Атаки на Active Directory: часть 5

defcon.ru/penetration-testing/18990

Пятая часть перевода статьи zer1t0, посвященная атакам на протокол аутентификации Kerberos.

Kerberos

Основы Kerberos

`Kerberos` является предпочтительным протоколом аутентификации в сетях Active Directory для учетных записей домена (его нельзя использовать в рабочих группах) и описан в RFC 4120. Kerberos фокусируется на использовании токенов, называемых «билетами», которые позволяют пользователю пройти аутентификацию.

Принципы Kerberos

Наиболее распространенными субъектами Kerberos являются пользователи и службы, причем последний тип используется чаще всего. Чтобы запросить билет для службы, необходимо указать имя субъекта-службы (SPN). Например, `HTTP/computer`. Существует несколько основных типов Kerberos, которые можно использовать для запроса службы: `NT-SRV-INST`, `NT-SRV-HST` или `NT-SRV-XHST`. С другой стороны, можно использовать участников для представления пользователей, и как правило они обычно используются для указания имени клиента, запрашивающего билет. Пользователь обычно представляется `SamAccountName` (например, «foo») с использованием типа `NT-PRINCIPAL`. Однако существует также тип `NT-ENTERPRISE`, который допускает более явные форматы для идентификации пользователя, например `SamAccountName@DomainFQDN` (например, «foo@contoso.local»). NT-ENTERPRISE может быть полезен для идентификации пользователей из разных доменов, при междоменном запросе.

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

Билеты

Билеты представляют собой частично зашифрованные структуры, которые содержат:

- целевой субъект (обычно служба), для которого применяется билет;
- информация, связанная с клиентом, такая как имя и домен;
- ключ для установления безопасных каналов между клиентом и сервисом;
- временные метки для определения периода, в течение которого билет действителен.

```
Ticket          ::= [APPLICATION 1] SEQUENCE {
        tkt-vno         [0] INTEGER (5),
        realm           [1] Realm,
        sname           [2] PrincipalName, -- Usually the service SPN
        enc-part        [3] EncryptedData -- EncTicketPart
}

EncTicketPart   ::= [APPLICATION 3] SEQUENCE {
        flags                   [0] TicketFlags,
        key                     [1] EncryptionKey, -- Session Key
        crealm                  [2] Realm,
        cname                   [3] PrincipalName,
        transited               [4] TransitedEncoding,
        authtime                [5] KerberosTime,
        starttime               [6] KerberosTime OPTIONAL,
        endtime                 [7] KerberosTime,
        renew-till              [8] KerberosTime OPTIONAL,
        caddr                   [9] HostAddresses OPTIONAL,
        authorization-data      [10] AuthorizationData OPTIONAL -- Includes a PAC
}
```

Определение билета

Сертификат атрибута привилегий

Помимо обычных данных билета, реализация Kerberos в Active Directory обычно включает в поле билета `authorization-data` важную структуру аутентификации Active Directory: PAC.

`PAC` (сертификат атрибута привилегий) содержит информацию о безопасности, связанную с клиентом:

- **домен клиента**: включает имя домена и SID (`LogonDomainName` и `LogonDomainId` соответственно);
- **пользователя клиента**: имя пользователя и RID пользователя (`EffectiveName` и `UserId` соответственно);
- **группы клиентов**: идентификаторы RID (`GroupId`) тех групп домена, к которым принадлежит пользователь;
- **другие группы**: PAC включает в себя другие SID (`ExtraSids`), которые ссылаются на не доменные группы, которые могут применяться для междоменной аутентификации, а также общеизвестные SID, используемые для указания особых характеристик.

Помимо информации о пользователе, PAC также содержит несколько подписей, используемых для проверки целостности PAC и данных билета.

- **подпись сервера**: подпись содержимого PAC, созданная с помощью того же ключа, который использовался для шифрования билета;
- **подпись KDC**: подпись подписи сервера, созданная с помощью ключа KDC. Это можно использовать для проверки того, что PAC был создан KDC, и предотвращения атак Silver Ticket.
- **подпись билета**: подпись содержимого билета, созданная с помощью ключа KDC. Эта подпись была недавно введена для предотвращения атаки `Bronze bit`.

Компоненты Kerberos

Kerberos использует билеты для аутентификации пользователей в службах. Но как они используются? Чтобы ответить на этот вопрос, необходимо сначала понять, какие компоненты участвуют в аутентификации Kerberos.

Первый компонент — это пользователь, который получает билеты и использует их для доступа к службам в домене (или лесу).

Затем у нас есть второй компонент — служба. В случае с Kerberos обычно говорится о сервере приложений (AP), то есть о машине, которая предлагает услугу, к которой пользователь хочет получить доступ. Точкой доступа здесь может быть любой компьютер домена.

И, наконец, нам нужно, чтобы кто-то предоставил пользователю билеты. Для этого предназначен KDC (Центр распределения ключей). KDC в Active Directory является контроллером домена, поскольку именно он имеет доступ к базе данных домена, необходимой для аутентификации пользователей.

В Kerberos TGT предоставляются службой/сервером аутентификации (AS), а ST — службой/сервером выдачи билетов (TGS). Обе службы запрашивают у KDC ключи Kerberos. Однако, поскольку все эти службы обычно работают на одном сервере, для простоты будем называть их просто KDC.

Типы билетов

Итак, теперь, когда у нас есть клиент, сервер приложений, KDC и билеты, давайте посмотрим, как работает протокол Kerberos. Для этого мы должны иметь в виду, что в протоколе Kerberos есть два типа билетов:

ST

Первый тип — это ST (Сервисные билеты), которые клиент предъявляет AP/службе, чтобы получить к ним доступ. KDC выдает ST для клиентов, которые запрашивают их.

В Active Directory клиент может получить ST для любой службы, зарегистрированной в базе данных домена, не имеет значения, имеет ли пользователь доступ к службе (Kerberos не обрабатывает авторизацию) или даже если служба не запущена.

ST предназначены для чтения целевой службой и никем другим, поскольку они включают информацию о клиенте, который должен быть аутентифицирован, и ключ сеанса для установления соединения с клиентом.

В случае Active Directory обычно целевыми субъектами являются службы, принадлежащие учетным записям пользователей (или учетным записям компьютеров, которые также являются пользователями в Active Directory). В этом случае TGT шифруются ключом владельца учетной записи службы.

Из этой информации мы можем сделать несколько выводов:

Во-первых, если мы знаем ключ (который получен из пароля), мы можем подделать билеты. С точки зрения Active Directory, если мы знаем ключ пользователя, мы можем создавать собственные билеты для доступа к любой службе этого пользователя. Эти индивидуальные билеты известны как Silver Ticket.

Например, если вы знаете пароль учетной записи компьютера (хранящийся в секретах LSA машины), вы можете создать Silver Ticket для службы SMB и получить доступ к машине как администратор.

Во-вторых, следует отметить, что если несколько сервисов принадлежат одному пользователю, они будут зашифрованы одним и тем же ключом. Вы можете использовать эту информацию вместе с тем, что целевая служба билета указана в незашифрованной части билета (поле `sname`). Следовательно, если изменить целевой сервис билета на другой сервис того же пользователя, билет будет работать для новой целевой службы.

Этот метод может быть полезен в некоторых ситуациях, например, если вы можете получить ST для администратора базы данных MSSQL на машине A (SPN = MSSQLSvc\machineA), вы можете изменить службу, чтобы она указывала на службу SMB той же самой машины. машина (SPN = CIFS\machineA) и получить доступ к машине A.

TGT
Чтобы получить ST от KDC, пользователь должен предъявить билет другого типа, TGT (билет на получение билетов). TGT — это как ST для KDC.

На самом деле, следуя принципу, согласно которому доступ к билету должен быть разрешен только целевому сервису, все TGT шифруются с помощью ключа учетной записи домена `krbtgt`, известного как ключ KDC. Следовательно, если вы можете получить ключ `krbtgt` (хранящийся в базе данных домена), вы можете создать собственные TGT, известные как Golden Ticket.

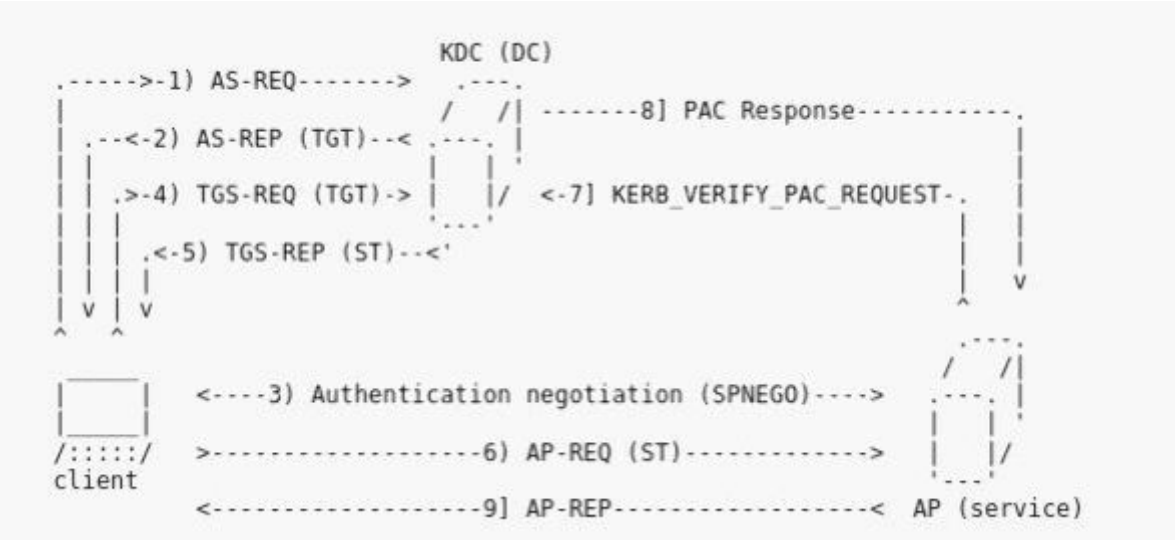
Поскольку вы можете создать билет для любого клиента, вы можете выдавать себя за любого пользователя в домене, включая администраторов. Golden Ticket можно даже использовать для компрометации всего леса, установив специальные привилегированные SID в PAC, например `Enterprise Admins`.

