Kerberos для специалиста по тестированию на проникновение. Часть 5. Делегирование, ограниченное на основе ресурсов

ardent101.github.io/posts/kerberos_rbcd

October 20, 2022

октября 20, 2022 · 8 мин · Ardent101

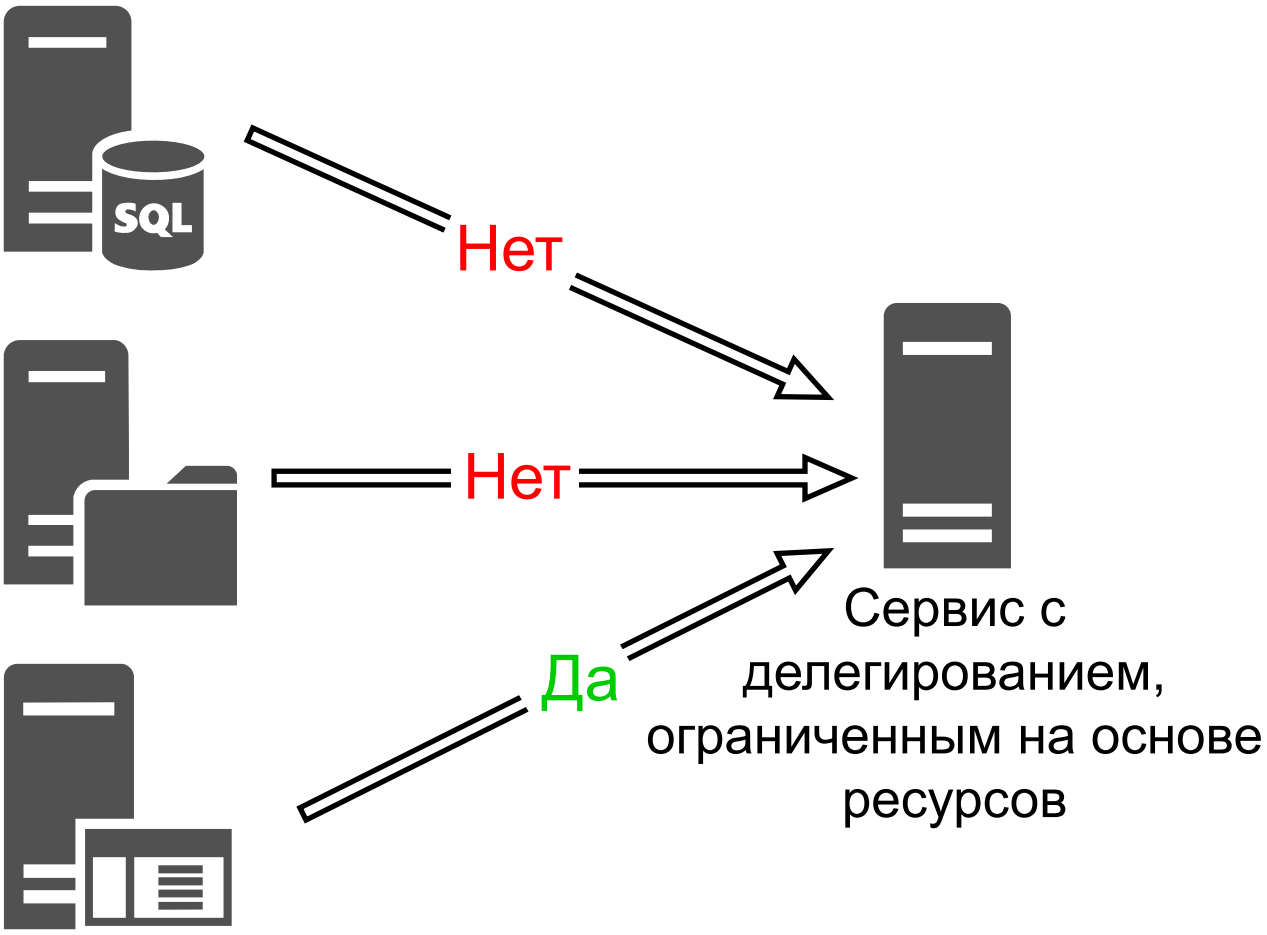
Ранее были рассмотрены механизмы неограниченного и ограниченного делегирования в Active Directory с помощью протокола Kerberos. Теперь перейдем к наиболее интересному с точки зрения проведения атак виду делегирования, а именно ограниченному на основе ресурсов.

Устройство ограниченного на основе ресурсов делегирования

Делегирование, ограниченное на основе ресурсов, появилось в Windows Server 2012.

Для настройки неограниченного и ограниченного видов делегирования требовалось наличие привилегии *SeEnableDelegation*, которой по умолчанию обладали только учетные записи с правами уровня администратора домена. При делегировании, ограниченном на основе ресурсов, сервис напротив самостоятельно определяет, кто может обращаться к нему от имени других пользователей.

