Атаки на Active Directory: часть 6

defcon.ru/penetration-testing/19011

Шестая часть перевода статьи zer1t0, посвященная атакам на сервисы S4U2proxy и S4U2self.

Ограниченное делегирование Kerberos

Неограниченное делегирование может быть опасным, поскольку оно позволяет полностью олицетворять клиента. Таким образом, чтобы создать более ограниченный механизм делегирования, Microsoft разработала два расширения Kerberos, известные как `Service for User` (S4U):

- Сервис для пользователя на прокси (S4U2proxy);
- Сервис для пользователя к себе (S4U2self).

Используя эти расширения, службы могут быть ограничены выполнением делегирования только в отношении набора разрешенных сторонних служб, и не требуется пользовательский `TGT`, что предотвращает его сохранение на сервере службы. Это известно как ограниченное делегирование.

> Информация предоставлена исключительно в ознакомительных целях. Не нарушайте законодательство!

S4U2proxy

Расширение `S4U2proxy` (Service for User to Proxy) позволяет службе запрашивать другую службу от имени клиента, используя клиентский ST (Сервисный билет), отправленный службе, вместо клиентского TGT.

Более того, в отличие от неограниченного делегирования, служба может запрашивать представляющую ST только для определенных служб из белого списка. Разрешенные службы определяются следующими атрибутами:

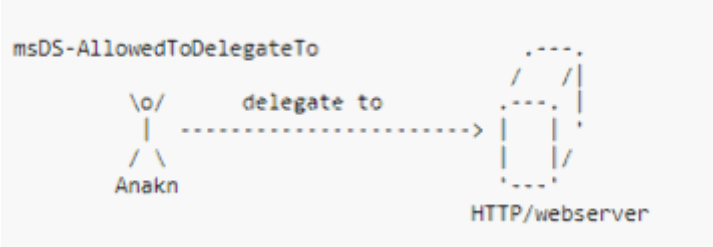
- Атрибут `msDS-AllowedToDelegateTo` учетной записи пользователя службы содержит список SPN (служб), для которых он (и его службы) может запрашивать ST от имени клиента. Этот список сервисов используется в классическом ограниченном делегировании. Чтобы изменить `msDS-AllowedToDelegateTo`, требуется `SeEnableDelegationPrivilege`;
- Учетная запись пользователя службы указана в атрибуте `msDS-AllowedToActOnBehalfOfOtherIdentity` целевой учетной записи пользователя службы. Этот список пользователей используется в ограниченном делегировании на основе ресурсов (RBCD).

Следующая команда показывают примеры этих атрибутов:

```
PS C:\Users\Administrator> get-aduser anakin -Properties msDS-AllowedToDelegateTo

DistinguishedName         : CN=Anakin,CN=Users,DC=contoso,DC=local
msDS-AllowedToDelegateTo  : {HTTP/webserver, HTTP/webserver.contoso.local}
SamAccountName            : anakin
SID                       : S-1-5-21-1372086773-2238746523-2939299801-1103
UserPrincipalName         : anakin@contoso.local
```

Пример msDS-AllowedToDelegateTo

Здесь службам пользователя `Anakin` разрешено выполнять делегирование службы «HTTP/webserver». Таким образом, `Anakin` может выдавать себя за любого пользователя (кроме защищенных).



Более того, поскольку можно изменить целевую службу билета, `Anakin` мог запросить билет для «HTTP/webserver» от имени клиента и изменить целевую службу на любую службу владельца «HTTP/webserver», поскольку все эти службы ST будут зашифрованы одним и тем же ключом Kerberos.

Например, если пользователем «HTTP/webserver» является `webserver$` (учетная запись пользователя компьютера веб-сервера), то `Anakin` может запросить билет для «HTTP/webserver» от имени клиента и использовать этот билет для доступа к службе SMB веб-сервера, изменив целевую службу на «`cifs/webserver`». Таким образом, `Anakin` может получить доступ к веб-серверу, выдавая себя за клиента.

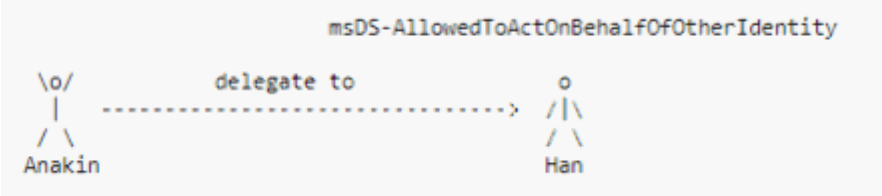
```
PS C:\Users\Administrator> get-aduser han -Properties PrincipalsAllowedToDelegateToAccount,msDS-AllowedToActOnBehalfOfOtherIdentity


DistinguishedName                       : CN=Han,CN=Users,DC=contoso,DC=local
Enabled                                 : True
GivenName                               : Han
msDS-AllowedToActOnBehalfOfOtherIdentity : System.DirectoryServices.ActiveDirectorySecurity
Name                                    : Han
ObjectClass                             : user
ObjectGUID                              : 356a7fb7-6cc0-4e09-a77f-b64e1677f2a8
PrincipalsAllowedToDelegateToAccount    : {CN=Anakin,CN=Users,DC=contoso,DC=local}
SamAccountName                          : han
SID                                     : S-1-5-21-1372086773-2238746523-2939299801-1109
Surname                                 :
UserPrincipalName                       : han@contoso.local
```

