Атаки на Active Directory: часть 7

defcon.ru/penetration-testing/19041

Заключительная седьмая часть перевода статьи zer1t0, посвященная аутентификации, групповым политикам и протоколам связи в Active Directory.

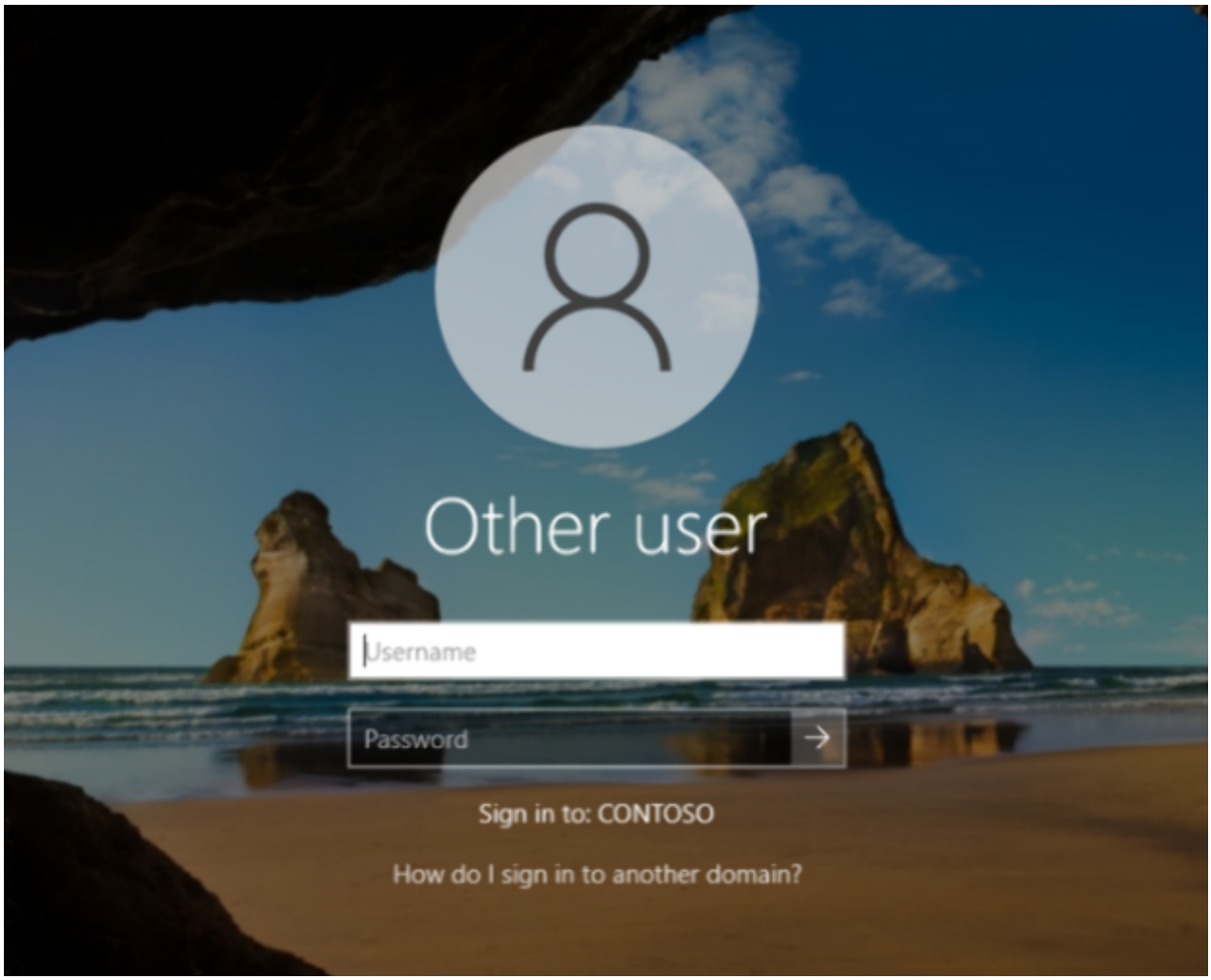
> Информация предоставлена исключительно в ознакомительных целях. Не нарушайте законодательство!

Типы аутентификации

Для входа пользователей как локально, так и удаленно, Windows определяет различные типы входа в систему, которые важны для злоумышленника по нескольким причинам. Во-первых, не каждый тип входа в систему может использоваться любым пользователем. Во-вторых, многие типы входа в систему кэшируют учетные данные в процессе `lsass` или даже в секретах `LSA`, которые могут быть восстановлены пентестером.

Интерактивный вход

Интерактивный вход в систему или локальный вход в систему происходит, когда происходит вход в систему на физической машине или при использовании `runas`. Учетные данные кэшируются в процессе `lsass` компьютера.



Интерактивный вход

```
runas /user: cmd
```

Интерактивный вход в систему с runas

При этом типе входа в систему, в случае локальных учетных записей, компьютер проверяет пароль, сравнивая свой хэш NT с хэшем, хранящимся в `SAM`. Если пользователь использует учетную запись домена, компьютер проверяет учетные данные пользователя, запрашивая `TGT Kerberos` у контроллера домена, который кэшируется на компьютере. Если контроллер домена недоступен, компьютер проверяет учетные данные пользователя в хранилище кэшированных учетных данных домена (DCC), которое кэширует учетные данные последних пользователей домена, вошедших в систему на компьютере. Если учетные данные домена не кэшированы, компьютер не сможет аутентифицировать пользователя.

После проверки подлинности хэш NT, полученный из пароля, сохраняется в процессе `lsass`. Для учетных записей домена также кэшируются ключи `Kerberos`, полученные из пароля пользователя, и билеты для обеспечения единого входа (`SSO`). На старых компьютерах кэшируется даже простой пароль.

Для выполнения интерактивного входа в систему может потребоваться `SeInteractiveLogonRight`, особенно на контроллерах домена или других компьютерах с Windows Server.

Вход по сети

Вход происходит, когда вы подключаетесь к удаленному компьютеру с помощью неинтерактивной службы, такой как `SMB`, `RPC`, `SQL` и т.д. Для такого входа вам требуется пароль, хэш NT или билет Kerberos, поэтому они восприимчивы к атакам Pass-The-Hash, передачей ключа или передачей билета. Одним из важных фактов является то, что учетные данные не кэшируются на удаленном компьютере, за исключением случаев, когда включено делегирование Kerberos.

Это, вероятно, тип входа в систему, который чаще всего используется злоумышленником из-за того, что также чаще всего используется легитимными пользователями, поскольку компьютеры постоянно подключаются друг к другу в домене.

Psexec, пакет impacket и удаленный Powershell (использующий WinRM по умолчанию) используют этот тип аутентификации, даже если они предоставляют интерактивную оболочку.

Вот несколько примеров аутентификации по сети:

```
dir \\ws01-10\Temp
```
Доступ к общему ресурсу

```
.\PsExec.exe \\dc01 cmd
```
Выполнение PsExec

В этом типе аутентификации клиент подключается к удаленному компьютеру и использует `SPNEGO` для согласования протокола аутентификации и использует Kerberos или NTLM. Поскольку при использовании любого из этих протоколов учетные данные пользователя не отправляются напрямую, они не могут быть кэшированы на целевом компьютере. Исключение составляет, если включено делегирование Kerberos.

Имейте в виду, что даже если вы можете выполнить вход в сеть, может быть много причин, по которым служба не может быть использована. Во-первых, брандмауэр, запрещающий подключение к удаленным службам, а во-вторых, многие службы, доступные для аутентификации через сеть, могут использоваться только администраторами.

Например, вы можете использовать вход в сеть для доступа к некоторым общим ресурсам удаленного компьютера, но не можете запустить оболочку с помощью `PsExec`, поскольку для этого требуется доступ к диспетчеру служб, доступ к которому имеют только администраторы.

Вход в качестве пакетного задания

Используется для запуска запланированных задач в контексте пользователя. Учетные данные будут кэшироваться в процессе `lsass` при выполнении задачи.

```
schtasks.exe /create /tn notepaddaily /tr notepad.exe /sc daily /ru CONTOSO\TaskUser /rp task1234!
```

Создание задачи с учетными данными пользователя

Имейте в виду, что пакетный вход в систему будет производиться при выполнении задачи, а не при ее создании. Так что, возможно, у вас есть привилегии для запуска в качестве задачи (например, `SeBatchLogonRight`), но вы не можете создать задачу. Например, у операторов резервного копирования есть право `SeBatchLogonRight`, но они не могут создавать задачи (по умолчанию).

При запуске задачи учетные данные проверяются и кэшируются, как и при интерактивном входе в систему.

Вход в качестве службы

Вход в качестве службы используется, когда служба будет запущена в контексте пользователя. Простой пароль хранится в секретах `LSA` компьютера, а учетные данные будут кэшироваться в процессе `lsass` при выполнении службы.

```
sc.exe create MySvc2 binpath= c:\windows\system32\notepad.exe obj=CONTOSO.local\svcUser password=svc1234!
```

Создание службы с учетными данными пользователя

Имейте в виду, что вход в качестве службы будет производиться при выполнении службы, а не при ее создании. Поэтому, возможно, даже если у вас есть привилегии для входа в качестве службы (например, `SeServiceLogonRight`), но вы не сможете создать ее.

При запуске службы учетные данные проверяются и кэшируются, как и при интерактивном входе в систему.

Вход в NetworkCleartext

В случае входа в систему `NetworkCleartext` пароль отправляется по сети на целевой компьютер в зашифрованном виде. Этот тип входа используется Powershell, когда указана проверка подлинности CredSSP. CredSSP выполняет сетевую аутентификацию с использованием NTLM или Kerberos и при создании зашифрованного канала отправляет пароль на целевой компьютер.

Следует отметить, что учетные данные кэшируются на целевом компьютере.

```
New-PSSession -Credential $(Get-Credential) -Authentication Credssp
```

Вход в систему NetworkCleartext с помощью удаленного управления Powershell

Вход с новыми учетными данными

Вход в систему `NewCredentials` происходит при использовании `runas` с расширением `/netonly`. Затем запущенный процесс будет использовать учетные данные только для удаленных подключений, сохраняя текущий сеанс пользователя для локальных операций.

Учетные данные кэшируются в локальном процессе `lsass`, чтобы их можно было использовать для сетевых подключений. Затем, когда процесс требует этого, он может выполнять вход в сеть для доступа к удаленным ресурсам домена.

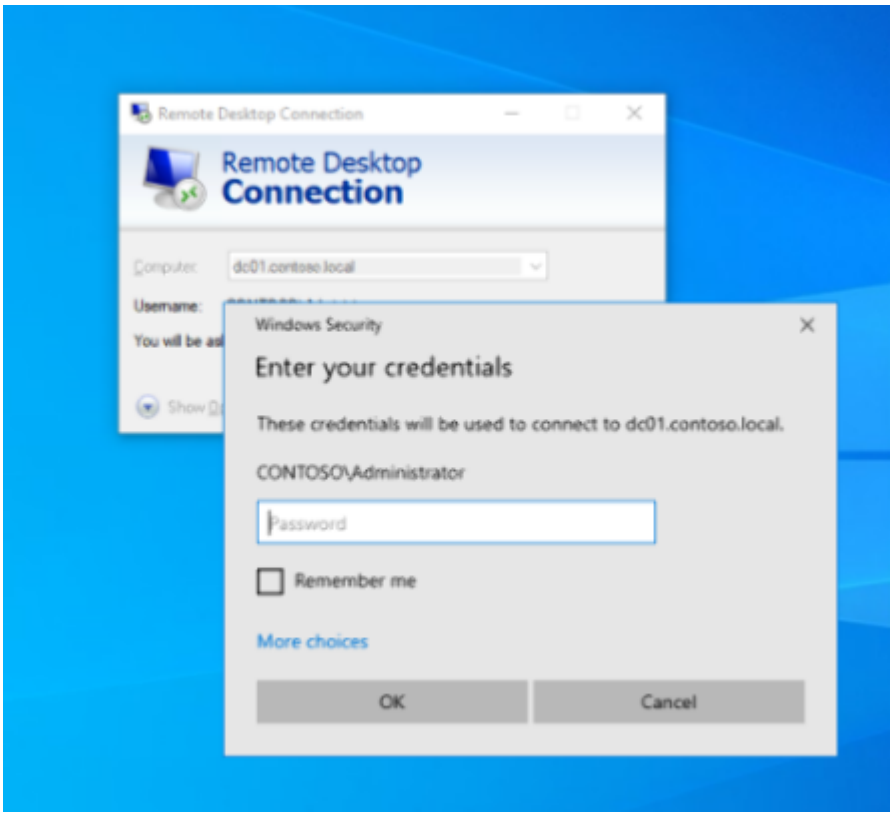
Вход NewCredentials с помощью runas

```
runas /netonly /user:CONTOSO\OtherUser cmd
```

Учетные данные не проверяются
до тех пор, пока не будет выполнено сетевое подключение, но кэшируются при выполнении команды `runas`, как и при интерактивном входе в систему (за исключением билетов `Kerberos`, поскольку они извлекаются при проверке учетных данных). Этот метод позволяет кэшировать поддельные учетные данные в процессе `lsass` и иногда используется BlueTeam для создания поддельных учетных данных для обнаружения злоумышленников.

