Детектирование аномалий при запросе TGT-билета по сертификату

securelist.ru/anomaly-detection-in-certificate-based-tgt-requests/107710

Alexander Rodchenko July 28, 2023

Авторы

Александр Родченко

Один из наиболее сложных, но в то же время эффективных методов получения неавторизованного доступа к ресурсам внутри корпоративной сети — это атака с использованием поддельных сертификатов. Суть ее заключается в создании злоумышленником таких сертификатов, которые будут приняты Центром распространения ключей (Key Distribution Center, KDC) и обеспечат доступ к ресурсам внутри сети атакуемой компании. Примером такой атаки является техника Shadow Credentials, которая позволяет авторизоваться под учетной записью пользователя, если атакующий может изменить атрибут msDS-KeyCredentialLink жертвы и добавить к нему сертификат для авторизации. Такие атаки трудно обнаружить, поскольку вместо кражи учетных данных злоумышленники работают с легитимными механизмами Active directory (AD) и недостатками конфигурации.

Тем не менее противостоять атакам с использованием поддельных сертификатов можно (и нужно). Я проанализировал практический опыт нашего сервиса MDR, после чего выявил несколько признаков присутствия таких атак внутри сети и разработал Proof-of-Concept утилиты, способной находить артефакты в AD, а также несколько правил детектирующей логики, которые можно добавить в SIEM. Но сначала несколько слов о том, в чем кроется неочевидный нюанс с аутентификацией Kerberos с использованием сертификатов.

Аутентификация Kerberos в AD и нюансы ее реализации

В современных корпоративных сетях, основанных на Active Directory, управление ресурсами происходит благодаря протоколу Kerberos. Пользователь может получить доступ к какому-либо сервису (объекту) внутри сети, только если он может предоставить этому объекту

билет, выданный Центром распространения ключей (Key Distribution Center, KDC) («Msg E» на рисунке ниже). Отвечающий за выдачу сервисных билетов компонент KDC называется «Сервер выдачи билетов» (Ticket Granting Server, TGS). При этом пользователь может получить TGS-билет от KDC, только если у него есть «билет для получения билетов» (Ticket Granting Ticket, TGT) («Msg B» на рисунке ниже). По сути, TGT-билет является доказательством успешного прохождения пользователем аутентификации, как правило, по паролю.