Пример msDS-AllowedToActOnBehalfOfOtherIdentity

Поскольку значение `msDS-AllowedToActOnBehalfOfOtherIdentity` является дескриптором безопасности в двоичном формате, необходимо запросить свойство `PrincipalsAllowedToDelegateToAccount`, которое выводит эти данные в удобном для человека формате.

С другой стороны, проверив `msDS-AllowedToActOnBehalfOfOtherIdentity` пользователя `Han`, мы обнаружим, что он позволяет пользователю `Anakin` выполнять делегирование всех своих служб.

Следовательно, `Anakin` может выдавать себя за любого пользователя (кроме защищенных) для любого сервиса пользователя `Han`.

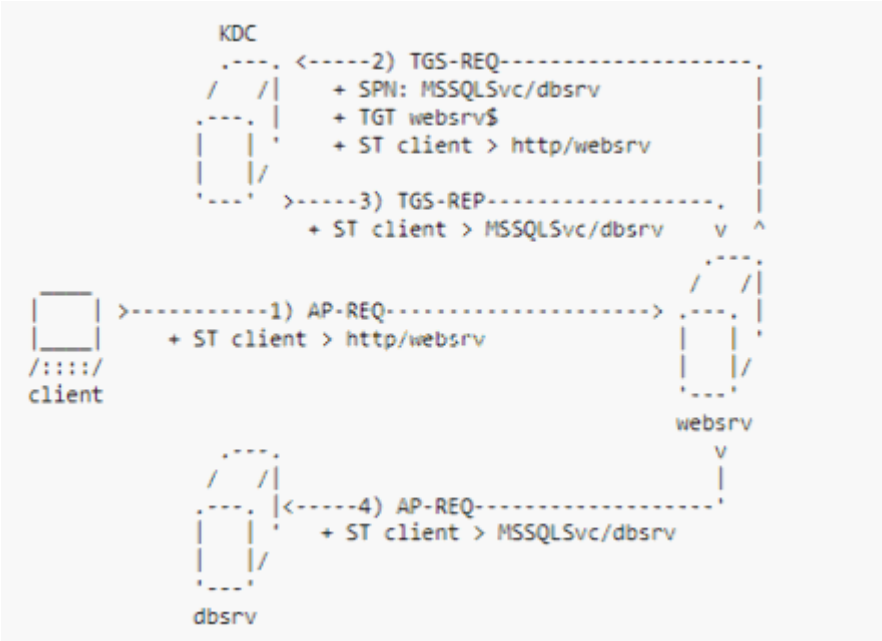


Кроме того, `KDC` также проверяет другие параметры, чтобы определить результат запроса `S4U2proxy`. Также учитывается, является ли клиент ST `FORWARDABLE` и защищен ли клиент от делегирования. Вы можете проверить правила в спецификации `MS-SFU`. Вкратце, правила следующие:

- Если PAC-тикета клиента ST => вернуть ошибку `KRB-AP-ERR-MODIFIED`;
- Если клиент ST не `FORWARDABLE` и клиент защищен => вернуть ошибку `KRB-ERR-BADOPTION - STATUS-ACCOUNT-RESTRICTED`;
- Если клиент ST не `FORWARDABLE` и `target_service` в `ms-AllowedToDelegateTo` => возвращает ошибку `KRB-ERR-BADOPTION - STATUS-ACCOUNT-RESTRICTED`;
- Если клиент ST `FORWARDABLE` и `target_service` в `ms-AllowedToDelegateTo` => возвращает `S4U2proxy ST`;
- Если пользователь службы в `target_service user msDS-AllowedToActOnBehalfOfOtherIdentity` => возвращает `S4U2proxy ST`.

Следует отметить одну любопытную вещь: можно получить `S4U2proxy ST` от клиента ST, не поддерживающего `FORWARDABLE`, с помощью ограниченного делегирования на основе ресурсов (`msDS-AllowedToActOnBehalfOfOtherIdentity`). За исключением случая, когда целевая служба также указана в `ms-AllowedToDelegateTo` (правило 3), где будет возвращена ошибка. Более того, вы можете заметить, что `S4U2proxy` возвращает форвардные тикеты.