Это можно сделать, потому что PAC содержит данные безопасности, относящиеся к пользователю, и не проверяется, верна ли информация, поэтому вы можете добавить любого пользователя в любую группу внутри билета. и даже создавать билетыы для несуществующих пользователей.

Чтобы получить TGT, пользователю обычно необходимо пройти аутентификацию в KDC, используя его учетные данные.

Получение билетов билетов

Теперь, когда мы знаем об ST и TGT, давайте более подробно рассмотрим, как работает Kerberos, а значит, как выдаются билеты:



Процесс Kerberos

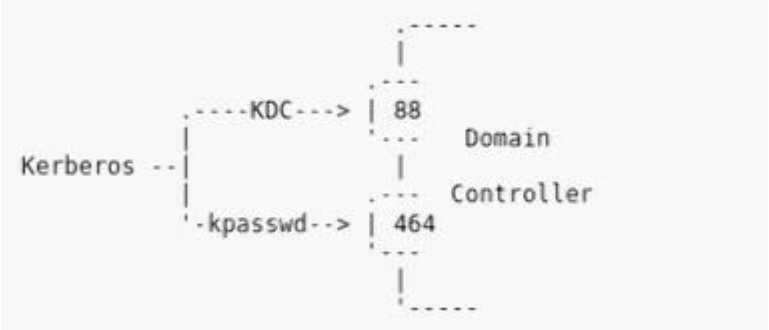
1. Клиент запрашивает TGT в AS (KDC), отправляя сообщение `AS-REQ`. В сообщение AS-REQ клиент может включить отметку времени, зашифрованную с помощью своего ключа Kerberos. Это называется предварительной аутентификацией Kerberos и иногда не требуется.
2. AS (KDC) проверяет метку времени и отвечает сообщением `AS-REP`, которое содержит две зашифрованные части: TGT, зашифрованный с помощью ключа KDC, и данные клиента, зашифрованные с помощью ключа клиента. Некоторая информация, такая как ключ сеанса, реплицируется в обеих частях, поэтому и пользователь, и KDC могут совместно использовать ее.
3. После этого клиент подключается к службе в точке доступа и согласовывает протокол аутентификации с `SPNEGO`. Если выбран Kerberos, клиенту необходимо получить ST для целевой службы.
4. Поэтому он запрашивает ST у KDC, отправляя `TGS-REQ`, который включает его

TGT и имя участника-службы целевой службы. Он также отправляет данные, зашифрованные с помощью сеансового ключа, такие как имя пользователя клиента и отметка времени, для проверки соединения.
5. KDC расшифровывает TGT своим ключом, получая таким образом доступ к имени пользователя и сеансовому ключу. KDC использует ключ сеанса для расшифровки имени пользователя, отправленного пользователем, чтобы проверить его правильность. Убедившись, что все правильно, KDC отвечает сообщением TGS-REP, которое содержит две зашифрованные части: ST для целевой службы, зашифрованную с помощью ключа пользователя службы, и данные клиента, зашифрованные с помощью ключа сеанса. Некоторая информация, такая как ключ сеанса службы, реплицируется в обеих частях, поэтому и пользователь, и служба могут совместно использовать ее.
6. Клиент отправляет ST службе в сообщении `AP-REQ` (внутри прикладного протокола). Служба расшифровывает ST и получает ключ сеанса службы и PAC. Служба будет использовать информацию безопасности PAC о клиенте, чтобы определить, есть ли у пользователя доступ к его ресурсам.
7. Если служба хочет проверить PAC, она может использовать протокол `Netlogon`, чтобы запросить у контроллера домена проверку подписи PAC с помощью KERB_VERIFY_PAC_REQUEST (необязательно).
8. Сервер проверит PAC и ответит кодом, указывающим, что PAC правильный (необязательно).
9. Наконец, если этого требует клиент, служба должна аутентифицировать себя, отвечая на сообщение AP-REQ сообщением AP-REP и используя сеансовый ключ в качестве доказательства того, что служба может расшифровать ST и, следовательно, это настоящий сервис, а не подделка (необязательно).

Kerberos, в отличие от NTLM, имеет сообщения, которые не включены в другой протокол приложения. Так обстоит дело с AS-REQ и TGS-REQ, которые отправляются непосредственно в DC.

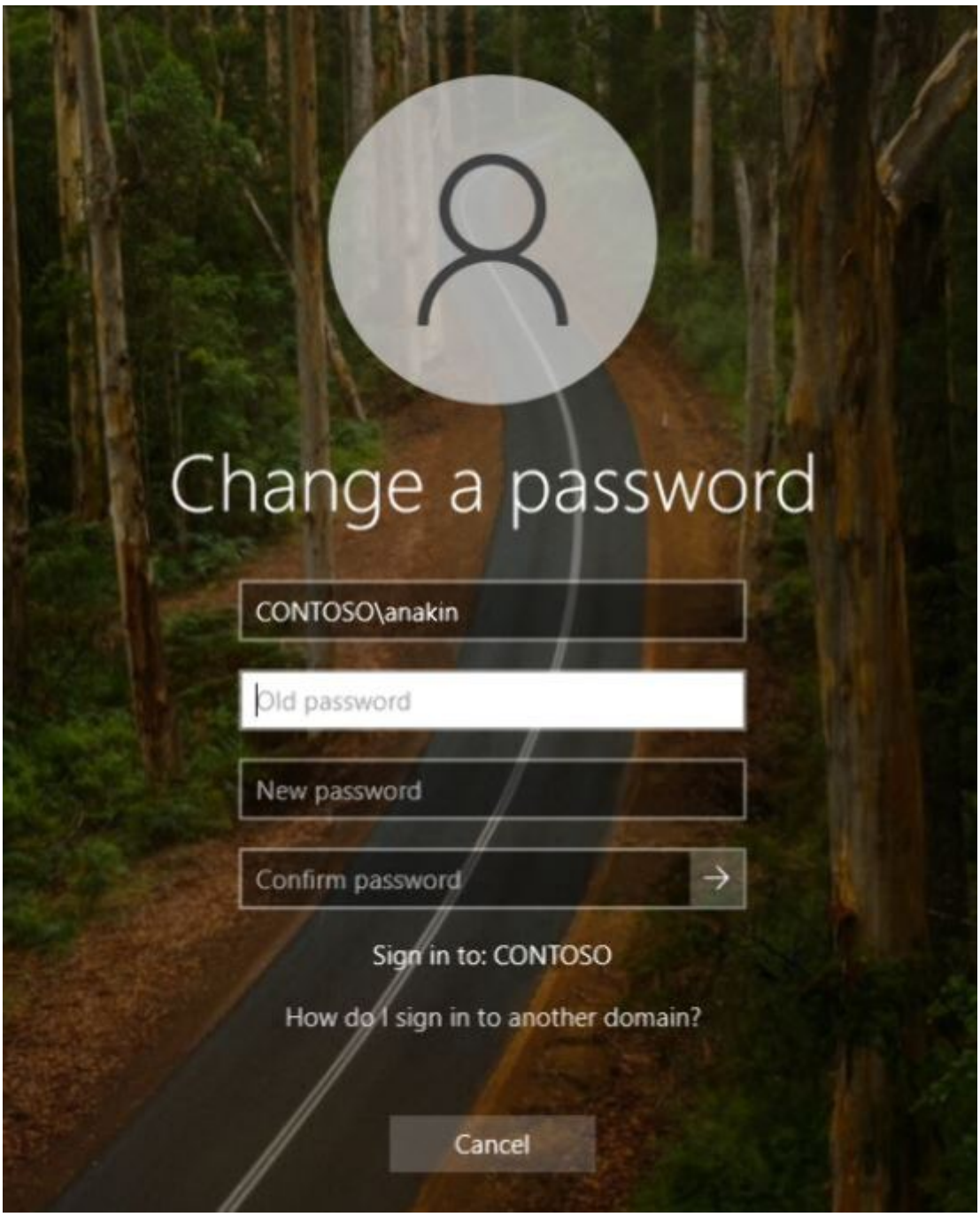
Службы Kerberos

Контроллер домена прослушивает Kerberos на портах `88/TCP` и `88/UDP`.



