SysAdmin Magazine

netwrix

№ 27

Active Directory

Tame the Beast

SysAdmin
Magazine

№ 27 August ‘17

SysAdmin Magazine is a free source of knowledge for IT Pros who are eager to keep a tight grip on network security and do the job faster.

The Sysadmin Magazine team
sysadmin.magazine@netwrix.com

Contents

Nirmal Sharma

MCSEx3, MCITP and Microsoft MVP
in Directory Services

5 How-tos for Active Directory Optimization Using PowerShell

Check these 5 how-tos to learn how to find information about such AD objects as inactive computers; users whose passwords never expire; AD site information and much more.

How-To #1

Find Inactive Computers in Active Directory

If you wish to collect stale computer accounts from Active Directory, you can always use the **Get-ADComputer** PowerShell cmdlet. As the name suggests, **Get-ADComputer** targets only computer accounts. **Get-ADComputer** does not provide any parameter that allows you to specifically collect stale computer accounts; however, it does feature a "-Filter" switch, which lets you specify a criterion.

To identify inactive computer accounts, you will always target those that have not logged on to Active Directory in the last last 90 days. To accomplish this goal, you need to target the LastLogonTimeStamp property and then specify a condition with the time as shown in the following PowerShell commands:

```
$DaysInactive = 90
$time = (Get-Date).Adddays(-($DaysInactive))
Get-ADComputer -Filter {LastLogonTimeStamp -lt $time}
-ResultPageSize 2000 -resultSetSize $null -Properties
Name, OperatingSystem, SamAccountName,
DistinguishedName
```

As you can see, the **$Time** variable holds a valid date, and the next PowerShell command is executed with a filter that is set to search only those computer accounts for which the **LastLogonTimeStamp** has not been updated in the last 90 days. If you wish to search computer accounts that have been inactive for more than 90 days, all you need to do is modify the **$DaysInActive** variable value. The current value is set at 90 days; however, you can specify your own value. To export output to a CSV file, add the **Export-CSV** PowerShell cmdlet as shown in the following command:

```
Get-ADComputer -Filter {LastLogonTimeStamp -lt $time}
-ResultPageSize 2000 -resultSetSize $null -Properties
Name, OperatingSystem, SamAccountName,
DistinguishedName | Export-CSV "C:\Temp\
StaleComps.CSV" –NoTypeInformation
```

How-To #2

Find Locked Out User Accounts in Active Directory

It isn't difficult to find locked-out user account information from Active Directory as long as you use PowerShell. The PowerShell cmdlet **Search-ADAccount** can provide you with a list of user accounts that have been locked out of the system, as is shown in the following PowerShell command:

```
Search-ADAccount -LockedOut -UsersOnly
-ResultPageSize 2000 -resultSetSize $null | Select-
Object Name, SamAccountName, DistinguishedName
| Export-CSV "C:\Temp\LockedOutUsers.CSV"
-NoTypeInformation
```

As you can see in the command above, the **Search-ADAccount** cmdlet supports specifying the **LockedOut** switch that only targets locked-out accounts. We have also specified the **UsersOnly** switch to ensure that the **Search-ADAccount** cmdlet targets only user accounts.

As part of the command, we are exporting output to a CSV file for easy tracking.

How-To #3

Get a List of Expired User Accounts in AD

One of the most important tasks that an Active Directory administrator performs is ensuring that expired user accounts are reported in a timely manner and that action is taken to immediately remove or disable them. Note that user accounts for which you set an expiration date are only created temporarily. For example, you might have created several user accounts to allow vendors to log on to the Active Directory. Similarly, you might have created user accounts for contractors. If you wish to see what accounts have expired, execute the following PowerShell command:

```
Search-ADAccount -Server $ThisDomain
-Credential $Creds -AccountExpired -UsersOnly
-ResultPageSize 2000 -resultSetSize $null|
Select-Object Name, SamAccountName,
DistinguishedName
```

Note the use of the **Search-ADAccount** PowerShell cmdlet again but with a different switch this time. The switch that we use is **AccountExpired**. As the name suggests, the **AccountExpired** switch helps you to collect user accounts that have expired.

