Атаки на Active Directory: часть 1

defcon.ru/penetration-testing/18872

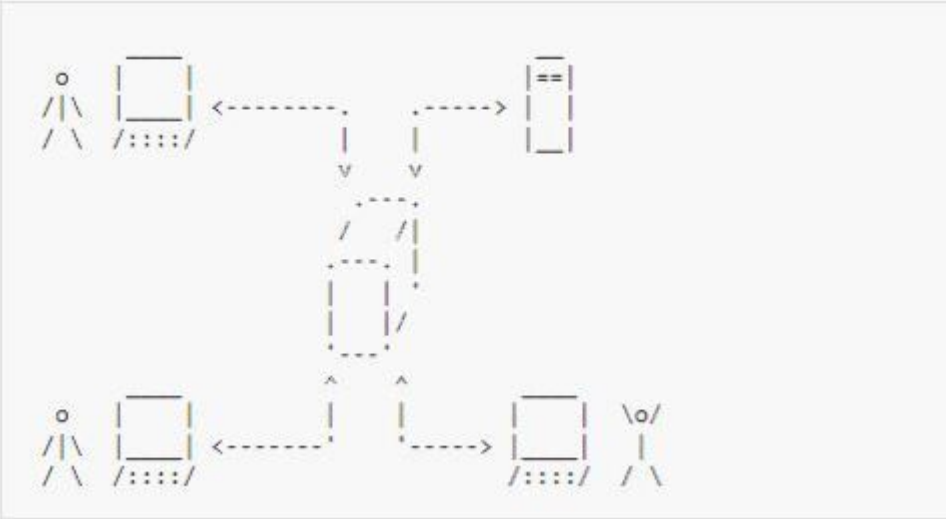
Это перевод статьи zer1t0, посвященный атакам на Active Directory. Цель статьи — рассмотреть **Active Directory** с точки зрения злоумышленника. Чтобы понять, как атаковать Active Directory (и любую другую технологию), я думаю, важно знать не только инструменты, но и то, как они работают, какие протоколы/механизмы они используют и почему эти механизмы/протоколы существуют.

На протяжении всей статьи будет использоваться Powershell, чтобы показать, как получить информацию из Active Directory. Для этого будет использоваться модуль ActiveDirectory Powershell, но вместо него можно использовать другие инструменты, такие как Powerview или ldapsearch.

> Информация предоставлена исключительно в ознакомительных целях. Не нарушайте законодательство!

Что такое Active Directory?

С моей точки зрения, Active Directory — это система, которая позволяет управлять набором компьютеров и пользователей, подключенных к одной сети, с центрального сервера. Конечно, это определение далеко не совсем точное, но я надеюсь, что оно достаточно простое, чтобы дать вам представление о том, что такое AD.



Представьте себе компанию с сотнями сотрудников, где каждый работает на своем (вероятно, Windows) компьютере. В этой компании есть несколько разных отделов, таких как продажи, отдел кадров, ИТ и т.д.

Теперь представьте, что отделу продаж требуется установить новую программу на своих рабочих станциях. Или что каждый день пользователь в другом офисе забывает свой пароль, и его нужно восстановить. Или что новая группа стажеров должна работать только с несколькими документами файлового сервера.

Должна ли IT-отдел устанавливать программу на все рабочие станции отдела продаж одну за другой? Должны ли они ходить каждый раз в разные офисы и восстанавливать пароль пользователя? Должны ли они создавать нового пользователя для каждого стажера на файловом сервере, который позволяет просматривать файлы только в каталоге?

В теории, они могли бы это сделать, хотя это создает много дополнительной работы и пустая трата ресурсов для компании. Но поскольку они умные люди, у них все компьютеры подключены к сети Active Directory, поэтому они могут выполнять все эти операции со своей рабочей станции. Active Directory позволяет это за счет ведения централизованной базы данных, в которой хранится вся информация о пользователях, компьютерах, политиках, разрешениях и т.д. Так, например, IT-отдел может подключиться к этой базе данных и создать новых пользователей для стажеров и назначить им разрешения на чтение файлов только в указанных каталогах определенных серверов их отделов.

