Управляемые учётных записи служб - MSA - нововведение в Windows Server 2008 R2, улучшающее безопасность и упрощающее управление

atraining.ru/managed-service-accounts-msa

2013-10-04T08:56:16+08:00

Привет.

Введение

Управляемые учётных записи служб – MSA – нововведение в Windows Server 2008 R2, призванное ощутимо улучшить безопасность, упростить управление и принести ряд других, мелких, но хороших моментов в такую достаточно известную издревле штуку как управление учётными записями сервисов. Исторически эта задача решалась путём создания пользовательской учётки и обрезания ей лишнего в плане возможностей, но всё же такой подход изначально не очень корректен – и теперь у нас есть новый тип объектов, благодаря которому данная задача решается лучше.

Оглавление

Начнём.

Преимущества MSA

Преимуществ использования специализированной сервисной учётной записи, а не просто пользовательской, достаточно много.

Автоматическая смена паролей

MSA автоматически меняют свои пароли, снимая данную задачу с плеч администраторов. Вспомним, какие проблемы обычно бывают, когда администратор

запускает какой-либо сервис от доменной учётки:

1. Лень менять пароль, поэтому часто ставят пункт “Password never expires”
2. Лень создавать каждой учётке действительно сложный пароль, и менять его по расписанию на такой же сложный
3. Пароль для доменной учётной записи хранится локально на той системе, где он используется для старта сервиса, что крайне небезопасно (грубо говоря, каждый, кто имеет доступ на уровне администратора к системе, может получить все пароли всех учётных записей, от которых стартуют сервисы)
4. В случае смены пароля у сервисной учётной записи, которая используется в нескольких местах, надо везде единовременно поменять этот пароль вручную, а потом на всякий случай проверить работоспособность сервисов

MSA лишены всех этих проблем – они автоматически создают пароль и меняют его по указанному расписанию.

Стойкий пароль

Пароль, который создают Managed Service Account’ы, имеет длину 240 (двести сорок) символов, из которых половина – буквы английского алфавита, а половина – цифры и другие символы, так что на тему bruteforce можно не беспокоиться :). Поэтому админская лень побеждена простейшим способом – не надо регулярно придумывать пачки сложных паролей для пачки служебных учёток, всё делается само и очень неплохо по качеству делается.

Избыточные права

Обычной учётной записи пользователя, используемой как сервисная, надо крайне осторожно “урезать” права до необходимого минимума – например, убрать её из `Domain Users` и сделать какую-нибудь специфическую группу типа `Domain Services`, которой уже адресно разрешать только нужное, а не нужное, но имеющееся у пользователей – типа возможности интерактивного входа – запретить. Это мало кто делает вообще, а тщательно не делает практически никто. В случае с MSA ситуация меняется – изначально у этого типа учёток никаких “лишних” прав типа “локальный вход на машину” нет. Соответственно, выигрывается время и уменьшается риск в плане безопасности.

Ограничения по протоколу аутентификации

Пользовательскую учётку можно ограничить только от работы с протоколом LM – это если ей сделать пароль длиннее 14ти символов, тогда LM-хэш просто не сможет быть корректно создан, и будет использоваться набор NTLMv1/NTLMv2/Kerberos. Но вот ограничить ещё жестче – труднее. Начиная с 2008 R2 возможность полностью выключить семейство протоколов NTLM появилась, но уровень хардкорности этого решения и количество трудностей, с которым сталкиваются администраторы, крайне велик – скажу проще, я знаю только одну инфраструктуру (исключая нашу фирменную, конечно :)), где NTLM в домене изжили полностью и всё работает на

Kerberos 5 r6. Так вот, у MSA можно штатно ограничить конкретную учётку конкретным протоколом аутентификации. Много интересного? Достаточно, чтобы попробовать. Вначале – то, для чего MSA сделаны и работают:

Продукты и сервисы, поддерживающие MSA

Этот список специфичен, но достаточно велИк.

- IIS (можно например запускать application pool'ы от MSA)
- AD LDS (можно запускать экземпляры AD LDS от MSA – правда, там надо будет кое-что дополнительно прописать в реестре, но работать будет)
- SQL Server 2008 R2 (в документации указано, что с ним не работает; однако, с 2008 R2 SP1 нормально работает – что с дефолтным, что с именованными экземплярами)
- Часть сервисов Exchange 2010 (кроме случаев CAS-кластера и ряда сценариев с Edge/HUB)

 Как понимаете, в ряде сценариев MSA крайне полезны, в ряде – просто полезны. Крайне полезны, допустим, для IIS – вот у Вас, например, несколько десятков сайтов. Каждому надо свой пул – это своя учётка, со своими правами, своей периодичностью ручной смены паролей, своими задачами "обрезания прав чтобы ничего лишнего". MSA для Вас в таком сценарии прекрасный выход – разово создали, identity у пулов разные, пароли разные, сами раз в месяц меняются, действий никаких не надо, урезать в правах "чтобы друг к другу или куда ещё случаем не зашли" – не надо. Сплошные преимущества. Перейдём к подготовке к работе с MSA.

