

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Part II of the *Best Practice Guide for Securing Active Directory Installations and Day-to-Day Operations* contains recommendations for managing domain controllers in a secure manner, detecting attacks, defending against known and unknown threats, and recovering from attacks. This guide provides guidelines for Active Directory administration, monitoring, and recovery that are designed to maintain a secure operating environment.

Introduction

Organizations require a network operating system (NOS) that provides secure access to network data by authorized users and rejects access by unauthorized users. For a Microsoft® Windows® 2000 network operating system, the Active Directory® directory service provides many key components needed for authenticating users and for generating authorization data for controlling access to network resources.

A breach in Active Directory security can result in the loss of network resource access by legitimate clients or in the disclosure of potentially sensitive information. Such information disclosure can occur for data that is stored on network resources or from the Active Directory database itself. To avoid these situations, organizations need more extensive information and support to ensure enhanced security for their NOS environments. This guide addresses this need for organizations that have new, as well as existing, Active Directory deployments.

A comprehensive plan for Active Directory security requires action in five areas:

- Protection of domain controllers against known threats
- Administrative policies and practices to maintain security
- Detection of attacks that have not been identified or mitigated previously
- Defense against Active Directory attacks
- Recovery from Active Directory attacks

The *Best Practice Guide for Securing Active Directory Installations and Day-to-Day Operations* comprises two parts. Part I of the guide contains recommendations for protecting domain controllers from potential attacks of known origin and recommendations for establishing secure administrative policies and procedures. Part II of the guide, which is presented here, contains recommendations for managing domain controllers in a secure manner, detecting attacks, defending against known and unknown threats, and recovering from attacks.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Scope of This Guide

Although NOS security relies on secure operations for all components in the operating system, the scope of this guide is limited to recommendations for operating, monitoring, and restoring Active Directory domain controllers and workstations used for Active Directory administration. Other security topics, such as secure network connectivity and secure clients, are not addressed in this guide.

Part II of this guide provides guidelines for Active Directory administration, monitoring, and recovery that are designed to maintain a secure operating environment. These guidelines, which can be applied to both new and existing Active Directory infrastructures, are organized into the following chapters:

- Maintaining Secure Active Directory Operations
- Monitoring the Active Directory Infrastructure
- Recovering from Active Directory Attacks

The recommendations in this guide take into consideration how an organization's domain controllers are deployed. Domain controllers can be deployed in datacenters for enterprise intranets, in branch offices, and in datacenters for extranets. In some cases, the guidelines vary in accordance with special circumstances that are encountered in each environment.

Audience

This guide is intended primarily for IT planners, architects, and managers who are responsible for establishing Active Directory deployment and operations practices. As a result, this guide emphasizes the decision-making process rather than procedures.

How to Use This Guide

The information in this guide is presented as if the reader's organization is planning its Active Directory deployment and operations. However, this information can be equally beneficial to an organization that is reviewing its current Active Directory security practices.

Proceed through the Active Directory security maintenance process as presented in this guide. Each phase of the Active Directory security maintenance process, such as maintaining domain controller and administrative workstation security, is contained in its own chapter. Each chapter topic begins with a discussion of how these security recommendations enhance security, and also discusses their cost in terms of complexity and performance. If a recommendation is impractical for a specific

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

deployment strategy, then that limitation is indicated. If alternate recommendations exist for a given Active Directory deployment, then this alternative is presented. Finally, the recommendations in each chapter are summarized in a checklist at the end of the chapter.

You can proceed to the next chapter after completing the checklist of recommendations at the end of the previous chapter.

Process for Securing Active Directory Installations and Operations

This guide focuses solely on the deployment and operation recommendations for creating a secure Active Directory system. Figure 1 depicts the process flow for the recommendations in this guide.

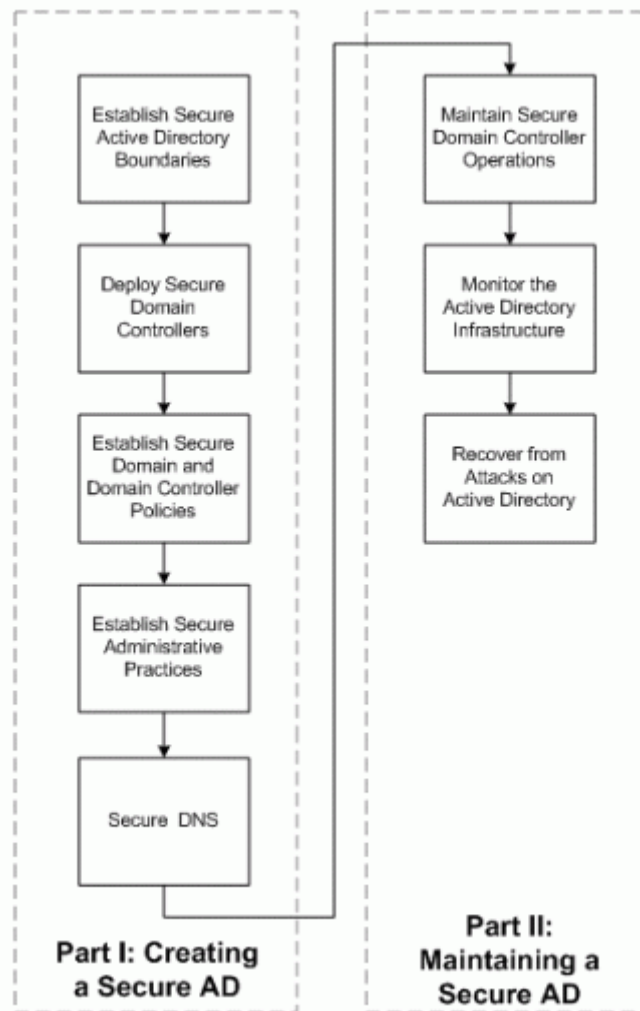


Figure 1: Process Flow for Securing Windows 2000 Active Directory

Parts I and II in the flowchart correspond to Parts I and II of this guide.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Part I of the process is designed to create a secure domain controller environment. To review Part I of this guide, see Best Practice Guide for Securing Active Directory Installations and Day-to-Day Operations: Part I at <http://www.microsoft.com/downloads/release.asp?ReleaseID=44459>.

Part II of the process is designed to help maintain a secure Active Directory infrastructure. This includes a list of day-to-day security operations to perform, specific events and resources to monitor, administrative activities to audit, and how to detect and respond to an attack. Finally, in the case of a security attack damaging some portion of the Active Directory infrastructure, Part II provides recommendations for system recovery.

Chapter 1 - Maintaining Secure Active Directory Operations

Once an organization has deployed their Windows 2000 domain controllers in accordance with the security recommendations laid out in Part I of this guide, it is essential that this level of domain controller security be maintained or even enhanced over time. Whether or not the environment will remain secure is determined in large part by the organization's IT operations practices.

Part I of this guide provides recommendations for deploying Active Directory securely, such as building and configuring domain controllers. Part II provides recommendations for maintaining Active Directory securely with such practices such as periodically auditing domain controller configurations to ensure that unauthorized changes have not occurred.

Note: This chapter contains recommendations for periodic *audits* of domain controller configurations, administrative group memberships, and administrative privileges. The term "audit" here refers generally to processes that regularly track security practices and configurations and not to events that are logged in the security event log. This ensures that security practices are being followed and that the intended security configurations remain in place.

Each organization should develop policies for domain controller administration to provide a basis for secure domain controller operations. A set of written and enforced policies will ensure that domain controller administrators:

- Clearly understand the security policies and security code of conduct of the organization.
- Can trust that the same level of security exists elsewhere in the organization.
- Can respond in a timely manner with the best solution for a problem if a security breach or incident occurs.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Important: Recommendations and procedures in this document assume that you are running Windows 2000 Server Service Pack 3 (SP3) or later for servers and Windows 2000 Professional Service Pack XX or later for Administrative workstations.

Designating Servers as High Security Server

Although security is an important consideration for all components of an organization's network, for some servers, designated as high security servers, security is of particular importance. The high security designation stems from the high security privileges that are associated with processes running on these servers. Designate any server in your organization as a high-security server when it:

- Runs a service in the context of a service administrator-level account.
- Is trusted for delegation.

When a server is "trusted for delegation," the server has the capability, when servicing a client request, of requesting services from another server under the client's security context. Since the requesting client could have arbitrarily high security privileges, the server can therefore assume arbitrarily high security privileges. Therefore, all servers that are "trusted for delegation" within the forest should be designated as high-security servers.

Based on these criteria, in addition to domain controllers there may be other high-security servers in your network which will require special day-to-day operations to remain secure. Protect all high-security servers by following these general guidelines for secure server operations.

- Practice regular, secure domain controller maintenance.
- Stay current on all security patches and hotfixes.
- Manage forest-wide configuration settings for Active Directory.
- Manage security of service administrator accounts.

This chapter provides specific recommendations for maintaining the security of domain controllers, other high-security servers, and administrative workstations.

Maintaining Domain Controller and Administrative Workstation Security

When your organization attains secure domain controller and administrative workstation configurations by implementing the recommendations presented in Part I of this guide, you begin operations. In a production environment, administrators perform day-to-day — and, occasionally —

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

special maintenance on domain controllers and administrative workstations. How these tasks are performed directly affects the level of domain controller and administrative workstation security that your organization can maintain.

Written policies and practices should exist for all domain controller maintenance operations, including:

- Domain controller backup and restore.
- Domain controller and administrative workstation hardware retirement.
- Domain controller and administrative workstation virus scans.

Establishing Domain Controller Backup and Restore Strategies

Administrators schedule regular system state backups on domain controllers to recover from the loss of Active Directory data and the loss of a domain controller. Depending upon its location, the failure of a domain controller can cause a serious disruption in service. As part of secure management and recovery operations, domain controller backups must be performed regularly and securely. System state backups on domain controllers differ from typical server backups and restores in its complexity because:

- Incremental backups are not possible.
- Not all domain controllers should be backed up.
- Backups from one domain controller cannot be used to restore a different domain controller.
- Restores are either authoritative or non-authoritative.
- Domain controllers are high-security servers, requiring special handling.

Due to the high-level security requirements, a secure backup and restore policy includes security practices that are not required for a typical server backup. A secure domain controller backup and restore strategy should include the following key practices:

- Avoid the use of a common, enterprise-wide account for backup.
- Limit domain controller backup hardware to locations with hardware and media security.
- Schedule regular domain controller backups, and destroy out-of-date backup media.
- Protect Backup Operators accounts.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

- Practice restoring domain controllers from backup media periodically.

Implement an enterprise-wide, published backup and restore policy that specifies which domain controllers should be backed up, who has permission to perform this function, how frequently domain controllers will be backed up, and how the backup media should be handled.

Dedicating a Backup Agent Service Account for Domain Controllers

The service account that is used to back up domain controllers must be a service administrator, and therefore highly privileged. To maintain a high level of security, backup agent service accounts used for backing up domain controllers should be different from the service accounts that are used for other server backups.

When a domain controller is promoted, a special built-in group, Backup Operators, is created in Active Directory. This group possesses the privileges necessary to backup and restore files on all domain controllers in the domain and hence its members are service administrators. As a general recommendation, membership in groups with service administrator privileges should be highly restricted. Users that are responsible for backing up data on application servers only (and not domain controllers) should therefore not be made a member of the Backup Operators group in Active Directory.

To back up a domain controller, the backup agent service running on the domain controller must run in the security context of an account with Backup Operator privileges (service administrator-level privileges). If the same backup agent service account is used for backups on both domain controllers as well as other application servers, then the application servers could potentially be compromised to gain access to this highly-privileged account.

An intruder, who gains access to such an application server with a backup agent service and compromises the backup agent service account, can gain access to administrative credentials. Therefore, a backup agent service account with service administrator credentials should only be used to perform backups for domain controllers. To maintain this separation, require that different backup agent service accounts for application servers and for domain controllers.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

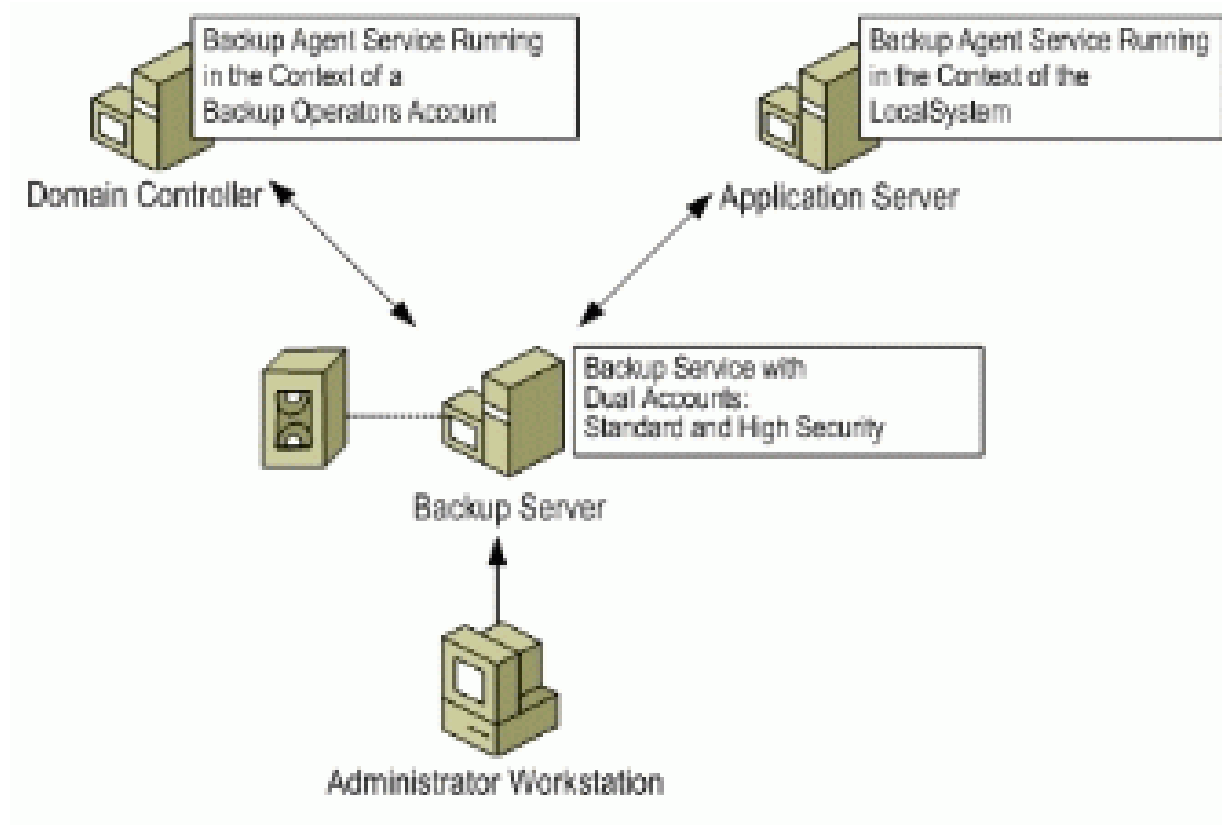


Figure 2: Distinction Between Backup Agent Service Accounts on Domain Controllers and Application Servers

Limiting Backup Services and Media Storage to Secure Locations

Provide domain controller backup media with the same level of physical security as the domain controllers themselves. Because the backup media contains all the information in the Active Directory database, theft of the backup presents the same risks as theft of a domain controller or a disk drive from a domain controller. An attacker could restore the information elsewhere and access Active Directory data.

To prevent individuals from gaining unauthorized access to backup media:

- Remove media from the backup hardware drive as soon as the backup process completes.
- Store backup media that you use on site in a secure location where access is audited.
- Store archival backup media securely off site.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

- Establish processes and procedures that require the signatures of authorized administrators when any archival backup media is brought back on site.

Choosing a Backup Strategy for Branch Offices

Implementing a domain controller backup strategy is straightforward for organizations with domain controllers located in a few secure locations. In such organizations, you can administer domain controller backups in a secure manner relatively easily.

In contrast, domain controller backups tend to be performed infrequently, if at all, in locations with the following characteristics:

- Limited facilities by way of its IT infrastructure or administrative staff.
- Limited ability to securely store the media on or off site.

A location with limited IT facilities might be a smaller, regional datacenter or branch office. For the purposes of this guide, locations with limited facilities are hereafter referred to as branch offices.

Infrequent backups at branch offices occur for several reasons:

- Purchasing and maintaining a backup system can be costly.
- Training and maintaining on-site administrators increases overhead costs.
- Limiting the locations where domain controller backup media must be protected enhances security.
- Because most branch offices are resource constrained, and Active Directory backups are very disk-intensive operations, backups may need to wait until weekends or other less busy periods.

Table 1 lists three alternatives for secure backup and restore practices at branch offices.

Table 1 Possible Backup and Restore Practices for Regional Datacenter and Branch Offices

Option	Advantages	Disadvantages
No domain controller backups at branch offices	Easy to secure Least administrative overhead	Higher risk of data loss Long delays possible when restoring a domain controller in the branch office

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Backups at all branch offices using remote backup systems (off-line media) in secure data centers	Relatively easy to secure Reduced risk of delays in restoring domain controller in branch office	More administrative overhead
Backups at all branch offices using local backup to disks (on-line media)	Low risk of data loss Least downtime when a domain controller fails in branch office.	Most administrative overhead Difficult to secure.

Some organizations may choose to eliminate regional datacenter and branch office domain controller backups altogether. Foregoing this process eliminates the cost and complexity of backups, as listed in Table 1. The disadvantages are that the failure of a domain controller in these locations exposes their users to the possibility of substantial downtime while a new domain controller is built and promoted.

Tape-backup infrastructures duplicating those in enterprise datacenters can be deployed at regional datacenters if the organization has a complex site topology, a large number of sites to back up, and insufficient connectivity between branch office sites and the enterprise datacenter.

Developing a Secure Remote Backup Process

If your organization chooses to backup branch office domain controllers to a centralized backup infrastructure, two secure strategies are possible. These are:

- Directly backing up domain controllers to a central location, or
- Temporarily backing up domain controllers to a secure, local disk share that can be downloaded later by a backup system in a central location.

If you plan to download data from domain controllers in branch offices or small datacenters directly to the backup devices in a secure location, determine if there is adequate network bandwidth and off-peak time to perform all the backups that you have planned. Backing up a domain controller is a disk-intensive operation that should be performed during off-peak hours or over the weekend. Backing up a large number of domain controllers to a single backup infrastructure can require more time than can be easily scheduled within off-peak times.

An alternative strategy is to write domain controller data from branch offices or small datacenters to local, secure disk storage first. This step avoids the need to deploy costly tape-backup infrastructure to field locations. It also increases security, because without distributed backup tapes, there is less

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

likelihood of the compromise of a single tape. Subsequently, the local backup to disks in branch offices can be downloaded to off-line media backup systems in enterprise data centers.

All field domain controllers should have a file share with the same designation. This share will contain only the most recently backed up copy of the system state of the domain controller. Should the domain controller operating system fail, this copy can be used to quickly restore the machine. To make this method secure, configure backup share permissions so that only domain administrators can access this shared drive.

Ensuring That the Required Backup Media Is Available When Needed

For each backup, verify that the backup runs to completion. Your organization's backup software determines how backup success is verified. For example, the NTBACKUP utility logs event ID 8019 upon the successful completion of a backup. Your organization's monitoring software, such as Microsoft® Operations Manager (MOM) 2000, can be configured to monitor for backup success or failure.

Note: Verify that your organization backs up your domain controllers with a frequency that is less than the Active Directory tombstone lifetime. To ensure this, you can recycle backup media after it exceeds 75 percent of the tombstone lifetime.

A check should be performed *weekly* in each domain to ensure that:

- At least two domain controllers have been successfully backed up that week.
- If backups are not successfully performed, then the problem should be escalated and resolved as a high priority.
- Backup media that is created has been clearly labeled with the name of the domain controller and the date the backup is created and then stored securely.

Include the name and roles of the domain controller in the label of the backup media to facilitate easy identification later. One suggested format for labeling the backup media that captures important information about the domain controller is:

FQCN.Build.OMRole.[GC].[MD5].TSL.YYMMDD.BKF, where:

- FQCN is the fully qualified computer name.
- Build is the build number of the operating system.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

- OMRole indicates the operations master role(s) held.
- GC indicates that this is a global catalog.
- MD5 indicates that a *message digest5* digital signature was added (optional).
- TSL is the terminal service license (Optional).

Managing Backup Operators Accounts

Active Directory contains a built-in group named Backup Operators. Members of this group are considered service administrators, because the group's members have the privilege to restore files, including the system files, on domain controllers. Membership in the Backup Operators group in Active Directory should be limited to those individuals who back up and restore domain controllers.

All member servers also contain a built-in group called Backup Operators that is local to each server. Individuals who are responsible for backing up applications on a member server should be made members of the local Backup Operators group on that server — as opposed to the Backup Operators group in Active Directory.

On a dedicated domain controller, you can reduce the number of members in the Backup Operators group. When a domain controller is used for running other applications, as it might be in a branch office, individuals who are responsible for backing up applications on the domain controller must also be trusted as service administrators, because they will have the privileges necessary to restore files, including the system files, on domain controllers.

By default, the Backup Operators group is empty. Its membership can be modified by members of the Administrators, Domain Administrators, and Enterprise Administrators groups. The Backup Operators group is not protected by the special default security descriptor settings on the *AdminSDHolder* object that are applied to other service administrator accounts. To protect the Backup Operators group in Active Directory, apply the same permissions that protect other service administrator accounts. These permissions are listed in Table 2,

Table 2 Security Descriptor to Protect the Backup Operators Group in Active Directory

Type	Name	Permission	Apply To
Allow	Administrators	List Contents Read All Properties Write All Properties	This Object Only

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

		Delete Read Permissions Modify Permissions Modify Owner All Validated Writes All Extended Rights Create All Child Objects Delete All Child Objects	
Allow	Authenticated Users	List Contents Read All Properties Read Permissions	This Object Only
Allow	Domain Admins	List Contents Read All Properties Write All Properties Read Permissions Modify Permissions Modify Owner All Validated Writes All Extended Rights Create All Child Objects Delete All Child Objects	This Object Only
Allow	Enterprise Admins	List Contents Read All Properties Write All Properties Read Permissions Modify Permissions Modify Owner All Validated Writes All Extended Rights Create All Child Objects Delete All Child Objects	This Object Only
Allow	Everyone	Change Password	This Object Only
Allow	Pre-Windows 2000 Compatible Access	List Contents Read All Properties Read Permissions	Special
Allow	SYSTEM	Full Control	This Object Only

Practicing Active Directory Recovery Procedures

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Backup media can be used to restore Active Directory data on a functioning domain controller (data restore) or to recover one or more nonfunctioning domain controllers (recovery).

Using backup media to recover an entire domain controller that has failed or been corrupted is a seldom-used practice. The most common method for recovering a single, failed domain controller is to promote a member server to a domain controller and then replicate the Active Directory data from another domain controller that is available online. However, if there are no available domain controllers that are free of corruption, you may need to resort to domain controller recovery from backup media. Because recovery procedures are performed infrequently, unknown problems may exist in your domain controller recovery practices, including the following:

- Failed backup media
- Incomplete or erroneous recovery procedures
- Lack of familiarity with procedures on the part of individuals who are responsible for domain controller recovery

To help avoid these problems, consider implementing the following recommendations:

- Verify the quality of your backup media by periodically performing a data restore on a domain controller.
- Ensure that your administrators are familiar with forest recovery procedures — before they are needed — by periodically performing a forest recovery.

Verifying That Backup Media Is in Good Condition

To help ensure that a restore from backup media will succeed, validate the quality of the backup media on a regular basis. Your organization should establish a practice of verifying the backup media quality with a frequency that is less than the Active Directory tombstone lifetime. Because backup media should be discarded before one tombstone lifetime has elapsed, this practice ensures that at least one useful backup exists at all times.

An Active Directory data restore includes the following steps:

- Use a test domain controller in a lab setting, isolated from the production forest.
- Follow the procedure for data restore from selected backup media.
- Verify that the data has been properly restored.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

- Tag the verified backup media with the date of verification.

To review the procedure for restoring the Active Directory data on a domain controller, see “Performing an Authoritative Restore of Directory Objects” in Chapter 3 of this guide.

Practicing Forest Recovery Procedures

To ensure that your organization can successfully recover one or all domain controllers from backup media, implement the following practices:

- Maintain a record of Active Directory data owners.
- Prepare an enterprise business recovery plan for the forest.
- Prepare a set of procedures for forest recovery.
- Practice at least annually restoring from backup media to ensure the quality of the media.

Practice an Active Directory forest recovery by performing the following tasks:

1. Promote an extra domain controller in each domain in the forest, as shown in Figure 3.
2. Isolate the new domain controllers from the production forest to create a scaled-down replica of your forest.
3. Add client computers and member servers to the isolated forest to represent the actual forest.
4. Practice your prepared forest recovery procedure in the scaled-down forest.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

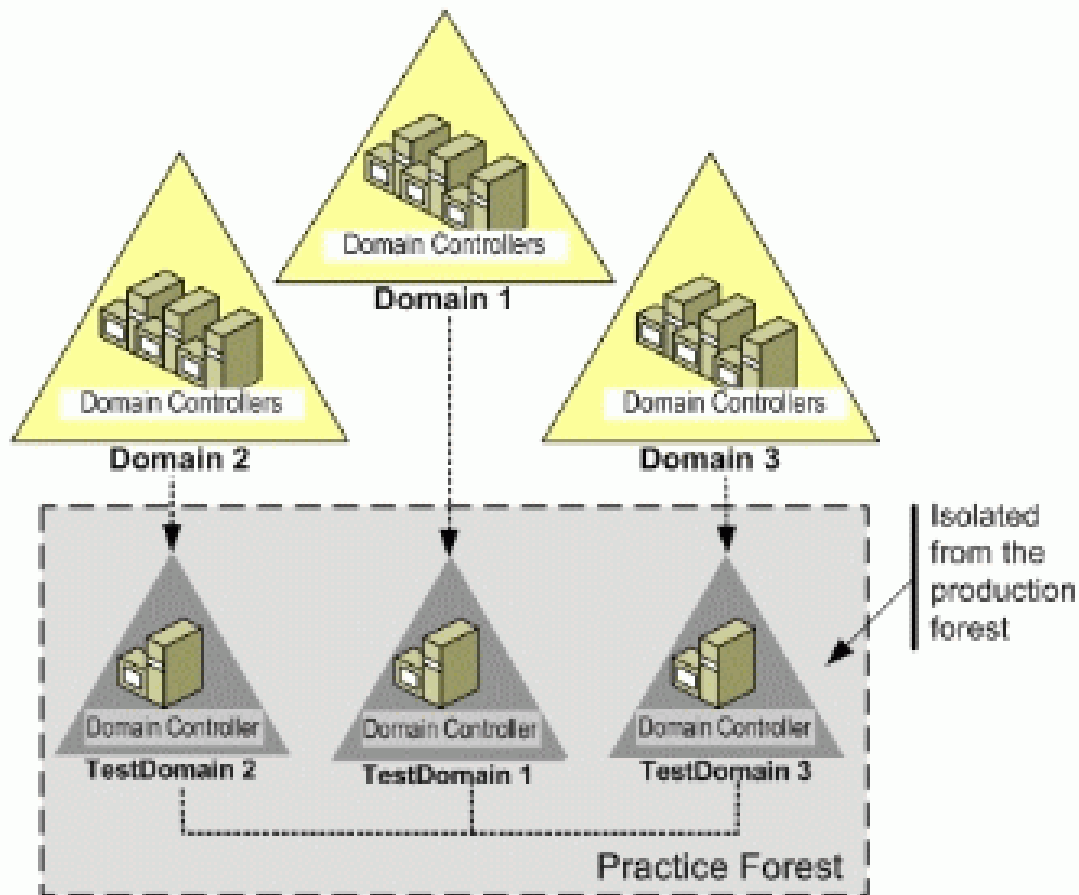


Figure 3: Isolating a Server from Each Domain to Practice Forest Recovery

To review the procedure for restoring the Active Directory forest, see “Best Practices: Active Directory Forest Recovery” at http://www.microsoft.com/windows2000/technologies/directory/AD/redir-bp_forests.asp.

Managing the Life Cycle of Domain Controller Hardware

An organization may regularly dispose of or recycle a significant number of servers, workstations and backup media. Domain controllers, administrative workstations and domain controller backup media contain sensitive information that should be secured. To protect against the recovery of sensitive information in recycled devices, you should have a written policy that specifies how domain controllers, administrative workstations, and their associated backup media are to be handled during the recycling process.

This policy should specify, if possible, the recommendations in Table 3.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Table 3 Recommendations for Disposing of Domain Controllers, Administrative Workstations, and Backup Media

Hardware Type	Recommendation
Hardware devices (computers, hard drives)	Must be managed to ensure that all data has been destroyed and is not recoverable before leaving the organization
Media (tapes, hard drives, SANs, optical disks, DVD-RAM)	Must be erased or degaussed with an approved utility before being reused
Other Media (CDs, microfiche)	Must be physically destroyed or degaussed

Running Antivirus Software on Domain Controllers and Administrative Workstations

On domain controllers, administrative workstations, and other high security server continue to:

- Run virus scans.
- Obtain regular virus signature updates from your antivirus software vendor.

Before initiating regular antivirus scanning, be aware that some antivirus software can interfere with the proper operation of domain controllers by:

- Interfering with directory database and log file access by the Extensible Storage Engine (ESE).
- Interfering with File Replication service (FRS) database and log file access by ESE
- Causing excessive replication by FRS.

The issues with domain controllers running antivirus software are addressed by the recommendations provided in Part I of this guide see “Running Antivirus Software on Domain Controllers and Administrative Workstations.,” in Best Practice Guide for Securing Active Directory Installations and Day-to-Day Operations: Part I at <http://www.microsoft.com/downloads/release.asp?ReleaseID=44459>.

Part 1 of this guide recommends that the SYSVOL folder be excluded from virus scanning. However, excluding SYSVOL increases the risk of a virus attack on a domain controller because viruses tend to attach to files that are executed such as binaries or scripts. The SYSVOL folder contains important executables such as logon and startup scripts.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

As a counter-measure, implement script signing to protect the integrity of scripts running on domain controllers and administrative workstations. For information on implementing script signing see “Running Antivirus Software on Domain Controllers and Administrative Workstations.” in Best Practice Guide for Securing Active Directory Installations and Day-to-Day Operations: Part I at <http://www.microsoft.com/downloads/release.asp?ReleaseID=44459>.

Note: As a best practice, enforce script signing at least on domain controllers and administrative workstations. As a general recommendation, enforce script signing all computers on the network. Operating systems that support script signing are Windows 2000 family of operating systems, Windows XP, and the Windows Server 2003 family.

Staying Current with Security Hotfixes and Service Packs

Throughout a domain controller’s life cycle, attackers may attempt to find and exploit security weaknesses in the operating system. The same is also true of other high-security servers designated in your network. Security bulletins, service packs, and hotfixes are periodically released to counter these threats. It is critical to remain up to date on these fixes on all high-security servers.

Microsoft security bulletins include a rating system to indicate the severity of the problem addressed by the security updates. Table 4 lists the ratings for security updates and provides a description of each rating.

Table 4 Severity Ratings for Security Bulletins and Associated Hotfixes

Rating	Definition
Critical	A vulnerability whose exploitation can allow the propagation of an Internet worm without user action
Important	A vulnerability whose exploitation can result in compromise of the confidentiality, integrity, or availability of users’ data or of the integrity or availability of processing resources
Moderate	A vulnerability whose exploitation is mitigated to a significant degree by factors such as default configuration, auditing, or difficulty of exploitation
Low	A vulnerability whose exploitation is extremely difficult or whose impact is minimal

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Before you can manage the updates that will be required, you need to control what you currently have in your production environment. Key information that is required to maintain a managed network includes:

- Accurate and up to date inventories of applications and operating systems
- Baselines for software applications and hardware configurations

Periodic reviews of inventories and software baselines should be planned as changes are introduced to the production environment.

Selecting a Security Update Strategy

In a small organization with Internet access from each server or workstation, a local administrator may handle updates directly. In this case, Windows Update Service downloads updates directly to each computer and notifies a local administrator on that computer that an update is available. When an administrator does not centrally manage server updates or if the administrator cannot ensure and enforce operating system versions, the network is considered to be unmanaged.

For an unmanaged environment, a local administrator is responsible for keeping the local computer up to date. Figure 4 illustrates this simple arrangement. In some cases, a large organization might also use this strategy to update member workstations but manage servers centrally.

