


# Базовая настройка Kerberos в IIS для пулов, исполняемых в контексте ApplicationPoolIdentity или gMSA, а также нюансы делегирования Constrained Delegation и Resource-Based Constrained Delegation

 [blog.it-kb.ru/2022/12/17/configuring-kerberos-authentication-for-iis-pools-running-as-applicationpoolidentity-or-gmsa-and-unconstrained-constrained-resource-based-constrained-delegation-to-cifs-file-server](https://blog.it-kb.ru/2022/12/17/configuring-kerberos-authentication-for-iis-pools-running-as-applicationpoolidentity-or-gmsa-and-unconstrained-constrained-resource-based-constrained-delegation-to-cifs-file-server)

Автор: Алексей Максимов

17.12.2022



В этой статье мы рассмотрим примеры базовой настройки протокола аутентификации **Kerberos** для пула приложений **IIS v10** на сервере с ОС **Windows Server 2022**. При этом мы отдельно поговорим о разных вариантах настройки в зависимости от типа учётной записи, в контексте которой выполняется пул приложений IIS. Кроме того, попробуем рассмотреть задачу подключения к веб-сервису IIS (фронтенд) стороннего файлового ресурса (бэкенд) и делегирования аутентификации пользователей от фронтенда к бэкенду.

Чтобы не запутаться в структуре изложения, будем использовать следующий план:

## Этап 1. Базовая настройка Kerberos для IIS

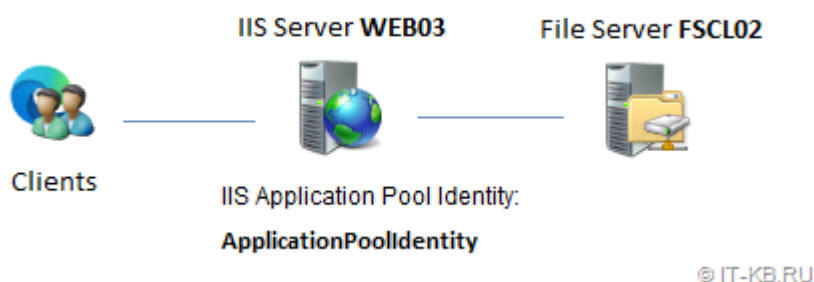
- Вариант 1. Пул IIS выполняется в контексте учётной записи самого веб-сервера
- Вариант 2. Пул IIS выполняется в контексте сервисной учётной записи gMSA

## Этап 2. Настройка разных типов делегирования аутентификации на примере IIS

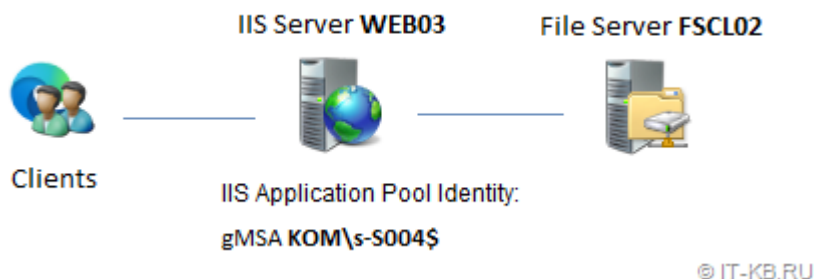
- Тип 1. Полное делегирование (Unconstrained Delegation)
- Тип 2. Ограниченное делегирование (Constrained Delegation)
  - Вариант 1. Пул IIS выполняется в контексте учётной записи самого веб-сервера
  - Вариант 2. Пул IIS выполняется в контексте сервисной учётной записи gMSA
- Тип 3. Ограниченное делегирование на основе ресурсов (Resource Based Constrained Delegation)
- Общие выводы по типам делегирования

На первом и втором этапе плана мы рассмотрим конкретные примеры настройки, исходя из двух разных вариантов работы пула приложений IIS:

Вариант 1. Пул IIS выполняется в контексте учётной записи самого веб-сервера:



Вариант 2. Пул IIS выполняется в контексте сервисной учётной записи gMSA:



## Этап 1. Базовая настройка Kerberos для IIS

Разные варианты запуска пула приложений IIS требуют разных действий по настройке поддержки Kerberos. Например, пул IIS может выполняться в контексте учётной записи самого веб-сервера (ApplicationPoolIdentity), в контексте учётной записи пользователя или в контексте сервисной учётной записи gMSA.

Вариант 1. Пул IIS выполняется в контексте учётной записи самого веб-сервера

1) Регистрируем в домене уникальную запись **SPN** вида HTTP/<fqdn> для доменной компьютерной учётной записи веб-сервера. Поиск и регистрацию SPN можно выполнить как с помощью утилиты setspn, так и с помощью **PowerShell**.

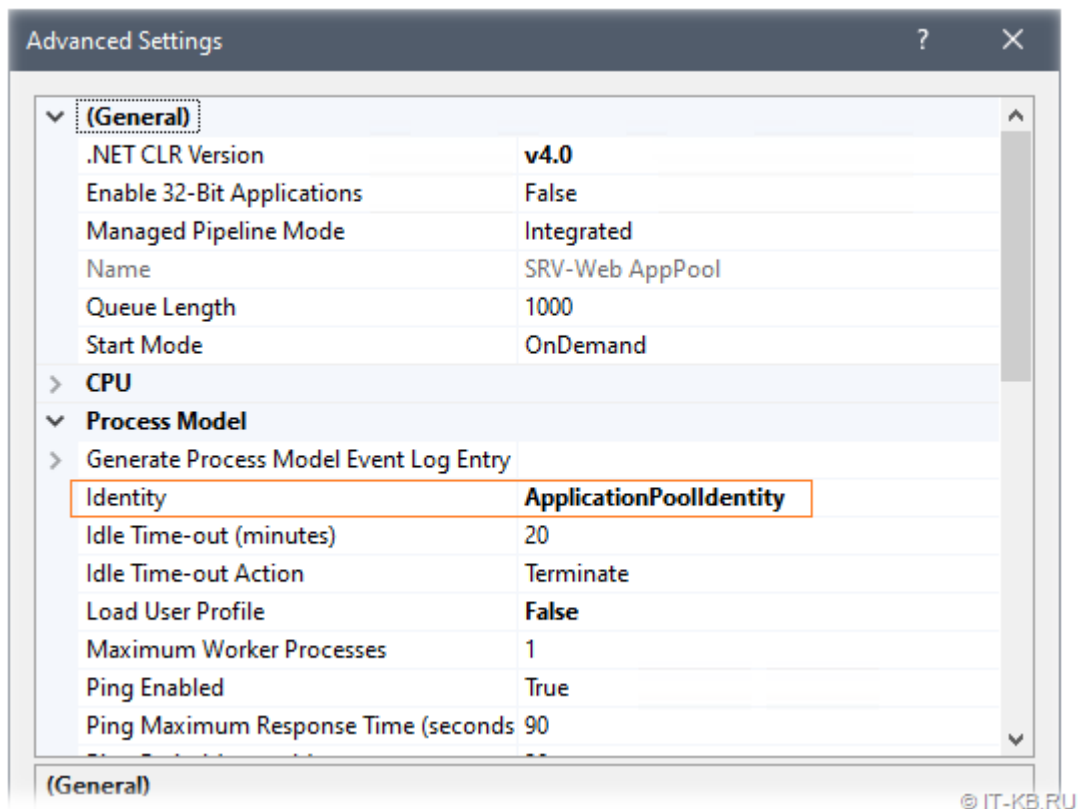
Пример команды для получения всех зарегистрированных SPN-записей для объекта типа компьютер:

```
Get-ADComputer -Identity "<имя учётной веб-сервера>" -Properties  
ServicePrincipalNames | Select-Object -ExpandProperty ServicePrincipalNames
```

В следующем примере регистрируем пару SPN-записей для компьютерной учётной записи веб-сервера:

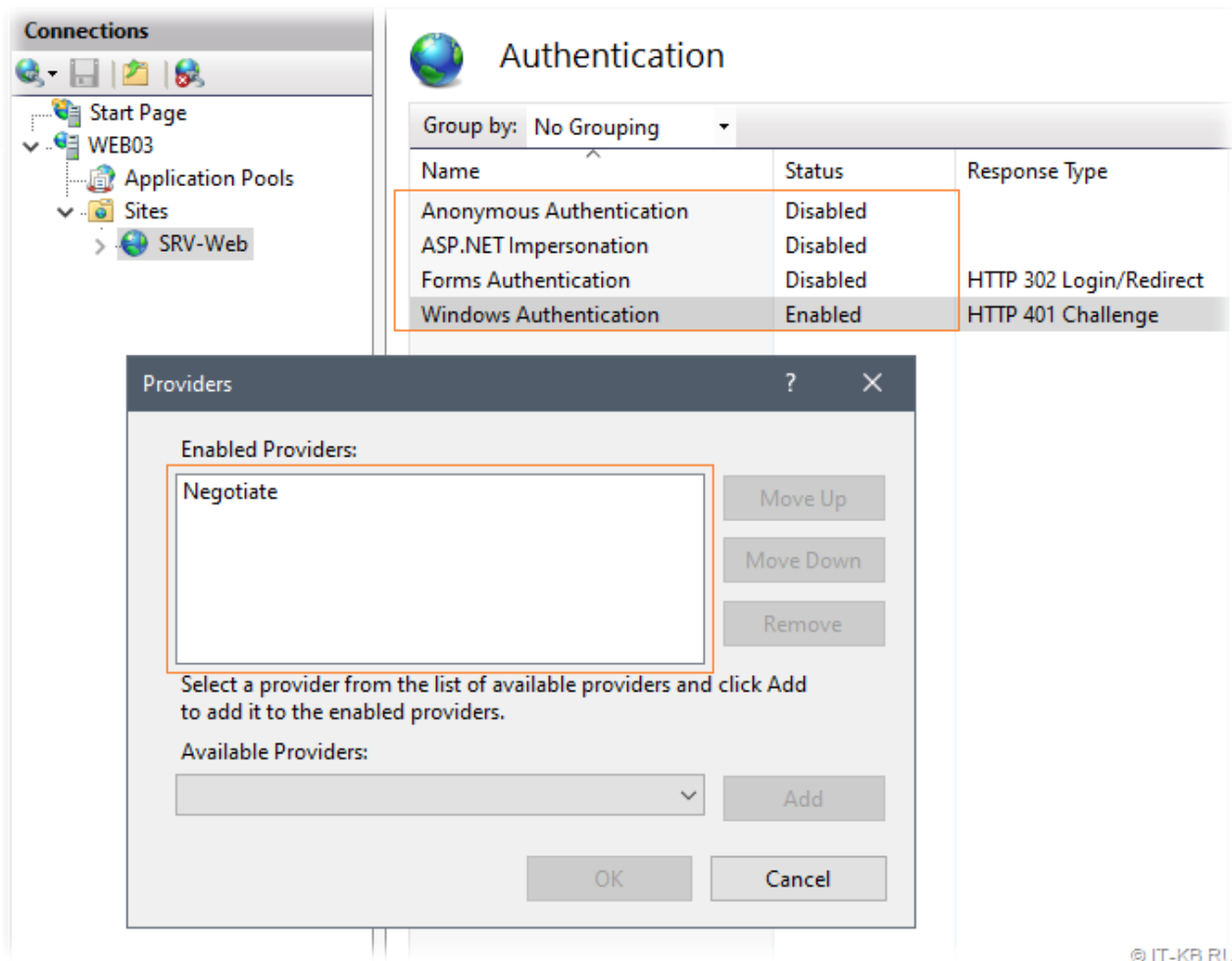
```
Set-ADComputer -Identity "WEB03" -ServicePrincipalNames @{Add='HTTP/SRV-  
Web.holding.com','HTTP/SRV-Web'}
```

2) Настраиваем в IIS пул приложений (Application Pool), используя в расширенных настройках пула в свойстве **"Identity"** значение **"ApplicationPoolIdentity"**.



3) В свойствах сайта в методах аутентификации выключаем анонимную аутентификацию, и включаем аутентификацию Windows.

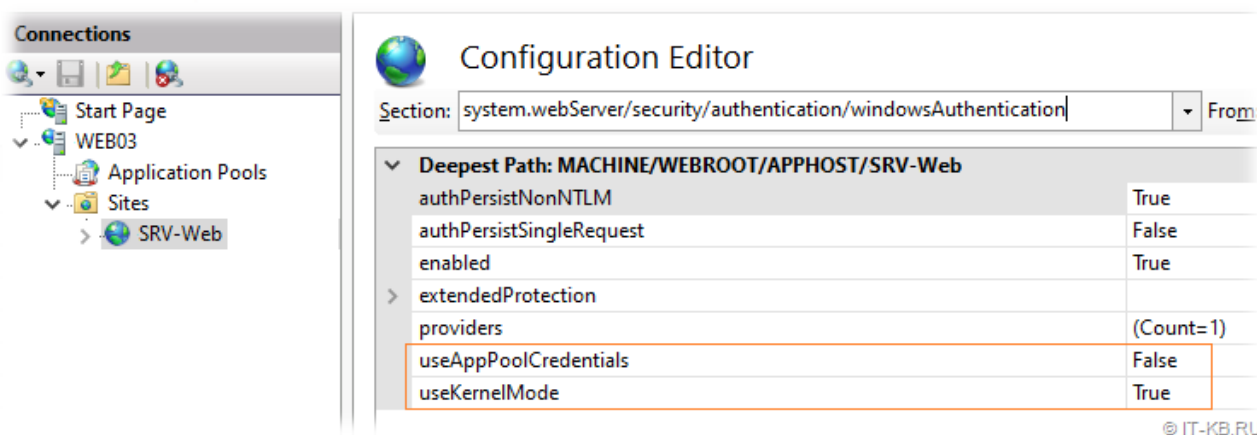
В свойствах аутентификации Windows убедимся в том, что провайдер **"Negotiate"** выставлен как приоритетный (в самом верху списка провайдеров). Если NTLM в чистом виде не нужен, можем его вообще удалить из списка включенных провайдеров.



© IT-KB.RU

Провайдер "Negotiate" будет приоритетно работать с протоколом Kerberos с возможностью понижения до протокола NTLM.

4) В редакторе правки конфигурации сайта "**Configuration Editor**" выбираем секцию **system.webServer/security/authentication/windowsAuthentication**. Здесь проверим, что опция "**useAppPoolCredentials**" выключена (False), а опция "**useKernelMode**" включена (True).



© IT-KB.RU

5) На клиентской машине проверяем обрабатывает ли аутентификация Kerberos.

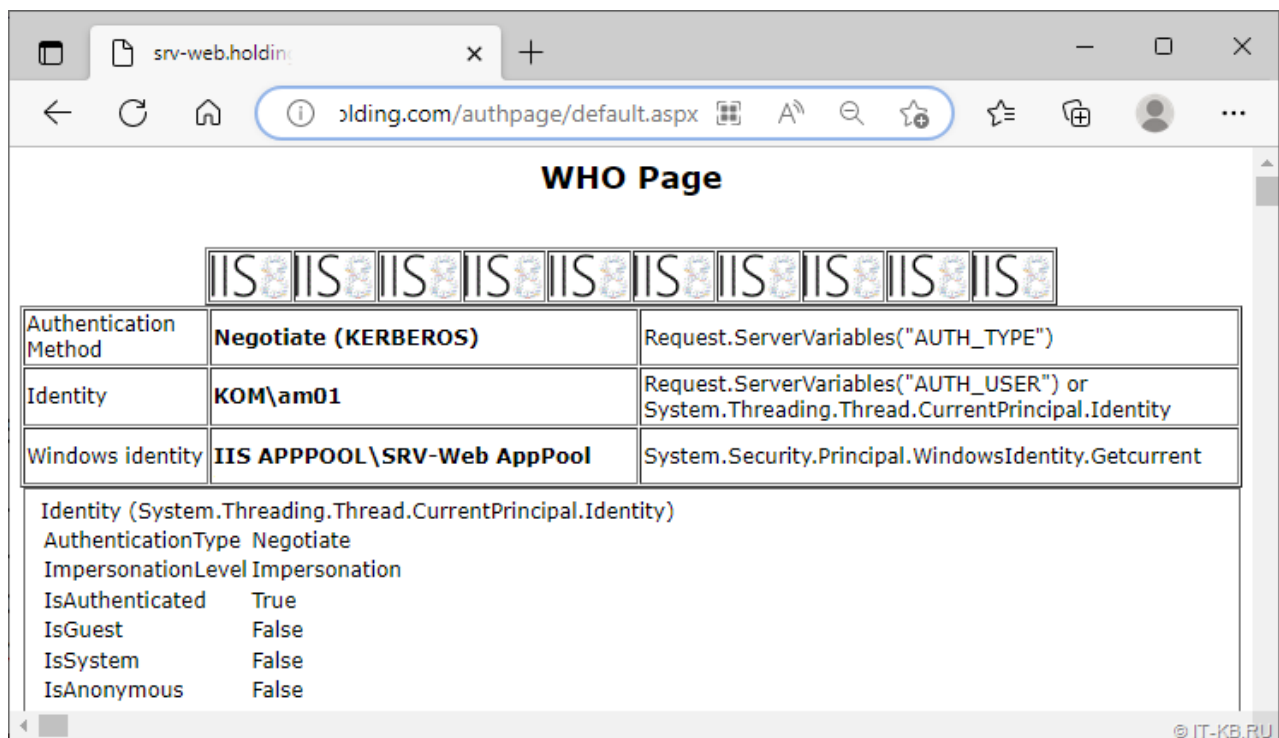
Перед проверкой, для чистоты эксперимента, на стороне веб-сервера перезапустим службу IIS (команда **iisreset**), а на клиентской машине закроем веб-браузер и очистим кеш билетов Kerberos в контексте текущего пользователя (команда **klist purge**).