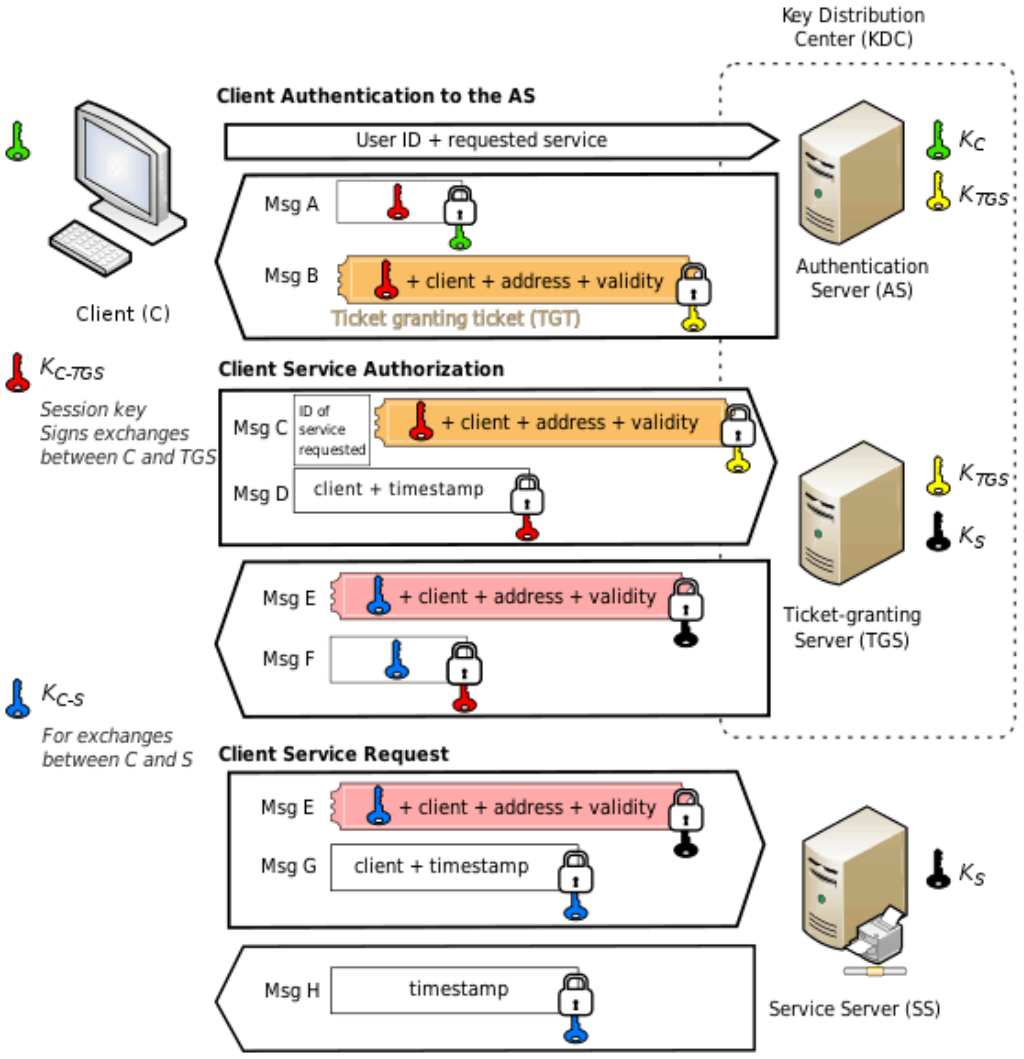
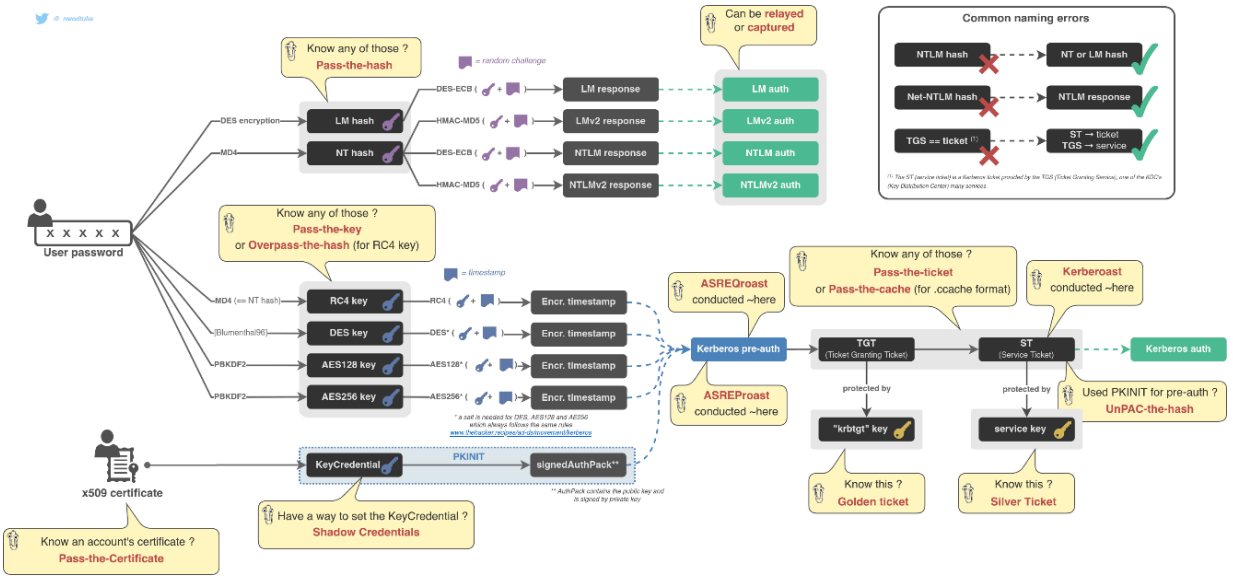


Схема аутентификации по протоколу Kerberos. Источник: https://en.wikipedia.org/wiki/Kerberos_(protocol)

Однако существует способ получить TGT-билет без знания пароля — по сертификату. Для этого нужно, чтобы KDC доверял предоставленному сертификату, а сертификат относился к субъекту, который запрашивает TGT-билет. Эта часть протокола Kerberos называется Public Key Cryptography for Initial Authentication (PKINIT), и с ее помощью достаточно просто организовать аутентификацию, если в корпоративной сети существует удостоверяющий центр (Certificate Authority), который выдает сертификаты для пользователей домена. Однако есть и альтернативный способ.



Источник: https://github.com/ShutdownRepo/The-Hacker-Recipes/blob/master/.gitbook/assets/Pass%20the%20things.png

Например, для того чтобы воспользоваться возможностями Microsoft Hello for Business, такими как авторизация с использованием PIN-кода или распознавания лица, необходимо, чтобы устройство, с которого осуществляется вход, имело в AD свой сертификат и KDC мог выписать TGT-билет по этому сертификату. Однако далеко не во всех сетях с Active Directory есть удостоверяющий центр. Поэтому для объектов AD был придуман атрибут msDS-KeyCredentialLink, в который можно записать сертификат. При выдаче TGT-билета служба KDC будет доверять этому сертификату. И это действительно хорошее решение, расширяющее возможности Microsoft Active Directory.