How-To #4

Get a List of AD Users Whose Passwords Never Expire

You might have created user accounts for which the passwords never expire. For example, you would always set the Password Never Expire attribute for user accounts that are utilized as service accounts, but you need to make sure that unwanted user accounts do not have the Password Never Expire attribute set. This is because, per security standards, every user is required to change his/her password within a certain time frame. For most organizations, every user is required to change his/her password within 90 days. You can use the following PowerShell commands and script to get a list of Active Directory users whose passwords never expire:

```
Search-ADAccount -PasswordNeverExpires -UsersOnly
-ResultPageSize 2000 -resultSetSize $null | Select-
Object Name, SamAccountName, DistinguishedName
| Export-CSV "C:\Temp\PassNeverExpiresUsers.CSV"
-NoTypeInformation
```

As you can see in the PowerShell command above, we use the PasswordNeverExpires switch that helps us query such users from Active Directory; the output is stored in the "C:\Temp\PassNeverExpiresUsers.CSV" file. If you wish to collect the same information from multiple Active Directory domains, you will use the PowerShell script that is describes later in this section. Please make sure to execute the script from a Windows Server 2012 or later operating system. The Active Directory forest name that is currently being used by the script is "Netwrix.Com." You must change the Active Directory forest name in the $CurForestName variable before executing the script. Make sure to create a directory by naming it "C:\Temp" on the local computer. You must also utilize a user account that has permission to access all Active Directory domains. The only permission that you require is a user account with Read only permissions in the destination domain.

```
$DomList = "C:\Temp\DomList.TXT"
remove-item $DomList -ErrorAction SilentlyContinue
$CurForestName="NWBlog.Com"
$GetForest=Get-ADForest $CurForestName
$Items = $R.Domains
ForEach ($Domains in $Items)
{
   Add-Content $DomList $Domain.Name
}
Write-Host "Starting Script..."
ForEach ($DomInFile in $DomList)
{
   $DisabledCompsCSV = "C:\Temp\DisabledAccounts_
Computers_"+$DomInFile+".CSV"
   Remove-item $DisabledCompsCSV -ErrorAction
SilentlyContinue
   $DisabledUsersCSV = "C:\Temp\DisabledAccounts_
Users_"+$DomInFile+".CSV"
   Remove-item $DisabledUsersCSV -ErrorAction
```

```
SilentlyContinue
    $InActiveUsersReport = "C:\Temp\
InactiveUsers_"+$DomInFile+".CSV"
    Remove-item $InActiveUsersReport -ErrorAction
SilentlyContinue

Get-ADComputer -Server $DomInFile –Filter {(Enabled
–eq $False)} –ResultPageSize 2000 –ResultSetSize
$null -Properties Name, OperatingSystem | Export-CSV
$DisabledCompsCSV -NoTypeInformation
    Search-ADAccount -Server $DomInFile –
AccountDisabled –UsersOnly –ResultPageSize 2000
–ResultSetSize $null | Select-Object SamAccountName,
DistinguishedName | Export-CSV $DisabledUsersCSV –
NoTypeInformation
    Search-ADAccount -Server $DomInFile –AccountInActive
–TimeSpan 90:00:00:00 –ResultPageSize 2000 –
ResultSetSize $null | ?{$_.Enabled –eq $True} | Select-
Object Name, SamAccountName, DistinguishedName |
Export-CSV $InActiveUsersReport –NoTypeInformation

}
Write-Host "Script Finished collecting required information.
Please check report files under C:\Temp folder"
```

How-To #5

Collect AD Site Information

PowerShell can help you when you wish to collect AD site information such as Active Directory site location, site options configured, ISTG assigned to the site, Site links and bridgehead servers. The PowerShell Script below uses the New-Object function to connect to Directory Context and then obtain all AD sites. All AD sites are stored in a variable. The script uses the ForEach loop to traverse through each site stored in the variable; it then appends the site information in a CSV file.