Удаленный интерактивный вход

Вход в `RemoteInteractive` используется при подключении к компьютеру через `RDP`. RDP использует CredSSP для удаленного входа в систему, поэтому пароль отправляется по сети на целевой компьютер, а учетные данные кэшируются в удаленном процессе `lsass`.



Вход в RemoteInteractive с использованием RDP

Аутентификация похожа на сетевой вход, но учетные данные отправляются на целевой компьютер, поэтому они кэшируются, как и при интерактивном входе в систему.

Чтобы иметь возможность войти на удаленный компьютер с помощью входа в `RemoteInteractive`, ваш пользователь должен быть частью пользователей удаленного рабочего стола или иметь право `SeRemoteInteractiveLogonRight` на целевом компьютере.

Авторизация

Как только клиент смог разрешить целевое имя хоста и пройти аутентификацию, целевая служба/программа/компьютер теперь должна знать о своих разрешениях, то есть знать имя пользователя и SID, а также группы, к которым он принадлежит. Как только эта информация станет известна, программа может решить, имеет ли пользователь достаточные привилегии для доступа к определенным объектам.

ACL

Дескриптор безопасности

Как проверить, есть ли у пользователя доступ к объекту? Проверяя его дескриптор безопасности. В Active Directory каждый объект базы данных имеет связанный с ним дескриптор безопасности в свойстве `NTSecurityDescriptor`. Дескриптор

безопасности хранится в двоичном формате, но его также можно преобразовать в строковый формат.

Дескриптор безопасности содержит следующую информацию о безопасности:

- `SID` владельца объекта;
- `SID` основной группы владельца;
- `DACL` (дискреционный список управления доступом, необязательно);
- `SACL` (системный список управления доступом, необязательно).

```
PS C:\> $(Get-ADUser anakin -Properties nTSecurityDescriptor).nTSecurityDescriptor | select Owner,Gro
up,Access,Audit | Format-List


Owner  : CONTOSO\Domain Admins
Group  : CONTOSO\Domain Admins
Access : {System.DirectoryServices.ActiveDirectoryAccessRule, System.DirectoryServices.ActiveDirectoryAccessRule,
         System.DirectoryServices.ActiveDirectoryAccessRule, System.DirectoryServices.ActiveDirectoryAccessRule...}
Audit  :
```

Получение дескриптора безопасности пользовательского объекта

В каждом дескрипторе безопасности может быть два `ACL` (списка управления доступом): `DACL` и `SACL`. ACL представляет собой список ACE (запись управления доступом). ACE в SACL определяют попытки доступа, которые будут генерировать журналы, и они могут быть полезны с точки зрения защиты.

Однако наиболее важной частью является DACL, который обычно присутствует во всех объектах, ACE которого определяет пользователей/группы, которые могут получить доступ к объекту, и тип разрешенного доступа. Обычно, когда кто-то обращается к объектному ACL, имеется в виду DACL.

ACE

Каждый ACE состоит из нескольких частей :

- **тип ACE** — указывает, предназначен ли `ACE` для разрешения или отказа в доступе (или доступа к журналу в случае SACL);
- **наследование** — указывает, наследуется ли `ACE`;
- **идентификация** — указывает участника (пользователя/группу), для которого применяется `ACE`. Основной `SID` сохраняется;
- **права** — указывает тип доступа, применяемый `ACE`;
- **тип объекта** — идентификатор `GUID`, указывающий расширенное право, свойство или дочерний объект в зависимости от флагов маски доступа. Устанавливается со значением `0`, если не используется;
- **тип наследования** — тип класса объекта, который может наследовать `ACE` от этого объекта.

```
PS C:\Users\Administrator> $(Get-ADUser anakin -Properties nTSecurityDescriptor).nTSecurityDescriptor.Access[0]


ActiveDirectoryRights : GenericRead
InheritanceType       : None
ObjectType            : 00000000-0000-0000-0000-000000000000
InheritedObjectType   : 00000000-0000-0000-0000-000000000000
ObjectFlags           : None
AccessControlType     : Allow
IdentityReference     : NT AUTHORITY\SELF
IsInherited           : False
InheritanceFlags      : None
PropagationFlags      : None
```

ACE учетной записи пользователя

Таким образом, ACE можно использовать как для предоставления доступа, так и для его ограничения. Следует отметить, что в случае, если участнику разрешен доступ и запрещен доступ разными ACE, то ACE отказа имеет предпочтение, и доступ запрещен.

С другой стороны, ACE могут наследоваться от родительских объектов базы данных (`OU` и контейнеров), и на самом деле наследуются большинство ACE, которые применяются к объектам. В случае, если унаследованный доступ противоречит не унаследованному `ACE`, то не унаследованный `ACE` определяет правило доступа. Таким образом, приоритет для `ACE` следующий:

- Явный отказ `ACE`;
- Явное разрешение `ACE`;
- Унаследованный запрет `ACE`;
- Унаследованное разрешение `ACE`.

Существует особый случай, который не ограничен `ACE`, и это владелец объекта. Владелец имеет неявное разрешение на изменение `ACE` объекта (право `WriteDacl`).

Кроме того, необходимо также учитывать, что в случае, если дескриптор безопасности не имеет `DACL` (DACL установлен со значением NULL), любой доступ к объекту есть у всех. Однако если дескриптор безопасности имеет пустой список `DACL` (нет `ACE` в `DACL`), то никто не имеет доступа к объекту.

Права

В `ACE` могут быть указаны следующие права :

- **Delete** — удаление объекта;
- **ReadControl** — чтение дескриптора безопасности, кроме `SACL`;
- **WriteDacl** — изменение объекта `DACL` в дескрипторе безопасности;
- **WriteOwner** — изменение владельца объекта в дескрипторе безопасности;
- **CreateChild** — создание дочерних объектов для контейнеров;
- **DeleteChild** — удаление дочерний объект для контейнеров;
- **ListContents** — список дочерних объектов для контейнеров. Объект скрыт от пользователя, если это право или `ListObject` не предоставлены;

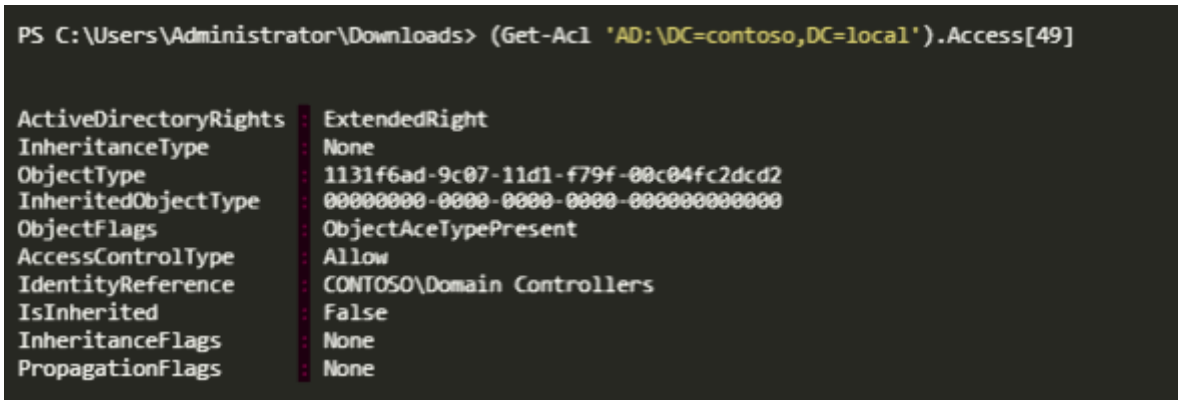
- **ReadProperty** — чтение свойства или набора свойств, указанного в типе объекта. Если тип объекта равен `0`, то все свойства могут быть прочитаны. Не позволяет читать конфиденциальные свойства;
- **WriteProperty** — изменение свойства, указанного в типе объекта. Если тип объекта равен `0`, то все свойства могут быть изменены;
- **WritePropertyExtended** — выполнение подтверждения записи. Возможно, наиболее интересной проверенной записью является самостоятельное членство для групп, которое позволяет добавить вашего текущего пользователя в группу с помощью `ACE`;
- **DeleteTree** — удаление все дочерние объекты с помощью операции удаления дерева;
- **ListObject** — список объектов. Объект скрыт от пользователя, если это право или `ListContents` не предоставлены;
- **ControlAccess** — специальное разрешение, которое можно интерпретировать по-разному в зависимости от типа объекта. Если тип объекта является `GUID` конфиденциального свойства, он дает разрешение на его чтение. Если `GUID` расширенного права зарегистрирован в схеме базы данных, то право предоставляется. Если тип объекта нулевой (`GUID` состоит из нулей), то предоставляются все расширенные права.

Существуют также некоторые общие права:

- **GenericRead** — ReadControl, ListContents, ReadProperty, ListObject;
- **GenericWrite** — ReadControl, WriteProperty, WritePropertyExtended;
- **GenericExecute** — ReadControl, ListContents;
- **GenericAll** — Delete, WriteDacl, WriteOwner, CreateChild, DeleteChild, DeleteTree, ControlAccess, GenericAll, GenericWrite.

Расширенных прав много, но вот одни из самых интересных:

- **User-Force-Change-Password** — изменение пароля пользователя, не зная текущего пароля для пользовательских объектов. Не путать с `User-Change-Password`, который требует знать пароль для его изменения;
- **DS-Replication-Get-Changes** — для репликации данных базы данных. Для объекта домена. Требуется для выполнения `dcsync`;
- **DS-Replication-Get-Changes-All** — для репликации секретных данных базы данных. Для объекта домена. Требуется для выполнения `dcsync`.



DS-Replication-Get-Changes-All непосредственно в домене

В предыдущем примере показан `ACE` `ExtendedRight`, который дает `DS-Replication-Get-Changes-All` для группы контроллеров домена.

Помимо объектов базы данных, на компьютерах с Windows также есть много защищаемых объектов, которые также защищены локальными `DACL`, управляемыми локальным компьютером. Среди этих объектов файлы/каталоги, процессы, ключи реестра или службы. Но поскольку `Domain Admins` по умолчанию добавляются в локальную группу Administrators на компьютерах, обычно администратор домена может получить доступ к любому локальному объекту на компьютере с Windows. Для проверки ACL файлов можно использовать такие инструменты, как Get-Acl, icacls.