После этого на клиенте снова откроем браузер и сначала просто пробуем открыть стартовую страницу сайта. Сайт должен открываться без каких-либо запросов аутентификации или ошибок.

В этот момент на клиентской системе с помощью команды **klist** можно проверить все кэшированные билеты Kerberos и убедиться в том, что в их списке появился билет для веб службы с SPN вида HTTP/srv-web.holding.com.

Дополнительно можем скачать диагностическую страницу, описанную в статье "[Microsoft Learn : ASP.NET Authentication test page](#)". Ссылка на архив на странице первоисточника битая, поэтому [вот копия](#).

Распаковываем содержимое архива в корневой каталог сайта в подкаталог /authpage и переходим в клиентском браузере по ссылке `srv-web.holding.com/authpage/default.aspx`



Authentication Method	Negotiate (KERBEROS)	Request.ServerVariables("AUTH_TYPE")
Identity	KOM\am01	Request.ServerVariables("AUTH_USER") or System.Threading.Thread.CurrentPrincipal.Identity
Windows identity	IIS APPPOOL\SRV-Web AppPool	System.Security.Principal.WindowsIdentity.GetCurrent

Identity (System.Threading.Thread.CurrentPrincipal.Identity)  
AuthenticationType Negotiate  
ImpersonationLevel Impersonation  
IsAuthenticated True  
IsGuest False  
IsSystem False  
IsAnonymous False

Как видим, доменный пользователь успешно аутентифицировался на веб-сайте с использованием аутентификации Kerberos.

После проверки на забываем удалить подкаталог /authpage из корневого каталога сайта.

Вариант 2. Пул IIS выполняется в контексте сервисной учётной записи gMSA

1) Создаём в домене сервисную учётную запись **gMSA** и разрешаем ей вход на веб-сервер. Пример создания учётной записи с помощью PowerShell можно найти [в статье Вики](#).

2) Регистрируем в домене уникальную запись **SPN** вида HTTP/<fqdn> для доменной учётной записи gMSA. Обратите внимание на то, что если SPN записи веб-службы ранее были заданы в свойствах учётной записи веб-сервера, то эти прежние SPN записи должны быть удалены, прежде чем их добавлять в свойства gMSA. Важно, чтобы соблюдалась уникальность SPN в рамках всего домена.

Проверим SPN-записи для компьютерной учётной записи веб-сервера и для учётной записи gMSA:

```
Get-ADComputer -Identity "WEB03" -Properties ServicePrincipalNames | Select-Object  
-ExpandProperty ServicePrincipalNames  
Get-ADServiceAccount -Identity "s-S004$" -Properties ServicePrincipalNames |  
Select-Object -ExpandProperty ServicePrincipalNames
```

В следующем примере мы удаляем пару SPN из компьютерной учётной записи веб-сервера (если, конечно, это требуется) и добавляем эту же пару записей в учётную запись gMSA, от имени которой в нашем случае работает пул IIS:

```
Set-ADComputer -Identity "WEB03" -ServicePrincipalNames @{Remove='HTTP/SRV-  
Web.holding.com', 'HTTP/SRV-Web'}  
Set-ADServiceAccount -Identity "s-S004$" -ServicePrincipalNames @{Add='HTTP/SRV-  
Web.holding.com', 'HTTP/SRV-Web'}
```

3) Настраиваем пул IIS для запуска от имени gMSA. При этом в окне указания учётной записи Identity не указываем пароль, а лишь указываем имя gMSA в формате вида DOMAIN\gMSA\$.



## Application Pools

This page lets you view and manage the list of application pools on the server. Application pools are associated with worker process isolation among different applications.

Filter:  Go  Group by: No Grouping

Name	Status	.NET CLR V...	Managed Pipel...	Identity	Applications
SRV-Web AppPool	Started	v4.0	Integrated	KOM\s-S004\$	1
DefaultAppPool	Started	v4.0	Integrated	ApplicationPoolIdentity	0
Advanced Settings					
<b>(General)</b>					
.NET CLR Version		v4.0			
Enable 32-Bit Applications		False			
Managed Pipeline Mode		Integrated			
Name		SRV-Web AppPool			
Queue Length		1000			
Start Mode		OnDemand			
<b>CPU</b>					
<b>Process Model</b>					
Generate Process Model Event Log Er					
Identity		KOM\s-S004\$			
Idle Time-out (minutes)		20			
Idle Time-out Action		Terminate			
Load User Profile		False			

© IT-KB.RU

4) В свойствах сайта в методах аутентификации выключаем анонимную аутентификацию, и включаем аутентификацию Windows.

В свойствах аутентификации Windows убедимся в том, что провайдер **"Negotiate"** выставлен как приоритетный (в самом верху списка провайдеров). Если NTLM в чистом виде не нужен, можем его вообще удалить из списка включенных провайдеров.

Connections

- Start Page
- WEB03
  - Application Pools
  - Sites
    - SRV-Web

### Authentication

Group by: No Grouping

Name	Status	Response Type
Anonymous Authentication	Disabled	
ASP.NET Impersonation	Disabled	
Forms Authentication	Disabled	HTTP 302 Login/Redirect
Windows Authentication	Enabled	HTTP 401 Challenge

Providers

Enabled Providers:

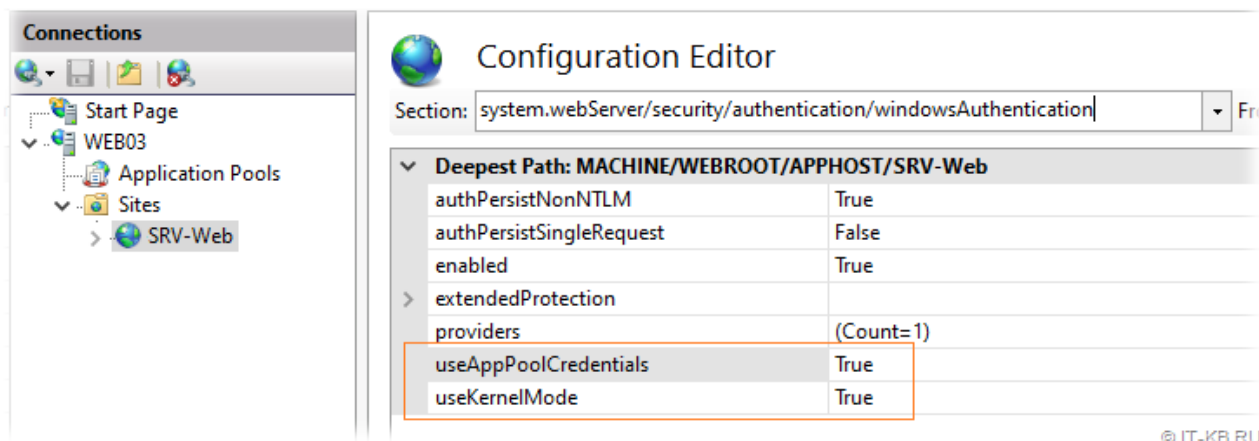
- Negotiate

Move Up

© IT-KB.RU

5) В редакторе правки конфигурации сайта "**Configuration Editor**" выбираем секцию **system.webServer/security/authentication/windowsAuthentication**. Здесь включаем опцию "**useAppPoolCredentials**" (True), и убеждаемся в том, что включена опция "**useKernelMode**" (True).