Порты Kerberos

Помимо KDC, в Kerberos есть еще одна служба, `kpasswd`, позволяющая менять пароли пользователей в домене. `kpasswd` можно найти в портах `464/TCP` и `464/UDP` контроллеров домена. Его можно использовать с утилитой ksetup, с экрана CTRL+ALT+DEL «Изменить пароль» или с помощью Rubeus changepw.



Утилита смены пароля использует kpasswd

Ключи Kerberos

Изменяя пароль, пользователь изменяет ключи Kerberos, используемые для шифрования сообщений и билетов Kerberos.

Существует много разных ключей, поскольку каждый из них используется для собственного алгоритма шифрования, используемого Kerberos. Возможные алгоритмы шифрования, используемые Kerberos, следующие:

- **RC4-HMAC**: ключ, используемый для RC4, представляет собой NT-хэш пользователя;
- **AES128-CTS-HMAC-SHA1-96**: Ключ, используемый для AES128, представляет собой хэш из 16 байтов, полученный из пароля пользователя (а также домена и имени пользователя);
- **AES256-CTS-HMAC-SHA1-96**: ключ, используемый для AES256, представляет собой хэш из 32 байтов, полученный из пароля пользователя (а также домена и имени пользователя);
- **DES-CBC-MD5**: этот ключ устарел, но ключ по-прежнему хранится в базе данных домена для пользователей.

В зависимости от выбранного алгоритма Kerberos использует нужный ключ, поэтому используя термин «ключ Kerberos» имеется в виду любой из возможных ключей, согласованных с помощью Kerberos. В Active Directory рекомендуется использовать AES256.

Базовые атаки Kerberos

Теперь, когда мы знаем основы Kerberos, давайте объясним некоторые атаки Kerberos.

Брутфорс Kerberos

Поскольку Kerberos — это протокол аутентификации, его можно использовать для проверки учетных данных пользователей в сети.

Кроме того, ошибки Kerberos очень многословны и позволяют выделить множество ситуаций при атаке методом полного перебора:

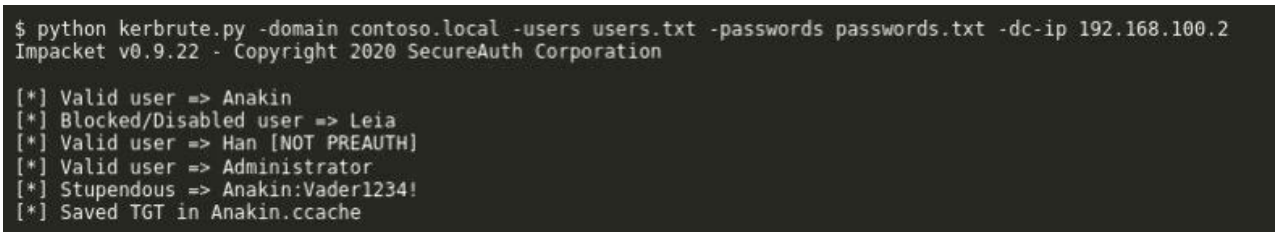
- `KDC_ERR_PREAUTH_FAILED` — неверный пароль;
- `KDC_ERR_C_PRINCIPAL_UNKNOWN` — недопустимое имя пользователя;
- `KDC_ERR_WRONG_REALM` — неверный домен;
- `KDC_ERR_CLIENT_REVOKED` — отключенный/заблокированный пользователь.

Проверяя сообщения об ошибках, вы можете не только проверить действительные учетные данные, но также перечислить учетные записи пользователей и узнать, не заблокировала ли ваша атака какую-либо учетную запись. Будьте осторожны, запуская такие атаки!

Еще одна вещь, которую следует иметь в виду, это то, что ошибки аутентификации регистрируются не с обычным событием ошибки входа в систему (код: `4625`), а с ошибкой предварительной аутентификации Kerberos (код: `4771`), которая поу молчанию не регистрируется.

Вы можете использовать Rubeus brute, kerbrute (Go), kerbrute (Python) или cerbero, чтобы запустить атаку грубой силы Kerberos.

$ python kerbrute.py -domain contoso.local -users users.txt -passwords passwords.txt -dc-ip 192.168.100.2
Impacket v0.9.22 — Copyright 2020 SecureAuth Corporation



Атака грубой силы Kerberos с помощью kerbrute.py

Kerberoasting

В Active Directory любой пользователь может запросить ST для любой службы, которая зарегистрирована в базе данных домена через SPN, независимо от того, запущена служба или нет. Более того, ST будет частично зашифрован ключом Kerberos (полученным из пароля) пользователя сервиса. Поэтому, если вы получили ST, вы можете попытаться взломать пароль пользователя службы, попытавшись расшифровать ST.

Большинство сервисов регистрируются в учетных записях компьютеров, которые имеют автоматически генерируемые пароли из 120 символов, которые меняются каждый месяц, поэтому взломать их ST невозможно.

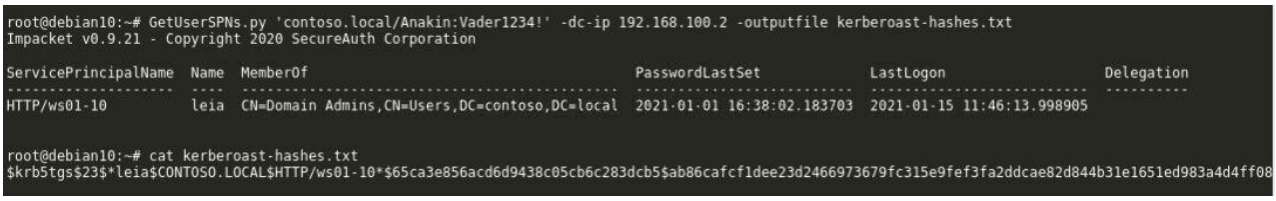
Однако иногда службы назначаются обычным учетным записям пользователей, которыми управляют люди, у которых могут быть слабые пароли. ST этих сервисов позволили бы взломать их, чтобы получить пароли пользователей.

Атака Kerberoast состоит в том, чтобы запрашивать ST для обычных учетных записей пользователей этих служб и пытаться взломать их, чтобы получить пароли пользователей. Обычно пользователи, у которых есть службы, также имеют и привилегии.

Проверить учетные записи пользователей с именами участников-служб можно с помощью любого клиента LDAP, используя следующий фильтр:

```
(&(samAccountType=805306368)(servicePrincipalName=*))
```

В частности, чтобы получить ST для взлома, вы можете использовать сценарий impacket GetUserSPNs.py, команду Rubeus kerberoast или сценарий Invoke-Kerberoast.ps1.



Kerberoast с GetUserSPNs.py

Когда у вас есть ST, вы можете попытаться взломать их с помощью hashcat. Чтобы их легко взломать, ST следует запрашивать в зашифрованном виде с помощью RC4, однако запросы могут быть обнаружены как обычный трафик (например, Microsoft ATA), поскольку для большинства запросов требуется шифрование AES256.

Также возможно выполнять Kerberoasting, не зная SPN служб. Помните, что вы можете запросить билет Kerberos для разных субъектов в домене, включая как службы, так и пользователей.

Таким образом, вы можете запросить билет для данного пользователя, который будет зашифрован с помощью ключа пользователя. Однако необходимо, чтобы у целевого пользователя была зарегистрирована какая-либо служба, чтобы получить для нее билет.

ST, полученный для пользователя в качестве целевого основного имени, также шифруется с помощью ключа пользователя, поэтому его можно использовать в Kerberoasting. Это также может быть полезно для перебора пользователей, поддерживающих kerberoaste, в случае, если вы не можете перечислить их через LDAP, поскольку основное имя пользователя без SPN не может быть разрешено.

Этот метод используется сценарием impacket GetUserSPNs.py. Его также можно использовать с командой Rubeus kerberoast с флагом /enterprise и командой cerbero kerberoast.

Кроме того, если у вас есть разрешение `Validated-SPN` для учетной записи пользователя, вы можете добавить имена участников-служб к этой учетной записи, сделав ее поддерживающей Kerberoasteable. Таким образом, вы можете запросить ST для этой службы учетной записи и попытаться взломать ее. По умолчанию учетные записи не имеют разрешений Validated-SPN на самих себя.

ASREProast

Большинству пользователей необходимо выполнить предварительную аутентификацию Kerberos, то есть отправить отметку времени, зашифрованную с помощью ключа Kerberos, в KDC в сообщении AS-REQ (для запроса TGT).

Однако в редких случаях предварительная аутентификация Kerberos отключается для учетной записи путем установки флага `DONT_REQUIRE_PREAUTH`. Таким образом, любой может выдать себя за эти учетные записи, отправив сообщение `AS-REQ`, и ответ `AS-REP` будет возвращен из KDC, что данные зашифрованы с помощью пользовательского ключа Kerberos.

```
AS-REP          ::= [APPLICATION 11] KDC-REP

KDC-REP         ::= SEQUENCE {
        pvno            [0] INTEGER (5),
        msg-type        [1] INTEGER (11 -- AS --),
        padata          [2] SEQUENCE OF PA-DATA OPTIONAL
        crealm          [3] Realm,
        cname           [4] PrincipalName,
        ticket          [5] Ticket, -- Encrypted with krbtgt key
        enc-part        [6] EncryptedData -- Encrypted with user key

}
```

Определение сообщения AS-REP

Вы не можете получить доступ к данным AS-REP напрямую, поскольку они зашифрованы с помощью ключа пользователя (который получен из пароля пользователя), но вы можете попытаться взломать офлайн-атаку, чтобы узнать пароль пользователя.