Затем, когда один из этих стажеров пытается войти на компьютер в сети `Active Directory`, компьютер обращается к центральной базе данных, чтобы проверить, существует ли стажер-пользователь (и правильно ли введен пароль). Таким образом, пользователи могут входить на любой из компьютеров компании (если у них есть разрешения), позволяя сотрудникам использовать только пользователя для выполнения всей своей работы на всех компьютерах компании (это могут быть рабочие станции, серверы баз данных, файловые серверы, и т.д).

Точно так же, если пользователь забудет пароль, он может сообщить об этом IT-отделу, и они могут изменить пароль пользователя в этой центральной базе данных (и пользователю будет предложено изменить этот пароль на новый, который знает только он).

В случае отдела продаж ИТ-отдел может создать новую политику в базе данных, которая указывает, что компьютеры этого отдела должны установить указанную программу, и как они должны это делать. Затем, когда рабочая станция продавцов прочтет базу данных, они узнают, что должны выполнить эту политику, и будет установлена новая программа.

Я надеюсь, что этот пример позволит вам понять, почему Active Directory так полезна и почему ее использует практически любая организация в мире. Вероятно, вы использовали его, обычно с компьютера, который требует от вас нажатия

`Ctrl+Alt+Del` перед запросом имени пользователя и пароля.

Но что произойдет, если кто-то сможет украсть пароль пользователя? Может ли он изменить пароли других пользователей? А доступ к базе? Теперь понятно, почему Active Directory так важна, давайте рассмотрим его элементы.

Домены

То, что мы называем сетью Active Directory, обычно называют доменом. **Домен** — это набор подключенных компьютеров, которые совместно используют базу данных Active Directory, которой управляют центральные серверы домена, называемые контроллерами домена.

Доменное имя

У каждого домена есть DNS-имя. Во многих компаниях имя домена такое же, как и у их веб-сайта, например `contoso.com`, в то же время есть внутренний домен, такой как `contoso.local`.

```
PS C:\Users\Anakin> $env:USERDNSDOMAIN
CONTOSO.LOCAL
PS C:\Users\Anakin> (Get-ADDomain).DNSRoot
contoso.local
```

Определить текущий домен пользователя из Powershell

```
PS C:\Users\Anakin> (Get-WmiObject Win32_ComputerSystem).Domain
contoso.local
```

Определить текущий домен компьютера из Powershell

В дополнение к DNS-имени каждый домен также можно идентифицировать по NetBIOS-имени. Например, домен `contoso.local` может иметь имя NetBIOS CONTOSO. Вы можете увидеть имя `NetBIOS`, используемое в операциях входа в систему, где пользователь идентифицируется, например, как `CONTOSO\Administrator`, где первая часть — это имя NetBIOS, а вторая — имя пользователя.

Наконец, домен можно идентифицировать по его `SID` (идентификатору безопасности). SID больше используется программами (использующими Windows API), чем пользователями, но вы должны знать, как его получить, если он понадобится.

```
PS C:\Users\Anakin> Get-ADDomain | select DNSRoot,NetBIOSName,DomainSID

DNSRoot       NetBIOSName DomainSID
-------       ----------- ---------
contoso.local CONTOSO     S-1-5-21-1372086773-2238746523-2939299801
```

Получить DNS-имя, NetBIOS-имя и SID домена

Лес

Использование DNS-имени очень полезно, так как оно позволяет создавать поддомены для управления. Например, у компании может быть корневой домен с именем `contoso.local`, а затем поддомены для разных (обычно крупных) отделов, например `it.contoso.local` или `sales.contoso.local`.

Active Directory предлагает множество способов организации инфраструктуры, поэтому то, как организация использует поддомены, варьируется от одного к другому. Некоторые создают поддомены для отделов, а другие используют их для разных офисов.

contoso.local лес

Представленное дерево доменов известно как «Лес». Имя леса совпадает с именем корневого домена дерева.

```
PS C:\Users\Anakin> Get-ADForest


ApplicationPartitions : {DC=DomainDnsZones,DC=contoso,DC=local, DC=ForestDnsZones,DC=contoso,DC=local}
CrossForestReferences : {}
DomainNamingMaster    : dc01.contoso.local
Domains               : {contoso.local}
ForestMode            : Windows2016Forest
GlobalCatalogs        : {dc01.contoso.local, dc02.contoso.local}
Name                  : contoso.local
PartitionsContainer   : CN=Partitions,CN=Configuration,DC=contoso,DC=local
RootDomain            : contoso.local
SchemaMaster          : dc01.contoso.local
Sites                 : {Default-First-Site-Name}
SPNSuffixes           : {}
UPNSuffixes           : {}
```