В некоторых статьях можно найти мнение о том, что опцию useKernelMode при работе пула от gMSA нужно выключать. Но есть также информация о том, что опцию useKernelMode при включённой опции useAppPoolCredentials можно не изменять, так как включённая опция useAppPoolCredentials имеет приоритет над опцией useKernelMode.



6) На клиентской машине проверяем обрабатывает ли аутентификация Kerberos.

Перед проверкой на стороне веб-сервера перезапустим службу IIS (команда **iisreset**), а на клиентской машине закроем веб-браузер и очистим кеш билетов Kerberos текущего пользователя (команда **klist purge**).

Снова переходим по ссылке [srv-web.holding.com/authpage/default.aspx](http://srv-web.holding.com/authpage/default.aspx) и смотрим, что покажет нам тестовая страница.



WHO Page		
Authentication Method	Negotiate (KERBEROS)	Request.ServerVariables("AUTH_TYPE")
Identity	KOM\am01	Request.ServerVariables("AUTH_USER") or System.Threading.Thread.CurrentPrincipal.Identity
Windows identity	KOM\s-S004\$	System.Security.Principal.WindowsIdentity.GetCurrent
Identity (System.Threading.Thread.CurrentPrincipal.Identity) AuthenticationType Negotiate ImpersonationLevel Impersonation IsAuthenticated True IsGuest False IsSystem False IsAnonymous False		

Обратим внимание на то, что теперь в поле Windows identity отображается учётная запись gMSA, от имени которой работает пул IIS.

## Этап 2. Настройка разных типов делегирования аутентификации на примере IIS

Описанной выше базовой настройки достаточно лишь для того, чтобы реализовать простую прямую аутентификацию Kerberos между клиентом и веб-сервером. Конфигурация усложняется в том случае, если возникает потребность предоставить доступ аутентифицированному клиенту веб-сервера к какому-либо стороннему ресурсу, находящемуся за веб-сервером, например, к базе данных на другом сервере с SQL Server или к файловой шаре на другом файловом сервере. В этом случае для поддержки, так называемого Double Hop, требуется настройка делегирования Kerberos. То есть, на уровне учётных записей домена мы должны разрешить фронтенд службе (веб-серверу) олицетворять пользователей на бэкэнд службе (например, файловом сервере).

С точки зрения Kerberos можно выделить три типа делегирования:

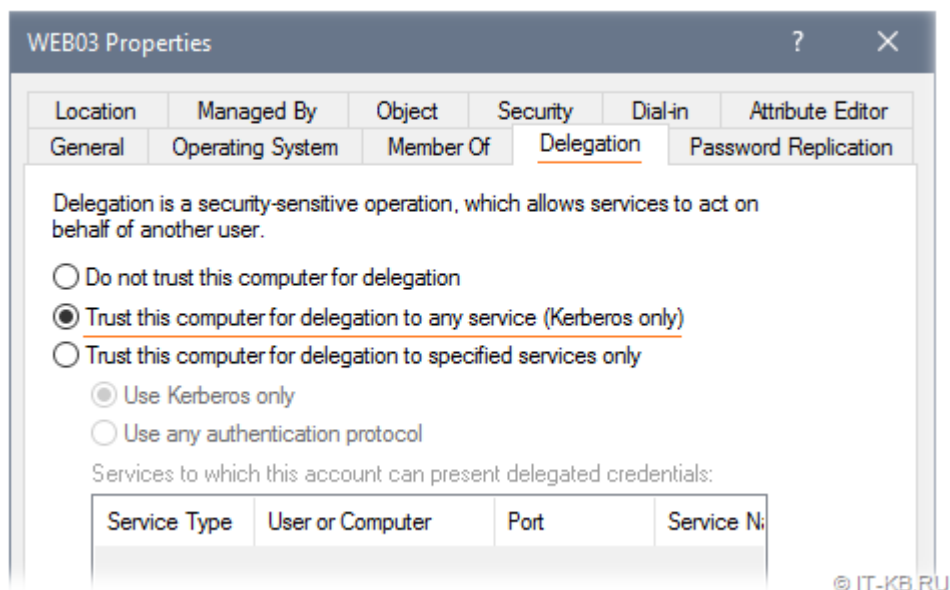
1. Полное делегирование (Unconstrained Delegation);
2. Классическое ограниченное делегирование (Constrained Delegation);
3. Ограниченное делегирование на основе ресурсов (Resource Based Constrained Delegation).

Тип 1. Полное делегирование (Unconstrained Delegation)

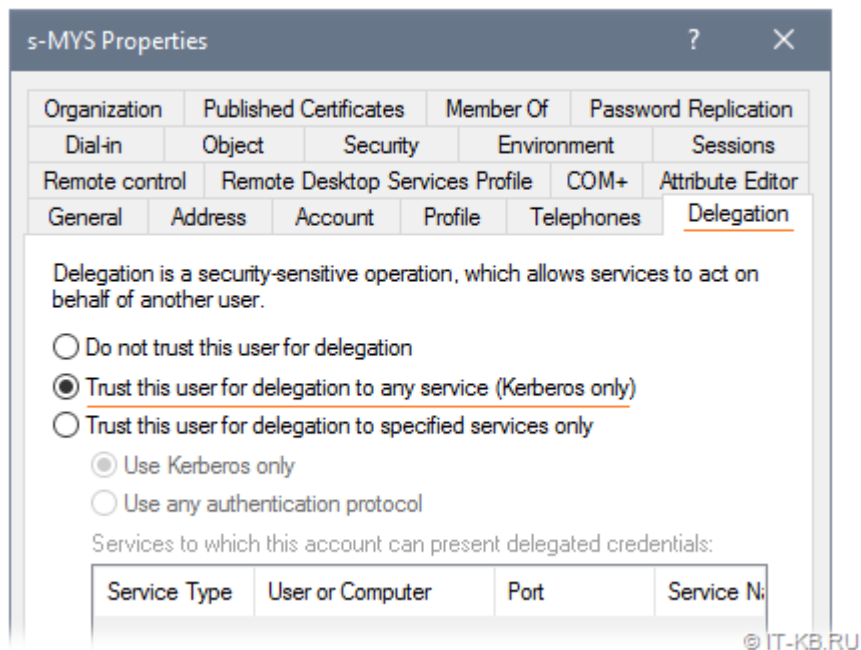
Для включения этого типа делегирования нужно обладать правами уровня администратора домена или привилегией **SeEnableDelegationPrivilege**. Включение этого типа делегирования влияет на содержимое атрибута **userAccountControl**

(включается флаг **TRUSTED\_FOR\_DELEGATION**).

В графической оболочке в свойствах учётных записей компьютеров на вкладке **"Delegation"** определяется вариантом **"Trust this computer for delegation to any service (Kerberos only)"**.



В свойствах учётных записей пользователей вкладка **"Delegation"** отображается только для пользователей, имеющих зарегистрированные SPN-записи в атрибуте **servicePrincipalName**. Здесь вариант полного делегирования называется по аналогии - **"Trust this user for delegation to any service (Kerberos only)"**.



Для сервисных учётных записей gMSA нет возможности управлять делегированием в графической оболочке, но можно сделать это с помощью PowerShell. Приведу примеры команд, используемых для управления делегированием gMSA.

Проверяем, включено ли полное делегирование:

```
Get-ADServiceAccount -Identity "s-S004$" -Property TrustedForDelegation
```

Включаем полное делегирование:

```
Set-ADServiceAccount -Identity "s-S004$" -TrustedForDelegation $true
```

Отключаем полное делегирование:

```
Set-ADServiceAccount -Identity "s-S004$" -TrustedForDelegation $false
```

Важно понимать то, что "Unconstrained Delegation" это самый "древний" и самый худший, с точки зрения безопасности, вариант делегирования Kerberos. Поэтому следует отказаться от его использования. Об опасности этого варианта написано не мало статей, например, ["Semperis : Unconstrained Delegation in Active Directory Leaves Security Gaps"](#).