Однако исходя из описанной выше логики, тот субъект, который может контролировать запись атрибута msDS-KeyCredentialLinky какого-либо объекта, сможет и получить билет за этот объект. В этом и заключается проблема.

Как происходит атака

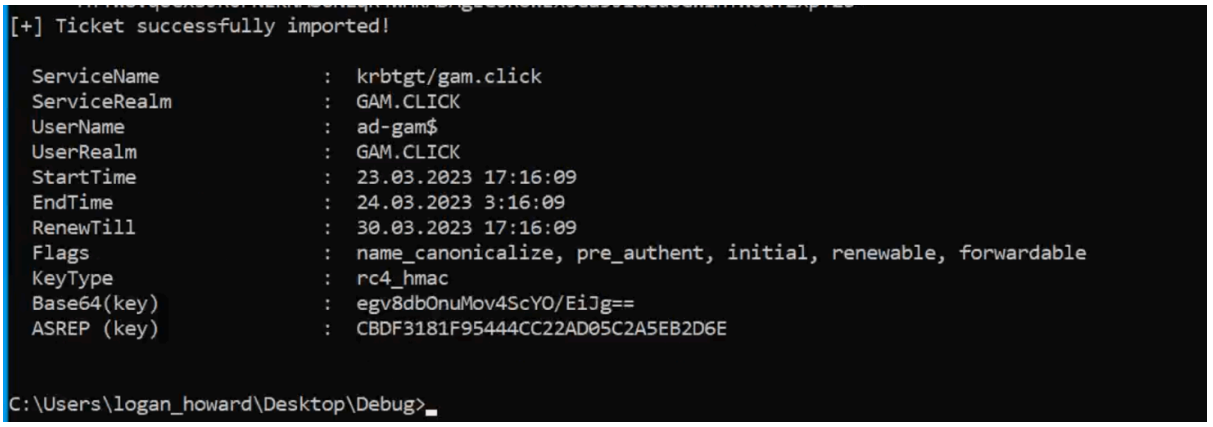
Проиллюстрируем один из возможных сценариев атаки:

1. Субъект logan_howard, обладая правами на запись любого атрибута в домене Active Directory, записывает публичный ключ в атрибут msDS-KeyCredentialLink для объекта контроллера домена (ad-gam$). Используется инструментарий Whisker.

```
C:\Users\logan_howard\Desktop\Debug>Whisker.exe list /target:ad-gam$
[*] Searching for the target account
[*] Target user found: CN=AD-GAM,OU=Domain Controllers,DC=gam,DC=click
[*] Listing deviced for ad-gam$:
[*] No entries!

C:\Users\logan_howard\Desktop\Debug>Whisker.exe add /target:ad-gam$
[*] No path was provided. The certificate will be printed as a Base64 blob
[*] No pass was provided. The certificate will be stored with the password 4cnlK4pSNUwYbHSB
[*] Searching for the target account
[*] Target user found: CN=AD-GAM,OU=Domain Controllers,DC=gam,DC=click
[*] Generating certificate
```

2. Субъект получает TGT-билет (используется инструментарий Rubeus), выписанный на контроллер домена.



3. Предъявив этот TGT, субъект получает TGS-билет для синхронизации парольной информации в домене (MS-DRSR: Directory Replication Service (DRS) Remote Protocol).
4. От имени субъекта злоумышленник «синхронизирует» хэш от учетной записи администратора домена (Administrator), чтобы выдать себя за него для получения доступа к данным и горизонтального перемещения внутри корпоративной сети. Эта атака называется DCSync, для нее используется mimikatz.

Где следует искать какие-либо артефакты?

Давайте сфокусируем внимание не на способе, которым можно заставить KDC доверять тому или иному сертификату, в том числе украденному или поддельному, а на том, что происходит в момент выдачи TGT-билета. В этот момент на контроллере домена происходит событие 4768: запрошен билет проверки подлинности Kerberos (TGT). Это событие может содержать артефакты от сертификата, по которому и осуществлялась аутентификация, — три поля: CertIssuerName, CertSerialNumber и CertThumbprint. Собственно, этими полями мы и ограничимся.

Каким инструментарием следует воспользоваться?

Для простоты и удобства мы сразу же будем обрабатывать все события в интерфейсе Kibana кластера ELK. Действительно, по умолчанию Logstash умеет преобразовывать битовые поля события 4768 в массив значений, характерных для билета из списка. Кроме

того, поиск пройдет значительно быстрее и приятнее. Я рекомендую использовать для лаборатории удобный набор Docker-конфигураций для быстрого старта ELK. И официальную инструкцию по настройке WinLogBeat.

Что мы можем сказать об этих событиях?