Информация о лесе с помощью Get-ADForest

В лесу у каждого домена есть своя база данных и свои собственные контроллеры домена. Однако пользователи домена в лесу также могут получить доступ к другим доменам леса. Это означает, что даже если домен может быть автономным (без необходимости взаимодействия с другими доменами), он не изолирован с точки зрения безопасности, поскольку пользователь из одного домена по умолчанию может получить доступ к ресурсам других доменов в том же лесу. Однако пользователи леса по умолчанию не могут получить доступ к ресурсам из других лесов, поэтому лес является логической структурой, которая может обеспечить изоляцию с точки зрению безопасности.

Как уже говорилось, у каждого домена есть свои собственные контроллеры домена, поэтому, если отдел разрастется, могут понадобиться выделенные контроллеры домена, которые обрабатывают запросы всех компьютеров в этом отделе. Вы можете добиться этого, создав новый поддомен, и пользователи по-прежнему смогут получать доступ к компьютерам в других поддоменах того же леса.

Функциональные режимы

Как и компьютеры Windows, домены/леса также могут иметь свою собственную «версию», которая называется функциональным режимом. В зависимости от режима домена/леса могут использоваться новые характеристики.

Названия режимов основаны на минимальной операционной системе `Windows Server`, необходимой для работы с ними. Существуют следующие функциональные режимы :

- Windows2000
- Windows2000MixedDomains
- Windows2003
- Windows2008
- Windows2008R2
- Windows2012
- Windows2012R2
- Windows2016

```
PS C:\Users\Administrator\Downloads> (Get-ADForest).ForestMode
Windows2016Forest
PS C:\Users\Administrator\Downloads> (Get-ADDomain).DomainMode
Windows2016Domain
```

Получить режим леса/домена

Если, например, вы найдете домен/лес с режимом `Windows2012`, вы можете понять, что все контроллеры домена являются как минимум `Windows Server 2012`. Вы должны знать режим, чтобы использовать некоторые характеристики домена, например, группе `Protected Users` требуется режим `Windows2012R2`.

Доверие

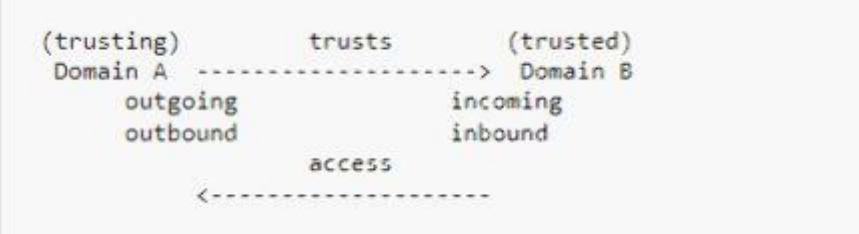
Пользователи могут получать доступ к другим доменам в тех же лесах, поскольку они связаны объединениями, называемыми `Trusts`.

Trusts — это соединение одного домена с другим. Не физическое сетевое соединение, а своего рода соединение для аутентификации/авторизации. Вы можете получить доступ к компьютерам в сети, которые находятся в других доменах, но вы не можете войти на эти компьютеры под своим пользователем этого домена. Это то, что доверие позволяет вам делать.

Направление доверия

Доверие – это направленное отношение, при котором одна сторона является доверяющей, а другая – доверенной. Когда эта связь установлена, пользователи доверенного домена могут получить доступ к ресурсам доверяющего домена.

Направление доверия противоположно направлению доступа. Например, если вы доверяете своей подруге, то вы позволяете ей получить доступ к вашему дому и есть вашу еду, когда она в ней нуждается.



Доверие от домена А к домену Б

Когда доверие направляется через ваш текущий домен, оно называется входящим доверием. Входящие доверительные отношения позволяют пользователям вашего домена получать доступ к другому домену.

С другой стороны, существуют исходящие доверительные отношения, которые переходят из вашего домена в другой. Поэтому пользователи другого домена могут получить доступ к вашему домену. И когда два домена связаны как входящим, так и исходящим доверием, говорят, что они связаны двунаправленным доверием (даже если на самом деле существует два доверия).