> Примечание: далее вместо “*ограниченное на основе ресурсов делегирование*” может использоваться общепринятая аббревиатура RBCD (от англ. Resource Based Constrained Delegation)



Общая идея делегирования, ограниченного на основе ресурсов

Пояснение простыми словами - раньше назначался ответственный, который сам решал к кому и от имени кого он может обращаться. Теперь каждый сам решает, кто может обращаться к нему от имени других.

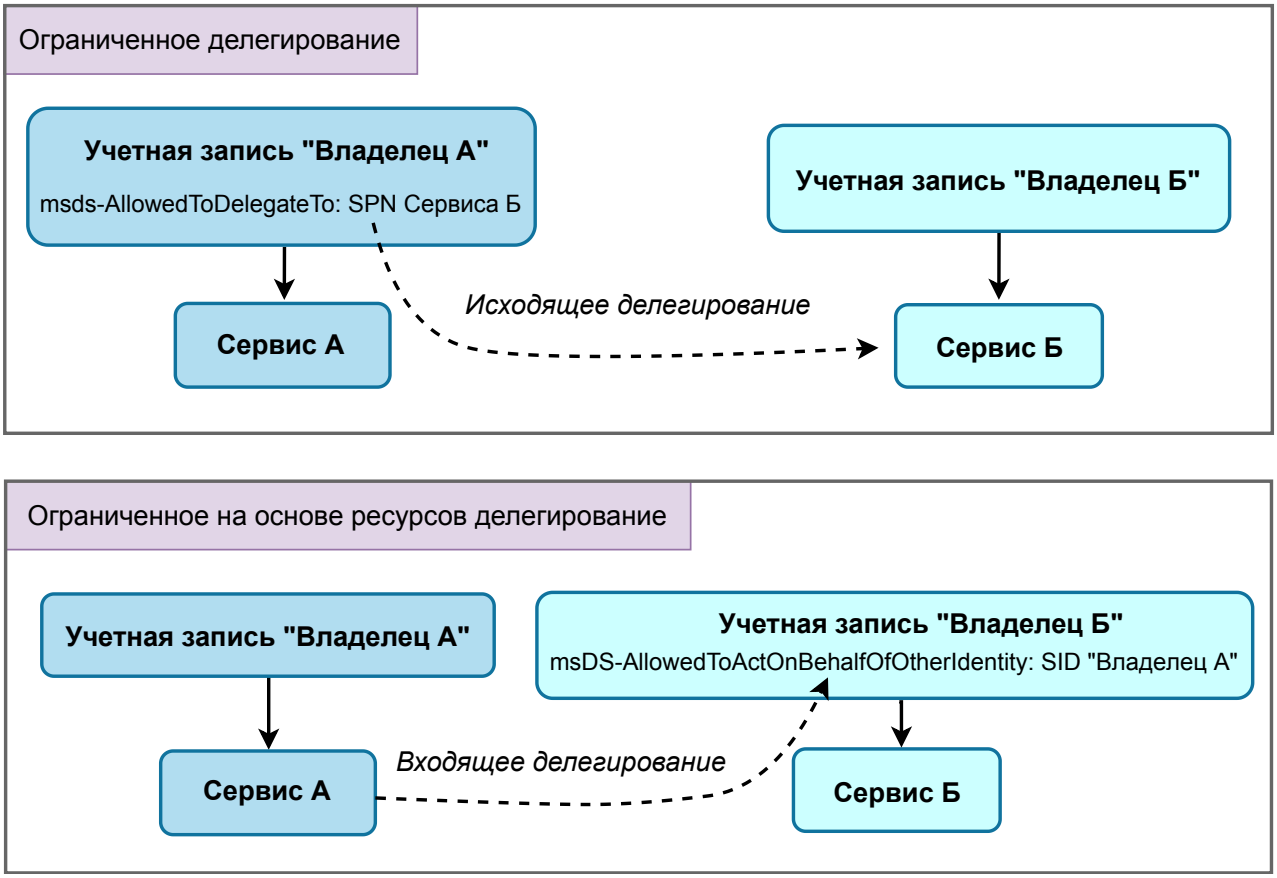
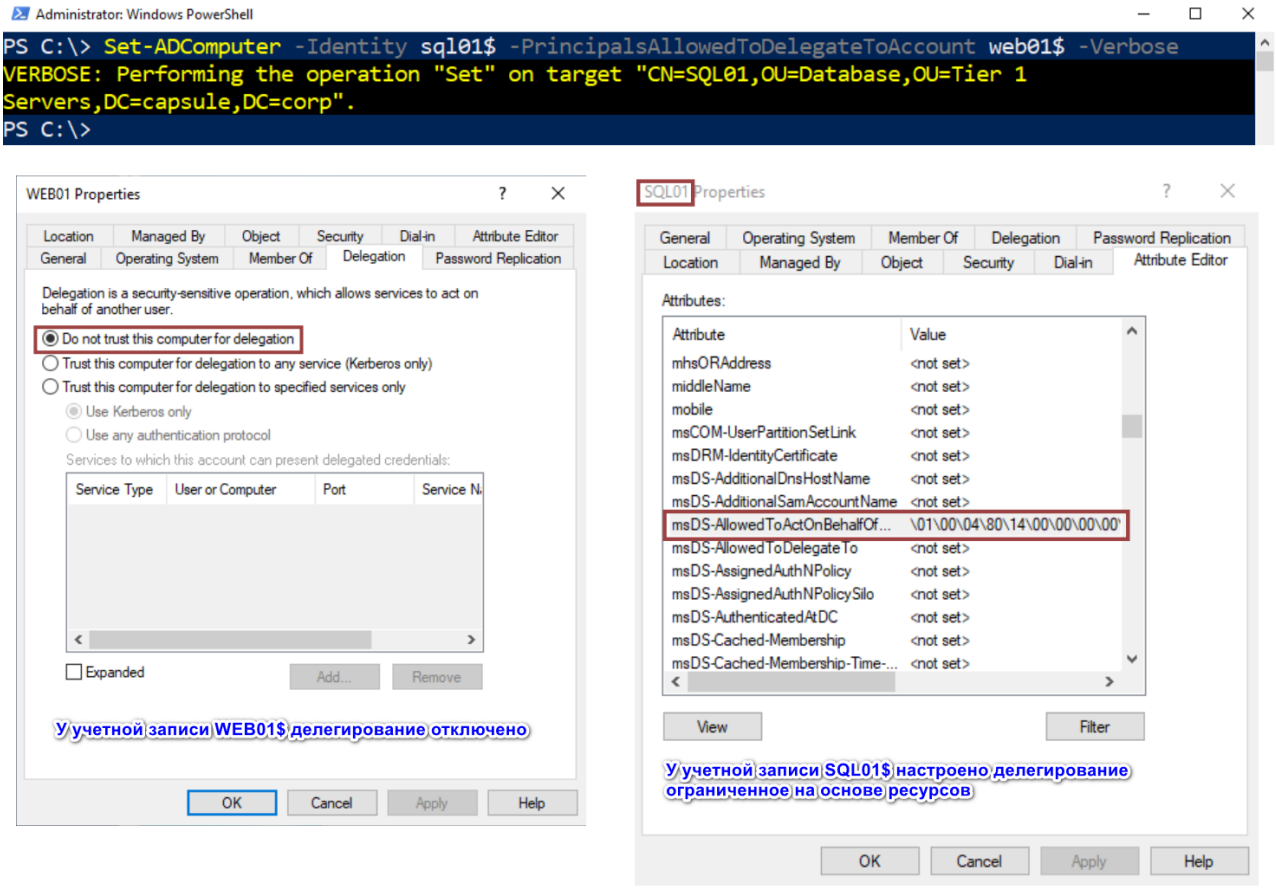