Once the PowerShell script mentioned below has been executed, you will have information for each Active Directory site stored in the "C:\Temp\ADSiteInfo.CSV" file. Note that the script connects to the current Active Directory Forest by default. If you wish to change the report file location, please modify the $ReportFile variable.

```
$ReportFile = "C:\Temp\ADSiteInfo.CSV"
Remove-item $ReportFile -ErrorAction SilentlyContinue
$ThisString="AD Site,Location,Site Option,Current IST-
G,Subnets,Servers,In Site Links,Bridgehead Servers"
Add-Content "$ReportFile" $ThisString

$CurForestName = "NetWrix.com"
$a = new-object System.DirectoryServices.ActiveDirectory.
DirectoryContext("Forest", $CurForestName)
[array]$ADSites=[System.DirectoryServices.ActiveDirecto-
ry.Forest]::GetForest($a).sites
$ADSites
ForEach ($Site in $ADSites)
{
    $SiteName = $Site.Name
    $SiteLocation = $site.Location
    $SiteOption = $Site.Options
    $SiteISTG = $Site.InterSiteTopologyGenerator

    [array] $SiteServers = $Site.Servers.Count
    [array] $SiteSubnets = $Site.Subnets.Count
    [array] $SiteLinks = $Site.SiteLinks.Count
    [array] $SiteBH = $Site.BridgeheadServers.Count

    $FinalVal=$SiteName+","+'"'+$SiteLoca-
```

[Infographics]

Top Cybersecurity Risks in the Asia Pacific Region

Netwrix conducted its 2017 IT Risks Survey to learn more about the security, compliance and operational issues that bother organizations worldwide. As part of the survey, we've extracted some pretty interesting findings about Asian-Pacific organizations based on feedback provided by IT specialists working in various industries in the Asia Pacific region.

Read Full Report

IT RISKS in Asia 2017

netwrix

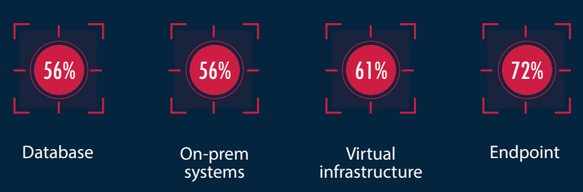
CONTEXT

85% Of organizations do not use any software for information security governance or risk management

60% Of IT operations teams are at least partially responsible for security

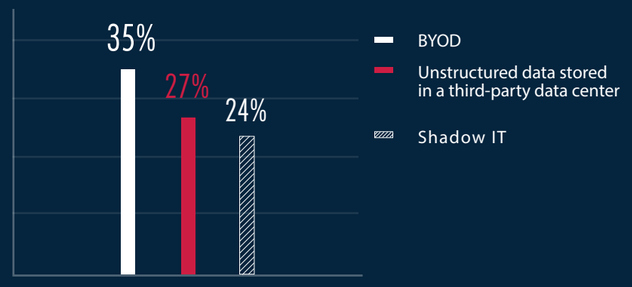
55% Of organizations do not have a separate information security function

MAIN SECURITY FOCUS



Database | On-prem systems | Virtual infrastructure | Endpoint

MOST NEGLECTED AREAS



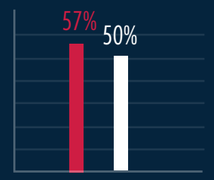
BIGGEST SECURITY RISKS

43% Employees

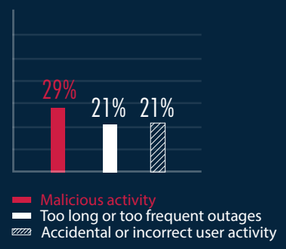
57% Hackers

64% of organizations had compliance issues in 2016

MAIN CAUSES OF SECURITY INCIDENTS IN 2016



MAIN CAUSES OF SYSTEM DOWNTIME IN 2016



WHERE VISIBILITY INTO USER ACTIVITY IS MOST NEEDED

ONLY **14%** of organizations are well prepared to beat cyber risks

WHY?

