Kerberoasting v2

habr.com/ru/articles/650889

CyberLympha February 10, 2022

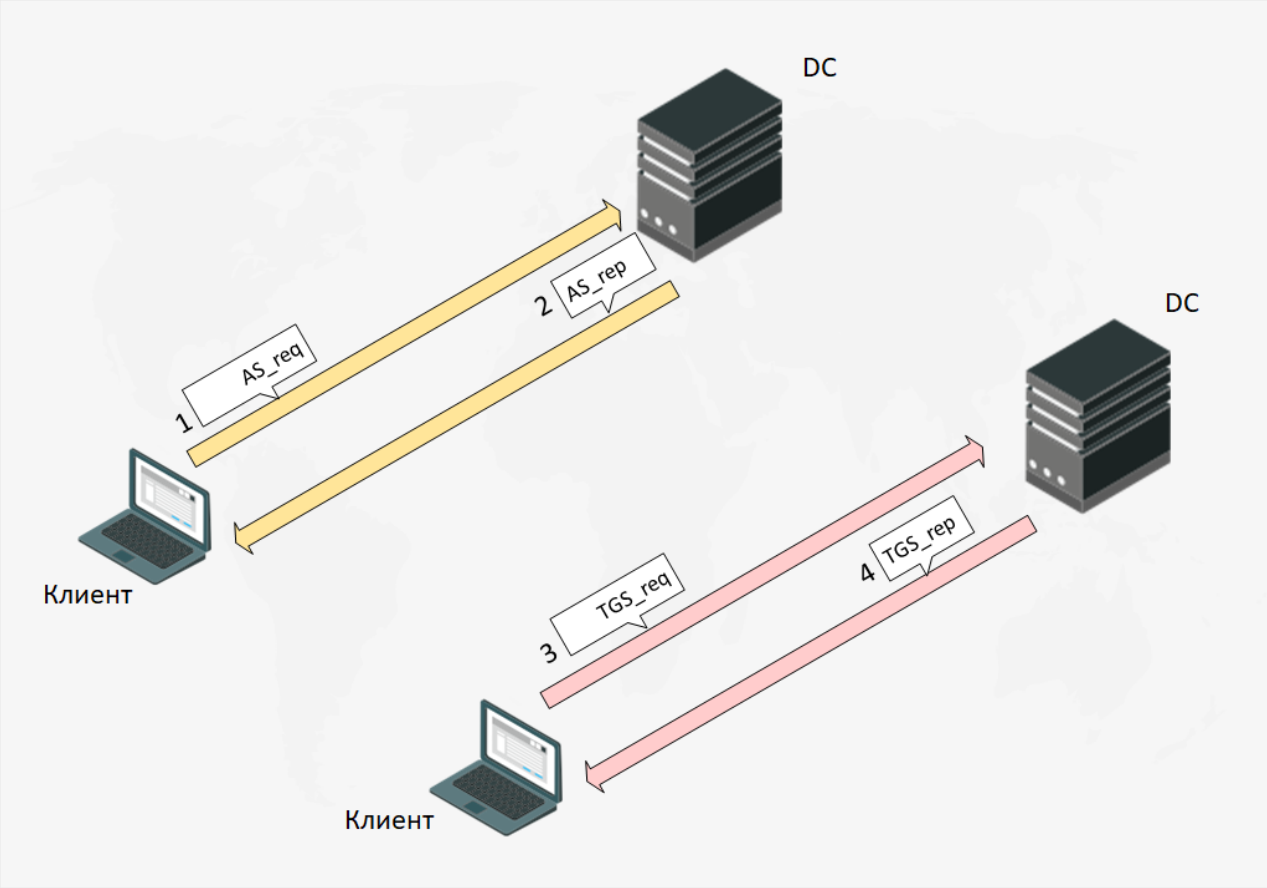


Рис.1. Схема потоков сообщений между Клиентом и DC

8 мин

33K

Название статьи

В статье «Итоги внутренних пентестов — 2020» от Positive Technologies сообщается, что в 61% внутренних тестирований на проникновение успешно применялась атака Kerberoasting. Это мотивировало меня разобраться в атаке, а также ответить на следующие вопросы: почему Kerberoasting так часто эксплуатируется и какие практики по защите существуют и успешно применяются на текущий момент.

Перед тем как перейти к сути атаки, полезно получить общее представление о том, как именно устроена проверка подлинности Kerberos.

Проверка подлинности

В проверке подлинности Microsoft по Kerberos участвуют:

- Key Distribution Center (KDC) – Центр распространения ключей Kerberos, одна из служб безопасности Windows server. Работает на контроллере домена (DC);
- Клиент, который хочет пройти проверку подлинности и получить доступ к сервису;
- Сервер, к сервису которого пользователь хочет получить доступ.

Схему общения между Клиентом и DC можно представить в виде потока сообщений:

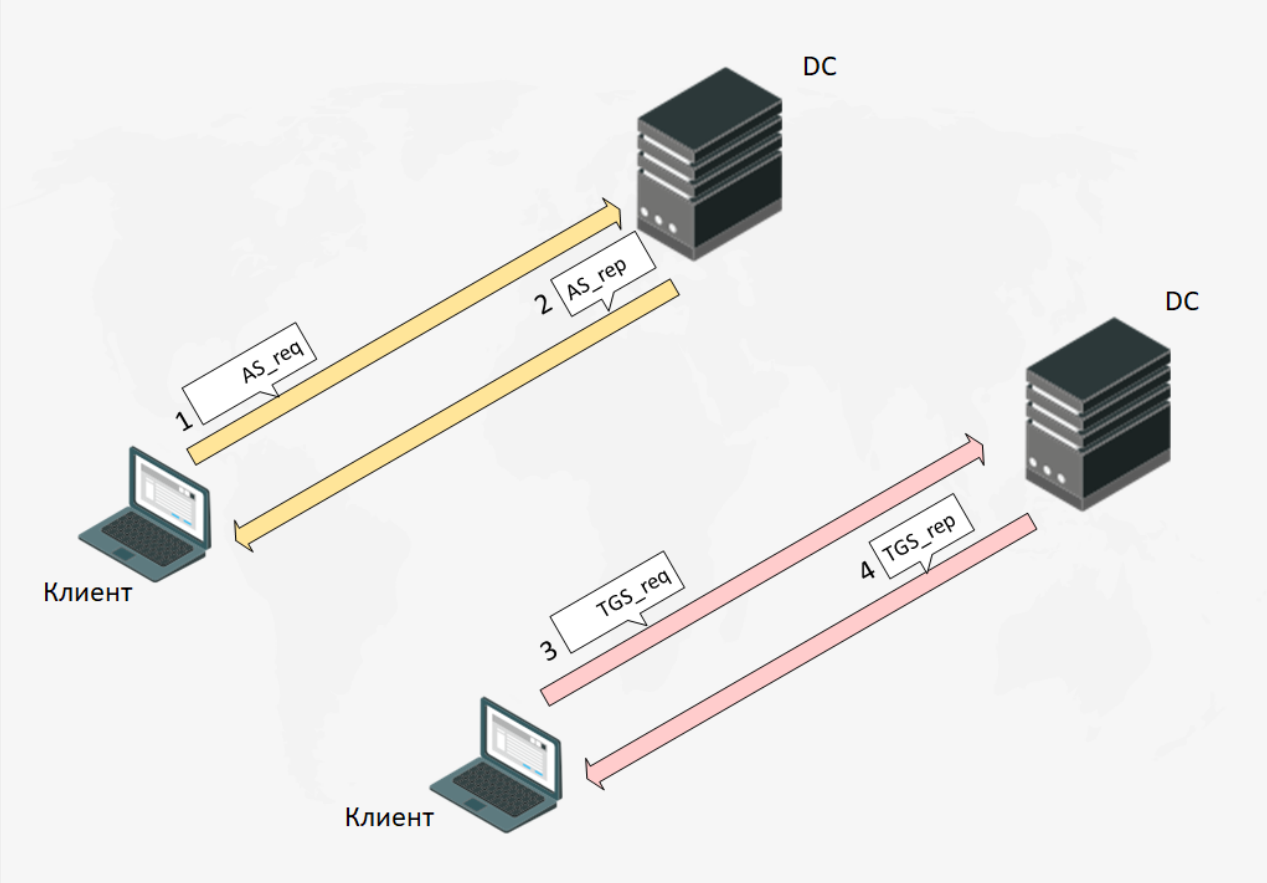


Рис.1. Схема потоков сообщений между Клиентом и DC

AS_req

Когда Клиент приступает к проверке подлинности, на DC отправляется сообщение AS_req (Authentication Service Request). AS_req-сообщение включает в себя UPN (UserPrincipalName), имя службы, к которой идет обращение (всегда krbtgt), а также штамп времени, зашифрованный с использованием хеша пароля учетной записи пользователя.

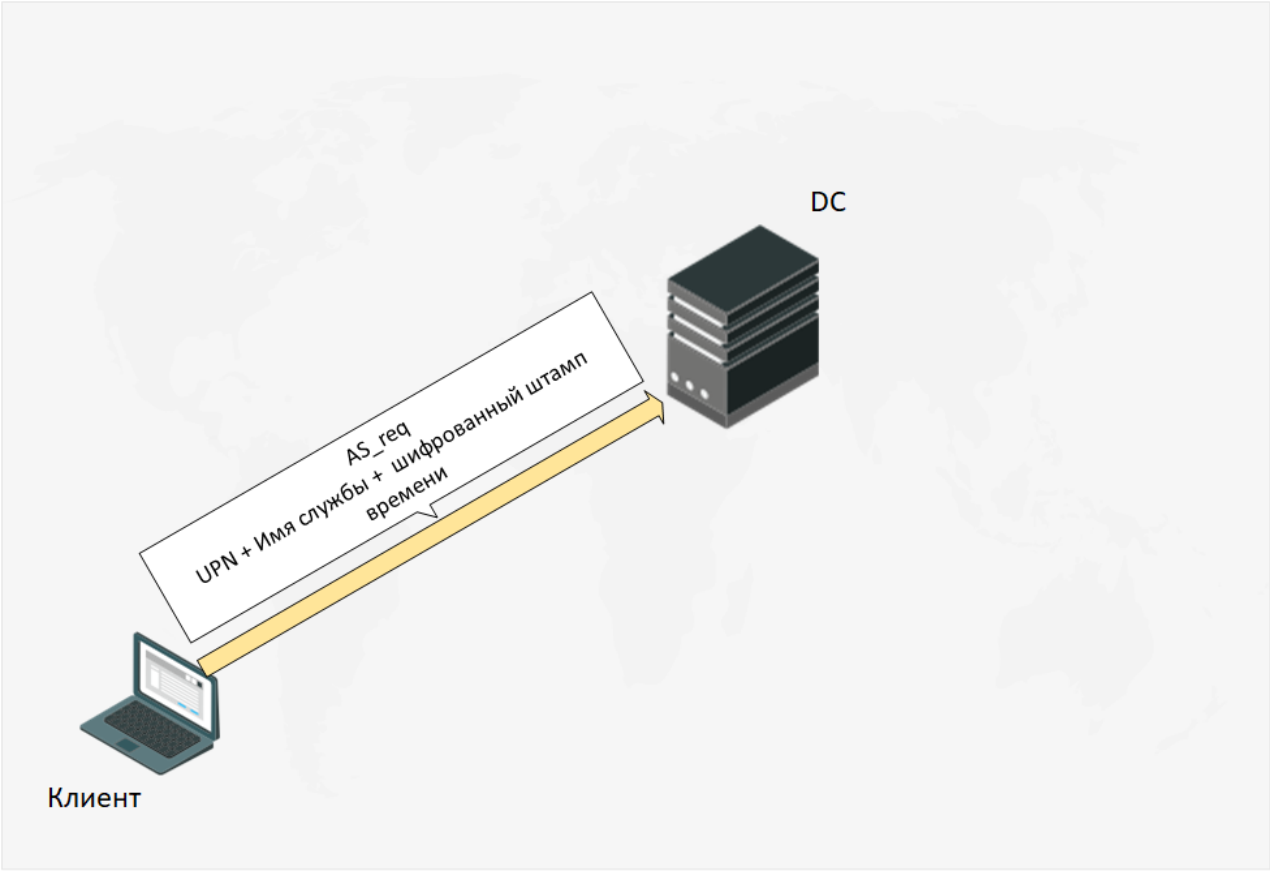


Рис.2. Клиент начинает предварительную проверку подлинности

На последней особенности основывается атака ASreqroasting. В случае MITM-атаки атакующий может перехватить AS_req-сообщение, чтобы затем извлечь из него зашифрованный штамп времени и пассивно перебрать с помощью hashcat (режим 7500 в случае, если штамп времени зашифрован RC4, или режим 19900, если штамп времени зашифрован AES256).Подробнее об эксплуатации можно прочитать по ссылке. Также стоит отметить, что атака ASreqroasting имеет низкую популярность, в сравнении с другими roast-атаками на kerberos, хотя и известна с 2014 года.