На тестовом стенде мы сгенерировали несколько событий запроса TGT по поддельному сертификату, который сгенерировали с помощью Whisker. Так эти события выглядят в тестовом окружении:

winlog.event_id	winlog.event_data.TicketOptionsDescription	winlog.event_data.CertIssuerName	winlog.event_data.CertSerialNumber	winlog.event_data.CertThumbprint
4768	[Renewable-ok, Renewable, Forwardable]	ad-gam$	00A10C4274496718DC	9D44A48B216AAB01D3DF1A59315EA13E9C79B45B
4768	[Renewable-ok, Renewable, Forwardable]	ad-gam$	00BA61092B330F639A	2111A043F1FF7C9F1A2EFB73A08BAC0AA672F9CB
4768	[Renewable-ok, Renewable, Forwardable]	ad-gam$	00A1018B1EB5AA9A94	CEA92EE5F7FF5E281EE1D7B20C42F1C0BB85BAEF
4768	[Renewable-ok, Renewable, Forwardable]	ad-gam$	00A1018B1EB5AA9A94	CEA92EE5F7FF5E281EE1D7B20C42F1C0BB85BAEF
4768	[Renewable-ok, Renewable, Forwardable]	ad-gam$	00A1018B1EB5AA9A94	CEA92EE5F7FF5E281EE1D7B20C42F1C0BB85BAEF
4768	[Renewable-ok, Renewable, Forwardable]	ad-gam$	00A1018B1EB5AA9A94	CEA92EE5F7FF5E281EE1D7B20C42F1C0BB85BAEF
4768	[Renewable-ok, Renewable, Forwardable]	ad-gam$	00A1018B1EB5AA9A94	CEA92EE5F7FF5E281EE1D7B20C42F1C0BB85BAEF

В рамках сервиса MDR мы наблюдаем несколько сотен тысяч событий запросов билета по **сертификату** в неделю, это достаточно широкая выборка, и поэтому на основании этих событий мы можем выявить некоторые закономерности:

Значительная часть событий — это запрос билета по сертификату на службу Microsoft Azure Active Directory (строчка «Azure» в агрегации на скриншоте ниже). Данные события нас не интересуют, их легко отфильтровать с помощью регулярного выражения по значению поля CertIssuerName в интерфейсе Kibana.

```
CertIssuerName.raw:/S\-1\-(5\-21|12\-1)\-([0-9]{8,10}\-){3}[0-9]{4,10}\/[a-f0-9]{8}\-([a-f0-9]{4}\-){3}[a-f0-9]{12}\/login\.windows\.net\/[a-f0-9]{8}\-([a-f0-9]{4}\-){3}[a-f0-9]{12}.*/
```

Также много событий для сертификатов, которые использует Microsoft Hello For Business (строчка «Hello4B self gen»). В этом случае данные о сертификате записаны в атрибут msDS-KeyCredentialLink, а ключ программно сгенерирован (NCRYPT_IMPL_SOFTWARE_FLAG). Для них характерны имена, начинающиеся с «CN=», и двузначный серийный номер, как правило, 01.

```
CertIssuerName.raw:CN=* AND CertSerialNumber.raw:01
```

Если компьютер имеет ключ, записанный в Trusted Platform Module (строчка «TPM enrolled»), то сертификат, который использует этот ключ, также может быть описан регулярными выражениями и нас не интересует.

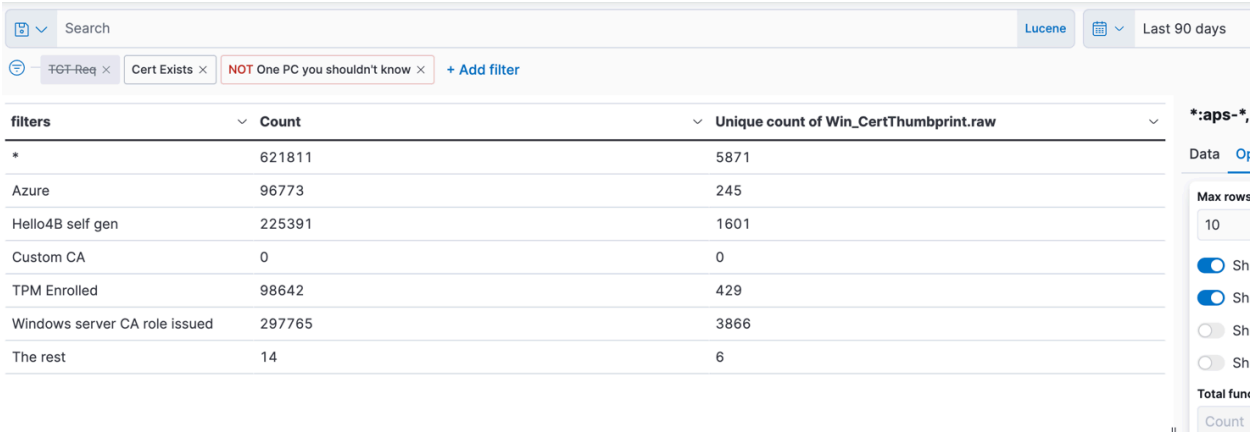
```
CertSerialNumber.raw:/[a-f0-9]{32}/ AND CertThumbprint.raw:/[a-f0-9]{40}/
```

