Tier 0 Account Operators

In Microsoft's Active Directory Security Groups blog post the permissions granted to the account operators are defined as:

The Account Operators group grants limited account creation privileges to a user. Members of this group can create and modify most types of accounts, including those of users, local groups, and global groups, and members can log in locally to domain controllers.

By default, this built-in group has no members, and it can create and manage users and groups in the domain

Account Operators Can Manage These permissions described above, there is no mention of computers. However account operators group has full control of most every computer on the domain. I want to show why the Account Operators groups has permissions over computers, and also what other objects in the domain they can control.

What is Schema?

The first step to get to our solution is to find how computers are mapped to schema another schema class in the Active Directory Schema. Then find how account operators are granted permissions to control the computer objects.

Per Windows-Active-Directory.com, the AD schema is defined as:

> Active Directory (AD) schema is a blueprint which describes the rules about the type of objects that can be stored in the AD as well as the attributes related to these objects. Objects, classes, and attributes are the building blocks of the schema object definition.

That's as deep as we'll go discussing 'what' the schema is today. Now to demonstrate how Computers are mapped into Groups and Users, and how you can search deeper other relationships in the schema.

Querying Schema Objects

First we need to export all schema objects from the domain and store them as a variable for easy searching:

```
#get schema root path with this
$schemaPath = (Get-ADRootDSE)
#Export all schema objects (and their properties) and store it as $schemaobjects
$schemaobjects = Get-ADObject -SearchBase ($schemaPath.SchemaNamingContext) -
LDAPFilter ` "(schemaidguid=*)" -Properties *
#The above code should store around ~5000 objects into the $schemaobjects
variable
```

According to the Microsoft definition for account operators, we need to search for any object that has ‘user’ or ‘group’ listed as their parent object. For this we will look at the attribute ‘subclassof’. Normally when you are looking for child objects in active directory you can look at the parent object, like a group, and look at an attribute like ‘memberof’ to see who is added to this group.

Schema is a different query

To find what objects are members of a parent class in a schema, you have to look at all the objects individually and **look UP to a parent object.**

Check out ‘user’ and ‘group’. What are they a subclassof?

```
$ObjsToCheck = @('user','group')

foreach($obj in $ObjsToCheck){

$schemaobjects |where-Object -Property CN -eq $obj|select cn,subclassof }
```

Topclass of user and group

See that ‘user’ is a member of ‘organizationalPerson’ and ‘group’ is a member of ‘top’ . The relationship is also outlined in the earlier schema picture.

Now to look up from all objects in order to find objects that are subclasses of ‘user’ and ‘group’. For this we query all objects (previously stored as $schemaobjects) and search for subclassof ‘user’ or ‘group’. In the code below, I searched for ‘user’. Change it to ‘group’ to find the relationships for group.

```
$usersubclass = $schemaobjects |Where-Object -Property subclassof -eq
'user'|select cn,subclassof

$usersubclass
```

Here we can see what schema objects are linked to User. This is where ‘computer’ is. By the inherited values, Microsoft defines a ‘computer’ as a ‘user’. Because account operators has access to all objects that are ‘user’ objects, account operators can control computer objects.

There are other privileged subclasses of computer objects that the Account Operators then inherits access to:

- ms-DS-Managed-Service-Account

- ms-DS-Group-Managed-Service-Account
- ms-Exch-Computer-Policy

These 3 classes of objects are extremely privileged in an Active Directory domain. Also listed in the classes of user is ‘ms-PKI-Key-Recovery-Agent’ access to the “Key Recovery Server”

Be aware of the fact that account operators can control the objects listed above. Not just users and computers

For reference sake, here is the hierarchical structure from group and user up to the top.

Bonus Content:

Diving a bit deeper, if you need to query what attributes are linked to a schema class, look at the “systemmaycontain” property on each class you want to investigate. You will also have to list all the auxiliary classes (supportive classes) linked to the schema class, as well as the schema linked to all parent classes, subclassof, to see all the attributes possible for a ‘user’ object on a domain :

```
$schemaclass = “user”

$s = Get-ADObject -SearchBase (Get-ADRootDSE).SchemaNamingContext -Filter {name -
like "$schemaclass"} -Properties MayContain,SystemMayContain

$s.auxiliaryclass

$s.maycontain

$s.systemmaycontain
```