57% Complexity of IT infrastructure

57% Lack of budget

57% Insufficient staff training

FOCUS OF FUTURE INVESTMENTS

43% Data breach

29% Cyber-sabotage

29% Fraud

WHY ORGANIZATIONS NEED VISIBILITY

71% To detect and mitigate human factor (errors, misuse, etc.)

71% To investigate security incidents

64% To ensure the security of network and information assets

64% To maximize system uptime

Larry Glusman

27 years in IT

Office 365 Tutorial

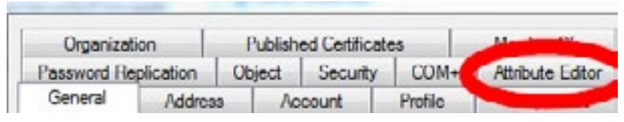
Using Active Directory to Add an Alias to an Office 365 Email Account

If you are using Office 365 with Azure AD Connect (or the older DirSync) you know that some changes to accounts cannot be made via the O365 admin portal. For instance, if someone gets married and changes their name, you may wish to add a new email address for them. If you try to add an alias (second email address) to an account, you will get an error similar to this:

error

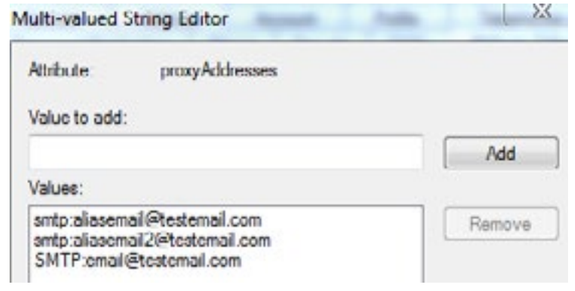
The operation on mailbox " [redacted] " failed because it's out of the current user's write scope. The action 'Set-Mailbox', 'EmailAddresses', can be performed on the object ' [redacted] ' because the object is being synchronized from your on-premises organization. This action should be performed on the object in your on-premises organization.

This error has made many people think they need to keep an Exchange Server up and running on their local network. Thankfully, that's not the case. You can easily add an alias via Active Directory Users and Computers (ADUC). To do this, open ADUC and find the User you want to modify. Make sure that Advanced Features is checked, under View on the top menu. Double click on the User then click on the Attribute Editor tab.



Scroll down to the Proxy Address field and double click to open it for editing. It may be blank, which is fine, or it may already have some information in it. If it's blank your first step is to add the existing email account in the format **SMTP:email@testemail.com**. Make sure to capitalize SMTP as that's how the default account is determined. For the alias account you want to add, use the format: **smtp:aliasemail@testemail.com**. You can add as many aliases as needed, just be sure that they all use lower case for smtp.

After entering the information, it should look something like this:



When done click OK until you are out of ADUC and then sit back and be patient. The cloud side will synchronize and show the new alias, but it isn't always fast. You can do a manual sync via Azure AD Connect / DirSync, but even then it can take some time to appear on the O365 side of things.

Russell Smith

Security Expert, IT consultant

Active Directory Audit

3 Ways to Secure Service Accounts in Active Directory

If you're looking for security weak spots in your organization, auditing service accounts isn't a bad place to start. Active Directory audit should include establishing the rights assigned to each account, the password strength, the last time it was reset, and whether it is a domain account, local account, Managed Service Account (MSA), or Group Managed Service Account (gMSA).

Services can be configured to run using the Local Service, Network Service, or Local System accounts, but these are shared with other processes and applications, and they can't be managed centrally. Although Local System has wide-ranging access to Windows, Local Service and Network Service are standard users with the Logon as a service NT right. Network Service is additionally able to connect to other devices on the network using the device's computer account.