ACL-атаки

Из-за огромного количества списков `ACL`, которые могут находиться в домене, управлять ими может быть сложно. Это может привести к нескольким неправильным конфигурациям, которые могут позволить злоумышленнику повысить привилегии в домене или даже в лесу (помните, что домены одного и того же леса связаны, поэтому вы можете добавить ссылку ACE на участников других доменов). Давайте рассмотрим некоторые неверные конфигурации:

- **Изменение пароля пользователя** — если у вас есть права `User-Force-Change-Password` или `GenericAll` на объект пользователя, вы можете взять на себя учетную запись, установив новый пароль;
- **Создание Kerberoasteable пользователя** — если вы можете написать имя участника-службы в свойстве `ServicePrincipalName` пользователя, вы можете выполнить атаку `Kerberoast` на эту учетную запись и попытаться взломать ее пароль. Для записи `SPN` требуется, чтобы у вас была проверенная запись `Validated-SPN` с `WritePropertyExtended`, `GenericWrite` или `GenericAll`;

- **Выполнение вредоносного сценария** — если вы можете изменить свойствоScriptPath пользователя с помощью WriteProperty, GenericWrite или GenericAll, то вы можете установить вредоносный файл, который будет выполняться при следующем входе пользователя в систему. Можно использовать путь UNC, чтобы указать на общий ресурс. Также может потребоваться включить флаг SCRIPT для свойства UserAccountControl;
- **Добавление пользователей в группу** — если вы можете изменить свойство участников группы с помощью WriteProperty, GenericWrite или GenericAll, вы можете добавить любого члена в группу;
- **Атака Kerberos RBCD** — если вы можете изменить учетную запись компьютера msDS-AllowedToActOnBehalfOfOtherIdentity с помощью WriteProperty, GenericWrite или GenericAll, то включится ограниченное делегирование на основе ресурсов Kerberos для другого пользователя к компьютерным службам и, наконец, будет получен доступ к компьютеру в качестве администратора;
- **Пароль LAPS** — если вы можете прочитать конфиденциальное свойство компьютера ms-Mcs-AdmPwd, используемое LAPS для хранения пароля локального администратора машины, то вы можете использовать его как доступ к машине в качестве локального администратора. Вы можете определить использование LAPS на компьютере, проверив, существует ли свойство ms-Mcs-AdmPwdExpirationTime в его учетной записи компьютера;
- **Атака DCSync** — если у вас есть расширенные права DS-Replication-Get-Changes и DS-Replication-Get-Changes-All на объект домена, то вы можете выполнить атаку DCSync, чтобы сбросить содержимое базы данных;
- **Злоупотребление объектом групповой политики** — если вы можете изменить контейнер групповой политики GPC-File-Sys-Path с помощью WriteProperty, GenericWrite или GenericAll, вы можете изменить объект групповой политики и выполнить код на компьютерах, затронутых объектом групповой политики;
- **Изменение ACL** — Если у вас есть права WriteDacl или GenericAll, то вы можете создать ACE, чтобы дать любое право на объект и выполнить некоторые из предыдущих атак. Кроме того, если у вас есть право WriteOwner, поскольку объект-владелец имеет неявное право WriteDacl, вы можете изменить владельца объекта на своего пользователя, а затем изменить ACL.

Помимо возможности повышения привилегий, списки ACL также могут быть весьма полезными и незаметными, если вы хотите создать лазейки, для сохранения собственного доступа в сети. Для создания бэкдоров есть несколько приемов сокрытия вредоносных ACE, описанных в официальном документе An ACE Up the Sleeve, написанном командой Specterops.

AdminSDHolder

`AdminSDHolder` — это специальный объект в базе данных, список `DACL` которого используется в качестве шаблона для дескриптора безопасности привилегированных субъектов.

```
PS C:\> Get-ADObject 'CN=AdminSDHolder,CN=system,DC=contoso,DC=local'

DistinguishedName                           Name          ObjectClass ObjectGUID
-----------------                           ----          ----------- ----------
CN=AdminSDHolder,CN=system,DC=contoso,DC=local AdminSDHolder container   7f34e8a5-ffbd-474a-b436-1e02b7b49984
```

Объект AdminSDHolder

Каждые 60 минут `SDProp` (распространитель дескрипторов безопасности) проверяет дескриптор безопасности этих привилегированных субъектов и заменяет их `DACL` копией `DACL AdminSDHolder` (если они отличаются). Это делается для предотвращения изменений в списках DACL этих субъектов, но если вы можете добавить пользовательские элементы управления доступом в список DACL AdminSDHolder, то эти новые элементы управления доступом также будут применяться к защищенным субъектам.

По умолчанию «защищены» следующие участники `AdminSDHolder`:

- Операторы счетов;
- Администратор;
- Администраторы;
- Операторы резервного копирования;
- Администраторы домена;
- Контроллеры домена;
- Гости домена;
- Администраторы предприятия;
- Ключевые администраторы предприятия;
- Корпоративные контроллеры домена только для чтения;
- Ключевые администраторы;
- krbtgt;
- Операторы печати;
- Контроллеры домена только для чтения;
- Репликатор;
- Администраторы схемы;
- Операторы сервера.

Привилегии

Если вы знакомы с платформой Windows, возможно, вы знаете о привилегиях пользователей, которые позволяют пользователям выполнять действия в обход ACL объектов. Например, `SeDebugPrivilege` на компьютере с Windows позволяет читать/записывать любую память процесса на компьютере, даже если у вас нет прав. В Active Directory также можно злоупотреблять некоторыми привилегиями (в основном в контроллерах домена):

- **SeEnableDelegationPrivilege**
  `SeEnableDelegationPrivilege` должен быть установлен в контроллере домена для пользователя (это локальная привилегия), а затем он позволяет изменять свойство `msDS-AllowedToDelegateTo` пользователей и флаги `TRUSTED_FOR_DELEGATION` и `TRUSTED_TO_AUTH_FOR_DELEGATION` из свойства `UserAccountControl`. Другими словами, `SeEnableDelegationPrivilege` позволяет управлять параметрами Kerberos `Unconstrained` и `Constrained Delegation` в домене, которые злоумышленник может использовать для повышения привилегий. По умолчанию предоставляется только учетной записи администратора.
- **SeBackupPrivilege**
  Привилегия резервного копирования позволяет читать любой файл контроллера домена, чтобы сделать его резервную копию, которую можно использовать для чтения базы данных домена. По умолчанию предоставляется группам `Backup Operators`, `Server Operators` и `Administrators`. Эта привилегия действует только при использовании API резервного копирования NTFS, доступ к которому можно получить с помощью утилиты wbadmin или Powershell WindowsServerBackup (оба требуют функции резервного копирования Windows Server). Вы также можете использовать `reg save` для доступа к секретам SAM и LSA.
- **SeRestorePrivilege**
  Привилегия восстановления позволяет записывать любой файл на контроллере домена из резервной копии. Это может позволить злоумышленнику изменить базу данных домена. По умолчанию предоставляется группам `Backup Operators`, `Server Operators` и `Administrators`. Вы можете использовать эту привилегию для изменения ключей реестра и обеспечения привилегированного выполнения команд.
- **SeTakeOwnershipPrivilege**
  С привилегией владения вы можете стать владельцем защищаемых объектов машины, таких как файлы, процессы или ключи реестра. Владелец объекта всегда может изменить разрешения объекта. Например, можно использовать вызов API `SetNamedSecurityInfo`, чтобы стать владельцем объекта.

Помимо привилегий, используемых в домене, также полезно знать об опасных привилегиях, которые могут быть полезны для повышения привилегий на компьютере с Windows. Обычно используются следующие:

- **SeDebugPrivilege**
  В режиме отладки для любого процесса на компьютере пользователь может вводить код в любой процесс, что может привести к повышению привилегий, или читать память процесса, что позволяет читать, например, секреты процесса lsass пользователей, вошедших в систему.

- **SeImpersonatePrivilege**
  Пользователь может получить токены безопасности других пользователей на компьютере. Если уровень представления — `SecurityDelegation`, то пользователь может представиться целевым пользователем на других компьютерах домена (маркеры `SecurityDelegation` связаны с учетными данными пользователя, такими как билеты Kerberos, которые можно использовать в сетевых подключениях). Если уровень маркера — `SecurityImpersonation`, то целевой пользователь может представляться только на локальном компьютере (полезно для повышения привилегий). `SeImpersonatePrivilege` предоставляется «`NT AUTHORITY\Network Service`», который обычно используется для запуска веб-серверов и тому подобного, поэтому, если вы можете скомпрометировать веб-сервер, возможно, вы можете выдавать себя за какого-то пользователя домена в сети. Но определенно, если вы хотите повысить привилегии с помощью `SeImpersonatePrivilege` на локальном компьютере, используйте potato.

Существуют и другие привилегии, которые можно использовать для повышения привилегий на компьютерах с Windows. Если они вас интересуют, вам следует проверить репозиторий token-priv от FoxGlove, который включает документ с их описанием и PoC для их использования, настоятельно рекомендуемый ресурс.

Групповая политика

Целью Active Directory является управление компьютерами и пользователями организации. И часть процесса управления осуществляется групповой политикой.

Групповая политика — это механизм, который позволяет применять набор правил/действий к пользователям и компьютерам сети Active Directory. Вот некоторые из возможностей:

- Отключить NTLM;
- Требование к созданию сложного пароля;
- Выполнение запланированных/немедленных задач;
- Создание локальных пользователей на компьютерах;
- Установка обоев по умолчанию;
- Синхронизация файлов с OneDrive и т.д.

Для определения правил создаются объекты групповой политики (GPO). Каждый объект групповой политики определяет ряд политик, которые можно применять к определенным компьютерам доменов. Кроме того, вы можете создавать политики, которые применяются ко всему компьютеру или сеансам пользователей. Например, вы можете выполнить сценарий при запуске компьютера или при входе пользователя в систему.

Область действия объекта групповой политики

При создании объекта групповой политики необходимо указать, к каким компьютерам он будет применяться. Для этого вам необходимо связать объект групповой политики с одним из следующих контейнеров базы данных:

- Домен;
- Организационная единица;
- Сайт (контейнер для группы компьютеров, которые физически расположены близко друг к другу, не рекомендуется для объектов групповой политики).

На компьютере с Windows также может быть локальная групповая политика. Таким образом, к машине на разных уровнях можно применить множество различных объектов групповой политики, которые обрабатываются в следующем порядке:

- Локально;
- Сайт;
- Домен;
- Организационная единица.

Здесь локальные GPO имеют наименьшее предпочтение, а GPO организационной единицы — наиболее предпочтительные.

Однако для объектов групповой политики Active Directory (не локальных) также возможно установить правило `No Override`. Таким образом, если установлено правило политики домена, никакие правила из подразделений не могут противоречить этому вышестоящему правилу.

Кроме того, с объектом групповой политики может быть связан запрос `WMI`, что позволяет отфильтровать компьютер, к которому будет применяться объект групповой политики. Например, чтобы применить политики только к компьютерам с Windows 7.

В домене каждый компьютер проверяет наличие обновлений политики каждые 90 минут, за исключением контроллеров домена, которые делают это каждые 5 минут. Вы также можете выполнить немедленную проверку с помощью `gpupdate`.

Каждый объект групповой политики идентифицируется идентификатором `GUID` и состоит из двух объектов: шаблона групповой политики и контейнера групповой политики.

Шаблон групповой политики

Шаблон групповой политики — это каталог в общей папке `SYSVOL`. Шаблоны могут находиться в `\\SYSVOL\Policies\`. Каждому каталогу шаблона присваивается имя с использованием `GUID` объекта групповой политики.

```
PS C:\> ls \\contoso.local\SYSVOL\contoso.local\Policies\

    Directory: \\contoso.local\SYSVOL\contoso.local\Policies


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        11/28/2020  10:02 AM                {31B2F340-016D-11D2-945F-00C04FB984F9}
d-----        11/28/2020  10:02 AM                {6AC1786C-016F-11D2-945F-00C04FB984F9}
d-----         4/19/2021   5:12 PM                {BE864EFE-6C07-4A53-A9D8-7EB6EB36BE5A}
```

Список шаблонов GP

Каждый каталог GPO содержит следующие элементы:

- **Каталог компьютера** — для политик на уровне компьютера;
- **Каталог пользователя** — для политик уровня пользователя;
- **GPT.INI** — основная информация о GPO, версия и отображаемое имя.

В этих каталогах могут быть самые разные файлы и каталоги, в которых можно найти INI-файлы конфигурации, в которых указаны значения ключей реестра, членов групп или сценариев для выполнения. И, если повезет, учетные данные в сценариях или файлах предпочтений групповой политики (GPP) с тегами `cpassword`. Для поиска учетных данных GPP можно использовать сценарий Get-GPPPasword.