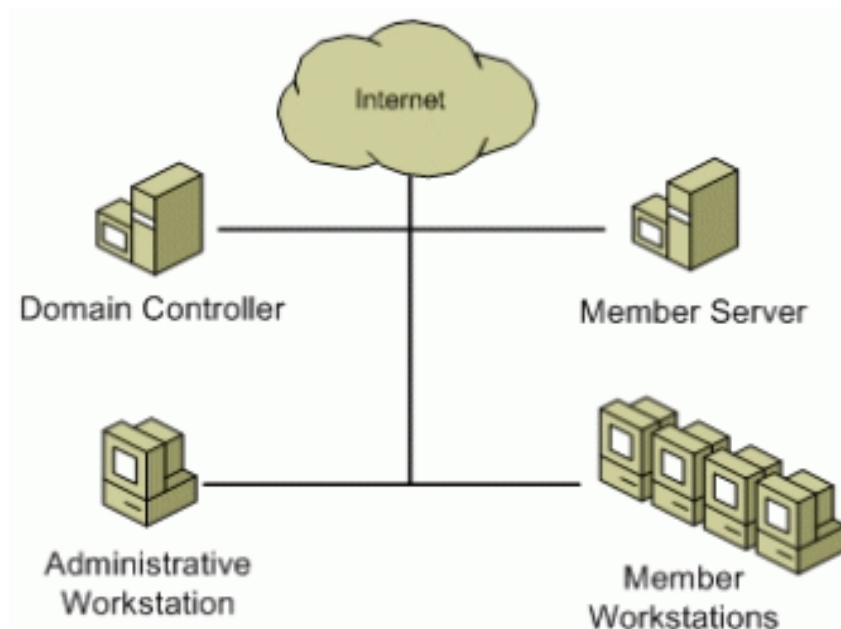


Figure 4: Strategy for Distributing and Installing Security Updates in a Simple Organization

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

In a large organization, an automatic method is required to ensure that all of the high-security servers and administrative workstations are current with regard to security updates. Figure 4 illustrates one recommended design for managing updates on domain controllers and administrative workstations.

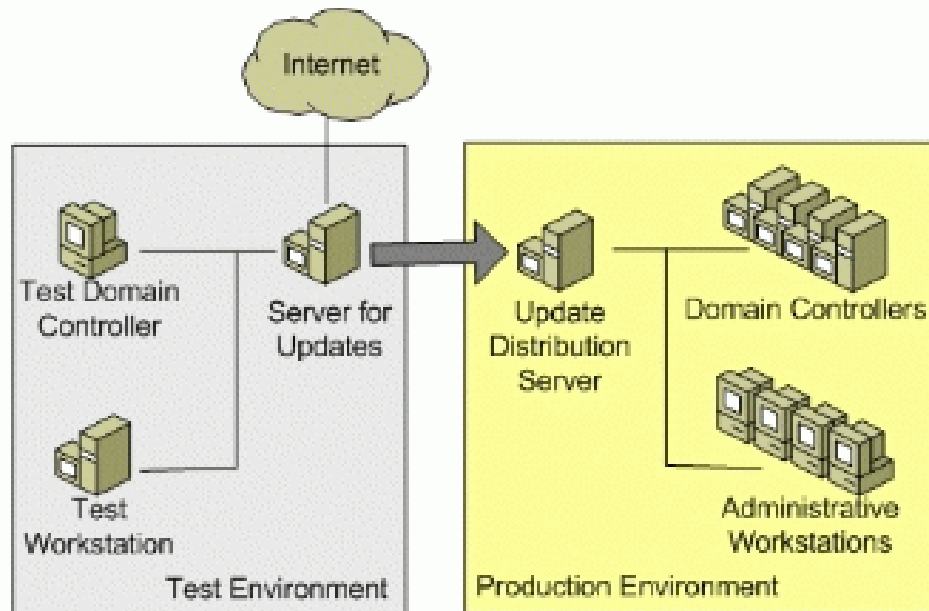


Figure 5: Strategy for Testing, Distributing, and Installing Security Updates in a Managed Environment

With this update strategy, security updates are downloaded automatically to a designated computer that is connected to the Internet and isolated from the production environment. Each update is checked for compatibility with the organization's critical applications in a test lab that is designed to mirror the normal production environment. If the applications function normally in the test lab, the update is copied to the update distribution server. A domain administrator configures the distribution server to push the update out to domain controllers and administrative workstations.

To optimize the update process for an organization, a number of variations on these two strategies are possible. The following section discusses the advantages and disadvantages of each method for automating updates.

Selecting Notification, Deployment, and Auditing Methods

Establish practices for handling update notification, installing, and auditing security updates to domain controllers, other high-security servers and administrative workstations. Consider automating the distribution and installation of security updates in your organization by one of a variety of methods, including the following:

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

- Microsoft Security Notification Service Newsletter
- Windows Update Service
- Software Update Services
- Management software, such as System Management Server

The following topics describe the features of each of these methods in more detail.

Microsoft Security Notification Service Newsletter

The Microsoft Security Notification Service newsletter is a free, subscription-based service that alerts administrators to new security updates. Register for this free service at:

<http://www.microsoft.com/info/PICWhyRegister.htm>.

An administrator must be responsible for reviewing any security updates and determining if the updates will be installed on domain controllers, other high-security servers and administrative workstations. When using this notification method, select a separate method for distributing and installing the updates.

Windows Update Service

With Windows Update Service, any computer that is connected to the Internet can automatically detect and download new operating system updates. You can configure Windows Update Service to either notify you of a pending update or automatically install the update. Any computer running Windows 2000 SP3 or later already has Windows Update Service installed.

Windows Update Service can be configured to automatically install a critical update or to notify an IT administrator that an update is available. You can configure Windows Update to install updates automatically, with or without confirmation, based on the security rating of the update, as listed in Table 4.

The possible limitations of using Windows Update Service as a means of applying service packs and hotfixes to domain controllers and high-security servers include the following:

- For security reasons, an organization's domain controllers and other high-security servers may not have Internet access.
- Automatically installing security updates bypasses the recommendation that all software be run in a test environment before installation in the production environment.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

- Selecting notification instead of automatic updates does not ensure that the service pack or hotfix is actually installed on all domain controllers and other high-security servers.

Software Update Services

Microsoft® Software Update Services (SUS) provides a solution to the problem of managing and distributing critical Windows patches that resolve known security vulnerabilities. SUS takes advantage of the technology used by Windows Update Service to enable enterprise-wide, centralized software update management. This software updates Windows® 2000 Service Pack 2 or later), Windows XP and Windows Server 2003. It requires that Internet Information Services (IIS) be enabled on the server running SUS.

One advantage of SUS is that the domain controllers and high-security servers using SUS updates do not require Internet access to receive the updates. SUS can be configured to automatically install critical updates or to notify an IT administrator that update are available. Table 4 lists the security ratings used by Windows Update Service and SUS.

The two possible strategies for managing updates on enterprise servers are referred to as managed servers and managed datacenter servers, respectively.

Managed Servers

When a new update has been successfully downloaded by the server managing automatic updates, SUS logs an event that indicates that an update is ready to install. When the Automatic Updates policy is enabled, SUS performs a scheduled installation, at a specified day and time, on all high-security servers. Automatic Updates then adds an icon to the notification area and displays a balloon to the first local administrator who logs on to a server stating that new updates are ready to install. The local administrator can choose not to install at this time. However, the Automatic Updates icon remains in the notification area so that the local administrator can install the updates any time before the scheduled installation time. If the installation has not occurred by the next scheduled installation time, Automatic Updates begins a countdown. If no local administrator stops the countdown, the installation proceeds automatically.

Managed Datacenter Servers

When a new update has been successfully downloaded by the server that manages automatic updates, SUS logs an event that indicates that an update is ready to install. In this case, the local administrator must manually install the update. An IT administrator remotely checks the system events on the server and sees an event stating that an update is ready to install. The IT administrator

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

determines when the next available system maintenance window occurs and executes a change notification plan. On the day and time of the scheduled maintenance, a local administrator logs on to the domain controller and manually installs the critical update.

Automatic Updates

Automatic Updates is the client component for SUS as well as the Windows Update Service. IT administrators can configure Automatic Updates on domain controllers and high-security servers with either group policy or registry settings. To make these configuration changes, see How to Configure Automatic Updates by Using group policy or Registry Settings at <http://support.microsoft.com/default.aspx?scid=kb;EN-US;328010&sd=tech>.

Note: When you configure Automatic Updates by using the policy registry keys, the policy overrides the preferences that are set by the local administrator. If an administrator removes the registry keys at a later date, the preferences that were set originally are used once again.

Systems Management Server

Most management software packages, such as Microsoft® Systems Management Server (SMS), can also be configured to automatically apply service packs and hotfixes.

SUS focuses on obtaining critical updates for Windows 2000 and Windows XP inside your corporate firewall as quickly as possible. It is not intended to serve as a replacement for an enterprise software distribution solution, such as SMS.

SMS can provide complete software management, including the distribution of security updates that solve operating system security and virus issues. To enhance SMS solutions for enterprise server management, security-patch improvements exist for Systems Management Server 2.0.

Table 5 provides a process for evaluating your options for managing security updates and determining which option to implement, based on your current IT practices.

Table 5 Strategies for Implementing a Security Update Solution

If You are currently...	Security Update Management Strategy
Using a solution that is successfully distributing security updates	Continue to use your current solution.
Using SMS and need a patch-	Implement the SMS security patch extensions

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

management solution	contained in Systems Management Server 2.0.
Not using SMS and need a patch-management solution	Consider SMS, SUS, or third-party solutions to determine which one best meets your needs.

Deploying Security Hotfixes and Service Packs

Establish practices that address obtaining, testing, and installing hotfixes and service packs by performing the following tasks:

1. Obtain a notification and download the most current security updates using one of the methods listed in Table 6.

Table 6 Methods of Handling Security Update Notification and Download

Notification and Download	Process for Notification and Download of Updates
Microsoft Security Notification Service with Systems Management Server	Receive email notifications of updates. An administrator must review the updates and determine if immediate distribution is required or if the update can wait until the next scheduled operating system update.
Windows Update Service with Automatic Update	Automatically download the update and then notify a local administrator that the update is available for installation.
Software Update Service with Automatic Update	Automatically download the update to the SUS server and then notify an administrator that it is available for testing and distribution to client computers.

You can configure Automatic Update, the client component used by both Windows Update Service and SUS, to send a notification, to download, or to download and install security patches. If your organization uses SMS to manage network servers, SMS can also be used to distribute security updates.

2. Evaluate the threat of the security weakness to your network environment.
3. Arrange to install the update immediately if the threat is great.
4. Test the security updates on domain controllers in a test environment.
5. Before deploying the security updates to the production environment, install and test the security update in a test lab running your organization's critical applications and operating

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

systems. The test domain controller and administrative workstation should be configured identically to those in your production environment.

6. Distribute and deploy security updates to domain controllers in your production environment using one of the methods provided in Table 7.

Table 7 Methods for Distributing and Deploying Security Updates

Deployment Method	Deploys Updates By
Windows Update Service	Configuring Windows Update to do one of the following: Inform an administrator who is interactively logged onto the Web server that updates are available and then install the updates. Automatically installing the updates.
SUS	Automatically distributing the update from an SUS server to domain controllers and administrative workstations (at a minimum). SUS can be configured to automatically install the update on computers based on some characteristic such as site, OU, or group membership.
SMS	Automatically distributing the update to domain controllers and administrative workstations (at a minimum) using SMS.

Check domain controller operating systems *weekly* to ensure that the most current service pack and hotfix upgrades have been applied to all domain controllers and administrative workstations. Table 8 describes methods for determining which service pack and hotfix upgrades have been applied to domain controllers and administrative workstations.

Table 8 Verifying Update Installations Based on Update Method

Audit Method	To Verify the Update Installation
Manual	List all service packs and security updates that have been applied to this computer with the Add or Remove Programs setting in the Control Panel.
SUS	Review SUS reports to determine which domain controllers and administrative workstations have received the update.
SMS	Create an SMS software audit through a software agent that lists domain controllers

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

	and administrative workstations that have received the update.
--	--

Managing Forest-wide Configuration Settings

Users, including anonymous users, can initiate LDAP queries of Active Directory, such as deep subtree searches. Inefficient LDAP queries can consume substantial domain controller resources, such as CPU capacity. This represents a potential denial-of-service threat with regard to Active Directory availability.

Reducing the MaxQueryDuration setting lowers the amount of time that a query runs before a domain controller considers the query to have timed out, and returns the error “timeLimitExceeded.” This increases domain controller resistance to denial-of-service attacks that exploit inefficient LDAP queries.

The MaxQueryDuration setting can be modified with the NTDSUTIL tool through the LDAP Policies menu:

- To check the current setting, see: “Viewing the MaxQueryDuration Setting with NTDSUTIL,” in Appendix B.
- To modify the current setting, see “Modifying Policy Settings with Ntdsutil,” in Appendix B.

Active Directory is configured with a default value for MaxQueryDuration of 120 (seconds). Setting this value to 30 (seconds) reduces the possibility of an LDAP query causing a denial-of-service attack.

Note: Be aware that in rare instances an application may require the longer time limit. If an application fails unexpectedly after the MaxQueryDuration setting has changed, try increasing this value or modifying the application.

Managing Service Administrator Account Security

The members of the service administrator groups represent the most powerful accounts in your forest and in each individual domain. To minimize security risks, follow the recommendations below to enforce strong oversight of administrative accounts.

Performing Periodic Audit Checks on Service Administrators

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Service administrators control the configuration and functioning of the directory service. Therefore, this responsibility should be given only to reliable, trusted individuals who demonstrate responsible ownership and who fully understand the operation of the directory.

Service administrators should also be completely familiar with your organization's policies regarding security and operations, and they should have demonstrated their willingness to enforce policies. All individuals who are granted a service administrator account should follow these practices:

- Reserve the service administrator account for tasks that require elevated privileges.
- Connect only to the corporate network with this account.
- Do not add this account to any local group on any host.
- Do not run any services under this account.
- Do not configure any system to logon automatically with this account.
- Notify "Accounts" to change or remove this account if your job responsibilities change.
- Do not use service administrator account privileges to obtain information or make modifications outside your job responsibilities.

Checking Service Administrator Trustworthiness

The following are some recommended practices for managing the membership of the service administrator groups:

- Members of the service administrator groups should undergo background checks before being granted service administrator account privileges.
- *Quarterly* audits of all individuals with service administrator credentials should be performed to ensure that they still have legitimate needs for this access.

These audits should evaluate the *least* privileges required by service administrators to perform their jobs, in a manner consistent with your organizational practices. In some cases, the user may no longer need any service administrator privileges. In other cases your organization's delegation model for Active Directory may have changed, triggering a reevaluation of least privileges.

Checking Service Administrator Group Memberships

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Having a large number of individuals in your organization with service administrator rights leaves the organization with a heightened vulnerability to “rogue administrator” attacks. Review the memberships of all service administrative groups on a *weekly* basis to:

- Review and justify any new members.
- Remove any individuals that no longer need service administrative access.
- Remove all Schema Admins group members if you are not currently modifying the schema.
- Remove any members that are from a different forest

Checking That Administrator Rights Are Properly Delegated

Review your organization’s delegation model *quarterly* and consider refining the model to minimize the:

- Number of service administrators
- Privileges required to perform these jobs

Members of the Backup Operators group can log on locally to domain controllers, archive files to backup media, and overwrite system files through restore operations. The only members of this group should be those individuals who perform domain controller backup and restore operations. To reduce the number of individuals who have these rights, do not make individuals who are responsible only for member server backup and restore operations, such as Microsoft® SQL Server operators, members of the Backup Operators group.

Avoid using the Account Operators group for strictly delegating a “data administration” task, such as account management. Because the default directory permissions give this group the ability to modify the membership of other service administrator groups, such as Server Operators, a member of the Account Operators group can elevate his or her privileges to become a service administrator.

By default, there are no members of the Account Operators group. Membership in this group should be left empty.

Managing Administrative Passwords and the Logon Process

As recommended, you should specify in your organization’s security practices that highly secure passwords be used especially for service administrator accounts. Recommendations for

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

implementations and strategies for managing administrative passwords and authentication are discussed in greater detail below.

Requiring Smart Cards for Administrative Logon

Require your service administrators to use smart cards, if possible, for their interactive logons. Besides forcing the administrative users to have physical possession of the cards to log on, smart cards also ensure the use of cryptographically strong passwords on the user accounts that are randomly generated. This helps protect against the theft of weak passwords to gain administrative access. To implement this strategy, you must have a public key infrastructure (PKI) available to authenticate the smart cards.

Require the use of smart cards for administrative logon by setting the smart card option on the administrative accounts. For the procedures to perform these tasks, see the Appendix in Best Practice Guide for Securing Active Directory Installations and Day-to-Day Operations: Part I at:

http://www.microsoft.com/windows2000/technologies/directory/AD/AD_SecurityPt1.asp

For more information on using smart cards authentication, see “The Smart Card Deployment Cookbook” at: <http://www.microsoft.com/technet/security/guidance/smrtcard/smrtcdb/default.asp>

Sharing Logons for Sensitive Administrative Accounts

For each account that is a member of the Enterprise Admins and Domain Admins groups in the forest root domain, assign two users to share that account, so that both users must be present for a successful logon with that account.

Shared administrative accounts can be implemented through either of the following methods. If you are using:

- Password-based credentials for administrative accounts, then split the password between the two administrators.
- Smart card-based credentials for administrative accounts, then split the physical possession of the smart card and knowledge of the personal identification number (PIN) for the smart card between the two administrators.

For more information on sharing administrative accounts, see “Controlling the Administrative Logon Process” in Best Practice Guide for Securing Active Directory Installations and Day-to-Day Operations: Part I at:

http://www.microsoft.com/windows2000/technologies/directory/AD/AD_SecurityPt1.asp.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Managing DS Restore Mode Administrator Passwords

The DS Restore mode password is set during the DCPROMO process and is only used during certain maintenance operations, such as directory database compression and Active Directory restore operation. Therefore, DS Restore mode passwords should be centrally stored and managed.

Like all service administrator passwords on domain controllers, these DS Restore mode passwords should be handled securely, following these recommendations:

- Require strong, complex, 14-character passwords.
- Ensure that a minimum number of trustworthy administrators can access to these passwords.
- If a user is granted temporary service administrator privilege, to compress the directory database, for example, change the password immediately.

Coordinate password information with the individual or team responsible for maintaining the password list *before* promoting a new domain controller. This helps to ensure that the password is available to service administrators and to no one else.

Maintaining Up to Date Baseline Information

Baseline information performs two distinct roles in maintaining Active Directory security: analyzing trends in values such as the directory size and providing a periodic snapshot of the configuration of the Active Directory infrastructure. In some cases, information recorded in a previous baseline snapshot can provide the fastest and easiest means of restoring this information to the directory.

Maintain up to date baseline information by completing the following tasks:

1. Create a baseline database of Active Directory infrastructure information.
2. Detect and verify infrastructure changes
3. Update Baseline information

Creating a Baseline Database of Active Directory Information

Table 9 lists the Active Directory infrastructure information that should be included in the baseline that is maintained and kept as up to date as possible.

Table 9 Information to Include In Active Directory Security Baseline

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Active Directory Infrastructure Component	Information to Record About Each Component
Audit policies	<ul style="list-style-type: none"> • List of policies that are enabled. • For each policy, if success, failure, or both are enabled for auditing.
GPOs	List of GPOs
Assignment of GPOs	List of containers and OUs that have GPOs assigned. For each container or OU, the specific GPOs that are assigned.
Existing trust relationships	List of trusts. Type of trust (two-way or one-way) Names of trusting and trusted domains
Domain Controllers OU	List of domain controllers.
Service Administrators OU	List of administrative workstations. List of service administrator accounts. List of service administrator account group memberships.
Operations Master role holders	List of domain controllers that currently hold the operations master roles.
Replication topology	List of sites. Configuration settings for each site. List of subnets within each site. Configuration settings for each subnet. List of manually created connections. Configuration settings for each manually created connection. List of site links. Configuration settings for each site link.
Database characteristics	Number of objects of each class in the database. Size of the database file (.DIT)
Service packs and hotfixes for domain controllers and administrative workstations	Current operating system. Current service packs. Current hotfixes. Current version of virus scan database. Current version of the virus scan software.
System state for domain	Collect initial system state status.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

controllers and administrative workstations	List of files that are expected to change as a normal part of activity. List of registry settings that are expected to change as a normal part of activity.
Current backup media exists	List of domain controllers with current backups
Verify directory backup media	Determine that backup media restores AD successfully
Audits of individuals with service administrator credentials	Review the needs of these individuals for the legitimate need for administrative access, and ensure they have the least privileges necessary to perform their function.

Detecting Active Directory Infrastructure Changes

Changes to the Active Directory infrastructure are identified by either a security log event, such as an addition to a service account group membership, or by an information change in the report generated for an infrastructure component.

As a part of your organization's ongoing operations, update the baseline information for the components in Table 9 anytime you make an infrastructure change. This will ensure your baseline documentation reflects the current standards within your organization. For example, if you create a new trust, immediately update your baseline documentation to reflect the addition of the new trust.

For infrastructure changes that do not generate events, reports should be generated on a specified interval. Use these reports to compare the actual configuration of Active Directory against the baseline you created previously.

Updating the Baseline Information

Once a change in the infrastructure information is detected, its validity as an authorized change should be ascertained as soon as possible. If the information has changed, the change must either be authorized, causing the baseline information to be updated or the change is unauthorized, causing the change to be rolled back to its previous state.

Based on the Active Directory infrastructure component, any changes might be detected on an ongoing basis or on a specified schedule selected for generating status reports, such as daily, weekly, or monthly. Specifying ongoing updates indicates that a baseline update should occur as soon as possible after an event is appears in the event log,

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Table 10 lists the Active Directory infrastructure components and the corresponding recommendations for establishing a:

- Method of detecting changes in specific infrastructure information
- Schedule for updating specific baseline information
- Schedule for reviewing specific baseline information

Table 10 How Infrastructure Changes Are Detected, the Frequency of Recommended Baseline Updates, and the Frequency of Baseline Reviews

Active Directory Infrastructure Component	Detection by	Update Baseline	Review Baseline
Audit policies	Event log	ongoing	Quarterly
GPOs	Event log	ongoing	Quarterly
Assignment of GPOs	Event log	ongoing	Monthly
Existing trust relationships	Event log	ongoing	Quarterly
Domain Controllers OU	Event log	ongoing	Monthly
Service Administrators OU	Event log	ongoing	Monthly
Operations Master role holders	Event log	ongoing	Monthly
Replication topology	Event log	ongoing	Monthly
Database characteristics	Automatic report	Daily	None
Service packs and hotfixes for domain controllers and administrative workstations	Automatic report	Weekly	None
System state for domain controllers and administrative workstations	Automatic report	Daily	None
Current backup media exists	Manual report	Weekly	None
Directory backup media can restore domain controllers	Manual report	Quarterly	None

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Audits of individuals with service administrator credentials	Manual report	Quarterly	Quarterly
--	---------------	-----------	-----------

Recommendations: Maintaining Secure Active Directory Operations

Follow the security recommendations presented in this chapter to help maintain a high level of security for Active Directory operations. In most instances, these recommendations apply to intranet datacenter, extranet datacenter, and branch office scenarios. However, some of the recommendations depend on the particular scenario. When the recommendations are scenario specific, notes are included to direct you to the section where the recommendation is discussed.

Maintaining Domain Controller and Administrative Workstation Security

The following table provides a checklist of recommendations for maintaining domain controller and administrative workstation security.

	Establishing Domain Controller Backup and Restore Strategies
<input type="checkbox"/>	Publish backup policies that specify which domain controllers are backed up, who backups domain controllers, how frequently they are backed up, and how backup media is handled.
<input type="checkbox"/>	Create a separate backup account for domain controllers requiring service administrative privileges
<input type="checkbox"/>	Store domain controller backup media in a secure location.
<input type="checkbox"/>	Consider backing up branch office domain controllers from media stored locally on disk.
<input type="checkbox"/>	Check backup media weekly for suitability.
<input type="checkbox"/>	Protect the Backup Operators group with special security descriptor settings.
<input type="checkbox"/>	Regularly verify that domain controller backup media is good by performing a data restore.
<input type="checkbox"/>	Practice performing a forest recovery yearly.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

	Managing the Life Cycle of Domain Controller Hardware
<input type="checkbox"/>	Follow recommendations for recycling domain controller hardware and media.
	Running Antivirus Software on Domain Controllers and Administrative Workstation
<input type="checkbox"/>	Run regular virus scans on domain controllers and administrative workstations.
<input type="checkbox"/>	Exclude certain Active Directory database and log files from virus scanning.
<input type="checkbox"/>	Exclude the SYSVOL directory tree from virus scanning.
<input type="checkbox"/>	Requiring script signing on domain controllers and administrative workstations.

Staying Current with Security Hotfixes and Service Packs

The following table provides a checklist of recommendations for staying current with security hotfixes and service packs.

	Selecting a Security Update Strategy
<input type="checkbox"/>	If you have a small organization that allows Internet access to servers consider implementing WUS.
<input type="checkbox"/>	If you currently have a solution that distributes security updates continue using the current solution.
<input type="checkbox"/>	If you currently have SMS and need a patch-management solution implement SMS patch extension.
<input type="checkbox"/>	If you do not have SMS implement SMS, SUS, or a third-party solution for patch management.
	Deploying Security Hotfixes and Service Packs
<input type="checkbox"/>	Select a method for security hotfix notification, distribution, and auditing.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

<input type="checkbox"/>	Check operating systems <i>weekly</i> to ensure that current service pack and hotfix upgrades have been applied
--------------------------	---

Managing Forest-wide Configuration Settings

The following table provides a checklist of recommendations for managing forest-wide configuration settings.

Managing Forest-Wide Configuration Settings	
<input type="checkbox"/>	Reduce the MaxQueryDuration value to 30 seconds with the NTDSUTIL tool.
<input type="checkbox"/>	If applications running on the computer no longer work, try increasing the MaxQueryDuration value

Managing Service Administrator Account Security

The following table provides a checklist of recommendations for managing service administrator account security.

Performing Periodic Audit Checks on Service Administrators	
<input type="checkbox"/>	Ensure that service administrators are familiar with your organization's security policies.
<input type="checkbox"/>	Perform periodic background checks of service administrator trustworthiness.
<input type="checkbox"/>	Periodically check service administrator group memberships
<input type="checkbox"/>	Periodically check for the proper delegation of service administrator rights.
Managing Administrative Passwords and the Logon Process	
<input type="checkbox"/>	Require smart cards on service administrator-level accounts

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

<input type="checkbox"/>	Implement a split password for the most sensitive administrator accounts.
<input type="checkbox"/>	Maintain a list of DS Restore mode passwords in a secure location.
<input type="checkbox"/>	Limit the number of administrators that have access to DS Restore mode passwords.

Maintaining Up to Date Baseline Information

The following table provides a checklist of recommendations for documenting baselin information.

Maintaining a Database of Baseline Information	
<input type="checkbox"/>	Document baseline information about Domain controllers and administrative workstations.
<input type="checkbox"/>	Update the baseline information listed in Table 9 at the frequency also provided in Table 10.

Chapter 2 - Monitoring the Active Directory Infrastructure

The monitoring (or “health-checking”) process gathers information about the security state of the Active Directory® directory service infrastructure, which includes the directory database, domain controllers, and administrative workstations for service administrators. To monitor your Active Directory infrastructure, perform the following tasks

1. Collect information in real time or at specified time intervals.
2. Compare this data with previous data or against a threshold value.
3. Respond to a security alert as directed in your organization’s practices.
4. Summarize security monitoring in one or more regularly scheduled reports.

Small organizations can manually handle security monitoring for Active Directory, but a large organization requires special monitoring software to collect and interpret this status. The complexity of Active Directory in a large organization makes manually collecting and reviewing security status

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

practically impossible. In addition, a large organization has numerous domain controllers and administrative workstations, which can best be supported by compiling status in a common database. After the information is centralized, you can review the security status of Active Directory as a whole.

Note: To provide comprehensive monitoring of Active Directory, all domain controllers and administrative workstations must be monitored. Any unmonitored computer represents a weakness in ensuring that Active Directory stays secure.

The monitoring described here is for the purpose of detecting security-sensitive configurations and not for the purpose of detecting intrusions. To detect intrusions, you need to provide additional auditing and monitoring.

The types of monitoring to perform to detect security-sensitive configurations include the following:

- Event notification

With event notification, you define thresholds for changes in the directory service, domain controller configuration, or other Active Directory infrastructure characteristics. When one of these characteristics changes enough to exceed the threshold, an event notification is generated, indicating a potential security breach.

For example, you can configure Active Directory and Microsoft® Windows 2000 Server to generate an entry in the security event log when changes are made to the site or subnet configuration in Active Directory. This software collects your monitoring information, including event log entries and performance counters, and then reports the events to operations console.

- Trend analysis

With trend analysis, you collect information as a number of data points that are only meaningful when they are examined over a period of time. Drastic changes in trends can indicate a potential security breach.

For example, you can collect status on available disk space every 15 minutes and determine if there is a steady increase in disk space usage. Over a period of time, you can analyze the trend in use of disk space to determine if a domain controller is under a potential disk space attack.

Monitor your Active Directory infrastructure by performing the following tasks:

1. Monitor changes to Active Directory.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

2. Monitor changes in domain controller status.
3. Monitor changes in system state and executables.

Monitoring Changes to Active Directory

Active Directory is an integral component in a domain's security mechanism. A big part of securing Active Directory installations is monitoring security-sensitive changes to Active Directory containers and objects. The security auditing recommendations presented in "Establishing Secure Domain Controller Policy Settings" in *Securing Active Directory Installations and Day-to-Day Operations: Part I* establishes audit settings for security-sensitive administration tasks and must be in place for you to monitor changes to Active Directory.

Note: Ensure that you detect changes in the Active Directory containers and objects within one hour after the change occurs.

Throughout this section, security-sensitive entries in the event logs are described. Because each Active Directory infrastructure contains unique information, any information that is unique has been converted to a variable. Table 11 lists the variables that are used in every table in this section. In addition to these variables, there are event-specific variables that are described in the discussion of an event.

Table 11 Variables Used In Monitoring Active Directory Events

Variable	Description
<username>	Unless otherwise noted, this variable indicates the user who performed the operation.
<computername>	Unless otherwise noted, this variable indicates the domain controller where the operation was performed.
<domain>	This variable indicates the domain where the operation was performed. For example DC=corp,DC=contoso,DC=com

Monitor changes to Active Directory containers and objects by performing the following tasks:

1. Monitor forest-level changes.
2. Monitor domain-level changes.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

3. Monitor changes in the Service Administrators OU.
4. Monitor changes to the disk space consumed by Active Directory objects.

Monitoring Forest-Level Changes

Forest-level changes in Active Directory have the broadest scope of any administrative operations, and as such they represent a large security attack surface. Forest-wide configuration changes include the following:

- Changes in the schema
- Promotion or demotion of domain controllers, especially global catalog servers
- Changes in replication topology, including changes in sites and subnets
- Changes in policies for Lightweight Directory Access Protocol (LDAP)
- Changes in the dSHeuristics attribute
- Changes in forest-wide operations master roles

Detecting Changes in the Schema

The Active Directory schema defines the structure of the directory service database. All object classes and attributes are defined in the Active Directory schema. Unauthorized changes in the schema pose a security threat, because an attacker can create, deactivate, or modify object class and attribute definitions. You cannot delete an object class or attribute, you can only deactivate an object class or attribute. Modifying an object class or attribute in the schema can disrupt Active Directory and cause denial of service for all computers and users that depend on Active Directory.

Detection

You can detect when a schema class or attribute is created, deleted, or modified by scanning the aggregated security event logs from all domain controllers for the event criteria that are specified in Table 12. The event fields that are common to all event actions (creations, deactivation, or modifications) are show in the first section of Table 12. The additional event fields for identifying each type of event action is shown in the remaining sections of Table 12.

The following variables are used in Table 12:

- *<objecttype>* can have one of following values:

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

- attributeSchema - indicates that the event references a schema attribute.
- classSchema - indicates that the event references a schema class.
- *<attributeclassname>* indicates the name of the class or attribute that is modified.
- *<propertychanged>* indicates the name of the property that is modified.

Table 12 Detecting Creations, Deletions, or Modifications in the Schema

Event field	Relevant values
	Common Fields For All Schema Events
Source	Security
Category	Directory Service Access
Type	Success
Event ID	565
User	<i><username></i>
Computer	<i><computername></i>
	Additional Fields For Schema Object Creation Event
Description:	Object Server: DS Object Type: <i><objecttype></i> Object Name: CN=Schema,CN=Configuration,DC= <i><domain></i> Accesses: Create Child
	Additional Fields For Schema Object Deactivation Event
Description:	Object Server: DS Object Type: <i><objecttype></i> Object Name: CN= <i><attributeclassname></i> ,CN=Schema,CN=Configuration,DC= <i><domain></i> Accesses: Write Property Properties: Write Property ?isDefunct
	Additional Fields For Schema Object Modification Event
Description:	Object Server: DS

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Object Type: <objecttype> Object Name: CN=<attributeclassname>,CN=Schema,CN=Configuration,DC=<domain> Accesses: Write Property Properties: Write Property ?<propertychanged>

Response

For creation and deactivation events, there is no method for determining the schema class or attribute that is created or deactivated. You need to compare existing classes and attributes with your documented list of valid classes and attributes, discussed in “Maintaining Up to Date Baseline Information” in Chapter 1, in this document.

The unauthorized modification of the Active Directory schema requires you to manually restore changes to the schema when the unauthorized schema modifications do not prevent service administrators from logging on. When service administrators are unable to log on, you must do a complete recovery of the forest. For more information about how to perform a recovery of the entire forest, see “Recovering from Catastrophic Forest-wide Corruption” in Chapter 3 of this guide.