Давайте теперь рассмотрим пример процесса S4U2proxy:



Процесс S4U2proxy

- Клиент аутентифицируется в службе веб-сервера `http/websrv`, отправляя ST;
- Когда веб-серверу `http/websrv` потребуется доступ к службе базы данных `MSSQLSvc/dbsrv` от имени клиента, он запрашивает ST для `MSSQLSvc/dbsrv`, используя клиентский ST и собственный TGT;
- KDC проверяет, разрешено ли пользователю службы `websrv$` запрашивать билеты делегирования для `MSSQLSvc/dbsrv`, следуя ранее обсуждавшимся правилам, и возвращает ST клиента для `MSSQLSvc/dbsrv`.

Подводя итог, обычно должно выполняться одно из этих условий:

- `MSSQLSvc/dbsrv` включен в атрибут `msDS-AllowedToDelegateTo` для `websrv$` (пользователь службы веб-сервера). Это классическое ограниченное делегирование;
- `websrv$` включен в атрибут `msDS-AllowedToActOnBehalfOfOtherIdentity` `dbsrv$` (пользователь службы `MSSQLSvc/dbsrv`). Это ограниченное делегирование на основе ресурсов;
- Веб-служба использует недавно приобретенный ST для аутентификации себя в базе данных, выдавая себя за клиента.

Кроме того, также возможно использовать S4U2proxy между доменами, однако в этом случае можно использовать только ограниченное делегирование на основе ресурсов.

S4u2proxy между доменами

- Мы предполагаем, что клиент уже отправляет свой ST сервису `websrv`. Затем `websrv` необходимо получить доступ к службе базы данных `MSSQLSvc/dbsrv` от имени пользователя;
- `websrv` запрашивает ST для `MSSQLSvc/dbsrv` от имени клиента, включая его собственный ST;
- KDC проверяет запрос и определяет, что запрошенная служба `bar.com` включена, поэтому он возвращает специальный межобластной TGT для запроса S4U2proxy в KDC `bar.com`;
- `websrv` проверяет ответ и обнаруживает этот специальный межобластной TGT для `S4U2proxy`. Но ему также нужен нормальный межобластной TGT для `bar.com`, поэтому он запрашивает его для KDC;
- KDC возвращает межобластной TGT `bar.com` для `websrv$`;
- Затем `websrv$` использует эти межобластные TGT, чтобы запросить у `bar.com` KDC для ST для `MSSQLSvc/dbsrv` от имени клиента;
- KDC проверяет запросы и определяет, что `websrv$` разрешено делегировать службе `MSSQLSvc/dbsrv` (используя RBCD), поэтому выдает ST для `MSSQLSvc/dbsrv`;
- `websrv` использует этот новый ST для доступа к службе `MSSQLSvc/dbsrv` от имени клиента.

S4U2self

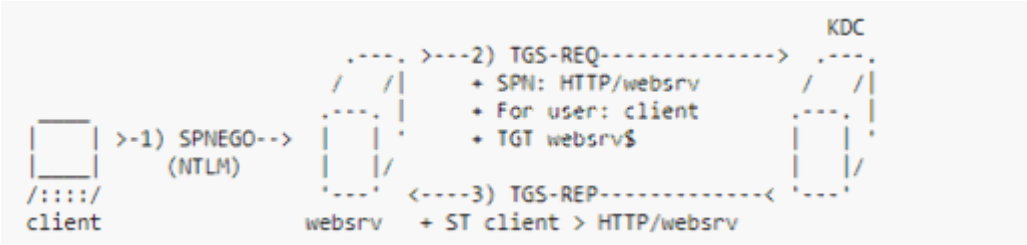
Расширение Kerberos `S4U2self` позволяет службе запрашивать билет от имени пользователя для себя, который затем можно использовать в `S4U2proxy`. Это сделано для того, чтобы разрешить делегирование Kerberos для тех клиентов, которые не поддерживают протокол Kerberos.

Чтобы иметь возможность использовать S4U2self, KDC проверяет флаг `UserAccountControl TRUSTED_TO_AUTH_FOR_DELEGATION` учетной записи пользователя службы. Чтобы изменить этот флаг, требуется `SeEnableDelegationPrivilege`.