Контейнер групповой политики

Чтобы компьютеры могли находить шаблоны групповой политики, в базе данных Active Directory хранится информация об объектах групповой политики в контейнере `CN=Policies,CN=System,DC=,DC=`. Каждый объект групповой политики хранится в объекте `GroupPolicyContainer`, содержащем объект групповой политики `GUID` и путь к шаблону групповой политики.

```
PS C:\> ls \\contoso.local\SYSVOL\contoso.local\Policies\

    Directory: \\contoso.local\SYSVOL\contoso.local\Policies


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        11/28/2020  10:02 AM                {31B2F340-016D-11D2-945F-00C04FB984F9}
d-----        11/28/2020  10:02 AM                {6AC1786C-016F-11D2-945F-00C04FB984F9}
d-----         4/19/2021   5:12 PM                {BE864EFE-6C07-4A53-A9D8-7EB6EB36BE5A}
```

Получение списка объектов групповой политики домена

GUID GPO отличается от GUID, используемого для идентификации каждого объекта в базе данных Active Directory. Также стоит обратить внимание, что если доступно изменение свойства объекта групповой политики `GPCFileSysPath`, то можно

установить контролируемый атакующим путь и создать вредоносный объект групповой политики, который может содержать вредоносные сценарии, которые будут выполняться на нескольких компьютерах.

С другой стороны, объекты базы данных домена, подразделений и сайтов связаны с объектами групповой политики с помощью свойства `GpLink`.

```
PS C:\> Get-ADObject -LDAPFilter '(gPLink=*)' -Properties CanonicalName,gpLink | select objectclass,CanonicalName,gplink | Format-List

objectclass     : domainDNS
CanonicalName   : contoso.local/
gplink          : [LDAP://cn={BE864EFE-6C07-4A53-A9D8-7EB6EB36BE5A},cn=policies,cn=system,DC=contoso,DC=local;1][LDAP://C
                  N={31B2F340-016D-11D2-945F-00C04FB984F9},CN=Policies,CN=System,DC=contoso,DC=local;0]

objectclass     : organizationalUnit
CanonicalName   : contoso.local/Domain Controllers
gplink          : [LDAP://CN={6AC1786C-016F-11D2-945F-00C04fB984F9},CN=Policies,CN=System,DC=contoso,DC=local;0]

objectclass     : organizationalUnit
CanonicalName   : contoso.local/web servers
gplink          : [LDAP://cn={BE864EFE-6C07-4A53-A9D8-7EB6EB36BE5A},cn=policies,cn=system,DC=contoso,DC=local;0]
```

Список доменов и подразделений со связанными объектами групповой политики

```
PS C:\> Get-ADObject -LDAPFilter '(gPLink=*)' -SearchBase "CN=Configuration,$((Get-ADDomain).DistinguishedName)" -Properties CanonicalName,gpLink

objectclass     : site
CanonicalName   : contoso.local/Configuration/Sites/mysite
gplink          : [LDAP://cn={BE864EFE-6C07-4A53-A9D8-7EB6EB368E5A},cn=policies,cn=system,DC=contoso,DC=local;0]
```

Список сайтов со связанными объектами групповой политики

Компьютер может определить объекты групповой политики, которые применяются к нему самому, изучив объекты `OU`, к которым он принадлежит, и объект домена.

Например, компьютер, на котором находится объект `CN=mypc,OU=workstations,OU=computers,DC=domain,DC=com`, будет применять объекты групповой политики рабочих станций (`workstations`) и компьютерных подразделений (`OU`) компьютера и домена `domain.com`.

Протоколы связи

В Active Directory существует множество протоколов, которые используются для связи между компьютерами. Их можно использовать для перемещения по сети и выполнения команд в разных компьютерах в средах, поэтому важно знать об их назначении и возможностях, которые они предлагают.

SMB

`SMB` (Server Message Block) — это протокол, широко используемый в сетях Active Directory (и любой другой сети Windows) для обмена файлами и обмена данными между компьютерами, как правило Windows.

Каждый компьютер Windows по умолчанию разрешает подключение к ней с использованием протокола SMB. Первоначально SMB работал через NetBIOS (службы сеансов), но в настоящее время его можно использовать непосредственно

через TCP. На компьютерах с Windows порт `445/TCP` открыт для обработки соединений SMB.

SMB и связанные протоколы/порты

Атакующему полезно знать об SMB, поскольку он используется для создания общих ресурсов, которые могут содержать ценную информацию и могут использоваться для извлечения информации с компьютеров.

Общие ресурсы

Общие ресурсы похожи на папки, которые компьютер использует для доступа к другим компьютерам/пользователям в сети. Вы можете получить список общих ресурсов с помощью команды `net view`, командлета Powershell `Get-SmbShare` или инструмента `smbclient`.

```
C:\> net view \\dc01.contoso.local /all
Shared resources at \\dc01.contoso.local

Share name  Type  Used as  Comment

-------------------------------------------------------------------------------
ADMIN$      Disk           Remote Admin
C$          Disk           Default share
IPC$        IPC            Remote IPC
NETLOGON    Disk           Logon server share
SYSVOL      Disk           Logon server share
The command completed successfully.
```

Общие ресурсы домена DC

Получить доступ к общим ресурсам других компьютеров можно таким же образом, как и к папке на локальном компьютере. Для доступа к общему ресурсу можно использовать путь `UNC`, например, `\\dc01.contoso.local\SYSVOL\` или сопоставить удаленный общий ресурс с локальным устройством с помощью команды `net use`.

Чтобы обратиться к целевому компьютеру в пути `UNC`, можно использовать его DNS-имя или имя NetBIOS. Например: `net view \\dc01.contoso.local` или `net view \\dc01`.

```
C:\> dir \\dc01\sysvol
 Volume in drive \\dc01\sysvol has no label.
 Volume Serial Number is 609D-528B

 Directory of \\dc01\sysvol

28/11/2020  11:02    <DIR>          .
28/11/2020  11:02    <DIR>          ..
28/11/2020  11:02    <JUNCTION>     contoso.local [C:\Windows\SYSVOL\domain]
               0 File(s)              0 bytes
               3 Dir(s)  20,050,214,912 bytes free
```

Список папок внутри общей папки

Общие ресурсы очень полезны для пользователей,позволяя получить доступ к файлам других компьютеров, не беспокоясь об использовании специальной программы или чего-то подобного. Следовательно, они также очень удобны для атакующих при перемещении файлов с одного компьютера на другой с целью их эксфильтрации.

Создание общего доступа, к которому каждый может получить доступ

```
net share Temp=C:\Temp /grant:everyone,FULL
```

Общие ресурсы по умолчанию

Некоторые ресурсы заканчиваются на `$`. Это общие ресурсы: `C$`, `ADMIN$` и `IPC$`, они присутствуют по умолчанию на любом компьютере с Windows.

Для доступа к `C$` и `ADMIN$` необходимо иметь права администратора на целевом компьютере. С помощью этих общих ресурсов (особенно `C$`) можно просматривать все компьютерные файлы. На самом деле, эти ресуррсы используются несколькими инструментами. Например, `PsExec` использует `ADMIN$` для развертывания двоичных файлов, отвечающих за выполнение данной команды.

Общий ресурс `IPC$` — это специальный общий ресурс, используемый для создания именованных каналов.

Общий доступ к домену по умолчанию

Помимо общих общих ресурсов, контроллеры домена также публикуют общие ресурсы `SYSVOL` и `NETLOGON` ресурсы, доступные для любого пользователя/компьютера в домене. Они используются для хранения файлов, к которым должны обращаться все компьютеры (по крайней мере, Windows) домена.

Общий ресурс `SYSVOL` обычно используется для хранения шаблонов групповой политики, используемых компьютерами для чтения групповых политик, развернутых в домене. Иногда эти политики содержат пароли. К общему ресурсу SYSVOL можно

получить доступ с `\\\SYSVOL` используя `UNC`.

```
PS C:\> dir \\contoso.local\SYSVOL\contoso.local

    Directory: \\contoso.local\SYSVOL\contoso.local


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        19/04/2021     17:12                Policies
d-----        28/11/2020     10:02                scripts
```

Список папок SYSVOL

`\\\\SYSVOL\\scripts` является псевдонимом для общего `NETLOGON` ресурса. Общий ресурс `NETLOGON` используется для хранения сценариев входа, которые необходимо выполнить для компьютеров домена.

Именные каналы

Общий ресурс `IPC$` не является каталогом, но он используется для создания именных каналов, которые позволяют процессам разных компьютеров взаимодействовать между собой с помощью таких механизмов, как `RPC` (удаленный вызов процедур).

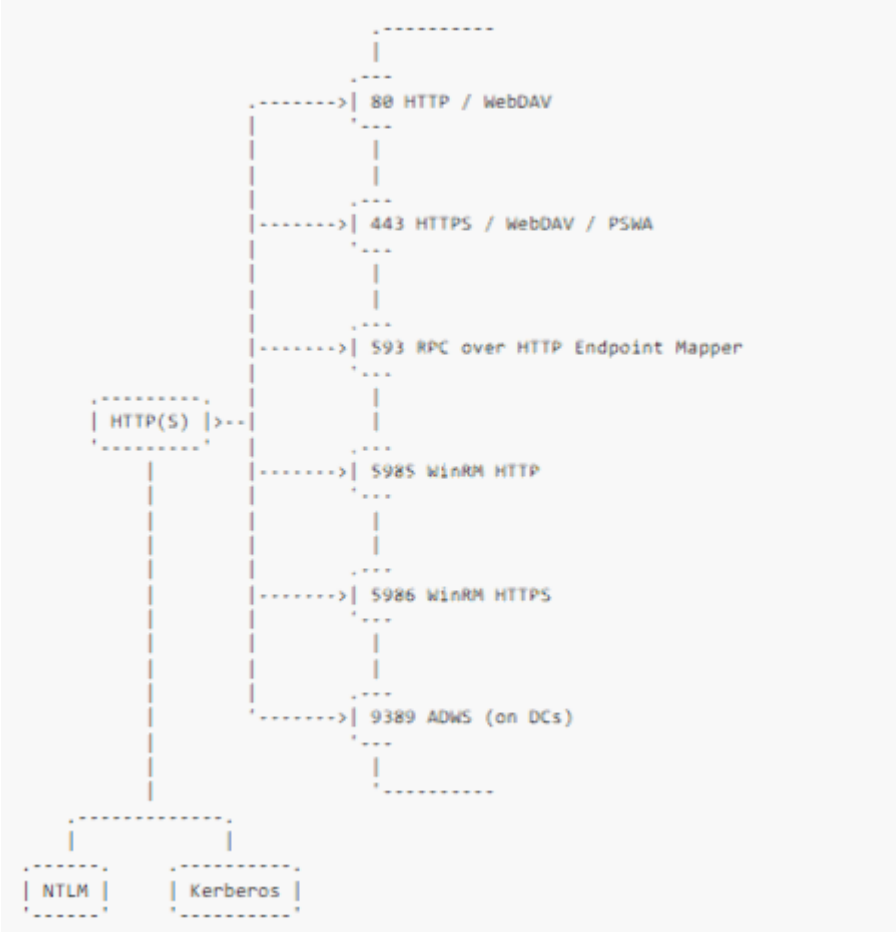
Именные каналы можно рассматривать как порты TCP, которые позволяют комапьютерам обмениваться данными между собой, но внутри протокола `SMB`. Они используются для вызовов `RPC`, что позволяет множеству протоколов обмениваться данными через `SMB`.

Обычно протоколы, работающие со стеком `RPC/SMB`, определяют известный именной канал, который можно использовать для связи с удаленной службой (та же идея, что и с портами TCP/UDP). Например, `RPC` использует именной канал `\pipe\netlogon` для обмена сообщениями протокола `Netlogon`.

HTTP

`HTTP` (протокол передачи гипертекста), вероятно, является самым известным протоколом приложений, поскольку это протокол Интернета. Но помимо своей основной роли в Интернете, также широко используется в Active Directory.

HTTP используется в качестве транспортного протокола многими другими протоколами приложений, присутствующими в домене Active Directory, такими как `WinRM` (и, следовательно, `Powershell Remoting`), `RPC` или `ADWS` (веб-службы Active Directory).

