```powershell
##############################################################################
# This script can (depending on what is commented out):
#    -create a file in 'C:\temp\Inactive_Windows_Computers' of all inactive (longer than 'daysInactive') enabled Windows computer accounts
#    -disable inactive (longer than 'daysInactive') enabled Windows computer accounts
#    -move inactive (longer than 'daysInactive') disabled Windows computer accounts to a specified OU
#    -delete inactive (longer than 'daysInactive') disabled Windows computer accounts
# WARNING: Be mindful of computer accounts that are in AD, but whose corresponding physical or virtual entities are incapable of
#                     logging in, say for instance, a non Domain member appliance (like a firewall)
# This script can easily be modified to handle user accounts instead of computer accounts.
##############################################################################
# For more, visit https://www.DanielLBenway.net
#-----------------------------------------------------------------------------
# http://blogs.technet.com/b/askds/archive/2009/04/15/the-lastlogontimestamp-attribute-what-it-was-designed-for-and-how-it-works.aspx
#-----------------------------------------------------------------------------
# With Domain Functional Level 2003 and higher there is LastLogonTimeStamp.
# Before DFL 2003 there was just LastLogon.
# Administrators can use LLTS to determine if a user account or a computer account has recently logged on.
# LLTS is not for real-time data, but is just accurate enough to find inactive user accounts and inactive computer accounts.
# LLTS has a default accuracy of 9-14 days to minimize replication traffic.
# LLTS default accuracy is sufficient to find users and computers that haven't logged on in more than 30 days.
# You can tighten LLTS down to 5 days by setting 'ms-DS-Logon-Time-Sync-Interval' to '5'.
# LLTS is replicated to every Domain Controller.
# LL is not replicated to every DC, so you'd have to query every DC in your domain.
# PwdLastSet is for user or computer account password, which for computer accounts is changed by default every 30 days by the Windows OS.
# PLS is replicated to all DCs.
# You should disable inactive (longer than 'daysInactive') enabled Windows user and computer accounts for a while (as a test) before deleting them.
# Be sure your AD Recycle Bin is enabled, your AD Tombstone Interval is set properly, and you've enforced Strict Replication Consistency on all DCs before you delete any
user or computer accounts.
##############################################################################
Throw "Do you really want to run the ENTIRE script?"     # This line prevents accidental usage.
#-----------------------------------------------------------------------------
cls
Import-Module ActiveDirectory
IF ((Test-Path "C:\temp") -eq $False) {MkDir "C:\temp"}
IF ((Test-Path "C:\temp\Inactive_Windows_Computers") -eq $False) {MkDir "C:\temp\Inactive_Windows_Computers"}
$now = Get-Date -format "yyyy.MM.dd.HHmm.ss"
$filePathAndName = "C:\temp\Inactive_Windows_Computers\" + $now + "-Inactive_Windows_Computer_Accounts.TSV"
$daysInactive = 90     # 45 or 90 days is a good choice here.
$referenceDate = (Get-Date).AddDays(-($daysInactive))
"Reference date: " + $referenceDate | Out-File -FilePath $filePathAndName -Encoding ASCII -Append
##############################################################################




##############################################################################
# This section of code writes a file of inactive (longer than 'daysInactive') Windows computer accounts, be they enabled or disabled.
# You should run this section and manually examine the file contents before running the other sections of this script.
# You should disable inactive (longer than 'daysInactive') enabled Windows computer accounts for a while (as a test) before deleting them.
# Be sure your AD Recycle Bin is enabled, your AD Tombstone Interval is set properly, and you've enforced Strict Replication Consistency on all DCs before you delete any
computer accounts.