Атака ASREProast состоит в том, чтобы идентифицировать пользователей без предварительной аутентификации Kerberos и отправить AS-REQ от их имени, чтобы получить часть данных, зашифрованных с помощью ключа пользователя в сообщении AS-REP. После получения выполняется атака взлома в автономном режиме с целью восстановления пароля пользователя. Фильтр LDAP для пользователей без предварительной аутентификации Kerberos:

```
(&(samAccountType=805306368)
(userAccountControl:1.2.840.113556.1.4.803:=4194304))
```

Вы можете использовать такие инструменты, как сценарий impacket GetNPUsers.py, команду asreproast Rubeus или сценарий ASREEProast.ps1 для извлечения зашифрованных данных AS-REP.

```
$ GetNPUsers.py 'contoso.local/Anakin:Vader1234!' -dc-ip 192.168.100.2 -outputfile asreproast-hashes.txt
Impacket v0.9.21 - Copyright 2020 SecureAuth Corporation

Name  MemberOf  PasswordLastSet             LastLogon                   UAC
----  --------  --------------------------  --------------------------  --------
han             2020-12-16 10:53:35.177156  2021-05-12 09:19:28.469863  0x410200

root@debian10:~# cat asreproast-hashes.txt
$krb5asrep$23$han@CONTOSO.LOCAL:73eea4275625972c2e224648c4766b5a$1bbdaba56bb6eba4ea8cb565221de2fe2b5a8ade3d1155e33aa4d786624b84a62a100d97412361c851dfbf
```

Если у вас есть пользователь TGT, вы можете взломать его с помощью hashcat. Вы можете запросить AS-REP с шифрованием RC4, чтобы его было легче взломать.

Pass the Key/Over Pass the Hash

Как вы могли заметить, для запроса TGT пользователю нужно использовать не пароль, а ключ Kerberos. Следовательно, если злоумышленник может украсть ключ Kerberos (хэш NT или ключи AES), его можно использовать для запроса TGT от имени пользователя без необходимости знать пароль пользователя.

Обычно в Windows ключи Kerberos кэшируются в процессе `lsass`, и их можно получить с помощью команды `mimikatz sekurlsa::ekeys`. Также вы можете создать дамп процесса lsass с помощью таких инструментов, как procdump, sqldumper или других, и извлечь ключи в автономном режиме с помощью mimikatz.

В случае Linux ключи Kerberos хранятся в файлах `keytab`, чтобы их можно было использовать для службы Kerberos. Файл keytab обычно можно найти в `/etc/krb5.keytab`, или в значении, указанном переменными среды `KRB5_KTNAME` или `KRB5_CLIENT_KTNAME`, или указанном в файле конфигурации Kerberos в `/etc/krb5.conf`.

После этого вы можете скопировать его на свой локальный компьютер и/или получить список его ключей с помощью klist (Kerberos MIT) или cerbero.

```
$ klist -k -Ke
Keytab name: FILE:/etc/krb5.keytab
KVNO Principal
---- --------------------------------------------------------------------------
   1 r2d2@contoso.local (DEPRECATED:arcfour-hmac)  (0xc49a77fafad6d3a9270a8568fa453003)
```

Чтение keytab с помощью klist

Получив ключи Kerberos, вы можете запросить TGT в Windows с помощью команды Rubeus asktgt.

На компьютере с Linux вы можете использовать утилиты MIT Kerberos011, чтобы создать keytab с ключом и запросить TGT, или использовать ключ напрямую с помощью скрипта impacket getTGT.py или команды cerbero ask.

```
$ cerbero ask -u contoso.local/Anakin --aes ecce3d24b29c7f044163ab4d9411c25b5698337318e98bf2903bbb7f6d76197e -k 192.168.100.2 -vv
INFO - Request contoso.local/Anakin TGT for contoso.local
INFO - Save contoso.local/Anakin TGT for contoso.local in /root/Anakin.ccache
```

Pass-The-Key с cerbero

Билеты Kerberos имеют два формата: `ccache` и `krb`. Первый используется машинами Linux для хранения билетов (обычно в файлах). Формат krb используется в Windows для хранения билетов в памяти lsass, а также является форматом для передачи билетов по сети. Вы можете конвертировать билеты из одного формата в другой с помощью скрипта ticket_converter.py или команды cerbero convert.

```
$ python ticket_converter.py ~/Anakin.ccache ~/Anakin.krb
Converting ccache => kirbi
```

Конвертирование билета с помощью ticket_converter.py

Кроме того, вы можете вычислить ключи Kerberos из пароля пользователя с помощью команды cerbero hash. Также ключи AES можно рассчитать с помощью Get-KerberosAESKey.ps1 или хэша NT с помощью нескольких строк Python.

```
$ cerbero hash 'Vader1234!' -u contoso.local/Anakin
rc4:cdeae556dc28c24b5b7b14e9df5b6e21
aes128:18fe293e673950214c67e9f9fe753198
aes256:ecce3d24b29c7f044163ab4d9411c25b5698337318e98bf2903bbb7f6d76197e
```

Вычисление ключей Kerberos с помощью cerbero

Pass the Ticket

Техника `Pass the Ticket` заключается в краже билета и связанного сеансового ключа и использовании их для олицетворения пользователя для доступа к ресурсам или услугам. Можно использовать как TGT, так и ST, но TGT предпочтительнее, поскольку они позволяют получить доступ к любой службе (используя ее для запроса ST) от имени пользователя, тогда как ST ограничены только одной службой (или несколькими, если SPN изменен на другой сервис того же пользователя).

В Windows билеты можно найти в памяти процесса lsass и извлечь их с помощью команды `mimikatz sekurlsa::tickets` или команды дампа Rubeus. Другая возможность состоит в том, чтобы сбросить процесс lsass с помощью таких инструментов, как procdump, sqldumper или других, и извлечь билеты в автономном режиме с помощью mimikatz или pypykatz. Эти команды извлекают билеты в формате krb.

```
PS C:\> .\procdump.exe -accepteula -ma lsass.exe lsass.dmp

ProcDump v10.0 - Sysinternals process dump utility
Copyright (C) 2009-2020 Mark Russinovich and Andrew Richards
Sysinternals - www.sysinternals.com

[12:03:17] Dump 1 initiated: C:\lsass.dmp
[12:03:18] Dump 1 writing: Estimated dump file size is 34 MB.
[12:03:18] Dump 1 complete: 34 MB written in 1.0 seconds
[12:03:18] Dump count reached.
```

Сброс памяти lsass с помощью procdump

```
$ pypykatz lsa minidump lsass.dmp -k /tmp/kerb > output.txt
INFO:root:Parsing file lsass.dmp
INFO:root:Writing kerberos tickets to /tmp/kerb
$ ls /tmp/kerb/
 lsass.dmp_51a1d3f3.ccache                                                    'TGS_CONTOSO.LOCAL_WS02-7$_WS02-7$_29a9c991.kirbi'
 lsass.dmp_c9a82a35.ccache                                                     TGT_CONTOSO.LOCAL_anakin_krbtgt_CONTOSO.LOCAL_6483baf5.kirbi
 TGS_CONTOSO.LOCAL_anakin_LDAP_dc01.contoso.local_contoso.local_f8a46ad5.kirbi 'TGT_CONTOSO.LOCAL_WS02-7$_krbtgt_CONTOSO.LOCAL_740ef529.kirbi'
'TGS_CONTOSO.LOCAL_WS02-7$_cifs_dc01.contoso.local_b9833fa1.kirbi'            'TGT_CONTOSO.LOCAL_WS02-7$_krbtgt_CONTOSO.LOCAL_77d63cf0.kirbi'
'TGS_CONTOSO.LOCAL_WS02-7$_cifs_dc01.contoso.local_bfed6415.kirbi'            'TGT_CONTOSO.LOCAL_WS02-7$_krbtgt_CONTOSO.LOCAL_7ac74bd6.kirbi'
'TGS_CONTOSO.LOCAL_WS02-7$_ldap_dc01.contoso.local_contoso.local_2129bc1c.kirbi' 'TGT_CONTOSO.LOCAL_WS02-7$_krbtgt_CONTOSO.LOCAL_fdb8b40a.kirbi'
'TGS_CONTOSO.LOCAL_WS02-7$_LDAP_dc01.contoso.local_contoso.local_719218c6.kirbi'
```

Получение билетов из дампа lsass с помощью pypykatz

С другой стороны, на Linux-машинах, входящих в домен, билеты хранятся по-другому. Билеты обычно можно найти по умолчанию в каталоге `/tmp` в файлах с форматом, `krb5cc_%{uid}`, где uid — это uid пользователя. Чтобы получить билеты, просто скопируйте файлы (если у вас есть права). Однако также возможно, что билеты хранятся в ключах ядра Linux, а не в файлах, но вы можете получить их с помощью tickey.

Чтобы быть уверенным, где билеты хранятся на компьютере с Linux, вы можете проверить файл конфигурации Kerberos `/etc/krb5.conf`.