Вы можете увидеть доверительные отношения вашего домена с помощью команды `nltest /domain_trusts`.

```
PS C:\Users\Administrator> nltest /domain_trusts
List of domain trusts:
    0: CONTOSO contoso.local (NT 5) (Direct Outbound) ( Attr: foresttrans )
    1: ITPOKEMON it.poke.mon (NT 5) (Forest: 2) (Direct Outbound) (Direct Inbound) ( Attr: withinforest )
    2: POKEMON poke.mon (NT 5) (Forest Tree Root) (Primary Domain) (Native)
The command completed successfully
```

Трасты домена poke.mon

Здесь мы видим, что наш текущий домен `poke.mon` и есть пара трастов. Исходящее доверие с `contoso.local` указывает, что его пользователи могут получить доступ к нашему домену `poke.mon`. Более того, есть второй двунаправленный траст `it.poke.mon`, который является поддоменом `poke.mon` и находится в том же лесу.

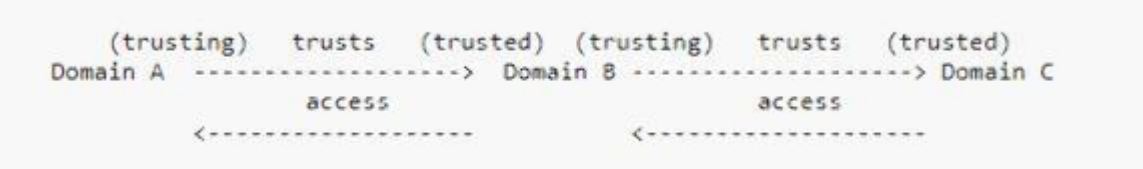
```
PS C:\Users\Anakin> nltest /domain_trusts
List of domain trusts:
    0: POKEMON poke.mon (NT 5) (Direct Inbound) ( Attr: foresttrans )
    1: CONTOSO contoso.local (NT 5) (Forest Tree Root) (Primary Domain) (Native)
The command completed successfully
```

Доверительные отношения contoso.local

Следовательно, если мы проверим доверие к `contoso.local`, мы увидим входящее соединение от `poke.mon`, что согласуется с предыдущей информацией. Таким образом, пользователи `contoso.local` могут получить доступ к файлам `poke.mon`.

Транзитивное доверие

Более того, доверие может быть транзитивным или нетранзитивным. Нетранзитивное доверие может использоваться только двумя сторонами доверия, доверяющей и доверенной. Транзитивное доверие может действовать как мост и использоваться для третьих доменов, связанных с доменами, которые связаны транзитивным доверием.



Три домена, связанные доверием

Например, если доверие между доменом **A** и **B** является транзитивным, то пользователи домена **C** могут получить доступ к домену **A** через оба доверия. Если доверие между доменами **A** и **B** было нетранзитивным, то пользователи домена **C** не могли получить доступ к домену **A**, но это могли бы сделать пользователи домена **B**.

Следовательно, по отношению к доменам в одном лесу все пользователи доменов могут получить доступ к другим доменам, потому что все родительские и дочерние домены связаны через двунаправленные транзитивные доверительные отношения. Таким образом, любой домен леса может пройти необходимые доверительные отношения для доступа к другому домену в том же лесу.

В лесу, чтобы разрешить доступ из любого домена к любому другому, все родительские и дочерние домены связаны двунаправленным транзитивным доверием.

contoso.local доверяет лесам

Таким образом, чтобы получить доступ к компьютерам `hr.contoso.local`, пользователь `webs.it.contoso.local` должен пройти через 3 траста.

Типы доверия

В Active Directory существует несколько типов доверия для разных целей:

- **Родительский-дочерний** — доверительные отношения по умолчанию, созданные между родительским доменом и его дочерним;
- **Лес** — доверие для обмена ресурсами между лесами. Таким образом, любой домен леса может получить доступ к любому домену другого леса (если это позволяют направление и транзитивность доверия). Если доверие леса настроено неправильно, оно может позволить получить контроль над другим лесом;
- **Внешний** — доверие для подключения к определенному домену, который находится в недоверенном лесу;
- **Область** — специальное доверие для соединения Active Directory и домена, отличного от Windows;
- **Ярлык** — когда два домена в лесу часто взаимодействуют, но не связаны напрямую, вы можете избежать перехода через множество доверительных отношений, создав прямое сокращенное доверие.