Тип 2. Ограниченное делегирование (Constrained Delegation)

Для включения этого типа делегирования нужно обладать правами уровня администратора домена или привилегией **SeEnableDelegationPrivilege**.

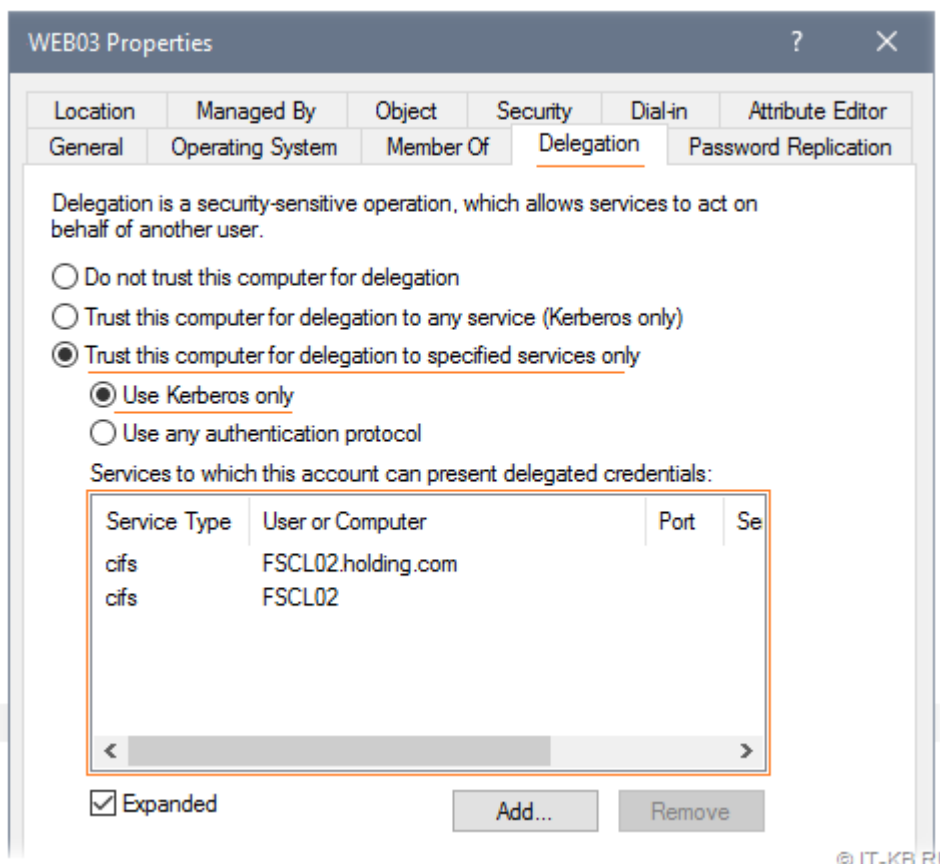
Фактически это делегирование настраивается с помощью изменения содержимого атрибута **msDS-AllowedToDelegateTo** для учётной записи, от имени которой работает фронтенд служба. В этот атрибут вписываются бэкэнд службы в формате SPN, в сторону которых фронтенд службе разрешено транслировать учётные данные аутентифицированных пользователей.

Порядок настройки ограниченного делегирования опять же зависит от того, в контексте какой учётной записи выполняется пул приложений IIS.

Вариант 1. Пул IIS выполняется в контексте учётной записи самого веб-сервера

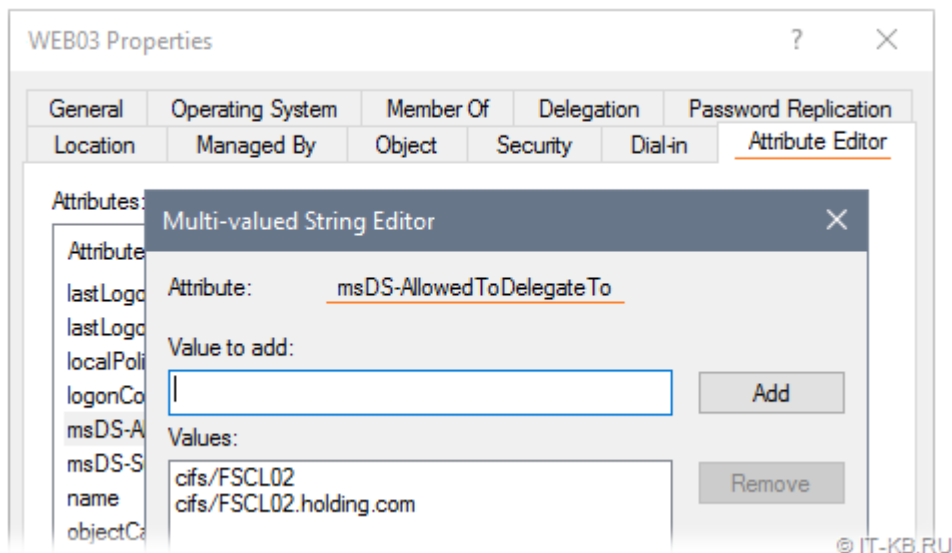
Например, у нас есть сайт IIS, пул которого работает в контексте учётной записи самого веб-сервера (Identity = ApplicationPoolIdentity) и мы подключаем в качестве виртуального каталога этого веб-сайта IIS файловую шару с отдельного файлового сервера, чтобы аутентифицированные пользователи сайта могли скачивать файлы с этой шары через этот виртуальный каталог.

В этом случае нам потребуется в свойствах компьютерной учётной записи веб-сервера выбрать вариант **"Trust this computer for delegation to specified services only" > "Use Kerberos only"** и в поле описания служб добавить службу **cifs** от файлового сервера.



При этом обратите внимание на то, что для возможности скачивания файлов с файлового сервера через веб-службу, нам потребуется обеспечить право на чтение в соответствующей файловой шаре как самим конечным пользователям, так и учётной записи веб-сервера. Если этого не сделать, то при попытке перехода по URL-ссылке, ведущей в подкаталоги шары файлового сервера, мы будем получать от веб-сервера ошибку HTTP 500. Как я понял, это связано с тем, что IIS пытается найти и прочитать файл web.config в конечном подкаталоге, и если учётной записи пула приложений IIS не хватает прав на чтение, чтобы выполнить эту операцию, то мы и получаем эту самую 500 ошибку.

После настройки делегирования, если мы заглянем в свойства атрибута **msDS-AllowedToDelegateTo** у учётной записи веб-сервера, то увидим указанные нами бэкэнд службы через вкладку "Delegation".



В принципе этой настройки достаточно, чтобы делегирование заработало и пользователи, успешно прошедшие аутентификацию Kerberos на веб-сайте IIS, могли скачивать файлы шары файлового сервера через виртуальный каталог веб-сайта.

Вариант 2. Пул IIS выполняется в контексте сервисной учётной записи gMSA

Если пул IIS работает в контексте учётной записи gMSA, то нам потребуется настроить делегирование не для учётной записи веб-сервера, а для учётной записи gMSA. Сделать это можно с помощью PowerShell, так как это описано в статье ["Microsoft Learn : Configuring Kerberos delegation for group Managed Service Accounts"](#).

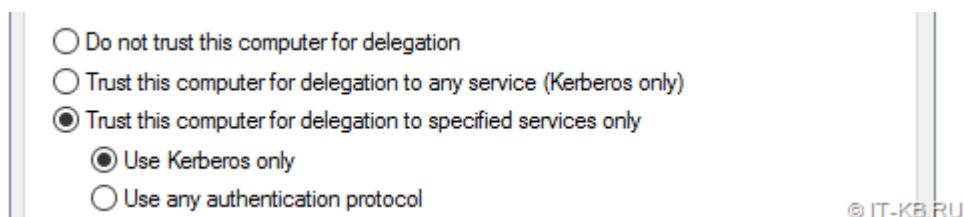
Чтобы проверить текущие настройки делегирования для учётной записи gMSA, выполним команду вида:

```
Get-ADServiceAccount -Identity "s-S004$" -Properties
TrustedForDelegation,TrustedToAuthForDelegation,msDS-AllowedToDelegateTo
```

Чтобы включить ограниченное делегирование с использованием только протокола аутентификации Kerberos выполним команду вида:

```
Set-ADAccountControl -Identity "s-S004$" -TrustedForDelegation $false -
TrustedToAuthForDelegation $false
```

Это аналогично выбору переключателя **"Trust this computer for delegation to specified services only" > "Use Kerberos Only"**.