When domain user accounts are used as service accounts, they provide a greater level of isolation and management, but passwords need to be managed manually or using a custom solution—or, as is often the case, passwords are rarely, if ever, reset.

Group Managed Service Accounts

There are two special types of accounts that can be used for services: MSAs and gMSAs. If you remember back to Windows Server 2008 R2, Microsoft introduced MSAs, but they came with so many caveats that the potential use cases were limited. The basic premise of MSAs was to provide the management and isolation of a domain user account but with automated password management. MSAs also provide simplified service principal name (SPN) management.

Microsoft improved MSAs in Windows 2012 with gMSAs. Unlike MSAs, gMSAs can be used on more than one device, run scheduled tasks, and work with applications such as IIS and Exchange. The Key Distribution Service (KDS) on Windows Server

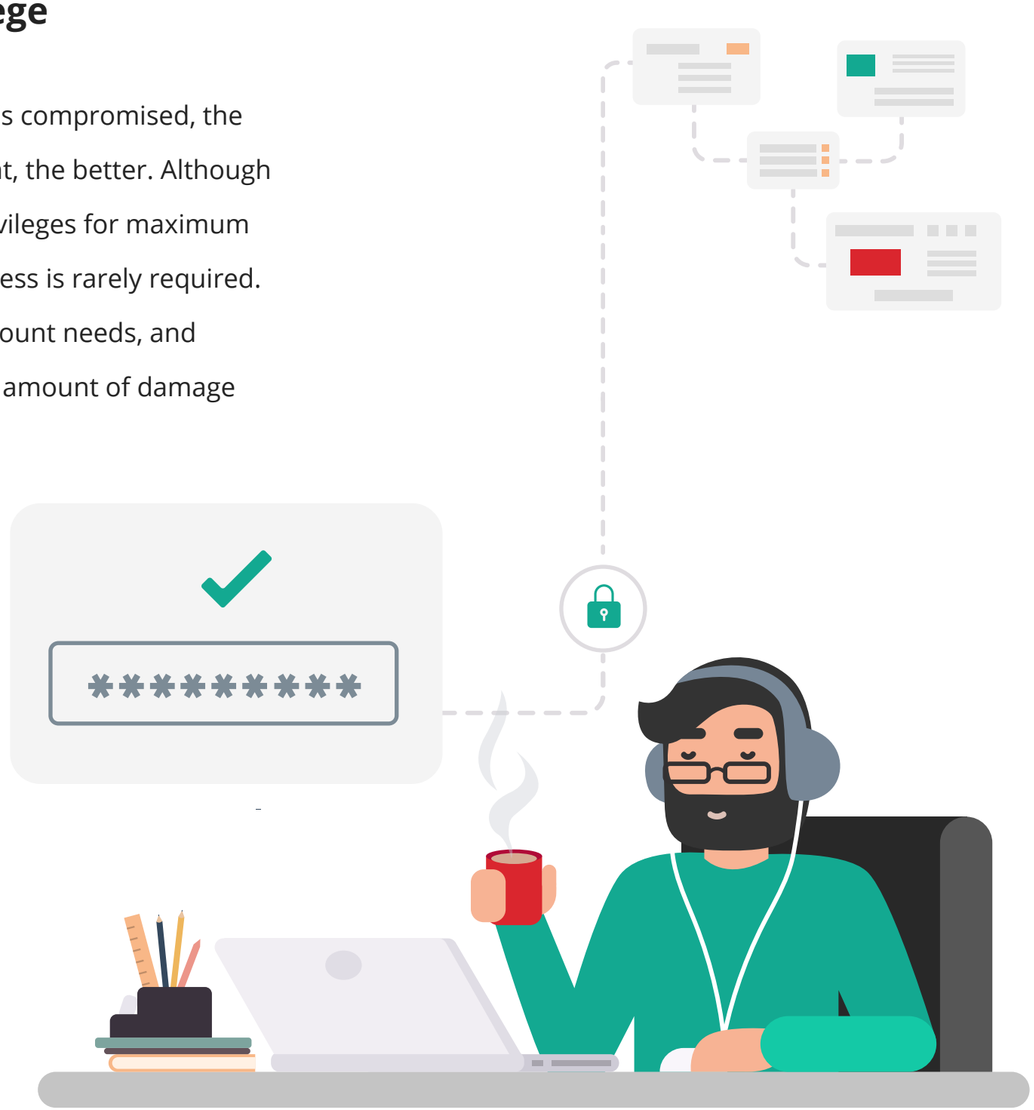
2012 domain controllers manages the 120-character password assigned to each gMSA. Before you can use gMSAs, you must have at least one Windows Server 2012 (or a higher domain) controller in your domain, as well as domain-joined devices running Windows Server 2012 or Windows 8 (and later). PowerShell is used to configure gMSAs.