Подготавливаем систему к работе с MSA

Данная технология достаточно новая, поэтому для её использования нужно следующее:

1. ОС на ядре NT 6.1 и выше, исключая домашние редакции Windows 7 (т.е. только Professional / Enterprise / Ultimate)
2. Установленный .NET 3.5 и выше
3. Модуль администрирования Active Directory для PowerShell
4. Установленный патч 2494158

Это всё (кроме патча, конечно) автоматически выполняется на контроллере домена на базе Windows Server 2008 R2, но вот для рядового сервера или рабочей станции надо будет добавлять данные компоненты. Также, если домен и лес работают не на уровне Windows Server 2008 R2 (т.е. нет ACTIVE_DIRECTORY_V61 в фичах LDAP на DC), то уровень необходимо поднять до 4го, путём привычных `adprep /forestprep` + `adprep /domainprep`.

Создаём MSA

MSA создаётся в Active Directory. Для них даже есть специальный контейнер, расположенный в доменном разделе и называющийся `Managed Service Accounts`. Его не видно в обычном варианте просмотра, если смотрите через Active Directory Users & Computers – включите в меню View расширенный вариант просмотра.
Примечание: Этот контейнер – место по-умолчанию, а вообще можно создать MSA в любом месте domain partition, работать будет

Создать MSA можно двумя способами – через PowerShell и через ADSI. Второй способ хуже, т.к. новорожденный объект не будет иметь всех минимально необходимых полей и будет нефункционален, поэтому мы рассмотрим вариант создания через PowerShell. Кто хочет ~~потра~~попробовать через ADSI – создайте объект класса `msDS-ManagedServiceAccount` и внимательно заполните все специфичные атрибуты. Для создания MSA мы будем использовать командлет `New-ADServiceAccount`. Какие у него будут параметры?
Примечание: Всех параметров – много. И у PowerShell – прекрасный help. Поэтому я рассматриваю те, которые действительно нужно учитывать при создании любой учётной записи, и те, про которые есть, что добавить специфического. Опциональные параметры вида `-Description` Вы отлично и с первого раза укажете верно, я уверен. :)

-Name

Это единственный обязательный параметр, т.е. самый простой вариант данного командлета выглядит как `New-ADServiceAccount -Name` *`имя учётки`*. Притом параметр `-Name` можно не задавать, а просто написать `New-ADServiceAccount` *`имя учётки`* – система поймёт, что если параметр единственный, то Вы просто хотите создать MSA с указанным именем, в контейнере по умолчанию и с настройками по умолчанию.

-AccountPassword

Если всё же очень хочется, то можно задать пароль у MSA руками. Однако, надо учесть, что MSA, являясь одновременно и объектом класса “пользователь”, и объектом класса “компьютер” (в официальной документации Microsoft изящно называет это `quasi-computer object`), подпадает под ограничения на учётные записи, накладываемые политиками Active Directory для данного домена. Т.е. если в политике стоит длина пароля от 10 символов, то выдать MSA пароль вида `P@ssw0rd` не получится. Если всё же захочется ввести пароль, то его надо будет вводить как SecureString – т.е. хэшем. Так безопаснее. Но с клавиатуры не очень удобно, поэтому если всё ж надумаете – вот так можно: `-AccountPassword (Read-Host -AsSecureString "AccountPassword")`

-AuthType

Имеет всего два варианта – `Negotiate` и `Basic`. Как понятно, `Basic` – не самый лучший вариант, и пригоден он лишь для сценария “службе надо показать кому-то plain-text пароль внутри SSL/TLS сессии”. Такое иногда бывает, в том же Exchange Server 2013, поэтому совсем сбрасывать со счетов такой вариант не стОит. `Negotiate` установлен по умолчанию.