Кроме того, KDC также проверяет наличие у пользователя службы каких-либо служб и значение атрибута `msDS-AllowedToDelegateTo`. Конкретные правила можно увидеть в спецификации MS-SFU, а вот сводка проверок, выполняемых KDC при получении запроса S4U2self:

- Если у пользователя службы нет никаких служб => вернуть ошибку `KDC-ERR-S-PRINCIPAL-UNKNOWN`.
- Если клиент защищен от делегирования => вернуть `non-FORWARDABLE ST`.
- Если флаг `TRUSTED_TO_AUTH_FOR_DELEGATION` пользователя службы установлен => вернуть `FORWARDABLE ST`.
- Если флаг `TRUSTED_TO_AUTH_FOR_DELEGATION` пользователя службы не установлен, а у пользователя службы есть службы в `ms-AllowedToDelegateTo` => вернуть `non-FORWARDABLE ST`.
- Если флаг `TRUSTED_TO_AUTH_FOR_DELEGATION` пользователя службы не установлен, а `ms-AllowedToDelegateTo` пользователя службы пусто => вернуть `FORWARDABLE ST`.

Рассмотрим пример:

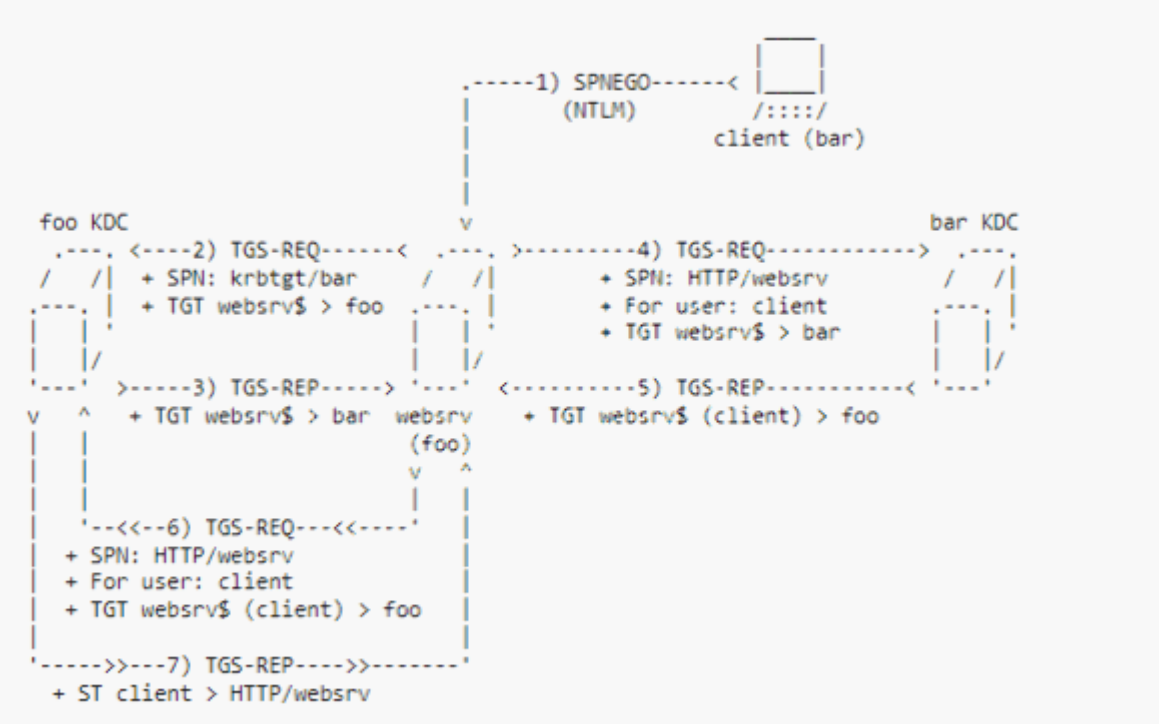


Процесс S4U2self

- Клиент проходит проверку подлинности в службе `HTTP/websrv` с использованием NTLM (или любого другого протокола проверки подлинности), поскольку он не поддерживает Kerberos;
- `Websrv` запрашивает `ST S4U2self` для клиента, отправляя `TGS-REQ` в KDC;

- KDC проверяет запросы, флаг `websrv$ TRUSTED_TO_AUTH_FOR_DELEGATION` и защищен ли клиент от делегирования. Если все правильно, KDC возвращает `HTTP/websrv ST` для клиента, который может или не может быть `FORWARDABLE` в зависимости от упомянутых переменных.

Более того, `S4U2Self` можно использовать в разных доменах. Давайте рассмотрим как это работает:

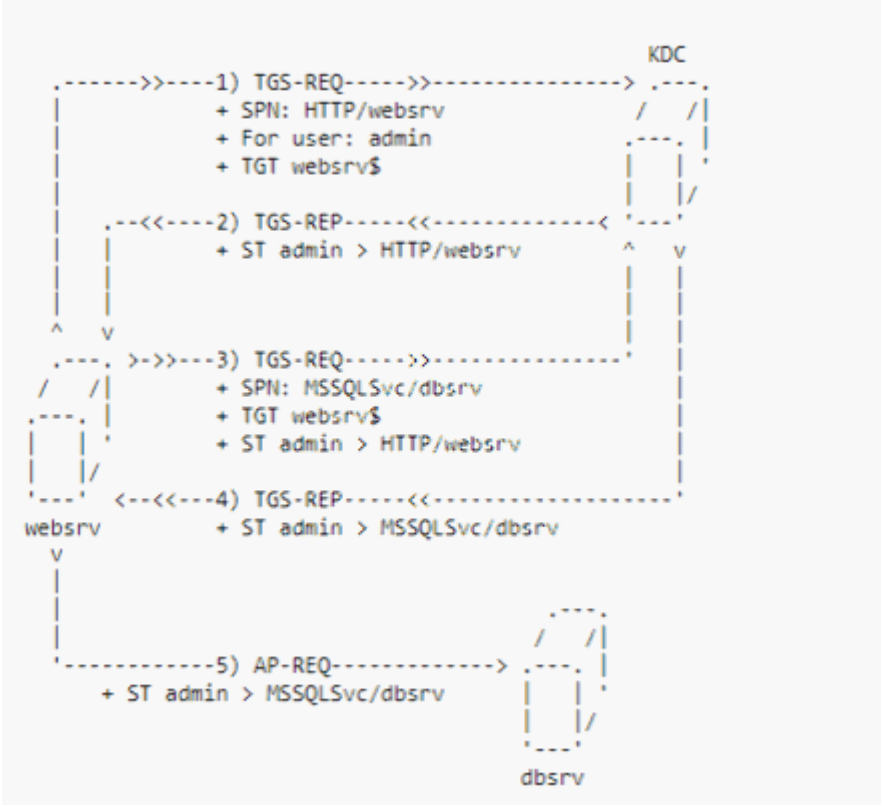


S4U2self в доменах

- Клиент проходит проверку подлинности в службе `HTTP/websrv` с использованием NTLM (или любого другого протокола проверки подлинности), поскольку он не поддерживает Kerberos;
- `websrv` определяет, что областью клиента является `bar`, поэтому отправляет `TGS-REQ` с запросом TGT для домена `bar`;
- KDC возвращает межобластной TGT для `barwebsrv`;
- `websrv` использует свой новый межобластной TGT, чтобы запросить у KDC `bar` HTTP/websrv ST для клиента;
- KDC определяет, что служба `HTTP/websrv` находится в домене `foo`, поэтому она не может выдать ST для службы `HTTP/websrv`, но возвращает реферальный TGT для домена `foo`, который указывает, что для клиента была запрошена HTTP/websrv ST;
- Затем `websrv` использует этот реферальный TGT, выданный KDC `bar`, чтобы запросить ST для клиента в KDC `foo`;
- KDC `foo` проверяет запрос и реферальный TGT и определяет, что HTTP/websrv ST для клиента может быть выдан.

S4U2self и S4U2proxy

Теперь, когда мы знаем, как работают `S4U2self` и `S4U2proxy`, давайте рассмотрим их совместное использование на примере.



S4U2self в цепочке с S4U2proxy

- `Websrv` запрашивает HTTP/websrv ST для пользователя-администратора в KDC с помощью `S4U2self` через `TGS-REQ`;
- KDC проверяет запросы, флаг `websrv$ TRUSTED_TO_AUTH_FOR_DELEGATION` и защищен ли администратор от делегирования. Если все правильно, KDC возвращает HTTP/websrv ST для клиента, который может или не может быть `FORWARDABLE` в зависимости от переменных, упомянутых в `S4U2Self`;
- Затем `websrv` запрашивает `MSSQLSvc/dbsrv` ST от имени администратора, используя ST `S4U2self` и свой собственный TGT;
- KDC проверяет, разрешено ли пользователю службы `websrv$` запрашивать билеты делегирования для `MSSQLSvc/dbsrv` в соответствии с правилами, упомянутыми в `S4U2Proxy`. Затем он возвращает `MSSQLSvc/dbsrv` ST для администратора;
- `websrv` использует `MSSQLSvc/dbsrv` ST для аутентификации в базе данных, выдавая себя за администратора.

Поэтому можно связать S4U2self и S4U2proxy, чтобы вы могли представлять любого пользователя (кроме защищенных от делегирования) для всех тех служб, которым пользователю службы разрешено выполнять ограниченное делегирование. И, конечно, также возможно использование S4U2self и S4U2proxy между доменами.

Атаки S4U

Давайте посмотрим, как можно злоупотреблять расширениями `Constrained Delegation` и `S4U` при тестировании на проникновение.