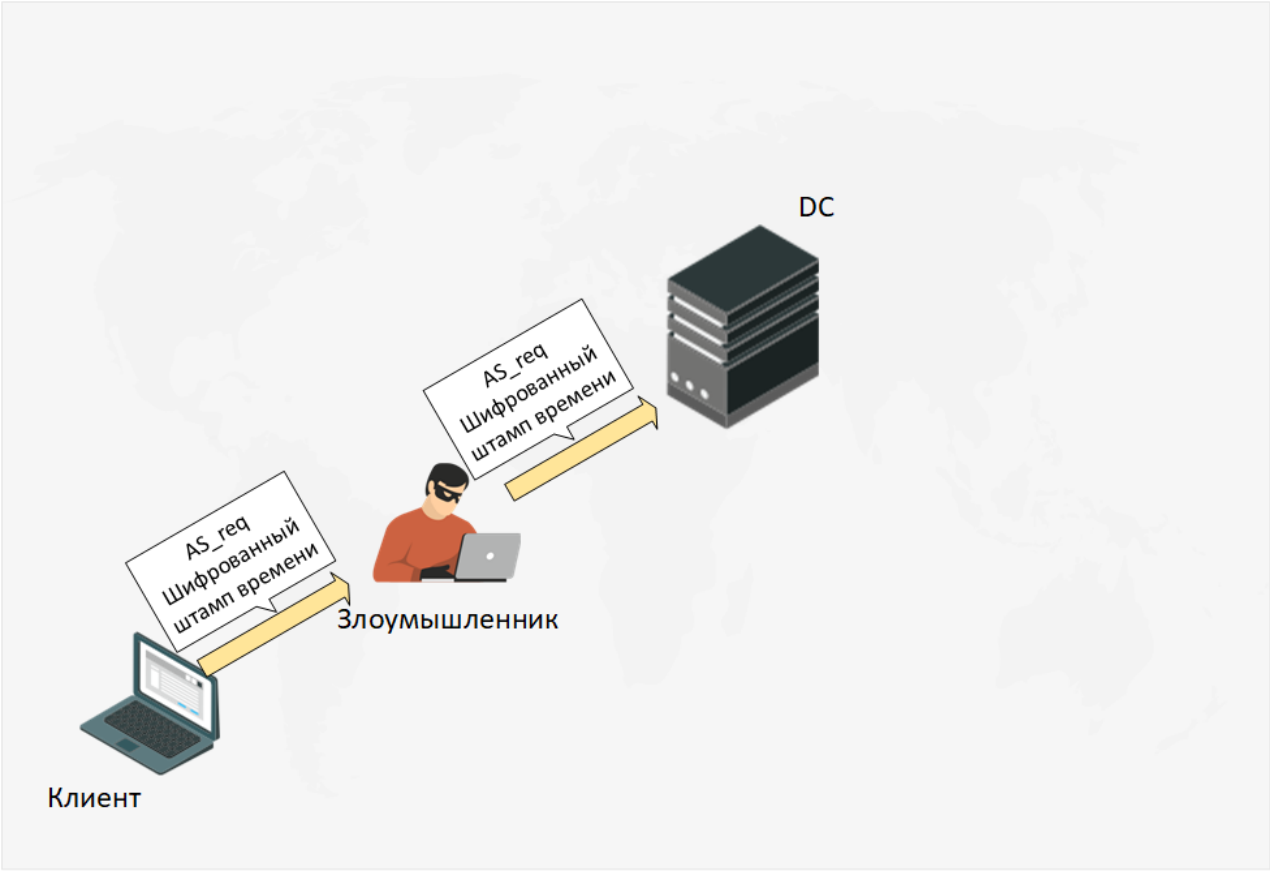


Рис.3. Атакующий реализует MITM и перехватывает AS_req-сообщение

AS_rep

Когда DC получает AS_req-сообщение, то в первую очередь расшифровывает штамп времени при помощи хеша пароля пользователя. Если зашифрованный штамп времени отличается от текущего в момент расшифровки более, чем на 5 минут (значение параметра Time Skew по умолчанию), то будет отправлен ответ PreAuth failed. В случае, когда штамп времени удается расшифровать, DC отправляет в ответ сообщение AS_rep (Authentication Service Reply). AS_rep-сообщение содержит в себе билет TGT (Ticket Granting Ticket), зашифрованный с использованием хеша пароля учетной записи krbtgt, и сеансовый ключ, зашифрованный с использованием хеша пароля учетной записи пользователя. Билет TGT также содержит в себе этот же самый сеансовый ключ. Сеансовый ключ необходим для шифрования следующего сообщения Клиента к DC.

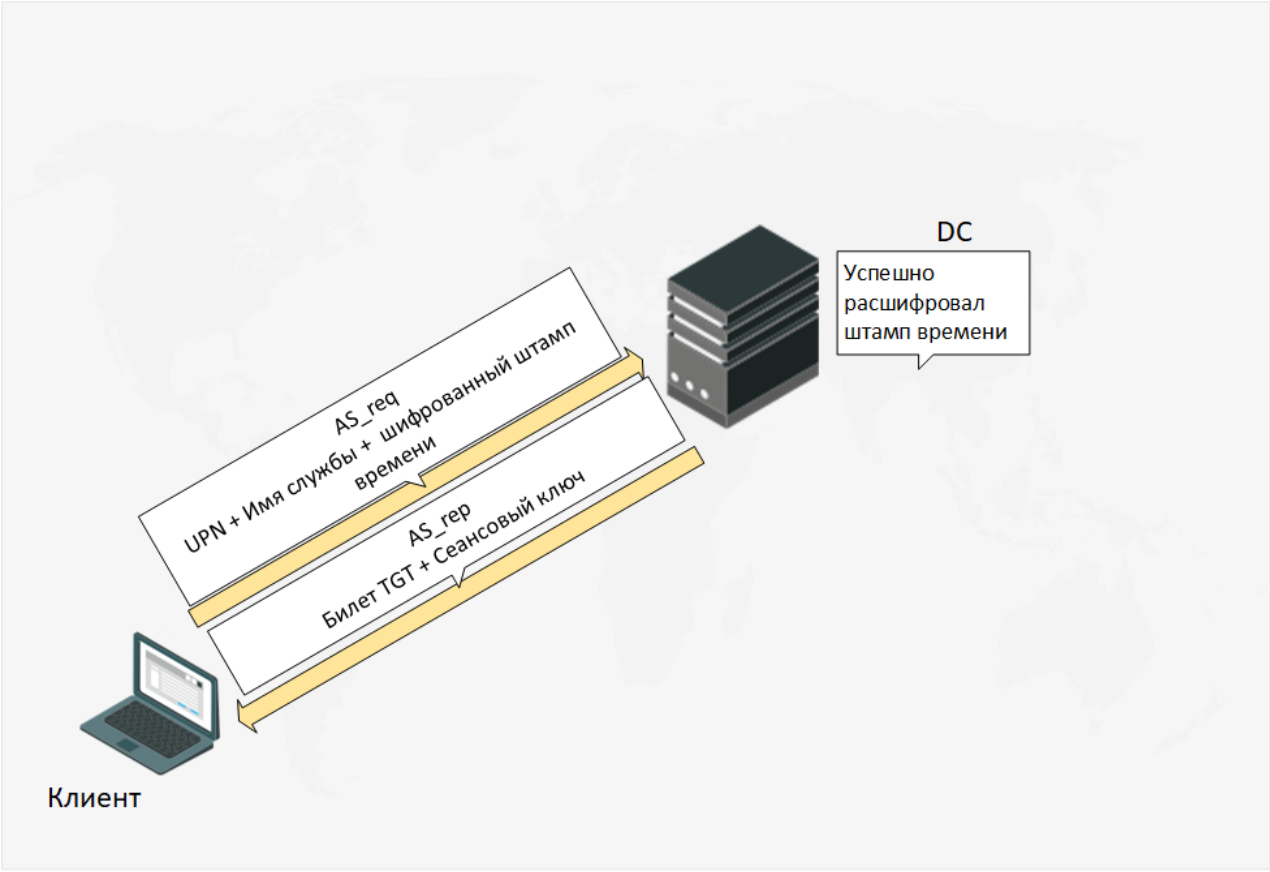


Рис.4. DC отправляет AS_rep-сообщение в ответ

И тут тоже не обошлось без своей roast-атаки. Ее суть в том, что подписание штампа времени в сообщении AS_req можно отключить для учетной записи пользователя (отключение предварительной проверки подлинности Kerberos). Это означает, что атакующий может перечислить учетные записи, для которых предварительная проверка подлинности отключена, и от их имени отправить AS_req-сообщение к DC и получить в ответ AS_rep-сообщение, которое, как мы уже знаем, содержит сеансовый ключ, зашифрованный с использованием хеша пароля учетной записи пользователя. Называется эта атака ASREProasting. Хотя учетные записи с отключенной предварительной проверкой подлинности являются редкостью, у этой атаки есть несколько преимуществ:

- можно применять, даже не имея доменной учетной записи (достаточно сетевого доступа к DC), правда тогда перечисление пользовательских учетных записей с отключенной предварительной проверкой подлинности может стать проблемой;
- полученный хеш, как и в случае с ASreqroast, можно перебирать пассивно (hashcat режим 18200).

