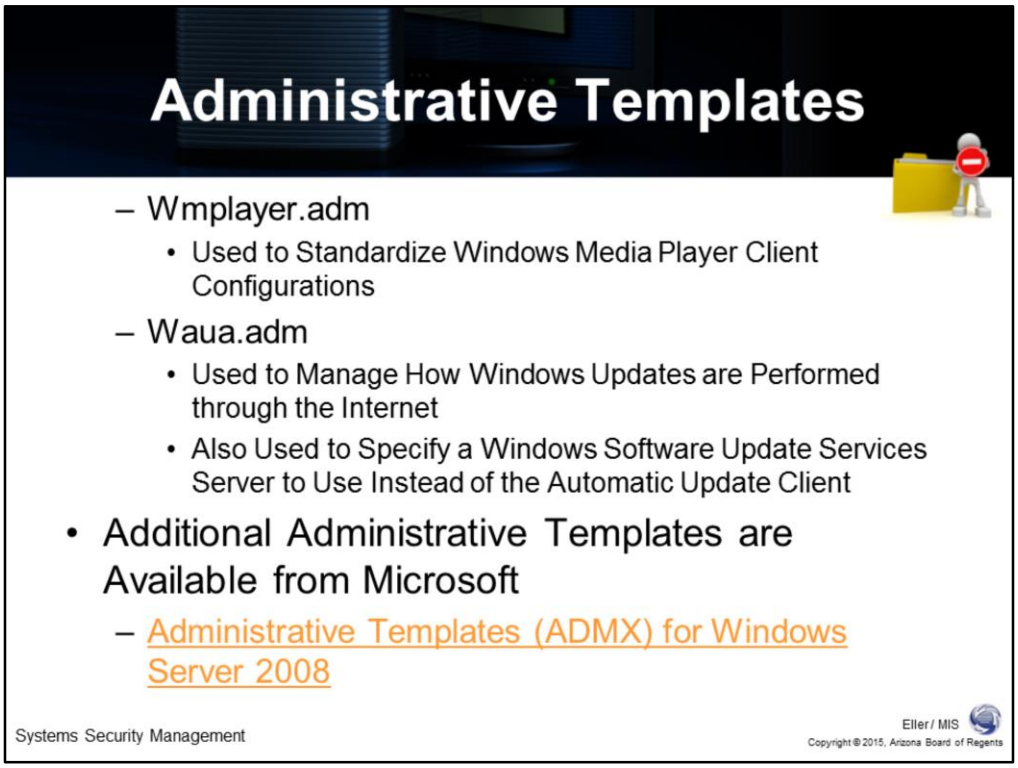
INETRES.adm – This template sets the defaults for managing Microsoft Internet Explorer on Windows 2000, XP, Vista, and Win7 client workstations.

INETSET.adm – This template is used for configuring advanced settings and additional Internet properties in Internet Explorer.

SYSTEM.adm – This is the default template for managing Windows 2000, XP, Vista, and Win7 client workstations.

WINDOWS.adm – This template is used for managing Windows 95, 98, and ME client workstations.

WINNT.adm – This template is used for managing Windows NT 4.0 client workstations.



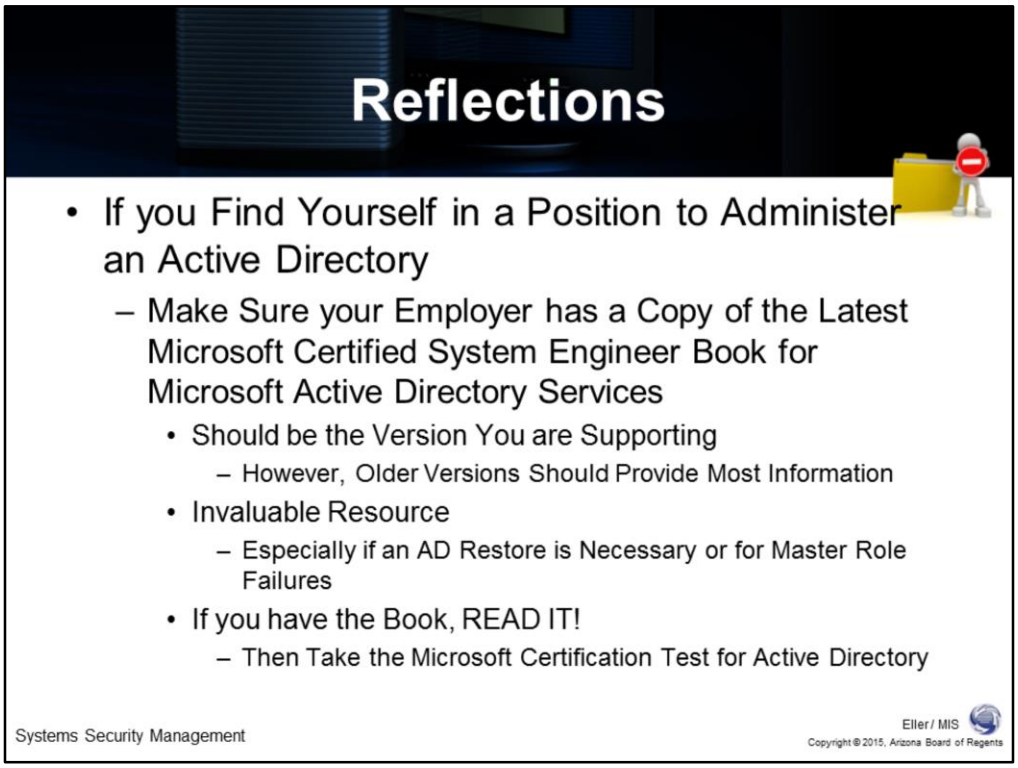
Administrative Templates (continued)