Чтобы использовать билеты на компьютере с Windows, вы должны внедрить их в процесс lsass, что можно сделать с помощью команды `mimikatz kerberos::ptt` или команды `Rubeus ptt`. Эти утилиты читают билеты в формате krb.

```
PS C:\> .\mimikatz.exe

  .#####.   mimikatz 2.2.0 (x64) #19041 Sep 18 2020 19:18:29
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz # kerberos::ptt pikachu-tgt.kirbi

 * File: 'pikachu-tgt.kirbi': OK
```

Внедрение TGT в текущую сессию Windows

После внедрения билетов в сеанс вы можете использовать любой инструмент для выполнения действий, выдавая себя за пользователя по сети, например psexec.

В Linux вы можете использовать билеты с утилитами impacket, указав переменную среды `KRB5CCNAME` на файл билета. Затем используйте утилиты с параметрами `impacket -k -no-pass`. Здесь нужны билеты в формате ccache.

Чтобы преобразовать билеты между форматами krb (Windows) и ccache (Linux), вы можете использовать скрипт ticket_converter.py или команду cerbero convert.

Golden/Silver ticket

В Active Directory TGT Kerberos шифруются с помощью ключей учетной записи krbtgt. Если знать ключи можно создать пользовательские TGT, известные как Golden Tickets.

Чтобы получить ключи krbtgt, вам необходимо получить доступ к базе данных Active Directory. Вы можете сделать это, выполнив удаленную атаку `dcsync` с помощью команды `mimikatz lsadump::dsync` или скрипта impacket secretsdump.py, либо

локально выгрузив файл NTDS.dit с помощью ntdsutil или vssadmin.

```
$ secretsdump.py 'contoso.local/Administrator@192.168.100.2' -just-dc-user krbtgt
Impacket v0.9.21 - Copyright 2020 SecureAuth Corporation

Password:
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:fe8b03404a4975e7226caf6162cfccba:::
[*] Kerberos keys grabbed
krbtgt:aes256-cts-hmac-sha1-96:5249e3cf829c979959286c0ee145b7e6b8b8589287bea3c83dd5c9488c40f162
krbtgt:aes128-cts-hmac-sha1-96:a268f61e103134bb7e975a146ed1f506
krbtgt:des-cbc-md5:0e6d79d66b4951cd
[*] Cleaning up...
```

Ключи krbtgt, полученные с помощью secretsdump.py

Точно так же можно создать пользовательский ST для службы, известной как Silver Ticket, если мы получим ключи Kerberos пользователя службы. Ключи для пользователя службы можно получить, изучив процесс lsass на машинах домена, на которых пользователь вошел в систему, выполнив Kerberoast, создав дамп базы данных Active Directory и т. д.

И в Golden Ticket, и в Silver Ticket билете можно создать билет для любого пользователя в домене и даже для несуществующего. Более того, мы можем дать высокие привилегии пользователю билета, изменив группы пользователей PAC и включив, например, группу «Администраторы домена», чтобы иметь привилегии администраторов домена.

Кроме того, мы должны подписать PAC билета с помощью ключа krbtgt, но это невозможно сделать для Silver Ticket, поскольку мы просто знаем ключ службы пользователя, здесь используется поддельная подпись, поскольку для службы довольно странно проверять с помощью контроллера домена подпись ПАК.

Для создания Golden Ticket и Silver ticket билетов вы можете использовать команду `mimikatz kerberos::golden` или скрипт impacket ticketer.py. Затем вы можете использовать их как любой билет. Если можете, используйте ключ AES256, чтобы избежать обнаружения такими решениями, как ATA.

```
$ ticketer.py -domain-sid S-1-5-21-1372086773-2238746523-2939299801 -domain contoso.local Administrator -aes 5249e3cf829c979959286c0ee145b7e6b8b8589287
Impacket v0.9.21 - Copyright 2020 SecureAuth Corporation

[*] Creating basic skeleton ticket and PAC Infos
[*] Customizing ticket for contoso.local/Administrator
[*]     PAC_LOGON_INFO
[*]     PAC_CLIENT_INFO_TYPE
[*]     EncTicketPart
[*]     EncAsRepPart
[*] Signing/Encrypting final ticket
[*]     PAC_SERVER_CHECKSUM
[*]     PAC_PRIVSVR_CHECKSUM
[*]     EncTicketPart
[*]     EncASRepPart
[*] Saving ticket in Administrator.ccache
```

Создаем Golden Ticket с помощью ticketer.py

После создания Golden Ticket дают вам права, указанные в билете, позволяя вам выдавать себя за любого пользователя, даже несуществующего, в домене и получать доступ к любой службе в домене. Следует иметь в виду, что после создания 3Golden Ticket его необходимо использовать в течение 20 минут, в противном случае KDC проверит информацию PAC, чтобы убедиться, что она верна.

Случаем использования Silver Ticket является доступ к компьютеру в качестве администратора, когда у вас есть пароль учетной записи домена компьютера. У вас нет прав администратора на компьютере с его учетной записью домена компьютера, но вы можете использовать его пароль для создания Silver Ticket для его служб и выдачи себя за администратора.

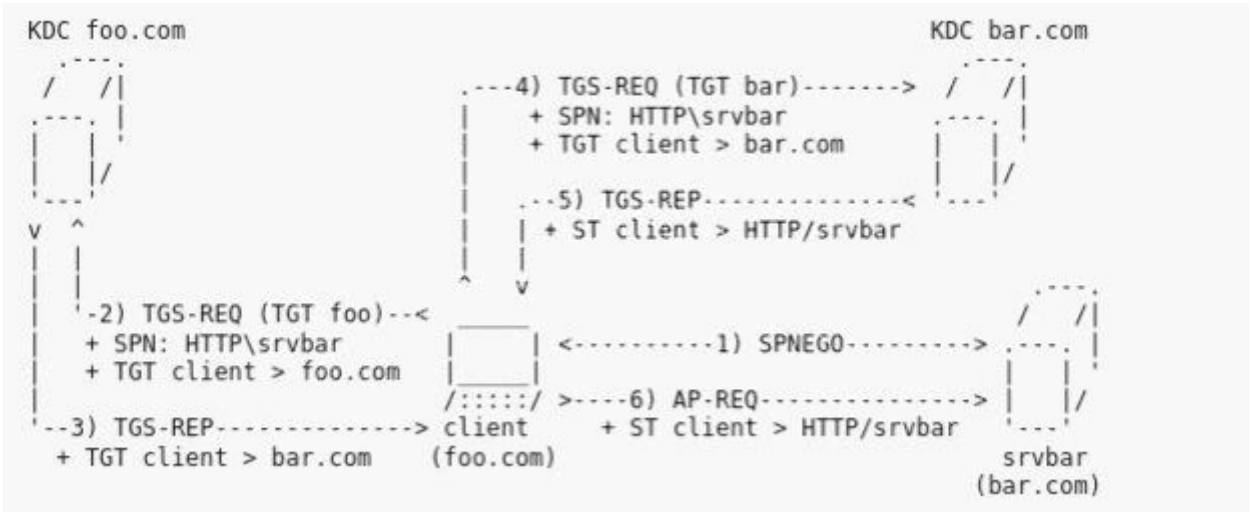
Таким образом, Silver Tickets можно использовать для доступа к сервису одного пользователя, а Golden Ticket — для доступа к любому сервису в домене.

Kerberos между доменами

Golden Ticket также можно использовать для компрометации всего леса. Но перед этим давайте рассмотрим, как работает Kerberos в доверенных доменах.

Ранее мы видел, что пользователь домена может получить доступ к службам в доверительных доменах (используя входящие или двунаправленные доверительные отношения). Процесс доступа к ресурсам внешнего домена также требует аутентификации, которую может обеспечить Kerberos.

Однако KDC (DC) может выдавать ST только для служб в своем домене, так как же работает Kerberos между доменами? Что ж, необходимо запросить ST на сервер DC внешнего домена, а для этого нам потребуется TGT для этого сервера. TGT для внешнего KDC, известного как межобластной TGT, выдается нашим KDC, когда мы запрашиваем ST для службы в другом домене. Шаги следующие:



Kerberos между доменами

- Клиент/пользователь из домена `foo.com` использует SPNEGO для согласования проверки подлинности Kerberos с желаемой службой, в данном случае `HTTP\srvbar` (веб-сервер на сервере srvbar) из домена `bar.domain`;
- Клиент запрашивает ST, используя свой TGT `foo.com`, для `HTTP\srvbar` в свой KDC, отправляя сообщение TGS-REQ;
- KDC определяет, что эта служба находится в доверенном домене `bar.com`. Таким образом, KDC `foo.com` создает TGT для `bar.com`, используя в качестве ключа шифрования (и подписи PAC) ключ межобластного доверия (секретный ключ, совместно используемый обеими сторонами доверия). Затем KDC возвращает TGT `bar.com` в сообщении TGS-REP. PAC, включенный в TGT `bar.com`, является копией TGT PAC `foo.com`;
- Клиент использует TGT `bar.com`, чтобы запросить HTTP\srvbar ST у KDC `bar.com`, отправив сообщение TGS-REQ. KDC `bar.com` проверяет билет, расшифровывая его с помощью межобластного ключа доверия.
- Затем он создает ST для HTTP\srvbar для клиента. Когда создается новый ST, PAC из TGT копируется и при необходимости фильтруется. Обычно удаляются лишние SID, не входящие в лес доверенного домена;
- Наконец, клиент использует ST для аутентификации в службе HTTP\srvbar.