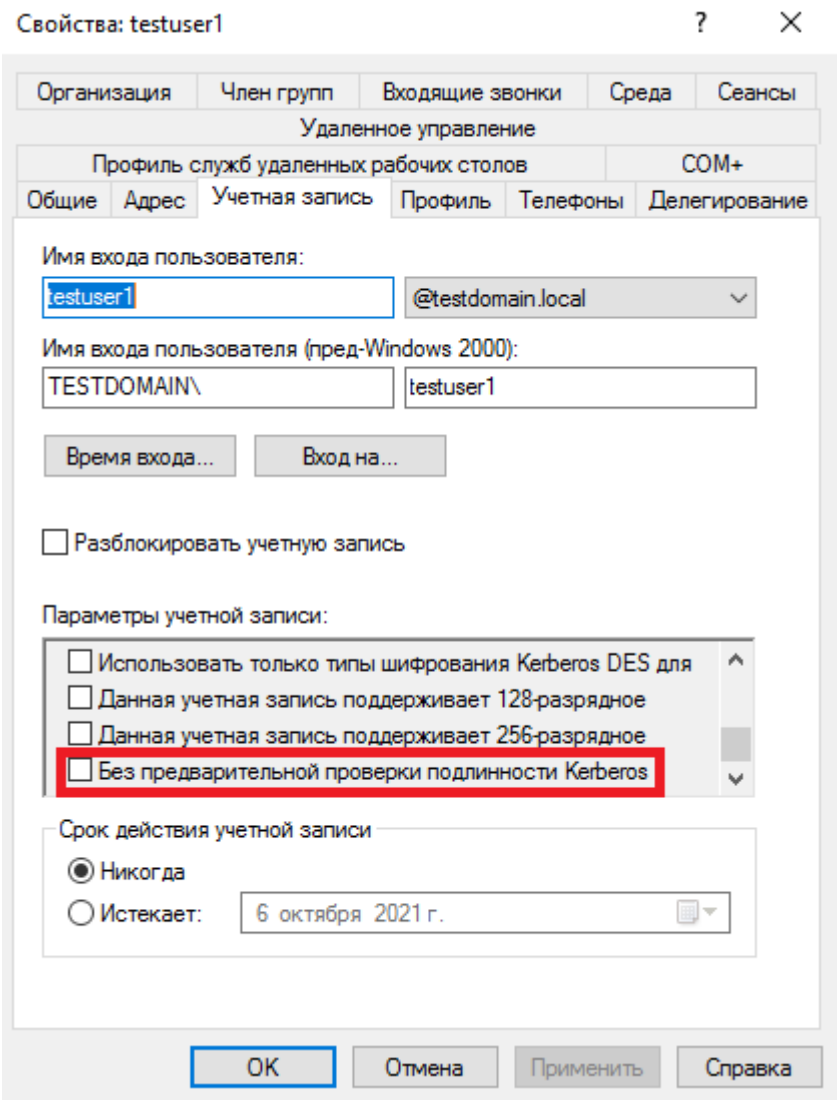


Рис.5. Настройка отключения предварительной проверки подлинности

TGS_req

Предварительная проверка подлинности пройдена. Теперь Клиент отправляет на DC сообщение TGS_req (Ticket Granting Service Request), содержащее в себе:

- SPN (Service Principal Name) – имя службы, для которой Клиент запрашивает доступ. Ассоциируется либо с компьютерной учетной записью, либо с пользовательской;
- UPN и штамп времени, зашифрованные с использованием сессионного ключа, полученного ранее;
- билет TGT.

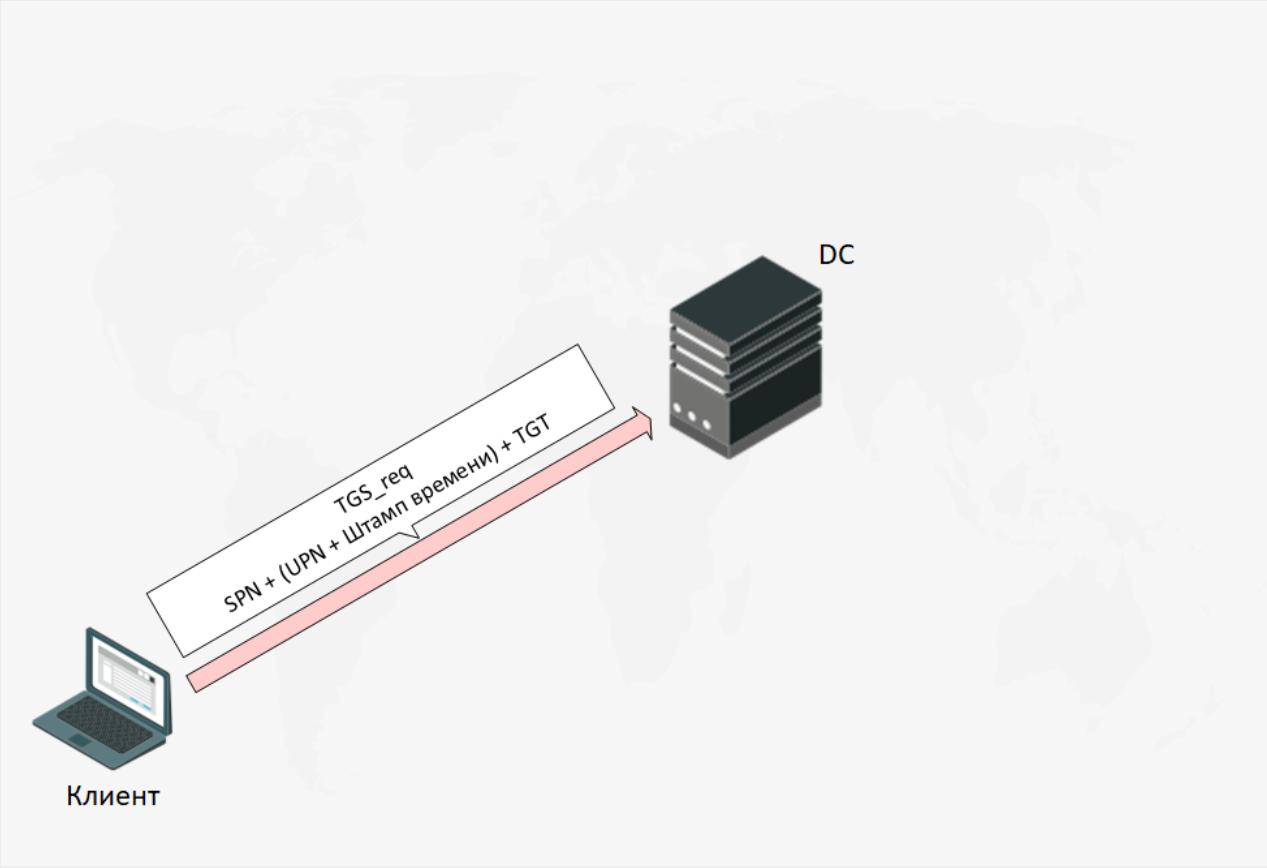


Рис.6. Клиент отправляет TGS_req-сообщение

TGS_rep

После получения TGS_req-сообщения DC проверяет SPN и срок действия билета TGT (по умолчанию билет TGT действителен 10 часов), расшифровывает и анализирует штамп времени. Если SPN указан верно, срок действия билета TGT не истек, и штамп времени находится в допустимом диапазоне, то DC отправляет Клиенту сообщение TGS_rep (Ticket Granting Service Reply), содержащее билет TGS (Ticket Granting Service), зашифрованный с использованием хеша пароля учетной записи, от имени которой запущен сервис. И именно последний факт делает возможным атаку Kerberoasting.