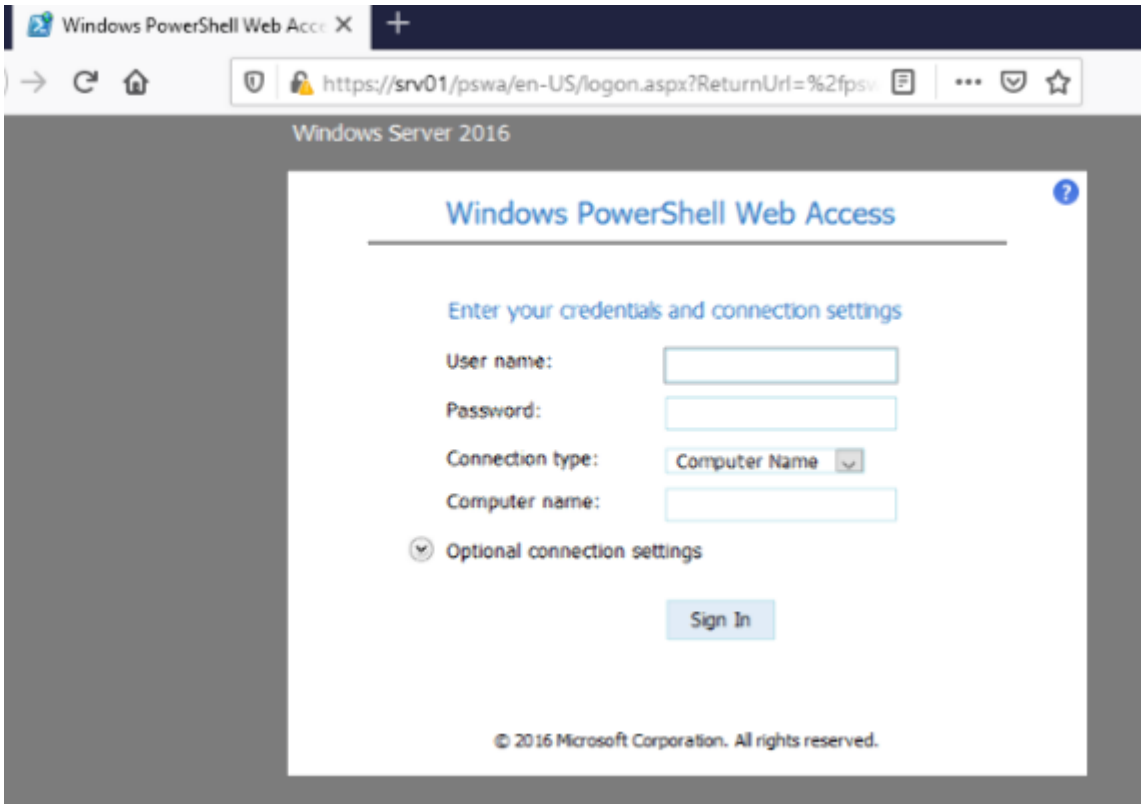


Порты, используемые службами HTTP в Active Directory

Для полной интеграции с Active Directory HTTP поддерживает аутентификацию как с помощью NTLM, так и с помощью Kerberos. Это важно с точки зрения безопасности, поскольку подразумевается, что HTTP-соединения подвержены атакам делегирования Kerberos или ретрансляции NTLM.

В случае ретрансляции NTLM особенно важно отметить, что HTTP-соединения не требуют подписи, поэтому они очень уязвимы для атак перекрестной ретрансляции NTLM. На самом деле, существует множество атак, таких как `PrivExchange` или захват компьютера `Kerberos RBCD`, которые полагаются на ретрансляцию NTLM с HTTP на LDAP. Если заставить компьютер выполнить HTTP-запрос, используя учетную запись домена компьютера с аутентификацией NTLM, то можно скомпрометировать компьютер с помощью Kerberos RBCD.

Что касается HTTP, на компьютерах с Windows можно установить веб-сервер `IIS`, который является основой для некоторых технологий, таких как `WebDAV` или `PSWA` (веб-доступ Powershell), которые можно включить `/pswa` в конечной точке.



Логин PSWA

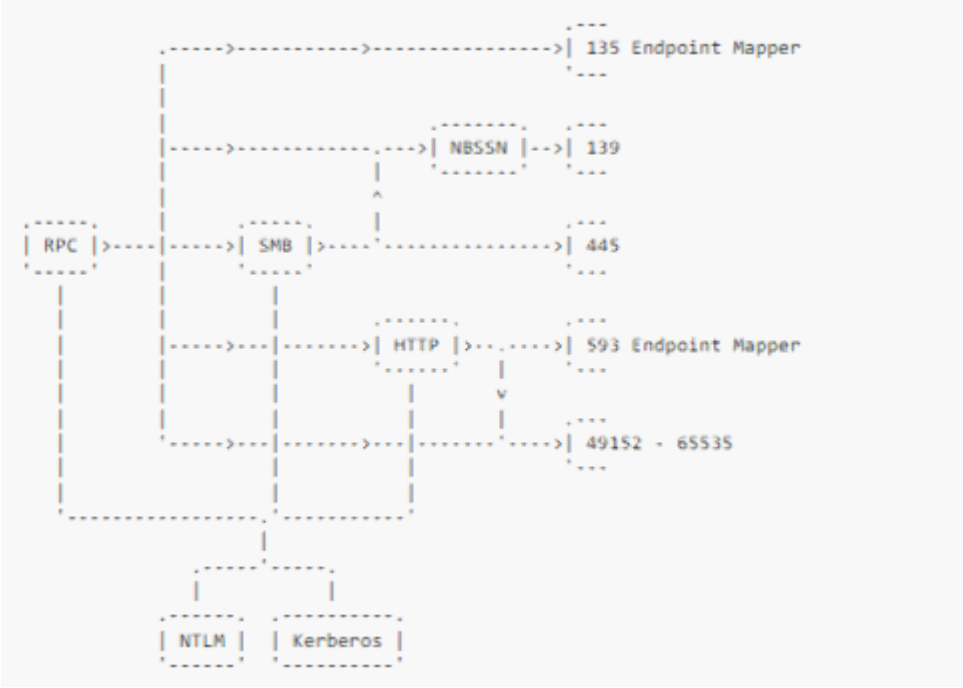
Кроме того, можно создать прокси-сервер `SOCKS` через HTTP в установке IIS с помощью pivotnacci.

RPC

`RPC` (удаленный вызов процедур) — это протокол, который позволяет программам с разных компьютеров взаимодействовать между собой, вызывая функции по сети. Microsoft разработала протокол `RPC` под названием `MSRPC`, который представляет собой модифицированную версию `DCE/RPC` с некоторыми расширениями (определенными в RPCE).

MSRPC может использовать разные транспортные протоколы, например:

- **TCP**, используя порт `135` для Endpoint Mapper и порты с `49152` по `65535` в качестве конечных точек;
- **SMB** с использованием именных каналов;
- **NetBIOS**;
- **HTTP**, используя порт `593` для Endpoint Mapper и порты с `49152` по `65535` в качестве конечных точек.



Протоколы и порты, связанные с RPC

MSRPC постоянно используется в домене компьютерами для связи между ними. Компьютеры Windows используют MSRPC для множества различных задач, таких как управление службами или чтение реестра других компьютеров.

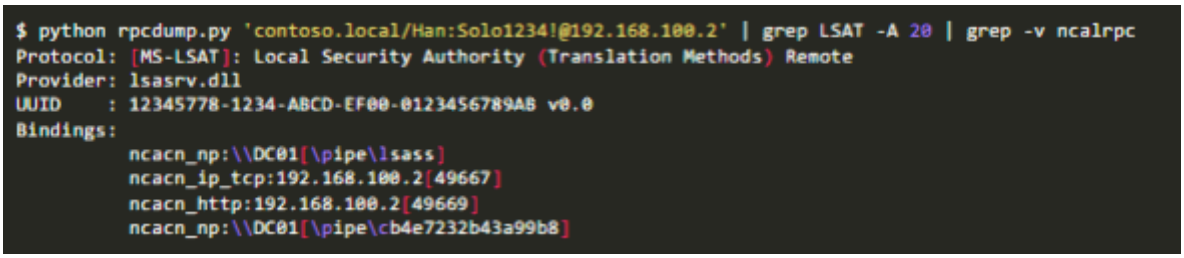
RPC также широко используется для связи программ на локальном компьютере через LRPC (локальный RPC) или ALPC (расширенный вызов локальных процедур).

Для выполнения всех этих задач Microsoft определила несколько интерфейсов MSRPC, которые определяют различные функции, что позволяет запрашивать/вызывать различные службы компьютера из удаленной программы.

Каждый интерфейс идентифицируется `UUID` (универсальным уникальным идентификатором), например `12345778-1234-ABCD-EF00-0123456789AB`, и для каждого интерфейса используются разные конечные точки. Некоторые интерфейсы имеют предопределенные конечные точки, например именные каналы. Например, диспетчер управления службами (`SCMR`) использует именной канал `\PIPE\svcctl`.

Однако для других интерфейсов удаленная конечная точка изменяется, поэтому для ее определения клиент RPC должен связаться с Endpoint Mapper (EPM) для разрешения удаленной конечной точки из `GUID`.

В зависимости от интерфейса могут использоваться разные транспортные протоколы. Вы можете использовать утилиты impacket rpcdump.py и rpcmap.py для обнаружения конечных точек RPC (и их протоколов), которые можно использовать для подключения к данной службе на удаленном компьютере. Кроме того, можно исследовать конечные точки RPC на локальном компьютере с помощью RpcView.



Список удаленных конечных точек интерфейса LSAT

Чтобы иметь представление о том, что можно сделать с помощью RPC, вот описания некоторых из наиболее часто используемых интерфейсов. Интерфейсы разделены по транспортным протоколам для понимания, когда разные порты на компьютьере открыты.

RPC через SMB

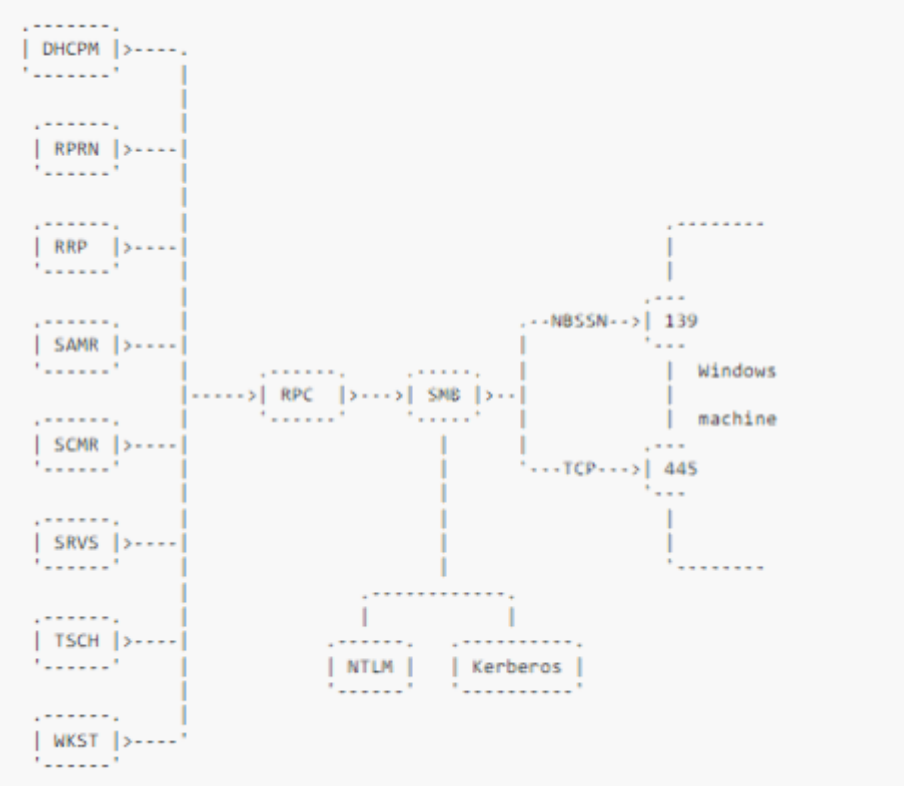
Следующие интерфейсы/протоколы RPC могут (и они обычно) использоваться через SMB:

DHCPM
DHCPM (DHCP Server Management) используется для управления конфигурацией DHCP-сервера.
RPRN
RPRN (Print System Remote) используется для управления печатью с удаленного компьютера. Можно использовать SpoolSample или printerbug.py, чтобы вызвать ошибку принтера через RPRN.
RRP
RRP (протокол удаленного реестра Windows) позволяет читать и изменять ключи реестра с удаленного компьютера. Можно использовать reg (если выводится ошибка «Сетевой путь не найден», вам нужно запустить службу «Удаленный реестр» на удаленной машине) или reg.py (это автоматически запускает службу «Удаленный реестр» с SRVS).) для управления удаленным реестром.
SAMR
SAMR (SAM Remote) позволяет подключать SAM (Security Account Manager) других компьютеров для управления пользователями и группами. Можно использовать samrdump.py для получения информации о локальных пользователях машины.
SCMR
SCMR (SCM Remote) используется для подключения к SCM (диспетчеру управления службами) других компьютеров для управления службами. Протокол, используемый утилитой PsExec для выполнения команд на удаленных компьютерах.
SRVS
Через SRVS (Server Service Remote) можно подключиться к удаленному компьютеру для управления подключениями, сеансами, общими ресурсами, файлами и транспортными протоколами. Можно использовать netview.py для перечисления сеансов или net view для перечисления общих ресурсов на удаленных машинах.
TSCH

TSCH (Task Scheduler Service Remote) используется для управления задачами на удаленных компьютерах. Можно использовать atexec.py, at или schtasks для создания удаленных задач.