Identifying When Domain Controllers Are Promoted or Demoted

The unauthorized promotion or demotion of domain controllers to and from the Active Directory infrastructure represents a serious compromise of Active Directory. An attacker can introduce a new, rogue domain controller to disrupt normal operations or to obtain a copy of the Active Directory database for the purposes of discovering passwords and other secure information that is stored in Active Directory. In addition, an attacker might remove a domain controller for the purposes of discovering the passwords of service administrator accounts.

Important: Give global catalog servers a higher priority in monitoring, because global catalog servers contain a complete copy of the Active Directory database, and as such they represent a large attack surface.

Detection

You can detect when a domain controller is promoted or demoted by scanning the aggregated security audit event logs from all domain controllers for the event criteria that are specified in Table 13. The event fields that are common to all event actions (promotions, or demotions) are show in the first

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

section of Table 13. The unique event fields for identifying each type of event action is shown in the remaining sections of Table 13.

Table 13 Detecting the Promotion or Demotion of a Domain Controller

Event field	Relevant values
	Common Fields For Domain Controller Promotion or Demotion Events
Source	Security
Category	Directory Service Access
Type	Success
Event ID	565
User	<username>
Computer	<computername>
	Additional Fields For Domain Controller Promotion Events
Description:	Object Server: DS Object Type: Computer Object Name: OU=Domain Controllers,DC=<domain> Accesses: Create Child
	Additional Fields For Domain Controller Demotion Events
Description:	Object Server: DS Object Type: Computer Object Name: OU=Domain Controllers,DC=<domain> Accesses: Delete Child

For domain controller promotions and demotions, you can identify the domain controller that is promoted or demoted by looking for the event listed in Table 14. The event listed in Table 14 occurs very close in the time, and on the same domain controller, to the event in Table 13 happened. <newdomaincontroller> is the name of the newly promoted or demoted domain controller.

Table 14 Identifying the Domain Controller That Is Promoted or Demoted

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Event field	Relevant values
Source	Security
Category	Directory Service Access
Type	Success
Event ID	565
User	<username>
Computer	<computername>
Description:	Object Server: DS Object Type: Computer Object Name: CN=<newdomaincontroller>,OU=Domain Controllers,DC=<domain> Accesses: Write Property

Response

The unauthorized promotion or demotion of a domain controller requires you to take further steps to ensure that the security of Active Directory is not further compromised. For more information on how to respond to the unauthorized promotion or demotion of a domain controller, see “Recovering from the Physical Breach of a Domain Controller” in Chapter 3 of this guide

Detecting Changes in the Replication Topology

Changes in replication topology include the addition or removal of Active Directory sites, subnets, and site links. An attacker can change the replication topology to:

- Disrupt the replication of Active Directory.

The sites and subnet to which domain controllers belong determine the Active Directory replication topology. An attacker can change the sites to which domain controllers belong and convince domain controllers, which are connected by slow-speed network segments, that they are in the same site and have high-speed network connectivity. This can potentially saturate the network utilization of the slow-speed network segments and prevent domain controllers from receiving timely updates.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

- Degrade performance by directing client traffic over slow-speed network segments or overloading a targeted number of domain controllers.

Client computers determine the domain controller to use for servicing Active Directory queries based on the subnets to which the domain controllers and the clients belong. An attacker can change the sites and subnets to which the domain controllers belong and convince the clients that a domain controller across a slow-speed network segment is the appropriate domain controller to use.

An attacker can also change the sites and subnets to convince a large number of clients that the same domain controller is the appropriate domain controller to use and saturate the domain controller with client requests.

Detection

You can detect when sites, subnets, site links, and connections are created, deleted, or modified by scanning the aggregated security audit event logs from all domain controllers for the event criteria that are specified in Table 15 for sites, Table 16 for subnets, Table 17 for site links, and Table 18 for connections. The event fields that are common to all event actions (creation, deletion, or modification) of sites, subnets, site links, and connections are show in the first section of Table 15, Table 16, Table 17, and Table 18. The unique event fields for identifying each type of event action is shown in the remaining sections of Table 15, Table 16, Table 17, and Table 18.

<sitename> is the name of the site that is modified. <subnetname> in Table 16 is the name of the subnet that is modified. <transporttype> in Table 17 is the type of transport for the site link and can either be IP or SMPT. <sitelinkname> in Table 17 is the name of the site link that is being modified. <connectionname> in Table 18 is the name of the connection that is being modified. <servername> is the name of the server that contains the connection. <modifiedproperties> is name of the properties that are modified.

Table 15 Detecting the Creation, Deletion, or Modification of Sites

Event field	Relevant values
	Common Fields For Site Creation, Deletion, or Modification Events
Source	Security
Category	Directory Service Access

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Type	Success
Event ID	565
User	<username>
Computer	<computername>
	Additional Fields For Site Creation Events
Description:	Object Server: DS Object Type: Computer Object Name: CN=Sites,CN=Configuration,DC=<domain> Accesses: Create Child
	Additional Fields For Site Deletion Events
Description:	Object Server: DS Object Type: site Object Name: CN=Sites,CN=Configuration,DC=<domain> Accesses: Delete Child
	Additional Fields For Site Modification Events
Description:	Object Server: DS Object Type: site Object Name: CN=<sitename>,Sites,CN=Configuration,DC=<domain> Accesses: Write Property Properties: Write Property < modifiedproperties >

Table 16 Detecting the Creation, Deletion, or Modification of Subnets

Event field	Relevant values
	Common Fields For Subnet Creation, Deletion, or Modification Events
Source	Security
Category	Directory Service Access
Type	Success
Event ID	565

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

User	<username>
Computer	<computername>
	Additional Fields For Subnet Creation Events
Description:	Object Server: DS Object Type: %{00000000-0000-0000-0000-000000000000} Object Name: CN=Subnets,CN=Sites,CN=Configuration,DC=<domain> Accesses: Create Child Properties: Create Child subnet
	Additional Fields For Subnet Deletion Events
Description:	Object Server: DS Object Type: Subnet Object Name: CN=Subnets,CN=Sites,CN=Configuration,DC=<domain> Accesses: Delete Child
	Additional Fields For Subnet Modification Events
Description:	Object Server: DS Object Type: Subnet Object Name: CN=<subnetname>,CN=Subnets,CN=Sites,CN=Configuration,DC=<domain> Accesses: Write Property Properties: Write Property < modifiedproperties >

Table 17 Detecting the Creation, Deletion, or Modification of Site Links

Event field	Relevant values
	Common Fields For Site Link Creation, Deletion, or Modification Events
Source	Security
Category	Directory Service Access
Type	Success
Event ID	565

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

User	<username>
Computer	<computername>
	Additional Fields For Site Link Creation Events
Description:	Object Server: DS Object Type: siteLink Object Name: CN=<transporttype>,CN=Inter-Site Transports,CN=Sites, CN=Configuration,DC=<domain> Accesses: Create Child
	Additional Fields For Site Link Deletion Events
Description:	Object Server: DS Object Type: siteLink Object Name: CN=<transporttype>,CN=Inter-Site Transports,CN=Sites, CN=Configuration,DC=<domain> Accesses: Delete Child
	Additional Fields For Site Link Modification Events
Description:	Object Server: DS Object Type: siteLink Object Name: CN=<sitelinkname>,CN=<transporttype>,CN=Inter-Site Transports, CN=Sites, CN=Configuration,DC=<domain> Accesses: Write Property Properties: Write Property < modifiedproperties >

Table 18 Detecting the Creation, Deletion, or Modification of Connections

Event field	Relevant values
	Common Fields For Connection Creation, Deletion, or Modification Events
Source	Security
Category	Directory Service Access
Type	Success
Event ID	565

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

User	<username>
Computer	<computername>
	Additional Fields For Connection Creation Events
Description:	Object Server: DS Object Type: nTDSConnection Object Name: CN=NTDS Settings,CN=<servername>,CN=Servers,CN=<sitename>,CN=Sites,CN=Configuration,DC=<domain> Accesses: Create Child
	Additional Fields For Connection Deletion Events
Description:	Object Server: DS Object Type: nTDSConnection Object Name: CN=NTDS Settings,CN=<servername>,CN=Servers,CN=<sitename>,CN=Sites,CN=Configuration,DC=<domain> Accesses: Delete Child
	Additional Fields For Connection Modification Events
Description:	Object Server: DS Object Type: nTDSConnection Object Name: CN=<connectionname>,CN=NTDS Settings,CN=<servername>,CN=Servers,CN=<sitename>,CN=Sites,CN=Configuration,DC=<domain> Accesses: Write Property Properties: Write Property < modifiedproperties >

Response

The unauthorized modification of replication topology requires you to perform an authoritative restore of the forest configuration from a known good backup. To restore the replication topology, you need to restore CN=Sites, CN=Configuration, DC=<ForestRootDomain> in the configuration directory partition. For more information about how perform an authoritative restore of the replication topology, see “Recovering from Data Tampering by Restoring Active Directory Data” in Chapter 3 of this guide.

You can also restore the replication topology manually. In “Maintaining Up to Date Baseline Information” in Chapter 1, of this guide, you documented the replication topology. Use this documentation to manually restore the configuration of the replication topology.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Detecting Changes in LDAP Policies

LDAP policies impose limits on LDAP queries that prevent specific operations from adversely affecting the performance of domain controllers, making it easier for domain controllers to resist denial-of-service attacks. LDAP policies are implemented through the use of objects of the queryPolicy class. You can create these objects can be created in the Query Policies container, which is a child of the directory service container in the configuration naming context.

Detection

You can detect modifications to LDAP policies by scanning the aggregated security event logs from all domain controllers for the event criteria that are specified in Table 19.

Table 19 Detecting the Modification of LDAP Policies

Event field	Relevant values
Source	Security
Category	Directory Service Access
Type	Success
Event ID	565
User	<username>
Computer	<computername>
Description:	Object Server: DS Object Type: queryPolicy Object Name: CN=Default Query Policy,CN=Query-Policies,CN=Directory Service,CN=Windows NT,CN=Services,CN=Configuration, DC=<domain> Accesses: Write Property Properties: Write Property IDAPAdminLimits

Response

The unauthorized modification of LDAP policies requires you to perform an authoritative restore of the policies from a known good backup. To restore LDAP policies, you need to restore the object CN=Default Query Policy, CN=Directory Service, CN=Windows NT, CN=Configuration,

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

DC=<ForestRootDomain> in the configuration directory partition. For more information about how to perform an authoritative restore of the LDAP policies, see “Recovering from Data Tampering by Restoring Active Directory Data” in Chapter 3 of this guide.

Detecting Changes in the dSHeuristics Attribute

The dSHeuristics attribute affects the behavior of various characteristics of the directory service, such as the ability to enable List Object functionality or the suppression of first/last name functionality in Ambiguous Name Resolution (ANR). dSHeuristics can affect the performance of domain controllers, and it can also make domain controllers resistant to denial-of-service attacks. Unauthorized changes to the dSHeuristics attribute can indicate an attempt to breach Active Directory security.

Detection

You can detect modification to the dSHeuristics attribute by scanning the aggregated security event logs from all domain controllers for event entries that have the fields listed in Table 20.

Table 20 Detecting the Modification to dSHeuristics

Event field	Relevant values
Source	Security
Category	Directory Service Access
Type	Success
Event ID	565
User	<username>
Computer	<computername>
Description:	Object Server: DS Object Type: nTDSservice Object Name: CN=Directory Service,CN=Windows NT,CN=Services,CN=Configuration, DC=<domain> Accesses: Write Property Properties: Write Property dSHeuristics

Response

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

The unauthorized modification of dSHueristics attribute requires you to perform an authoritative restore of the dSHueristics from a known good backup. To restore the dSHueristics attribute, you need to restore object CN=Directory Service, CN=Windows NT, CN=Services, CN=Configuration, DC=<ForestRootDomain> in the configuration directory partition. For more information about how perform an authoritative restore of the dSHueristics attribute, see “Recovering from Data Tampering by Restoring Active Directory Data” in Chapter 3 of this guide.

Detecting Changes in Forest-wide Operations Master Roles

Changes in forest-wide operations master roles (also known as flexible single master operations, or FSMO) are important to security because they affect the entire forest. Forest-wide operations master roles include the following:

- Schema master
- Domain naming master

Because forest-wide operations master roles are assigned to specific computers, any unauthorized change in the operations master roles can be an indication of a breach in Active Directory security.

Detection

You can detect changes in forest-wide operations master roles by scanning the aggregated security audit event logs from all domain controllers for the event criteria that are specified in Table 21. The event fields that are common to all event actions (changes in schema master or domain naming master roles) are show in the first section of Table 21. The additional event fields for identifying the individual forest-wide operations master roles is shown in the remaining sections of Table 21. In the case of changes in forest-wide operations master roles, <computername> is the name of the domain controller which currently holds the operations master role.

Table 21 Detecting Changes in the Forest-wide Operations Master Roles

Event field	Relevant values
	Common Fields For Changes in Forest-wide Operations Master Roles
Source	Security
Category	Directory Service Access
Type	Success

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Event ID	565
User	<username>
Computer	<computername>
	Additional Fields For Changes in Schema Master Role
Description:	Object Server: DS Object Type: dMD Object Name: CN=Schema,CN=Configuration,DC=<domain> Accesses: Control Access Properties: Control Access Change Schema Master
	Additional Fields For Changes in Domain Naming Master Role
Description:	Object Server: DS Object Type: crossRefContainer Object Name: CN= Partitions,CN=Configuration,DC=<domain> Accesses: Control Access Properties: Control Access Change Domain Master

Response

When you detect unauthorized changes in forest-wide operations master roles, immediately transfer back, or if necessary seize, the forest-wide operations master roles to the configuration that you have documented. Documenting forest-wide operations master roles is discussed in “Documenting and Updating Baseline Information” in Chapter 1. For more information about transferring or seizing operations master roles, see “Transferring Operations Master Roles” and “Seizing Operations Master Roles” in the Microsoft Windows 2000 Active Directory Operations Guide at:
<http://www.microsoft.com/windows2000/techno/administration/activedirectory/adops.asp>.

Monitoring Domain-Level Changes

Changes in Active Directory domains affect all users, workstations, member servers, and domain controllers in the domains, and as such they represent a large security attack surface. These domain-wide configuration changes include the following:

- Changes in domain-wide operations master roles

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

- Changes in trusts
- Changes in permissions for Administrators and Domain Admins through modification of AdminSDHolder
- Changes in the GPOs for the Domain container and the Domain Controllers OU.
- Changes in the GPO assignments for the Domain container and the Domain Controllers OU.
- Changes in the membership of built-in groups, such as Administrators and Backup Operators
- Changes to the audit policy settings for the domain.

Detecting Changes in Domain-wide Operations Master Roles

Changes in domain-wide operation master roles are critical to security because they affect the entire domain. Domain-wide operation master roles include the following:

- Relative ID (RID) master
- Primary domain controller (PDC) emulator master
- Infrastructure master

Because domain-wide operations master roles are assigned to specific computers, any unauthorized change in the operations master roles indicates a breach in Active Directory security.

Detection

You can detect changes in domain-wide operations master roles by scanning the aggregated security audit event logs from all domain controllers for the event criteria that are specified in Table 22. The event fields that are common to all event actions (changes in RID master, PDC emulator master, or infrastructure master roles) are show in the first section of Table 22. The additional event fields for identifying the individual domain-wide operations master roles is shown in the remaining sections of Table 22. In the case of changes in domain-wide operations master roles, *<computername>* is the name of the domain controller which currently holds the operations master role.

Table 22 Detecting Changes in the Domain-wide Operations Master Roles

Event field	Relevant values
-------------	-----------------

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

	Common Fields For Changes in Domain-wide Operations Master Roles
Source	Security
Category	Directory Service Access
Type	Success
Event ID	565
User	<username>
Computer	<computername>
	Additional Fields For Changes in RID Master Role
Description:	Object Server: DS Object Type: rIDManager Object Name: CN= RID Manager \$,CN=Configuration,DC=<domain> Accesses: Control Access Properties: Control Access Change RID Master
	Additional Fields For Changes in PDC Emulator Master Role
Description:	Object Server: DS Object Type: domainDNS Object Name: DC=<domain> Accesses: Control Access Properties: Control Access Change PDC
	Additional Fields For Changes in Infrastructure Master Role
Description:	Object Server: DS Object Type: infrastructureUpdate Object Name: CN= Infrastructure,DC=<domain> Accesses: Control Access Properties: Control Access Change Infrastructure Master

Response

When you detect changes in domain-wide operations master roles, immediately restore the domain-wide operations master roles to their original configuration. Documenting your domain-wide

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

operations master roles is discussed in “Documenting and Updating Baseline Information” in Chapter 1. For more information about transferring or seizing operations master roles, see “Transferring Operations Master Roles” and “Seizing Operations Master Roles” in the Microsoft Windows 2000 Active Directory Operations Guide at:

<http://www.microsoft.com/windows2000/techinfo/administration/activedirectory/adops.asp>.

Detecting Changes in Trusts

Any changes in domain trusts can cause domain-wide disruptions in Active Directory. User access to resources in one domain may depend on a trust to another domain where the user’s account resides. Breaking the trust between these domains prevents users in one domain from accessing resources in the other domain. The addition of an authorized trust can indicate a breach in Active Directory security.

Detection

You can detect changes in domain trusts by scanning the aggregated security audit event logs from all domain controllers for the event criteria that are specified in Table 23. The event fields that are common to all event actions (creation or removal of a trust) are show in the first section of Table 23. The additional event fields for identifying fields that uniquely identify a creation or removal of a trust are shown in the remaining sections of Table 23. *<trusting/trusteddomain>* in Table 23 is the name of the trusted or trusting domain created or removed.

Table 23 Detecting Changes in Domain Trusts

Event field	Relevant values
	Common Fields For Changes in Domain Trusts
Source	Security
Category	Policy Change
Type	Success
User	<i><username></i>
Computer	<i><computername></i>
	Additional Fields For Creating A Domain Trust
Event ID	610

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Description:	Domain Name: <trusting/trusteddomain>
	Additional Fields For Removing a Domain Trust
Event IDI	611
Description:	Domain Name: <trusting/trusteddomain>

Response

If you detect unauthorized changes to domain trusts, restore the original trust relationships between the affected domain and other domains. To restore the original trust relationships, you would consult to your organization's documentation for authorized trust relationships in "Documenting and Updating Baseline Information" in Chapter 1. Use this documentation to rebuild the trusts manually. For more information about how to configure domain trusts, see "Creating External Trusts" in the Microsoft Windows 2000 Active Directory Operations Guide at:

<http://www.microsoft.com/windows2000/techno/administration/activedirectory/adops.asp>

Detecting Changes in AdminSDHolder

Active Directory contains a mechanism to protect user accounts and groups that are members of service administrator groups. Every hour, the domain controller that holds the PDC emulator operations master role in the domain checks that the DACLs of these user accounts are identical to the permission list of a special AdminSDHolder object. The PDC emulator modifies any differing permission list, so that it is again identical to the permission list of AdminSDHolder.

Because any changes in the permission list for the AdminSDHolder object are applied to the members of the service administrator groups, changes to the AdminSDHolder object represent a significant security risk.

Detection

You can detect modification to the security descriptor of the AdminSDHolder object by scanning the aggregated security event logs from all domain controllers for event entries that have the fields listed in Table 24.

Table 24 Detecting the Modification to AdminSDHolder Object

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Event field	Relevant values
Source	Security
Category	Directory Service Access
Type	Success
Event ID	565
User	<username>
Computer	<computername>
Description:	Object Server: DS Object Type: container Object Name: CN= AdminSDHolder,CN=System,DC=<domain> Accesses: Write DAC

Response

If you detect unauthorized changes to the AdminSDHolder object, restore the AdminSDHolder object. To restore the AdminSDHolder object, you need to restore CN=AdminSDHolder, CN=System, DC=<DomainName> in the domain directory partition. For more information about how perform an authoritative restore of the AdminSDHolder object, see “Recovering from Data Tampering by Restoring Active Directory Data” in Chapter 3 of this guide.

Detecting Changes in Group Policy Security Settings

The information for creating or modifying Group Policy objects (GPOs) to configure the security settings on domains and domain controllers is presented in “Establishing Secure Domain Controller Policy Settings” in Chapter 4, in Part I. These security settings affect the entire domain and all domain controllers in the domain. Any unauthorized changes to these security settings could compromise the security of the domain. For example, an attacker might change the number of failed attempts before an account is locked in a group policy to make determining passwords easier.

Detection

You can detect changes in GPOs by scanning the aggregated security audit event logs from all domain controllers for the event criteria that are specified in Table 25. The event fields that are common to all event actions (creation, deletion, or modification) are show in the first section of Table

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

25. The additional event fields for identifying the individual actions are shown in the remaining sections of Table 25. *<modificationaccess>* in Table 25 can be any type of access that modifies the GPO, such as Write Property. In the modification event, *<guid>* is the GUID for the GPO that is modified.

Table 25 Detecting Changes in the GPOs

Event field	Relevant values
	Common Fields For Changes in GPOs
Source	Security
Category	Directory Service Access
Type	Success
Event ID	565
User	<i><username></i>
Computer	<i><computername></i>
	Additional Fields For Creation of GPOs
Description:	Object Server: DS Object Type: groupPolicyContainer Object Name: CN=Policies,CN=System,DC= <i><domain></i> Accesses: Create Child
	Additional Fields For Deletion of GPOs
Description:	Object Server: DS Object Type: groupPolicyContainer Object Name: CN=Policies,CN=System,DC= <i><domain></i> Accesses: Delete Child
	Additional Fields For Modification of GPOs
Description:	Object Server: DS Object Type: groupPolicyContainer Object Name: CN= <i><guid></i> ,CN=Policies,CN=System,DC= <i><domain></i> Accesses: <i><modificationaccess></i>

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

You can determine the *friendly* name of the GPO that was modified by from the <guid> in the description. The friendly name is the name that you see in a console such as Active Directory Users and Computers. For more information about converting the <guid> of a GPO to the friendly name of the GPO, see “Converting the GUID of a GPO to a Friendly Name” in “Appendix B: Deployment Procedures” in this document.

Response

If you detect unauthorized changes to the GPOs, restore the security settings to their previous configuration. To restore the group policy security settings, you need to restore the CN=Policies, CN=System, DC=<domain> in the domain directory partition. For more information about how perform an authoritative restore of the group policy security settings, see “Recovering from Data Tampering by Restoring Active Directory Data” in Chapter 3 of this guide.

You can also restore the GPOs manually. In “Maintaining Up to Date Baseline Information” in Chapter 1, in this document, you documented the GPOs. Use this documentation to manually restore the configuration of the GPOs.

Detecting Changes in GPO Assignments for the Domain Container and Domain Controllers OU

In addition to monitoring for changes in the GPOs, monitor for changes in the assignment of the GPO to a container within Active Directory. A change in the GPO assignment affects only the container where the assignment is made. AGPO can be assigned to one or more containers.

In “Establishing Secure Domain Controller Policy Settings” in Chapter 4, in Part I of this guide, GPOs were created and assigned to the Domain and to the Domain Controllers OU. Monitor for changes to the GPO assignments on both the Domain container and Domain Controllers OU.

Detection

You can detect changes in GPO assignments on the Domain container and the Domain Controllers OU by scanning the aggregated security audit event logs from all domain controllers for the event criteria that are specified in Table 26. The event fields that are common to all event actions (modifications to GPO assignments on the Domain container or the Domain Controllers OU) are show in the first section of Table 26. The additional event fields for identifying the Domain container or Domain Controllers OU are shown in the remaining sections of Table 26.

Table 26 Detecting Changes in the GPO Assignments

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Event field	Relevant values
	Common Fields For Changes in GPO Assignments
Source	Security
Category	Directory Service Access
Type	Success
Event ID	565
User	<username>
Computer	<computername>
	Additional Fields For Modification of GPO Assignments to Domain Container
Description:	Object Server: DS Object Type: domainDNS Object Name: DC=<domain> Accesses: Write Property Properties: Write Property gPLink gPOptions
	Additional Fields For Modification of GPO Assignments to Domain Controllers OU
Description:	Object Server: DS Object Type: organizationalUnit Object Name: OU=DomainControllers,DC=<domain> Accesses: Write Property Properties: Write Property gPLink gPOptions

Response

If you detect unauthorized changes to the GPO assignments in the controlled OU subtree, restore the security settings to their previous configuration. To restore the GPO assignments for the Domain container, you need to restore DC=<domain> in the domain directory partition. To restore the GPO assignments for the Domain Controllers OU, you need to restore OU=Domain Controllers,DC=<domain> in the domain directory partition. For more information about how perform

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

an authoritative restore of GPO assignments, see “Recovering from Data Tampering by Restoring Active Directory Data” in Chapter 3, in this document.

You can also restore the GPO assignments for the Domain container and the Domain Controllers OU manually. In “Maintaining Up to Date Baseline Information” in Chapter 1, in this document, you documented the GPO assignments for the Domain container and the Domain Controllers OU. Use this documentation to manually restore the configuration of the GPO assignments.

In addition, any unauthorized changes in the GPO assignments for the Domain container or the Domain Controllers OU can indicate the existence of a rogue administrator account. For more information about how to recover from a rogue administrator attack, see “Recovering from a Rogue Administrator Attack” in Chapter 3 of this guide.

Detecting Changes in the Membership of Built-in Groups

The information for creating an organizational unit (OU) subtree for controlling the group policy settings and administration of the service administrators and administrative workstations is in “Establishing Secure Service Administrator Practices” in Chapter 4, in Part I of this guide. However, the built-in service administrator groups, such as Administrators and Backup Operators, cannot be moved to the controlled OU subtree. You need to detect any changes in the group membership of these built-in groups. Any unauthorized changes to these groups can compromise the security of the domain.

Detection

You can detect changes in the membership of the built-in groups by scanning the aggregated security event logs from all domain controllers for the event criteria that are specified in Table 27.

`<accountaddedremoved>` is the name of the account that was added or removed from one of the built-in groups. `<group>` is the name of the built-in group that was modified.

Table 27 Detecting Changes in the Membership of Built-in Groups

Event field	Relevant values
Source	Security
Category	Account Management
Type	Success

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Event ID	636
User	<username>
Computer	<computername>
Description:	Member Name: <accountaddedremoved> Target Domain: Builtin Target Account ID: BUILTIN\<group>

Response

If you detect unauthorized changes in the group membership of built-in groups, immediately restore the built-in group membership to the original list of members. Documenting your built-in group membership is discussed in “Documenting and Updating Baseline Information” in Chapter 1. For more information about changing group membership, see “Manage groups” in Windows 2000 Server Help.

In addition, any unauthorized changes in the group membership of the built-in groups can indicate the creation of a rogue administrator account. For more information about how to recover from a rogue administrator attack, see “Recovering from a Rogue Administrator Attack” in Chapter 3 of this guide.

Detecting Changes in the Audit Policy Settings for the Domain Controllers OU

The audit policy settings for the Domain Controllers OU affect your notification of security-related events. As a result, any change in the audit policy can affect the notifications you receive and can also be an indication of a rogue administrator. The audit policies for receiving notification of security-related events that affect the administration of Active Directory were discussed in “Establishing Domain Controller Audit Policy Settings” in Chapter 4, in Part I. Review the audit policy settings described there and use them as a baseline for your audit policy settings.

Detection

You can detect changes in the audit policy settings for the domain by scanning the aggregated security event logs from all domain controllers for the event criteria that are specified in Table 28. You can identify the new Audit policy settings, <newpolicysettings>, by looking at the Description event field.

Table 28 Detecting Changes in the Audit Policy Settings

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Event field	Relevant values
Source	Security
Category	Account Management
Type	Success
Event ID	612
User	NT AUTHORITY\SYSTEM
Computer	<computername>
Description:	Audit Policy Change: New Policy: <newpolicysettings>

The following is an example of how the new audit policy settings, <newpolicysettings>, are displayed in the Description event field of event 612. A plus sign (+) indicates that the corresponding success or failure of the event category is audited. A minus sign (-) indicates the corresponding success or failure of the event category is not audited.

Audit Policy Change:

New Policy:

Success Failure

- + - Logon/Logoff
- - Object Access
- + + Privilege Use
- + - Account Management
- + - Policy Change
- + - System
- - Detailed Tracking
- + - Directory Service Access
- + - Account Logon

Note: When the "Audit policy change" policy is enabled for either success or failure in the Default Domain Policy or Default Domain Controllers Policy group policy objects (GPO), a success event, event 617, is logged in the Windows 2000 Security log regardless of whether or not a policy change occurred. For more information about this behavior, see Microsoft Knowledge Base Article - 272460 "Information About Event 617 in the Security Event Log" at <http://support.microsoft.com/?kbid=272460>.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Response

If you detect unauthorized changes in the audit policy settings for the domain, immediately restore the audit policies to their original settings. In “Maintaining Up to Date Baseline Information” in Chapter 1, in this document, you documented the audit policy settings. Use this documentation to manually restore the configuration of the audit policy settings.

In addition, any unauthorized changes in the audit policy settings can indicate the creation of a rogue administrator account. For more information about how to recover from a rogue administrator attack, see “Recovering from a Rogue Administrator Attack” in Chapter 3 of this guide.

Monitoring Changes in the Service Administrators OU

The information for creating the Service Administrators OU is in “Establishing Secure Service Administration Practices” in Chapter 5, in Part I, illustrated in Figure 6. Any changes to the objects in the Service Administrators OU might indicate a possible security attack.

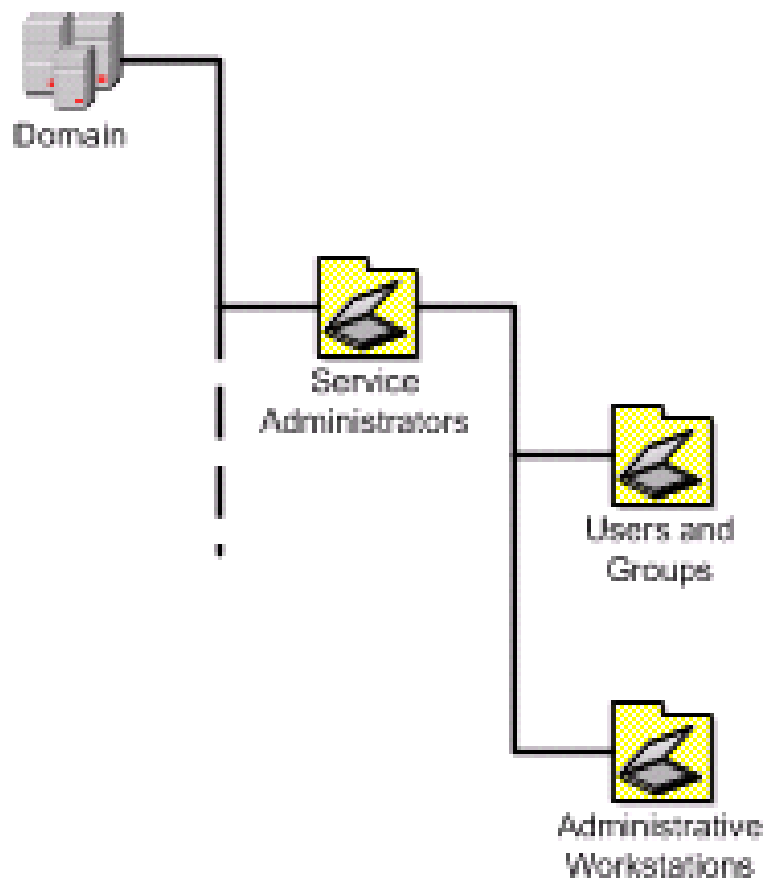


Figure 6: Placement of Service Administrators OU in a Domain

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Monitor for the following changes in the Service Administrators OU:

- Changes in service administrator accounts
- Changes in GPOs for the controlled OU subtree
- Changes in GPO assignments for the controlled OU subtree

Detecting Changes in Service Administrator Accounts

Any unauthorized changes to the service accounts in the Service Administrators OU can indicate a potential attack, including the possible creation of a rogue service administrator account. These changes include:

- The creation, deletion, and modification of service administrator user and group accounts in the Users and Groups OU.
- The addition and deletion of computers in the Administrative Workstations OU.

Detection

Table 29 includes a section for the event fields that are common to all event actions (addition, deletion, and modification) for the service administrator accounts stored in the Users and Groups OU. The additional event fields for identifying types of action are shown in the remaining sections of Table 29. *<objecttype>* in Table 29 can be either user or group, depending on the object being created, deleted, or modified. *<user/groupname>* is the name of the user or group that is modified. *<changedattribute>* is the attribute of the user or group that is modified. *<propertysetname>* is the name of the property set of which the changed attribute is a member.