Но, наверное, самый популярный случай — это использование сертификатов, которые выписаны Microsoft Certificate Authority (строчка «Windows server CS role issued»). Эта служба может быть включена в компьютере под управлением серверной версии Microsoft Windows. Тут, наверное, следует заметить, что если вы самостоятельно осуществляете мониторинг вашей локальный инфраструктуры и не являетесь MSSP-провайдером, то вам будет значительно проще отфильтровать этот случай просто по имени CertIssuerName — по названию вашего CA-сервера (скорее всего, единственного на каждый домен в лесу). И действительно, даже большие корпоративные сети имеют достаточно небольшое количество CA, которые могут выдавать сертификаты. Но даже если вы — провайдер услуг безопасности, то для вас все равно не представит большого труда выяснить имена всех серверов клиентских PKI для того, чтобы их отфильтровывать. Я же хочу показать некоторую закономерность в **других** полях.

```
CertSerialNumber.raw:/[a-f0-9]{38}/ AND CertThumbprint.raw:/[a-f0-9]{40}/
```

Также могут встречаться сторонние реализации PKI, сертификатам которых доверяют серверы Kerberos при выдаче билетов. Например, у нас в мониторинге встречается специализированное ПО от фирмы Lanaco (не более десяти запросов за 30 дней). Мы его также можем отфильтровать.

Давайте посмотрим на реальных данных, какие запросы мы можем отфильтровать. Для этого можно построить следующую агрегацию, используя описанные выше регулярные выражения:



Агрегация событий запроса билета по сертификату в сервисе Kaspersky MDR

Обратим внимание на строчку «Rest» — это оставшиеся неотфильтрованными события (их 13 штук), их можно посмотреть в деталях. Обратите внимание на поле CertIssuerName, подробности будут ниже.

13 hits

Show chart

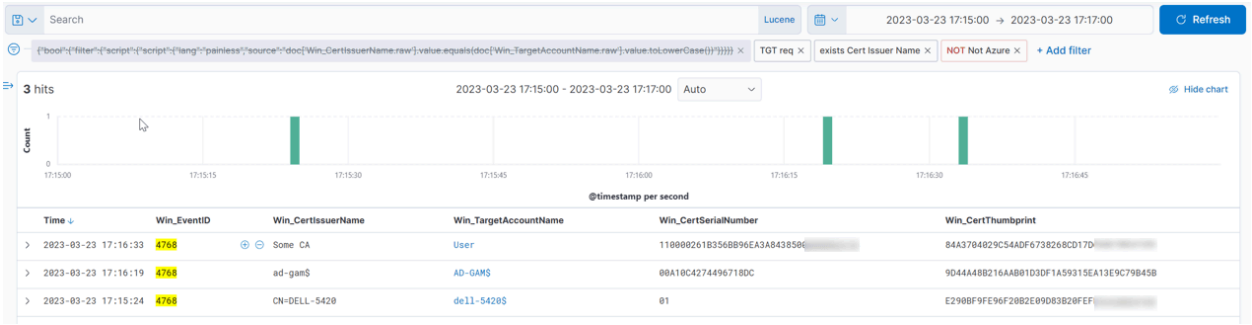
Time	Win_EventID	Win_CertIssuerName	Win_CertSerialNumber	Win_CertThumbprint	Win_TargetAccountName	Win_Status
2023-04-12 14:49:04	4768	ad-gam$	00CD5AC8DD61E4AFB6	D59D1BF48EED3BAFB7C5959944DC5FE99178DD0A	AD-GAM$	0x0
2023-04-04 19:36:22	4768	ad-gam$	7BA123B0F26C9DB9	7E3C6019BD510342748515 10A32574AAB5FC2F4C	AD-GAM$	0x0
2023-03-31 18:20:16	4768	ad-gam$	7BA123B0F26C9DB9	7E3C6019BD510342748515 10A32574AAB5FC2F4C	AD-GAM$	0x0
2023-03-31 12:24:59	4768	ad-gam$	7BA123B0F26C9DB9	7E3C6019BD510342748515 10A32574AAB5FC2F4C	AD-GAM$	0x0
2023-03-31 12:06:22	4768	ad-gam$	7BA123B0F26C9DB9	7E3C6019BD510342748515 10A32574AAB5FC2F4C	AD-GAM$	0x0
2023-03-31 11:58:42	4768	ad-gam$	7BA123B0F26C9DB9	7E3C6019BD510342748515 10A32574AAB5FC2F4C	AD-GAM$	0x0
2023-03-26 20:57:13	4768	CN=jane	00	6D7F4E6D1D4284A9F2A1DE374374C1DDB444974E	jane	0x0
2023-03-23 17:16:19	4768	ad-gam$	00A10C4274496718DC	9D44A48B216AAB01D3DF1A59315EA13E9C79B45B	AD-GAM$	0x0
2023-03-22 17:22:19	4768	ad-gam$	00A1018B1EB5AA9A94	CEA92EE5F7FF5E281EE1D7B20C42F1C0BB85BAEF	AD-GAM$	0x0
2023-03-06 18:52:10	4768	CN=jane	00	23EE63F02F430A568205437162EBD110CC1FC946	jane	0x0
2023-03-06 17:52:51	4768	CN=jane	00	78CA5897A760F152F8F8E51A6CD715D0B57F5C8B	jane	0x0
2023-03-06 16:38:49	4768	CN=jane	00	2E05375B9A22E5A0A1478CA7F1638A17C9AB5434	jane	0x0
2023-03-03 22:45:06	4768	CN=jane	00	48EB274B55813D5DFA5C4FEBB002FD81C91ADA8F	jane	0x0