Чтобы найти учетные записи, использующие ограниченное делегирование, необходимо найти учетную запись с включенным `UserAccountControl TRUSTED_TO_AUTH_FOR_DELEGATION` или со значениями атрибутов `msDS-AllowedToDelegateTo` или `msDS-AllowedToActOnBehalfOfOtherIdentity`. Вот фильтр LDAP, который вы можете использовать для поиска учетных записей ограниченного делегирования:

```
(|
  (UserAccountControl:1.2.840.113556.1.4.803:=16777216)
  (msDS-AllowedToDelegateTo=*)
  (msDS-AllowedToActOnBehalfOfOtherIdentity=*)
)
```

Фильтр LDAP для получения учетных записей, связанных с ограниченным делегированием

Чтобы найти учетные записи, связанные с ограниченным делегированием, вы можете использовать такие инструменты, как Powerview, скрипт impacket findDelegation.py, модуль Powershell ActiveDirectory или ldapsearch.

```
(|
  (memberof:1.2.840.113556.1.4.1941:=CN=Protected Users,CN=Users,DC=<domain>,DC=<dom>)
  (userAccountControl:1.2.840.113556.1.4.803:=1048576)
)
```

Фильтр LDAP для получения учетных записей, защищенных от делегирования

После того, как вы нашли учетные записи и хотите выполнить некоторые связанные операции Kerberos, существует множество инструментов, которые позволяют выполнять запросы билетов через расширения S4U и получать ST для произвольных пользователей, чтобы они выдавали себя за них. Вы можете использовать утилиты MIT kerberos (ktutitl, kinit, kvno), Rubeus, скрипт impacket getST.py или cerbero.

Кроме того, в случае выполнения команд учетной записи с привилегиями `SYSTEM` на компьютере с ограниченным делегированием можно использовать S4U2self и S4U2proxy с небольшим количеством кода Powershell:

```
# Code made by Lee Christensen (@tifkin_) and Will Schroeder (@harmj0y)
# Source: https://www.harmj0y.net/blog/activedirectory/s4u2pwnage/

# translated from the C# example at https://msdn.microsoft.com/en-us/library/ff649317.aspx

# load the necessary assembly
$Null = [Reflection.Assembly]::LoadWithPartialName('System.IdentityModel')

# execute S4U2Self w/ WindowsIdentity to request a forwardable TGS for the specified user
$Ident = New-Object System.Security.Principal.WindowsIdentity @('Administrator@FOO.LOCAL')

# actually impersonate the next context
$Context = $Ident.Impersonate()

# implicitly invoke S4U2Proxy with the specified action
ls \\DC01.FOO.LOCAL\C$

# undo the impersonation context
$Context.Undo()
```

Преимущество ограниченного делегирования заключается в том, что во многих случаях (включено RBCD или TrustedToAuthForDelegation) вы можете представляться пользователями без какого-либо взаимодействия. Однако, поскольку количество сервисов, к которым вы можете получить доступ, ограничено, вы должны знать о тех сервисах, которые могут быть полезны при делегировании:

- **LDAP контроллера домена**
  Служба LDAP Active Directory используется для управления учетными записями, включая ее разрешения, поэтому, если вы можете выдавать себя за администратора в отношении службы LDAP, вы можете предоставить любые привилегии любой учетной записи пользователя, которой вы управляете. Примером может служить предоставление прав произвольному ползователю для выполнения атаки `DCSync` и компрометации домена.
- **SMB любого компьютера**
  В случае, если вам разрешено выдавать себя за любого пользователя в службе SMB, вы можете получить доступ ко всем файлам на компьютере, выполнять команды с помощью таких инструментов, как psexec, и выполнять другие действия через вызовы RPC.
- **Службы MSSQL**
  Помимо содержания конфиденциальных данных, которые могут быть важным флагом для получения при тестировании на проникновение, серверы MSSQL также могут позволять пользователям выполнять команды с помощью `xp_cmdshell`, злоупотреблять ретрансляцией NTLM, выполняя HTTP-запросы через `xp_dirtree` к серверу `WebDAV` и многие другие варианты.
- **Сервис krbtgt**
  Если учетной записи разрешено делегировать службу `krbtgt`, она может запросить у TGT любую учетную запись.

И помните, что даже если вам разрешено делегировать не напрямую в один из этих сервисов через Classic Constrained Delegation (атрибут `ms-AllowedToDelegateTo`), а в один сервис одного и того же пользователя, вы можете изменить целевой сервис в тикете. Например, если вам разрешено делегировать службу HTTP компьютера,

