Визитка

ИВАН КОРОБКО, сертифицированный специалист MCP, автор более 50 статей и двух книг. Занимается созданием различных приложений для Active Directory

Управление ресурсами пользователя
без перезагрузки компьютера

Можно ли сотруднику избежать перезагрузки рабочей станции после включения его учетной записи в новую группу безопасности? Рассмотрим варианты решения данной проблемы

Среди повседневных задач системного администратора встречается предоставление доступа пользователям к сетевым ресурсам – дискам или принтерам. Алгоритм операции прост: системный администратор в оснастке Active Directory Users and Computers включает учетную запись пользователя в соответствующую группу безопасности. После этого пользователю необходимо перезагрузить свою рабочую станцию. Если работу администратора можно автоматизировать с помощью веб-приложения или какого-либо сценария, то пользователя на первый взгляд нельзя избавить от перезагрузки. Поставленную задачу можно решить несколькими способами, однако рассказ пойдет о наиболее удачном из них.

Типы групп безопасности

Ключ к решению задачи находится в группах безопасности, а точнее, в их типах. В каталоге Active Directory существует несколько типов групп [1], о существовании которых знает любой системный администратор (см. таблицу 1).

Разработчики компании Microsoft говорят о принципиальном различии глобальных и универсальных групп – их области действия, однако не распространяются о том, что изменения членства в глобальных группах для учетной записи пользователя осуществляется мгновенно, т. е. сотруднику нет необходимости перезагружать свою рабочую станцию. Именно этой особенностью универсальных групп и воспользуемся для решения задачи.

Схема управления группами безопасности

На рис. 1 приведена схема изменения структуры групп безопасности. На ней видно, что между учетной записью пользователя и группами, членом которых он является, помещается еще одна группа безопасности, жестко связанная с учетной записью пользователя. Исключение составляют только встроенные в домене локальные группы, которые не могут быть членами универсальных групп.

К локальным группам относятся такие, как Domain Admins, Domain Users и т.д. Изменение членства учетных записей пользователей в этих группах осуществляется только при создании нового пользователя. При первом входе сотрудника в Сеть под новым именем у новой учетной записи неизбежно осуществляется формирование списка групп, в том числе встроенных. Для реализации оговоренной схемы следует выполнить следующие действия:

- Изменить область действия групп к Universal, кроме служебных групп безопасности.
- Создать универсальные группы безопасности в соответствии с существующими учетными записями пользователей.
- Реализовать связку учетной записи пользователя и соответствующей группы безопасности, добавить ее в свойства учетной записи пользователя.
- Перенести список универсальных групп безопасности из учетной записи пользователя в новую группу.

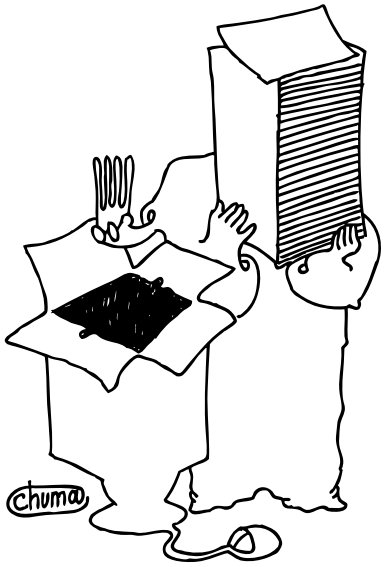
Изменение области действия групп

Область действия группы безопасности и ее тип определяются совокупным параметром GroupType. В таблице 2 приведены значения атрибута, соответствующие различным типам групп.

В листинге 1 приведен пример сценария, который изменяет область действия группы на Universal. Во время работы скрипта осуществляются поиск всех групп безопасности и изменение их типа. Во время работы сценария может возникнуть ошибка, поскольку выполнить преобразование группы в Universal невозможно, если в нее входит хотя бы одна глобальная группа. Из-за этого необходимо несколько раз выполнить сценарий.