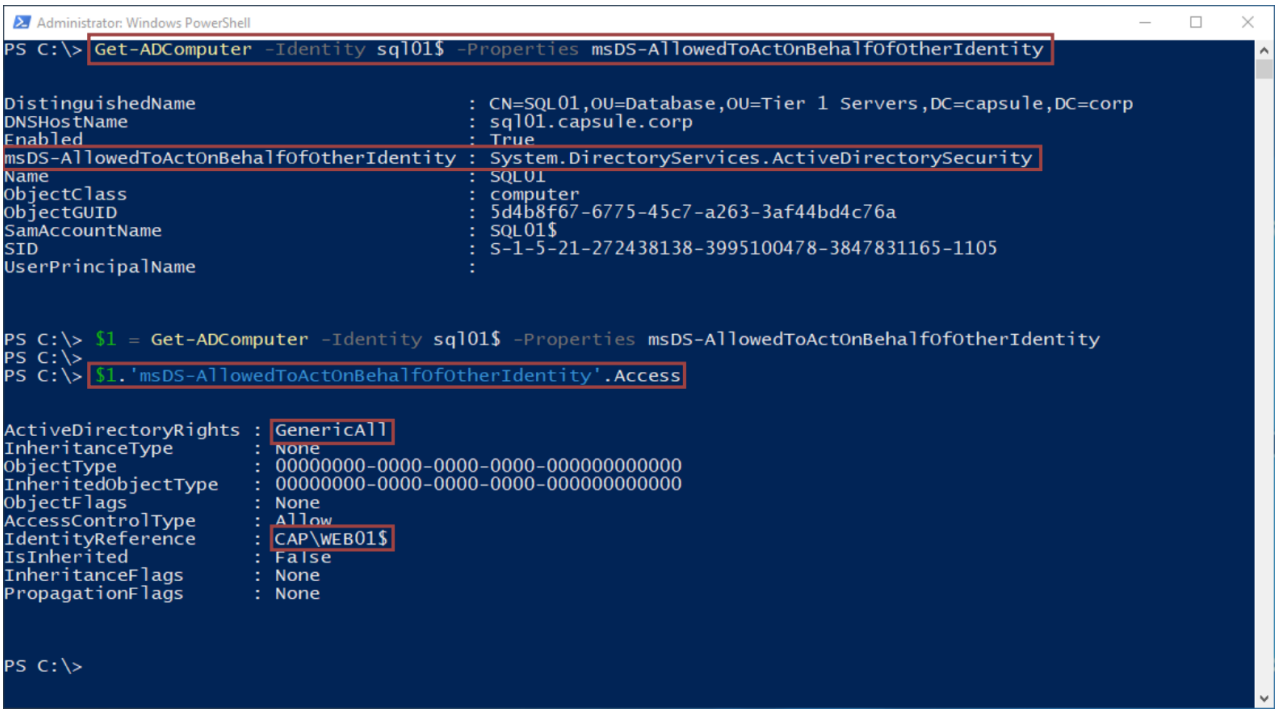
Иллюстрация различий между ограниченным делегированием и RBCD