WKST

WKST (Workstation Service Remote) используется для управления/запроса некоторых параметров рабочей станции, таких как имя хоста, версия ОС, пользовательские сеансы или домен компьютера. Можно использовать WKST с netview.py для перечисления сеансов.



Протоколы RPC, работающие через SMB

Кроме того, существуют некоторые интерфейсы RPC, предназначенные для использования в домене для запроса контроллеров домена:

BKRP

BKRP (удаленный протокол BackupKey) используется для передачи ключей `DPAPI` в домене Active Directory. Вы можете использовать `mimikatz lsadump::backupkeys` или dpapi.py backupkeys для получения резервных ключей `DPAPI` с контроллера домена.

LSAD

LSAD (политика домена LSA) — это удаленный интерфейс для `LSA` (локальный орган безопасности) для управления пользователями, доверием и другими вещами, связанными с безопасностью. Используется вместе с `LSAT`.

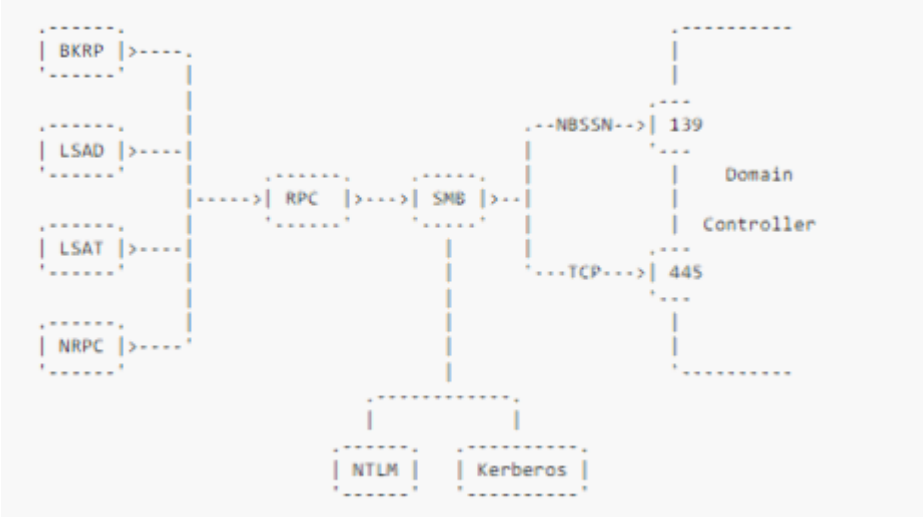
LSAT

LSAT (методы трансляции LSA) позволяет преобразовывать `SID` в основные имена. Используется вместе с `LSAD`. Вы можете использовать lookupsid.py для

перечисления пользователей на основе `SID`.

NRPC

NRPC (протокол удаленного входа в систему) используется в доменах, чтобы позволить компьютерам аутентифицировать пользователей, запрашивая контроллер домена. Также используется между контроллерами домена разных доменов для аутентификации пользователей разных доменов с помощью `NTLM`. Кроме того, он позволяет получать такую информацию, как: информация о пользователях, доверительные отношения домена или список контроллеров домена. Можно использовать nltest для выполнения нескольких запросов. Этот протокол также известен уязвимостью `Zerologon`.



Протоколы RPC, работающие через SMB (контроллер домена)

RPC через TCP

Кроме того, есть некоторые интерфейсы RPC, которые нельзя использовать через SMB, но можно использовать их напрямую через TCP:

DRSR

DRSR (Directory Replication Service Remote) — это протокол, используемый контроллерами домена для репликации данных. Его также можно использовать для злоумышленника с достаточными привилегиями для репликации учетных данных пользователей домена, выполнив атаку `dcsync` с помощью `mimikatz lsadump::dcsync` или impacket secretsdump.py.

DCOM

DCOM (Distributed COM) используется для взаимодействия с объектами `COM` (Component Object Model) удаленных компьютеров. COM-объекты очень полезны и могут использоваться для многих целей, например, для выполнения команд, которые можно выполнить с помощью dcomexec.py.

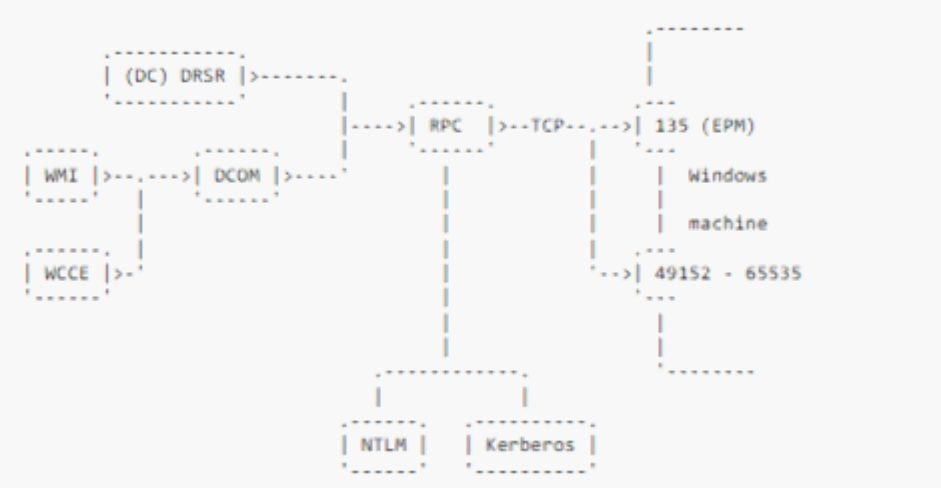
WMI

WMI (Windows Management Instrumentation Remote) — это реализация `Microsoft CIM` (Common Information Model), построенная на основе COM-объектов, которая позволяет запрашивать и управлять различными частями компьютера Windows из

единого интерфейса. Является очень универсальным и может использоваться с `wmic`, командлетами `Powershell`, такими как `Get-WmiObject`, или сценариями WMI impacket, такими как wmiexec.py.

WCCE

WCCE (протокол регистрации клиентских сертификатов Windows) — это интерфейс `DCOM`, который позволяет пользователям запрашивать сертификаты и другие службы, связанные с ЦС в `ADCS`. Его можно использовать с certreq или Certify.



Протоколы RPC, работающие через TCP

WinRM

Помимо RPC, также можно использовать WinRM (удаленное управление Windows) для связи и выполнения операций на других компьютерах. WinRM — это реализация Microsoft спецификации `WS-Management` (Web Services-Management), которая определяет протокол для управления компьютерами с использованием SOAP через HTTP.

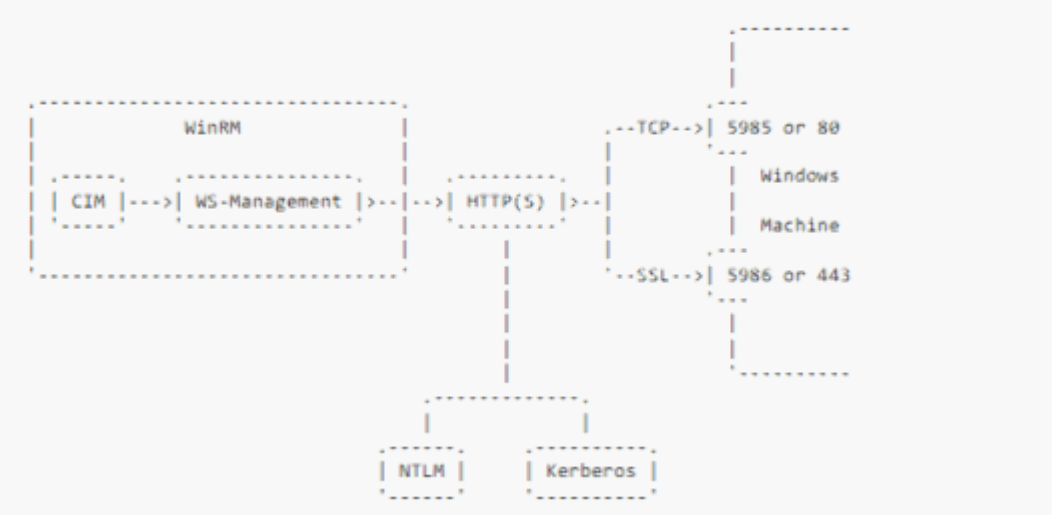
WinRM использует некоторые расширения, определенные в `WSMAN` и `WSMV`, для доступа к объектам `CIM` на удаленных компьютерах. Эти объекты `CIM` похожи на обновление объектов `WMI`. Можно получить доступ к объектам `CIM` на локальных и удаленных компьютерах с помощью командлетов CIM, таких как `Get-CimInstance`. Кроме того, можно использовать `winrs` для выполнения действий на удаленных компьютерах с помощью WinRM.

```
PS C:\> Get-CimInstance CIM_OperatingSystem -ComputerName dc01 | Format-List

SystemDirectory : C:\Windows\system32
Organization    :
BuildNumber     : 17763
RegisteredUser  : Windows User
SerialNumber    : 00431-10000-00000-AA522
Version         : 10.0.17763
PSComputerName  : dc01
```

Использование CIM для получения информации с удаленного компьютера

По умолчанию служба WinRM прослушивает порт `5985` для соединений HTTP и порт `5986` для соединений HTTPS. По умолчанию используется HTTP, поскольку сообщения WinRM шифруются на верхнем уровне. Однако WinRM можно настроить на использование обычных HTTP-портов `80` и `443` для соединений HTTP и HTTPS соответственно.



Стек протоколов WinRM

Удаленное взаимодействие Powershell

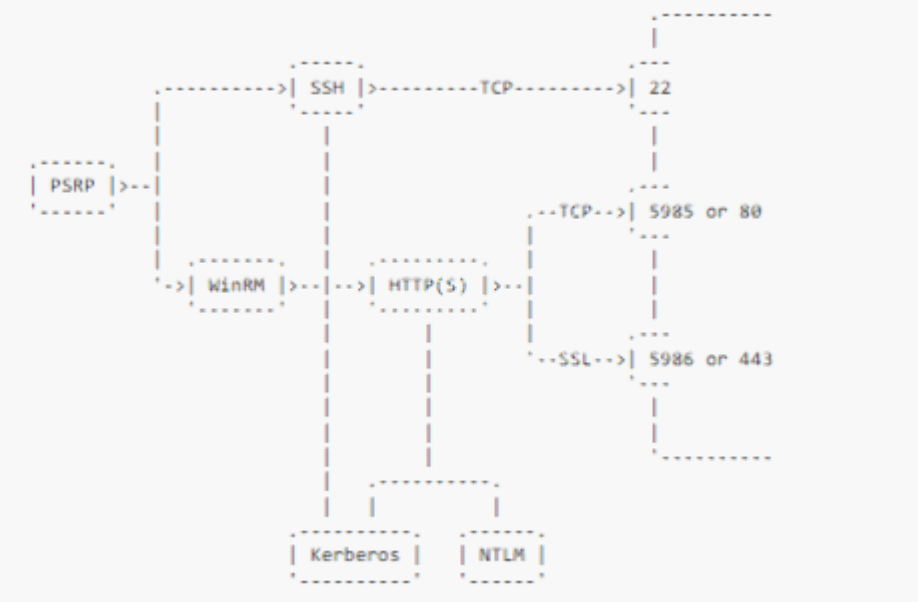
Одной из замечательных утилит для управления системами является удаленное взаимодействие Powershell, которая позволяет клиенту устанавливать сеанс Powershell на удаленных компьютерах и выполнять все виды задач. Удаленное взаимодействие Powershell включено по умолчанию в серверных версиях Windows (а не в клиентских, как в Windows 10), начиная с Windows Server 2012 R2.

```
PS C:\> $pw = ConvertTo-SecureString -AsPlainText -Force -String "Admin1234!"
PS C:\> $cred = New-Object -typename System.Management.Automation.PSCredential -argumentlist "contoso\Administrator",$pw
PS C:\>
PS C:\> $session = New-PSSession -ComputerName dc01 -Credential $cred
PS C:\> Invoke-Command -Session $session -ScriptBlock {hostname}
dc01
PS C:\> Enter-PSSession -Session $session
[dc01]: PS C:\Users\Administrator\Documents>
```

Удаленный сеанс PowerShell с учетными данными в открытом виде

Первоначально удаленное взаимодействие Powershell было построено поверх протокола WinRM. Однако предполагалось, что он будет использоваться на компьютерах с Linux, поэтому он также поддерживает SSH в качестве транспортного протокола.