Long Passwords

Not all applications are compatible with gMSAs, so sometimes a domain user account is the best option. Microsoft recommends passwords of at least 25 characters for service accounts, and a process for changing service account passwords should also be implemented. Additionally, place service accounts in a dedicated organizational unit (OU) in AD so that they can be managed separately from other accounts, and make sure that a strong password policy is applied to them.

Principle of Least Privilege

In the event that a service account is compromised, the fewer rights assigned to the account, the better. Although it's easy to assign administrator privileges for maximum compatibility, such far-reaching access is rarely required. Find out what rights the service account needs, and assign only those rights to limit the amount of damage that can be done.



Add Sensitive User Accounts to the Active Directory Protected Users Group

Microsoft introduced the Protected Users group in Windows Server 2012 R2 and Windows 8.1, and it's designed to harden accounts that are group members, in particular to protect against pass-the-hash attacks by disabling the use of NT LAN Manager (NTLM), a legacy authentication protocol that's still present in Windows for backwards compatibility.

The extra protections are only provided when users log in to Windows Server 2012 R2 or Windows 8.1, and the full list of defenses requires the domain functional level to be set to Windows Server 2012 R2 (or higher).

Protected Users is primarily intended for use with domain and enterprise administrator accounts, which are particularly susceptible to attack, as when compromised they provide wide-open access to systems. That's not to say that other user accounts, that might be considered a target, can't be added to Protected Users. But because of the tight restrictions placed on members of Protected Users, it's essential to perform thorough testing beforehand.

> **Always make sure that you keep at least one highly-privileged account outside of Protected Users group**

There are no workarounds for the restrictions, so it's wise to make sure that if you intend to add highly-privileged accounts, like domain and enterprise administrators, that at least one that's not used for regular administration tasks, is left outside of the group.

The following protections are enabled for members of the Protected Users group when logging in from a supported device:

- Cached credentials are blocked. A domain controller must be available to authenticate the user
- Long-term Kerberos keys cannot be used to log in
- Plaintext passwords are not cached for Windows Digest authentication or default credential delegation (CredSSP), even when the relevant policies are enabled
- NTLM one-way function (NTOWF) is blocked

When DES and RC4 are blocked, all domain controllers (DC) must be running Windows Server 2008 (or later), and before adding users to Protected Users, their passwords should be changed against a Windows Server 2008 DC so that the AES keys are stored in Active Directory.