Для настройки RBCD требуется право на запись в атрибут *msDS-AllowedToActOnBehalfOfOtherIdentity* учетной записи. В указанном атрибуте записываются и хранятся в двоичном виде идентификаторы безопасности (SID) учетных записей, сервисам которых разрешено олицетворять других пользователей при обращении к сервисам учетной записи с настроенным RBCD.

Рассмотрим пример. Удобный графический интерфейс для настройки RBCD отсутствует, поэтому приходится использовать Powershell:



Пример настройки у SQL01 RBCD для WEB01



Значение атрибута msDS-AllowedToActOnBehalfOfOtherIdentity в читаемом виде

Для работы механизма ограниченного на основе ресурсов делегирования также используются расширения *S4U2Self* и *S4U2Proxy*.

Рассмотрим работу указанных расширений при RBCD на следующем примере: пусть у учетной записи “Владелец Б” настроено RBCD, позволяющее **любому** сервису учетной записи “Владелец А” олицетворять незащищенных пользователей при обращении к **любому** своему сервису.

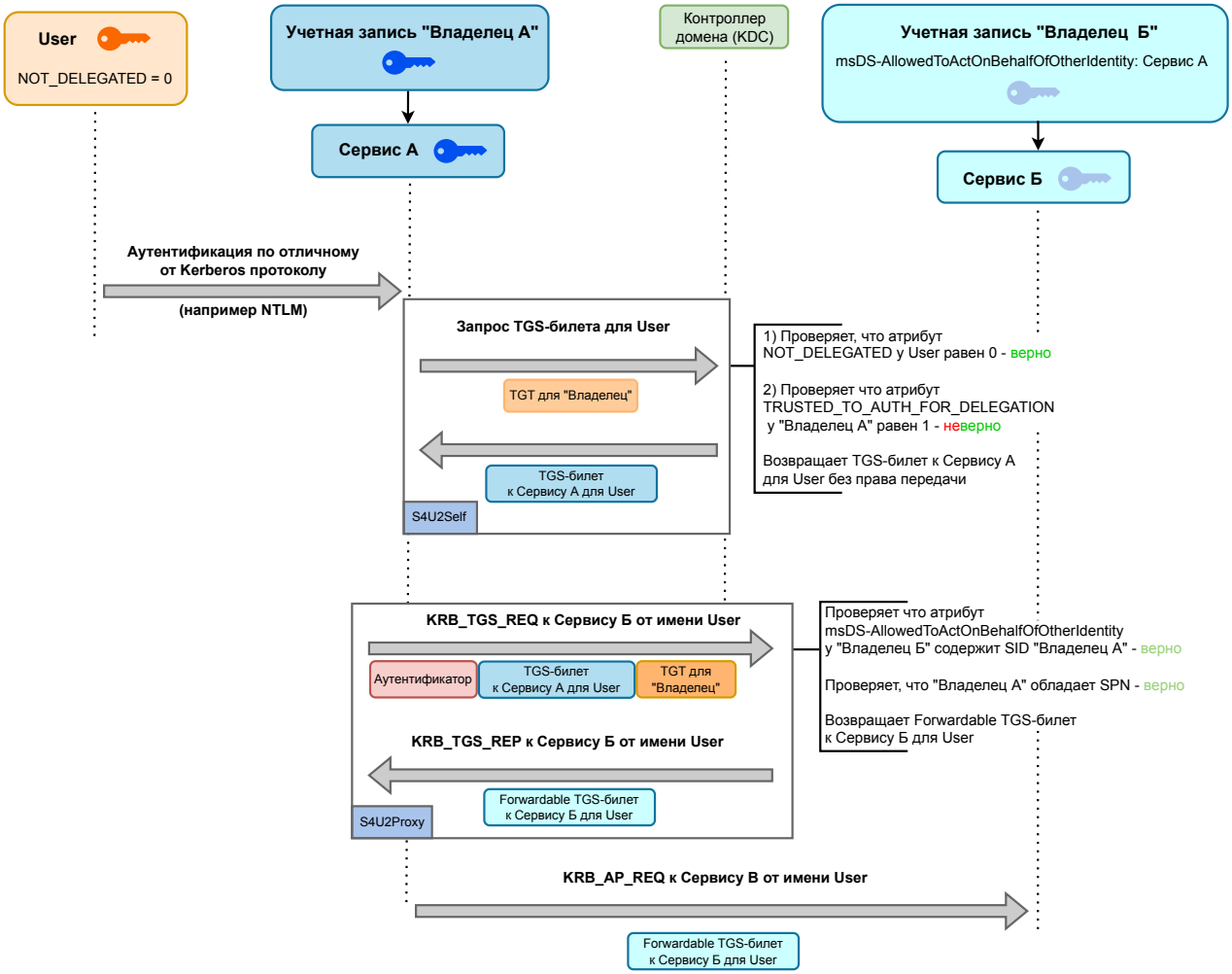


Иллюстрация обмена сообщений при RBCD

Рассмотрим процесс RBCD по пунктам:

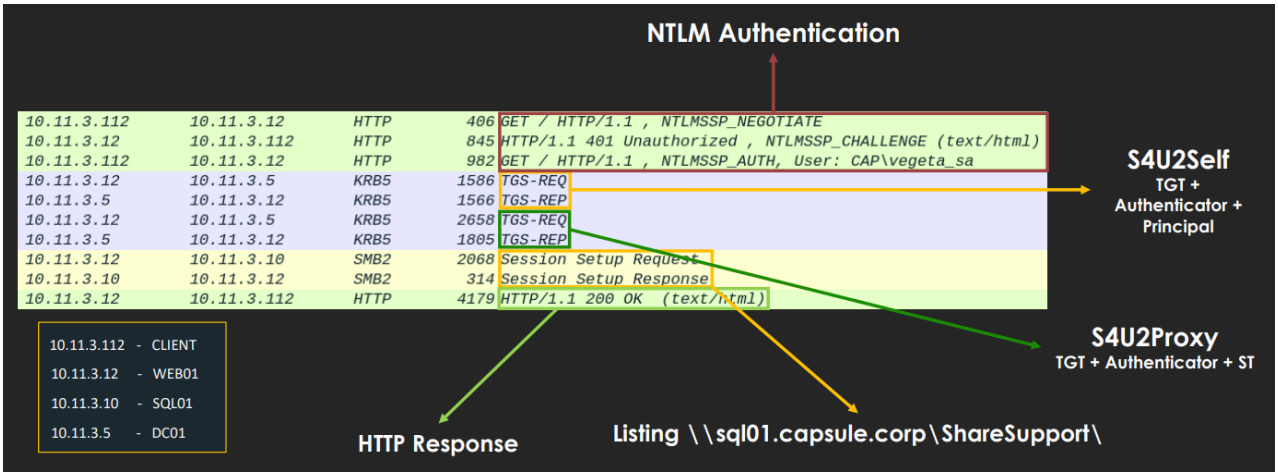
0. Каким образом User прошел аутентификацию к Сервису А неважно, допустим, что с использованием отличного от Kerberos протокола NTLM.