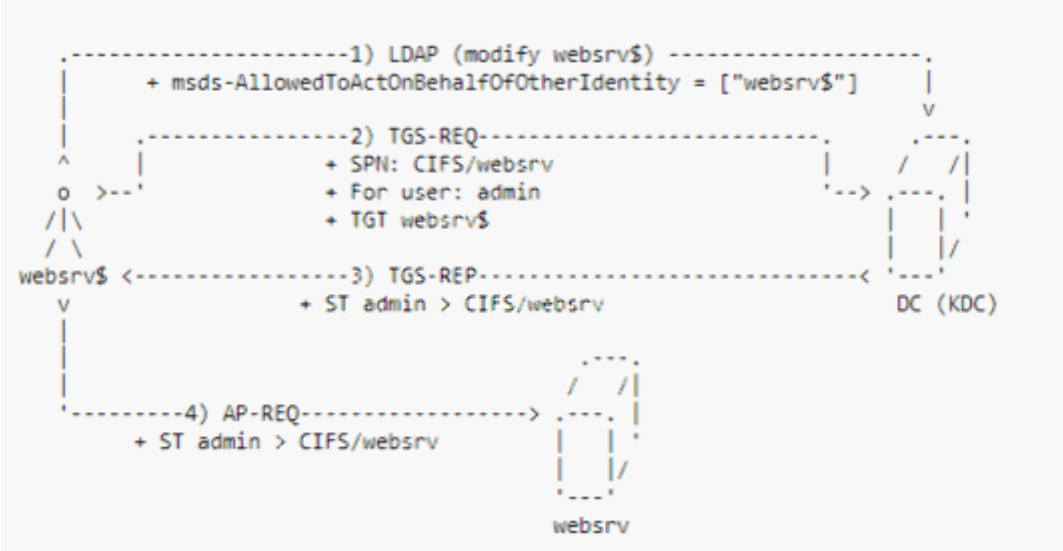
например `HTTP/websrv`, вы можете изменить целевую службу на `CIFS/websrv` для доступа к компьютеру (если служба HTTP выполняется в контексте компьютера). Кроме того, если вы можете делегировать какую-либо службу контроллера домена, вы, вероятно, можете изменить службу билетов, чтобы использовать ее для доступа к службе LDAP.

Чтобы представлять пользователя для службы, вам потребуется ограниченное делегирование на основе ресурсов (RBCD) или классическое ограниченное делегирование с переходом протокола (S4U2self). Вы можете включить RBCD для учетной записи, разрешив запись в ее атрибуте `ms-AllowedToActOnBehalfOfOtherIdentity` и указание на учетную запись, которая имеет хотя бы одну службу (для использования S4U2self).

Если у вас нет учетной записи с хотя бы одной службой, вы можете создать одну учетную запись компьютера, злоупотребив квотой компьютеров, которая позволяет пользователям по умолчанию создавать до 10 учетных записей компьютеров в домене. Это можно сделать с помощью Powermad или impacket addcomputer.py. После создания учетной записи компьютера (пользователь может выбрать пароль учетной записи компьютера) создавший ее пользователь может назначать ей службы. Таким образом, вы можете получить учетную запись с сервисами.

Более того, учетные записи по умолчанию имеют разрешения на редактирование собственного атрибута `ms-AllowedToActOnBehalfOfOtherIdentity`. Поэтому, если вы можете получить учетные данные (например, хэш NT, ключи Kerberos или TGT) учетной записи компьютера, вы можете включить RBCD для произвольного пользователя на этом компьютере. Таким образом, вы можете использовать RBCD, чтобы выдать себя за администратора в службе CIFS (SMB) компьютера и поставить компьютер под угрозу.

Поскольку у вас есть данные учетной записи компьютера, вы можете включить RBCD для себя (отражающий RBCD). Таким образом, вам просто нужно использовать S4U2self, чтобы запросить билет для службы CIFS компьютера, чтобы получить ST для компрометации хоста. Это работает даже для олицетворения защищенных учетных записей пользователей. Если вам интересно, этот метод необходим, поскольку учетные записи компьютеров домена по умолчанию не имеют разрешений на удаленный доступ к самому компьютеру в качестве администратора.