Table 29 Detecting Changes to the Users and Groups OU

Event field	Relevant values
	Common Fields For Changes to the Users and Groups OU
Source	Security
Category	Account Management
Type	Success
Event ID	565

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

User	<username>
Computer	<computername>
	Additional Fields For Creations in the Users and Groups OU
Description:	Object Server: DS Object Type: <objecttype> Object Name: OU=Users and Groups,OU=Service Administrators,DC=<domain> Accesses: Create Child
	Additional Fields For Deletions in the Users and Groups OU
Description:	Object Server: DS Object Type: <objecttype> Object Name: OU=Users and Groups,OU=Service Administrators,DC=<domain> Accesses: Delete Child
	Additional Fields For Modifications to the Accounts in the Users and Groups OU
Description:	Object Server: DS Object Type: <objecttype> Object Name: CN=<user/groupname>,OU=Users and Groups,OU=Service Administrators, DC=<domain> Accesses: Write Property Properties: Write Property <propertysetname> <changedattribute>

To find the name of the account added or deleted in Table 29, look for events listed in Table 30. Table 30 includes a section for the event fields that are common to all event actions (addition and deletion) for the service administrator accounts stored in the Users and Groups OU. The additional event fields for identifying types of action are shown in the remaining sections of Table 30. *newusername*> in Table 30 is the name of the account that is created. *<targetusername>* in Table 30 is the name of the account that is removed.

Table 30 Identifying the Accounts Created or Deleted

Event field	Relevant values
	Common Fields For Creations or Deletions in the Users and Groups OU

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Source	Security
Category	Account Management
Type	Success
User	<username>
Computer	<computername>
	Additional Fields For Creations in the Users and Groups OU
Event ID	624
Description:	New Account Name: <newusername>
	Additional Fields For Deletions in the Users and Groups OU
Event ID	630
Description:	Target Account Name: <targetusername>

Table 31 includes a section for the event fields that are common to all event actions (addition and deletions) for the Administrator Workstations OU. The additional event fields for identifying types of action are shown in the remaining sections of Table 31. The **Object Type** in Table 31 should only be computer. If an object type other than computer is created in the Administrator Workstations OU, treat the object as a rogue object.

Table 31 Detecting Changes to the Administrator Workstations OU

Event field	Relevant values
	Common Fields For Changes to the Administrator Workstations OU
Source	Security
Category	Directory Service Access
Type	Success
Event ID	565
User	<username>

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Computer	<computername>
	Additional Fields For Additions to the Administrator Workstations OU
Description:	Object Server: DS Object Type: computer Object Name: OU=Administrator Workstations,OU=Service Administrators,DC=<domain> Accesses: Create Child
	Additional Fields For Deletions to the Administrator Workstations OU
Description:	Object Server: DS Object Type: computer Object Name: OU=Administrator Workstations,OU=Service Administrators,DC=<domain> Accesses: Delete Child

Response

If you detect unauthorized changes in the service administrator accounts, immediately restore the list of service administrators and group membership lists to the original list of members. Documenting your service administrator accounts is discussed in “Documenting and Updating Baseline Information” in Chapter 1.

In addition, any unauthorized changes in the service administrator accounts can indicate the creation of a rogue administrator account. For more information about how to recover from a rogue administrator attack, see “Recovering from a Rogue Administrator Attack” in Chapter 3 of this guide.

Detecting Changes in GPOs for the Service Administrators OU

The information for creating or modifying GPOs to configure the security settings on service administrators and administrative workstations is presented in “Establishing Secure Service Administration Practices” in Chapter 5, in Part I. These security settings affect the entire controlled OU. Any unauthorized changes to these security settings could compromise the security of the domain, because the service administrator accounts and secured workstations are stored in this controlled OU.

Detection

You can detect changes in GPOs by scanning the aggregated security audit event logs from all domain controllers for the event criteria that are specified in Table 32. The event fields that are

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

common to all event actions (creation, deletion, or modification) are show in the first section of Table 32. The additional event fields for identifying the individual actions are shown in the remaining sections of Table 32. *<modificationaccess>* in Table 32 can be any type of access that modifies the GPO, such as Write Property. In the modification event, *<guid>* is the GUID for the GPO that is modified.

Table 32 Detecting Changes in the GPOs

Event field	Relevant values
	Common Fields For Changes in GPOs
Source	Security
Category	Directory Service Access
Type	Success
Event ID	565
User	<i><username></i>
Computer	<i><computername></i>
	Additional Fields For Creation of GPOs
Description:	Object Server: DS Object Type: groupPolicyContainer Object Name: CN=Policies,CN=System,DC= <i><domain></i> Accesses: Create Child
	Additional Fields For Deletion of GPOs
Description:	Object Server: DS Object Type: groupPolicyContainer Object Name: CN=Policies,CN=System,DC= <i><domain></i> Accesses: Delete Child
	Additional Fields For Modification of GPOs
Description:	Object Server: DS Object Type: groupPolicyContainer Object Name: CN= <i><guid></i> ,CN=Policies,CN=System,DC= <i><domain></i> Accesses: <i><modificationaccess></i>

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

You can determine the *friendly* name of the GPO that was modified by from the <guid> in the description. The friendly name is the name that you see in a console such as Active Directory Users and Computers. For more information about converting the <guid> of a GPO to the friendly name of the GPO, see “Converting the GUID of a GPO to a Friendly Name” in “Appendix B: Deployment Procedures” in this document.

Response

If you detect unauthorized changes to the GPOs, restore the security settings to their previous configuration. To restore the group policy security settings, you need to restore the CN=Policies, CN=System, DC=<domain> in the domain directory partition. For more information about how perform an authoritative restore of the group policy security settings, see “Recovering from Data Tampering by Restoring Active Directory Data” in Chapter 3 of this guide.

You can also restore the GPOs manually. In “Maintaining Up to Date Baseline Information” in Chapter 1, in this document, you documented the GPOs. Use this documentation to manually restore the configuration of the GPOs.

Detecting Changes in GPO Assignments for the Service Administrators OU

In addition to monitoring for changes in the GPOs, monitor for changes in the assignment of the GPO to the Service Administrators OU. A change in the GPO assignment can indicate that an attacker is attempting to weaken security-related group policy settings. Any unauthorized changes to the GPO assignment in the Service Administrator OU can indicate a potential attack and the possible existence of a rogue service administrator account.

Detection

You can detect changes in GPO assignments on the Service Administrators OU by scanning the aggregated security audit event logs from all domain controllers for the event criteria that are specified in Table 33.

Table 33 Detecting Changes in the GPO Assignments on the Service Administrators Controlled OU

Event field	Relevant values
Source	Security

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Category	Directory Service Access
Type	Success
Event ID	565
User	<username>
Computer	<computername>
Description:	Object Server: DS Object Type: domainDNS Object Name: OU=Service Administrators,DC=<domain> Accesses: Write Property Properties: Write Property gPLink gPOptions

Response

If you detect unauthorized changes to the GPO assignments in the Service Administrators OU, restore the security settings to their previous configuration. To restore the GPO assignments for the Service Administrators OU, you need to restore the OU=Service Administrators, DC=<domain> in the domain directory partition. For more information about how perform an authoritative restore of GPO assignments in the Service Administrator OU, see “Recovering from Data Tampering by Restoring Active Directory Data” in Chapter 3, in this document.

You can also restore the GPO assignments for the Service Administrator OU manually. In “Maintaining Up to Date Baseline Information” in Chapter 1, in this document, you documented the GPO assignments for the Service Administrators OU. Use this documentation to manually restore the configuration of the GPO assignments.

In addition, any unauthorized changes in the GPO assignments for the Service Administrators OU can indicate the existence of a rogue administrator account. For more information about how to recover from a rogue administrator attack, see “Recovering from a Rogue Administrator Attack” in Chapter 3 of this guide.

Monitoring for Disk Space Consumed by Active Directory Objects

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

One of many potential types of attacks is the creation of rogue objects or the modification of existing objects in Active Directory and the resulting consumption of available disk space. When the available disk space is exhausted, the Active Directory database is unable to be enlarged for legitimate objects.

The types of Active Directory objects attacks that can be performed include the creation of:

- A sufficiently large number of normal-sized objects, so that the objects consume the available disk space, also known as a *rogue object flood attack*.
- New objects or modification of existing objects, so that a limited number of extraordinarily large-sized objects consume the available disk space, also known as a *large-sized object attack*.

Monitoring for a Rogue Object Flood Attack

In this situation, the attacker creates an inordinately large number of normal-sized objects over a period of time. The attacker can create the objects over a long or short period of time. If you monitor objects over a long period of time, it can be difficult to determine which objects are the rogue objects.

Detection

Detect a rogue object flood attack by performing the following tasks:

1. Create the ObjCountByClass.vbs script.

For more information about the ObjCountByClass.vbs script source and how to create the script, see “Monitoring the Number of Objects In a Domain” in “Appendix B: Deployment Procedures” in this document.

Note: This script requires Windows 2000 with Service Pack 3 or later and Windows Scripting Host version 5.6 or later.

2. For each domain in the forest, run ObjCountByClass.vbs daily to collect the number of each objects of each object class type in the domain.

You must run the ObjCountByClass.vbs script in the cscript environment from a command line (or a batch file) by entering the following command:

```
cscript ObjCountByClass.vbs
```

Note: The ObjCountByClass.vbs script must be run in the cscript environment because the Windows Scripting Host (WSH) environment does not support the necessary objects.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

3. Collect the data files created by ObjCountByClass.vbs into a database or spreadsheet to enable trend analysis of object type counts in the domain.

The ObjCountByClass.vbs script creates a status file in comma-separated value (CSV) format, named ObjectClassCount-*date-time*.csv, where *date* is the date the status file was created and *time* is the time that the status file was created. You can import the status file into Microsoft® Excel or Microsoft Access for tracking historical trends and performing forensic analysis.

Generate an alert when an inordinate number of objects, for a specific object type, are created over a period of time.

Response

As an immediate response, delete the reserve file on the affected domain controllers to create free disk space so that normal operation can be restored immediately. As a long-term response, identify the rogue objects in Active Directory and then remove them. For more information about how to respond and recover from a rogue object flood attack, see “Recovering from a Rogue Object Flood Attack” in Chapter 3 of this guide.

Monitoring for a Large-sized Object Attack

In this situation, the attacker creates a limited number of objects, but each object is extraordinarily large in size. These attacks tend to happen over a relatively short period of time. Because the objects are large in size, only a few objects are required to consume the available disk space.

Detection

Detect a large-sized object attack that consumes disk space by performing the following tasks:

1. For each domain controller in the forest, create alerts in Performance Logs and Alerts that detect when only 10 percent of free disk space is available on the disk volume that contains the Active Directory database.

For more information about creating alerts to monitor available disk space, see “Monitoring Changes in Domain Controller Status” later in this chapter.

2. For each domain in the forest, run ObjCountByClass.vbs daily to collect the number of each objects of each object class type in the domain.

You must run the ObjCountByClass.vbs script in the cscript environment from a command line (or a batch file) by entering the following command line and then pressing ENTER:

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

cscript ObjCountByClass.vbs

When the size of the database is increasing significantly, but the number of objects is not increasing proportional to the growth, then a limited number of large-sized objects are consuming the database, and subsequently the available disk space.

Response

As an immediate response, delete the reserve file on the affected domain controllers to create free disk space so that normal operation can be restored immediately. As a long-term response, identify the rogue objects in Active Directory and then remove them. For more information on how to respond and recover from a large-sized object attack, see “Recovering from an Object Growth Attack” in Chapter 3 of this guide.

Monitoring Changes in Domain Controller Status

In addition to monitoring changes in Active Directory, monitor the domain controllers in your Active Directory infrastructure. The overall health of Active Directory is only as good as the cumulative health of the domain controllers in your organization.

The domain controllers deployment recommendations presented in “Establishing Secure Domain Controller Policy Settings” in *Securing Active Directory Installations and Day-to-Day Operations: Part I* help to ensure that the domain controllers are deployed in a secure manner. Monitoring security-sensitive changes to the domain controller status helps to ensure that the domain controllers stay secure.

Monitor changes to the domain controller status by performing the following tasks:

1. Monitor domain controller availability to ensure that domain controllers are active and have not been restarted.
2. Monitor changes in domain controller performance counters for system resources.

Monitoring Domain Controller Availability

One of the primary reasons for ensuring that domain controllers are secure is to help guarantee domain controller uptime. From a security perspective, availability can be affected when an attacker removes a domain controller or performs denial-of-service attacks on a domain controller. When a domain controller is unavailable, the security implication is that the domain controller might be physically breached.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Detect when a domain controller is unavailable by performing the following tasks:

1. Monitor domain controllers to determine if the domain controllers are active.
2. Monitor the event log for domain controller restarts.

Monitoring Domain Controllers for Active Status

One of the best ways to verify that a domain controller is online and servicing client requests is to send an LDAP query to the domain controller. By performing this query, you can determine:

- The active or inactive status of the domain controller, if the domain controller responds.
- The current utilization of the domain controller, by how quickly the domain controller responds.

Detection

Most monitoring software provides the ability to periodically interrogate a computer and determine if the computer is active and servicing requests. For example, Microsoft Operations Manager has an Active Directory Management Pack that periodically queries a domain controller to determine if the domain controller is online and servicing client requests.

You can also determine if a domain controller is online and servicing client requests by executing the following LDAP query:

```
*****
!* File:      DSPing.vbs          *
!* Created:   March 2003         *
!* Version:   1.0                *
!*           *
!* Description: Diagnostic utility that attempts to connect to the *
!*             rootDSE entry of an AD domain controller to return *
!*             the DnsHostName property. The purpose is to provide *
!*             an equivalent utility to ping that checks for the *
!*             availability of the directory service on an Active *
!*             Directory domain controller. *
!*           *
!* Compatibility: This script requires WSH 5.6, CScript, ADSI *
!*             and access to Active Directory *
*****
```

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Option Explicit

'Define any constants used in this script

Const LDAP_SERVER_DOWN = &h8007203a

'Declare global variables

Dim objArgs, strServerName, strMessage

'Use Named Arguments collection for the command line argument.

'The WSHArguments Object is included in WSH version 5.6 and later

Set objArgs = WScript.Arguments.Named

strServerName = objArgs.Item("dc")

If WScript.Arguments.Named.Count < 1 Then

 Call Usage()

 WScript.Quit

Elseif Not Wscript.Arguments.Named.Exists("dc") Then

 Call Usage()

 WScript.Quit

Else

 strMessage = PingDS(strServerName)

 WScript.Echo strMessage

End If

'* Routine: Usage

Sub Usage()

 WScript.Echo "Usage: dsping /dc:target_name" & VbCrLf & _

 "For target_name, specify the ip address or name (NetBIOS name or FQDN)" & VbCrLf & _

 "of an Active Directory domain controller."

End Sub

'* Function: PingDS

Function PingDS(ServerName)

 Dim objRootDSE, strDNSHostName

 On Error Resume Next

 Set objRootDSE = GetObject("LDAP://" & serverName & "/rootDSE")

 If err.number = LDAP_SERVER_DOWN Then

 PingDS = ServerName & " did not reply."

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

```
Else
  On Error GoTo 0
  strDNSHostName = "LDAP://" & objRootDSE.Get("DnsHostName")
  PingDS = "DnsHostName: " & strDNSHostName & " replied."
End If
End Function
```

Note: This script requires Windows 2000 with Service Pack 3 or later and Windows Scripting Host version 5.6 or later.

Check the availability of your domain controllers every 15 minutes to ensure that a domain controller has not been physically breached.

Response

A domain controller that fails to respond to queries can indicate that the domain controller has been physically breached. For more information on how to respond to a physically breached domain controller, see “Recovering from the Physical Breach of a Domain Controller” in Chapter 3 of this guide.

A domain controller that does not respond to queries within a given period of time can indicate the domain controller is under some type of denial-of-service attack. These attacks can include attacks performed by a rogue administrator or rogue object flood attacks.

For more information about how to recover from a rogue administrator attack, see “Recovering from a Rogue Administrator Attack” in Chapter 3 of this guide. For more information about how to recover from a rogue object flood attack, see “Recovering from a Rogue Object Flood Attack” in Chapter 3 of this guide.

Monitoring for Domain Controller Restarts

The unauthorized restart of a domain controller can indicate that a domain controller has been physically breached. When the SYSKEY password is not stored on the domain controller, the unauthorized restart of a domain controller can also indicate the presence of a rogue service administrator because the SYSKEY password is only given to service administrators.

Detection

You can detect domain controller restarts by scanning the aggregated system event logs from all domain controllers for the event criteria that are specified in Table 34.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Table 34 Criteria for Detecting Domain Controller Restarts

Event field	Relevant values
Source	Security
Category	System Event
Type	Success
Event ID	512
User	NT AUTHORITY\SYSTEM
Computer	<computername>
Description:	Windows NT is starting up.

Response

The unauthorized restart of a domain controller can indicate that the domain controller has been physically breached and that a rogue administrator might exist. For more information on how to respond to a physically breached domain controller, see “Recovering from the Physical Breach of a Domain Controller” in Chapter 3 of this guide. For more information about how to recover from a rogue administrator attack, see “Recovering from a Rogue Administrator Attack” in Chapter 3 of this guide.

Monitoring Changes in Domain Controller Performance Counters

You can also monitor changes in Windows 2000 performance counters on a domain controller to determine its general health. Because the focus of many attacks is to negatively affect the health of the domain controller, examining these performance counters can give you an indication that the domain controller is under attack.

The domain controller health indicators can also be indicative of non-security-related issues as well. Resolving these issues, regardless of whether the intent was malicious or not, is the same. With security-related changes to health indicators, the reason behind the change in domain controller health indicators is malicious intent. If you suspect malicious intent, you need to determine if a rouge user instigated the attack.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Table 35 lists the Windows 2000 system resource performance counters that you should monitor, the threshold values that indicate a domain controller health problem, and the rationale for the threshold values.

Table 35 Windows 2000 System Resource Performance Counters for Security-related Monitoring

Performance Counter	Indication of Health Problem	Rationale
Object: Processor Counter: % Processor Time Instance: _Total	Greater than 80% utilization on all processors	Greater than 80% processor utilization indicates that the processor is over-utilized and the domain controller might be under a denial-of-service attack.
Object: LogicalDisk Counter: % Free Space Instance: all instances	Less than 25% of the total disk space of each drive on the domain controller	When less than 25% of the total disk space is available on the disk that contains the database, a rogue object flood attack might be occurring.
Object: Memory Counter: Available Bytes Instance: N/A	Less than 10% of the physical memory in the domain controller	When less than 10% of the physical memory is available, this indicates that the memory is over-utilized and the domain controller might be under a denial-of-service attack.
Object: Memory Counter: Pages/sec Instance: N/A	Increase of any kind	A significant increase in pages/sec in conjunction with low Memory object and Available Bytes counter indicates that the memory is over-utilized and that the domain controller might be under a denial-of-service attack.

Table 36 lists the LDAP performance counters that you should monitor, the threshold values that indicate a domain controller health problem, and the rationale for the threshold values.

Table 36 LDAP Performance Counters for Security-related Monitoring

Performance	Indication of Health	Rationale
-------------	----------------------	-----------

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Counter	Problem	
Object: NTDS Counter: LDAP Bind Time Instance: N/A	Trend of increase in length of time.	A significant increase in the length of time to perform an LDAP bind can indicate that the domain controller is over utilized and might be under a denial-of-service attack.
Object: NTDS Counter: LDAP Successful Binds/sec Instance: N/A	Trend of decrease in number of successful LDAP binds.	A significant decrease in the number of successful LDAP binds can indicate that the domain controller is over utilized and might be under a denial-of-service attack.

Monitoring Changes in System State and Executables

You should monitor changes in the system state and in the executable files on domain controllers and administrative workstations to help detect when an attempt to compromise a domain controller or administrative workstation occurs. Compromising one of these computers can represent a first step in further compromising the security of the Active Directory infrastructure.

Monitor changes to the domain controllers and administrative workstations, because these are the computers where service administrators log on and perform administrative functions. Because service administrators frequently log on to these computers, attackers focus their attention on these computers.

The *System state*, which is stored locally on each domain controller or administrative workstation, includes:

- Configuration for the operating system and critical applications.
- Registry and other configuration files.

If attackers modify the system state, they can render a domain controller unusable to the point of disrupting normal Active Directory operation and preventing users from using Active Directory. Or, the attacker can modify the system state to introduce a rogue application that can compromise the security of the domain controller or administrative workstation.

Attackers can also place executable files, such as viruses or Trojan horse programs, on the domain controllers and administrative workstations. These executables can be files that:

- Are in addition to the existing executables.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

With this type of attack, the attacker places new executables on the computer and then modifies the system state to run the rogue application. The next time the computer restarts, the rogue application is run and the security of Active Directory or the operating system is compromised.

- Replace existing executables.

With this type of attack, attackers replace existing executables on the computer with a rogue application. The next time a service administrator or the operating system attempts to run the valid, replaced application, the rogue application is run and the security of Active Directory or the operating system is compromised.

The monitoring of changes in the system state and executable files on domain controllers and administrative workstations is accomplished by having *system state monitoring software*. This section describes the characteristics of system state monitoring software. You can purchase software that does this type of monitoring, such as Microsoft System Management Server 2000 or other similar non-Microsoft software. You can also create your own software to perform system state monitoring.

Monitor changes in the system state and executable files on domain controllers and administrative workstations by performing the following tasks:

1. Identify how the software that monitors changes in the system state and executables on domain controllers and administrative workstations operate.
2. Create a baseline reference for the operating system and system state.
3. Monitor changes in the baseline reference.

Identifying the Characteristics of System State and Executable Monitoring Software

Monitor the domain controllers and administrative workstations for changes in the system state and executables as close to real time as possible. The sooner you detect a change in the system state or executables, the more you can limit the extent to which Active Directory is compromised.

The following are the characteristics of what the system state monitoring software should perform so that it can detect changes in the system state and executables:

- Creates a point-in-time list of the system state configuration settings and software inventory of the executables.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

The point-in-time snapshot of the computer provides a baseline for the software to use in determining when any changes in the system state or executables occur.

- Periodically compares the current system state configuration settings and list of executables to the point-in-time list and inventory.

The software compares the current system state and list of executables with the baseline to determine if:

- Any changes to the system state occur, such as additions, removals, or updates to the registry.
- New executables are placed on the computer.
- Changes in the size, date time stamp, owner, and other attributes of any existing executables.
- Any previously existing executable has been removed.

Creating an Operating System Baseline Reference

Immediately after you install your domain controller and administrative workstation, create a baseline reference of the system state. Before you create the baseline for a domain controller or an administrative workstation, ensure that you:

- Install the domain controller and administrative workstation in a secure configuration.

To install the domain controller, use the recommendations in “Deploying Secure Domain Controllers” in Chapter 2 of *Securing Active Directory Installations and Day-to-Day Operations: Part I*. Install an administrative workstation by using the recommendations in “Securing Service Administrator Workstations” in Chapter 4 in *Securing Active Directory Installations and Day-to-Day Operation: Part I*.

- Install any recent service packs and hotfixes as recommended in “Staying Current with Security Hotfixes and Service Packs” in Chapter 1, in this document.
- Run a virus scan on all disk volumes as recommended in “Running Antivirus Software on Domain Controllers and Administrative Workstations” in Chapter 1 in this document.

If the configuration of the domain controller or administrative workstation is not stabilized before you create the baseline, the monitoring software reports any updates as changes in the system state and executables. These legitimate changes may be misinterpreted as attempts to compromise the

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

integrity of Active Directory. So, if you must perform updates after creating the baseline, notify operation staff of the scope of the updates and the computers that are going to be updated.

Monitoring Changes in the Baseline Reference

The primary decision with regard to monitoring changes in the baseline reference for domain controllers and administrative workstations is how often to monitor for changes. Monitor as frequently as possible to ensure that you can limit the window in which the Active Directory infrastructure is compromised. However, the monitoring of changes consumes system resources, such as an increase in disk activity or processor utilization, on the computer being scanned.

Follow these recommendations when monitoring for changes in the baseline reference:

- Monitor for changes in the baseline reference for domain controllers and administrative workstations weekly at a minimum.

Determine the tradeoff between the frequency of monitoring for changes and the consumption of system resources by the monitoring process when scanning for changes in the system state and the executables.

- Perform baseline comparisons during periods of time when the domain controllers and administrative workstations are minimally used.

Because the monitoring process consumes system resources, scan for changes in the system state and the executables during periods of time when system resource use is minimal.

- Exclude or ignore changes that are a normal part of the operating system's function.

Many system state settings and files change as a normal part of the operating system's function. For example, the paging file, Active Directory database files, log files, and some registry entries change as a normal part of the day-to-day operation of domain controllers. There are similar changes on administrative workstations.

Most system state monitoring software are configured by default to exclude the system state settings and files for the operating system. In addition, you can determine if other files need to be excluded by observing the system state settings and files that change and then excluding the files that you determine are a legitimate changes in the system state. You can ignore these files to ensure that only actual security breaches are reported.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Important: You can ignore files that are modified by the operating system, because they are not executables. Consider any modification of an executable to be a compromise in security for a domain controller or an administrative workstation.

Recommendations: Monitoring the Active Directory Infrastructure

Following the security recommendations described earlier in this chapter helps to minimize the security risks involved in monitoring the Active Directory infrastructure. Of course, as previously mentioned, you should consider the recommendations described in other chapters when considering how best to enhance your comprehensive Active Directory security.

In most instances, these recommendations are intended for intranet datacenter, extranet datacenter, and branch office scenarios. However, some of the recommendations depend on the particular scenario. When the recommendations are scenario specific, references are included to direct you to the sections of this guide where the recommendations are discussed.

Monitoring Changes to Active Directory

The following table provides a checklist of recommendations for monitoring forest-level and domain-level changes to Active Directory and monitoring changes in service administrator accounts, administrative workstations, and disk space.

Monitoring Forest-level Changes	
<input type="checkbox"/>	Detect changes in the Active Directory schema.
<input type="checkbox"/>	Identify when domain controllers are added or removed.
<input type="checkbox"/>	Detect changes in replication topology.
<input type="checkbox"/>	Detect changes in LDAP policies.
<input type="checkbox"/>	Detect changes in dSHeuristics.
<input type="checkbox"/>	Detect changes in forest-wide operations master roles.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Monitoring Domain-level Changes	
<input type="checkbox"/>	Detect changes in domain-wide operations master roles.
<input type="checkbox"/>	Detect changes in trusts.
<input type="checkbox"/>	Detect changes in AdminSDHolder.
<input type="checkbox"/>	Detect changes in GPOs for the Domain container and the Domain Controllers OU.
<input type="checkbox"/>	Detect changes in GPO assignments for the Domain container and the Domain Controllers OU.
<input type="checkbox"/>	Detect changes in the membership of the built-in groups.
<input type="checkbox"/>	Detect changes in the audit policy settings for the domain.
Monitoring Service Administrator and Administrative Workstation Changes	
<input type="checkbox"/>	Detect changes in service administrator accounts.
<input type="checkbox"/>	Detect changes in GPOs for the Service Administrators controlled subtree.
<input type="checkbox"/>	Detect changes in GPO assignments for the Service Administrators controlled subtree.
Monitoring for Disk Space Consumed by Active Directory Objects	
<input type="checkbox"/>	Monitor for an inordinately large number of normal-sized objects.
<input type="checkbox"/>	Monitor for a limited number of extraordinarily large-sized objects.

Monitoring Domain Controller Status

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

The following table provides a checklist of recommendations for monitoring the status of individual domain controllers.

Monitoring Domain Controller Availability	
<input type="checkbox"/>	Monitor domain controllers for active status.
<input type="checkbox"/>	Monitor domain controllers for restarts.
Monitoring Changes in Domain Controller Performance Counters	
<input type="checkbox"/>	Detect changes in domain controller system resources.
<input type="checkbox"/>	Detect changes in LDAP responsiveness.

Monitoring Changes in System State and Executables

The following table provides a checklist of recommendations for monitoring changes in the system state and executables of domain controllers and administrative workstations.

Identifying the Characteristics of System State and Executable Monitoring Software	
<input type="checkbox"/>	Identify the characteristics of the software that monitors changes in the system state and executables on domain controllers and administrative workstations.
Creating an Operating System Baseline Reference	
<input type="checkbox"/>	Create a baseline of the system state and executables for the operating system to be used for future comparison.
Monitoring Changes in the Baseline Reference	
<input type="checkbox"/>	Monitor changes in the current system state and executables by comparing them to the baseline reference.

Chapter 3 - Recovering from Active Directory Attacks

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Chapter 2 of this document provides recommendations for monitoring your network and identifying abnormal activity that might indicate that Active Directory is under attack. This chapter provides recommendations for returning the Active Directory® directory service and database to its pre-attack state, if possible. Recommendations for locating the attacker and neutralizing the attack are outside of the scope of this guide.

If you can confirm an attack on Active Directory has occurred, ascertain immediately if there have been any unauthorized modifications to Active Directory or its database. If so, begin the recovery process to:

- Reestablish the prescribed Active Directory security configuration.
- Restore the normal directory service and configuration.
- Restore Active Directory database content.

Note: This chapter assumes that an attack has occurred in the past. Do not attempt to recover Active Directory if it is still under attack. Instead, focus on stopping the attack.

This chapter describes recovery recommendations for several types of attacks. In many cases, the recovery process is slow, and it might be several days before Active Directory functions normally again. If necessary, make modifications to these recommendations based on the needs of your environment. However, use caution when doing so; relaxing these security recommendations might leave your network open to attack again.

This chapter provides guidance for recovering from Active Directory attacks in the following situations:

- Physical breach of a domain controller
- Rogue administrator attack
- Catastrophic forest-wide corruption
- Data tampering attack
- Rogue object flood attack
- Rogue object growth attack

Recovering from the Physical Breach of a Domain Controller

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

If an unauthorized person gains access to the Active Directory data stored on a domain controller, that domain controller is considered to be *physically breached*. A domain controller is physically breached when an unauthorized person has been able to:

- Gain physical access to a domain controller.
- Steal, copy, or access Active Directory database files.
- Access the backup media for a domain controller.

When a domain controller has been physically breached, assume that all information that is stored on the domain controller, including the Active Directory database, is now available to the attacker. In such a situation, your organization should implement a plan for minimizing the security damage caused by the breach.

Recover from the physical breach of a domain controller by performing the following tasks:

1. Create a new backup of Active Directory.
2. Remove the account for the breached domain controller from Active Directory.
3. Reset all service administrator account passwords.
4. Reset the KRBTGT password twice.
5. Change all user account passwords.
6. Review memberships in all service administrator groups.
7. Review installed software on all domain controllers and service administrator workstations.
8. Review all group policy settings and logon scripts.
9. Find and remove rogue user accounts.
10. Create new backups.

Create a new backup of Active Directory

During the recovery process, you will be making many changes to Active Directory. Create a backup of Active Directory to ensure that, if necessary, you have a recent backup from which to restore.

Removing the Account for the Breached Domain Controller from Active Directory

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

As long as the computer account for the breached domain controller remains in Active Directory, that computer can continue to function as a domain controller on your network and it can continue to perpetrate changes to Active Directory. Remove the computer account for the breached domain controller from Active Directory as soon as possible after the breach has been discovered.

Force the immediate removal of the breached domain controller using the NTDSutil.exe tool, which is in Support Tools on your Windows 2000 CD. For information on how to remove a domain controller from Active Directory, see “Removing an Active Directory Domain Controller” in the Appendix B.

If you are responding to a rogue service administrator attack, you must remove the rogue service administrator account from Active Directory. To remove the account, simply find the account in Active Directory Users and Computers, right-click the account, and click Delete.

Resetting All Service Administrator Account Passwords

Service administrator accounts are highly privileged, and therefore they should be dealt with more urgently than other user accounts. After an Active Directory attack, reset all service administrator account passwords immediately. This minimizes the chance that an attacker will have sufficient time to crack one of these passwords and use it to log on to your domain.

You must reset each password individually. For information on how to reset service administrator passwords, see “Resetting Passwords” in Appendix B.

For information about which groups qualify as service administrator accounts, see Securing Service Administrator Accounts in Part 1, Chapter 5, of this guide.

Rendering Current Ticket-Granting Tickets invalid

You must reset the Key Distribution Service Account (KRBTGT) password *twice* to invalidate ticket-granting tickets (TGT) issued since the attack. The password must be reset twice to effectively invalidate TGTs.

With Kerberos, users are granted a TGT by KRBTGT upon authentication. Users present this TGT to gain access to network resources for the lifetime of the ticket, 10 hours by default. The KRBTGT encrypts TGTs with its password.

If an attacker compromises a password for a user account, logs on to the domain, and receives a TGT, the attacker is able to access network resources with that TGT for the lifetime of the ticket. This occurs even if the user’s password is changed after the attacker received the TGT.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

However, if you reset the KRBTGT password, the TGT is invalidated because domain controllers are not able to decrypt it. You must reset the password twice because domain controllers attempt to decrypt the TGT with passwords stored in history, two by default.

Important: You must be running Windows 2000 SP2 or later to perform this procedure. If you are not, replication issues can result. If replication is interrupted, use the Net Stop tool to stop replication on the domain controller and the repadmin tool to start replication. A complete discussion of these tools is outside of the scope of this guide. These tools are documented in the online help for Windows 2000.

To reset the KRBTGT account password, open Active Directory Users and Computers, and find the krbtgt user account. Right-click the account, and then click **Reset password**.

Changing All User Account Passwords

One threat that is associated with a breached domain controller is an impersonation attack. If the attacker has offline access to data in Active Directory, the attacker can attempt to compromise user account passwords with a password-cracking tool. If passwords are compromised, the attacker can impersonate an authenticated user. The attacker can log on to the domain and perform privileged tasks if a service administrator password is compromised.