Листинг 1. Изменение типа группы безопасности на Universal Security

```
# Определение имени домена
$objD=([ADSI]"LDAP://RootDSE").defaultNamingContext
$domain= ([ADSI]"LDAP://$objD").path
# Поиск всех групп безопасности в домене
$objS=New-Object System.DirectoryServices.DirectorySearcher
$objS.Filter="(&(objectclass=group))"
$objS.SearchRoot=$domain
$result=$objS.FindAll()
# Обработка списка групп безопасности
$result |% {
    if (([ADSI]$_.path).isCriticalSystemObject -eq $TRUE) {}
```



Будьте очень внимательны и разработайте план миграции структуры каталога Active Directory

```
    else
        { # Фильтрация некритических объектов
        $tempObj=[ADSI]$_.path
        # Изменение типа группы безопасности на Universal
        # Security
        $tempObj.put("grouptype","-2147483640")
        $tempObj.setinfo()
        Write-Host $tempObj.name
        }
    }
Write-Host "End Of Script"
```

Создание универсальных групп безопасности на основе учетных записей пользователей

Данный скрипт выполняется не для всего домена, а для указанного контейнера. Такая мера позволит не обрабатывать лишние учетные записи и ускорит скорость работы сценария. Используя предлагаемую масштабируемость, можно протестировать его работу на небольшом количестве пользователей. Все внешние настройки скрипта рекомендуется хранить во внешнем файле. Наиболее удачный формат – XML. Такими параметрами для данного сценария являются следующие (см. листинг 2):

Идентификатор контейнера, в котором хранятся пользователи. В качестве идентификатора рекомендуется использовать имя контейнера или его GUID. Поскольку имя контейнера может дублироваться в домене, то надежнее всего использовать 128-битный уникальный идентификатор.

Идентификатор контейнера, в котором будут созданы группы безопасности, связанные с учетными записями пользователей. Рекомендуется сформировать в данном контейнере линейную структуру, поскольку иерархическое расположение этих групп не играет никакой роли. В качестве идентификатора рекомендуется также использовать GUID.

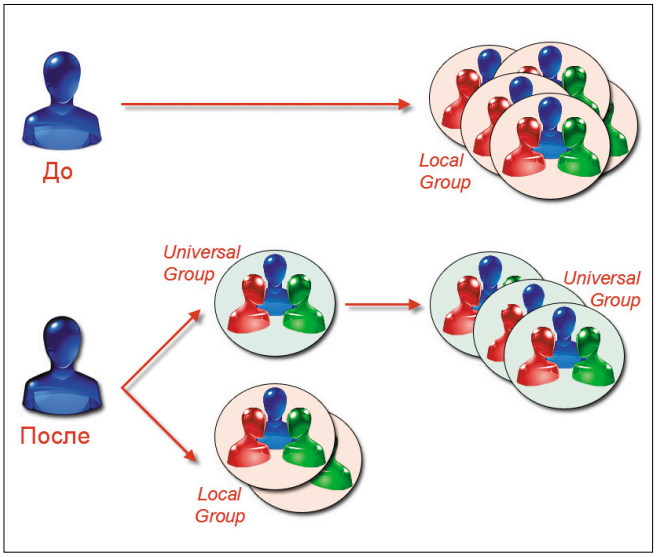
Таблица 1. Типы групп безопасности в каталоге Active Directory

Тип группы	Область действия
Universal	Любой домен или лес
Global	Любой домен
Domain Local	Только локальный домен

Таблица 2. Расшифровка значений параметра GroupType

Тип группы	Значение
Global Security Group	-2147483646
Local Security Group	-2147483644
BuiltIn Group	-2147483643
Universal Security Group	-2147483640
Global Distribution Group	2
Local Distribution Group	4
Universal Distribution Group	8

Рисунок 1. Схема членства в группах учетной записи пользователя



Связка учетной записи пользователя и создаваемой группы. Необходимость связки обусловлена тем, что двух одинаково названных объектов среди групп и учетных записей пользователя создать невозможно. Рекомендуется в качестве идентификатора связки использовать уникальный префикс перед сокращенным именем в Сети (login), например, un$_.

Пример XML-файла приведен в листинге 2.