Любопытно, что обычно TGT между областями шифруются с использованием алгоритма RC4 вместо AES256.

Атака на историю SID

Что интересно в этом процессе, так это то, как PAC копируется между заявками при междоменном взаимодействии. Такое поведение может позволить злоумышленнику, способному создать Golden Ticket для домена, скомпрометировать весь лес.

Как мы видели ранее, в PAC есть поле для включения дополнительных SID, которые идентифицируют специальные объекты. Это поле обычно используется для включения тех SID, которые хранятся в атрибуте `SIDHistory`.

История SID используется для миграции. Когда пользователь мигрирует из домена в другой, привилегии пользователя сбрасываются, создается новый SID, пользователь добавляется в новые группы и т. д. Однако SID из групп, к которым пользователь принадлежит в старом домене, сохраняются. в атрибуте История SID.

Затем, когда пользователь хочет получить доступ к ресурсам в старом домене, его исторические SID добавляются в поле дополнительных SID PAC. Таким образом, старый домен может просмотреть эти SID и предоставить пользователю его старые привилегии, что позволит ему получить доступ к ресурсам старого домена.

Однако дополнительные SID могут быть опущены (не скопированы в ST PAC) в соответствии с политикой фильтрации SID. Как правило, домены разрешают SID из других доменов в лесу (по умолчанию), но отбрасывают дополнительные SID из внешних лесов в соответствии с правилом `ForestSpecific`, поскольку лес является

границей безопасности Active Directory. Кроме того, домены одного и того же леса также могут быть помещены в карантин, что позволит удалить лишние SID с помощью политики `QuarantinedWithinForest`.

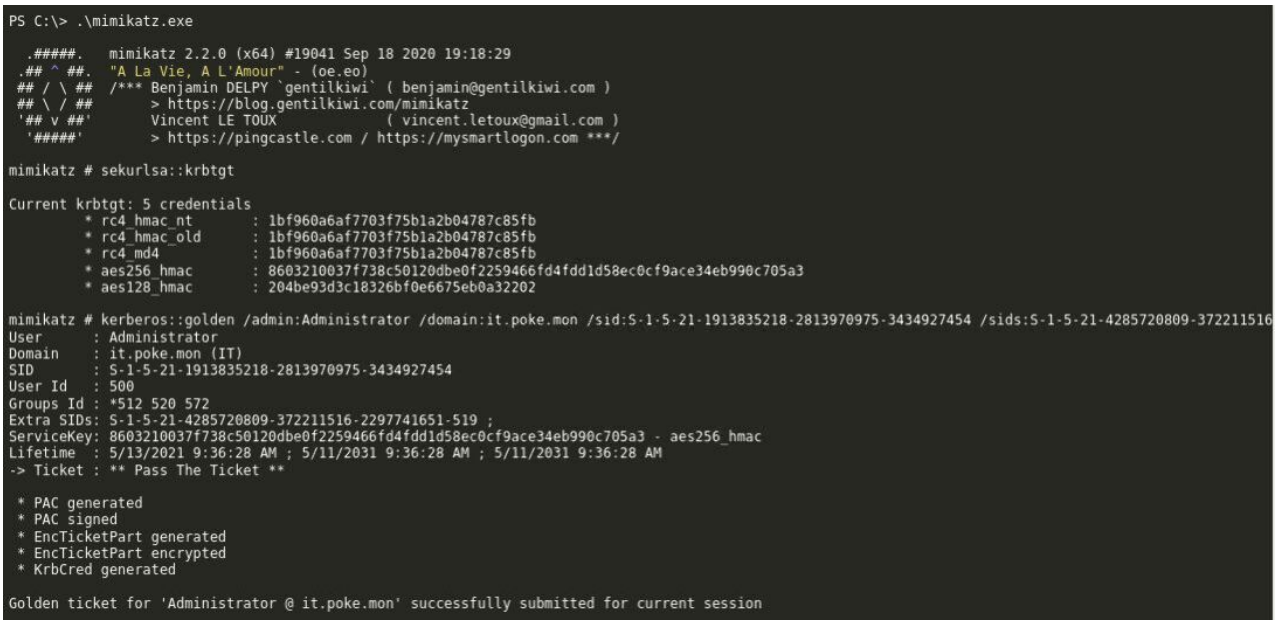
Напротив, история SID может быть включена в доверии между доменами разных лесов с некоторыми ограничениями. Допускаются группы с SID целевого (доверяющего) леса, у которых RID выше `1000`. Следовательно, административные группы, такие как «Администраторы домена» (RID = 512), чей RID ниже 1000, фильтруются, но группы с более высоким RID, принадлежащие этим административным группам (становятся также административными группами), такие как административные группы `Exchange`.

Затем, если история SID редактируется, могут быть введены административные привилегии для других доменов. Например, если вы внедрите SID `Enterprise Admins` в историю SID пользователя, то пользователь может иметь административные привилегии во всем лесу. Атрибут SID History можно редактировать непосредственно в базе данных Active Directory с помощью команды `mimikatz misc::addsid`.

Однако, как мы уже говорили ранее, история SID копируется в PAC TGT, поэтому, если мы можем создать Golden Ticket, мы можем внедрить нужные SID истории непосредственно в атрибут дополнительных SID PAC. Затем, когда мы используем этот билет «Golder», его PAC копируется в межобластной TGT. Впоследствии, при использовании этого межобластного TGT для получения ST для службы во внешнем домене, если этот домен находится в том же лесу, привилегированные SID могут быть скопированы в ST PAC, что дает нам привилегии, которые у нас были изначально вводится в билет Golder.

Интересным SID для внедрения является один из «Администраторов предприятия», эта группа существует только в корневом домене леса и по умолчанию добавляется как член всех групп «Администраторы домена» всех доменов в лесу.

На самом деле, если вы скомпрометируете корневой лес домена и создадите Golden Ticket, который включает группу «Администраторы предприятия» (чей RID равен 519 и включен по умолчанию для impacket и mimikatz), вам не нужно создавать Golden Ticket с дополнительными SID, поскольку у вас уже есть разрешения на управление всем лесом, даже доменами, помещенными в карантин (поскольку нет дополнительных SID для фильтрации). Добавление «Enterprise Admins» к дополнительным SID необходимо только в том случае, если вы скомпрометировали некорневой домен и хотите скомпрометировать другой домен леса, за исключением доменов, помещенных в карантин, которые фильтруют дополнительные SID.



Pass-The-Ticket с корпоративными администраторами в дополнительных SID

Тем не менее, для выполнения атаки dcsync в другом домене, возможно, с использованием групп SID «Контроллеры домена предприятия» (S-1-5-9) и «Контроллеры домена» (S-1-5-21-domain-516).

Для создания Golden Ticket вы можете использовать команду `mimikatz kerberos::golden` или скрипт impacket ticketer.py, аналогичный созданию Golden Ticket, но с добавлением дополнительных SID. Если можете, используйте ключ AES256, чтобы избежать обнаружения такими решениями, как ATA.

Межрегиональный TGT

Как мы видели, использование Kerberos для операций между областями вводит новый тип TGT, межобластной TGT. Этот TGT точно такой же, как и обычный, за исключением того, что он зашифрован ключом доверия между областями, который является секретным ключом, позволяющим обеим сторонам доверия общаться между собой. Секретный ключ хранится как ключ учетной записи пользователя, представляющей доверие.

Чтобы получить ключ доверия между областями, обычно вам нужно сделать дамп базы данных домена. Более того, есть одна ситуация, когда через Kerberoast можно было получить trust-key.

Когда создаетсядоверие, пароль может быть выбран человеком, поэтому существует вероятность того, что будет установлен слабый пароль. Затем вы можете получить межобластной TGT, зашифрованный с помощью ключа доверия, и попытаться взломать его, чтобы получить пароль доверия (который используется для генерации всех ключей доверия Kerberos). Но помните, что пароль доверия, как и машинные пароли, обычно меняются каждые 30 дней.

Наконец, после получения ключа доверия для создания билета между областями можно использовать команду `mimikatz kerberos::golden` или скрипт impacket ticketer.py. Затем вы можете использовать его как любой билет. Билеты между трастами шифруются с помощью ключа RC4, то есть хэша NT доверительной учетной записи.

Делегация Kerberos

Как нам кажется, Kerberos позволяет пользователям аутентифицироваться и получать доступ к сервису по всему домену или даже в других доменах. Однако иногда службам, к которым осуществляется доступ, необходимо олицетворять пользователя, чтобы общаться с третьей службой.

Например, веб-сервер, на котором зарегистрирован пользователь (с помощью Kerberos), должен выполнять некоторые действия в базе данных от имени пользователя. Однако, когда пользователь аутентифицируется на веб-сервере, он получает пользовательский ST только для себя, но для олицетворения пользователя ему также требуется пользовательский ST для службы базы данных.

Чтобы справиться с этой ситуацией, можно использовать делегирование Kerberos. Эта функция предоставляет механизмы, позволяющие службе получать ST для третьей службы от имени клиента.

Для выполнения делегирования Kerberos в Active Directory существует два подхода:

Неограниченное делегирование
Подразумевает отправку пользовательского TGT внутри ST службе, что позволяет ей полностью олицетворять клиента по отношению к KDC с помощью клиентского TGT.
Ограниченное делегирование
Предоставляет механизмы, расширения службы для пользователя (S4U), которые позволяют службе запрашивать ST от имени пользователя без необходимости использования клиентского TGT и только для определенных разрешенных служб.