#-----------------------------------------------------------------------------
$headerLine = "Windows computer" + [char]09 + "OS" + [char]09 + "status" + [char]09 + "LastLogonTimeStamp" + [char]09 + "PwdLastSet"
$headerLine | Out-File -FilePath $filePathAndName -Encoding ASCII -Append
Write-Host "Getting Windows computer objects from AD…"
$inactiveWindowsComputers = Get-ADComputer -Filter {(OperatingSystem -Like "*Windows*") -and (LastLogonTimeStamp -lt $referenceDate) -and (PwdLastSet -lt $referenceDate)}
-Properties Name, OperatingSystem, LastLogonTimeStamp, PwdLastSet, enabled
Write-Host "Processing inactive Windows computer objects from AD, and writing output file…"
ForEach ($inactiveWindowsComputer in $inactiveWindowsComputers)
    {
    $convertedLastLogonTimeStamp = [DateTime]::FromFileTime($inactiveWindowsComputer.LastLogonTimeStamp)
    $convertedPwdLastSet = [DateTime]::FromFileTime($inactiveWindowsComputer.PwdLastSet)
    If ($inactiveWindowsComputer.Enabled -eq $True) {$accountStatus = "enabled"}
    Else {$accountStatus = "disabled"}
    $outputLine = $inactiveWindowsComputer.Name + [char]09 + $inactiveWindowsComputer.OperatingSystem + [char]09 + $accountStatus + [char]09 +
$ConvertedLastLogonTimeStamp.ToString('yyyy.MM.dd') + [char]09 + $ConvertedPwdLastSet.ToString('yyyy.MM.dd')
    $outputLine | Out-File -FilePath $filePathAndName -Encoding ASCII -Append
```

```
    }
Write-Host "Inactive Windows computer account output file is complete."
################################################################################




################################################################################
# This section of code disables inactive (longer than 'daysInactive') enabled Windows computer accounts.
# You should disable inactive (longer than 'daysInactive') enabled Windows computer accounts for a while (as a test) before deleting them.
# Be sure your AD Recycle Bin is enabled, your AD Tombstone Interval is set properly, and you've enforced Strict Replication Consistency on all DCs before you delete any
computer accounts.
#-------------------------------------------------------------------------------
# Write-Host "Getting and disabling inactive enabled Windows computer accounts from AD…"
# Get-ADComputer -Filter {(OperatingSystem -Like "*Windows*") -and (LastLogonTimeStamp -lt $referenceDate) -and (PwdLastSet -lt $referenceDate) -and (enabled -eq $True)} -
Properties OperatingSystem, LastLogonTimeStamp, PwdLastSet, enabled |
#      Set-ADComputer -Enabled $False
# Write-Host "Inactive enabled Windows computer account disabling is complete."
################################################################################




################################################################################
# This section of code moves inactive (longer than 'daysInactive') disabled Windows computer accounts to a specified OU (recently active disabled windows computer accounts
are not moved).
# You will need to edit the OU path below.
# You should disable inactive (longer than 'daysInactive') enabled Windows computer accounts for a while (as a test) before deleting them.
# Be sure your AD Recycle Bin is enabled, your AD Tombstone Interval is set properly, and you've enforced Strict Replication Consistency on all DCs before you delete any
computer accounts.
#-------------------------------------------------------------------------------
# Write-Host "Getting and moving inactive disabled Windows computer accounts to their new OU…"
# Get-ADComputer -Filter {(OperatingSystem -Like "*Windows*") -and (LastLogonTimeStamp -lt $referenceDate) -and (PwdLastSet -lt $referenceDate) -and (enabled -eq $False)}
-Properties OperatingSystem, LastLogonTimeStamp, PwdLastSet, enabled |
#      Move-ADObject -TargetPath "OU=DisabledComputers,DC=Domain,DC=com"
# Write-Host "Inactive disabled Windows computer account moving is complete."
################################################################################




################################################################################
# This section of code deletes inactive (longer than 'daysInactive') disabled Windows computer accounts (recently active disabled Windows computer accounts are not
deleted).
# You should disable inactive (longer than 'daysInactive') enabled Windows computer accounts for a while (as a test) before deleting them.
# Be sure your AD Recycle Bin is enabled, your AD Tombstone Interval is set properly, and you've enforced Strict Replication Consistency on all DCs before you delete any
computer accounts.
#-------------------------------------------------------------------------------
# Write-Host "Getting and deleting inactive disabled Windows computer accounts from AD…"
# Get-ADComputer -Filter {(OperatingSystem -Like "*Windows*") -and (LastLogonTimeStamp -lt $referenceDate) -and (PwdLastSet -lt $referenceDate) -and (enabled -eq $False)}
-Properties OperatingSystem, LastLogonTimeStamp, PwdLastSet, enabled |
#     Remove-ADComputer
# Write-Host "Inactive disabled Windows computer account deleting is complete."
################################################################################
```