Листинг 2. Пример XML-файла config.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<Root>
<Prefix Value="un$_"/>
<UsersOU GUID="010DAAF18C8E5042A144A3C14F251D3D"/>
<GroupsOU GUID="6C18D99597280B4BB4EC5F8DF8AEBBBB"/>
</Root>
```

Чтение данных из XML-файла реализуется с помощью встроенного провайдера [XML] и командлета Get-Content. После того как содержимое XML-файла в виде дерева загружено в память, очень просто получить нужное значение, указав в объектной модели заданные в файле теги. В листинге 3 приведен пример определения значения префикса.

Листинг 3. Чтение данных из XML-файла

```
[XML]$obj = Get-Content .\config.xml
$Prefix = $obj.Root.Prefix.Value
Write-Host $Prefix
```

Определение GUID пользователя может быть реализовано несколькими способами, в том числе с помощью сценария или оснастки Active Directory в расширенном режиме.

Для его определения с помощью сценария выполнятся поиск всех объектов в каталоге Active Directory с указанным именем с помощью объекта System.DirectoryServices. DirectorySearcher (см. листинг 4). В результате выполнения сценария администратор выбирает нужный из найденных идентификаторов по RDN-пути контейнера.

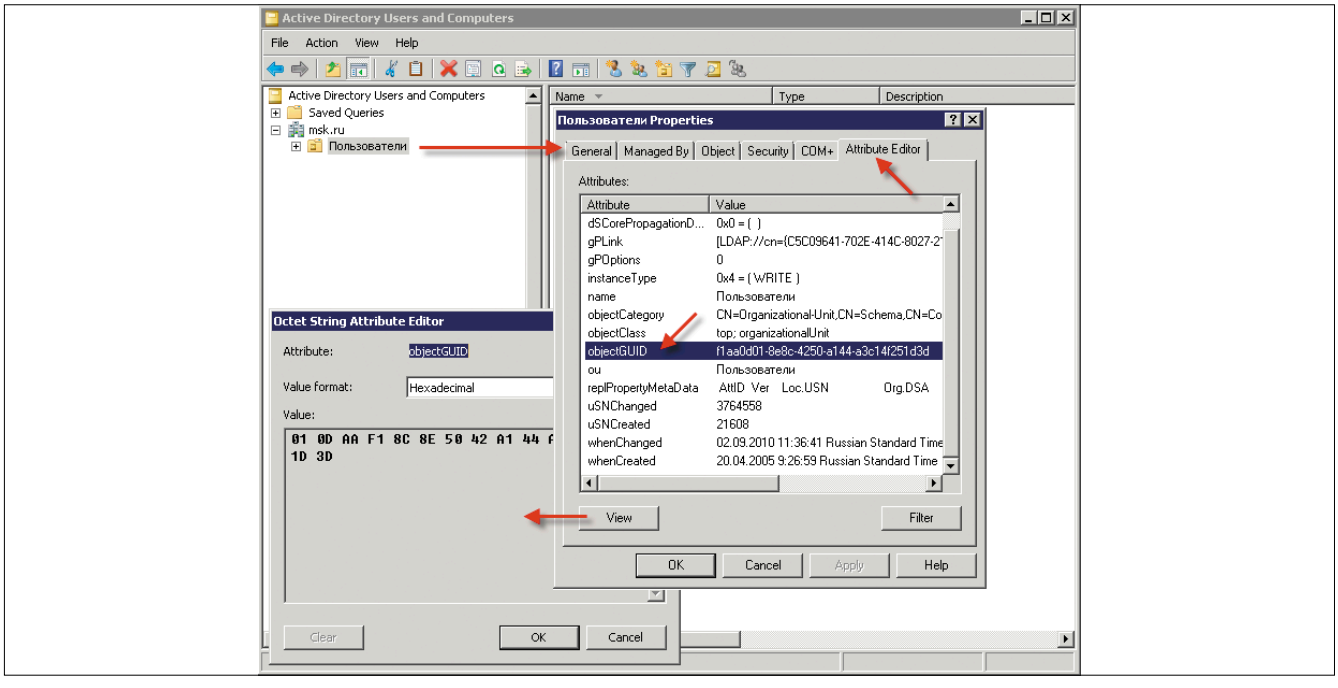
Листинг 4. Определение GUID-объекта в каталоге Active Directory