I Can not See the Protected Users Group in my Domain

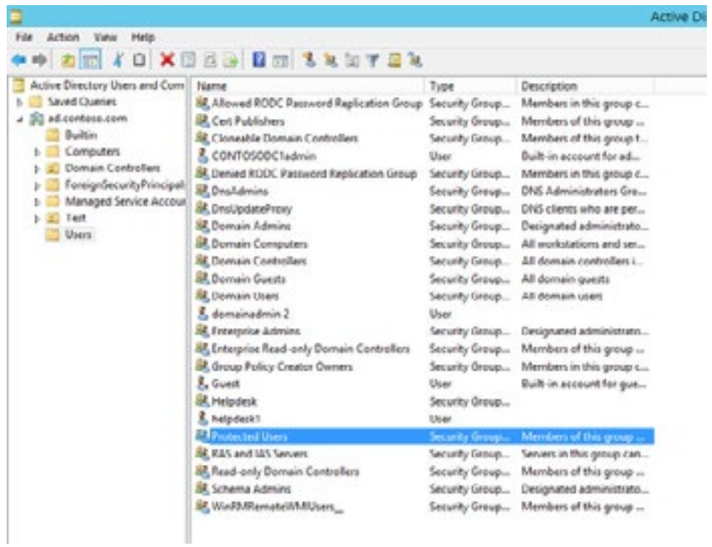
I Can't See the Protected Users Group in my Domain
If you don't have the Protected Users group in your domain, you will need to make sure there is at least one Windows Server 2012 R2 DC, and transfer the Primary Domain Controller (PDC) Emulator Flexible Single Master Operation (FSMO) role to a Windows Server 2012 R2 DC. If necessary, the PDC role can be transferred back to its original location.

To check if you have the Protected Users group in your domain, log in to Windows Server 2012 R2 as a domain administrator:

- Open Server Manager from the Start screen
- Select Active Directory Users and Computers from the Tools
- In the left pane, expand your domain and click Users

If Protected Users is present in the domain, you should see it on the right. Users can be added to Protected Users, as you would add them to any AD group. Using PowerShell for example, to add the admin1 user account:

```
Add-ADGroupMember –Identity 'Protected Users' –Members admin1
```



To transfer the PDC emulator FSMO role to a Windows Server 2012 R2 DC, log in to the DC using a domain administrator account, and open a PowerShell prompt using the icon on the desktop taskbar. Now type the command below and press ENTER, replacing DC6 with the name of the Windows Server 2012 R2 DC:

```
Move-AdDirectoryServerOperationMasterRole -Identity DC6 –OperationMasterRole pdcemulator
```

If you want to confirm the new or original location of the PDC emulator FSMO role, run the command below, replacing ad.contoso.com with the name of your AD domain:

```
Get-AdDomain ad.contoso.com | Format-List pdcemulator
```

Free Tool of the Month

Netwrix Auditor for Active Directory

Download Free Tool

The free edition of Netwrix Auditor for Active Directory provides visibility into what's happening inside your domain by tracking logons and all changes to AD users, groups, organizational units, GPO links and various policies. Daily activity summaries sent by this free Active Directory software detail every change and logon that happened during the last 24 hours, including the before and after values for each modification.

Report example

Netwrix Auditor for Active Directory

Activity Summary

Removed	1
Modified	2

Action	Object type	What	Item	Where	When	Workstation
Modified	Group	\com\Enterprise\Builtin\ Remote Desktop Users	enterprise.com	dc3.enterprise.com	4/14/2017 4:59:19 AM	atl-mkt021.enterprise.com
Security Local Group Member Added: "\com\Enterprise\Users\Bill Hops"						
Modified	User	\com\Enterprise\Users\ Guest	enterprise.com	dc3.enterprise.com	4/14/2017 4:59:30 AM	atl-mkt021.enterprise.com
User Account Enabled						
Removed	User	\com\Enterprise\Users\ Diana Abraham	enterprise.com	dc3.enterprise.com	4/14/2017 4:59:40 AM	atl-mkt021.enterprise.com

This message was sent by Netwrix Auditor from **au-srv-fin.enterprise.com.**

Try Netwrix Auditor 9.0:

Shield Your IT Environment from Ransomware & Malicious Insiders

netwrix.com/auditor9.html

Corporate Headquarters:
300 Spectrum Center Drive,
Suite 200 Irvine, CA 92618

Phone: 1-949-407-5125
Toll-free: 888-638-9749
EMEA: +44 (0) 203-318-02