1. Сервис А обращается к контроллеру домену для получения TGS-билета “на себя” от имени пользователя User.
2. Контроллер домена проверяет, что у учетной записи “Владелец А” активен флаг *TRUSTED_TO_AUTH_FOR_DELEGATION*. Указанный флаг **не активен**, поэтому в ответ отправляется обычный *nonforwardable* TGS-билет без права передачи для User к Сервису А. Таким образом S4U2Self работает также, как и в случае классического ограниченного делегирования.

3. С использованием полученного TGS-билета Сервис А обращается к контроллеру домена для получения *Forwardable* TGS-билета к Сервису Б от имени User. Контроллер домена проверяет, что в атрибуте *msDS-AllowedToActOnBehalfOfOtherIdentity* учетной записи “Владелец Б” установлен SID учетной записи “Владелец А”. Также контроллер домена проверяет, что “Владелец А” обладает хотя бы одним SPN. В результате контроллер домена отправляет в ответ *Forwardable* TGS-билет.

> Указанное поведение является важной отличительной особенностью. При RBCD *S4U2Proxy* возвращает *Forwardable* TGS-билет при предоставлении *Nonforwardable* билета, в то время как при ограниченном делегировании *S4U2Proxy* возвращает *Forwardable* TGS-билет только при предоставлении *Forwardable* билета.