Доверительный ключ

Технически, когда вы используете доверие, существует связь между контроллером домена вашего домена и контроллером домена целевого домена (или промежуточного домена). Способ установления связи зависит от используемого протокола (это может быть NTLM, Kerberos и т.д.), но в любом случае контроллеры домена должны совместно использовать ключ для обеспечения безопасности связи. Этот ключ известен как ключ доверия и создается при установлении доверия.

При создании доверия в базе данных домена создается доверительная учетная запись с именем, заканчивающимся на `$`. Затем создается доверенный ключ, как пароль доверенного пользователя (в виде хеша NT и ключей Kerberos).

Пользователи

Одним из ключевых моментов использования Active Directory является управление пользователями. Каждая организация управляет своими пользователями по-разному, устанавливая для них форматы имен, назначая разные разрешения и т.д. Чтобы упростить управление пользователями в Active Directory, они хранятся в виде объектов в центральной базе данных, к которыми можно обращаться и манипулировать ими из любой точки домена, если у вас достаточно прав.

Свойства пользователя

Идентификаторы пользователей

Пользовательский объект хранит множество различных данных, но в первую очередь следует учитывать те атрибуты, которые позволяют нам идентифицировать пользователя.

Для идентификации пользователя обычно используется имя пользователя, которое хранится в атрибуте `SamAccountName`. Кроме того, `SID` (идентификатор безопасности) также может использоваться для идентификации пользователя.

SID пользователя похож на SID домена и фактически представляет собой комбинацию SID домена и RID пользователя (относительный идентификатор), который является последним числом, которое появляется в SID пользователя.

```
PS C:\Users\Anakin> Get-ADUser Anakin


DistinguishedName : CN=Anakin,CN=Users,DC=contoso,DC=local
Enabled           : True
GivenName         : Anakin
Name              : Anakin
ObjectClass       : user
ObjectGUID        : 58ab0512-9c96-4e97-bf53-019e86fd3ed7
SamAccountName    : anakin
SID               : S-1-5-21-1372086773-2238746523-2939299801-1103
Surname           :
UserPrincipalName : anakin@contoso.local
```

Получить информацию о пользователе

В этом случае SID домена `S-1-5-21-1372086773-2238746523-2939299801` и RID пользователя `1103`. Некоторые инструменты отображают `SID` в своих выходных данных вместо имени пользователя (поскольку оно используется в некоторых структурах, таких как дескрипторы безопасности), поэтому вам следует знать его формат, чтобы идентифицировать его.

Кроме того, `DistinguishedName` используется `API LDAP` для идентификации объектов, поэтому, если вы запрашиваете базу данных с помощью `LDAP` (что является одним из наиболее распространенных способов), вы, вероятно, увидите ссылки на объекты через его файлы `DistinguishedName`.

«Секреты» пользователя

Кроме того, база данных также должна хранить секреты пользователя, чтобы позволить контроллеру домена аутентифицировать пользователя. Пароль пользователя не хранится в открытом виде, но сохраняются следующие полученные из него секреты: хеш NT (и хеш LM для старых учетных записей).

Ключи Kerberos

Излишне говорить, что пользовательские секреты не могут быть получены пользователями без прав администратора. Даже компьютеры домена не могут получить к ним доступ, но оставляют аутентификацию контроллеру домена.

Чтобы получить пользовательские секреты, вам нужны права администратора (или эквивалентные) для сброса базы данных домена с помощью атаки `dcsync` или захвата `C:\Windows\NTDS\ntds.dit` файла с контроллера домена.

Хеши LM/NT

Хеши LM и NT хранятся как в локальной базе данных Windows `SAM`, так и в базе данных `Active Directory NTDS` для аутентификации локальных пользователей и пользователей домена соответственно. Эти хеши, как LM, так и NT, имеют длину 16 байт.