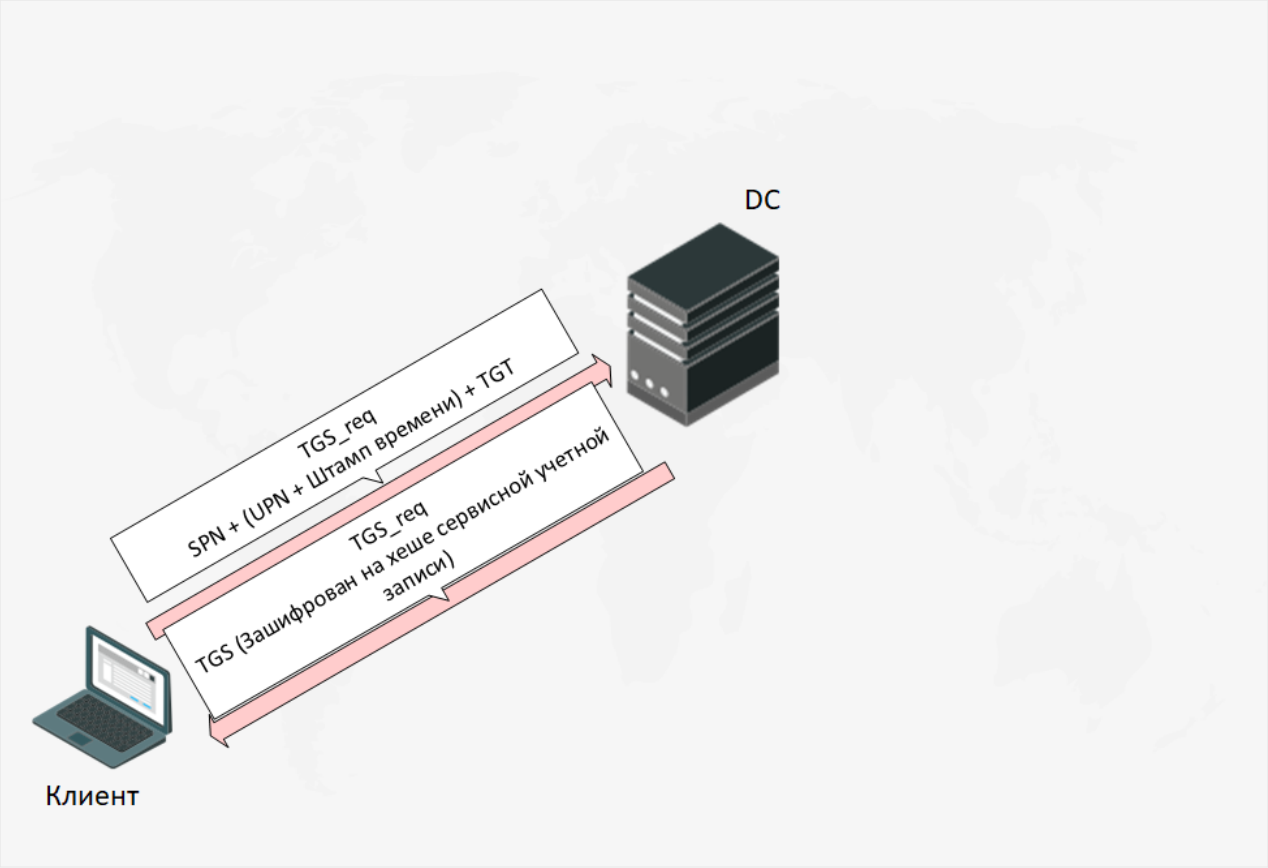


Рис.7. Клиент получает билет TGS

Далее в рамках проверки подлинности Kerberos идут сообщения AP_req и AP_rep, но их описывать мы не станем, так как рассмотренных сообщений достаточно для понимания атаки.

Kerberoasting

Kerberoasting является возможным по двум причинам. Во-первых, DC не занимается авторизацией Клиента, то есть имеет ли право Клиент посещать те или иные сервисы — это не вопрос DC. И поэтому атакующий, имея одну лишь доменную учетную запись, может создать легитимный запрос билета TGS ко всем SPN в домене. Стоит заметить, что атакующего интересуют SPN, ассоциирующиеся с учетными записями пользователей, а не компьютеров, ввиду нецелесообразности восстановления паролей последних. Вторая причина уже озвучивалась выше: билеты TGS шифруются с использованием хеша сервисной учетной записи. Это позволяет атакующему восстановить пароль сервисной учетной записи при условии, что пароль недостаточно стойкий.

Атаку можно разделить на несколько этапов:

1. Атакующий приступает к аутентификации в домене (AS_req и AS_rep).
2. Атакующий использует билет TGT для запроса получения билета TGS для конкретного SPN (TGS_req и TGS_rep).
3. Атакующий извлекает хеш зашифрованного билета TGS из TGS_rep.

Несколько общих фактов об атаке:

- легкость в эксплуатации. В сети можно найти огромное количество инструкций, как выполнять перечисление SPN запрос билетов TGS, получение хешей зашифрованных билетов TGS, а также как реализовать все действия разом, к примеру, с помощью модуля Impacket GetUserSPNs.py;
- для реализации атакующему достаточно иметь доменную учетную запись с любым уровнем привилегий и сетевой доступ к DC по порту UDP/88;
- полученные хеши сервисных учетных записей можно в пассивном режиме подвергнуть перебору.

Атака прочно вошла в арсенал как пентестеров, так и хакеров. Обиженный участник «партнерской программы» вымогателя Conti опубликовал обучающие «гайды» хак-группы, в которых Kerberoasting фигурирует как одна из приоритетных атак. Конкретно в «гайде» написано, что в случае проникновения в крупный домен, именно Kerberoasting целесообразно применять в первую очередь.

Пример реализации атаки с помощью GetUserSPNs.py:

```
ntpdate 10.23.53.26; GetUserSPNs.py -request -dc-ip 10.23.53.26
TESTDOMAIN.local/testuser1:Testuser! > kerberoast.txt
```

где:

- ntpdate – синхронизация времени на машине атакующего с временем на DC. Без данной команды рискуем получить ошибку «KRB_AP_ERR_SKEW(Clock skew too great)»;
- 10.23.53.26 – IP-адрес DC;
- GetUserSPNs.py -request -dc-ip – запуск скрипта с опциями -request и -dc-ip;
- TESTDOMAIN.local – имя домена;
- testuser1:Testuser! – логин и пароль доменной учетной записи;
- > kerberoast.txt – записываем вывод скрипта GetUserSPNs.py в текстовый файл.