```
# Имя искомого контейнера
$ou="Пользователи"
# Определение имени домена
$domain="LDAP://"+ ↵
        ([ADSI]"LDAP://RootDSE").defaultnamingcontext
# Создание объекта для поиска
$objS=New-Object System.DirectoryServices.DirectorySearcher
$objs.Filter="(&(objectclass=organizationalunit)(name=$ou))"
$objs.SearchRoot=$domain
# Поиск объектов по заданным критериям
$Result = $objs.FindAll()
$Result | %{
# Определение GUID найденного объекта
    $GUID=([ADSI]$_.path).GUID
    # Определение RDN-пути к объекту
    $Path=$_.path
    # Вывод данных в консоль
    Write-host $GUID " "  $path
    }
```

В оснастке Active Directory Users and Computers определить GUID также не составляет труда. Для этого необходимо, инициализировав расширенный режим работы оснастки (View → Advanced Features) в свойствах нужного контейнера, перейти во вкладу Attribute Editor. В появившемся диалоговом окне установите курсор на атрибут objectGUID и нажмите на кнопку View в нижней части окна. В новом диалоговом окне в формате Hexadecimal скопируйте значение GUID (см. рис. 2). Все, что осталось, – удалить лишние пробелы.

Замечание: механизм обращения к объекту по GUID очень прост. Вместо RDN-пути достаточно указать GUID-объекта, например:

Рисунок 2. Определение GUID-объекта в оснастке Active Directory Users and Computers



```
([ADSI]"LDAP://<GUID= 010DAAF18C8E5042A144A3C14F251D3D>").path
```

Алгоритм сценария (см. листинг 5), создающего учетные записи групп на основе имен имеющихся пользователей, выглядит следующим образом. На первом этапе осуществляются чтение данных их XML-файла и получение доступа к соответствующим объектам в каталоге Active Directory. Далее выполняется поиск учетных записей пользователей и формирование на его основе имени группы безопасности. Затем осуществляется создание Universal групп безопасности в Active Directory. Во время создания определяют тип группы и в поле description записывают имя пользователя.

Листинг 5. Определение GUID-объекта в каталоге Active Directory

```
# Подключение к XML-файлу
cd "…"
[XML]$obj = Get-Content .\config.xml
# Чтение данных из XML-файла
$StoreOU=$obj.Root.GroupsOU.GUID
$UsersOU=$obj.Root.UsersOU.GUID
$Prefix=$obj.Root.Prefix.Value
# Поиск учетных записей пользователей
$objSearch=New-Object ↵
        System.DirectoryServices.DirectorySearcher
$objSearch.Filter="(&(objectClass=person) ↵
        (!(objectClass=computer)))"
$objSearch.SearchRoot="LDAP://<GUID=$UsersOU>"
$result = $objSearch.FindAll()

$result | % {
$GroupName = $Prefix + $_.properties.samaccountname
$GroupDescrition = $_.properties.description
Write-Host $GroupName
# Создание новой группы
    $objGroup = [ADSI]"LDAP://<GUID=$StoreOU>"
    $objNewGroup = $objGroup.Create("group", "cn=$GroupName")
    $objNewGroup.put("samaccountname","$GroupName")
    $objNewGroup.put("description","$GroupDescrition ")
    $objNewGroup.put("grouptype","-2147483640")
    # Добавление пользователя в группу
    $objNewGroup.PutEx(3,"member", ↵
        @([string]([ADSI]$_.path).distinguishedName))
    $objNewGroup.setinfo()
}
Write-Host "End Of Work"
    }
```

Связка учетной записи пользователя и соответствующей группы безопасности

Добавить учетную запись пользователя в группу лучше всего в момент ее создания, присвоив значение атрибуту member. Работа с элементами массива осуществляется с помощью метода PutEx, который имеет три параметра. Первый из них идентифицирует производимую операцию. (Список операций и соответствующие им значения приведены в таблице 3.) Второй параметр определяет имя атрибута объекта, третий – массив присваиваемых значений.

Для сокращения листинга рекомендуется не создавать дополнительный сценарий включения учетных записей пользователей в соответствующие группы безопасности, а добавить всего одну строку (выделена красным цветом) в листинг 5.

Изменение основной группы безопасности пользователя

По умолчанию в качестве основной группы безопасности установлена Domain Users, имеющая идентификатор 513. Идентификатором группы является последний октек SID-пользователя (см. рис. 3), и именно он указывается в качестве значения атрибута primaryGroupID.

Изменение атрибута осуществляется так же, как и всех остальных атрибутов, – с помощью метода Put(). В листинге 6 приведен сценарий изменения основных групп у всех учетных записей пользователей, находящихся в указанном контейнере. Каждому пользователю соответствует своя основная группа, имена которых отличаются на префикс.

В этом примере уделим внимание функции, определяющей и анализирующей SID-пользователя. При вызове функции ей передается параметр – имя объекта. Затем получают его описатель с помощью функции NTAccount, который затем преобразуют в SID. Полученное значение преобразуют в массив, разделяя его с помощью символа «-». Функцией возвращается последний элемент этого массива – идентификатор группы.

Листинг 6. Изменение значений Primary Group учетных записей пользователей контейнера