LM и NT хэши пароля

```
Password: 123456
LM hash: 44EFCE164AB921CAAAD3B435B51404EE
NT hash: 32ED87BDB5FDC5E9CBA88547376818D4
```

Однако хеши LM довольно слабые, поэтому они не используются, начиная с Windows Vista/Server 2008. Процедура создания хеша LM следующая:

- пароль пользователя преобразовывается в верхний регистр (что облегчает атаку полным перебором);

- если пароль пользователя меньше 14 символов, он дополняется символами `NULL` до тех пор, пока его длина не станет 14. Если пароль больше 14 символов, он усекается (поэтому бесполезно использовать пароли более 14 символов);
- затем пароль разбивается на две строки по 7 байт каждая;
- каждая 7-байтовая строка используется в качестве ключа для шифрования `KGS!+#$%` строки с использованием криптографического алгоритма `DES`. В результате получается два хэша;
- два результирующих значения объединяются для формирования хэша LM (можно взломать каждую часть отдельно).

```
upper_password = to_uppercase(password)
14_password = truncate_to_14_bytes(upper_password)

7_part1, 7_part2 = split_7(14_password)

hash1 = des(7_part1, "KGS!+#$%")
hash2 = des(7_part2, "KGS!+#$%")

lm_hash = hash1 + hash2
```

Псевдокод вычисления хэша LM

С другой стороны, хеш NT немного сильнее, но для его вычисления не используется соль, поэтому его можно взломать, используя предварительно вычисленные значения (например, радужные таблицы). Хеш NT вычисляется путем применения алгоритма `MD4` (который устарел) непосредственно к `Unicode` (в частности, кодировке `UTF-16LE`) пароля пользователя.

```
nt_hash = md4(encode_in_utf_16le(password))
```

Псевдокод вычисления хэша NT

Очень часто хеш NT называется хешем NTLM, однако это может сбивать с толку, поскольку протокол NTLM также использует хеши, называемые хешами NTLM. В этой статье хеш NTLM является хешем протокола NTLM.

Многие инструменты позволяют извлекать хэши LM и NT и обычно возвращают вывод с несколькими строками, по одной на пользователя, в формате `<username>:<rid>:<LM>:<NT>:::`. В случае, если LM не используется, его значение будет `aad3b435b51404eeaad3b435b51404ee` (LM-хеш пустой строки).

```
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:6535b87abdb112a8fc3bf92528ac01f6:::
user:1001:aad3b435b51404eeaad3b435b51404ee:57d583aa46d571502aad4bb7aea09c70:::
```

Формат дампа хэшей

Для пентестера важно распознавать хеши NT, поскольку, даже если они не являются паролями пользователей, они используются для аутентификации на машинах Windows, поэтому они очень полезны. Их можно использовать для выполнения атак `Pass-The-Hash` или `Overpass-the-Hash`, чтобы выдавать себя за пользователей на удаленных машинах.

Кроме того, вы можете попытаться взломать хеши LM и NT с помощью hashcat, чтобы восстановить исходный пароль. Если вам повезет и LM-хеш присутствует, подбор пройдет быстро.

Ключи Kerberos

Помимо хэшей LM/NT, сохраняются ключи `Kerberos`, полученные из пароля пользователя и используемые в протоколе аутентификации Kerberos. Ключи Kerberos можно использовать для запроса билета Kerberos, который представляет пользователя при проверке подлинности Kerberos. Существует несколько разных ключей, и разные используются для разной поддержки шифрования Kerberos:

- **Ключ AES 256** — используется алгоритмом `AES256-CTS-HMAC-SHA1-96`. Это тот, который обычно используется `Kerberos` и его должен использовать пентестер, чтобы избежать срабатывания аварийных сигналов;
- **Ключ AES 128** — используется алгоритмом AES128-CTS-HMAC-SHA1-96;
- **Ключ DES** — используется устаревшим алгоритмом DES-CBC-MD5;
- **Ключ RC4** — это NT-хэш пользователя, используемый алгоритмом RC4-HMAC.

```
$ secretsdump.py 'contoso.local/Administrator@192.168.100.2' -just-dc-user anakin
Impacket v0.9.21 - Copyright 2020 SecureAuth Corporation

Password:
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
contoso.local\anakin:1103:aad3b435b51404eeaad3b435b51404ee:cdeae556dc28c24b5b7b14e9df5b6e21:::
[*] Kerberos keys grabbed
contoso.local\anakin:aes256-cts-hmac-sha1-96:ecce3d24b29c7f044163ab4d9411c25b5698337318e98bf2903bbb7f6d76197e
contoso.local\anakin:aes128-cts-hmac-sha1-96:18fe293e673950214c67e9f9fe753198
contoso.local\anakin:des-cbc-md5:fbba85fbb63d04cb
[*] Cleaning up...
```

Ключи Kerberos, извлеченные из базы данных домена

Эти ключи можно использовать в атаке `Pass-The-Key` для получения билета для олицетворяемого пользователя. Затем вы можете использовать этот билет Kerberos для аутентификации в различных службах домена от имени пользователя.