To prevent the attacker from using compromised passwords, render these passwords useless by forcing the expiration of all user account passwords in the domain that contains the breached domain controller. When you force the expiration of account passwords, users must change their passwords the next time they log on.

If a user is logged on when passwords expire, the password will have to be changed the next time a logon is attempted. If the attacker manages to crack the user's password before the password is changed and logs on as that user, the attacker might be able to change the password. If this occurs, the user receives a notification to lock and unlock the computer to verify credentials. Because the password provided by the user does not match the new password set by the attacker, the user cannot unlock the computer. This results in a support call, at which point the user account should be disabled and the password reset.

If the breached domain is trusted by another domain, and if any of the users in the breached domain have administrator permissions in the trusting domain, you must expire all passwords in the trusting domain as well. If the attacker cracks a user account password that is also an administrator in another domain, the attacker has effectively compromised the other domain as well.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Note: For best security, reset all service administrator account passwords and expire all user account passwords, including service accounts. However, it is less essential to do so if one or both of the following situations describe your environment:

- If you enabled SYSKEY so that you must provide a password or insert a floppy disk containing the key on machine reboots on the breached domain controller, and the password or floppy disk are not available to the attacker.
- If your organization uses two-factor authentication for all users, such as smart cards or biometrics devices. In this case, you must still change all service account passwords.

Securely and efficiently change all user account passwords by performing the following tasks:

1. Prepare the PDC emulator for globally resetting passwords.
2. Force the expiration of user account passwords in batches.

Preparing the PDC Emulator for Globally Resetting Passwords

Before you force the expiration of all user account passwords, you must adequately prepare your network to handle this volume of password changes gracefully. The Windows 2000 domain controller that holds the primary domain controller operations master role (PDC emulator) can become overloaded and experience performance degradation in the following situations.

- When a large number of users change their account passwords in a short period of time.
- When users attempt to log on to a computer before the password change replicates to all domain controllers.
- If NTLM is used for authentication, when users attempt to access network resources before the new password replicates to all domain controllers in the domain.

If the PDC emulator becomes overloaded, users might be denied access to computers and resources. For more information on why the PDC emulator can become overloaded in these situations, see “Appendix A: Overloading the PDC Emulator” in the Appendix A

Prepare your Active Directory infrastructure to reduce the load on the PDC emulator by performing the following tasks:

1. Direct general client requests away from the PDC emulator.
2. Isolate the PDC emulator in its own site.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

3. Disable password change forwarding to the PDC emulator.
4. Reduce the notification delay interval for replication.

Directing General Client Requests Away from the PDC Emulator

As a general best practice, ensure that the PDC emulator is available to perform tasks that only it can perform by directing general client request traffic away from it. This can be accomplished by assigning appropriate priority and weight values to the service (SRV) resource records for the PDC emulator in the corresponding DNS database. This helps to ensure that tasks that can be performed by other domain controllers are not sent to the PDC emulator.

Assign priority and weight values to the PDC emulator so that its priority is higher and its weight is lower than other domain controllers in the domain. The DC locator uses these values to determine which domain controller is most appropriate for servicing a client. A domain controller with a higher priority is less likely to be contacted. If priorities are equal between domain controllers, the one with lower weight is less likely to be contacted.

Important: For these settings to take effect, you must stop and restart the Netlogon service on the PDC emulator. To stop the Netlogon service, at the command line type “net stop netlogon”, To restart the Netlogon service, at the command line, type “net start netlogon”.

For specific instructions on how to configure the priority and weight for domain controllers, see “Changing the Priority for DNS SRV Records” and “Changing the Weight for DNS SRV Records” in Appendix B of this guide.

Isolating the PDC Emulator in its Own Site

Another criterion that is used by the DC locator to preferentially select a domain controller to fulfill a client request is the site to which the domain controller belongs. The DC locator always prefers a domain controller that resides in the same site as the client. You can decrease the number of requests that are received by the PDC emulator by isolating the PDC emulator in its own site in which there are no clients. This site is created solely for the purpose of isolating the PDC emulator and not for designating a geographic separation.

Move the PDC emulator to the new site by performing the following tasks:

1. Create a new site.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

For this procedure, see “Create a Site Object” in the Active Directory Operations Guide, Appendix B — Procedures Reference at:

<http://www.microsoft.com/technet/prodtechnol/ad/Windows2000/maintain/opsguide/Part2/ADOGdApB.asp>

2. Create a new subnet for the site.

For this procedure, see “Create a Subnet Object” in the Active Directory Operations Guide, Appendix B — Procedures Reference at:

<http://www.microsoft.com/technet/prodtechnol/ad/Windows2000/maintain/opsguide/Part2/ADOGdApB.asp>

3. Move the PDC emulator to the new site by changing its IP address to match a subnet in the new site.

For this procedure, see “Change the Static IP Address of a Domain Controller” in Active Directory Operations Guide, Appendix B: Procedures Reference at:

<http://www.microsoft.com/technet/prodtechnol/ad/Windows2000/maintain/opsguide/Part2/ADOGdApB.asp>

Disabling Password Change Forwarding to the PDC Emulator

After the PDC emulator is moved to its own site, you can further decrease the load on it by disabling password change forwarding to the PDC emulator. Password change forwarding is enabled by default on all domain controllers. When it is enabled, the PDC emulator, regardless of the site where it resides, is notified immediately of password changes and updated with the new password. When many users change their passwords at once, the PDC emulator receives many notifications, which generates a significant load on the PDC emulator, causing its performance to degrade.

Disabling password forwarding prevents the immediate notification of password changes to the PDC emulator only if the PDC emulator is in a separate site. However, the PDC emulator still receives the new password through normal replication during the next scheduled replication period.

Note: Disabling password forwarding only takes effect on domain controllers running Windows 2000 SP2 and later.

Password change forwarding should be disabled on *every* domain controller in the domain. To disable password change forwarding on a domain controller, modify the entry under the following registry key:

HKLM\SYSTEM\CurrentControlSet\Services\Netlogon\Parameters

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Entry name: AvoidPdcOnWan

Data type: REG_DWORD

Value: 0 (disabled)

The default value for this entry is 1 (enabled). A value of 0, or no value, disables password change forwarding.

You can create a script that modifies this value on all domain controllers in the domain automatically by performing the following tasks:

1. Create the ComputerSearch.vbs script and copy it to a domain controller in the affected domain. For information about how to create the script see “Identifying Computers to Receive New Registry Settings” in the Appendix B.
2. Create a list of domain controllers in the domain by typing the following command at a command prompt:
3. Cscript computersearch.vbs /r:DC
4. Apply the new registry value to the listed domain controllers by creating and running the ApplyReg.vbs script.

Important: When password change forwarding is disabled, the PDC emulator no longer has the most up-to-date password information. This can result in users being denied access to network resources before the password change has replicated to all domain controllers. You can help mitigate this by reducing the notification delay intervals for Active Directory replication, as described in the next section.

Reducing the Notification Delay Interval for Active Directory Replication

To minimize the chance that users will be denied access to resources due to replication latency following a password change, reduce notification delay intervals for Active Directory replication. After you disable password change forwarding to the PDC emulator, the PDC emulator is no longer available to differentiate between a bad password and a password that has recently been changed but not replicated. Therefore, the chance that users are denied access to resources increases.

To minimize this possibility of users being denied access to resources, decrease the notification delay intervals for Active Directory replication. This expedites the replication of password changes to other domain controllers in the site.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

The notification delay intervals for Active Directory replication should be reduced on every domain controller in the domain. To decrease the notification delay interval, you must modify two registry entries. Both of these entries reside under the following registry key:

HKLM\SYSTEM\CurrentControlSet\Services\NTDS\Parameters

Entry name: ReplicatorNotifyPauseAfterModify

Data type: REG_DWORD

Value: 30 (secs)

This entry value determines the number of seconds that the domain controller on which the password change occurs waits before sending a notification to its first replication partner to initiate replication.

The default value for this setting is 300 seconds. It is recommended that you reduce this setting to 30 seconds. After all user passwords have been changed, restore the default setting.

Entry name: ReplicatorNotifyPauseBetweenDSAs

Data type: REG_DWORD

Value: 3 (secs)

This entry value determines the number of seconds that the domain controller on which the password change occurs waits between subsequent notifications to all other replication partners. The default value for this setting is 30 seconds. It is recommended that you reduce this setting to 3 seconds. After all user passwords have been changed, restore the default setting.

You can create a script that modifies these value on all domain controllers in the domain automatically by performing the following tasks:

1. Create the ComputerSearch.vbs script and copy it to a domain controller in the affected domain. For information about how to create the script see "Identifying Computers to Receive New Registry Settings with ComputerSearch.vbs" in Appendix B.
2. Create a list of domain controllers in the domain by typing the following command at a command prompt:
3. Cscript computersearch.vbs /r:DC
4. Apply the new registry value to the listed domain controllers by creating and running the ApplyReg.vbs script.

Reducing the Interval for Active Directory Replication Between Sites

To decrease the possibility that users are denied access to necessary resources in specific sites, you can reduce the interval for Active Directory replication between specific sites. The default replication

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

interval is 180 minutes. You can decrease this interval, particularly for the site that contains the PDC emulator, and accelerate password changes replicating between these sites.

Replication traffic uses network bandwidth. Do not reduce the interval for replication between sites to such a low value that bandwidth becomes saturated. The appropriate value is dependent on your environment.

After the password reset and change process is complete, restore these values to their original configuration.

To configure the replication between site interval, open Active Directory Sites and Services, right-click the site that you want to configure and click Properties. In Replicate every, enter the number of minutes you want to wait between replication.

Forcing the Expiration of User Account Passwords in Batches

If all user account passwords are changed at once, the available bandwidth for replication can become saturated. Forcing the expiration of user passwords in smaller batches decreases the impact on available replication bandwidth.

The size of the batch that you choose varies, depending on your environment. Start with batches that seem small and easily manageable for your environment; 3000 to 5000 user accounts might be a good starting point. If the available replication bandwidth is relatively unaffected by the batch size you select, increase the batch size during each iteration until you find the best size for your environment.

You can employ a variety of methods or criteria to help group your users into batches for the purpose of password change. The order in which you reset passwords should make sense for your organization. Create batches of user accounts based on, for example:

- Account location in a OU subtree.
- Range of alphabetized account names.
- Account membership in large groups or distribution lists.

Service account passwords also need to be expired and changed by the service administrator in charge of that service. Service accounts are found in Active Directory Users and Computers and might have been created by service administrators or by the service itself. Expire these passwords using the same method as expiring user account passwords. The appropriate service administrator

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

must change the password and go to each server that runs the service to manually update the password information.

The user interface for Active Directory allows only one password to expire at a time. To automate the process, use a script that enumerates user accounts and that forces the expiration of the password associated with each user account by setting the attribute **pwdLastSet** on each account to the value 0. This forces users to change their passwords the next time they log on.

For a sample script for forcing account password expiration, see “Forcing Password Expiration Using a Script” in Appendix B.

After you run the script for one batch, allow some time for users to log on and change their passwords and for that password information to replicate throughout the domain before resetting passwords for the next batch of users.

Keep in mind, however, that waiting to change a batch of passwords represents a security trade-off. The longer that account passwords remain unchanged, the longer the attacker has to compromise a password and infiltrate your organization’s network.

Reviewing the Memberships of All Service Administrator Groups

The service administrator groups are very privileged groups. To inflict the most damage, attackers usually try to gain access to an account, such as a service administrator account, that allows them to perform highly privileged tasks.

In addition to compromising passwords for existing service administrator accounts, attackers might also create user accounts and add them to the service administrator groups. If you do not find and remove these accounts, attackers will retain easy, privileged access to your network in the future. You must search for these groups manually, looking for user accounts that you do not recognize.

Review all users in service administrator groups and remove suspicious accounts from the service administrator group. If the account belongs to an authorized user, the user will make it known to your organization’s help desk.

For information about which groups qualify as service administrator accounts, see “Securing Service Administrator Accounts” in Part I, Chapter 5, of this guide.

Reviewing Installed Software on Domain Controllers and Administrative Workstations

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

In an ideal situation, you completely rebuild computers that you know have been compromised by reformatting the hard drive and reinstalling all software on the domain controller. This is the most secure reaction to attack, because reformatting the hard disk removes any back doors that attackers leave so that they can enter your network again.

In many cases, completely rebuilding the domain controller is not feasible. If your organization cannot completely rebuild the domain controller, plan to carefully review all software that has been installed on domain controllers and administrative workstations. Make sure that all settings are set appropriately for each application, and check for new applications that may have been added to the domain controller.

The intruder may have cracked some passwords, making it possible to log on to certain workstations and to install malicious software. The potential also exists that the attacker has already installed malicious software, such as a Trojan horse.

To respond to the threat of the installation of a Trojan horse:

- Examine services, applications, password filtering software, and notification packages.
- Examine all executables to ensure that these have not experienced tampering.
- Run a virus scan on all domain controllers and administrative workstations.

Reviewing All Group Policy Settings and Logon Scripts

If the attacker gains permissions and privileges on the level of service administrator, the attacker can alter any security safeguards that you have in place for your organization, including policy settings and logon scripts. After an attack, it is especially important that you check all policy settings and logon scripts to ensure that they have not been tampered with.

The most efficient way to check the validity of most policy settings is to run the Security Configuration and Analysis tool, comparing the template that you created (see corresponding section in Part 1) to the current settings for the domain. The tool reports any inconsistencies in this comparison. For information on how to use the Security Configuration and Analysis tool, see “Analyzing Current Security Settings” in Appendix B.

Note: If the template might have been modified by an attacker, review the settings in the template to ensure they are still accurate.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

If an attacker can access logon scripts and inject malicious code into these scripts, the malicious code will run on every client computer that authenticates on the network. After an attack is detected, logon scripts must be analyzed to ensure that they have not been modified.

Finding and Removing Rogue User Accounts

When an attacker is able to log on to a domain with a privileged account, the attacker might create one or more new, rogue user accounts. It is important to find this type of account or accounts. If you do not, the attacker can use this account to gain access to your network in the future and perpetrate other attacks.

The attacker might have added local accounts to administrative workstations or high security servers. These accounts can be instrumental for the attacker to perpetrate a Trojan horse attack on your network. Review the local administrator group and privileged accounts on administrative workstations and high security servers to ensure that the accounts are valid.

To find rogue accounts efficiently, you need to find an authoritative source of the users that exist on your network, such as a Human Resources database. Examine all user objects and verify that each object maps to a legitimate user. Disable all user accounts that do not have an associated legitimate user. If the user account is needed, a support call will result from the account being disabled. Investigate each call to ensure that the need for the account is valid. Delete all accounts that are not valid.

Creating New Backups

If you are unsure as to when a domain controller has been physically breached, you cannot be sure that any of your current backups are safe. Create a fresh backup as soon as the recovery is complete.

Recovering from a Rogue Administrator Attack

Service administrator accounts are the most highly privileged accounts in Active Directory. Service administrator accounts are considered to be *rogue* if one of the following has occurred:

- A service administrator account has been compromised.
- A user has elevated the privileges of a user account to the level of a service administrator.
- A trusted service administrator has become an attacker.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Rogue service administrators can perform virtually any malicious act. If you have detected and removed a rogue service administrator account from your network, the recovery procedure that you must perform is the same as the recovery for a physically breached domain controller, with one exception. Instead of removing the breached domain controller account from Active Directory, you must remove the rogue administrator account from Active Directory.

To recover from a rogue administrator attack, follow the recommendations described above in the “Recovering from the Physical Breach of a Domain Controller” earlier in this chapter.

For information about which groups qualify as service administrator accounts, see “Securing Service Administrator Accounts” in Part I, Chapter 5, of this guide.

Recovering from Catastrophic Forest-wide Corruption

If your Active Directory schema is corrupt or if irreparable changes are made to Active Directory data that is then replicated throughout your forest, you may have to recover the forest. In this process, you restore one domain controller for each domain from the last known good backup. All other domain controllers are restored by running the Active Directory Installation Wizard. Because the process is quite extensive and because data is usually lost, only recover the forest as a last resort when all other Active Directory restorative procedures have failed.

There are three stages to recovering your forest: pre-recovery, recovery, and post-recovery. To complete the forest recovery, it is recommended that you perform the tasks detailed in the following sections.

A complete description of each task is outside the scope of this guide. The “Best Practices: Active Directory Forest Recovery” paper contains a detailed explanation of and instructions for how to perform each task. For detailed information on performing each task, see Best Practices: Active Directory Forest Recovery at <http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=3EDA5A79-C99B-4DF9-823C-933FEBA08CFE>.

Performing Tasks in Preparation for a Forest Recovery

Before you recover your forest, perform the following tasks:

1. To ensure that a forest recovery is necessary, check with Microsoft Product Support Services.
2. Document your current forest structure by:

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

- a. Making a list of all domain controllers.
 - b. Noting which domain controllers have a backup that was made before the corruption, if you know when the corruption occurred.
3. Identify the forest root domain (the first domain in a new forest).

The first domain controller that you restore will come from this domain.
 4. For every other domain, identify the domain controller with the most recent valid backup.

These domain controllers will be the recovery domain controllers for all other domains.
 5. Shut down all domain controllers in the forest.

Performing Forest Recovery Tasks

Until you are directed to return the domain controllers to the production network, recover your forest while the domain controllers are physically isolated in a test environment.

Recover the first domain controller in the forest by performing the following tasks:

1. Restore a domain controller from the forest root domain from the last known good backup.

For details of restoring a domain controller, see “Restore from Backup Media for Authoritative Restore” in the Active Directory Operations Guide, Appendix B - Procedures Reference at <http://www.microsoft.com/technet>

[/prodtechnol/ad/Windows2000/maintain/opsguide/Part2/ADOGdApB.asp](http://prodtechnol/ad/Windows2000/maintain/opsguide/Part2/ADOGdApB.asp).

Important: It is essential that you begin the recovery with a domain controller in the forest root domain. Then repeat steps 1 through 12 for one domain controller in each additional domain in the forest.

2. Verify that the data on the domain controller has not been affected by the failure.

If the Active Directory data is damaged, repeat step 1 using a different backup.
3. Mark this SYSVOL as “primary,” because this is the first domain controller in the domain.
4. Ensure that the local DNS Server service is installed and running on the domain controller.
5. If the restored domain controller is enabled as a global catalog, disable the global catalog flag.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

6. Raise the value of the current RID pool by 100,000.
7. Seize all domain-wide and forest-wide operation master roles (also known as *flexible single master operations* or *FSMO*).

Note: You must seize these roles because, at this point, this is the only functional domain controller in the forest. When you repeat these steps for other domains, seize only the domain-wide FSMO roles.

8. Clean up the metadata for all other domain controllers in the domain.
9. Delete all server and computer objects in the domain except for this domain controller.
10. Reset the computer account password of this domain controller twice.
11. Reset the **krbtgt** account password twice.

For a discussion of the **krbtgt** account, see “Rendering Current Ticket-Granting Tickets invalid” earlier in this chapter.

12. Reset the trust password (Trusted Domain Object password) twice for all trusted domains.
13. Repeat steps 1 through 12 on a single domain controller from each domain in the forest before continuing with the forest recovery.
14. Finally, return the new domain controllers to the network, and rebuild the remaining domain controllers by performing the following tasks:
 - a. Join the recovered domain controllers to your network.
 - b. Ensure that the global catalog flag is enabled for the domain controller in the forest root domain.
 - c. Rebuild all other domain controllers in the forest by running the Active Directory Installation Wizard.

Performing Tasks Required to Complete the Forest Recovery

After your forest is functional again, complete the recovery process by performing the following tasks:

1. Delete any DNS records for domain controllers that have not been recovered.
2. Delete any WINS records for domain controllers that have not been recovered.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

3. Restore the operation master roles to the domain controllers that owned those roles in the pre-failure configuration.
4. Enable the global catalog flag on domain controllers that were global catalog servers in the pre-failure configuration.
5. Restore or reinstall any software applications that were running on domain controllers before recovery.

A detailed explanation of and instructions for how to perform each task is beyond the scope of this guide. For detailed information on performing each task, see Best Practices: Active Directory Forest Recovery at <http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=3EDA5A79-C99B-4DF9-823C-933FEBA08CFE>.

Recovering from Data Tampering by Restoring Active Directory Data

Tampering with or destroying certain Active Directory data types might not cause Active Directory to fail, but normal directory functioning might be disrupted. Examples of this type of tampering include modifying group memberships, altering SACLs on OUs, and removing topology elements of the Active Directory infrastructure.

Tampering of this type can be easily detected because some clients are unable to access network resources. It can be difficult to reconstruct missing Active Directory data if your organization does not maintain records of the exact configuration and security policies for Active Directory objects.

In some cases, you can recover to a pre-attack state by simply reapplying security templates, group policy settings, or registry settings. To restore subtree and leaf data, authoritatively restore only the affected objects from the last known good backup. In these situations, you can then update all other domain controllers with this new information.

Performing an Authoritative Restore of Directory Objects

Until directed to return the domain controller to the production network, perform these tasks in a physically isolated test environment.

Restore subtree and leaf data in the affected domain by performing the following tasks:

1. Identify a domain controller that contains tampered data and that has a last known good backup.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

2. Disconnect the domain controller from the production network.
3. Boot the domain controller into DSRestore Mode, and perform a normal restore from backup.

For details, see “Restore from Backup” in the Active Directory Operations Guide, Appendix B — Procedures Reference, at:

<http://www.microsoft.com/technet/prodtechnol/ad/Windows2000/maintain/opsguide/Part2/ADOGdApB.asp>

4. Restart the domain controller.
5. Verify that the tampered data is present, but that it is restored to its correct data values.

If the Active Directory data is still missing or incorrect, repeat steps 1 through 3, using an earlier backup.

6. Boot the domain controller into DSRestore Mode, and perform an authoritative restore of the tampered data.

For details, see “Restoring Active Directory subtree and Leaf Data from Backup” in the Active Directory Operations Guide, Appendix B — Procedures Reference, at:

<http://www.microsoft.com/technet/prodtechnol/ad/Windows2000/maintain/opsguide/Part2/ADOGdApB.asp>

7. Connect the domain controller to the production network.
8. Restart the domain controller.
9. Verify that the tampered data is restored on this domain controller.
10. Verify that the restored version of the tampered data replicates to all other domain controllers.

Detailed instructions for performing an authoritative restore of tampered objects is beyond the scope of this guide. For background information and detailed instructions for performing these tasks, see “Best Practices: Active Directory Forest Recovery” at: <http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=3EDA5A79-C99B-4DF9-823C-933FEBA08CFE>

Recovering from a Rogue Object Flood Attack

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Domain controllers are vulnerable to an attack that adds large numbers of rogue objects to Active Directory. Such an attack can cause a domain controller to run out of disk space on the database's drive potentially causing other legitimate object additions, modifications, or deletions to fail.

After an attacker has been blocked from accessing the network, to recover disk space on domain controllers, perform the following tasks:

1. Delete the reserve file on all affected domain controllers.
2. Create a backup of the recovery domain controller.
3. Find and delete the rogue objects.
4. Accelerate the purging of deleted objects (optional).
5. Verify that all deleted objects have been purged.
6. Create a fresh backup of a clean domain controller.

Deleting the Reserve File

If you previously added a reserve file to the Active Directory database drive, you can recover disk space by deleting the reserve file on all affected domain controllers. Freeing up some disk space helps your network to function during the recovery process. For information about the reserve file, see "Creating a Reserve File to Enable Recovery from Disk-Space Attacks" in Part I, Chapter 3, of this guide.

If a reserve file does not exist, or if queued objects fill the disk space originally set aside through the reserve file (after the file has been deleted), you must ascertain how you can recover disk space while the recovery is taking place. One possibility is moving the database files on all affected domain controllers to larger, dedicated drives.

Creating a Backup of the Recovery Domain Controller

You perform the recovery on one domain controller. Choose a domain controller on which to perform the recovery, and create a fresh backup.

It is possible for an attack to occur slowly over time, as opposed to all objects being added to the directory at once. During the period of time in which rogue objects are added to Active Directory, legitimate objects may also be created. In the process of purging rogue objects, these legitimate

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

objects might also be purged. If this occurs, it is easier to restore these objects from backup than it is to recreate the objects from scratch.

Create a backup for the recovery domain controller before you proceed with the recovery, so that you have a record of all objects.

Finding and Deleting Rogue Objects

Before you can delete all rogue objects, you must find them. If you already have an idea as to which objects are rogue objects, use those leads to determine which objects can be deleted. The steps detailed below should only be used if you have no idea as to when the attack occurred; these steps can be time consuming.

To find all the rogue objects that were added to Active Directory, you must determine when the attack began and which object is the first rogue object. Then, you have to examine all objects that were created since the first rogue object was created, looking for other rogue objects. If you do not know which object is the first rogue object, you have to search Active Directory for that object.

If you have been collecting periodic object counts, as recommended in “Monitoring for Disk Space Consumed by Active Directory Objects” in Chapter 2 of this guide, you can determine when the attack took place by performing the following tasks:

1. Disconnect an affected domain controller from the network.
2. Create the ObjCountbyClass.vbs and ObCMAudit.vbs scripts on a workstation and copy them to the affected domain controller. These scripts are required for rogue object detection.

For information about how to create ObjCountbyClass.vbs script, see “Monitoring the Number of Objects In a Domain With ObjCountByClass.vbs” in Appendix B of this guide. For information about how to create ObCMAudit.vbs script, see “Identifying Objects Created or Modified Within A Period of Time With ObCMAudit.vbs” in the Appendix B of this guide.

3. Identify the current number of objects by object class by running the following command:
4. `cscript ObjCountbyClass.vbs`
ObjCountbyClass.vbs creates an output file, ObjCountbyClass -*date-time*.csv, which contains the number of objects by object class.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

5. Examine the output file from ObjCountbyClass.vbs and compare this output to previous ObjCountbyClass.vbs outputs. By doing so, you can determine when objects were added and what types of objects experienced a trend in growth.

The output of ObjCountbyClass *-date-time.csv*, is a .CSV file, so you can view the output with an application such as Microsoft Excel.

6. Identify a list of the objects created or modified during the timeframe determined in step 5 by running the following command:
7. `cscript ObCMAudit.vbs [/d:date] [/t:time] /h:length`

Where:

- *date* is the date when you detect the disk space attack. This parameter is optional.
- *time* is the time when you detect the disk space attack. This parameter is optional.
- *length* is the length of time, in hours, before the time you detected the disk space attack based on the trend in the growth of the rogue objects discovered in step 5.

For example, if you detected a disk space attack at 3:00pm on April 1, 2003, and you thought the disk space attack took place over the last 20 hours, you would run the following command:

```
Cscript ObCMAudit.vbs /d:04-01-2003 /t:1500 /h:20
```

ObCMAudit.vbs creates an output file, ObCMAudit-*date-time.csv*, that contains a list of the objects created or modified during the length of time.

8. Examine the list and look for rogue a large number of objects of the type identified in Step 5.

The output of ObCMAudit.vbs, the Creating a List of Objects' Size script is a .CSV file, so you can export itview the output with an application such as Microsoft Excel to a spreadsheet and examine each object's size. Look for a large number of objects of the same object type that has experienced a trend in growth during the period of time discovered in Step 4.

9. Based on the analysis of the list in Step 5, delete each object that seems to be a rogue object.

If you have not been collecting periodic object counts, you can still find the first rogue object by performing the following tasks.

1. Determine the usnCreated of the object most recently added to Active Directory.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Note: A general guideline for how to find the first rogue object follows, assuming that you have no indication as to when the attack began. Your situation might demand some modifications.

2. Using the LDP.exe tool, perform the following search.
3. BaseDN: Root domain
4. Scope: subtree
5. Filter: (usnCreated>=1000)
6. Attributes: usnCreated
7. Sorted: descending order of usnCreated
8. Paged: yes

The first item in the output contains the usnCreated value of the object most recently added to Active Directory. Let the usnCreated value for that object be represented by *maxUsnCreated* for the remainder of this procedure.

9. Examine recently created objects, and try to discern which is the first rogue object:
 - a. Divide *maxUsnCreated* by 2.

The resulting value is *SearchValue*. Search Active Directory for all objects that have a usnCreated equal to or greater than *SearchValue* with the following query.

BaseDN: Root domain

Scope: subtree

Filter: (usnCreated>=SearchValue)

Attributes: usnCreated, DN, objectCategory

Sorted: descending order of usnCreated

Paged: yes

- b. This search generates a list of objects with a usnCreated value greater than *SearchValue*. Examine each of these objects, looking for rogue objects.
- c. If rogue objects are not found, continue adjusting the value of *SearchValue* by dividing by two and examining the objects until you find rogue objects.
- d. The first rogue object is the rogue object with the lowest usnCreated value. After you think you have found the first rogue object, examine many objects created before this object to ensure that it is the first rogue object.

10. Discern which objects created after the first rogue object are rogue objects and which are legitimate.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

11. Determine the usnCreated of the first rogue object (*RogueObjectusnCreated*), and then create a list of objects created since the attack by performing the following search:
 12. BaseDN: Root domain
 13. Scope: subtree
 14. Filter: (usnCreated>=RogueObjectusnCreated)
 15. Attributes: DN
 16. Sorted: descending order of usnCreated
 17. Paged: yes
18. Sort the results of this search so that child objects are listed before parents.

With the list sorted like this, you can be sure that you delete child objects before parent objects. A parent object cannot be deleted unless all its child objects are deleted first.

19. Examine each object that is returned in this search, looking for patterns in rogue objects. Some patterns to look for include the following:

- Objects that are created with the same parent object.
- Objects that have similar attributes.

20. Write a script that automatically deletes objects that follow the pattern for rogue objects.

If no pattern is discernable, you may have to delete all rogue objects manually.

Accelerating the Purging of Deleted Objects (Optional)

After the objects are deleted from Active Directory, they still take up disk space until their tombstone lifetime has expired and the garbage collection interval has passed. The tombstone lifetime determines how long deleted objects are kept on disk until they are collected and completely purged from the disk. The garbage collection interval determines how long the computer waits before deleted objects are collected and completely purged from the disk, after the tombstone lifetime has expired.

The default for this setting is 60 days; however, your organization may have previously modified this setting. If you cannot wait for the duration of the tombstone lifetime, you can decrease this value.

Note: If the reserve file frees up enough disk space for your network to function properly until the tombstone lifetime expires, skip this section.

The tombstone lifetime setting and the garbage collection interval are forest-wide settings. Do not modify them unless it is necessary to do so.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Reduce the time required to eliminate rogue objects by performing the following tasks:

1. Reduce the tombstone lifetime.

To ensure that all domain controllers are synchronized, you can reduce the tombstone lifetime to the recommended value of 14 days. However, if your environment demands a quicker recovery, adjust this value as necessary. Do not assign a tombstone lifetime value that is less than the time it takes for all domain controllers in your forest to synchronize. The minimum value allowed for the tombstone lifetime is two days.

Important: Do not decrease the value of the tombstone lifetime so that it is less than the amount of time that it takes for all domain controllers to synchronize. To be safe, wait for two times the maximum latency period. Otherwise, phantom objects will result and these objects will not be garbage collected.

2. Use the LDAP tool to make the modification described below:
3. Object: cn=Directory Service,cn=WindowsNT,cn=Service,cn=Configuration,dc=<DOMAIN>
4. ,Attribute:TombstoneLifetime Value: n days
5. Decrease the garbage collection interval to its minimum value of one hour.
6. As the tombstone lifetime expires for the deleted rogue objects, delete them permanently as quickly as possible to free up disk space.
7. Use the LDAP tool to make the modification described below:
8. Object: cn=Directory Service,cn=WindowsNT,cn=Service,cn=Configuration,dc=<DOMAIN> ,
9. Attribute:GarbageCollection Value: 1 hour

Verifying That All Deleted Objects Have Been Purged

Once the tombstone lifetime has expired and enough time has passed for the deleted objects to be garbage collected, check the event log for event 700. This event indicates that garbage collection is complete and online disk defragmentation has begun.

You can increase the number of events that are logged when garbage collection is running by modifying the entry under the following registry key:

HKLM\SYSTEM\CurrentControlSet\Services\NTDS\Diagnostics

Entry name: 6 Garbage Collection

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Data type: REG_DWORD

Value: 3

Changing the value of this registry entry allows event 1646 to be logged, which is logged after event 700. Event 1646 details how much disk space is freed during garbage collection, and it is logged when the garbage collection process is complete.

Creating a Fresh Backup of a Clean Domain Controller

Depending on how long it takes for you to realize that an attack has occurred, it is possible that all the backups that you have on hand contain rogue objects. Create a fresh backup as soon as all the rogue objects are deleted from Active Directory. You can choose any clean domain controller. Creating this backup minimizes the chances that, if you have to restore a domain controller from backup, you will also restore the rogue objects and have to perform this procedure again.

Recovering from an Object Growth Attack

Domain controllers are vulnerable to an attack that creates new, extraordinarily large objects or that modifies attributes of legitimate objects, resulting in these objects becoming extraordinarily large. After these objects are compromised and modified, they are considered to be rogue objects.

This type of attack can cause a domain controller to run out of disk space on the database's drive, potentially causing other legitimate object additions, modifications, or deletions to fail.

Recover from an object growth attack by performing the following tasks:

1. Delete the reserve file on all affected domain controllers.
2. Find and remove the rogue objects.
3. Create a fresh backup of a clean domain controller.