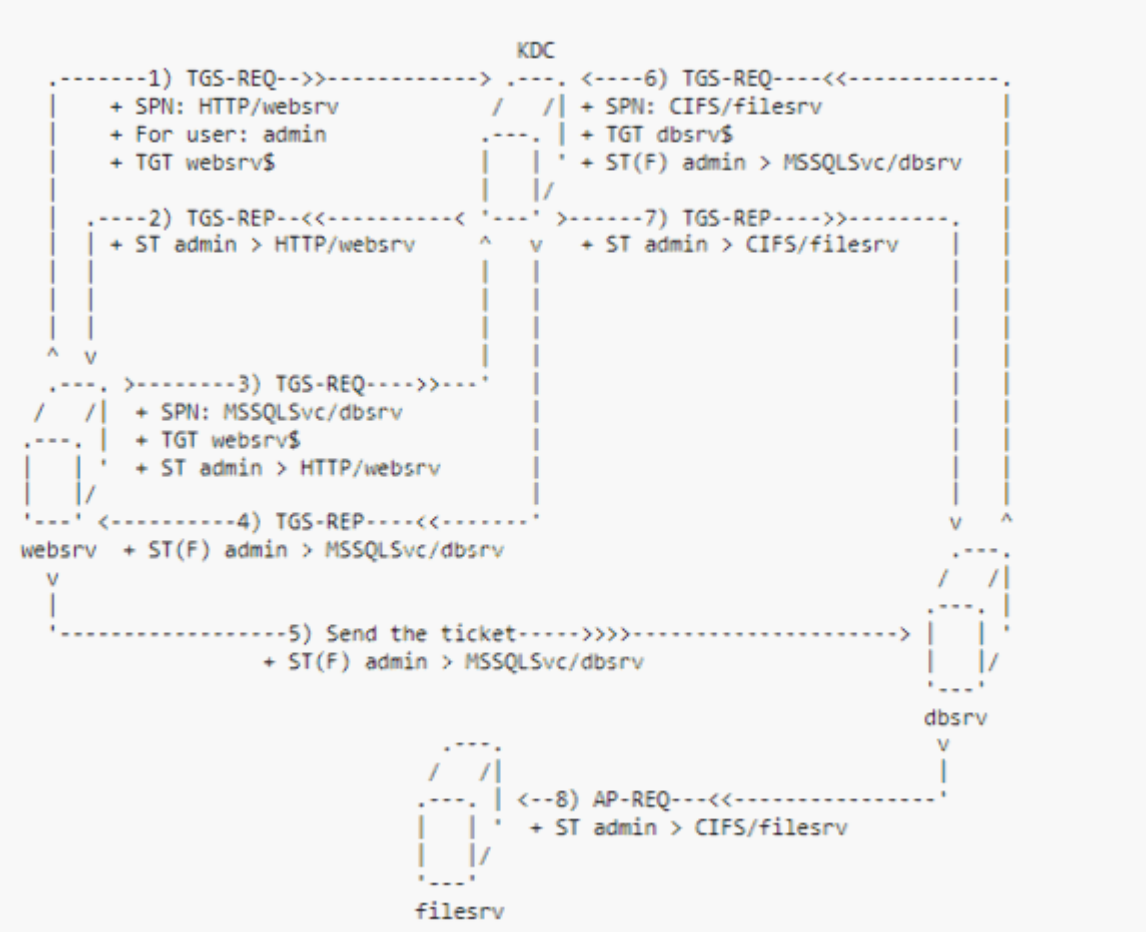
Отражающая атака RBCD

Тем не менее, получить учетные данные компьютера перед компрометацией машины может быть сложно, но если вы можете заставить компьютер сделать HTTP-запрос с аутентификацией NTLM к хосту, которым вы управляете, вы можете использовать перекрестный NTLM для ретрансляционной атаки с HTTP на LDAP, чтобы включить RBCD для учетной записи компьютера для учетной записи, которой вы управляете.

Для этого вы можете воспользоваться клиентом WebDAV, установленным по умолчанию на рабочих столах Windows. Например, вы можете инициировать аутентифицированный HTTP-запрос, используя процедуру `xp_dirtree` базы данных MSSQL (для этого вы можете использовать bad_sequel.py).

Однако возможно, что вы скомпрометируете учетные записи с классическим ограниченным делегированием, в которых не включен переход протокола (S4U2self), поэтому вы не сможете запросить билет для любого пользователя. В этом случае вы можете использовать RBCD для имитации смены протокола. Это означает, что вы можете включить RBCD со скомпрометированной учетной записи на другую учетную запись, чтобы другая учетная запись могла запрашивать билет для любого пользователя на скомпрометированную учетную запись, которую можно пересылать, поскольку она создается S4U2proxy, таким образом имитируя `Protocol Transition`.

Это может быть немного сложно, поэтому давайте рассмотрим пример, когда `dbsrv` скомпрометирован и имеет классическое ограниченное делегирование, но без смены протокола. Однако `websrv` также скомпрометирован и может использоваться для перехода на протокол RBCD. Затем включается RBCD с `websrv` на `dbsrv`, и `websrv` используется для имитации перехода протокола и, наконец, получает ST администратора для компрометации `filesrv` следующим образом:



Использование RBCD в качестве перехода протокола

На первых четырех этапах `websrv` использует S4U2self и S4U2proxy для получения пересылаемого ST `MSSQLSvc/dbsrv` для администратора, таким образом имитируя переход протокола. Затем `websrv` отправляет этот ST администратора в `dbsrv`, который использует его для S4U2proxy и запрашивает ST CIFS/filesrv для администратора, что позволяет скомпрометировать `filesrv`.

Практическая подготовка

Если материал показался вам интересным, и хотите на практике разобраться, как это работает — пройдите Корпоративные лаборатории Pentestit — программу практической подготовки в области информационной безопасности.