

Audyty grup z zagnieżdżoną grupą z innej domeny

Komendę uruchamiamy jako Administrator w powershell na systemie z zainstalowanymi przystawkami RSAT.

```
# Pytaj użytkownika o nazwę grupy, kontroler domeny (DC) i poświadczenia
$Group = Read-Host -Prompt "Podaj nazwę grupy"
$DC = Read-Host -Prompt "Podaj kontroler domeny bądź jej nazwę (dc.example.local / example.local)"
$Credential = Get-Credential -Message "Podaj poświadczenia użytkownika do dostępu do domeny $DC"

# Pobierz członków grupy z określonego kontrolera domeny
$membership = Get-ADGroup $Group -Properties Member -Server $DC -Credential $Credential
$membership.Member | Group-Object { ($_ -split '(?=DC=)',2)[1] } |
ForEach-Object {
    [adsis] $ldap = 'LDAP://{0}' -f $_.Name
    [string] $domain = $ldap.Name

    foreach($member in $_.Group) {
        $obj = Get-ADObject $member -Server $domain -Credential $Credential
        [pscustomobject]@{
            MemberOf      = $membership.Name
            Domain        = $domain
            Name           = $obj.Name
            DSN           = $obj.DistinguishedName
            SamAccountName = $obj.SamAccountName
            ObjectClass    = $obj.ObjectClass
        }
    }
}
```

Wersja z zapisem pliku do CSV

```

# Pytaj użytkownika o nazwę grupy, kontroler domeny (DC) i poświadczenia
$Group = Read-Host -Prompt "Podaj nazwę grupy"
$DC = Read-Host -Prompt "Podaj kontroler domeny bądź jej nazwę (dc.example.local / example.local)"
$Credential = Get-Credential -Message "Podaj poświadczenia użytkownika do dostępu do domeny $DC"

# Pobierz członków grupy z określonego kontrolera domeny
$membership = Get-ADGroup $Group -Properties Member -Server $DC -Credential $Credential

# Zbieramy obiekty członków grupy
$results = foreach ($domainGroup in $membership.Member | Group-Object { ($_ -split '(?=DC=)',2)[1] }) {
    [adsis] $ldap = "LDAP://$($domainGroup.Name)"
    [string] $domain = $ldap.Name

    foreach ($member in $domainGroup.Group) {
        $obj = Get-ADObject $member -Server $domain -Credential $Credential
        [pscustomobject]@{
            MemberOf      = $membership.Name
            Domain        = $domain
            Name           = $obj.Name
            DSN           = $obj.DistinguishedName
            SamAccountName = $obj.SamAccountName
            ObjectClass    = $obj.ObjectClass
        }
    }
}

# Pytanie do użytkownika czy zapisać wynik
$saveCsv = Read-Host -Prompt "Czy chcesz zapisać wynik do pliku CSV? (t/n)"

if ($saveCsv -eq 't') {
    $path = Read-Host -Prompt "Podaj pełną ścieżkę do pliku CSV"

    try {
        $results | Export-Csv -Path $path -NoTypeInfo -Encoding UTF8
        Write-Host "Wynik zapisano do pliku: $path" -ForegroundColor Green
    }
    catch {
        Write-Host "Wystąpił błąd podczas zapisywania pliku:" -ForegroundColor Red
        Write-Host $_.Exception.Message -ForegroundColor Red
    }
}

```

```
}  
else {  
    Write-Host "Wynik nie został zapisany do pliku." -ForegroundColor Yellow  
}  
  
# Możesz też wyświetlić wyniki na ekranie jeśli chcesz  
$results
```

Revision #4

Created 10 March 2025 13:51:07 by Tomasz Konieczny

Updated 5 August 2025 11:49:59 by Tomasz Konieczny