```
Function DetectSID
{
# Чтение значения передаваемого функции параметра
$object=$args[0]
# Определение SID-группы безопасности
$AdObj = New-Object ↵
        System.Security.Principal.NTAccount($object)
$strSID = $AdObj.Translate( ↵
        [System.Security.Principal.SecurityIdentifier])
# Определение последнего октека SID-группы безопасности
$Value=$strSID.Value.split("-")
$ID=[int]$Value[$Value.count-1]
# Возвращение полученного значения
return $ID
}
# Подключение к XML-файлу
CD "…"
[XML]$obj = Get-Content .\config.xml
# Чтение данных из XML-файла
$StoreOU=$obj.Root.GroupsOU.GUID
$UsersOU=$obj.Root.UsersOU.GUID
$Prefix=$obj.Root.Prefix.Value
# Поиск учетных записей пользователей в указанном
# контейнере
$objSearch=New-Object ↵
        System.DirectoryServices.DirectorySearcher
$objSearch.Filter="(&(objectClass=person) ↵
        (!(objectClass=computer)))"
$objSearch.SearchRoot="LDAP://<GUID=$UsersOU>"
$result = $objSearch.FindAll()
$result | % {
    # Определение имени связанной группы
    $GroupName = $Prefix + $_.properties.samaccountname
    # Определение идентификатора группы из SID с помощью
    # функции
    $newID=DetectSID $GroupName
    Write-Host "User: " $_.properties.samaccountname
    Write-Host "Link Group: " $GroupName
    Write-Host "OLD Primary Group ID: " ↵
        $_.properties.primarygroupid
    Write-Host "NEW Primary Group ID: " $newID
```

Таблица 3. Операции с объектом, доступные для метода PutEx()

Код операции	Выполняемое действие
1	Опустошение элементов массива
2	Обновление значения с заменой его на новое значение
3	Добавление элементов массива к существующим
4	Удаление указанного значения из существующего массива

```
    Write-Host ↵
        "***********************************************"
    # Получение доступа к объекту (пользователь)
    $objID=[ADSI]$_.Path
    # Изменение типа группы
    $objID.Put("primaryGroupID",$newID)
    # Запись изменений в каталог Active Directory
    $objID.setInfo()
    }
```

Замечание: в PowerShell функция всегда объявляется до ее вызова. Поэтому в начале скрипта всегда описывают используемые функции, а его тело располагают в его конце.

Перенос списка универсальных групп безопасности из учетной записи пользователя в соответствующую ему группу безопасности

Задача переноса списка групп состоит из двух частей: копирование списка из одного места в другое, удаление из первоисточника.

Реализовать копирование учетных записей групп от пользователя к группе безопасности можно двумя способами: модернизировав первый сценарий или создав отдельный.

Предпочтительным является первый вариант, поскольку для воплощения в жизни второго варианта необходимо почти полностью повторить предыдущий.

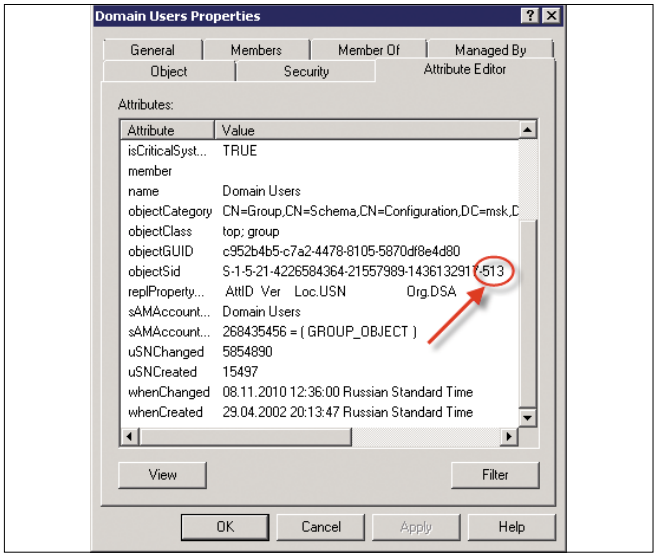
В листинг 5 после процедуры создания группы безопасности и вызова метода setInfo() необходимо добавить фрагмент сценария, приведенного в листинге 7.

Напомним, что из листинга 5 используется объект $MemberOf – список RDN-путей к каталогу и объект objNew Group – новая группа.

Листинг 7. Копирование списка групп безопасности из пользователя в новую группу