UserAccountControl

Одним интересным свойством пользовательского класса является `UserAccountControl` (UAC) (не путайте его с механизмом контроля учетных записей, чтобы избежать выполнения программ с повышенными правами на компьютерах с

Windows).

Свойство `UserAccountControl` содержит ряд флагов, которые очень важны для безопасности и домена и используются во многих атаках, упомянутых в этом посте. Вот самые актуальные:

- **ACCOUNTDISABLE** — Учетная запись отключена и не может быть использована;
- **DONT_REQUIRE_PREAUTH** — Учетная запись не требует предварительной аутентификации Kerberos;
- **NOT_DELEGATED** — Эта учетная запись не может быть делегирована с помощью делегирования Kerberos;
- **TRUSTED_FOR_DELEGATION** — Неограниченное делегирование Kerberos включено для этой учетной записи и ее служб. Для его изменения требуется SeEnableDelegationPrivilege;
- **TRUSTED_TO_AUTH_FOR_DELEGATION** — Расширение Kerberos S4U2Self включено для этой учетной записи и ее служб. Для его изменения требуется SeEnableDelegationPrivilege.

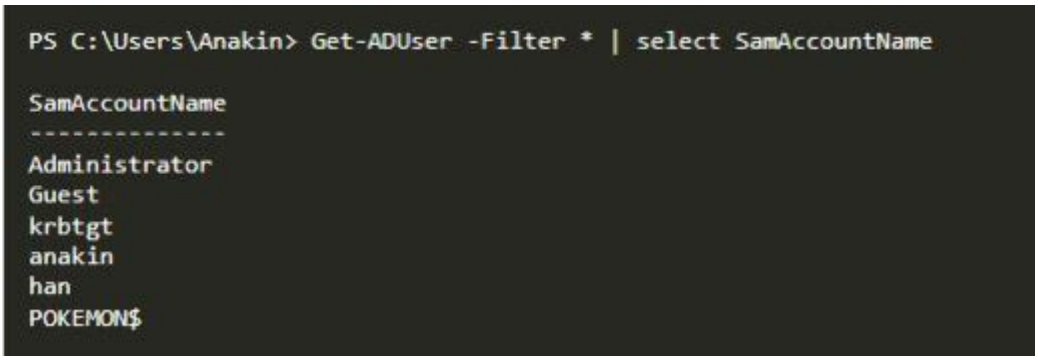
Другие свойства пользователя

Есть и другие свойства, которые могут быть полезны в пентесте:

- **Description** — Описание пользователя. Он может дать представление о правах пользователя, а иногда даже включает пароль;
- **AdminCount** — Указывает, защищен ли пользователь (или группа) объектом `AdminSDHolder`, или он был защищен. Поскольку иногда не обновляется, используйте его только как ссылку;
- **MemberOf** — Группы, членом которых является пользователь. Это свойство является логическим и генерируется из свойства группы `Members`;
- **PrimaryGroupID** — Основная группа пользователя. Эта группа не отображается в свойстве `MemberOf`;
- **ServicePrincipalName** — Службы пользователя. Может быть полезен для атаки `Kerberoast`;
- **msDS-AllowedToDelegateTo** — Список служб, для которых пользователь (и его собственные службы) может олицетворять клиентов с помощью ограниченного делегирования Kerberos. Для его изменения требуется `SeEnableDelegationPrivilege`.

Важные пользователи

Для обращения с пользователями есть несколько вариантов, таких как команда `net user /domain` команда или `Powershell`. Для вывода списка пользователей не требуется иметь специальные привилегии, это может сделать любой пользователь.



Список пользователей с Powershell

Как вы могли заметить, тестовый домен небольшой, с небольшим количеством пользователей, но в действительности будут сотни или тысячи пользователей. Поэтому важно различать, что действительно важно. Это может быть немного сложно, так как это зависит от организации, но обычно члены ИТ-отдела используют привилегированных пользователей, им это нужно для работы.