Также возможно использовать Powershell через веб-браузер, если включен Powershell Web Access (PSWA).



Стек протоколов удаленного взаимодействия Powershell

Чтобы использовать удаленное взаимодействие Powershell, можно использовать несколько `PSSession CmdLet` для выполнения команд на удаленных компьютерах. Также из Linux можно установить Powershell или использовать evil-winrm.

Будьте осторожны при проведении пентеста, так как Powershell имеет множество функций ведения журналов.

Доверенные хосты

По умолчанию удаленное взаимодействие Powershell позволяет подключаться ко всем компьютерам в домене с помощью Kerberos. Однако, если необходимо подключить компьютер из другого домена, то нужно добавить этот IP-адрес к значению `TrustedHost` (или использовать «`*`»). В этом случае нужно настроить `TrustedHost` на клиенте, а не на сервере.

```
PS C:\> Set-Item wsman localhost\client\TrustedHosts -Value * -Force
```

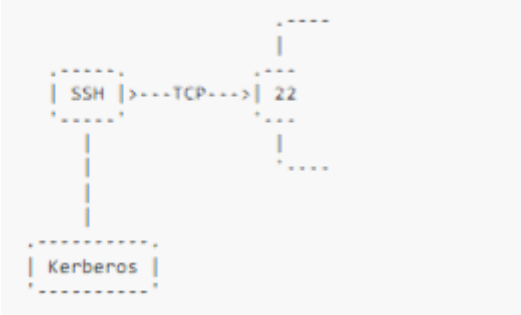
Настройка TrustedHost в клиенте, чтобы разрешить подключение к любому компьютеру

SSH

SSH (Secure Shell) — это широко используемый протокол для доступа и управления системами Unix, такими как Linux, но с 2018 года он также доступен для Windows. Даже если это не связано напрямую с Active Directory, обычно ко многим компьютерам Linux, развернутым в домене, можно получить доступ через SSH, поэтому необходимо знать, как это работает и что можно с ним делать.

Службы SSH по умолчанию прослушивают порт `22`.

SSH-порт



SSH — универсальный протокол, который позволяет пользователю получить оболочку в удаленной системе, передавать файлы (с помощью утилиты `scp`) и устанавливать туннели SSH.

Он активно используется компьютерами Linux, и, возможно, вы можете использовать его для перемещения между компьютерами домена, если сможете найти некоторые ключи ssh или действительные учетные данные пользователя.

```
$ ssh foo@db.contoso.local
foo@db.contoso.local's password:
Linux db 4.19.0-14-amd64 #1 SMP Debian 4.19.171-2 (2021-01-30) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Apr 26 11:23:20 2021 from 192.168.122.1
foo@db:~$ hostname
id
```

Сеанс SSH в db.contoso.local от имени пользователя foo

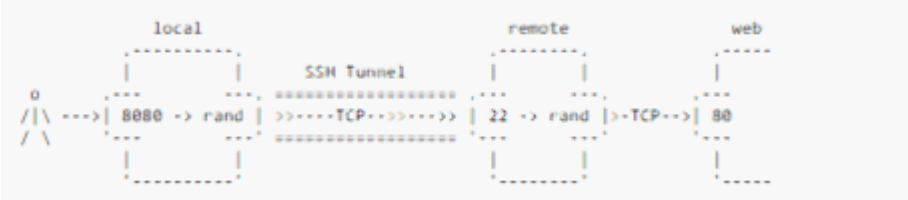
Более того, это также может быть с Kerberos, если целевой компьютер добавлен в домен. Аутентификацию Kerberos можно использовать, включив аутентификацию `GSSAPI` (с помощью `-o GSSAPIAuthentication=yes`).

SSH-туннелирование

Туннелирование SSH позволяет перенаправлять соединения с портов локального компьютера на удаленный и наоборот, поэтому это может быть очень полезно для обхода брандмауэров и сегментации сети.

SSH поддерживает три типа переадресации портов:

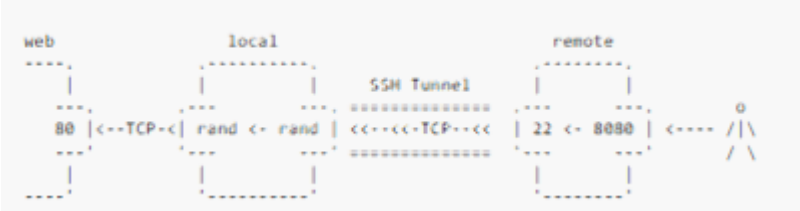
Переадресация локального порта
В этом случае вы можете сопоставить локальный порт с портом, доступным для удаленного компьютера. Например, если удаленный компьютер `remote.contoso.local` может получить доступ к веб-сайту `web.contoso.local:80`, который недоступен для вашего компьютера, вы можете сопоставить локальный порт, например, `8080`, с портом `80 web.contoso.local` с выполняющимся соединением SSH `ssh -L 8080:web.contoso.local:80 user@remote.contoso.local`. Затем вы можете получить доступ к удаленной веб-странице, подключившись к локальному порту `8080`.



Переадресация локального порта SSH

Переадресация удаленного порта

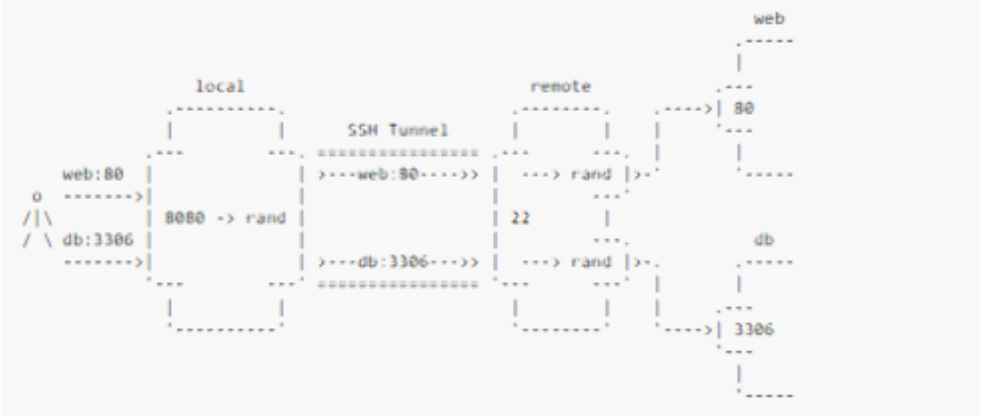
Переадресация удаленного порта противоположна переадресации локального порта. В этом случае можно сделать так, чтобы удаленный компьютер мог получить доступ к порту, доступному вашему компьютеру. Если, например, получить доступ к веб-странице, `web.contoso.local:80`, но удаленный компьютер не может, то вы можете сопоставить порт, такой как `8080` удаленной машины, с портом `80` `web.contoso.local` с помощью следующей команды `ssh -R 8080:web.contoso.local:80 user@remote.contoso.local`. Таким образом, люди, которые подключаются к порту `8080` удаленного компьютера, смогут получить доступ к веб-серверу.



Переадресация удаленного порта SSH

Динамическая переадресация портов

Наконец, динамическая переадресация портов позволяет взаимодействовать с любым портом, доступным для удаленного компьютера, путем создания прокси-сервера `SOCKS`. Вы указываете локальный порт, где прокси-сервер `SOCKS` будет слушать, и он будет пересылать все ваши запросы на удаленный компьютер через `SSH`, а затем на целевой компьютер и порт. Например, вы можете настроить прокси-сервер SOCKS на порт `8080` с помощью следующей команды `ssh -D 8080 user@remote.contoso.local`.

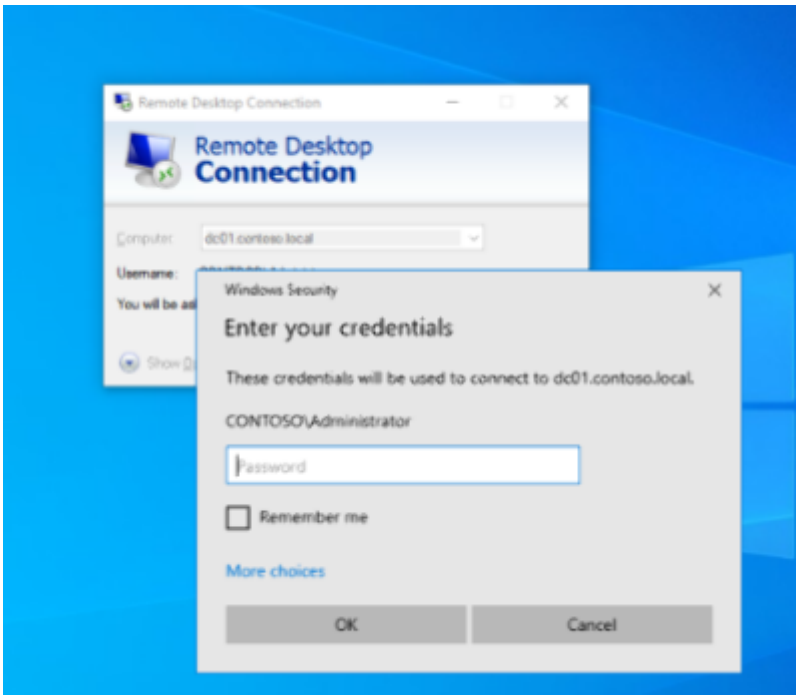


Динамическая переадресация портов SSH

Иногда переадресация TCP отключена на серверах SSH, что не позволяет использовать их для создания туннелей SSH. В этих случаях можно использовать SaSSHimi для создания туннелей.

RDP

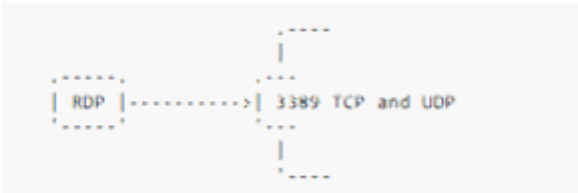
RDP (протокол удаленного рабочего стола) — это протокол, который позволяет подключаться к другим компьютерам, предоставляя графический интерфейс пользователя. Обычно используется в средах Windows для подключения и управления удаленными компьютерами, поскольку и клиент, и сервер включены в Windows по умолчанию.



RDP-клиент для Windows

Проверить использует ли компьютер RDP можно, проверив, открыты ли порты `3389/TCP` или `3389/UDP`.

RDP-порт



Однако, чтобы получить доступ к этому компьютеру, пользователь должен быть членом `Administrators` или `Remote Desktop Users` локальной группы. Кроме того, будьте осторожны, поскольку в Windows разрешен только графический сеанс, поэтому подключение через RDP может привести к выходу другого пользователя из системы.

Нужно иметь в виду, что, когда машина подключена через RDP, учетные данные пользователя отправляются по сети на целевую машину (начиная с поставщика CredSSP), поэтому пользователи, подключенные через RDP, подвержены защите учетных данных путем сброса памяти процесса `lsass`.

