

```

#Requires -Version 3.0
#Requires -Modules ActiveDirectory,GroupPolicy
$HTMLCSS="@
<style>
table {
    # border-collapse: collapse;
    border-width: 1px; border-style: solid; border-color: black; border-collapse:
collapse;
}
h2 {text-align:center}
th, td {
    #padding: 8px;
    #text-align: left;
    #border-bottom: 1px solid #ddd;
    border-width: 1px; padding: 3px; border-style: solid; border-color: black;
}
tr:hover{background-color:#f5f5f5}
</style>
"@

#Variable initialization
#$domainName = All forest domains
#$ReportPath = Path where script datas will be stored
#$date = Current date to format text and be sure they are unique
#$dnsfilename is dns file name for dns configuration
#$usersCSV is file name for exported users in each domain
#$computerCSV is file name for exported computers in each domains
#$logfile name is file name for file contains script log for troubleshoot.
$domainName=(get-adforest).domains
$ReportPath = "c:\temp\audit"
$date = (get-date -Format "dd_MM_yyyy_HH_mm")
$htmlReportFileName = "ADReport_{$date}.html"
$DNSFileName = "Dns_{$date}.txt"
$usersCSV = "users_{$date}.csv"
$computersCSV = "computers_{$date}.csv"
$logFileName = "AuditADps1_{$date}.log"

#Give time source from all DCs
function Get-TimeSource {
    Write-EZLog -Category INF -Message "### Start Get-TimeSource ###"
    Invoke-Command -ComputerName (Get-ADDomainController -Filter {domain -eq
$domain}) -Server $domain hostname -ScriptBlock {
        [PSCustomObject]@{
            Server = $env:COMPUTERNAME
            TimeSource = w32tm /query /source
        }
    } | Select-Object Server,TimeSource | Sort-Object server
    Write-EZLog -Category INF -Message "### End Get-TimeSource ###"
}

function Get-LastUpdateDate{
    param (
        [string]$domain
    )
    Write-EZLog -Category INF -Message "### Start Get-LastUpdateDate ###"

    $query= Invoke-Command -ComputerName (Get-ADDomainController -Filter {domain -
eq $domain}) -Server $domain hostname -ScriptBlock {
        $objects= get-hotfix | Select-Object @{N = "KB"; E = {($_.properties |
where-Object {$_.name -eq "hotfixid"}).value}},
        @{N = "Date"; E = {[datetime]($_.properties | where-Object {$_.name -eq
"installedon"}).value}}
        $objects | Sort-Object Date -Descending | Select-Object -First 1
    }
    $query | Select-Object pscomputername,KB,date | Sort-Object pscomputername
    Write-EZLog -Category INF -Message "### End Get-LastUpdateDate ###"
}

#Provide all roles installed in the AD domain servers
#The parameterer -computerName doesn't exist with get-windowsfeature with
powershell3.
#For this reason, invoke-command is used here.
function Get-RoleServer
{

```

```

param (
    [string]$domain
)
$servers=Get-ADDomainController -Filter * -Server $domain
foreach ($server in $servers)
{
    $query = invoke-command -ComputerName $server.hostname -ScriptBlock {Get-
WindowsFeature | where-object {$_.installed -eq $true -and $_.featuretype -eq
"role"}} }
    $obj=[pscustomobject]@{
        ServerName = $server.hostname
        Role = $query.name -join ", "
    }
    $obj
}
}

Function Get-AllowedDHCPservers
{
    $query = Get-ADObject -SearchBase (get-ADRootDSE).configurationNamingContext -
Filter "objectclass -eq 'dhcpclass' -AND Name -ne 'dhcproot'"
    if($null -eq $query)
    {
        [PSCustomObject]@{
            AllowedDHCP = "<N/A>"
        }
    }
    else
    {
        $query.name
    }
}

$TimeSource = $TimeSource | ConvertTo-Html -Fragment
$LastUpdateTime = $LastUpdateTime | ConvertTo-Html -Fragment
$RoleServer = $RoleServer | ConvertTo-Html -Fragment
$DHCPAllowed = $DHCPAllowed | ConvertTo-Html -Fragment -property
@{label='Allowed DHCP' ; expression = {$$_}}

$HTML = @"
<!DOCTYPE html>
<html>
<head>
<title>Report</title>
<meta name="generator" content="PowerShell" />
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

$HTMLCSS
</head>
<body>
$HTMLtitle

<h3>Time source</h3>
$TimeSource

<h3>Date of the last update</h3>
$LastUpdateTime

<h3>Roles Installed</h3>
$RoleServer

<h3>Allowed DHCP servers</h3>
$DHCPAllowed

$End Date $(get-date -Format "dd/MM/yyyy HH:mm")
</body>
</html>
"@

$HTML | out-file -FilePath $HTMLfilename
#endregion

Invoke-Item $HTMLfilename

foreach ($domain in (Get-ADForest).domains)

```

```
{
  Write-EZLog -Category INF -Message "----- Start domain $domain -----
-----" -ToScreen

  Write-EZLog -Category INF -Message "Test ReportFolder exist" -ToScreen

  if (-not(Test-Path $ReportPath\$domain))
  {
    New-Item -Path "$ReportPath\$domain" -Type Directory | Out-Null
  }
  Write-EZLog -Category INF -Message "TimeSource" -ToScreen
  $TimeSource= Get-TimeSource-domain $domain
  Write-EZLog -Category INF -Message "LastUpdateTime" -ToScreen
  $LastUpdateTime= Get-LastUpdateTime-domain $domain
  Write-EZLog -Category INF -Message "RoleServer" -ToScreen
  $RoleServer= Get-RoleServer-domain $domain
  Write-EZLog -Category INF -Message "DHCPAllowed" -ToScreen
  $DHCPAllowed = Get-AllowedDHCPservers
}
```