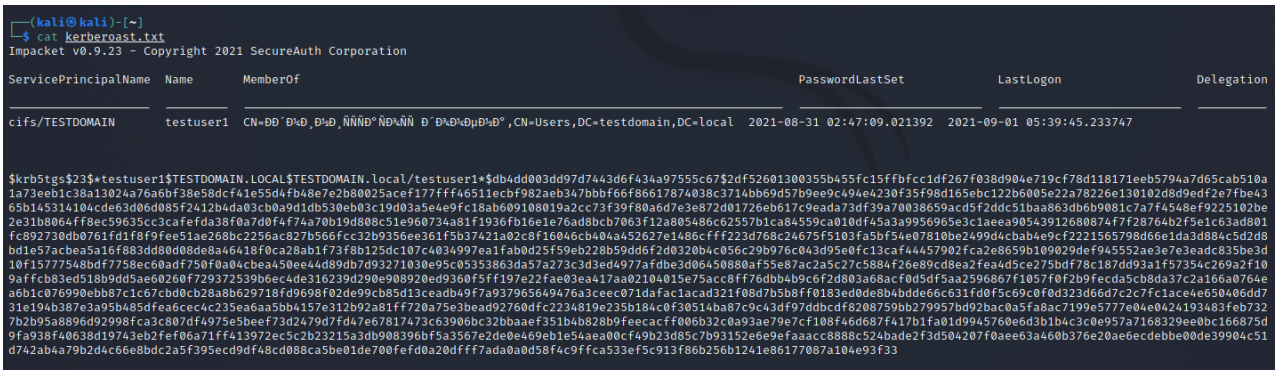


Рис.8. Вывод файла kerberoast.txt

Далее хеш подвергается перебору и, если пароль недостаточно стойкий, атакующий получает пароль сервисной учетной записи.

Противодействие

1. Парольная политика и уменьшение привилегий сервисных учетных записей

Самое просто противодействие – установить для всех сервисных учетных записей пароли длиной 25-30 символов, а также убедиться, что сервисные учетные записи имеют только необходимые привилегии и, к примеру, не состоят в группе администраторов домена. Звучит это все достаточно просто, но, к сожалению, эффективность Kerberoasting показывает, что в подавляющей части доменов данные рекомендации не выполняются.

2. SPN honeypot

Существует эффективный способ обнаружения Kerberoasting, заключающийся в создании учетной записи и SPN, которые не будут использоваться (созданный SPN не связан ни с одним реальным приложением). Клиенты Kerberos никогда не запросят билет TGS для ложного SPN, поэтому, если в журнале безопасности DC появляется соответствующее событие 4769, то использование Kerberoasting может быть замечено.

Пример события 4769 (запрос билета TGS), сигнализирующего о проведении Kerberoasting:

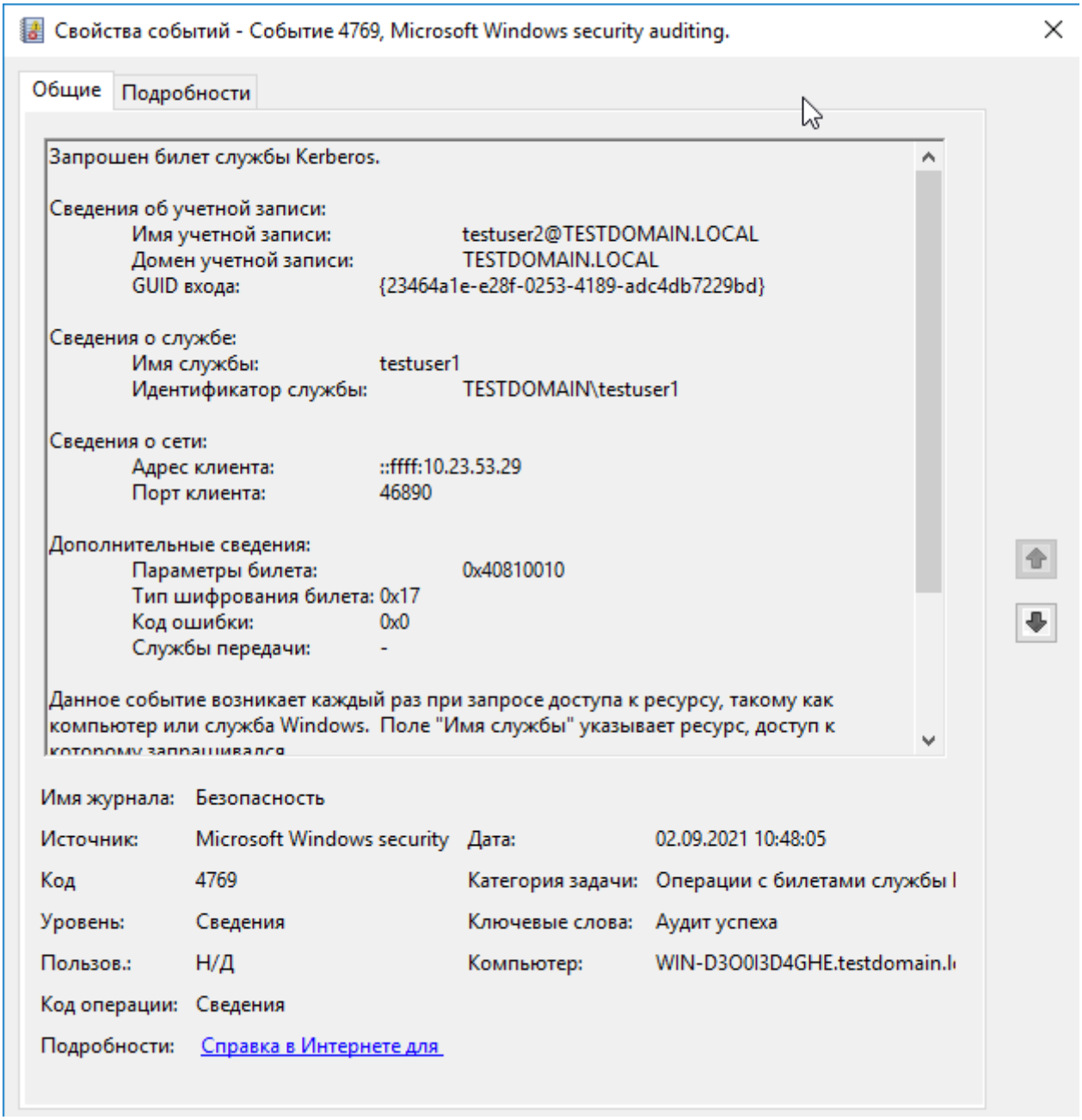


Рис.9. Событие 4769, указывающее на запрос билета TGS

В данном событии:

- Testuser2@TESTDOMAIN.LOCAL – скомпрометированная учетная запись, которую атакующий использует для запроса билета TGS;
- Testuser1 – учетная запись, являющаяся ловушкой. С этой учетной записью ассоциируется ложный SPN;
- 10.23.53.29 – ip-адрес, с которого проводится атака.

Таким образом, у защитников есть возможность обнаружить не только факт атаки, но и то, с какой машины она ведется.

3. FAST (или Kerberos armoring)

Flexible Authentication Secure Tunneling (FAST) или Kerberos Armoring – настройка безопасности DC, обеспечивающая защищенный канал между клиентом Kerberos и KDC в рамках сообщений AS_req, AS_rep, TGS_req и TGS_rep. Поддерживается, начиная с Windows Server 2012 и Windows 8. Подробно механизм работы описан в RFC 6113 и RFC 4851.

Возможная реализация использования Kerberos Armoring следующая:

1. Активируем поддержу Kerberos Armoring на DC. Для этого заходим в **Group Policy Management**, переходим к **политике контроллера домена по умолчанию**, открываем контекстное меню правой кнопкой мыши, выбираем **изменить**. Далее выбираем **Конфигурация компьютера -> Политики -> Административные шаблоны -> Система -> Центр распространения ключей**. В правой области открываем **Поддержка KDC требований, комплексной проверки подлинности и защиты Kerberos** и устанавливаем состояние на **Включено**, а в параметрах выбираем **Отклонять запросы проверки подлинности без защиты**.