Дополнительные возможности Майкрософт

Active Directory является центральным элементом сетевой экосистемы, и многие другие продукты Microsoft использовали/расширяли ее для различных целей. Этот раздел включает некоторое программное обеспечение Microsoft, о котором злоумышленник должен знать, если оно установлено в домене.

ADCS

Службы сертификатов Active Directory (ADCS) — это одна из ролей сервера, представленных в Windows Server 2008, которая предоставляет пользователям настраиваемые службы для создания и управления сертификатами инфраструктуры открытых ключей (PKI), которые можно использовать для шифрования и цифровой подписи электронных документов, сообщений электронной почты и сообщений.

Приложения, поддерживаемые ADCS, включают безопасные беспроводные сети, виртуальные частные сети (VPN), безопасность интернет-протокола (IPSec), защиту доступа к сети (NAP), шифрование файловых систем (EFS), вход в систему с помощью смарт-карты и многое другое.

Вот соответствующие инструменты:

LAPS

`LAPS` (Local Administrator Password Solution) — это утилита для управления паролями локальных администраторов компьютеров домена. LAPS рандомизирует пароли локальных администраторов, чтобы избежать повторного использования учетных данных, и периодически меняет их.

Для этого LAPS добавляет к объектам-компьютерам домена два свойства: `ms-Mcs-AdmPwd` и `ms-Mcs-AdmPwdExpirationTime`.

`ms-Mcs-AdmPwd` хранит локальный пароль администратора компьютера, и его можно увидеть только в том случае, если ему предоставлено явное разрешение. Если можно получить пароль локального администратора, вы можете подключиться к компьютеру (используя аутентификацию NTLM) с правами администратора.

Другое свойство `ms-Mcs-AdmPwdExpirationTime` может быть прочитано кем угодно (по умолчанию), поэтому для идентификации компьютеров, управляемых LAPS, можно искать объекты-компьютеры, содержащие это свойство.

Exchange

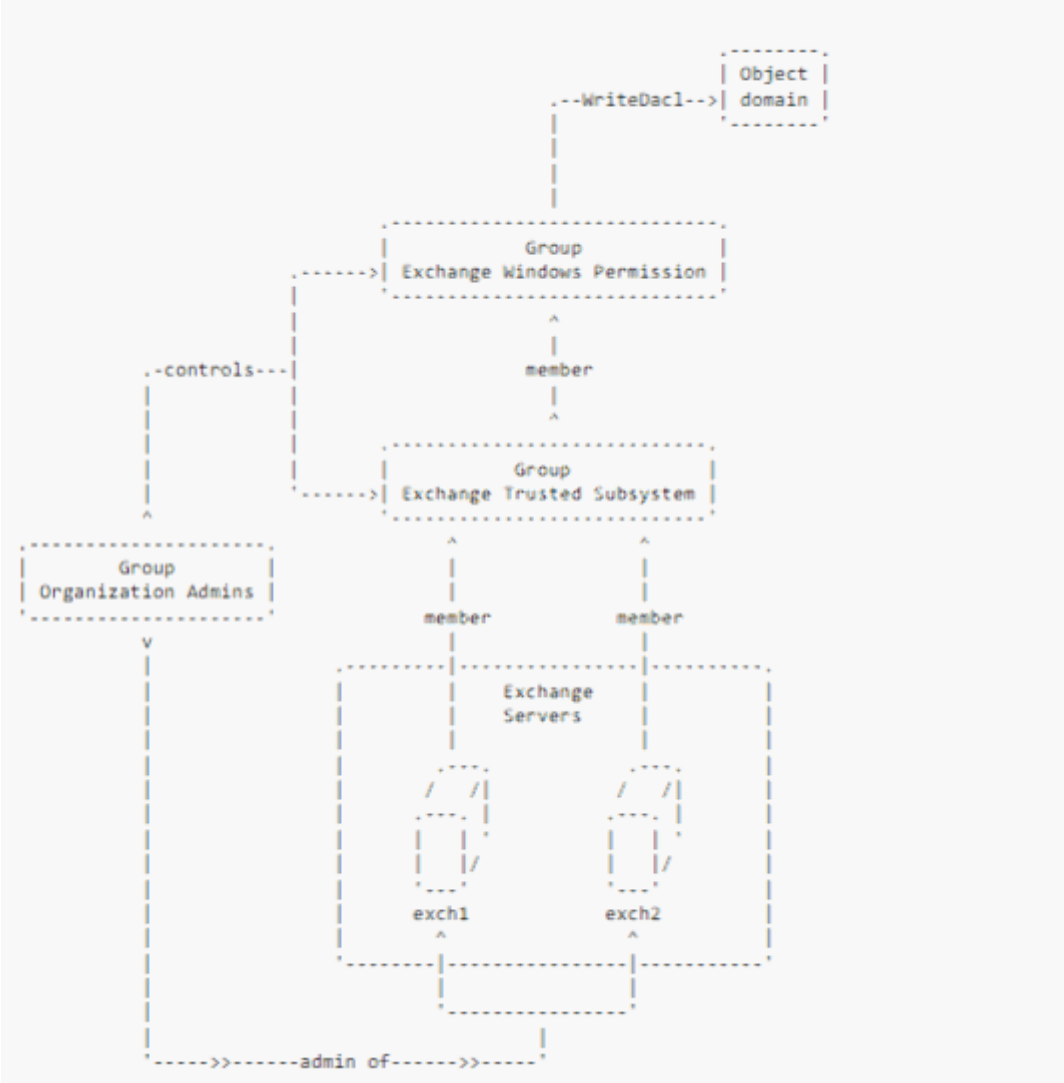
Exchange — это почтовый сервер, разработанный Microsoft, который может быть установлен на серверах Windows и интегрирован с Active Directory.

При установке Exchange в домене создается несколько групп и `ACE`.

Кроме того, группа `Exchange Trusted Subsystem`, к которой принадлежат все серверы Exchange, является членом группы `Exchange Windows Permissions`. Таким образом, компрометация любого сервера Exchange может позволить злоумышленнику получить права на компрометацию всего домена.

Возможно, самым известным злоупотреблением разрешениями Exchange была атака `PrivExchange`, которая использовала уязвимость на серверах Exchange, которая позволяла пользователю принудительно установить HTTP-аутентифицированное соединение с сервера Exchange на другой компьютер. Затем, выполнив атаку NTLM Relay с HTTP на LDAP, сервер Exchange был вынужден предоставить права `DCsync` произвольной учетной записи пользователя. Microsoft также выпустила исправление для этой уязвимости в обновлении за февраль 2019 года.

Более того, группа `Organization Admins` (также добавленная Exchange) может контролировать членство в `Exchange Windows Permissionsи Exchange Trusted Subsystem`. Кроме того, `Organization Admins` локальный администратор серверов Exchange, поэтому членство в этой группе также позволит пользователю скомпрометировать весь домен.



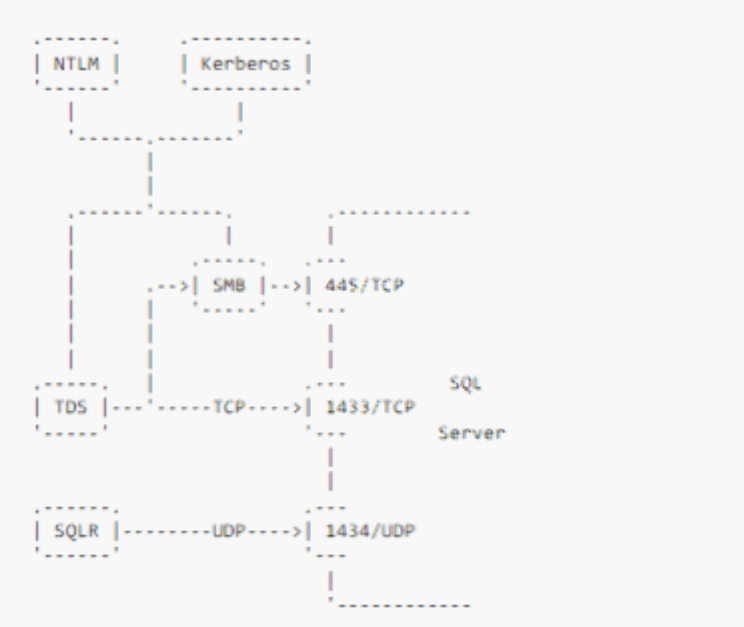
Группы обмена и разрешения

SQL-сервер

Microsoft SQL Server (MSSQL) — это система управления базами данных, созданная Microsoft. Обычно он устанавливается на компьютерах с Windows Server, прослушивает TCP-порт `1433`, и многие веб-приложения используют его в качестве базы данных.

SQL Server прослушивает TCP-порт `1433`, и к нему можно подключиться, используя учетные данные домена, поскольку он использует протокол `TDS`, совместимый с аутентификацией NTLM и Kerberos.

Для связи с сервером SQL можно использовать протокол TDS напрямую через TCP или с помощью SMB. В случае использования TCP порт по умолчанию — `1433`, но также можно использовать динамический порт.



Порт и протоколы SQL-сервера

При использовании динамического порта выбирается случайный TCP-порт. Чтобы удаленный клиент мог обнаружить этот порт, SQL Server должен быть включен на порту `UDP 1434`, ожидая запросов `SQLR` (разрешение SQL Server). Можно использовать инструмент impacket mssqlinstance.py для обнаружения динамического порта SQL-сервера.

Запрос к браузеру SQL Server

Здесь можно увидеть, что порт SQL Server — `50377`, теперь можно использовать клиент SQL Server, такой как `HeidiSQL`, `SQL Server Management Studio` или `PowerUpSQL`, для подключения к базе данных.

```
$ mssqlinstance.py 192.168.100.19
Impacket v0.9.21 - Copyright 2020 SecureAuth Corporation

[*] Instance 0
ServerName:SRV01
InstanceName:SQLEXPRESS
IsClustered:No
Version:15.0.2000.5
[*] Instance 1
ServerName:SRV01
InstanceName:MSSQLSERVER
IsClustered:No
Version:15.0.2000.5
tcp:50377
```

```
PS C:\> . .\PowerUpSQL.ps1
PS C:\> Get-SQLQuery -Query "Select @@version" -Instance "srv01,50377"

Column1
-------
Microsoft SQL Server 2019 (RTM) - 15.0.2000.5 (X64) ...
```

SQL-запрос с динамическим портом

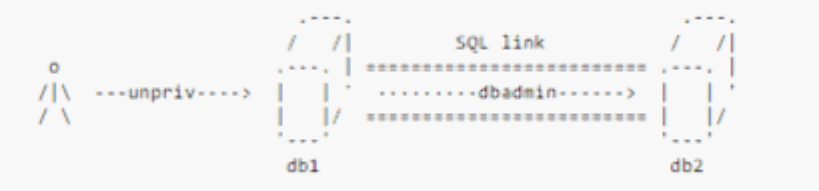
Важным аспектом SQL-сервера является возможность выполнения команд через команду `xp_cmdshell`, если это разрешено.

Иногда в неправильно сконфигурированных средах, даже если команда `xp_cmdshell` отключена, у пользователя достаточно привилегий, чтобы включить ее с помощью директивы `sp_configure`.

Кроме того, команда `xp_dirtree` может быть полезна для доступа к файлам в сети (с использованием путей UNC) или для выполнения аутентифицированных запросов к другим машинам с использованием учетной записи компьютера домена, чтобы вспомнить хэши NTLM для взлома или выполнить ретрансляцию NTLM.

Кроме того, невероятно полезной характеристикой для атакующего могут быть ссылки на SQL Server. SQL Server позволяет создавать связи с другими источниками данных, такими как другие базы данных SQL.

Эти ссылки интересны тем, что даже если они созданы привилегированным пользователем, например администратором, они могут использоваться любым пользователем и позволяют выполнять команды на удаленных машинах с привилегиями создателя ссылки.



Использование ссылки, созданной dbadmin

Кроме того, если вам нравится выполнять поворот через SQL Server, вы также можете преобразовать его в прокси-сервер SOCKS с помощью mssqlproxy.

Чтобы узнать о других способах злоупотребления SQL-серверами, вы можете использовать инструментарий PowerUpSQL.

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.