Поговорим о том, как работает делегирование Kerberos. Но сначала давайте рассмотрим меры против делегирования, которые мешают делегированию работать.

Меры против делегирования Kerberos

Существует два механизма, позволяющих избежать делегирования определенной учетной записи пользователя (олицетворения в Kerberos):

- установка флага `NOT_DELEGATED` в атрибуте `UserAccountControl` учетной записи пользователя;
- добавление пользователя в группу защищенных пользователей.

Любая из этих мер применяется к учетной записи пользователя, делегирование будет невозможно. Поэтому важно знать, какие учетные записи защищены, и чтобы найти их, вы можете использовать запрос LDAP со следующим фильтром:

```
(|
  (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

Чтобы найти защищенные делегированные учетные записи, вы можете использовать такие инструменты, как Powerview, модуль Powershell ActiveDirectory или ldapsearch.

Неограниченное делегирование Kerberos

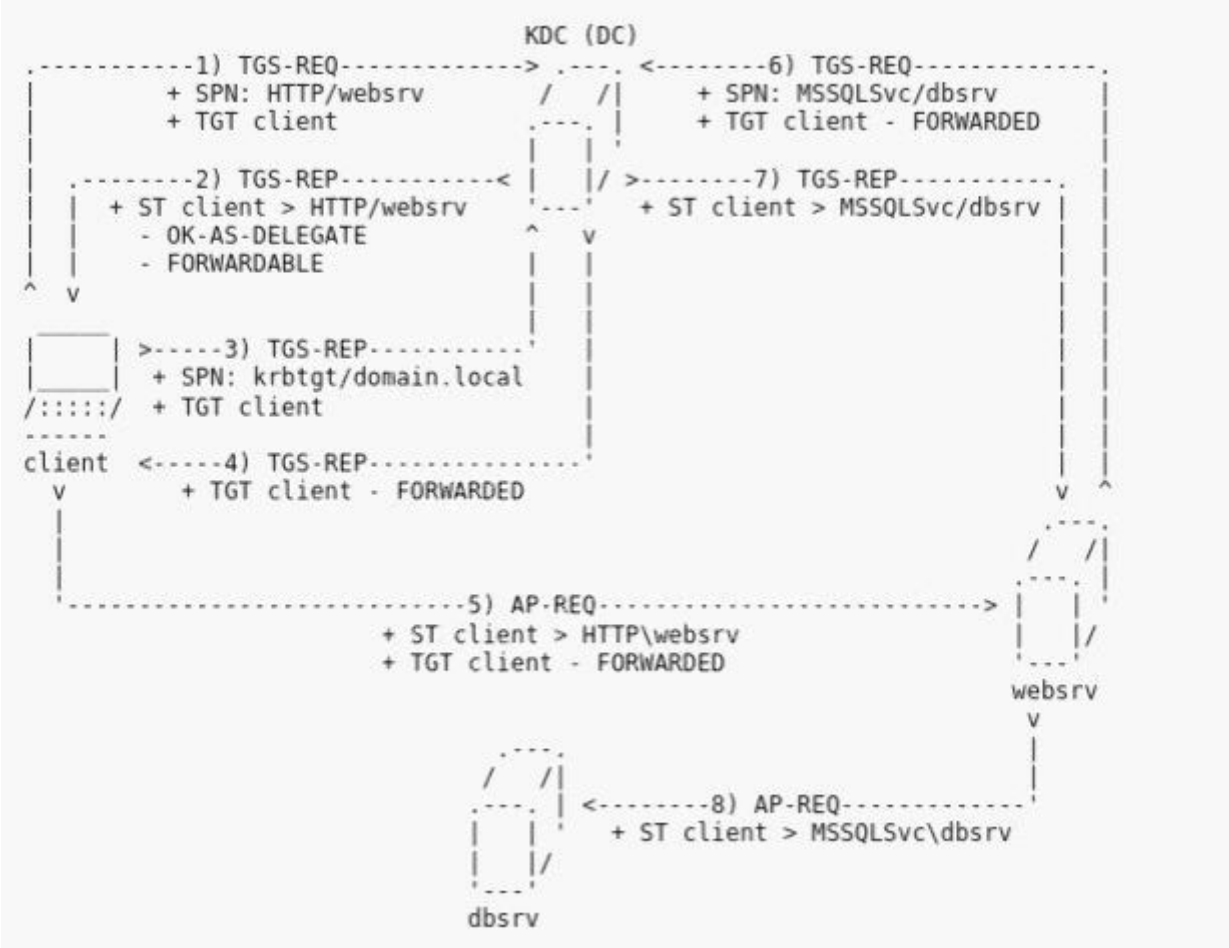
При неограниченном делегировании Kerberos служба может олицетворять пользователя клиента, поскольку при этом службе отправляется собственный TGT. Затем служба может использовать TGT пользователя (без каких-либо ограничений) для запроса новых ЗБ для других служб от имени клиента.

KDC устанавливает флаг `OK-AS-DELEGATE` в ST для любой службы, владелец которой — пользователь службы, имеющей установленный флаг `UserAccountControl TRUSTED_FOR_DELEGATION`. Проверяя флаги OK-AS-DELEGATE и FORWARDABLE, клиент знает, должен ли он запрашивать отправку TGT целевой службе, чтобы разрешить неограниченное делегирование.

В случае клиентов, являющихся членами группы «Защищенные пользователи» или имеющих установленный флаг `UserAccountControl NOT_DELEGATED`, когда флаг `FORWARDABLE` в протоколе ST.

Кроме того, для установки флага TRUSTED_FOR_DELEGATION в учетной записи пользователя требуется SeEnableDelegationPrivilege.

Давайте рассмотрим пример:



Неограниченный процесс делегирования

- клиент запрашивает ST для службы `HTTP\websrv` (веб-служба на сервере websrv), используя свой TGT. Служба HTTP\websrv принадлежит пользователю `websrv$` (помните, что имена пользователей `[[#computer-accounts][computer account]]` заканчиваются на `=$=`);
- KDC проверяет, установлен ли флаг TRUSTED_FOR_DELEGATION для `websrv$`. Следовательно, KDC возвращает клиенту ST для HTTP\websrv с флагом OK-AS- DELEGATE (и FORWARDABLE);
- клиент проверяет флаг OK-AS-DELEGATE, указывающий, что служба использует делегирование, поэтому он решает запросить у KDC сообщение FORWARDED TGT для отправки службе;
- KDC возвращает TGT с установленным флагом FORWARDED;
- клиент отправляет ST с включенным FORWARDED TGT в websrv, чтобы получить доступ к службе `HTTP\websrv`;
- Иногда `HTTP\websrv` должен олицетворять клиента для доступа к службе базы данных, расположенной в dbsrv. Поэтому веб-служба запрашивает ST для MSSQLSvc\dbsrv от имени клиента, используя полученный клиентский TGT;
- KDC возвращает клиенту ST для доступа к службе `MSSQLSvc\dbsrv`;
- наконец, служба `HTTP\websrv` использует ST для доступа к `MSSQLSvc\dbsrv`, выдавая себя за клиента.

Вероятно, наиболее важным фактом, который следует иметь в виду, является то, что любой ST, который будет отправлен на HTTP\websrv, будет содержать TGT от клиента. Поэтому, если кто-то скомпрометирует сервер websrv, он сможет получить все эти TGT и использовать их для олицетворения любого из клиентов с помощью атаки Pass the Ticket.

Для получения билетов с компьютера Windows (включая делегированные TGT) можно использовать команду `mimikatz sekurlsa::tickets` или команду дампа Rubeus. Другой подход заключается в сбросе процесса lsass с помощью таких инструментов, как procdump, sqldumper или других, и извлечении билетов в автономном режиме с помощью mimikatz или pypykatz.

Но помните, что TGT включены во все ЗБ для служб учетной записи, для которой установлен флаг `UserAccountControl TRUSTED_FOR_DELEGATION`. Поэтому в предыдущем примере, где учетная запись компьютера websrv$ была владельцем службы HTTP\websrv, любой ST, запрошенный для любой другой службы websrv$, такой как, например, CIFS\websrv (для доступа к общим ресурсам SMB), также будет содержать клиент TGT.

Для идентификации учетных записей с неограниченным делегированием вы можете использовать следующий фильтр LDAP:

```
(UserAccountControl:1.2.840.113556.1.4.803:=524288)
```

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

Таким образом, если вы скомпрометируете сервер, учетная запись которого имеет неограниченное делегирование, вы сможете собрать TGT всех клиентов, которые к нему подключаются. Вы можете использовать несколько методов фишинга, чтобы заставить пользователей подключаться к вашему серверу, например, создавать файлы, которые подключаются к вашей скомпрометированной машине для получения TGT Kerberos, аналогично методам, используемым для взлома хэшей NTLM.

Кроме того, вы можете получить TGT учетной записи компьютера, заставив его подключиться к вашему серверу с ошибкой принтера. Ошибка принтера использует вызов RPC интерфейса RPRN RPC, который позволяет любым « аутентифицированным пользователям » указать целевому компьютеру сервер для подключения через SMB.

Чтобы вызвать ошибку принтера, вы можете использовать инструмент SpoolSample или скрипт printerbug.py. Вы должны передать имя хоста в качестве аргумента, чтобы целевая машина могла использовать Kerberos. Если вы предоставите IP-адрес, то для аутентификации будет использоваться NTLM, и делегирование

выполняться не будет. Кроме того, вы можете сканировать компьютеры, на которых включена служба спулинга (по умолчанию она включена), с помощью скрипта SpoolerScan.ps1.

Отслеживать появление TGT можно с помощью команды Rubeus.

Для этого нам нужно изменить SPN учетной записи неограниченного делегирования, которую мы скомпрометировали. Плохая новость заключается в том, что для этого нам нужно разрешение Validated-SPN, которое по умолчанию не предоставляется собственной учетной записи. Однако в случае учетных записей компьютеров они могут добавлять имена участников-служб по умолчанию, соответствующие их именам хостов, которые, к счастью, включают имена хостов, добавленные в `msDS-AdditionalDnsHostName`, которые могут быть изменены самой учетной записью. Затем мы можем добавить в качестве нового имени хоста имя хоста нашей машины и создать таким образом имена участников-служб, которые указывают на нашу машину. Мы можем сделать это с помощью addspn.py. Также мы можем добавить SPN с помощью утилиты setspn.

Чтобы имя хоста указывало на нашу машину, мы можем создать пользовательскую запись ADIDNS с помощью Powermad или dnstool.py.

Затем мы можем использовать принтер или методы фишинга, чтобы заставить пользователей аутентифицироваться на нашем сервере. Наконец, чтобы вспомнить TGT, мы можем использовать krbrelayx.

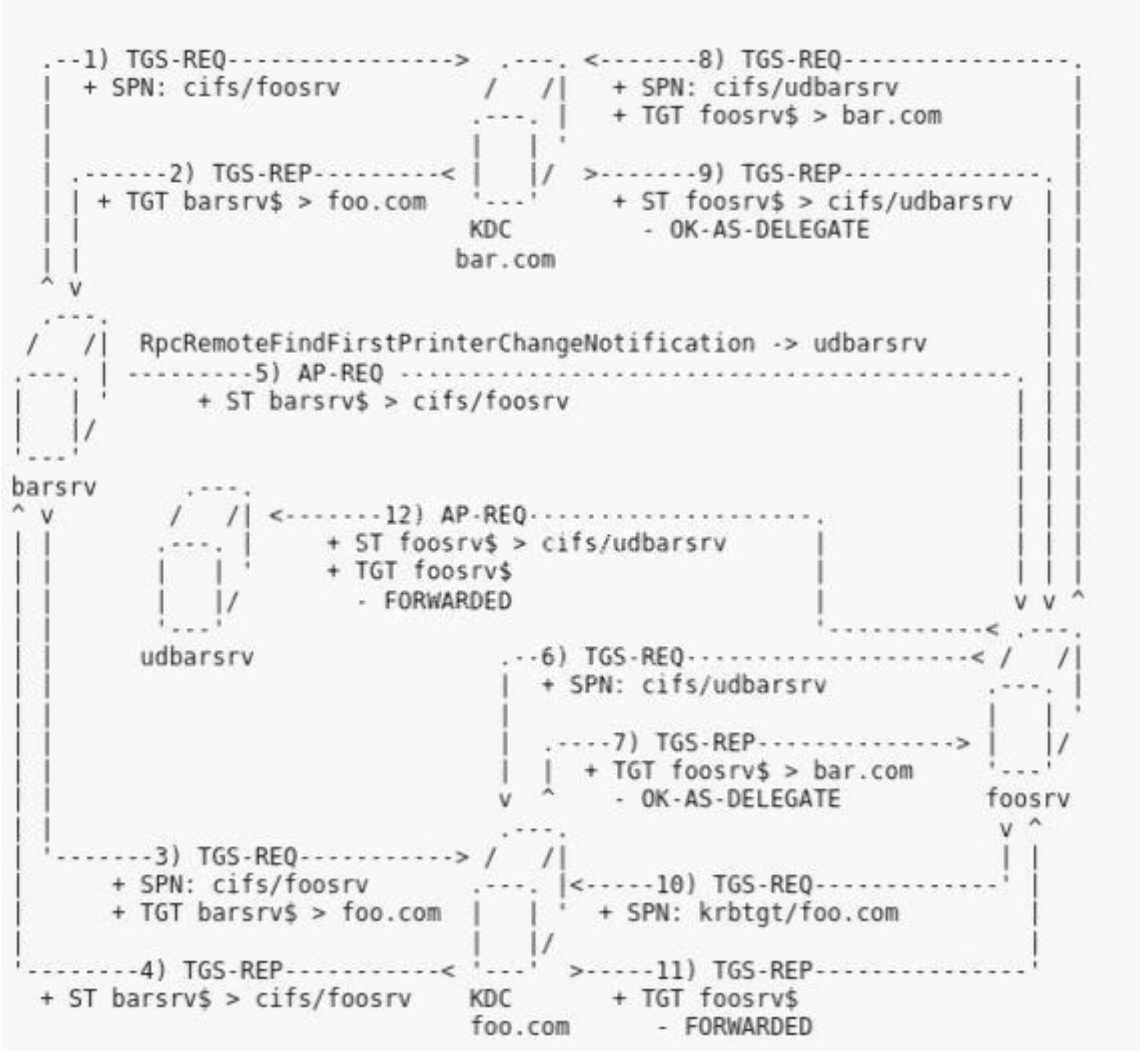
Очень интересным случаем, позволяющим скомпрометировать домен, является выполнение ошибки принтера на контроллере домена, чтобы заставить его подключиться к нашему скомпрометированному серверу. Таким образом, вы можете получить TGT для учетной записи DC и использовать его для запуска атаки DCSync.

Неограниченное делегирование Kerberos между лесами

На самом деле, этот метод также можно использовать для двунаправленных доверительных отношений между лесами, в которых включено делегирование TGT, чтобы скомпрометировать другой лес. Обычно делегирование TGT было включено по умолчанию, но Microsoft выпустила исправление, которое отключает делегирование TGT по умолчанию.

В следующей последовательности описываются сообщения Kerberos, участвующие в атаке с ошибкой принтера между доменами в случае включения делегирования TGT. Злоумышленник отправляет вызов RPC от barsrv к foosrv, чтобы указать этому последнему подключиться к udbarsrv, который имеет неограниченное делегирование. После этого TGT foosrv$ (пользователь домена foosrv) может быть получен в udbarsrv.

Шаги следующие:



Сообщения Kerberos в ошибке принтера в разных доменах (делегирование включено)

- barsrv (из домена bar.com) отправляет TGS-REQ в KDC `bar.com` с запросом ST для службы SMB (cifs) foosrv (поскольку ошибка принтера использует RPC через SMB);
- KDC `bar.com` проверяет, находится ли запрошенная служба в доверенном домене `foo.com` и выдает TGT barsrv$ для этого домена;
- Затем barsrv использует свой TGT `foo.com`, чтобы запросить у KDC ST `foo.com` для службы `cifs/foosrv`;
- Затем KDC `foo.com` возвращает ST для barsrv$ для cifs/foosrv;
- Затем barsrv аутентифицируется с помощью foosrv и выполняет вызов ошибки принтера `RpcRemoteFindFirstPrinterChangeNotification`, указывая foosrv на подключение к серверу udbarsrv (домен bar.com) с помощью SMB;
- foosrv запрашивает у KDC `foo.com` ST для службы SMB udbarsrv (cifs/udbarsrv);
- KDC проверяет, находится ли запрошенная служба `foo.com` в доверенном домене `bar.com` и выдает TGT для foosrv$ для этого домена. Этот TGT включает флаг OK-AS-DELEGATE, который указывает, что делегирование TGT включено для `bar.com` от `foo.com`;
- Затем foosrv использует новый TGT, чтобы запросить у KDC `bar.com` ST для cifs/udbarsrv;

- KDC `bar.com` возвращает ST для foosrv$ для cifs/udbarsrv. Для этого ST установлен флаг OK-AS-DELEGATE, указывающий, что службы используют неограниченное делегирование;
- Таким образом, foosrv проверяет, что cifs/udbarsrv использует делегирование и разрешено делегирование `bar.com`, поэтому запрашивает у KDC `foo.com` перенаправленный TGT;
- KDC `foo.com` возвращает TGT для пользователя foosrv$ на сервер foosrv;
- Наконец, foosrv подключается к udbarsrv и выполняет аутентификацию, включая собственный TGT. Теперь злоумышленник на этой машине может вспомнить TGT и использовать его для доступа к foosrv.

В этом примере barsrv и udbarsrv являются разными серверами, чтобы показать, что они могут быть разными машинами, но ошибка принтера также может использоваться для указания на повторное подключение к той же машине, которая выполняет вызов RPC. Кроме того, KDC также могут быть серверами, которые выполняют или получают вызов ошибки принтера. В этом примере использовалось много разных машин для представления различных сообщений и ролей Kerberos в атаке.

В связи с этим важно знать, что в контроллерах домена (KDC) разрешено неограниченное делегирование, поэтому компрометация доменного контроллера домена может привести к компрометации других лесов с двунаправленными доверительными отношениями, в которых включено делегирование TGT.

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.