Deleting the Reserve File

If you previously added a reserve file to the Active Directory database drive, you can recover disk space by deleting the reserve file on all affected domain controllers. Freeing up some disk space helps your network to function during the recovery process. For information about the reserve file, see "Creating a Reserve File to Enable Recovery from Disk-Space Attacks" in Part 1, Chapter 3, of this guide.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

If a reserve file does not exist, you must ascertain how you can recover disk space while the recovery is taking place. One possibility is moving the database files on all affected domain controllers to larger, dedicated drives.

Finding and Removing the Rogue Objects

To recover disk space, you must find the rogue objects and remove them from the database. If the objects are created by the attacker and fulfill no useful purpose on your network, you can delete them from the database. However, if the rogue objects are legitimate and are modified, you can modify the objects so that the rogue object attributes are removed. When the objects are heavily modified and you are unable to modify the objects, delete and recreate the objects.

Remove rogue objects that have become extraordinarily large by performing the following tasks:

1. Disconnect an affected domain controller from the network.
2. Create the ObCMAudit.vbs and ObjMemUse.vbs scripts on a workstation and copy them to the affected domain controller. The scripts are required for rogue object detection.

For information about how to create ObCMAudit.vbs script, see “Identifying Objects Created or Modified Within A Period of Time With ObCMAudit.vbs” in Appendix B of this guide. For information about how to create ObjMemUse.vbs script, see “Identifying Large-sized Objects with ObjMemUse.vbs” in Appendix B of this guide.

3. Identify the potential rogue objects by running the following command:
4. `cscript ObCMAudit.vbs [/d:date] [/t:time] /h:length`

Where:

- *date* is the date when you detect the disk space attack. This parameter is optional.
- *time* is the time when you detect the disk space attack. This parameter is optional.

length is the length of time, in hours, before the time you detected the disk space attack.

Specify a length of time, in hours, that is equal to or greater than the last two intervals that you have for detecting disk space attack. Free disk space is monitored at specific intervals, for example, once every hour. You want to identify objects that have been created or modified during the last two intervals (two hours) to ensure that you capture a sufficient number of rogue objects.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

For example, if you detected a disk space attack at 3:00pm on April 1, 2003, and you monitor disk space utilization every hour, you would run the following command:

```
cscript ObCMAudit.vbs /d:04-01-2003 /t:1500 /h:2
```

ObCMAudit.vbs creates an output file, ObCMAudit-date-time.csv, that contains a list of the objects created or modified during the length of time.

5. Determine the size of the objects created or modified within a specific timeframe by running the following command:

6. `cscript ObjMemUse.vbs /f:input_file`

Where `input_file` is the name of the file created by ObCMAudit.vbs in step 3.

ObjMemUse.vbs creates an output file, ObjMemeUse-date-time.csv, that contains a list of the objects created in step 3, along with their size.

7. Examine the list, and look for rogue large objects.

The output of ObjMemeUse-date-time.csv, is a .CSV file; you can view the output with an application such as Microsoft Excel.

8. For each object that seems to be questionably large, determine if the object should remain in Active Directory and do one of the following:
 - a. If the object should not remain in Active Directory, delete it.
 - b. If the object should stay in Active Directory, determine what has been modified in the object that has made it very large, and modify the object so that it is an appropriate size. If the object has been heavily modified, it might be easiest to delete the object and recreate it.

Creating a Fresh Backup of a Clean Domain Controller

Depending on how long it takes for you to realize that an attack has occurred, it is possible that all the backups that you have on hand contain rogue objects. Create a fresh backup as soon as all the rogue objects are deleted from Active Directory. You can choose any clean domain controller. Creating this backup minimizes the chances that, if you have to restore a domain controller from backup, you will also restore the rogue objects and have to perform this procedure again.

Recommendations: Recovering from Active Directory Attacks

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

If Active Directory is attacked in one of the methods presented in this chapter, following the recommended recovery procedures helps to ensure that Active Directory is recovered securely. Each attack has its own set of recommendations.

Recovering from the Physical Breach of a Domain Controller

The following table provides a checklist of recommendations for recovering from a physically breached domain controller.

	Removing the Account for the Breached Domain Controller from Active Directory
<input type="checkbox"/>	Use the NTDSUtil.exe Support Tool to remove the breached domain controller from Active Directory.
	Resetting All Service Administrator Account Passwords
<input type="checkbox"/>	Use Active Directory Users and Computers to reset all service administrator passwords.
<input type="checkbox"/>	Distribute new passwords to service administrators.
	Rendering Current Ticket-Granting Tickets Invalid
<input type="checkbox"/>	Use Active Directory Users and Computers to reset the Key Distribution Service Account (krbtgt) password twice.
	Changing All User Account Passwords
<input type="checkbox"/>	Prepare for password expiration by: Isolating the PDC emulator in its own site. Reducing the notification delay interval for replication. Reducing the interval for Active Directory replication between sites.
<input type="checkbox"/>	Force the expiration of all user account passwords in batches.
	Reviewing the Memberships of All Service Administrator Groups
<input type="checkbox"/>	Review all users in service administrator groups and delete all users of whom you are suspicious.
	Examining Installed Software on Domain Controllers and Administrative Workstations
<input type="checkbox"/>	Review all software installed on domain controllers and administrative workstations looking

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

<input type="checkbox"/>	for Trojan horses and viruses.
<input type="checkbox"/>	Run an antivirus scan.
Reviewing All Group Policy Settings and Logon Scripts	
<input type="checkbox"/>	Run the Security Configuration and Analysis Tool comparing your security template to the current settings on the domain controller. Restore template settings wherever there is conflict.
<input type="checkbox"/>	Examine all logon scripts for evidence of tampering or code injection.
Finding and Removing Rogue User Accounts	
<input type="checkbox"/>	Review all group membership, looking for accounts that might have been created by the attacker.
Creating New Backups	
<input type="checkbox"/>	Create a new backup after the recovery is complete.

Recovering from a Rogue Administrator Attack

The following table provides a checklist of recommendations for recovering from a rogue service administrator attack.

Removing the Rogue Administrator Account from Active Directory	
<input type="checkbox"/>	Use Active Directory Users and Computers to delete the rogue administrator account.
Resetting All Service Administrator Account Passwords	
<input type="checkbox"/>	Use Active Directory Users and Computers to reset all service administrator passwords.
<input type="checkbox"/>	Distribute new passwords to service administrators.
Rendering Current Ticket-Granting Tickets Invalid	

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

<input type="checkbox"/>	Use Active Directory Users and Computers to reset the Key Distribution Service Account (krbtgt) password twice.
	Changing All User Account Passwords
<input type="checkbox"/>	Prepare for password expiration by: Directing general client requests away from the PDC emulator. Isolating the PDC emulator in its own site. Disabling password change forwarding to the PDC emulator. Reducing the notification delay interval for replication. Reducing the interval for Active Directory replication between sites.
<input type="checkbox"/>	Force the expiration of all user account passwords in batches by: Dividing users into small batches. Creating and running a script that expires all passwords for user accounts in that batch.
	Reviewing the Memberships of All Service Administrator Groups
<input type="checkbox"/>	Review all users in service administrator groups and delete all users of whom you are suspicious.
	Examining Installed Software on Domain Controllers and Administrative Workstations
<input type="checkbox"/>	Review all software installed on domain controllers and administrative workstations looking for Trojan horses and viruses.
<input type="checkbox"/>	Run an antivirus scan.
	Reviewing All Group Policy Settings and Logon Scripts
<input type="checkbox"/>	Run the Security Configuration and Analysis Tool comparing your security template to the current settings on the domain controller. Restore template settings wherever there is conflict.
<input type="checkbox"/>	Examine all logon scripts for evidence of tampering or code injection.
	Finding and Removing Rogue User Accounts
<input type="checkbox"/>	Review all group membership, looking for accounts that might have been created by the attacker.
	Creating New Backups
	Create a new backup after the recovery is complete.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

<input type="checkbox"/>	
--------------------------	--

The following table provides a checklist of recommendations for recovering from a catastrophic forest-wide corruption.

	Recovering from Catastrophic Forest-wide Corruption
<input type="checkbox"/>	Perform pre-recovery tasks.
<input type="checkbox"/>	Perform recovery tasks.
<input type="checkbox"/>	Perform post recovery tasks.

Recovering from Data Tampering by Restoring Active Directory Data

The following table provides a checklist of recommendations for recovering from data tampering.

	Performing an Authoritative Restore of Directory Objects
<input type="checkbox"/>	Choose an affected domain controller with a last known good backup and disconnect it from the network.
<input type="checkbox"/>	Perform a normal restore the tampered data.
<input type="checkbox"/>	Perform an authoritative restore of the data.
<input type="checkbox"/>	Verify that the data is restored and reconnect the domain controller to the network.
<input type="checkbox"/>	Verify that the data is restored on all other domain controllers.

Recovering from a Rogue Object Flood Attack

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

The following table provides a checklist of recommendations for recovering from a rogue object flood attack. A rogue object flood attack occurs when objects are added or modified so that Active Directory runs out of disk space.

	Deleting the Reserve File
<input type="checkbox"/>	Delete the reserve file on all affected domain controllers to free up disk space.
	Creating a Backup of the Recovery Domain Controller
<input type="checkbox"/>	Create a backup of the domain controller on which you will perform the recovery.
	Finding and Deleting Rogue Objects
<input type="checkbox"/>	Use the LDP.exe tool to search Active Directory for the first rogue object.
<input type="checkbox"/>	Delete rogue objects manually or with a script.
	Accelerating the Purging the Rogue Objects (Optional)
<input type="checkbox"/>	Decrease the tombstone lifetime to no less than twice the maximum replication latency for your network.
<input type="checkbox"/>	Decrease the garbage collection interval.
<input type="checkbox"/>	Wait for all deleted objects to be purged.
	Verifying That All Deleted Objects Have Been Purged
<input type="checkbox"/>	Look for event 700 in the event log indicating that garbage collection is complete and online defragmentation has been triggered.
	Creating a Fresh Backup of the Clean Domain Controller
<input type="checkbox"/>	Create a new backup after the recovery is complete.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Recovering from a Rogue Object Growth Attack

The following table provides a checklist of recommendations for recovering from a rogue object growth attack. A rogue object growth attack occurs when object attributes are added or modified, causing the object to grow to an extraordinarily large size, so that Active Directory runs out of disk space.

Deleting the Reserve File	
<input type="checkbox"/>	Delete the reserve file on all affected domain controllers to free up disk space.
Finding and Deleting Rogue Objects	
<input type="checkbox"/>	Disconnect an affected domain controller from the network.
<input type="checkbox"/>	Create the ObCMAudit.vbs and ObjMemUse.vbs scripts on a workstation and copy them to the affected domain controller.
<input type="checkbox"/>	Create a list of potential rogue objects by running the ObCMAudit.vbs script.
<input type="checkbox"/>	Determine the size of each object listed by running the ObjMemUse.vbs script.
<input type="checkbox"/>	Examine the object sizes, modifying or deleting rogue objects.
Creating a Fresh Backup of the Clean Domain Controller	
<input type="checkbox"/>	Create a new backup after the recovery is complete.

Appendix A: Overloading the PDC Emulator

The PDC emulator maintains the authoritative list of account passwords. In the default configuration, when a user changes the account password, the domain controller that processed the password change immediately forwards the new password to the PDC emulator.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

When you force passwords to expire in response to a domain controller breach, thousands of users might change their passwords at the same time. Resetting all these passwords at once would result in thousands of password updates being forwarded to the PDC emulator, potentially overloading the PDC emulator.

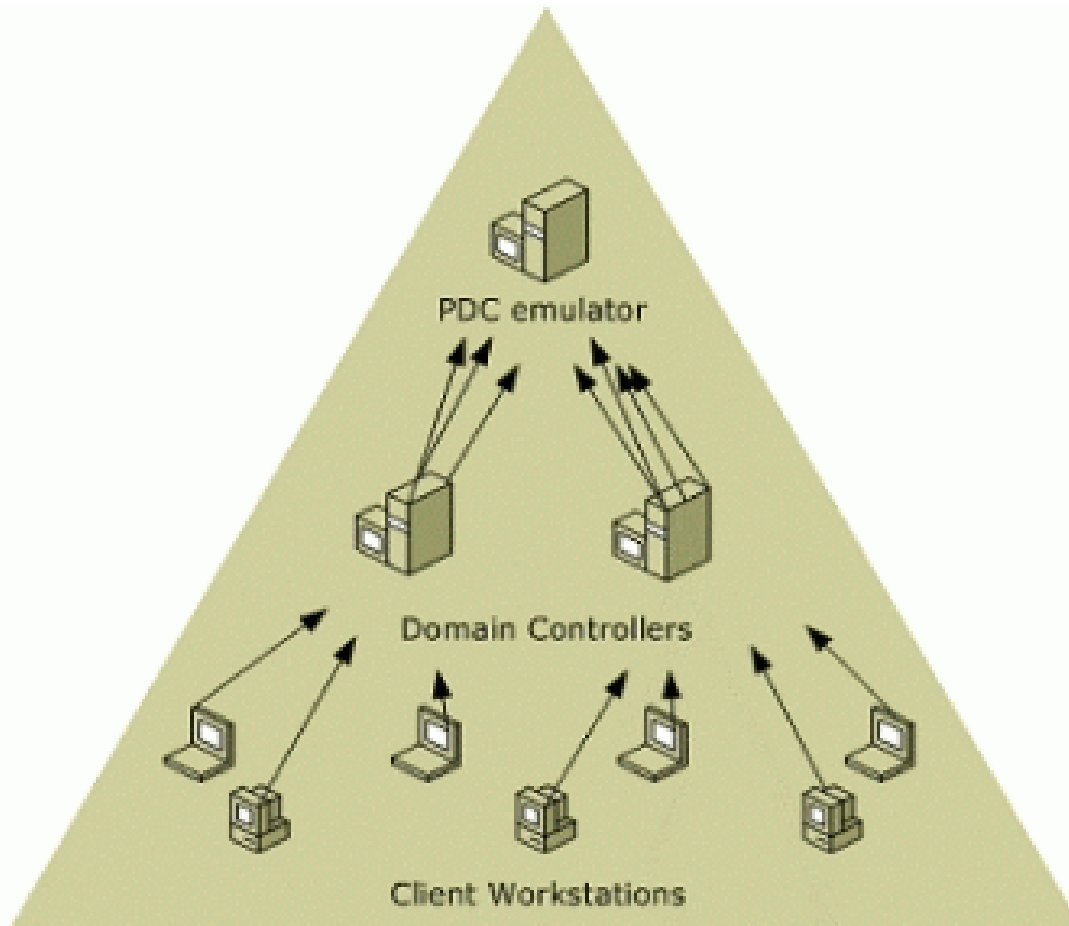


Figure 7: Overloading the PDC Emulator with Password Changes

All other domain controllers receive new password information through normal replication. As a result, after an initial password change, there is some latency before the other domain controllers receive new passwords. This behavior only presents a problem if NTLM is used for authentication. If a user attempts to log on to another computer or access a network resource with the new password during this window, the password information provided by the user does not match the password information stored on the authenticating domain controller. Realizing that its password information could be obsolete, the authenticating domain controller contacts the PDC emulator to attempt to verify the provided password. If thousands of users are all doing this simultaneously, the PDC emulator might

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

not be able to accommodate all of the requests. These requests might eventually time-out, resulting in the user being denied access to the computer or network resource.

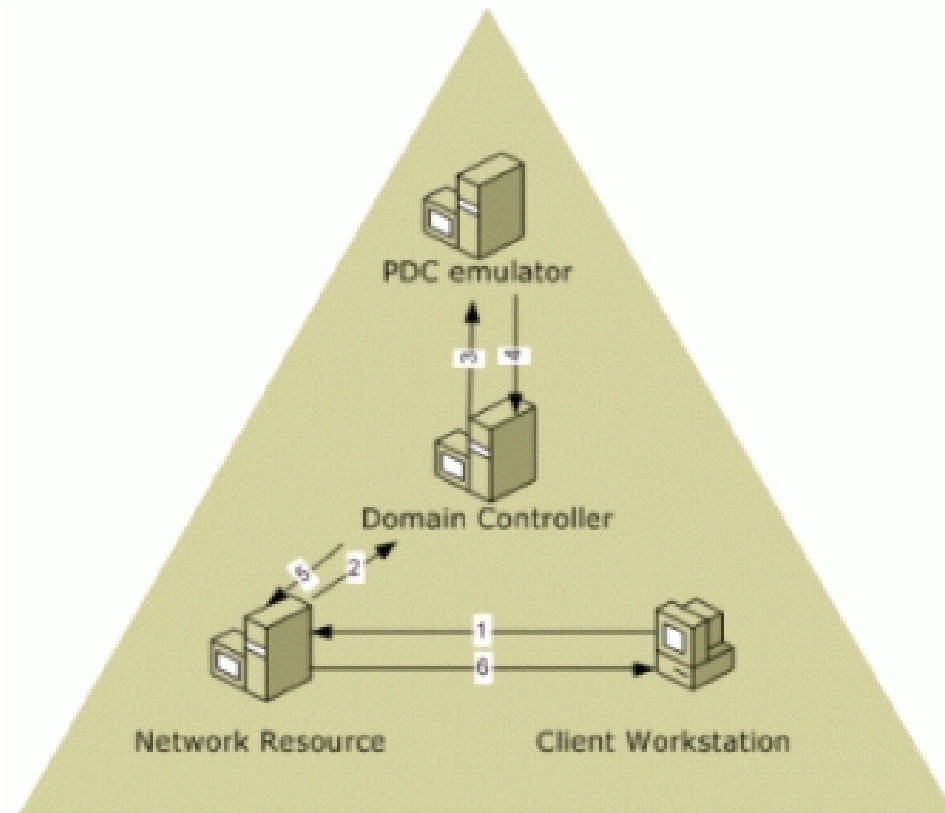


Figure 8: Accessing Network Resources Immediately After a Password Change

The steps involved in accessing a network resource after a password change, but before that password has replicated to all domain controllers in environments where NTLM is used are as follows. When a user attempts to access a network resource, the resource contacts a domain controller to authorize the access. If the domain controller cannot verify the client's identity (because the password has changed), then it contacts the PDC emulator to check for an updated password (3). PDC emulator verifies the client's identity and forwards this information to the domain controller (4) where the user's password is updated. The domain controller then contacts the network resource (5) and allows access if the password provided was correct and if the user has the permissions to access the resource. The network resource in turn forwards the requested information to the client (6).

If all users change their passwords at once and then attempt to log on or access a network resource, the PDC emulator can become overloaded.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

```
9.  '* Description: Utility that applies the contents of a .reg file to  *
10. '*           the registry of each available and accessible  *
11. '*           computer specified in the csv report created by the  *
12. '*           ComputerSearch.vbs script.  *
13. '*           This script generates a report that includes the  *
14. '*           name of each computer where the .reg file import  *
15. '*           was attempted and whether the application was  *
16. '*           successful at applying the change.  *
17. '*           *
18. '* Compatibility: This script requires WSH 5.6, CScript, ADSI, WMI  *
19. '*           and access to Active Directory  *
20. *****
21. Option Explicit
22. 'Define constants
23. Const ForReading = 1
24. Const TristateUseDefault = -2
25. Const ForAppending = 8
26. 'Declare global variables
27. Dim objArgs,strRptFileName,strRegFileName
28. Dim objDictionary,strFileName,objFSO,objFile,objRegFile
29. Dim objRptFile,objRegExp,objShell,iRecord,strLine
30. Dim arrComptInfo,strDN,strComputer,strNotes
31. Dim objExec,strPingStdOut,objRegistry
32. Dim objTextStream,ColRootKey,Key,strKey,strKeyPath
33. Dim strStatus,iCnt
34. Call CheckForCScript()
35. 'Use Named Arguments collection for the command line argument.
36. 'The WSHArguments Object is included in WSH version 5.6 and later
37. Set objArgs = WScript.Arguments.Named
38. strRptFileName = objArgs.Item("f")
39. strRegFileName = objArgs.Item("r")
40. If WScript.Arguments.Named.Count < 2 Then
41.     WScript.Echo "You must specify a csv file name " & VbCrLf & _
42.     "of a file created by ComputerSearch.vbs and " & VbCrLf & _
43.     "the name of the .reg file to apply." & VbCrLf & VbCrLf & _
44.     SampleCommandLine()
```

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

```
45. WScript.Quit
46. End If
47. 'Use a dictionary object for the constants that define
48. 'the root keys
49. Set objDictionary = CreateObject("Scripting.Dictionary")
50. objDictionary.Add "HKEY_LOCAL_MACHINE",&h80000002
51. objDictionary.Add "HKEY_CLASSES_ROOT",&h80000000
52. objDictionary.Add "HKEY_CURRENT_USER",&h80000001
53. objDictionary.Add "HKEY_USERS",&h80000003
54. objDictionary.Add "HKEY_CURRENT_CONFIG",&h80000005
55. 'Call the GenFileName function to create a unique file name for the report
56. strFileName = GenFileName("ApplyReg")
57. 'Create a text file (.csv format) to hold the
58. 'results of the report.
59. Set objFSO = CreateObject("Scripting.FileSystemObject")
60. 'This will overwrite the file if it already exists.
61. Set objFile = objFSO.CreateTextFile(strFileName,True)
62. objFile.Close
63. Set objFile = objFSO.OpenTextFile(strFileName,ForAppending)
64. 'Write the headings for this csv file
65. objFile.WriteLine "DistinguishedName," & _
66. "Computer Name,Status,Notes"
67. 'Get the registry file
68. Set objRegFile = objFSO.GetFile(strRegFileName)
69. 'Open the computer search report file for reading
70. Set objRptFile = objFSO.OpenTextFile(strRptFileName,ForReading)
71. 'Skip the header row of the report
72. objRptFile.SkipLine
73. 'Create the Regular Expression object
74. Set objRegExp = New RegExp
75. 'Set some global properties for the RegExp object
76. objRegExp.IgnoreCase = FALSE
77. objRegExp.Global = TRUE
78. objRegExp.MultiLine = FALSE
79. 'Create the Wscript.Shell object for accessing the Exec method
80. Set objShell = CreateObject("WScript.Shell")
```

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

```
81. iRecord = 1
82. WScript.Echo "Report records processed: "
83. Do While objRptFile.AtEndOfStream <> TRUE
84.   strLine = objRptFile.ReadLine
85.   'Bind to dn of a computer listed in the report file.
86.   arrComptInfo = Split(BindDN(strLine),"||")
87.   strDN = arrComptInfo(0)
88.   strComputer = arrComptInfo(1)
89.   'An invalid dn is specified in the report file
90.   If strComputer = "None" Then
91.     strNotes = arrComptInfo(2) & " distinguished name specified"
92.     strStatus = "Unable to connect to the specified dn."
93.   Else
94.     strNotes = arrComptInfo(2) & " name used for remote connection."
95.     'Before connecting to the computer, use ping to see if you get a response.
96.     'A WMI connection attempt is not used because WMI's connection timeout interval
97.     'is too long
98.     Set objExec = objShell.Exec("ping -n 2 -w 1000 " & strComputer)
99.     strPingStdOut = LCase(objExec.StdOut.ReadAll)
100.    'Test whether ping was successful
101.    If InStr(strPingStdOut, "reply from ") Then
102.      'Attempt to connect to the registry provider on a remote computer
103.      Set objRegistry=GetObject("winmgmts:{impersonationLevel=impersonate}!\" &_
104.        strComputer & "\root\default:StdRegProv")
105.      On Error Resume Next
106.      If Err.Number <> 0 Then
107.        'Store the following line to write to the status column for this computer.
108.        'Typical causes of failure:
109.        'WMI not running or operator does not have permission to make a remote connection.
110.        strStatus = Err.Description
111.        Err.Clear
112.      On Error GoTo 0
113.    Else
114.      'Perform the registry update
115.      Set objTextStream = objRegFile.OpenAsTextStream(ForReading, TristateUseDefault)
116.      Do While objTextStream.AtEndOfStream <> TRUE
```

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

```
117.      'Read a line of data
118.      strLine = objTextStream.ReadLine
119.      'Match root keys and paths
120.      objRegExp.Pattern = "(^[HKEY_\\w+[^\\]])(.*)"
121.      If objRegExp.Test(strLine) = TRUE Then 'Create the key
122.          Set ColRootKey = objRegExp.Execute(strLine)
123.          For Each Key in ColRootKey
124.              strKey = Mid(Key.SubMatches(0),2)
125.              strKeyPath = Mid(Key.SubMatches(1),2,Len(Trim(Key.SubMatches(1)))-2)
126.              objRegistry.CreateKey objDictionary(strKey),strKeyPath
127.          Next
128.      Else 'Create value names and values for the key
129.          'Match strings: "valuenam"="value"
130.          objRegExp.Pattern = "(\\x22.*\\x22=\\x22)(.*)"
131.          If objRegExp.Test(strLine) = TRUE Then
132.              Call CreateString(strLine)
133.          End If
134.          'Match Binary data: "valuenam"=hex:
135.          objRegExp.Pattern = "(\\x22.*\\x22=hex:)(.*)"
136.          If objRegExp.Test(strLine) = TRUE Then
137.              Call CreateBinary(strLine)
138.          End If
139.          'Match DWORD data: "valuenam"=dword:
140.          objRegExp.Pattern = "(\\x22.*\\x22=dword:)(.*)"
141.          If objRegExp.Test(strLine) = TRUE Then
142.              Call CreateDWORD(strLine)
143.          End If
144.          'Match Expandable string: "valuenam"=hex(2):
145.          objRegExp.Pattern = "(\\x22.*\\x22=hex\\(2\\):)(.*)"
146.          If objRegExp.Test(strLine) = TRUE Then
147.              Call CreateExpString(strLine)
148.          End If
149.          'Match Multistring: "valuenam"=hex(7):
150.          objRegExp.Pattern = "(\\x22.*\\x22=hex\\(7\\):)(.*)"
151.          If objRegExp.Test(strLine) = TRUE Then
152.              Call CreateMultiString(strLine)
```

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

```
153.     End If
154.     'The registry Key pattern is true. Therefore, the script has encountered
155.     'another registry key to apply.
156.     End If
157.     Loop
158.     strStatus = "Registry update applied."
159.     End If
160. Else
161.     'Store the following line to write to the status and notes columns for this computer
162.     strStatus = "Host unreachable"
163.     strNotes = "Verify that this host is online."
164.     End If
165. End If
166. objFile.WriteLine Chr(34) & strDN & Chr(34) & "," & strComputer & _
167.     "," & strStatus & "," & strNotes
168. 'Display a record counter
169. For iCnt = 1 to Len(iRecord)
170.     WScript.StdOut.Write Chr(8)
171. Next
172. WScript.StdOut.Write iRecord
173. iRecord = iRecord + 1
174. Loop
175. WScript.Echo VbCrLf & VbCrLf & "The report data has been saved to: " & _
176.     strfileName & "." & VbCrLf & _
177.     "Import or open the CSV data in a spreadsheet" & VbCrLf & _
178.     "or database program to determine which computers were updated."
179. *****
180. '* Routine: CreateString
181. *****
182. Sub CreateString(Line)
183.     Dim ColEntries,Entry,strValueName,strValue
184.     Set ColEntries = objRegExp.Execute(Line)
185.     For Each Entry in ColEntries
186.         strValueName = Mid(Entry.SubMatches(0),2,Len(Trim(Entry.Submatches(0)))-4)
187.         strValue = Mid(Entry.SubMatches(1),1,Len(Trim(Entry.Submatches(1)))-1)
188.         objRegistry.SetStringValue objDictionary(strKey),strKeyPath,strValueName,strValue
```

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

```
189. Next
190. End Sub
191. *****
192. '* Routine: CreateBinary
193. *****
194. Sub CreateBinary(Line)
195. Dim arrValue(),ColEntries,Entry,strValueName,arrHexValue,i,HexVal
196. Set ColEntries = objRegExp.Execute(Line)
197. For Each Entry in ColEntries
198.     strValueName = Mid(Entry.SubMatches(0),2,Len(Trim(Entry.Submatches(0)))-7)
199.     'Use the split function to create an array so that each hex value can be
200.     'appended with an &h value in the next For Each statement
201.     arrHexValue = Split(Trim(Entry.SubMatches(1)),",")
202.     'Declare a dynamic array to hold the properly formatted hex values to
203.     'be passed to the WMI SetBinaryValue method.
204.     i=0
205.     For Each HexVal in arrHexValue
206.         Redim Preserve arrValue(i)
207.         arrValue(i) = "&h" & HexVal
208.         i=i + 1
209.     Next
210.     objRegistry.SetBinaryValue objDictionary(strKey),strKeyPath,strValueName,arrValue
211.     Redim arrValue(0)
212. Next
213. End Sub
214. *****
215. '* Routine: CreateDWORD
216. *****
217. Sub CreateDWORD(Line)
218. Dim ColEntries,Entry,strValueName,intValue
219. Set ColEntries = objRegExp.Execute(Line)
220. For Each Entry in ColEntries
221.     strValueName = Mid(Entry.Submatches(0),2,Len(Trim(Entry.Submatches(0)))-9)
222.     'Convert the hex value that will be passed to the WMI
223.     'SetDWORDValue method, to a decimal data type
224.     intValue = CInt("&h" & Trim(Entry.SubMatches(1)))
```

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

```
225. objRegistry.SetDWORDValue objDictionary(strKey),strKeyPath,strValueName,intValue
226. Next
227. End Sub
228. *****
229. '* Routine: CreateExpString
230. *****
231. Sub CreateExpString(Line)
232. Dim arrValue(),ColEntries,Entry,strValueName,strValueFirstLine
233. Dim strValueMiddleLines,strValueLastLine,strExpandableString,HexVal
234. Dim strExpandableStringFormatted,arrlItems
235. Set ColEntries = objRegExp.Execute(Line)
236. For Each Entry in ColEntries
237.   strValueName = Mid(Entry.SubMatches(0),2,Len(Trim(Entry.SubMatches(0)))-10)
238.   strValueFirstLine = Trim(Entry.SubMatches(1))
239. Next
240. 'Read another line to test for data that might belong to an expandable string entry
241. strLine = objTextStream.ReadLine
242. 'Match additional lines of expandable-string data: nn,nn,nn...\
243. objRegExp.Pattern = "(^s{2}[0-9{1,2}|a-f{1,2},,]+\$$)"
244. Do While objRegExp.Test(strLine) = TRUE
245.   Set ColEntries = objRegExp.Execute(strLine)
246.   For Each Entry in ColEntries
247.     strValueMiddleLines = strValueMiddleLines & Trim(Entry.SubMatches(0))
248.   Next
249. 'Read another line to test for data that might belong to the middle lines of
250. 'an expandable string entry
251.   strLine = objTextStream.ReadLine
252. Loop
253. 'Match the last line of a expandable-string expression
254. objRegExp.Pattern = "(^s{2}[0-9{1,2},|a-f{1,2},,]+[0-9{1,2}$|a-f{1,2}$])"
255. If objRegExp.Test(strLine) Then
256.   Set ColEntries = objRegExp.Execute(strLine)
257.   For Each Entry in ColEntries
258.     strValueLastLine = Trim(Entry.SubMatches(0))
259.   Next
260. End If
```

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

```
261. 'Combine the expandable-string value
262. strExpandableString = strValueFirstLine & strValueMiddleLines & strValueLastLine
263. 'Remove the "\" character from strExpandableStringValue
264. objRegExp.Pattern = "\"
265. strExpandableStringFormatted = objRegExp.Replace(strExpandableString,"")
266. 'Convert to an array for formatting each value
267. arrItems = Split(strExpandableStringFormatted,",")
268. For Each HexVal in arrItems
269.   Redim Preserve arrValue(i)
270.   If HexVal <> "00" Then
271.     arrValue(i) = CStr(Chr(CInt("&h" & HexVal)))
272.     strData = strData & arrValue(i)
273.     i= i + 1
274.   End If
275. Next
276. 'Add the expandable string to the registry
277. objRegistry.SetExpandedStringValue
    objDictionary(strKey),strKeyPath,strValueName,strData
278. End Sub
279. *****
280. '* Routine: CreateMultiString
281. *****
282. Sub CreateMultiString(Line)
283.   Dim arrArgData(),ColEntries,Entry,strValueName,strValueFirstLine
284.   Dim strValueMiddleLines,strValueLastLine,strMultiString
285.   Dim strMultiStringFormatted,arrLine,iArgs
286.   Dim HexVal,arrItems,arrValue,strData
287.   Set ColEntries = objRegExp.Execute(Line)
288.   For Each Entry in ColEntries
289.     strValueName = Mid(Entry.SubMatches(0),2,Len(Trim(Entry.Submatches(0)))-10)
290.     strValueFirstLine = Trim(Entry.SubMatches(1))
291.   Next
292. 'Read another line to test for data that might belong to a multistring entry
293. strLine = objTextStream.ReadLine
294. 'Match additional lines of multi-string data:  nn,nn,nn...\
295. objRegExp.Pattern = "(^s{2}[0-9{1,2}]a-f{1,2},]+\\"$)"
```