```
ForEach($Element in $MemberOF)
    {
    $AddMembers=[ADSI]("LDAP://"+$Element)
    $AddMembers.PutEx(3,"member", ↵
        @([string]$objNewGroup.distinguishedName))
    $AddMembers.setInfo()
    }
```

Рисунок 3. Идентификатор безопасности



Замечание: поле memberOf, отображаемое в объектной модели, фиктивное. Поэтому управление членством в группах безопасности осуществляется только через атрибут member:

```
$Obj1.PutEx(3,"member", @([string]$Obj2.distinguishedName))
```

Вторая часть – удаление из свойств учетной записи пользователя во вкладке Member Of универсальных групп. Некорректная работа данного сценария может повлечь за собой очищение списка членов всех перечисленных в ней групп безопасности, поэтому он не входит в другие сценарии и перед применением на действующем рабочем пространстве требует проверки в тестовой среде и полного резервного копирования каталога Active Directory.

Для удаления группы (см. листинг 8) из списка осуществляется обращение к группе и удаление конкретной записи из ее членов, это достигается с помощью метода PutEx(), при этом первый параметр равен 4 (см. таблицу 3).

Листинг 8. Изменение основной группы в учетной записи пользователя

```
# Подключение к XML-файлу
CD "…"
[XML]$obj = Get-Content .\config.xml
# Чтение данных из XML-файла
$StoreOU=$obj.Root.GroupsOU.GUID
$UsersOU=$obj.Root.UsersOU.GUID
$Prefix=$obj.Root.Prefix.Value
# Поиск учетных записей пользователей в указанном контейнере
$objSearch=New-Object ↵
        System.DirectoryServices.DirectorySearcher
$objSearch.Filter="(&(objectClass=person) ↵
        (!(objectClass=computer)))"
$objSearch.SearchRoot="LDAP://<GUID=$UsersOU>"
$result = $objSearch.FindAll()
$result | % {
# Определение RDN-пути удаляемой учетной записи
$UserName = $_.properties.distinguishedname
# Формирование списка групп, членом которых является
# пользователь
$MemberOf = $_.Properties.memberOf
    ForEach($element in $memberof)
    {
    # Получение доступа к группе безопасности
    $objS  = [ADSI]"LDAP://$element"
    # Определение типа группы безопасности: Universal
    if ($objS.grouptype -eq -2147483640 )
        {
        # Удаление пользователя из группы безопасности
            $objS.name
            $objs.PutEx(4,"member",@($UserName))
            $objs.setinfo()
        }
    }
}
Write-Host "End Of Work"
```

***

Внедряя такую систему, не забудьте, что изменился принцип работы системного администратора. Теперь администратору и некоторым сервисам и сценариям регистрации пользователей в Сети необходимо работать не с учетными записями учетных записей пользователей, а с прилинкованными к ним группами безопасности. Изменение конфигурации этих сервисов потребует некоторого времени. Будьте внимательны и разработайте план миграции структуры каталога Active Directory. И не забывайте создавать резервные копии. EOF

1. Group Type and Scope Usage in Windows – http://support.microsoft.com/kb/231273.