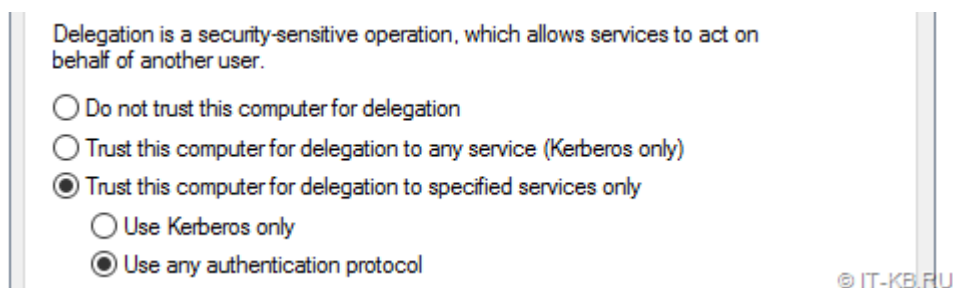


Это как раз то, что нужно сделать нам, исходя из нашего примера с фронтом на веб-сервере и бэкендом на файловом сервере.

В тех случаях, когда нужно включить ограниченное делегирование с использованием любых протоколов аутентификации, используется команда следующего вида:

```
Set-ADAccountControl -Identity "s-S004$" -TrustedForDelegation $false -  
TrustedToAuthForDelegation $true
```

Это аналогично выбору переключателя **"Trust this computer for delegation to specified services only" > "Use Any Authentication Protocol"**. Обратите внимание на то, что в этом случае в атрибуте **userAccountControl** включается флаг **TRUSTED\_TO\_AUTH\_FOR\_DELEGATION**.

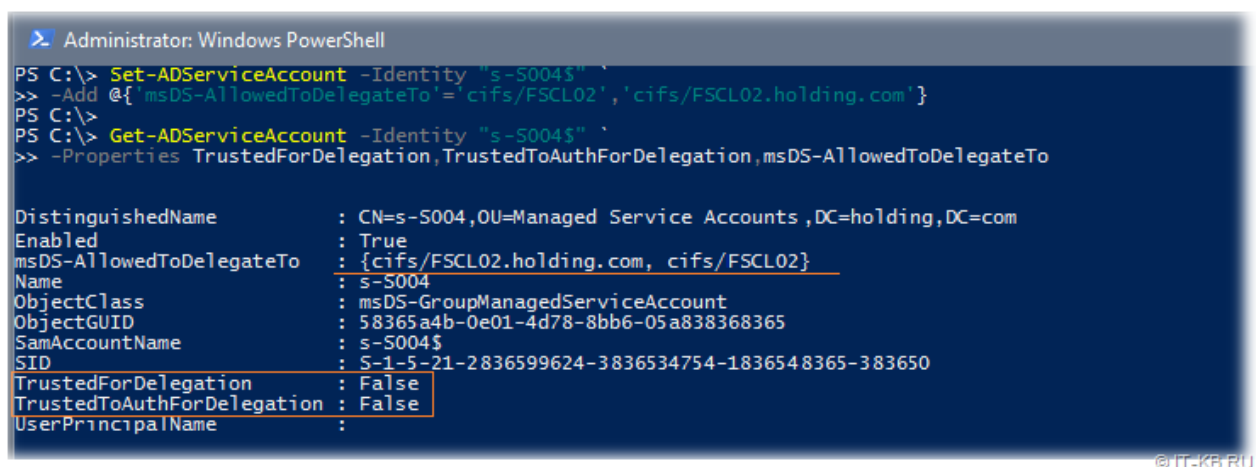


Следующим шагом нам нужно заполнить перечень SPN для разрешённых бэкенд служб в атрибуте **msDS-AllowedToDelegateTo** учётной записи gMSA. В нашем примере это делается следующей командой:

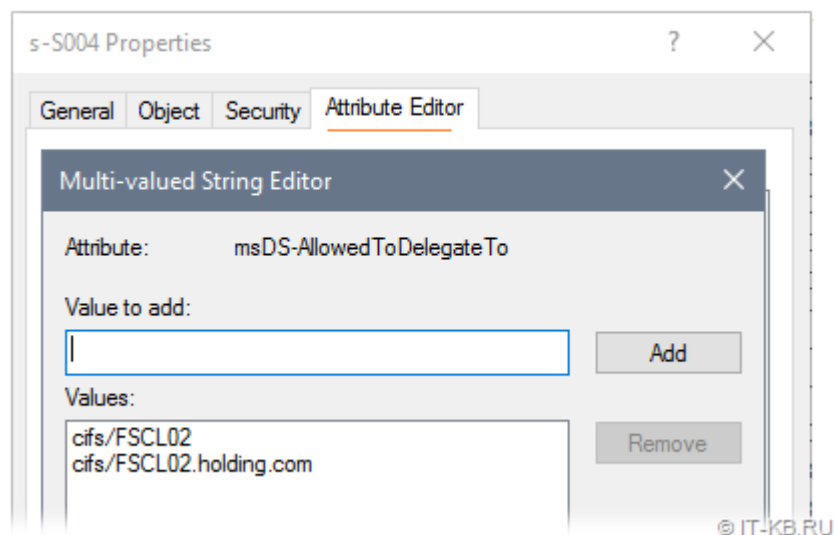
```
Set-ADServiceAccount -Identity "s-S004$" -Add @{'msDS-  
AllowedToDelegateTo'='cifs/FSCL02','cifs/FSCL02.holding.com'}
```

По окончании настройки снова проверяем свойства учётной записи gMSA:

```
Get-ADServiceAccount -Identity "s-S004$" -Properties  
TrustedForDelegation,TrustedToAuthForDelegation,msDS-AllowedToDelegateTo
```



Соответственно, если теперь в графической оболочке заглянем в свойства учётной записи gMSA, то увидим заполненный атрибут **msDS-AllowedToDelegateTo** (редактировать его при необходимости, можно и здесь, а не через PowerShell).



Обратите внимание на то, что, в рамках нашего примера, для возможности скачивания файлов с файлового сервера через веб-службу, нам потребуется обеспечить право на чтение в соответствующей файловой шаре как самим конечным пользователям, так и учётной записи gMSA, от имени которой выполняется пул IIS на веб-сервере.

Для вступления изменений в силу на веб-сервере перезапустим IIS командой **iisreset** (или просто перезагрузим веб-сервер).

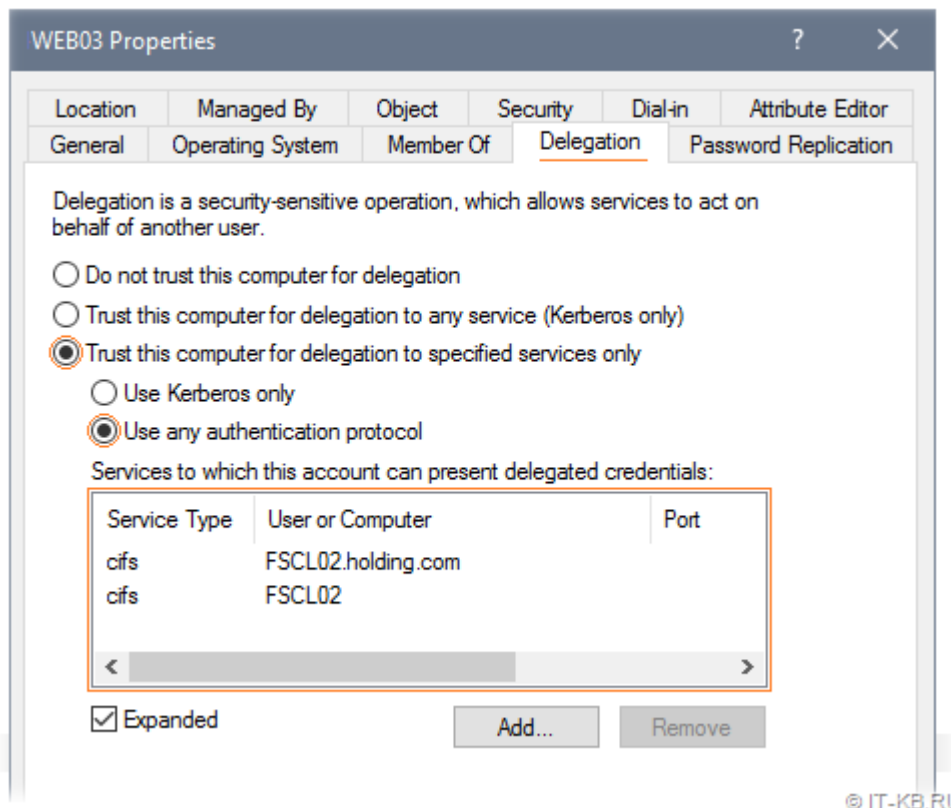
Теоретически проделанной настройки должно быть достаточно, чтобы делегирование заработало и пользователи, успешно прошедшие аутентификацию Kerberos на веб сайте IIS, могли скачивать файлы шары файлового сервера через виртуальный каталог веб-сайта. Но фактически это не так, так как на практике при ограниченном делегировании доступа к файловой службе (**CIFS**) мы столкнёмся с проблемой, описанной в статье ["KB2602377 : Constrained delegation for CIFS fails with ACCESS\\_DENIED error"](#).