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

```
296. Do While objRegExp.Test(strLine) = TRUE
297.   Set ColEntries = objRegExp.Execute(strLine)
298.   For Each Entry in ColEntries
299.     strValueMiddleLines = strValueMiddleLines & Trim(Entry.SubMatches(0))
300.   Next
301.   'Read another line to test for data that might belong to the middle
302.   'lines of a multistring entry
303.   strLine = objTextStream.ReadLine
304. Loop
305. 'Match the last line of a Multistring expression
306. objRegExp.Pattern = "(^s{2}[0-9{1,2},|a-f{1,2},]+[0-9{1,2}$|a-f{1,2}$])"
307. If objRegExp.Test(strLine) Then
308.   Set ColEntries = objRegExp.Execute(strLine)
309.   For Each Entry in ColEntries
310.     strValueLastLine = Trim(Entry.SubMatches(0))
311.   Next
312. End If
313. 'Combine the expandable-string value
314. strMultiString = strValueFirstLine & strValueMiddleLines & strValueLastLine
315. 'Remove the "\" character from strMultiStringValue
316. objRegExp.Pattern = "\"
317. strMultiStringFormatted = objRegExp.Replace(strMultiString, "")
318. 'Each line is delimited by "00,00,00". Create separate strings for each line.
319. 'Each line is an argument in the array supplied to the SetMultiStringValue method.
320. arrLine = Split(strMultiStringFormatted,"00,00,00")
321. iArgs = 0
322. 'Convert each item in each line to string data
323. For Each Line in arrLine
324.   If Line <> "" AND Trim(Line) <> "," Then
325.     'Remove commas padding the beginning and/or the end.
326.     If Instr(1,Line,",") = 1 Then
327.       Line = Mid(Line,2)
328.     End If
329.     If Instr(Len(Line),Line,",") = Len(Line) Then
330.       Line = Mid(Line,1,Len(Line) - 1)
331.     End If
```

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

```
332. 'Split each item in each line for conversion.
333. arrItems = Split(Line,",")
334. i=0
335. 'Convert each hex value to character data.
336. For Each HexVal in arrItems
337.     Redim Preserve arrValue(i)
338.     If HexVal <> "00" Then
339.         arrValue(i) = CStr(Chr(CInt("&h" & HexVal)))
340.         strData = strData & arrValue(i)
341.         i= i + 1
342.     End If
343. Next
344. Redim Preserve arrArgData(iArgs)
345. arrArgData(iArgs) = strData
346. strData = ""
347. iArgs = iArgs + 1
348. End If
349. Next
350. objRegistry.SetMultiStringValue
    objDictionary(strKey),strKeyPath,strValueName,arrArgData
351. Redim arrArgData(0)
352. End Sub
353. *****
354. '* Routine: CheckForCScript
355. *****
356. Sub CheckForCScript
357. 'This script must run from cscript because
358. 'it uses the WScript.StdOut property.
359. 'Test the script host and if it's not cscript,
360. 'instruct the operator on how to run the script.
361. If Right(LCase(WScript.FullName),11) <> LCase("cscript.exe") Then
362.     WScript.Echo "This script must run from cscript." & _
363.     VbCrLf & "Example: cscript ApplyReg.vbs /f:ComptSearch.csv /r:RegFile.reg"
364.     WScript.Quit
365. End If
366. End Sub
```

**Best Practice Guide for Securing Active Directory
Installations and Day to Day Operations Part II**
By Kathleen Cole, Jennifer Bayer, Doug Steen

```
367. *****
368. '* Function: PadZero
369. *****
370. Function PadZero(dtValue)
371.   If Len(dtValue) = 1 Then
372.     PadZero = 0 & dtValue
373.   Else
374.     PadZero = dtValue
375.   End If
376. End Function
377. *****
378. '* Function: GenFileName
379. *****
380. Function GenFileName(prefix)
381. 'Create a unique time stamped name for the text file
382. Dim dtDate,strYear,strMonth,strDay,strDate
383. Dim dtNow,strHour,strMinute,strSecond,strTime
384. dtDate = Date()
385. strYear = Mid(Year(dtDate),3)
386. strMonth = PadZero(Month(dtDate))
387. strDay = PadZero(Day(dtDate))
388. strDate = strYear & strMonth & strDay & "-"
389. dtNow = Now()
390. strHour = PadZero(Hour(dtNow))
391. strMinute = PadZero(Minute(dtNow))
392. strSecond = PadZero(Second(dtNow))
393. strTime = strHour & strMinute & strSecond
394. GenFileName = prefix & "-" & strDate & strTime & ".csv"
395. End Function
396. *****
397. '* Function: SampleCommandLine
398. *****
399. Function SampleCommandLine()
400. SampleCommandLine = _
401.   "For example, to apply registry changes specified in" & VbCrLf & _
402.   "a file named SysAdmin.reg, to all computers listed in" & VbCrLf & _
```

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

```
403. "ComptSearch.csv, type:" & VbCrLf & _
404. "CScript ApplyReg.vbs /r:SysAdmin.reg /f:ComptSearch.csv"
405. End Function
406. *****
407. '* Function: BindDN
408. *****
409. Function BindDN(Line)
410. Const E_ADS_PROPERTY_NOT_FOUND = &h8000500D
411. Const LDAP_NO_SUCH_OBJECT = &h80072030
412. Dim strDN,objComputer,strComptName
413. 'Use Instr to get the distinguishedName column for writing to the
414. 'first column of the report.
415. strDN = Mid(Line,2,Instr(1,Line,Chr(34) & Chr(44))-2)
416. 'Bind to the computer in AD
417. On Error Resume Next
418. Set objComputer = GetObject("LDAP://" & strDN)
419. If Err.Number = LDAP_NO_SUCH_OBJECT Then
420. 'The GetObject method was unable to bind to the specified
421. 'dn. This probably means the computer is not listed in AD.
422. BindDN = strDN & "||None||Invalid"
423. Else
424. 'Get the dNSHostName of the computer
425. strComptName = objComputer.Get("dNSHostName")
426. 'If the dNSHostName attribute is set, use it for the registry update.
427. If Err.number <> E_ADS_PROPERTY_NOT_FOUND Then
428. BindDN = strDN & "||" & strComptName & "||Host"
429. Err.Clear
430. 'If the dNSHostName attribute is not set then use
431. 'the cn for the registry update.
432. Else
433. strComptName = objComputer.Get("cn")
434. BindDN = strDN & "||" & strComptName & "||NetBIOS"
435. End If
436. End If
437. Err.Clear
438. On Error GoTo 0
```

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

439. End Function

440. Save the file as ApplyReg.vbs

Analyzing Current Security Settings

This tool creates a log file with the current security configuration of the computer analyzed.

Requirements

- Credentials: Domain Admins or Enterprise Admins

To analyze the current security settings:

1. Open **Security Configuration and Analysis** administrative tool.
2. In the console tree, right-click **Security Configuration and Analysis** and then click **Open Database**.

If you are creating a new database:

- a. For **File name**, type **FileName** and then click **Open**.
 - b. Select a template and then click **Open**.
3. If you are opening an existing database, select the database and then click **Open**.
 4. In the details pane, right-click **Security Configuration and Analysis**, and then click **Analyze Computer Now**.
 5. Do one of the following:
 - a. To use the default log, in **Error log file path**, click **OK**.
 - b. To specify a different log, in **Error log file path**, type a valid path and file name.

Changing the Priority for DNS SRV Records

Perform this procedure to raise the priority for the PDC emulator's DNS SRV records in the domain where you are expiring all user passwords. Use Regedit.exe to perform this procedure.

Requirements

- Credentials: Domain Admins or Enterprise Admins

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

To change the priority of a domain controller

1. On the PDC emulator, in the **Run** dialog box, type **regedit**, and press ENTER.
2. In the registry editor, navigate to
HKLM\SYSTEM\CurrentControlSet\Services\Netlogon\Parameters
3. Click **Edit**, click **New**, and then click **DWORD value**.
4. For the new value name, type **LdapSrvPriority**, and press ENTER.
5. Double-click the value name that you just typed to open the **Edit DWORD Value** dialog box.
6. Enter a value from 0 through 65535. The default value is 0. The higher the value that you choose relative to the other domain controllers in the domain, the less likely that this domain controller will be contacted.
7. Choose **Decimal** as the **Base** option.
8. Click **File**, and then click **Exit** to close the registry editor.

Important: For this setting to take effect, you must stop and restart the Netlogon service on the PDC emulator. To stop the Netlogon service, at the command line type “net stop netlogon”. To restart the Netlogon service, at the command line, type “net start netlogon”.

Changing the Weight for DNS SRV Records

Perform this procedure to lower the weight of the PDC emulator’s DNS SRV records in the domain where you are expiring all user passwords. Use Regedit.exe to perform this procedure.

Requirements

- Credentials: Domain Admins or Enterprise Admins

To change the weight of the PDC emulator’s DNS SRV Record:

1. On the PDC emulator, in the **Run** dialog box, type **regedit**, and press ENTER.
2. In the registry editor, navigate to
HKLM\SYSTEM\CurrentControlSet\Services\Netlogon\Parameters.
3. Click **Edit**, click **New**, and then click **DWORD value**.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

4. For the new value name, type **LdapSrvWeight** and press ENTER. (The value name is not case sensitive.)
5. Double-click the value name you just typed to open the **Edit DWORD Value** dialog box.
6. Enter a value from 0 through 65535. The default value is 100. The lower the value that you choose relative to the other domain controllers in the domain, the less likely that this domain controller will be contacted.
7. Choose **Decimal** as the **Base** option.
8. Click **File**, and then click **Exit** to close the registry editor.

Important: For this setting to take effect, you must stop and restart the Netlogon service on the PDC emulator. To stop the Netlogon service, at the command line type “net stop netlogon”, To restart the Netlogon service, at the command line, type “net start netlogon”.

Converting the GUID of a GPO to a Friendly Name

When an event log entry is generated because a GPO is modified, the GUID of the GPO that is modified is listed in the Description field of the event. You can determine the *friendly name* of the GPO from the GUID. The friendly name is the name that is show in consoles, such as Active Directory Users and Computers.

Requirements

- Credentials: Domain Admins
- Tools: Ldifde.exe, Notepad.exe

To convert the GUID of a GPO to a friendly name

1. Find the event in the event log for the modification of the GPO.
2. Double-click the event entry in the log.
3. Click the copy button on the event, illustrated in Figure 7.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

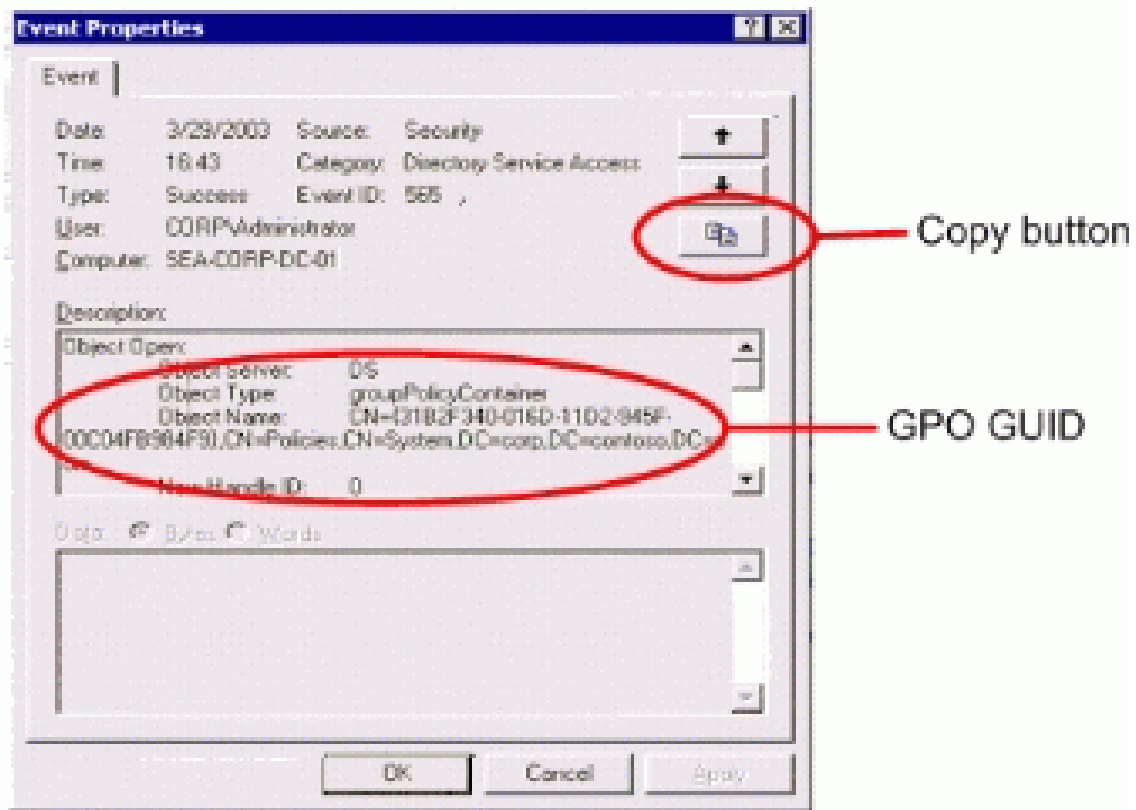


Figure 9: GPO Modification Event

4. Start Notepad.exe.
5. Paste the event copied in Step 3 into Notepad.
6. At a command prompt, type the following *without* pressing ENTER:
 7. `ldifde -f output.txt -d "<domain>" -r "cn=<guid>" -l displayName,cn`
 8. Paste the GUID from Notepad to *<guid>* in the command line, and then press ENTER.

The following is an example of how the command line appears after pasting the GUID from Notepad:

```
ldifde -f output.txt -d "DC=corp,DC=contoso,DC=com" -r  
"cn={31B2F340-016D-11D2-945F-00C04FB984F9})" -l displayName,cn
```

9. Open output.txt to view the friendly name of the GPO.

Forcing Password Expiration Using a Script

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

Forcing the expiration of all user account passwords and then manually resetting the passwords would be impractical in any but a very small organization. For larger organizations, consider forcing the expiration of user account passwords with a script like the one provided below.

Description

This script forces users to change their account passwords the next time they logon.

Script Code

```
Set objConnection = CreateObject("ADODB.Connection")
objConnection.Open "Provider=ADsDSOObject;"
Set objCommand = CreateObject("ADODB.Command")
objCommand.ActiveConnection = objConnection
objCommand.CommandText = _
"<LDAP://ou=Management,dc=NA,dc=fabrikam,dc=com>" & _
"(&(objectCategory=Person)(objectClass=user));" & _
"ADsPath;subtree"
Set objRecordSet = objCommand.Execute
While Not objRecordset.EOF
  strADsPath = objRecordset.Fields("ADsPath")
  Set objUser = GetObject(strADsPath)
  objUser.Put "pwdLastSet", 0
  objUser.SetInfo
  objRecordSet.MoveNext
Wend
WScript.Echo objRecordSet.RecordCount & _
" user account objects must change the " & _
"password at next logon."
objConnection.Close
Disclaimer
```

The sample scripts are not supported under any Microsoft standard support program or service. The sample scripts are provided AS IS without warranty of any kind. Microsoft further disclaims all implied warranties including, without limitation, any implied warranties of merchantability or of fitness for a particular purpose. The entire risk arising out of the use or performance of the sample scripts and documentation remains with you. In no event shall Microsoft, its authors, or anyone else involved in the creation, production, or delivery of the scripts be liable for any damages whatsoever (including,

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use of or inability to use the sample scripts or documentation, even if Microsoft has been advised of the possibility of such damages.

Identifying Computers to Receive New Registry Settings with ComputerSearch.vbs

You can use the ComputerSearch.vbs script to help identify computers objects in Active Directory so that you can subsequently apply registry settings to the list of computers identified by ComputerSearch.vbs. Before you can use the ComputerSearch.vbs script, you must create the script based on the source code included in this document.

Note: This script requires Windows 2000 with Service Pack 3 or later and Windows Scripting Host version 5.6 or later.

The ComputerSearch.vbs script has the following syntax:

```
cscript ComputerSearch.vbs [/r:role]
```

Where *role* can be "DC" for domain controllers or "MC" for member computers.

If you omit role, then ComputerSearch.vbs will return both domain controllers and member computers.

ComputerSearch.vbs creates an output file, ComputerSearch-*date-time*.csv, which contains a list of the computer objects found in Active Directory.

To create the ComputerSearch.vbs script

1. Open Notepad
2. Copy or type the following script into Notepad
3. *****
4. '* File: ComputerSearch.vbs *
5. '* Created: March 2003 *
6. '* Version: 1.0 *
7. '* *
8. '* Description: Diagnostic utility that returns a report containing *
9. '* the distinguished names of computers in the domain *
10. '* (domain controllers, member computers or both). This *
11. '* report can be viewed in a spreadsheet or database *
12. '* program and it can be read by ApplyReg.vbs to apply *

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

```
13. '* registry changes to all the computers listing in '*
14. '* the report. '*
15. '* '*
16. '* Compatibility: This script requires WSH 5.6, CScript, ADSI, '*
17. '* and access to Active Directory '*
18. '*****
19. Option Explicit
20. 'Define any constants used in this script
21. 'Denotes a workstation:
22. Const ADS_UF_WORKSTATION_TRUST_ACCOUNT = &h1000
23. 'Denotes a DC:
24. Const ADS_UF_SERVER_TRUST_ACCOUNT = &h2000
25. Const ForAppending = 2
26. Dim objArgs,strRole
27. Dim objRootDSE,strDomain
28. Dim objConnection,objCommand,objRecordSet
29. Dim strFileName,objFSO,objFile
30. Call CheckForCScript
31. 'Use Named Arguments collection for the command line argument.
32. 'The WSHArguments Object is included in WSH version 5.6 and later
33. Set objArgs = WScript.Arguments.Named
34. strRole = UCase(objArgs.Item("r"))
35. If WScript.Arguments.Named.Count < 1 Then
36. WScript.Echo "No role was specified on the command-line." & VbCrLf & _
37. "Therefore, the report will list both domain controller and" & VbCrLf & _
38. "member computers in the domain." & VbCrLf & _
39. "Possible roles are: dc (domain controller) or mc (member computer)."
40. End If
41. If WScript.Arguments.Named.Exists("r") Then
42. If strRole <> "DC" AND strRole <> "MC" Then
43. WScript.Echo SampleCommandLine()
44. WScript.Quit
45. End If
46. End If
47. 'Use the RootDSE object for upcoming search operation
48. Set objRootDSE = GetObject("LDAP://rootDSE")
```

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

```
49. 'Bind to the current domain
50. 'Use to specify the search base for the LDAP search
51. strDomain = "<LDAP://" & _
52.   objRootDSE.Get("defaultNamingContext") & ">"
53. Set objConnection = CreateObject("ADODB.Connection")
54. objConnection.Open "Provider=ADsDSOObject;"
55. Set objCommand = CreateObject("ADODB.Command")
56. objCommand.ActiveConnection = objConnection
57. objCommand.CommandText = strDomain & _
58.   ":(objectCategory=computer);" & _
59.   "distinguishedName,userAccountControl;subtree"
60.   "distinguishedName,userAccountControl,samAccountName,dNSHostName;subtree"
61. 'Specify page size for this command object.
62. 'This is necessary to avoid overutilization of server
63. 'and network resources. Also, by default,
64. 'only 1000 records will be returned if paging isn't
65. 'specified. The domain might contain more
66. 'than 1000 computer objects.
67. objCommand.Properties("Page Size") = 256
68. objCommand.Properties("Asynchronous") = True
69. objCommand.Properties("Cache results") = False
70. 'Run the computer object query
71. Set objRecordSet = objCommand.Execute
72. 'Create a unique file name (timestamp) using the GenFileName function
73. strFileName = GenFileName("ComputerSearch")
74. 'Create a text file (.csv format) to hold the
75. 'results of the class test.
76. Set objFSO = CreateObject("Scripting.FileSystemObject")
77. 'This will overwrite the file if it already exists.
78. Set objFile = objFSO.CreateTextFile(strFileName,True)
79. objFile.Close
80. Set objFile = objFSO.OpenTextFile(strFileName,ForAppending)
81. Call PopulateReport
82. objConnection.Close
83. WScript.Echo VbCrLf & "The report data has been saved to: " & strfileName & "."
84. *****
```

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

```
85. '* Routine: PopulateReport
86. *****
87. Sub PopulateReport
88.   On Error Resume Next
89.   Dim strRecord,iCnt,i
90.   i=0
91.   ObjFile.WriteLine "distinguishedName,Role"
92.   While Not objRecordset.EOF
93.     If strRole = "" Then
94.       If ADS_UF_SERVER_TRUST_ACCOUNT AND
          objRecordset.Fields("userAccountControl") Then
95.         strRecord = GenRecord & "Domain Controller"
96.         objFile.WriteLine strRecord
97.         i=i+1
98.       ElseIf ADS_UF_WORKSTATION_TRUST_ACCOUNT AND
          objRecordset.Fields("userAccountControl") Then
99.         strRecord = GenRecord & "Workstation or Server"
100.        objFile.WriteLine strRecord
101.        i=i+1
102.      End If
103.    ElseIf strRole = "DC" Then
104.      If ADS_UF_SERVER_TRUST_ACCOUNT AND
          objRecordset.Fields("userAccountControl") Then
105.        strRecord = GenRecord & "Domain Controller"
106.        objFile.WriteLine strRecord
107.        i=i+1
108.      End If
109.    ElseIf strRole = "MC" Then
110.      If ADS_UF_WORKSTATION_TRUST_ACCOUNT AND
          objRecordset.Fields("userAccountControl") Then
111.        strRecord = GenRecord & "Workstation or Server"
112.        objFile.WriteLine strRecord
113.        i=i+1
114.      End If
115.    End If
116.    'Progress indicator
```

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

```
117. For iCnt = 1 to Len(i) + 35
118.     WScript.StdOut.Write Chr(8)
119. Next
120.     WScript.StdOut.Write "Current number of computers found: " & i
121.     objRecordset.MoveNext
122. Wend
123. End Sub
124. *****
125. '* Routine: CheckForCScript
126. *****
127. Sub CheckForCScript
128. 'This script must run from cscript because
129. 'it uses the WScript.StdOut property.
130. 'Test the script host and if it's not cscript,
131. 'instruct the operator on how to run the script.
132. If Right(LCase(WScript.FullName),11) <> LCase("cscript.exe") Then
133.     WScript.Echo "This script must run from cscript." & _
134.     VbCrLf & "Example: cscript ComputerSearch.vbs"
135.     WScript.Quit
136. End If
137. End Sub
138. *****
139. '* Function: PadZero
140. *****
141. Function PadZero(dtValue)
142.     If Len(dtValue) = 1 Then
143.         PadZero = 0 & dtValue
144.     Else
145.         PadZero = dtValue
146.     End If
147. End Function
148. *****
149. '* Function: GenFileName
150. *****
151. Function GenFileName(prefix)
152. 'Create a unique time stamped name for the text file
```

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

```
153. Dim dtDate,strYear,strMonth,strDay,strDate
154. Dim dtNow,strHour,strMinute,strSecond,strTime
155. dtDate = Date()
156. strYear = Mid(Year(dtDate),3)
157. strMonth = PadZero(Month(dtDate))
158. strDay = PadZero(Day(dtDate))
159. strDate = strYear & strMonth & strDay & "-"
160. dtNow = Now()
161. strHour = PadZero(Hour(dtNow))
162. strMinute = PadZero(Minute(dtNow))
163. strSecond = PadZero(Second(dtNow))
164. strTime = strHour & strMinute & strSecond
165. GenFileName = prefix & "-" & strDate & strTime & ".csv"
166. End Function
167. *****
168. '* Function: SampleCommandLine
169. *****
170. Function SampleCommandLine()
171. SampleCommandLine = _
172. "Specify the /r parameter to limit the computer list to" & VbCrLf & _
173. "only domain controllers or only member computers in the domain." & VbCrLf & _
174. "ComputerSearch.vbs /r:dc" & VbCrLf & _
175. "or" & VbCrLf & _
176. "ComputerSearch.vbs /r:mc"
177. End Function
178. *****
179. '* Function: GenRecord
180. *****
181. Function GenRecord()
182. Dim strRecord
183. strRecord = chr(34) & objRecordset.Fields("distinguishedName") & chr(34) & ","
184. GenRecord = strRecord
185. End Function
186. Save the file as ComputerSearch.vbs.
```

Identifying Large-sized Objects with ObjMemUse.vbs

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

You can use the ObjMemUse.vbs script to help find rogue objects in Active Directory that are consuming a significant amount of disk space. Before you can use the ObjMemUse.vbs script, you must create the script based on the source code included in this document.

Note: This script requires Windows 2000 with Service Pack 3 or later and Windows Scripting Host version 5.6 or later.

The ObjMemUse.vbs script has the following syntax:

```
cscript ObjMemUse.vbs /f:input_file [/s:object_size]
```

Where:

- *input_file* is the name of the file, including path, created by ObCMAudit.vbs.
- *object_size* is the minimum size, in Kb, of an object that you consider to be large. If *object_size* is omitted, the ObjMemUse will consider any objects larger than 50Kb to be large object.

ObjMemUse.vbs creates an output file, ObjMemUse-*date-time*.csv, which contains a list of the objects listed in the file created by ObCMAudit.vbs and the size of each object.

Note: ObjMemUse.vbs provides only an estimate of an object's size. The estimate can vary by as much as 4K - 8K. This means that an object that is less than 4K in size might be estimated as 0K. However, when the object is large enough that the 4K variance is negligible, then the difference between the size of the object and the estimate returned by ObjMemUse.vbs is negligible.

To create the ObjMemUse.vbs script.

1. Open Notepad.
2. Copy or type the following script into Notepad.

```
3. *****
4.  '* File:      ObjMemUse.vbs                *
5.  '* Created:   April 2003                  *
6.  '* Version:   1.0                        *
7.  '*           *
8.  '* Description: Security diagnostic utility that creates a report *
9.  '*           containing objects that appear to be large. This *
10. '*           utility uses a report created by ObjCMAudit.vbs *
```

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

```
11. '*          to estimate the size of the created or modified      *
12. '*          objects.                                           *
13. '*          *
14. '* Notes:          This script uses the Raw performance counter class *
15. '*          for two reasons:                                     *
16. '*          1. The cooked (calculated) counter is not currently *
17. '*          available in Windows 2000.                          *
18. '*          2. The cooked counter uses a cooking type of        *
19. '*          PERF_COUNTER_LARGE_RAWCOUNT, which doesn't require *
20. '*          any further calculations but the cooked counter     *
21. '*          is 64-bits in length to support very large values. *
22. '*          *
23. '* Compatibility: This script requires WSH 5.6, CScript, ADSI, WMI, *
24. '*          and access to Active Directory                       *
25. '*          *****
26. Option Explicit
27. 'Define any constants used in this script
28. Const ForAppending = 2
29. Const ForReading = 1
30. 'Declare global variables
31. Dim objArgs,strRptFileName,intSize,strFileName
32. Dim objFSO,objFile,objRptFile
33. Dim objRootDSE,strDomain,objADObject
34. Dim strLine,strDN
35. Dim intWSBefore,intWSAfter
36. Dim i,iCnt,intObjSizeBaseline
37. Dim intCurObjSize,intCalculatedSize
38. Call CheckForCScript()
39. 'Use Named Arguments collection for the command line argument.
40. 'The WSHArguments Object is included in WSH version 5.6 and later
41. Set objArgs = WScript.Arguments.Named
42. strRptFileName = objArgs.Item("f")
43. intSize = Round(Abs(objArgs.Item("s"))) * 1024
44. 'Verify the command-line arguments
45. If ValidArgsTesting() = False Then WScript.Quit
46. 'Call the GenFileName function to create a unique file name for the report
```

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

```
47. strFileName = GenFileName("objMemUse")
48. 'Create a text file (.csv format) for the report.
49. Set objFSO = CreateObject("Scripting.FileSystemObject")
50. 'This will overwrite the file if it already exists.
51. Set objFile = objFSO.CreateTextFile(strFileName,True)
52. objFile.Close
53. Set objFile = objFSO.OpenTextFile(strFileName,ForAppending)
54. 'Write the headings for this csv file
55. objFile.WriteLine "DistinguishedName," & _
56. "Change in Working Set Memory (bytes)"
57. Set objRptFile = objFSO.OpenTextFile(strRptFileName,ForReading)
58. 'Skip the header row of the report
59. objRptFile.SkipLine
60. Set objRootDSE = GetObject("LDAP://rootDSE")
61. strDomain = objRootDSE.Get("defaultNamingContext")
62. Set objRootDSE = Nothing
63. 'Tare the local property cache w/a default AD object.
64. Set objADObject = GetObject("LDAP://cn=BuiltIn," & strDomain)
65. 'Check memory before calling GetInfo
66. intWSBefore = CheckMemory(0)
67. objADObject.GetInfo
68. 'Check memory after calling GetInfo
69. intWSAfter = CheckMemory(0)
70. Set objADObject = Nothing
71. intCurObjSize = intWSAfter - intWSBefore
72. 'Calculate baseline working set memory
73. intObjSizeBaseline = BaseLine()
74. WScript.Echo "Report records processed: "
75. i=1
76. While Not objRptFile.AtEndOfStream
77. 'Extract a line from the report
78. strLine = objRptFile.ReadLine
79. 'Return everything except for the first quotation mark
80. strLine = Mid(strLine,2)
81. 'Return everything up to the second quotation mark
82. strDN = Split(strLine,chr(34))
```

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

```
83. 'Bind to the object
84. Set objADObject = GetObject("LDAP://" & strDN(0))
85. 'Check memory before calling GetInfo
86. intWSBefore = CheckMemory(0)
87. objADObject.GetInfo
88. 'Check memory after calling GetInfo
89. intWSAfter = CheckMemory(0)
90. Set objADObject = Nothing
91. intCalculatedSize = ((intWSAfter - intWSBefore - intObjSizeBaseline)/2)
92. If intCalculatedSize >= intSize Then
93.     'Write data to the new report file
94.     objFile.WriteLine Chr(34) & strDN(0) & chr(34) & "," & intCalculatedSize
95. End If
96. For iCnt = 1 to Len(i)
97.     WScript.StdOut.Write Chr(8)
98. Next
99.     WScript.StdOut.Write i
100. i = i + 1
101. Wend
102. WScript.Echo VbCrLf & VbCrLf & "The report data has been saved to: " & _
103. strfileName & "." & VbCrLf & _
104. "Import or open the CSV data in a spreadsheet" & VbCrLf & _
105. "or database program to analyze object size estimates."
106. *****
107. '* Routine: CheckForCScript
108. *****
109. Sub CheckForCScript
110. 'This script must run from cscript because
111. 'it uses the WScript.StdOut property.
112. 'Test the script host and if it's not cscript,
113. 'instruct the operator on how to run the script.
114. If Right(LCase(WScript.FullName),11) <> LCase("cscript.exe") Then
115.     WScript.Echo "This script must run from cscript." & _
116.     VbCrLf & "Example: cscript ObjMemUse.vbs"
117.     WScript.Quit
118. End If
```

**Best Practice Guide for Securing Active Directory
Installations and Day to Day Operations Part II**
By Kathleen Cole, Jennifer Bayer, Doug Steen

```
119. End Sub
120. *****
121. '* Function: ValidArgsTesting
122. *****
123. Function ValidArgsTesting
124.   If WScript.Arguments.Named.Count < 1 Then
125.     WScript.Echo "You must specify a csv file name " & VBCrLf & _
126.       "of a file created by ObjCMAudit.vbs." & VbCrLf & _
127.       SampleCommandLine()
128.     ValidArgsTesting = False
129.   ElseIf Wscript.Arguments.Named.Exists("f") AND Wscript.Arguments.Named.Count = 1
       Then
130.     WScript.echo "No size value was specified. Therefore, objects that require" & _
131.       VbCrLf & "50kb or more of the local property cache will be reported."
132.     intSize = 50 * 1024
133.     ValidArgsTesting = True
134.   ElseIf WScript.Arguments.Named.Exists("s") AND IsNumeric(intSize) = FALSE Then
135.     WScript.Echo "The size value that you specified is not valid." & _
136.       VbCrLf & "You must specify a number." & _
137.       VbCrLf & SampleCommandLine()
138.     ValidArgsTesting = False
139.   ElseIf WScript.Arguments.Named.Count >= 1 AND _
140.     WScript.Arguments.Named.Exists("f") = false Then
141.     WScript.Echo "The /f: parameter is required."
142.     WScript.Echo VbCrLf & SampleCommandLine()
143.     ValidArgsTesting = False
144.   Else
145.     WScript.Echo "Objects that require " & intSize & "kb or more " & _
146.       "than the Builtin container object in the local property" & _
147.       VbCrLf & "cache will be reported." & _
148.       VbCrLf & "Note, this script rounds the number specified to the nearest" & _
149.       VbCrLf & "whole number. Negative numbers are not allowed and are automatically" & _
150.       VbCrLf & "converted to positive values."
151.     ValidArgsTesting = True
152.   End If
153. End Function
```

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

```
154. *****
155. '* Function: SampleCommandLine
156. *****
157. Function SampleCommandLine()
158.     SampleCommandLine = _
159.     "For example, to create a report on all objects" & _
160.     VbCrLf & "that consume 250kb of memory to complete" & VbCrLf & _
161.     " a binding operation, based on computers listed in" & VbCrLf & _
162.     " a file named obcmReport.csv, type:" & VbCrLf & _
163.     "ObjMemUse.vbs /f:obcmReport.csv /s:250"
164. End Function
165. *****
166. '* Function: BaseLine
167. *****
168. Function BaseLine
169.     Dim blnVal,intPreObjSize,intCount
170.     intPreObjSize = 0
171.     intCount= 0
172.     blnVal = False
173.     Do Until blnVal = True
174.         If intPreObjSize = intCurObjSize Then
175.             intCount = intCount + 1
176.             If intCount = 10 then
177.                 blnVal = True
178.                 'intObjSizeBaseline = intCurObjSize
179.                 Baseline = intCurObjSize
180.             End If
181.         Else
182.             intCount = 0
183.             intPreObjSize = intCurObjSize
184.             Set objADObject = GetObject("LDAP://cn=BuiltIn," & strDomain)
185.             intWSBefore = CheckMemory(0)
186.             objADObject.GetInfo
187.             intWSAfter = CheckMemory(0)
188.             Set objADObject = Nothing
189.             intCurObjSize = intWSAfter - intWSBefore
```

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

```
190. End If
191. Loop
192. End Function
193. *****
194. '* Function: CheckMemory
195. *****
196. Function CheckMemory(index)
197. Dim blnValue,intPreWorkingSet,intCurWorkingSet,intMemorySameCount
198. intMemorySameCount = 0
199. intPreWorkingSet = 0
200. intCurWorkingSet = _
201. (GetObject("winmgmts:Win32_PerfRawData_PerfProc_Process='cscrip").WorkingSet)
202. blnValue = False
203. Do Until blnValue = True
204. If intPreWorkingSet = intCurWorkingSet Then
205. intMemorySameCount = intMemorySameCount + 1
206. If intMemorySameCount = 10 then
207. blnValue = true
208. CheckMemory = intCurWorkingSet
209. End If
210. Else
211. intMemorySameCount = 0
212. intPreWorkingSet = intCurWorkingSet
213. intCurWorkingSet = _
214. (GetObject("winmgmts:Win32_PerfRawData_PerfProc_Process='cscrip").WorkingSet)
215. End If
216. Loop
217. End Function
218. *****
219. '* Function: PadZero
220. *****
221. Function PadZero(dtValue)
222. If Len(dtValue) = 1 Then
223. PadZero = 0 & dtValue
224. Else
225. PadZero = dtValue
```

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

```
226. End If
227. End Function
228. *****
229. '* Function: GenFileName
230. *****
231. Function GenFileName(prefix)
232. 'Create a unique time stamped name for the text file
233. Dim dtDate,strYear,strMonth,strDay,strDate
234. Dim dtNow,strHour,strMinute,strSecond,strTime
235. dtDate = Date()
236. strYear = Mid(Year(dtDate),3)
237. strMonth = PadZero(Month(dtDate))
238. strDay = PadZero(Day(dtDate))
239. strDate = strYear & strMonth & strDay & "-"
240. dtNow = Now()
241. strHour = PadZero(Hour(dtNow))
242. strMinute = PadZero(Minute(dtNow))
243. strSecond = PadZero(Second(dtNow))
244. strTime = strHour & strMinute & strSecond
245. GenFileName = prefix & "-" & strDate & strTime & ".csv"
246. End Function
247.     Save the file as ObjMemUse.vbs
```

Identifying Objects Created or Modified Within a Period of Time with ObCMAudit.vbs

You can use the ObCMAudit.vbs script to identify objects that are created or modified in a domain within a given period of time to help you identify any rogue objects in Active Directory. Before you can use the ObCMAudit.vbs script, you must create the script based on the source code included in this document.

