

Get AD Group Member List Script

Save this as .ps1 file and run as admin in Domain Controller.

Simple

```
$groups = (Get-ADGroup -Filter * | Select name -ExpandProperty name)

$table=@()
$record=@{
    "groupname" = ""
    "username"=""
}

foreach ($g in $groups){
    $i = 0
    $members = (Get-ADGroupMember -Identity $g | select name,samaccountname)

    foreach ($m in $members){
        $record."groupname"=$g
        $record."UserName"=$m.samaccountname

        $objrec= New-Object psobject -Property $record
        $table += $objrec
        $i+=1
    }
}

if($i -eq 0){
    $record."groupname"=$g
    $record."UserName"=""
    $objrec= New-Object psobject -Property $record
    $table += $objrec
}
```

```

}

$dateinfile = (Get-Date).AddMonths(-1).ToString('yyyy-MM')
$domain = (Get-WmiObject -Namespace root\cimv2 -Class Win32_ComputerSystem | Select Domain).Domain
$table |Export-Csv "C:\temp\$dateinfile - $domain - Group Member List.csv" -NoTypeInfoInformation

```

Detailed

```

Import-Module ActiveDirectory

$date=Get-Date -f yyyy-MM-dd
$domain=(Get-WmiObject win32_computersystem).Domain
$csvexport=$null
$csvlocation="C:\temp\$domain-GroupReview-$date.csv"
If ((Test-Path C:\temp) -eq $false){
    New-Item C:\temp -ItemType Directory
}
If ((Test-Path $csvlocation) -eq $true){
    Remove-Item $csvlocation
}

$toprint=@()
$record=@{
    "Name" = ""
    "GroupScope"=""
    "Description"=""
    "adminCount"=""
    "ManagedBy"=""
    "samaccountname"=""
    "membertype"=""
    "WhenCreated"=""
    "WhenChanged"=""
}

$groupaccounts=Get-ADGroup -filter * -Properties * | Select
"Name","GroupScope","Description","adminCount","ManagedBy","WhenCreated","WhenChanged"
foreach ($grp in $groupaccounts) {
    $isempty=$true
    if ($grp.adminCount -eq 1){
        $record."adminCount"="Yes"
    }
}

```

```

} else {
    $record."adminCount"=""
}
if ($grp.ManagedBy){
    $record."ManagedBy"=$grp.ManagedBy.Split(",")[0].replace("CN=", "")
} else {
    $record."ManagedBy"=""
}

$record."Name"=$grp.Name
$record."GroupScope"=$grp.GroupScope
$record."Description"=$grp.Description
$record."samaccountname"=""
$record."membertype"=""
$record."WhenCreated"=$grp.WhenCreated.ToString("dd-MMM-yyyy HH:mm:ss")
$record."WhenChanged"=$grp.WhenChanged.ToString("dd-MMM-yyyy HH:mm:ss")

try{
    $groupmembers = (Get-ADGroupMember -Identity $grp.Name | select name,samaccountname,objectClass)
    foreach ($m in $groupmembers){
        if($m.objectClass){
            $membertype=$m.objectClass
        }else{
            $membertype="builtin group"
        }
        $record."samaccountname"=$m.samaccountname+"<br>"+$record."samaccountname"
        $record."membertype"=$membertype+"<br>"+$record."membertype"

        $isempty=$false
    }

    if($isempty -eq $true){
        $record."ManagedBy"=$grp.ManagedBy
        $record."samaccountname"="-"
        $record."membertype"="-"
    }

    $objrec= New-Object psobject -Property $record
    $toprint += $objrec
} catch{ # For OWFT foreignSecurityPrincipal objects

```

```

$members = dsquery group -name $grp.Name | dsget group -members
$foreignAccount = [System.Collections.ArrayList]::new()
$nonForeignAccount = [System.Collections.ArrayList]::new()

foreach ($m in $members){
    if ($m.Contains("ForeignSecurityPrincipals")) {
        $foreignAccount.add($m.ToString().replace('"',''))
    }else{
        $nonForeignAccount.add($m.ToString().replace('"',''))
    }
}

$foreignObjects = Get-ADObject -Filter {ObjectClass -eq "foreignSecurityPrincipal"} -Properties msds-
principalname,memberof
foreach ($fObject in $foreignAccount){
    $faccountname = ($foreignObjects | Where-Object DistinguishedName -eq $fObject ).'msds-
principalname'

    $record."samaccountname"=$faccountname+"<br>"+$record."samaccountname"
    $record."membertype"='foreignSecurityPrincipal'+"<br>"+$record."membertype"
}

foreach ($obj in $nonForeignAccount){
    if($obj -ne ""){
        $fullname=$obj.Split(',')[0].replace('CN=', '')
        $userObj = Get-ADUser -Filter {name -like $fullname} | Select-Object SamAccountName,ObjectClass

        $record."samaccountname"=$userObj.SamAccountName+"<br>"+$record."samaccountname"
        $record."membertype"=$userObj.ObjectClass+"<br>"+$record."membertype"
    }
}

$objrec= New-Object psobject -Property $record
$toprint += $objrec
}

$sn=0
foreach($tt in $toprint){
    $sn=$sn+1
    $csvexport += @(

```

```
[pscustomobject]@{
    "SN"=$sn
    "Group Name" = $tt.Name
    "Group Scope" = $tt.GroupScope
    "Description" = $tt.Description
    "Admin Rights" = $tt.adminCount
    "Managed By" = $tt.ManagedBy
    "Member Name" = $tt.samaccountname.Replace("<br>","`n").ToString()
    "MemberType" = $tt.membertype.Replace("<br>","`n").ToString()
    "Creation Date" = $tt.WhenCreated
    "Modified Date" = $tt.WhenChanged
}
)
}

$csvexport | Export-csv -Path $csvlocation -NoTypeInfoInformation
```

Revision #3

Created 1 January 2024 20:08:07 by aki

Updated 1 January 2024 20:35:33 by aki