Possible Template Choices (continued)

WMPLAYER.adm – This template is used to standardize Windows Media Player client configurations.

WAUA.adm – This template is used to manage how Windows Updates are performed over the Internet. It is also used to specify a Windows Software Update Server to use instead of using the Automatic Update Client. This is very important if an organization wants to only install specific, approved updates on computers automatically.

In addition to these administrative templates, Microsoft has made available a number of different possible templates for a variety of uses. SysAdmins can download these templates directly from Microsoft's website at no charge.

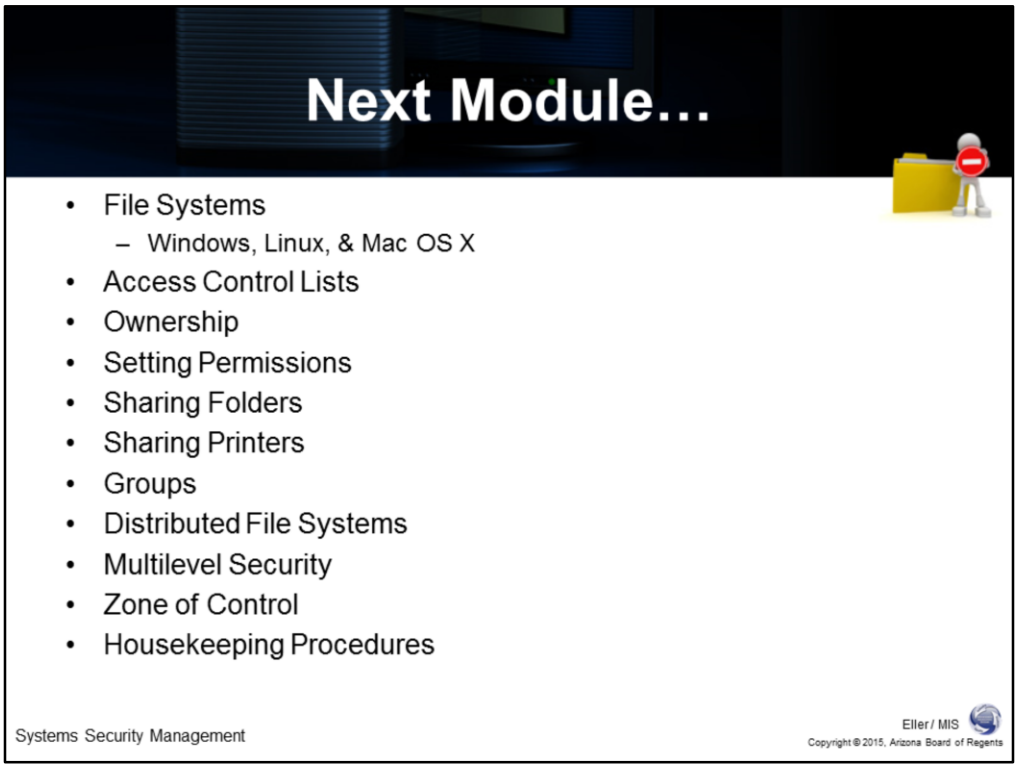


Reflections

These are my own personal reflections based on personal experience with Microsoft's Active Directory platform.

Should you find yourself in a position to administer an Active Directory, make sure your employer has a copy of the latest Microsoft Certified System Engineer book for Microsoft Active Directory Services. This book will include valuable information that you will likely never need to remember, but is great to have in the event you need the information at your fingertips. I would recommend getting the version of the book which matches the version of Active Directory you are supporting; however, you can use other versions (older or newer) as they will contain most of the same information.

This is an invaluable resource that is well worth the money, especially if there is ever a need to perform an AD restore or Master Role seizure. If you have access to this book, read it! Then I would recommend taking the certification test for Active Directory as this will help improve your knowledgebase and your career.



In the next module we will be discussing file, directory, and shared resource security. This will include:

- File Systems
- Access Control Lists
- Ownership
- Setting Permissions
- Sharing Folders and Printers
- Groups
- Distributed File Systems
- Applying Multilevel Security
- Zone of Control
- Housekeeping Procedures

References

Active Directory Schema. (2009, July 7). Microsoft Developer Network. Retrieved from http://msdn.microsoft.com/en-us/library/ms675085(VS.85).aspx.

Windows 2000 Active Directory Services. (2000). Microsoft Press. Penguin Books Canada, Ltd.: Canada.