Эта статья предлагает нам 2 обходных решения.

Первое обходное решение - это не использовать для ограниченного делегирования к службе CIFS сервисную учётную запись, а использовать учётную запись компьютера. То есть предполагается, что в нашем случае пул IIS (фронтэнд-служба), обращающийся к внешней сетевой шаре (бэкенд-служба) должен работать не в контексте gMSA, а в контексте системы (в IIS в свойствах пула параметр Identity = ApplicationPoolIdentity).

Если же всё-таки по каким-то причинам нам нужно, чтобы пул IIS работал от имени gMSA, то можно воспользоваться вторым обходным решением из вышеупомянутой статьи. Это решение подразумевает то, что помимо выполнения настройки делегирования для gMSA, нам потребуется сделать дополнительную "костыльную" настройку делегирования для компьютерной учётной записи веб-сервера.





После сделанных изменений перезагружаем веб-сервер WEB03.

В конечном итоге, в рассматриваемом нами примере, для того, чтобы заработало делегирование к CIFS для учётной записи gMSA (s-004\$), используемой для пула IIS на веб-сервере (WEB03) с подключением к шару стороннего файлового сервера (FSCL02), потребуется реализовать следующую "костыльную" и абсурдную, на мой взгляд, конфигурацию:

```
Administrator: Windows PowerShell

PS C:\> Get-ADServiceAccount -Identity "s-004$" -Properties * | `
>> Select SamAccountName,TrustedForDelegation,TrustedToAuthForDelegation,msDS-AllowedToDelegateTo | fl

SamAccountName           : s-004$
TrustedForDelegation      : False
TrustedToAuthForDelegation : False
msDS-AllowedToDelegateTo  : {cifs/FSCL02, cifs/FSCL02.holding.com}

PS C:\> Get-ADComputer -Identity "WEB03" -Properties * | `
>> Select SamAccountName,TrustedForDelegation,TrustedToAuthForDelegation,msDS-AllowedToDelegateTo | fl

SamAccountName           : WEB03$
TrustedForDelegation      : False
TrustedToAuthForDelegation : True
msDS-AllowedToDelegateTo  : {cifs/FSCL02, cifs/FSCL02.holding.com}
```

© IT-KB.RU

После этого пользователи веб-сервера действительно смогут скачивать файлы через виртуальный каталог, ссылающийся на шару на файловом сервере.

В случае, если нам потребуется вернуть учётную запись gMSA в исходное состояние и полностью отключить делегирование, то можно использовать команды следующего вида:



```
Set-ADAccountControl -Identity "s-S004$" -TrustedForDelegation $false -  
TrustedToAuthForDelegation $false  
Set-ADServiceAccount -Identity "s-S004$" -Clear 'msDS-AllowedToDelegateTo'
```

### Тип 3. Ограниченное делегирование на основе ресурсов (Resource Based Constrained Delegation)

---

Ограниченное делегирование на основе ресурсов Resource Based Constrained Delegation (RBCD) - это технология, доступная начиная с Windows Server 2012.

Если при использовании классического Constrained Delegation нам требуется настройка атрибутов msDS-AllowedToDelegateTo и userAccountControl для учётной записи, от имени которой работает фронтенд-служба (в нашем примере это учётная запись пула IIS и/или самого веб-сервера), то при настройке RBCD используется обратный подход, то есть делегирование настраивается в свойствах учётной записи бэкенд-сервиса (в нашем примере это учётная запись файлового сервера).

Более того, классическое Constrained Delegation для настройки требует привилегии уровня администратора домена, а для настройки RBCD достаточно иметь лишь доступ на изменение учётной записи бэкенд-сервиса (файлового сервера). При этом в ходе настройки делегирования нет необходимости указывать SPN служб, то есть доверительные отношения выстраиваются на уровне учётных записей домена (подразумевается доверие любых служб в рамках отношений между бэкэндом и фронтендом). Разумеется, это имеет как свои плюсы, так и свои минусы, но концептуально RBCD позиционируется, как более простой и удобный тип делегирования, чем классическое Constrained Delegation. Например, RBCD позволяет управлять делегированием между разными доменами Active Directory, в то время как классическое делегирование работает лишь в рамках одного домена.

Этот тип делегирования работает с помощью управления атрибутом **msDS-AllowedToActOnBehalfOfOtherIdentity** и настраивается только через PowerShell с помощью параметра *-PrincipalsAllowedToDelegateToAccount* в командлетах Set-ADUser, Set-ADComputer, Set-ADServiceAccount.

Рассмотрим типичные команды, используемые для настройки RBCD.

Проверяем текущее состояние настроек RBCD для учётных записей разного типа:

```
Get-ADComputer -Identity "<имя учётной записи компьютера бэкенд-службы>" -  
Properties PrincipalsAllowedToDelegateToAccount  
Get-ADServiceAccount -Identity "<имя учётной записи gMSA бэкенд-службы>$" -  
Properties PrincipalsAllowedToDelegateToAccount
```

Настраиваем делегирование на примере служб, работающих от имени учётных записей gMSA:

```
Set-ADServiceAccount -Identity "<имя учётной записи gMSA бэкенд-службы>$" -  
PrincipalsAllowedToDelegateToAccount $(Get-ADServiceAccount -Identity "<имя  
учётной записи gMSA фронтенд-службы>$")
```

Если нужно делегировать несколько фронтэнд-служб, то можно воспользоваться конструкцией следующего вида

```
$BackendSVC = "<имя учётной записи gMSA бэкэнд-службы>$"  
$FrontendSVC1 = Get-ADServiceAccount -Identity "<имя учётной записи gMSA фронтэнд-службы 1>$"  
$FrontendSVC2 = Get-ADServiceAccount -Identity "<имя учётной записи gMSA фронтэнд-службы 2>$"  
$Frontends = @($FrontendSVC1,$FrontendSVC2)  
Set-ADServiceAccount -Identity $BackendSVC -PrincipalsAllowedToDelegateToAccount  
$Frontends
```