Более того, по умолчанию встроенный пользователь `Administrator` является самой привилегированной учетной записью домена. Он может выполнять любые действия на любом компьютере. Таким образом, если вы сможете скомпрометировать эту учетную запись, вы сможете получить полный контроль над доменом и даже над лесом, используя атаку истории SID.

Кроме того, учетная запись `krbtgt` также очень важна. Его секреты (хеш NT и ключи Kerberos) используются для шифрования билетов (в частности, `TGT`), используемых Kerberos, что позволяет аутентифицировать пользователей. Если вам удастся скомпрометировать учетную запись `krbtgt`, вы сможете создать `Golden Tickets`. Обычно эту учетную запись можно скомпрометировать только путем сброса базы данных домена, поскольку она используется только в контроллерах домена, для чего потребуются права администратора в домене.

Учетные записи компьютеров

Еще одна вещь, которую следует учитывать, это то, что в организации у каждого человека есть свой пользователь, и даже у некоторых людей, таких как ИТ-отдел, может быть несколько пользователей на каждого для выполнения различных задач. Более того, также у каждого компьютера домена есть свой пользователь, так как им тоже нужно выполнять свои действия в домене.

Разница между учетными записями пользователей и учетными записями компьютеров заключается в том, что первые хранятся как экземпляры класса `User` в базе данных, тогда как другие хранятся как экземпляры класса `Computer` (который является подклассом класса `User`). Более того, имена учетных записей компьютеров представляют собой имена хостов компьютеров, заканчивающиеся знаком доллара `$`.

Вы можете проверить это, выполнив следующую команду:

```
PS C:\> Get-ADObject -LDAPFilter "objectClass=User" -Properties SamAccountName | select SamAccountName

SamAccountName
--------------
Administrator
Guest
DC01$
krbtgt
anakin
WS01-10$
WS02-7$
DC02$
han
POKEMON$
```

Список всех пользователей домена

Как видите, пользователей намного больше, чем при использовании команды `Get-ADUser`, поскольку теперь включены подклассы класса User. Вы можете оценить, что новые учетные записи заканчиваются знаком доллара и, кажется, имеют имя компьютера. Например, `DC01$` и `DC02$` для Контроллеров домена, `WS01-10$` и `WS02-7$` для рабочих станций.

Более того, объекты-компьютеры также сохранили информацию об их операционной системе, которую можно получить из атрибутов `OperatingSystem` или файлов `OperatingSystemVersion`.

Кроме того, во многих организациях существуют правила выбора имен компьютеров и пользователей, поэтому, если вы в состоянии понять имена, вы можете быть осведомлены об использовании компьютеров и учетных записей пользователей, а также о том, какие из них могут быть привилегированные или содержат доступ к важной информации. Кроме того, вы можете проверить другие атрибуты объектов, например `Description`, чтобы найти там больше информации (и даже пароли в открытом виде). Для этой цели может быть полезен командлет `Find-DomainObjectPropertyOutlier Powerview`.

Доверенные учетные записи

Однако есть также учетная запись `POKEMON$`, которая появляется в обоих `Get-ADUser` и `Get-ADObject`, но имя которой заканчивается знаком доллара. Это может быть обычный пользователь (нет проблем с созданием имен пользователей, заканчивающихся на `$`), однако, как мы видели ранее, существует доверие с `poke.mon` доменом.

При установлении доверия в каждом домене создается связанный пользовательский объект для хранения ключа доверия. Имя пользователя — это NetBIOS-имя другого домена, заканчивающееся `$` (аналогично имени учетной

записи компьютера). Например, в случае доверия между доменами FOO и BAR, домен FOO будет хранить ключ доверия в пользователе BAR$, а домен BAR будет хранить его в пользователе FOO$.

```
PS C:\> Get-ADUser  -LDAPFilter "(SamAccountName=*$)" | select SamAccountName

SamAccountName
--------------
POKEMON$
```

Список доверенных учетных записей в домене

Пользовательский объект POKEMON$ используется для хранения ключей доверия, которые являются хешем NT или ключами Kerberos (в зависимости от контекста используется один или другой). Если вы можете получить секреты этой учетной записи, вы можете создавать билеты Kerberos между областями.

Практическая подготовка

Если материал показался вам интересным, и хотите на практике разобраться, как это работает — пройдите Корпоративные лаборатории Pentestit — программу практической подготовки в области информационной безопасности.