4. Сервис А обращается от имени User к Сервису Б с использованием полученного *Forwardable* TGS-билета.

> Еще раз отметим:
>
> - *S4U2Self* при RBCD возвращает *Nonforwardable* TGS-билет.
> - *S4U2Proxy* всегда в случае успеха возвращает *Forwardable* TGS-билет.



Пример сетевого трафика при делегировании ограниченном на основе ресурсов

Классическая атака на RBCD

Условия для проведения атаки:

- Возможность изменять содержимое атрибута *msDS-AllowedToActOnBehalfOfOtherIdentity* атакуемой учетной записи.
- Контроль над учетной записью, обладающей SPN.

Результат успешной атаки: доступ с административными правами к серверу, предназначенному для функционирования сервиса, работающего от имени учетной записи с настроенным RBCD.

На первом шаге атакующий записывает SID подконтрольной учетной записи, обладающей SPN, в атрибут *msDS-AllowedToActOnBehalfOfOtherIdentity* атакуемой учетной записи. Таким образом для подконтрольной учетной записи предоставляется право на олицетворение практически любого пользователя при обращении к сервису атакуемой учетной записи.

```
rbcd.py 'DomainFQDN'/'Username':'Password' -delegate-from
'ControlledAccountWithSPN' -delegate-to 'Target$' -dc-ip 'DC_IP' -action write
```

На втором шаге атакующий выполняет S4U2Self запрос TGS-билета от имени административного пользователя к сервису работающему из-под контролируемой учетной записи. В результате запроса атакующий получает Nonforwardable TGS-билет.

```
getST.py -spn "cifs/target" -impersonate $AdminAccountName -dc-ip
$DomainController $DomainFQDN/$ControlledAccountWithSPN:$Password
```

На третьем шаге атакующий с использованием полученного Nonforwardable TGS-билета выполняет S4U2Proxy запрос. В результате успешного запроса атакующий получает Forwardable TGS-билет к сервису атакуемой учетной записи от имени административного пользователя. На этом атака считается завершенной.

На первый взгляд атака может показаться не очень применимой на практике. Действительно, условие наличия возможности изменения атрибута *msDS-AllowedToActOnBehalfOfOtherIdentity* у атакуемой учетной записи, выглядит довольно требовательным, не смотря на то, что наличие специальной привилегии уровня администратора домена для изменения указанного атрибута не требуется. Кроме того не очень понятно, как получить учетную запись, обладающую SPN.

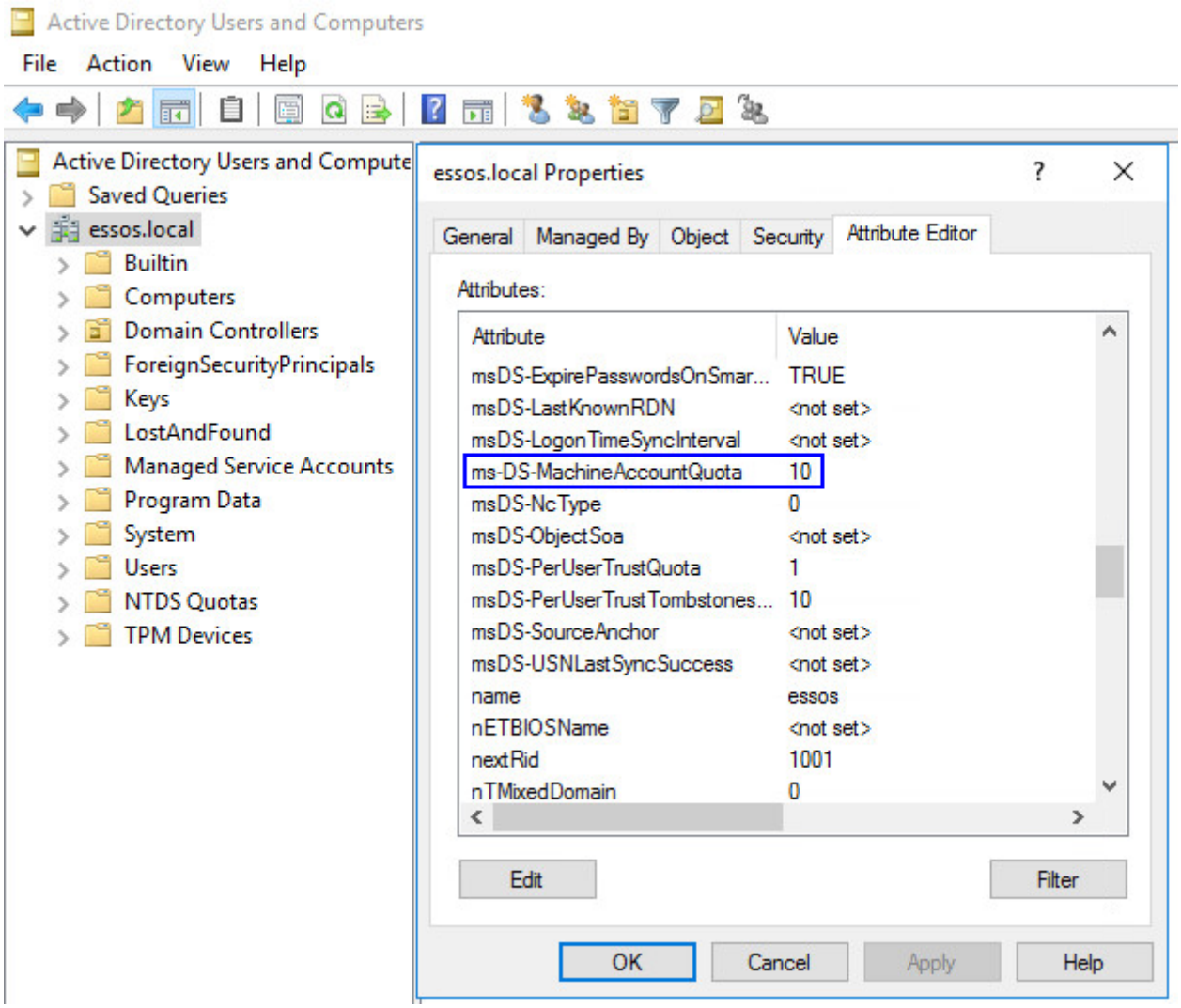
Тем не менее существует ряд особенностей, расширяющих перечень условий для успешной реализации рассмотренной атаки. Рассмотрим некоторые из них.

