

[PowerShell - Listing aged out computer accounts in Active Directory](#)

✘ Do you remember having to use OldCmp.exe from JoeWare (great tools by the way) to capture a list of computer accounts which have not been attached to your Active Directory for a period of time. Luckily with PowerShell and Quest ActiveRoles you can do this with a simple two line script.

Here's the code: <http://gist.github.com/492494>



Find Old Computers In Active Directory

[PowerShell - Get serial numbers for computers in Active Directory](#)

✘ There are a lot of posts about pulling data from a file to do actions against computers/users. While this is valuable, I prefer to do dynamic capturing of computer and user objects directly from my Active Directory.

This is a simple script which will capture a list of computers from Active Directory and subsequently pull the serial number using the **Get-WMIObject** CmdLet using the **Win32_BIOS** class.

The neat addition over many other scripts I have used is the addition of the **-ErrorAction SilentlyContinue** option which prevents offline computers from sending error information in the output. Add some error handling and you now have a list of computers which are offline or inaccessible.

As with many of the scripts I will show you, you need to install ActiveRoles CmdLets from Quest Software (follow the link in the Gist)

Here's the code: <http://gist.github.com/492475>

```
Get-ADComputer -searchroot "OU=computers,DC=example,DC=com" -SizeLimit 0 |
ForEach-Object {
    $hostname = $_.Name
    $serialnumber = (Get-WMIObject Win32_BIOS -computer $hostname -ErrorAction
    SilentlyContinue).SerialNumber
    if ([not $serialnumber]) {
        Add-Content d:\logfiles\offlinelists.txt "$hostname is in offline"
    }
    else {
        Write-Host "$hostname,$serialnumber" > output.txt
        Add-Content d:\logfiles\serialnumberlist.txt "$hostname,$serialnumber"
    }
}
```

List of serial numbers from Active Directory computers

PowerShell bundle for TextMate

I've just installed this bundle for TextMate and it is nothing short of awesome. TextMate = happiness. ☹

<http://github.com/caisson/powershell.tmbundle>

With these powers I can become invincible!!!*

Now if only I was a proper programmer I could really work some magic with these tools ☹

* **TextMate bundles do not actually make you invincible**

PowerShell - Active Directory - Email users with password expiry < 14 days

PowerShell - Active Directory - Email users with password expiry under 14 days ☹

This script is handy for users who are not typically on your network (remote access, email only etc...). Normally users are warned at 14 days before expiry unless they are not logging into Active Directory through the workstation in which case they will not receive warnings until the password expires.

Simply set this to run daily and it will locate and email anyone who has a password expiry less than 14 days. To change the criteria, simply modify the If statement.

You need to install the ActiveRoles CmdLets from Quest Software (follow the link in the Gist)

Here's the code: <http://gist.github.com/469656>

```

EmailADPasswordExpiryUnder14days.ps1
# Requires ActiveRoles CmdLets from Quest Software (They're free and awesome) -
• http://www.quest.com/powershell/activeroles-server.aspx
Get-QADUser -SizeLimit 0 | Select-Object samAccountName,mail,PasswordStatus |
Where-Object {$_.PasswordStatus -ne "Password never expires" -and $_.PasswordStatus -ne "Expired" -and
• $_.PasswordStatus -ne "User must change password at next logon." -and $_.mail -ne $null} |
ForEach-Object {
    $today = Get-Date
    $logdate = Get-Date -format yyyyMMdd
    $samaccountname = $_.samAccountName
    $mail = $_.mail
    $passwordstatus = $_.PasswordStatus
    $passwordexpiry = $passwordstatus.Replace("Expires at: ","")
    $passwordexpirydate = Get-Date $passwordexpiry
    $daystoexpiry = ($passwordexpirydate - $today).Days
    if ($daystoexpiry -lt 14) {
        $emailFrom = "PasswordManagement@yourdomain.com"
        $emailTo = "$mail"
        $subject = "Your password will expire in $daystoexpiry days"
        $body = "Please change your password to prevent loss of access to your corporate systems`n`n"
        $body += "If you are unable to change your password, please contact the help desk"
        $attachment = "C:\batch\docs>PasswordChange.doc" # Attach a HowTo document for changing passwords
        $smtpserver = "smtp.yourdomain.com"
        Send-MailMessage -To $emailTo -From $emailFrom -Subject $subject -Body $body -Attachments $attachment
    • -SmtpServer $smtpserver
        Write-Host "Email was sent to $mail on $today"
        Add-Content c:\batch\logs\maillog$logdate.txt "Email was sent to $mail on $today"
    }
}
Send-MailMessage -To "administrator@yourdomain.com" -From "PasswordManagementProcess@yourdomain.com"
• -Subject "Password change log for $today" -Body "This is the log from $today" -Attachments
• "d:\batch\logs\maillog$logdate.txt" -SmtpServer $smtpserver

```

Email Password Expiry Under 14 days