Note: This script requires Windows 2000 with Service Pack 3 or later and Windows Scripting Host version 5.6 or later.

The ObCMAudit.vbs script has the following syntax:

```
cscript ObCMAudit.vbs [/d:date] [/t:time] /h:length
```

Where:

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

- *date* is the date when you detected the disk space attack. This parameter is optional.
- *time* is the time you detected the disk space attack. This parameter is optional.
- *length* is the length of time, in hours, prior to the time you detected the disk space attack based on the trend in the growth of the rogue objects discovered in Step 4. ObCMAudit.vbs has no parameters.

ObCMAudit.vbs creates an output file, ObCMAudit-*date-time*.csv, that contains a list of the objects created or modified during the length of time.

To create the ObCMAudit.vbs script.

1. Open Notepad.
2. Copy or type the following script into Notepad.

```
3. *****
4. '* File:      ObjCMAudit.vbs          *
5. '* Created:   March 2003             *
6. '* Version:   1.0                    *
7. '*           *
8. '* Description: Security diagnostic utility that creates a report *
9. '*           containing all objects created or modified within a *
10. '*           specified time window. Use this tool to build a *
11. '*           report of object creation and modification activity. *
12. '*           You can then complete two common tasks with the *
13. '*           report. *
14. '*           1. Use the ObjMemUse.vbs file to read the report *
15. '*           and estimate the size of the created objects. *
16. '*           2. Import the csv file into a spreadsheet program *
17. '*           or a database program to analyze the results. *
18. '*           *
19. '* Compatibility: This script requires WSH 5.6, CScript, ADSI, WMI *
20. '*           and access to Active Directory *
21. *****
22. Option Explicit
23. 'Define any constants used in this script
24. Const ForAppending = 2
```

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

```
25. 'Declare global variables
26. Dim objArgs,intHours,dtFromDate,dtFromTime,dtVal,intLocalTime
27. Dim objRootDSE
28. Dim strFileName,objFSO,objFile
29. Dim objConnection,objCommand
30. Dim arrNamingContexts,NamingContext,strContainer
31. 'This script must run from cscript because
32. 'it uses the WScript.StdOut property.
33. 'Test the script host and if it's not cscript,
34. 'instruct the operator how to run the script.
35. If Right(LCase(WScript.FullName),11) <> LCase("cscript.exe") Then
36.   WScript.Echo "This script must run from cscript." & _
37.     VbCrLf & SampleCommandLine() & VbCrLf & _
38.     VbCrLf & "Check the locale setting of your system for the correct" & _
39.     " date and time format."
40.   WScript.Quit
41. End If
42. 'Use Named Arguments collection for the command line argument.
43. 'The WSHArguments Object is included in WSH version 5.6 and later
44. Set objArgs = WScript.Arguments.Named
45. intHours = clnt(objArgs.Item("h"))
46. dtFromDate = objArgs.Item("d")
47. dtFromTime = objArgs.Item("t")
48. If WScript.Arguments.Named.Exists("d") AND _
49.   WScript.Arguments.Named.Exists("t") Then
50.   dtVal = DateValue(dtFromDate) & " " & TimeValue(dtFromTime)
51.   If IsDate(dtVal) = False Then
52.     WScript.Echo "The date or time value that you specified is not valid." & _
53.       VbCrLf & "You must specify a valid date and time." & _
54.       VbCrLf & SampleCommandLine()
55.     WScript.Quit
56.   End If
57. ElseIf WScript.Arguments.Named.Count = 2 Then
58.   WScript.Echo "A command line parameter is missing." & _
59.     VbCrLf & SampleCommandLine() & VbCrLf _
60.     VbCrLf & "Check the locale setting of your system for the correct" & _
```

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

```
61.     " date and time format."
62.     WScript.Quit
63.     ElseIf WScript.Arguments.Named.Count = 1 AND _
64.     WScript.Arguments.Named.Exists("h") Then
65.     WScript.Echo "No date and time endpoint was provided. " & VbCrLf & _
66.     "Therefore, the time window endpoint is based upon " & VbCrLf & _
67.     "the current system time."
68.     dtVal = Now()
69.     End If
70.     If WScript.Arguments.Named.Count < 1 Then
71.     WScript.Echo "You must specify a time window (in hours) " & VbCrLf & _
72.     "within which objects were created or modified." & VbCrLf & _
73.     SampleCommandLine()
74.     WScript.Quit
75.     Else
76.     intLocalTime = LocalTime()
77.     'Call the GenFileName function to create a unique file name
78.     strFileName = GenFileName("ObjCMAudit")
79.     'Create a text file (.csv format) to hold the
80.     'results of the report.
81.     Set objFSO = CreateObject("Scripting.FileSystemObject")
82.     'This will overwrite the file if it already exists.
83.     Set objFile = objFSO.CreateTextFile(strFileName,True)
84.     objFile.Close
85.     Set objFile = objFSO.OpenTextFile(strFileName,ForAppending)
86.     'Write the headings for this csv file
87.     objFile.WriteLine "DistinguishedName,Class,Action,Date"
88.     'Create the ADSI OLE DB provider object
89.     Set objConnection = CreateObject("ADODB.Connection")
90.     objConnection.Open "Provider=ADsDSOObject;"
91.     'Create the command object and set the ActiveConnection
92.     'property equal to the command object
93.     Set objCommand = CreateObject("ADODB.Command")
94.     objCommand.ActiveConnection = objConnection
95.     'Specify Page size for this command object.
96.     'The domain and configuration containers are likely to contain
```

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

```
97. 'large result sets
98. objCommand.Properties("Page Size") = 256
99. objCommand.Properties("Asynchronous") = True
100. 'This will be used as a forward only recordset
101. 'so caching isn't necessary and, if a large result set
102. 'is returned, caching will consume too much memory.
103. objCommand.Properties("Cache results") = False
104. 'Use the RootDSE object for upcoming binding operations
105. Set objRootDSE = GetObject("LDAP://rootDSE")
106. 'Get the DNs for all naming contexts defined on the DC
107. arrNamingContexts = objRootDSE.GetEx("namingContexts")
108. 'Enumerate the list of Naming Contexts and write a multi-line
109. 'report file.
110. For Each NamingContext in arrNamingContexts
111. 'Set the strContainer variable equal to the binding string of the container.
112. strContainer = "<LDAP://" & NamingContext & ">"
113. WScript.Echo VbCrLf & "Reporting on objects in the " & _
114. NamingContext & " container" & VbCrLf & _
115. "created or modified between " & DateAdd("h",-intHours,dtVal) & " and " & dtVal
116. Call CheckDates(strContainer,intHours)
117. Next
118. 'Clean up
119. Set objRootDSE = Nothing
120. End If
121. WScript.Echo VbCrLf & VbCrLf & "The report data has been saved to: " & _
122. strFileName & "." & VbCrLf & _
123. "Import or open the CSV data in a spreadsheet" & VbCrLf & _
124. "or database program and/or rename the file and use" & VbCrLf & _
125. "the report data to analyze object size using the ObjMemUse.vbs script." & VbCrLf & _
126. "For example, rename " & strFileName & " to ocm.csv and then run the " & VbCrLf & _
127. "object size analysis script by typing: cscript ObjMemUse.vbs /f:ocm.csv"
128. *****
129. '* Routine: CheckDates
130. *****
131. Sub CheckDates(Container,Hours)
132. Dim objRSContainer,dtWhenCreated,dtWhenChanged
```

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

```
133. Dim dtAdjwhenCreated,dtAdjwhenChanged
134. Dim dtWChgDiff,dtWCreDiff
135. Dim strDN,arrClass,iCnt,i
136. 'Construct the query.
137. 'Note: Using objectClass as an attribute to return because objectCategory
138. 'does not contain a value for all classes.
139. objCommand.CommandText = Container & _
140. ";&(objectCategory=*)(!objectCategory=foreignSecurityPrincipal);" & _
141. "distinguishedName,whenCreated,whenChanged,objectClass;subtree"
142. 'Run the query
143. Set objRSContainer = objCommand.Execute
144. i = 0
145. WScript.Echo "Records processed:"
146. 'Iterate the recordset
147. While Not objRSContainer.EOF
148.   dtWhenCreated = objRSContainer.Fields("whenCreated")
149.   dtWhenChanged = objRSContainer.Fields("whenChanged")
150.   'Test for null values in whenChanged and whenCreated
151.   'For example, whenCreated and whenChanged are not mandatory attributes
152.   'and are not set for objects that are instantiated from the crossRefContainer
153.   'structural class
154.   If (IsNull(dtWhenCreated) OR dtWhenCreated < CDate("01/01/1990")) OR _
155.     (IsNull(dtWhenChanged) OR dtWhenChanged < CDate("01/01/1990")) Then
156.     If IsNull(dtWhenCreated) AND IsNull(dtWhenChanged) Then
157.       objFile.WriteLine ReadAndWrite(objRSContainer,"Created and Changed","not set")
158.     ElseIf IsNull(dtWhenCreated) Then
159.       objFile.WriteLine ReadAndWrite(objRSContainer,"Created","not set")
160.     ElseIf IsNull(dtWhenChanged) Then
161.       objFile.WriteLine ReadAndWrite(objRSContainer,"Changed","not set")
162.     End If
163.   Else
164.     'Convert the time returned to local time by adding or subtracting
165.     'from GMT.
166.     dtAdjwhenCreated = DateAdd("h",intLocalTime,dtWhenCreated)
167.     dtAdjwhenChanged = DateAdd("h",intLocalTime,dtWhenChanged)
168.     'Take the number of hours and date time endpoint provided by
```

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

```
169. 'the operator and return all objects created from that hour forward
170. 'The date value (dtVal) is the date and time endpoint of the test.
171. dtWChgDiff = DateDiff("h",dtAdjwhenChanged,dtVal)
172. dtWCreDiff = DateDiff("h",dtAdjwhenCreated,dtVal)
173. 'The sgn function is necessary in the next two If Then evaluations.
174. 'Otherwise, these evaluations will return values that are greater than
175. 'the defined date/time endpoint.
176. If ((Hours >= Cint(dtWChgDiff) AND Sgn(Cint(dtWChgDiff)) <> -1) AND _
177. (Hours >= Cint(dtWCreDiff) AND Sgn(Cint(dtWCreDiff)) <> -1)) Then
178. objFile.WriteLine ReadAndWrite(objRSContainer,"[Created] \ [Changed]", _
179. "[" & dtAdjwhenCreated & "]" \ "[" & dtAdjwhenChanged & "]")
180. Elseif Hours >= Cint(dtWChgDiff) AND Sgn(Cint(dtWChgDiff)) <> -1 Then
181. objFile.WriteLine ReadAndWrite(objRSContainer,"Changed","[" & dtAdjwhenChanged
    & "]")
182. Elseif Hours >= Cint(dtWCreDiff) AND Sgn(Cint(dtWCreDiff)) <> -1 Then
183. objFile.WriteLine ReadAndWrite(objRSContainer,"Created","[" & dtAdjwhenCreated &
    "]")
184. End If
185. End If
186. For iCnt = 1 to Len(i)
187. WScript.StdOut.Write Chr(8)
188. Next
189. WScript.StdOut.Write i
190. i = i + 1
191. objRSContainer.MoveNext
192. Wend
193. End Sub
194. '*****
195. '* Function: LocalTime
196. '*****
197. 'Get local time zone offset. Requires WMI and the Win32_TimeZone class
198. 'which is part of Windows 2000 and later.
199. Function LocalTime()
200. Dim objWMIService,colTimeZones,objTimeZone
201. Set objWMIService =
    GetObject("winmgmts:{impersonationLevel=Impersonate}!\.\root\cimv2")
```

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

```
202. Set colTimeZones = objWmiService.InstancesOf("Win32_TimeZone")
203. For Each objTimeZone In colTimeZones
204.   LocalTime = Fix(objTimeZone.Bias / 60)
205. Next
206. End Function
207. *****
208. '* Function: SampleCommandLine
209. *****
210. Function SampleCommandLine()
211.   SampleCommandLine = VbCrLf & _
212.     "Example: To test a 4 hour time window on 03-07-2003" & VbCrLf & _
213.     "from 9:30 AM and 50 seconds to 1:30 PM and 50 seconds, type:" & VbCrLf & _
214.     "cscript ObjCMAudit.vbs /h:4 " & _
215.     "/d:03-07-2003 /t:1:30:50PM" & VbCrLf & VbCrLf & _
216.     "You can also enter a time value using a 24-hour clock." & VbCrLf & _
217.     "Example: To test a 2 hour time window on 03-07-2003" & VbCrLf & _
218.     "from 14:00 (2:00pm) to 16:00 (4:00pm), type:" & VbCrLf & _
219.     "cscript ObjCMAudit.vbs /h:2 " & _
220.     "/d:03-07-2003 /t:16:00"
221. End Function
222. *****
223. '* Function: PadZero
224. *****
225. Function PadZero(dtValue)
226.   If Len(dtValue) = 1 Then
227.     PadZero = 0 & dtValue
228.   Else
229.     PadZero = dtValue
230.   End If
231. End Function
232. *****
233. '* Function: GenFileName
234. *****
235. Function GenFileName(prefix)
236.   'Create a unique time stamped name for the text file
237.   Dim dtDate,strYear,strMonth,strDay,strDate
```

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

```
238. Dim dtNow,strHour,strMinute,strSecond,strTime
239. dtDate = Date()
240. strYear = Mid(Year(dtDate),3)
241. strMonth = PadZero(Month(dtDate))
242. strDay = PadZero(Day(dtDate))
243. strDate = strYear & strMonth & strDay & "-"
244. dtNow = Now()
245. strHour = PadZero(Hour(dtNow))
246. strMinute = PadZero(Minute(dtNow))
247. strSecond = PadZero(Second(dtNow))
248. strTime = strHour & strMinute & strSecond
249. GenFileName = prefix & "-" & strDate & strTime & ".csv"
250. End Function
251. *****
252. '* Function: ReadAndWrite
253. *****
254. Function ReadAndWrite(RecordSet,Status,AttribValue)
255. Dim strDN,arrClass,strClassName
256. strDN = RecordSet.Fields("distinguishedName")
257. arrClass = RecordSet.Fields("objectClass")
258. strClassName = arrClass(UBound(arrClass))
259. ReadAndWrite = Chr(34) & strDN & Chr(34) & "," & strClassName & "," & _
260. Status & "," & AttribValue
261. End Function
262. Save the file as ObCMAudit.vbs
```

Modifying Policy Settings with Ntdsutil

This procedure allows an administrator to manually modify a policy setting. If recording policy settings is performed periodically, then any changes to policies can be quickly discovered and repaired.

Ntdsutil.exe is located in the Support tools folder on the Windows 2000 installation CD-ROM.

Requirements

- Credentials: Domain Admins
- Tools: Ntdsutil.exe

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

To view current policy settings on your domain controllers:

1. From a command line, type **Ntdsutil**
2. At the Ntdsutil command prompt, type **LDAP policies**
3. At the LDAP policy command prompt, type **set < setting > to < settingValue >**
4. At the server connection command prompt, type **connect to server < DomainControllerName >**
5. Modify current LDAP policy settings

For example; **Set MaxPoolThreads to 8**
6. To save the changes, type **Commit Changes**
7. When finished, type **q**

Note: This procedure only shows the Default Domain Policy settings. If you apply your own policy setting, you cannot see it.

Monitoring the Number of Objects In a Domain With ObjCountByClass.vbs

You can use the ObjCountByClass.vbs script to monitor the number of objects in a domain. Before you can use the ObjCountByClass.vbs script, you must create the script based on the source code included in this

document.<http://www.microsoft.com/technet/prodtechnol/ad/Windows2000/maintain/opsguide/Part2/ADOGdApB.asp>.

Note: This script requires Windows 2000 with Service Pack 3 or later and Windows Scripting Host version 5.6 or later.

The ObjCountByClass.vbs script has the following syntax:

```
cscript ObjCountByClass.vbs
```

ObjCountByClass.vbs has no parameters.

ObjCountByClass.vbs creates an output file, ObjCountByClass -*date-time*.csv, which contains the number of objects by object class (where *date* is the date you ran ObjCountByClass.vbs and *time* is the time you ran ObjCountByClass.vbs).

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

To create the ObjCountByClass.vbs script

1. Start Notepad
2. Copy or type the following script into Notepad
3. *****
4. '* File: ObjCountByClass.vbs *
5. '* Created: April 2003 *
6. '* Version: 1.0 *
7. '* *
8. '* Description: Diagnostic utility that counts the number of objects *
9. '* by objectClass, created in all containers *
10. '* defined in the namingContexts attribute of RootDSE. *
11. '* Use this tool to build an object creation trend *
12. '* analysis. Output is placed in a date and time-stamped *
13. '* CSV report file with a beginning prefix of *
14. '* ObjCountByClass. Import the csv file into a *
15. '* a spreadsheet program or a database program for *
16. '* trend analysis. *
17. '* *
18. '* Compatibility: This script requires WSH 5.6, CScript, ADSI, *
19. '* and access to Active Directory *
20. *****
21. Option Explicit
22. 'Define any constants used in this script
23. Const ForAppending = 2
24. 'Declare global variables
25. Dim dtDate,dtTime,strDate,strTime
26. Dim objRootDSE,strSchema,strDomain,strConfig
27. Dim objConnection,objCommand,objRSSchema
28. Dim objFSO,objFile,strCount,strFileName
29. Dim arrNamingContexts,NamingContext,strContainer
30. Call CheckForCScript
31. 'Set the date and time values. These are used by
32. 'the GenFileName function and appear in the first
33. 'and second column of the report.

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

```
34. dtDate = Date()
35. dtTime = Time()
36. 'Formatted date and time values for report columns 1 and 2
37. strDate = DateFormat(dtDate)
38. strTime = TimeFormat(dtTime)
39. 'Use the RootDSE object for upcoming binding operations
40. Set objRootDSE = GetObject("LDAP://rootDSE")
41. 'Get the DNs for all naming contexts defined on the DC
42. arrNamingContexts = objRootDSE.GetEx("namingContexts")
43. 'Bind to the current schema to obtain a list of all ClassSchema objects
44. strSchema = "<LDAP://" & _
45.   objRootDSE.Get("schemaNamingContext") & ">"
46. 'Create the ADSI OLE DB provider object
47. Set objConnection = CreateObject("ADODB.Connection")
48. objConnection.Open "Provider=ADsDSOObject;"
49. 'Create the command object and set the ActiveConnection
50. 'property equal to the command object
51. Set objCommand = CreateObject("ADODB.Command")
52. objCommand.ActiveConnection = objConnection
53. 'Create the schema query
54. objCommand.CommandText = strSchema & _
55.   ":(objectClass=classSchema);" & _
56.   "IDAPDisplayName,distinguishedName;subtree"
57. 'Specify page size for this command object.
58. 'This is necessary to avoid overutilization of server
59. 'and network resources. Also, by default,
60. 'only 1000 records will be returned if paging isn't
61. 'specified. The schema might contain more
62. 'than 1000 classSchema objects.
63. objCommand.Properties("Page Size") = 256
64. objCommand.Properties("Asynchronous") = True
65. 'Caching is required because the schema recordset
66. 'iterated multiple times.
67. objCommand.Properties("Cache results") = True
68. 'Run the schema query
69. Set objRSSchema = objCommand.Execute
```

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

```
70. 'Create a unique file name (timestamp) using the GenFileName function
71. strFileName = GenFileName("ObjCountByClass")
72. 'Create a text file (.csv format) to hold the
73. 'results of the class test.
74. Set objFSO = CreateObject("Scripting.FileSystemObject")
75. 'This will overwrite the file if it already exists.
76. Set objFile = objFSO.CreateTextFile(strFileName,True)
77. objFile.Close
78. Set objFile = objFSO.OpenTextFile(strFileName,ForAppending)
79. 'Write the column headings to the report file
80. objFile.WriteLine "dtDate,dtTime,strContainer,strObjectClass,intTotal"
81. 'Enumerate the list of Naming Contexts and write a multi-line
82. 'report file.
83. For Each NamingContext in arrNamingContexts
84.     'Set the strContainer variable equal to the binding string of the container.
85.     strContainer = "<LDAP:// " & NamingContext & ">"
86.     'Write the column headings
87.     Call CountObjects(NamingContext,strContainer)
88. Next
89. 'Clean up
90. Set objConnection = Nothing
91. 'Close the file
92. objFile.Close
93. WScript.Echo VbCrLf & "The report data has been saved to: " & strfileName & "."
94. WScript.Echo "Import or open the CSV data in a spreadsheet or database program."
95. '*****
96. '* Routine: CheckForCScript
97. '*****
98. Sub CheckForCScript
99.     'This script must run from cscript because
100.     'it uses the WScript.StdOut property.
101.     'Test the script host and if it's not cscript,
102.     'instruct the operator on how to run the script.
103.     If Right(LCase(WScript.FullName),11) <> LCase("cscript.exe") Then
104.         WScript.Echo "This script must run from cscript." & _
105.         VbCrLf & "Example: cscript objCountByClass.vbs"
```

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

```
106. WScript.Quit
107. End If
108. End Sub
109. *****
110. '* Routine: CountObjects
111. *****
112. Sub CountObjects(NamingContext,Container)
113. Dim i,strClassName,strDN,objRSContainer,intRecordCount,iCnt,iPercent
114. Dim objField,colObjClass,strClassValue
115. i = 0
116. 'Use a While Wend loop to test for each type of class object
117. WScript.Echo VbCrLf
118. While Not objRSSchema.EOF
119. 'Set the class and dn variables equal to the two attributes
120. 'contained in the Fields collection.
121. strClassName = objRSSchema.Fields("IDAPDisplayName")
122. strDN = objRSSchema.Fields("distinguishedName")
123. 'objectClass must be returned in the resultset so that the last entry
124. 'of the multi-valued attribute can be returned as the objectclass
125. 'in the report.
126. objCommand.CommandText = Container & _
127. ";(objectCategory=" & strDN & ");" & _
128. "objectClass;subtree"
129. 'Caching isn't necessary because this recordset is read only once.
130. objCommand.Properties("Cache results") = False
131. Set objRSContainer = objCommand.Execute
132. If i = 0 Then
133. WScript.Echo "Counting objects in:" & " " & NamingContext
134. End If
135. 'Get the class name from the recordset. The last entry is
136. 'all that's required in the objectClass multi-valued attribute
137. While Not objRSContainer.EOF
138. For Each objField in objRSContainer.Fields
139. colObjClass = objField.Value
140. For Each strClassValue in colObjClass
141. 'The last entry for strClassValue becomes the value
```

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

```
142.     'of strClassName
143.     strClassName = strClassValue
144.     Next
145.     Next
146.     intRecordCount = objRSContainer.RecordCount
147.     If intRecordCount > 0 Then
148.         objFile.WriteLine strDate & "," & strTime & "," & _
149.             Chr(34) & NamingContext & Chr(34) & "," & strClassName & "," & intRecordCount
150.     End If
151.     objRSContainer.MoveNext
152. Wend
153. 'Progress indicator
154. For iCnt = 1 to Len(iPercent) + 22
155.     WScript.StdOut.Write Chr(8)
156. Next
157. iPercent = (Round(i/objRSSchema.RecordCount,2) * 100) + 1
158. WScript.StdOut.Write "Percentage complete: " & iPercent & "%"
159. i = i + 1
160. objRSSchema.MoveNext
161. Wend
162. 'Clean up
163. Set objRSContainer = Nothing
164. objRSSchema.MoveFirst
165. End Sub
166. *****
167. '* Function: PadZero
168. *****
169. Function PadZero(dtValue)
170.     If Len(dtValue) = 1 Then
171.         PadZero = 0 & dtValue
172.     Else
173.         PadZero = dtValue
174.     End If
175. End Function
176. *****
177. '* Function: GenFileName
```

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

```
178. '* NOTE, This is slightly modified from the other GenFileName functions
179. '* the initial date and time values are calculated outside of the
180. '* function because the values are used in the first and second columns
181. '* of the report.
182. *****
183. Function GenFileName(prefix)
184. 'Create a unique time stamped name for the text file
185. Dim strYear,strMonth,strDay,strDate
186. Dim strHour,strMinute,strSecond,strTime
187. strYear = Mid(Year(dtDate),3)
188. strMonth = PadZero(Month(dtDate))
189. strDay = PadZero(Day(dtDate))
190. strDate = strYear & strMonth & strDay & "-"
191. strHour = PadZero(Hour(dtTime))
192. strMinute = PadZero(Minute(dtTime))
193. strSecond = PadZero(Second(dtTime))
194. strTime = strHour & strMinute & strSecond
195. GenFileName = prefix & "-" & strDate & strTime & ".csv"
196. End Function
197. *****
198. '* Function: DateFormat
199. *****
200. Function DateFormat(DateVal)
201. Dim strYear,strMonth,strDay
202. 'Get the date for column 1 of each row.
203. strYear = Year(DateVal)
204. strMonth = PadZero(Month(DateVal))
205. strDay = PadZero(Day(DateVal))
206. DateFormat = strYear & "/" & strMonth & "/" & strDay
207. End Function
208. *****
209. '* Function: TimeFormat
210. *****
211. Function TimeFormat(TimeVal)
212. Dim strHour,strMinute,strSecond
213. 'Get the time for column 2 of each row.
```

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

214. strHour = PadZero(Hour(TimeVal))
215. strMinute = PadZero(Minute(TimeVal))
216. strSecond = PadZero(Second(TimeVal))
217. TimeFormat = strHour & ":" & strMinute & ":" & strSecond
218. End Function
219. Save the file as ObjCountByClass.vbs

Removing an Active Directory Domain Controller

Use the NTDSUtil.exe Support tool to perform this procedure. Ntdsutil.exe is located in the Support tools folder on the Windows 2000 installation CD-ROM.

To remove a selected Active Directory domain controller:

1. Run the ntdsutil command and at the prompt, type **metadata cleanup**
2. This returns the metadata cleanup command prompt.
3. Type **connection**
This returns the connection command prompt.
4. At the connection command prompt, type **connect to server < ServerName>**
5. Type **quit**
This returns the metadata cleanup command prompt.
6. Type **select operation target**
This returns the operational target command prompt.
7. Type **list sites**
A numbered list of sites is displayed.
8. Type **select site < SiteNumber>**
9. Type **list domains in site**
A numbered list of domains is displayed.
10. Type **select domain < DomainNumber>**

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

11. Type **list servers in site**

A numbered list of servers for a domain in a site is displayed.

12. Type **select server < ServerNumber >**

13. At the select operation target command prompt, type **quit**

14. At the metadata cleanup command prompt, type **remove selected server**

Resetting Passwords

Perform this procedure when administrator passwords must be quickly reset, such as when a forest-wide password reset is required because passwords might have been compromised.

Requirements

- Credentials: Domain Admins or Enterprise Admins

To reset a password

1. Open Active Directory Users and Computers, and locate the service administrator accounts.
2. Right-click an account, and then click **Reset password**.
3. Type the new password in **New Password** and **Confirm New Password**, and then click **OK**.
4. Distribute the new password to the service administrator.
5. Repeat steps 2 through 4 for each service administrator account.

Securing Scripts with Script Signing

Two alternatives exist for creating signed scripts. For those interested in developing their own script host, the Windows Product SDK contains a set of tools for signing scripts (signcode.exe and chktrust.exe). When writing your own script host, call Win32 API WinVerifyTrust. This API verifies the trust on a .VBS or .JS file.

Alternatively, the Windows Script Host (WSH) 5.6 ships with a signer object to create and verify signed scripts. The following JScript® code creates a signed file.

```
var Signer = new ActiveXObject("Scripting.Signer");  
var File = "c:\\myfile.vbs";
```

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

```
var Cert = "Jane Q. Programmer";  
var Store = "my";  
Signer.SignFile(File, Cert, Store);
```

The following sample, in this case as VBScript code, verifies the signing on a file:

```
Dim Signer, File, ShowUI, FileOK  
Set Signer = CreateObject("Scripting.Signer")  
File = "c:\newfile.wsf"  
ShowUI = True  
FileOK = Signer.VerifyFile(File, ShowUI)  
If FileOK Then  
    WScript.Echo File & " is trusted."  
Else  
    WScript.Echo File & " is NOT trusted."  
End If
```

Viewing Current Policy Settings

This procedure allows an administrator view current LDAP policy settings. This procedure collects information that is required for the baseline database.

Ntdsutil.exe is located in the Support tools folder on the Windows 2000 installation CD-ROM.

Requirements

- Credentials: Domain Admins
- Tools: Ntdsutil.exe

Ntdsutil.exe is located in the Support tools folder on the Windows 2000 installation CD-ROM.

To view current policy settings on your domain controllers:

1. From a command line, type **Ntdsutil**
2. At the Ntdsutil command prompt, type **LDAP policies**
3. At the LDAP policy command prompt, type **connections**
4. At the server connection command prompt, type **connect to server < DomainControllerName>**

Best Practice Guide for Securing Active Directory Installations and Day to Day Operations Part II

By Kathleen Cole, Jennifer Bayer, Doug Steen

View current LDAP policy settings.

5. At the server connection command prompt, type **q**
6. At the LDAP policy command prompt, type **show values**

A display of the policies as they exist appears.

Acknowledgements

Developed by the Windows Server Content Group

Program Managers: Arun Nanda

Writers: Kathleen Cole, Jennifer Bayer, Doug Steen

Editor: Jim Becker

Lab Staff: Robert Thingwold, David Meyer

Lab Partners: Hewlett Packard Corporation and Cisco Systems, Inc.

We thank the following people for reviewing the guide and providing valuable feedback:

Eric Fitzgerald, Andy Harjanto, Smitha Vuppuluri, Jason Garms