Развернутый список неотфильтрованных событий запроса билета по сертификату

Исследуем код Whisker

Как я уже упоминал, в нашем примере сертификат был сгенерирован в самой утилите Whisker с параметрами по умолчанию. Сама процедура генерации самоподписанного сертификата описана тут.

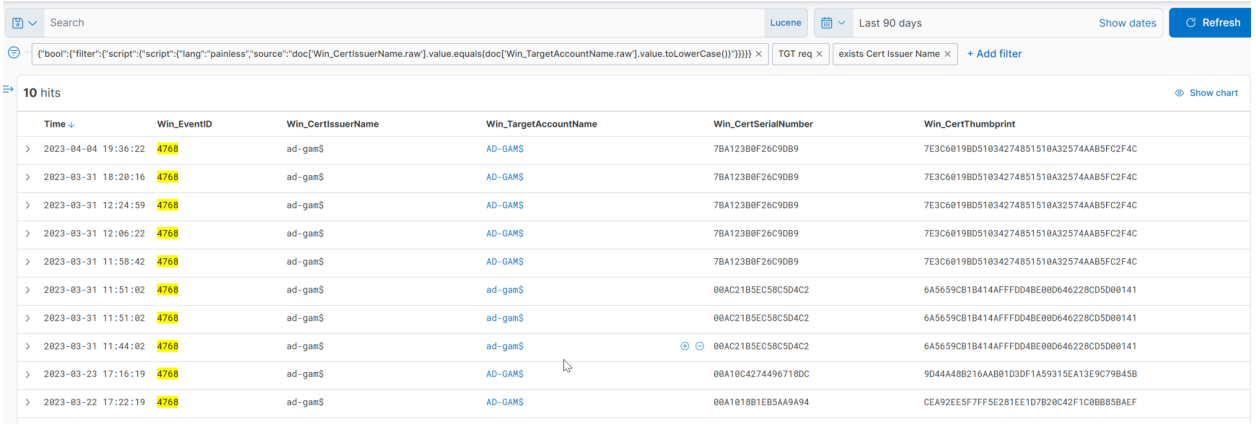
Whisker, как мы видим, пытается выдавать свои сертификаты за сертификаты от Windows Hello For Business (в случае с программным созданием ключевой пары). Однако в оригинальных сертификатах (когда ПК под управлением Windows самостоятельно сгенерировал сертификат для использования этого функционала) есть ошибка — в поле CertIssuerName записана нотация DN (от англ. Distinguished Name) в формате «CN=…». Инструментарий атакующих лишен этой ошибки — это и подозрительно. Можете сравнить вторую и третью строку с данными, полученными на тестовом стенде, но в продуктовой системе MDR (также см. ниже).



Мы можем прямо в Kibana добавить painless-скрипт, способный отыскать все события 4768, в которых CertIssuerName регистронезависимо совпадает с TargetAccountName.

```
{
  "query": {
    "bool": {
      "filter": {
        "script": {
          "script": {
            "lang": "painless",
            "source": "doc['CertIssuerName.raw'].value.equals(doc['TargetAccountName.raw'].value.toLowerCase())"
          }
        }
      }
    }
  }
}
```

Таких событий десять, и все они относятся к использованию утилиты Whisker.



Исследуем поля с флагами билета

Теперь рассмотрим поле «winlog.event_data.TicketOptionsDescription» в событиях с тестового стенда за произвольный интервал времени, где встречается как поддельный, так и легитимный запрос TGT.