Как получить учетную запись с SPN

На уровне домена существует атрибут *ms-DS-MachineAccountQuota* (*MAQ*), который отвечает за количество машинных учетных записей, которое непривилегированный пользователь указанного домена может добавить.

> В разделе “общая теория” описывалась разница между пользовательскими и машинными учетными записями.

По умолчанию значение атрибута *ms-DS-MachineAccountQuota* равно 10.



Пример содержимого MAQ

Проверить текущее значение *MAQ* можно например с помощью специального модуля к CrackMapExec:

```
crackmapexec ldap $DC -u $Username -p $Password -d $DomainFQDN -M MAQ
```

Как правило, машинная учетная запись сразу после создания уже имеет несколько SPN. Кроме того создатель машинной учетной записи обладает рядом прав в отношение указанной учетной записи, в том числе позволяющих добавить новые SPN.

Таким образом, захватив пользовательскую непривилегированную учетную запись, атакующий может добавить машинную учетную запись для дальнейшего использования при проведении атак на RBCD.

```
addcomputer.py -computer-name 'evilcomputer$' -computer-pass $GeneratedPass -dc-ip $DC_IP $Domain_FQDN/$Username:$Password
```

В скрипте addcomputer.py присутствует параметр -method, который отвечает за выбор протокола с использованием которого будет осуществляться добавление машинной учетной записи. Есть две опции: SAMR или LDAPS. По умолчанию используется SAMR, так как LDAPS не всегда может быть доступен. Важно

отметить, что в отличие от LDAPS, при добавлении машинной учетной записи с помощью SAMR SPN не создаются. В таком случае необходимо дополнительно задействовать скрипт addspn.py.

```
addspn.py -u '$DomainFQDD'\'evilcomputer$' -p $LM:$NT -s anyRecordName.$DomainFQDN $DC_FQDN
```

Атака без контроля над учетной записью с SPN

Не так давно был открыт способ проводить RBCD-атаки и при отсутствии возможности добавлять машинные учетные записи в домен, в том числе когда атрибут *ms-DS-MachineAccountQuota* равен 0. В этом случае наличие учетной записи с SPN несущественно и для атаки достаточно обладать обычной непривилегированной учетной записью пользователя. Хороший материал на эту тему под названием “Делегируй меня полностью, или Новый взгляд на RBCD-атаки в AD” написал snovvcrash. Не вижу смысла дублировать и настоятельно рекомендую ознакомиться с указанной статьей.

> Один момент, который хочется отметить: рассматриваемый способ подразумевает смену пароля подконтрольной пользовательской учетной записи. Таким образом атака приводит к неработоспособности указанной учетной записи со стороны атакуемой организации, то есть “отказу в обслуживании”, что следует принимать во внимание.

Изменение *msDS-AllowedToActOnBehalfOfOtherIdentity* с помощью ACL

Обладание одним из прав в отношении атакуемой учетной записи:

- GenericAll
- GenericWrite
- WriteDACL
- WriteOwner

позволяет атакующему изменять атрибут *msDS-AllowedToActOnBehalfOfOtherIdentity*. Подробнее почему это так можно посмотреть на русском языке в докладе “Другой взгляд атакующего на ACL в AD”.