-DisplayName

Это – то имя, которое будет отображаться в Active Directory. Возможно, Вам будет удобнее сделать его развёрнутым – вида “SQL2008 на buhsrv01”, потому что работать с MSA Вы все равно будете по её имени – если задали, допустим, `-Name svc28`, а домен называется `atraining.local`, то её “технологическое имя” так и будет – `ATRAINING\svc28$`. Доллар в конце – ну, потому что это ж quasi-компьютер :)

-Instance

Удобный параметр, если надо “клонировать” MSA. Суть проста – можно при создании MSA указать через этот параметр другую, уже существующую; в обязательном порядке надо будет задавать только имя, остальные параметры, если Вы их не укажете явно, скопируются из указанной в `-Instance` MSA.

-PassThru

Если делаете pipe, то надо учитывать, что по умолчанию командлет не возвращает никаких значений. Хотите его попросить вернуть их – введите этот параметр, тогда результатами выполнения можно будет воспользоваться в следующем командлете.

-Path

DN того контейнера, в котором должна появиться создаваемая MSA, если Вас не устраивает дефолтный `CN=Managed Service Accounts,DC=...`

-SAMAccountName

Вот тут начинается интересное. Этот параметр позволяет “перекрыть” фактически используемое имя учётки – т.е. если Вы зададите и `-Name` и `-SamAccountName`, то имя для отображения в Active Directory будет из `-Name`, а идентификатор для логина – из `-SamAccountName`. Технологически максимум длины – 256 символов; однако, в документации существует неточность и указано, что “в целях совместимости со старыми ОС идентификатор не следует делать более 20 символов”. По факту – больше 14 не надо, т.е. где-то это всё упирается в древний код SAM’а, где ещё помнят времена NBName и NBNS/WINS. Экспериментально проверено, что ряд сервисов Exchange 2010 с установленным SP2 не может корректно запуститься с MSA при имени >14 символов. При подсчёте количества символов в имени не забудьте, что если Вы вручную не добавите доллар в конце – он добавится сам, и он тоже – символ.

-ServicePrincipalNames

Когда учётка MSA попробует “пойти наружу” с того сервера, на котором она будет работать, её могут спросить – от какого имени она действует? Соответственно, ей надо будет предъявить один из существующих у неё SPN. Допустим, Вы запускаете SQL Server 2008 R2 от MSA – Вам надо явно задать SPN, из-под которого будет идти работа. Это можно сделать так: `-ServicePrincipalNames -@{Add="svc_sql1\atraining.ru:1433"}` Если надо несколько – так: `-ServicePrincipalNames -@{Add="svc_sql1\host1.atraining.ru:1433"}; {Add="MSSQLSVC/host1.atraining.ru:5050"}` Выполнили? ОК. В принципе, всё, что нужно, Вы сделали – учётка создана, Вы можете открыть её и просмотреть, чтобы убедиться, что заданные Вами параметры повлияли на нужные атрибуты в Active Directory. Теперь нам надо назначить MSA хозяина – того, кто будет её беречь в жизненных невзгодах, и менять ей пароль, и всё другое, так необходимое в доменной жизни.

Привязываем MSA к серверу

Это будет делать командлет `Add-ADComputerServiceAccount`. По сути дела, эта операция достаточно проста – она добавляет сведения о том, что надо заботиться об этой MSA, в учётную запись computer’a в Actove Directory. Поэтому самая простая форма запуска будет выглядеть так: `Add-ADComputerServiceAccount -Identity sqlsrv15 -ServiceAccount svc_sql_15` (добавляем на сервер с именем `sqlsrv15` MSA с именем `svc_sql_15`) Компьютер при этом не должен быть в онлайне, и вообще живым – добавление произойдёт в любую существующую в Active Directory учётную запись компьютера. Факт добавления отследить просто – зайдите в атрибут с названием `msDS-HostServiceAccount` у данного компьютерного объекта и визуально убедитесь в наличии там DN добавляемой MSA. Как понятно, этот атрибут – массив, потому что один компьютер может поддерживать на себе несколько MSA. Что можно добавить в этой команде из полезного? Довольно мало; разве что упомяну, что можно сразу добавить несколько учёток, написав их имена в `-ServiceAccount` через запятую – например `-ServiceAccount svc_sql_15,svc_sql_16`. Интереснее – добавление данной учётки на сервер. Именно после этой операции он не просто поймёт, что к нему “приписана” учётная запись, но и начнёт иметь возможность выполнять с ней полезные действия.