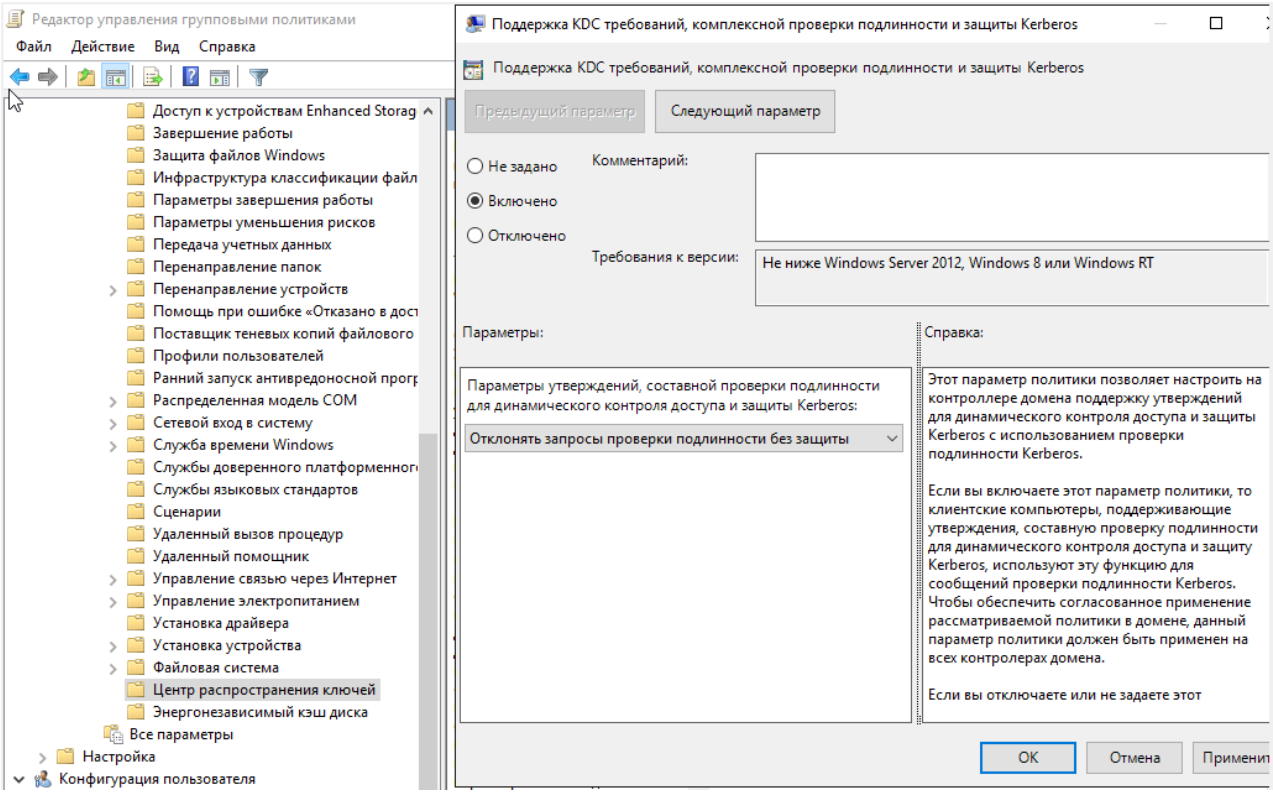


Рис.10. Включение поддержки Kerberos Armoring на DC

2. Активируем поддержку Kerberos Armoring на Клиенте Kerberos. Для этого заходим в **Group Policy Management**, переходим к **политике контроллера домена по умолчанию**, открываем контекстное меню правой кнопкой мыши, выбираем **изменить**. Далее выбираем **Конфигурация компьютера -> Политики -> Административные шаблоны -> Система -> Kerberos.** В правой области открываем **Поддержка клиентами Kerberos требований, комплексной проверки подлинности и защиты Kerberos** и устанавливаем состояние на **Включено**.

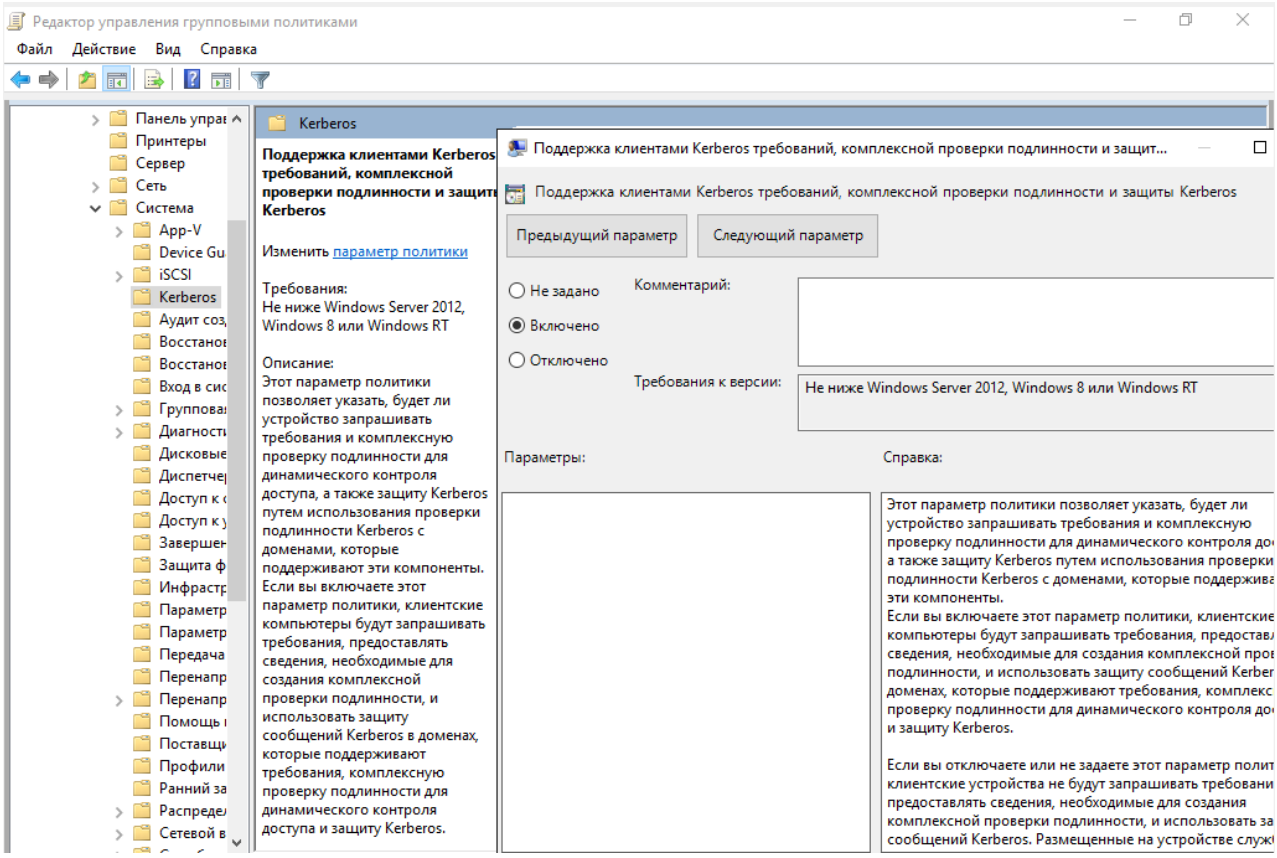
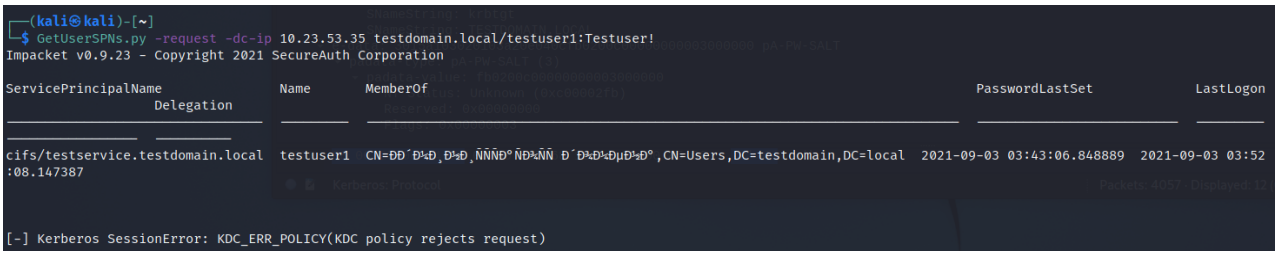
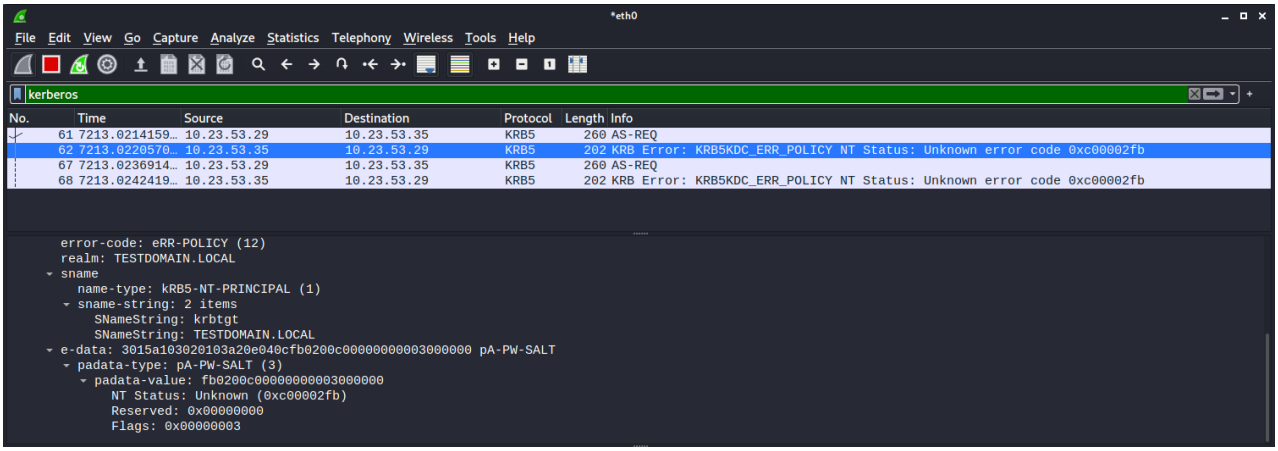