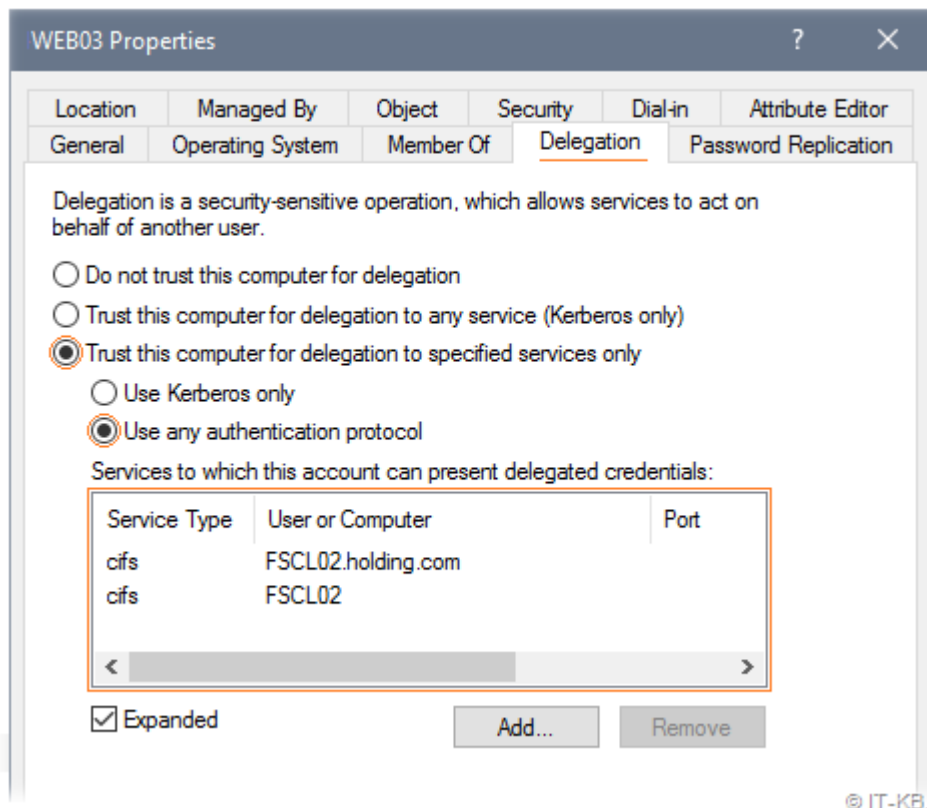
Экспериментируя с настройкой RBCD можно обнаружить не совсем очевидную особенность, которую можно отнести к разряду недостатков. Дело в том, что в атрибут **msDS-AllowedToActOnBehalfOfOtherIdentity** нельзя записать несколько значений разного типа. То есть, либо мы туда пишем учётные записи типа компьютер, либо учётные записи типа gMSA. При попытке указания коллекции из учётных записей разного типа, мы получим от PowerShell ошибку следующего вида:

```
Set-ADComputer : An AD Property Value Collection may only contain values of the  
same type. Specified value of type  
'Microsoft.ActiveDirectory.Management.ADComputer' does not match the existing type  
of 'Microsoft.ActiveDirectory.Management.ADServiceAccount'...
```

Вернёмся к нашему примеру, с вариантом, в котором для учётной записи gMSA (s-S004), от имени которой выполняется пул IIS на веб-сервере (WEB03), требуется разрешить передачу учётных данных пользователей веб-сервера на сторонний файловый сервер (FSCL02). В этом примере команда настройки RBCD будет иметь следующий вид:

```
Set-ADComputer -Identity "FSCL02"-PrincipalsAllowedToDelegateToAccount $(Get-  
ADServiceAccount "s-S004$")
```

И опять же, с теоретической точки зрения, такой настройки должно быть достаточно, чтобы пользователи смогли скачивать файлы с шары файлового сервера через веб-приложение IIS. Однако, на практике это работать не будет, предположительно, всё из-за той же проблемы, которую мы упомянули ранее со ссылкой на статью [KB2602377](#). То есть, в этом случае, чтобы делегирование заработало, опять потребуется сделать дополнительную "костыльную" настройку делегирования для компьютерной учётной записи веб-сервера.



После сделанных изменений перезагружаем веб-сервер WEB03.

В конечном итоге, в рассматриваемом нами примере, для того, чтобы заработало делегирование RBCD, потребовалось реализовать следующую, опять же "костыльную", конфигурацию:

```
Administrator: Windows PowerShell

PS C:\> Get-ADServiceAccount -Identity "s-S004$" -Properties * | `
>> Select SamAccountName, TrustedForDelegation, TrustedToAuthForDelegation, `
>> msDS-AllowedToDelegateTo, PrincipalsAllowedToDelegateToAccount | fl

SamAccountName           : s-S004$
TrustedForDelegation      : False
TrustedToAuthForDelegation : False
msDS-AllowedToDelegateTo  : {}
PrincipalsAllowedToDelegateToAccount : {}

PS C:\> Get-ADComputer -Identity "WEB03" -Properties * | `
>> Select SamAccountName, TrustedForDelegation, TrustedToAuthForDelegation, `
>> msDS-AllowedToDelegateTo, PrincipalsAllowedToDelegateToAccount | fl

SamAccountName           : WEB03$
TrustedForDelegation      : False
TrustedToAuthForDelegation : True
msDS-AllowedToDelegateTo  : {cifs/FSCL02, cifs/FSCL02.holding.com}
PrincipalsAllowedToDelegateToAccount : {}

PS C:\> Get-ADComputer -Identity "FSCL02" -Properties * | `
>> Select SamAccountName, TrustedForDelegation, TrustedToAuthForDelegation, `
>> msDS-AllowedToDelegateTo, PrincipalsAllowedToDelegateToAccount | fl

SamAccountName           : FSCL02$
TrustedForDelegation      : False
TrustedToAuthForDelegation : False
msDS-AllowedToDelegateTo  : {}
PrincipalsAllowedToDelegateToAccount : {CN=s-S004,OU=Managed Service Accounts,DC=holding,DC=com}
```

© IT-KB.RU

После этого пользователи веб-сервера действительно смогли скачивать файлы через виртуальный каталог, ссылающийся на шару на файловом сервере.

В случае, если нам потребуется вернуть учётную запись бэкенда в исходное состояние и полностью отключить делегирование RDCB, то для очистки атрибута **msDS-AllowedToActOnBehalfOfOtherIdentity** можно использовать команду следующего вида:

```
Set-ADComputer -Identity "FSCL02" -PrincipalsAllowedToDelegateToAccount $null
```

Общие выводы по типам делегирования

Итак, мы рассмотрели три типа делегирования и теперь можно оценить ситуацию в целом.

Однозначно можно сказать, что следует воздерживаться от применения на практике полного делегирования (Unconstrained Delegation), и если где-то требуется настройка делегирования, то всегда правильней использовать ограниченное делегирование (Constrained Delegation). Что-же касается ограниченного делегирования на основе ресурсов (Resource Based Constrained Delegation), то от него у меня двойные впечатления. С одной стороны RBCD более прост и действительно может оказаться полезен, например, в междоменных отношениях. С другой стороны, потеря контроля над чувствительными операциями делегирования со стороны администраторов домена в пользу администраторов отдельно взятых ресурсов, может привести к неприятным ситуациям с точки зрения информационной безопасности. В качестве наглядного примера можно ознакомиться со статьёй ["Decoder's Blog : Donkey's guide to Resource Based Constrained Delegation Exploitation – from simple user to \(almost\) DA"](#).

Есть также справедливое мнение, что, в принципе, все три типа делегирования потенциально опасны, и степень рисков может напрямую зависеть от правильности выбора типа делегирования и корректности настройки делегирования: ["harmj0y : Another Word on Delegation"](#).

Если у вас возникнет желание оценить уровень "развязности" настройки делегирования в текущей доменной инфраструктуре, то получить сводную картину по учётным записям с настроенным делегированием поможет [скрипт](#), позаимствованный из статьи ["Microsoft Learn : Get rid of accounts that use Kerberos Unconstrained Delegation"](#).

Что же касается примера с делегированием IIS->FS, которое мы сквозным образом рассматривали в ходе этой статьи, то у меня сложилось впечатление, что, в данном конкретном случае, для запуска пула IIS будет более правильно использовать контекст самой системы веб-сервера (Identity = ApplicationPoolIdentity), нежели контекст учётной записи gMSA. Так как преимущества, которые нам даёт использование gMSA, в данном конкретном случае, из-за костылей [KB2602377](#) "помножаются на ноль", делая конечную конфигурацию более громоздкой и менее безопасной.

---

## Дополнительные источники информации:

- [Jawahar Ganesh S : Setting up Kerberos Authentication for a Website in IIS](#)
- [Microsoft Learn : Troubleshoot Kerberos failures - Internet Information Services](#)
- [Database Administrators Stack Exchange : how to stop using sql server login credentials in a linked server?](#)
- [Microsoft Learn : Making the second hop in PowerShell Remoting - PowerShell](#)
- [hackndo : Kerberos Delegation](#)
- [Patrick Keisler : Setup Kerberos Constrained Delegation for Group Managed Service Accounts](#)
- [Josh Corrick : Kerberos Constrained Delegation with Group Managed Service Accounts](#)
- [Mark Southwell : Resource Based Kerberos Constrained Delegation](#)
- [Nichlas Falk : Re-becoming the securest constrained delegation we never weren't](#)
- [Jeff Warren : Resource-Based Constrained Delegation Abuse](#)
- [Daniel López Jiménez : You Do \(Not\) Understand Kerberos Delegation](#)

12 Оценок