Добавляем MSA на сервер

Первым делом – не забудем, что речь про сервер, на котором есть .NET 3.5.1, Powershell 2.0 и патч, про который я упоминал в самом начале. После – командлет `Install-ADServiceAccount`. В базовом варианте он прост: `Install-ADServiceAccount -Identity имя MSA` Но есть специфика – например, можно задать пароль MSA в явном виде, используя параметр `-PromptForPassword`, если хочется задать пароль в интерактивном режиме: `Install-ADServiceAccount -Identity имя MSA -PromptForPassword` Или задать пароль сразу: `Install-ADServiceAccount -Identity имя MSA -AccountPassword (ConvertTo-SecureString`

`-AsPlainText "пароль" -Force)` Зачем это может понадобиться? Есть редкий сценарий, когда Вы используете MSA на сервере, который имеет доступ только до RODC. В этом случае сервер не сможет “сходить и обновить” пароль у MSA – и надо будет задать пароль у MSA при инсталляции вручную. Установили? Теперь используем.

Используем MSA

Разные варианты использования – разные особенности.

MSA для пулов IIS

Для того, чтобы пул запустился от имени MSA, достаточно открыть консоль управления IIS, выбрать Application Pools, выбрать нужный пул, у него – Advanced Settings, там в разделе Process Model выбрать Identity, и задать имя учётной записи вида *ДОМЕН\MSA*. Пароль должен быть пустым, система сообразит, про что речь. Не забудьте значок доллара в конце имени MSA :). После – перезапустите пул и убедитесь, что всё ОК.

MSA для NT-сервисов

Ситуация аналогична – задайте имя MSA и не задавайте пароль. Помните, что это, помимо автоматизации обновления пароля – нормальная учётная запись, с SID’ом, членством в группах, SPN’ами, привилегиями и прочим.

MSA для AD LDS

Застрельная процедура с бубном описана здесь, к ней добавить особо нечего: http://technet.microsoft.com/ru-ru/library/ff641729(WS.10).aspx. Разве что статья старая, MSA с SQL Server уже работают. В общем-то всё из интересного. Остальное так же, как у обычных учётных записей. Другие операции у MSA нужны реже, но про них тоже надо упомянуть.

Сбрасываем пароль MSA

Если потребовалось всё же вручную сбросить пароль у MSA – это просто: `Reset-ADServiceAccountPassword -Identity имя учётки` Сбросить пароль, как понятно, можно только у той учётки, которая была установлена на данный сервер – сервер не может пойти в AD и поменять пароль у “чужой” MSA.

Добавляем MSA в группу Active Directory

MSA точно так же, как и любая другая учётка, может состоять в группах. Добавить её можно через стандартный графический интерфейс Active Directory, либо через командлет: `Add-ADGroupMember "имя группы" "DN MSA"` Ничего особенного, в общем.

Меняем SPN'ы у MSA

Это несложно – достаточно зайти в свойства объекта MSA в Active Directory, и поправить там поле `servicePrincipalName`. Можно и через PowerShell: `Set-ADServiceAccount -Name имя -ServicePrincipalNames @{Add="добавляемый SPN"}` Как понятно, вместо Add могут быть и другие операции. Replace, например, будет такого вида: `Set-ADServiceAccount -Name имя -ServicePrincipalNames @{Replace="заменяемый SPN","новый SPN"}` Синтаксис здесь будет такой же, как и у стартового задания SPN.

Удаляем MSA

Удаление MSA состоит из трёх разных операций – удаление с локальной машины, удаление из computer account'а и удаление записи MSA как таковой из Active Directory. Если Вам надо просто привязать MSA к другому серверу, то надо сделать только одну операцию – зайти на локальный сервер и выполнить

```
Remove-ADServiceAccount –Identity svc_sql_15
```

Для чистоты, конечно, ещё нужно зайти в атрибуты этого объекта в Active Directory и убедиться, что там из списка привязанных MSA всё тоже удалилось. Если нет – удалить вручную, редактированием атрибута.

Ну а совсем удалить MSA несложно – можно через ADUC, можно через ADSI, да хоть через ldp.exe – но раз уж всё через PowerShell, то надо действовать так:

```
Remove-ADComputerServiceAccount –Identity host1.atraining.local -ServiceAccount svc_sql_15
```

Вместо заключения

MSA – крайне полезная и несложная штука. Она и увеличивает безопасность, и упрощает администрирование, так что грех не использовать. Я бы рекомендовал Вам перейти на MSA везде, где это технически возможно.

Но, несмотря на то, что технология полезная и в этом варианте, и ощутимое КПД от её использования будет, в Windows Server 2012 у MSA появилось ещё больше возможностей – это и расширенный функционал, и поддержка учётных записей для групп (т.е. веб-кластеры работать с MSA смогут), и многое другое. Это – в следующей статье про MSA.

Удач!