Рис.11. Включение поддержки Kerberos Armoring на Клиенте Kerberos

Затем запускаем на машине атакующего GetUserSPNs.py и получаем ошибку KDC_ERR_POLICY(KDC policy rejects request):



Перехватив сообщения Kerberos через Wireshark, видим ошибку NT STATUS: Unknown error code 0xc00002fb:



С помощью https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-erref/596a1078-e883-4972-9bbc-49e60bebca55 узнаем значение ошибки 0xc00002fb: An invalid request was sent to the KDC.

Таким образом, DC отказывает атакующему в предварительной проверке подлинности, так как ожидает построения защищенного канала для передачи сообщений AS_req и AS_rep.

4. gMSA

Групповые управляемые учетные записи служб или gMSA (Group Managed Service Accounts) – тип учетных записей в AD для безопасного запуска служб. Для gMSA генерируется пароль длиной 240 символов, который по умолчанию меняется каждые 30 дней. Пароль управляется AD и не хранится локально в системе, поэтому его нельзя извлечь из дампа процесса LSASS. Для аутентификации gMSA использует только Kerberos. Поддерживается, начиная с Windows server 2012. Подробнее можно прочитать по ссылке.

Возможная реализация использования gMSA следующая:

1. Создается доменная группа для серверов, которым будет разрешено использование групповой сервисной учетной записи:

```
New-ADGroup testgMSA -GroupScope Global -PassThru -Verbose
```

Где testgMSA – имя создаваемой доменной группы

```
PS C:\Windows\system32> New-ADGroup testgMSA -GroupScope Global -PassThru -Verbose
ПОДРОБНО: Выполнение операции "New" над целевым объектом "CN=testgMSA,CN=Users,DC=testdomain,DC=local".


DistinguishedName : CN=testgMSA,CN=Users,DC=testdomain,DC=local
GroupCategory     : Security
GroupScope        : Global
Name              : testgMSA
ObjectClass       : group
ObjectGUID        : 80ceebde-607d-4617-99a0-b4be9ef532f7
SamAccountName    : testgMSA
SID               : S-1-5-21-246992039-600058196-3259959164-1106
```

Сервер WIN-D300I3D4GHE добавляется в группу testgMSA:

```
Add-AdGroupMember -Identity testgMSA -Members WIN-D300I3D4GHE$
```

```
PS C:\Windows\system32> Add-AdGroupMember -Identity testgMSA  -Members WIN-D3OOI3D4GHE$
PS C:\Windows\system32>
PS C:\Windows\system32>
PS C:\Windows\system32>
```

2. Создается групповая учетная запись gMSA:

```
New-ADServiceAccount -name gmsa -DNSHostname gmsa.testdomain.local -
PrincipalsAllowedToRetrieveManagedPassword testgMSA -Verbose
```

Где gmsa – создаваемая групповая управляемая учетная запись.

```
PS C:\Windows\system32> New-ADServiceAccount -name gmsa -DNSHostName gmsa.testdomain.local -PrincipalsAllowedToRetrieveM
anagedPassword testgMSA -Verbose
ПОДРОБНО: Выполнение операции "New" над целевым объектом "CN=gmsa,CN=Managed Service Accounts,DC=testdomain,DC=local".
PS C:\Windows\system32>
PS C:\Windows\system32>
PS C:\Windows\system32>
```

3. Выполняется настройка возможности использования учетной записи gMSA на сервере WIN-D300I3D4GHE:

```
Install-ADServiceAccount gmsa
```

Установка возможности использования gmsa на сервере

```
Test-ADServiceAccount gmsa
```

Тест возможности использования gmsa на сервере:

```
PS C:\Users\Администратор.WIN-D3OOI3D4GHE> Install-ADServiceAccount gmsa
PS C:\Users\Администратор.WIN-D3OOI3D4GHE> Test-ADServiceAccount gmsa
True
PS C:\Users\Администратор.WIN-D3OOI3D4GHE>
```

4. Последним этапом проводится настройка запуска сервиса от имени gMSA.

Таким образом, получаем возможность запускать сервисы от имени учетной записи gMSA, пароль которой, в случае получения атакующим зашифрованного хеша билета TGS, восстановить не получится.

Описанная в статье технология имеет практическое применение в CL DATAPK, флагманском продукте компании СайберЛимфа для защиты систем автоматизации. Больше информации о компании и разработках СайберЛимфа доступно на сайте компании.