> Примечание: наличие приведенных прав можно проверить или поискать с помощью Bloodhound.

Изменение *msDS-AllowedToActOnBehalfOfOtherIdentity* через Relay-атаки

Рассмотрим еще один немаловажный способ. Ранее было отмечено, что атакуемая учетная запись может самостоятельно изменить значение своего атрибута *msDS-AllowedToActOnBehalfOfOtherIdentity* и наличие административных привилегий для

этого не требуется. Но как заставить учетную запись изменить значение своего атрибута *msDS-AllowedToActOnBehalfOfOtherIdentity* и вписать туда SID учетной записи контролируемой атакующим?

Ответ: с помощью Relay-атаки.

Если вкратце, то в результате указанной атаки осуществляется перехват NetNTLMv1/2 запроса на аутентификацию и перенаправление полученных аутентификационных данных с последующим выполнением команд от имени атакуемой учетной записи, в частности можно изменить значение атрибута *msDS-AllowedToActOnBehalfOfOtherIdentity*.

Для начинающего разбираться во внутреннем тестировании на проникновение читателя приведенная выше формулировка скорее всего будет затруднительна для понимания. Изначально было желание рассмотреть RBCD в связке с Relay-атаками, но тогда возникают следующие проблемы:

- Тема Relay-атак довольно обширна и заслуживает отдельной, может и не одной, статьи. Описание принципа и условий проведения Relay-атак выходит за рамки настоящего материала.
- Если продолжить без объяснения Relay-атак, то разрывается логика повествования. От читателя начинает требоваться наличие знаний, которые до этого не обсуждались. Ранее в предыдущих статьях материал преподносился последовательно “с нуля” и на мой взгляд это очень важное свойство от которого не хочется отступать.

Таким образом было принято решение завершить статью об атаках на RBCD. Но это не означает, что автор не вернется к этой теме в будущем, например после материала по Relay-атакам.

Дополнительные материалы для дальнейшего изучения

Рекомендации

- Провести инвентаризацию домена на предмет наличия учетных записей с неиспользуемым настроенным RBCD. Использовать RBCD только при необходимости.
- Добавить критически важные учетные записи домена в группу Protected Users или активировать опцию “Account is sensitive and cannot be delegated” в атрибутах указанных учетных записей.
- По возможности назначать владельцами сервисов выделенные пользовательские учетные записи. Обеспечить сложность и периодическую сменяемость паролей к указанными учетным записям.
- Установить атрибут ms-DS-MachineAccountQuota равным 0.

Противодействовать Relay-атакам:

- Подписывать SMB и LDAP сообщения на всех объектах, входящих в состав домена.
- Настроить и использовать LDAP channel binding.
- Отключить службу печати «Print Spooler» на всех объектах домена, где использование указанной службы не требуется.
- Скорректировать настройки межсетевых экранов с целью ограничения возможности осуществления принудительной аутентификации. Хорошая статья на эту тему.
- Исключить использование протокола NetNTLM v1. По возможности отказаться от использования NetNTLM v2 (труднодостижимо на практике).
- Использовать актуальные версии ОС с установленными критическими обновлениями, что в частности, помогает защититься от атак “Drop the Mic” или “PrivExchange”.
- Исключить вхождение группы “Authenticated Users” в предустановленную стандартную группу “Pre-Windows 2000 Compatible Access”.
- При помощи групповых политик отключить использование на объектах, функционирующих в составе домена, небезопасных широковещательных протоколов NetBIOS, LLMNR, а также службы автоматической настройки прокси (WPAD) и протокола IPv6.

> Примечание: к исполнению рекомендаций следует подходить индивидуально. Дело в том, что зачастую нельзя дать универсальную рекомендацию, подходящую для всех сетей и доменов. Прежде чем что-то предпринимать следует убедиться в целесообразности вносимых изменений. Например, протокол IPv6 или служба WPAD могут быть необходимы при работе и тогда вместо отключения следует выполнить корректировку настроек.

Используемые источники

- Статья, заложившая основы для RBCD атак “Wagging the Dog: Abusing Resource-Based Constrained Delegation to Attack Active Directory”.
- Отличный доклад про делегацию в Kerberos: “You Do (Not) Understand Kerberos Delegation” от Daniel López Jiménez (видео и презентация)
- Доклад “Delegating Kerberos to bypass Kerberos delegation limitation” от Charlie Bromberg (видео и презентация)
- Материалы с Hacker Recipes от Charlie Bromberg (Shutdown)
- Статья: “Kerberos (III): How does delegation work?” от Eloy Pérez
- Посты из телеграмм канала “CyberSecrets”
- Delegate or Escalate? The Dangers of Kerberos Delegation (Jared Yeo) - презентация
- Delegate to the Top: Abusing Kerberos for arbitrary impersonations and RCE (Matan Hart) - доклад презентация