Documents | Field statistics BETA

Columns | 1 field sorted

@timestamp	winlog.event...	winlog.event_data.TicketOptionsDescription	winlog.event_data.CertIssuerName	winlog.event_data.CertSerialNumber
2023-03-22 17:48:31.242	4768	[Renewable-ok, Renewable, Forwardable]	ad-gam$	00BA61092B330F639A
2023-03-22 17:43:48.025	4768	[Renewable-ok, Name-canonicalize, Renewable, Forwardable]	-	-
2023-03-22 17:43:16.602	4768	[Renewable-ok, Name-canonicalize, Renewable, Forwardable]	-	-
2023-03-22 17:43:12.706	4768	[Renewable-ok, Name-canonicalize, Renewable, Forwardable]	-	-
2023-03-22 17:43:12.675	4768	[Renewable-ok, Name-canonicalize, Renewable, Forwardable]	-	-
2023-03-22 17:43:09.832	4768	[Renewable-ok, Name-canonicalize, Renewable, Forwardable]	-	-
2023-03-22 17:40:37.159	4768	[Renewable-ok, Name-canonicalize, Renewable, Forwardable]	-	-
2023-03-22 17:27:17.085	4768	[Renewable-ok, Renewable, Forwardable]	ad-gam$	00A1018B1EB5AA9A94
2023-03-22 17:25:43.728	4768	[Renewable-ok, Renewable, Forwardable]	ad-gam$	00A1018B1EB5AA9A94
2023-03-22 17:25:04.840	4768	[Renewable-ok, Renewable, Forwardable]	ad-gam$	00A1018B1EB5AA9A94
2023-03-22 17:23:44.238	4768	[Renewable-ok, Renewable, Forwardable]	ad-gam$	00A1018B1EB5AA9A94
2023-03-22 17:22:14.150	4768	[Renewable-ok, Renewable, Forwardable]	ad-gam$	00A1018B1EB5AA9A94
2023-03-22 17:06:56.355	4768	[Renewable-ok, Name-canonicalize, Renewable, Forwardable]	-	-
2023-03-22 17:06:32.065	4768	[Renewable-ok, Name-canonicalize, Renewable, Forwardable]	-	-

В глаза бросается отсутствие флага Name-canonicalize, который играет важную роль в инфраструктуре Kerberos. Дело в том, что у службы или учетной записи может быть несколько основных имен. Например, если хост известен под несколькими именами, службы на его основе могут иметь несколько имен Service Principal Name (SPN). Чтобы клиент не нуждался в запросе билета для каждого из имен, KDC может предоставить ему информацию о сопоставлении в процессе получения учетных данных. Этот функционал и запрашивается при включении флага Name-canonicalize. Если установлена опция «канонизировать», то KDC **может** в ответе и TGT-билете изменить имена и SPN как клиента, так и сервера. Но в нашем случае этого флага нет, и это подозрительно. Найдем все билеты без этого флага, но запрошенные с помощью PKINIT (по сертификату). Запрос сделаем по продуктовым данным Kaspersky MDR.

Win_TicketOptionsList.raw:("Forwardable" AND "Renewable" AND "Renewable-ok" AND -"Name-canonicalize") | Lucene | Last 30 days | Show dates

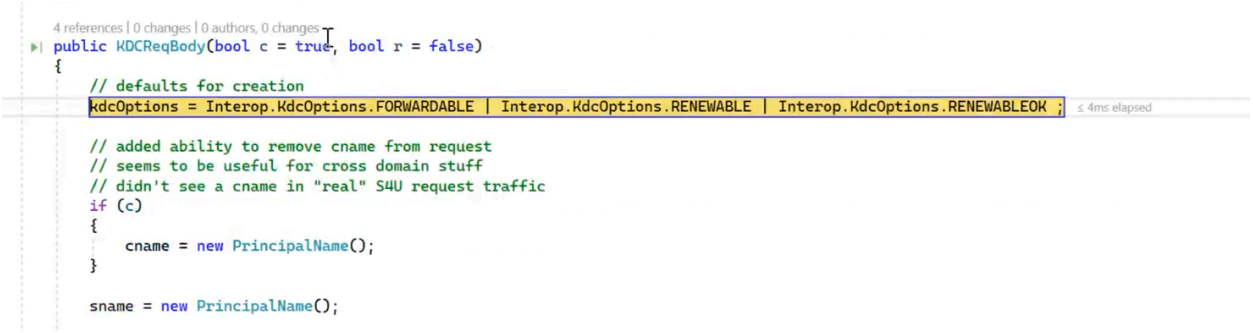
TGT req × | exists Cert Issuer Name × | computer_name: AD-gam.gam.click × | NOT Client's TP × | NOT That certificate are using on clients VMWare guest OS under custom Linux distro for signing × | + Add filter

16 hits

Time	computer_name	netremoteipv4str	Win_TargetAccountName	Win_CertIssuerName	Win_TicketOptionsList
2023-04-04 19:36:22	AD-gam.gam.click	192.168.232.1	AD-GAM$	ad-gam$	Forwardable, Renewable, Renewable-ok
2023-03-31 18:20:16	AD-gam.gam.click	192.168.232.1	AD-GAM$	ad-gam$	Forwardable, Renewable, Renewable-ok
2023-03-31 12:24:59	AD-gam.gam.click	192.168.232.1	AD-GAM$	ad-gam$	Forwardable, Renewable, Renewable-ok
2023-03-31 12:06:22	AD-gam.gam.click	192.168.232.1	AD-GAM$	ad-gam$	Forwardable, Renewable, Renewable-ok
2023-03-31 11:58:42	AD-gam.gam.click	192.168.232.129	AD-GAM$	ad-gam$	Forwardable, Renewable, Renewable-ok
2023-03-31 11:51:02	AD-gam.gam.click	192.168.232.129	ad-gam$	ad-gam$	Forwardable, Renewable, Renewable-ok
2023-03-31 11:51:02	AD-gam.gam.click	-	ad-gam$	ad-gam$	Forwardable, Renewable, Renewable-ok
2023-03-31 11:44:02	AD-gam.gam.click	192.168.232.1	ad-gam$	ad-gam$	Forwardable, Renewable, Renewable-ok
2023-03-26 23:51:05	DC.test.local	192.168.192.145	Administrator	test-DC-CA	Forwardable, Renewable, Renewable-ok
2023-03-26 23:47:04	DC.test.local	192.168.192.145	Administrator	test-DC-CA	Forwardable, Renewable, Renewable-ok
2023-03-26 22:52:44	DC.test.local	192.168.192.145	administrator	test-DC-CA	Forwardable, Renewable, Renewable-ok
2023-03-26 20:57:13	DC.test.local	192.168.192.145	jane	CN=jane	Forwardable, Renewable, Renewable-ok
2023-03-26 19:51:50	DC.test.local	192.168.192.145	DC$	test-DC-CA	Forwardable, Renewable, Renewable-ok

В результате мы можем видеть активность (за последние 30 дней) от Whisker + Rubeus на нашем стенде (хост AD-Gam) и работу моего коллеги (остальное) по тестированию группы уязвимостей слабых настроек AD CS, которые объединили под общим названием ADCS ESC или Certified Pre-Owned. Также есть одно отфильтрованное по имени сертификата ложное срабатывание и один инцидент, отправленный клиенту. Это неплохая конверсия.

Давайте рассмотрим на примере Rubeus, почему при запросе билета не выставляется флаг Name-canonicalize.



```
4 references | 0 changes | 0 authors, 0 changes
public KDCReqBody(bool c = true, bool r = false)
{
    // defaults for creation
    kdcOptions = Interop.KdcOptions.FORWARDABLE | Interop.KdcOptions.RENEWABLE | Interop.KdcOptions.RENEWABLEOK ;  ≤ 4ms elapsed

    // added ability to remove cname from request
    // seems to be useful for cross domain stuff
    // didn't see a cname in "real" S4U request traffic
    if (c)
    {
        cname = new PrincipalName();
    }

    sname = new PrincipalName();
```

Да, действительно, инструментарий Rubeus нарочно не выставляет этот флаг. Более того, то же самое делает Impacket — де-факто стандартный инструментарий для работы с Kerberos (и не только) для специалистов по анализу защищенности. Этим и объясняется то, что в ходе поиска билетов без флага мы также нашли и работу моего коллеги, который тестировал ESC-техники, — используемый им инструментарий базируется на Impacket. Такого рода утилит, благодаря простоте кода и популярности языка, очень много.

А что же сам атрибут msDS-KeyCredentialLink?

Мы можем сравнить два атрибута: один легитимно выставленный в процессе настройки Hello For Business, а другой — установленный Whisker. И разница между ними есть. Занимаясь сравнением этих атрибутов, я и написал утилиту, которая позволит вам найти артефакты от нелегитимной установки атрибута, например в результате использования Whisker.

Вы можете самостоятельно скачать и использовать эту утилиту в среде разработки, во время отладки попробуйте найти и сравнить ключевые отличия в «злом» и «добром» атрибуте.

На что нужно обратить внимание:

- Существует ли DeviceId у атрибута msDS-KeyCredentialLink (формат GUID). Если да, и при этом объекта с таким идентификатором нет в домене, — это подозрительно. Если же объект есть и при этом относится к коннектору Azure AD, то это похоже на легитимный случай.
- Поле Flags не содержит MFANotUsed. Как правило, в легитимном случае содержит.
- KeyMaterial имеет длину, отличную от 270 байт, — именно такие артефакты оставляет Whisker.
- Почти идентичные KeyApproximateLastLogonTimeStamp и Однако это менее надежный показатель, его лучше не использовать.

Резюмируем

Описанные выше атаки сравнительно эффективны, но их можно обнаруживать в момент использования поддельного сертификата. В этом специалисту по информационной безопасности помогают знание своей инфраструктуры (в идеале — составление списка всех активных ключей) и мониторинг. При этом понимание некоторых общих шаблонов и артефактов атак с использованием поддельного сертификата может оказать значительную поддержку, а моя утилита упростит процесс поиска.