

# Неприметные токены. Часть 1. Теория

[ardent101.github.io/posts/tokens\\_theory](https://ardent101.github.io/posts/tokens_theory)

March 2, 2023

марта 2, 2023 · 12 мин · Ardent101



## Вступление

В ходе тестирования на проникновение нередко удается получить доступ с правами уровня локального администратора к какому-то сетевому объекту, функционирующему под управлением операционной системы семейства Windows.

Следующим этапом, как правило, является повышение привилегий до администратора домена. Существует множество способов пост эксплуатации конечной системы при наличии к ней административного доступа. Наиболее перспективным методом представляется поиск учетных данных, таких как:

- пароли в открытом виде
- NT-хэши паролей
- TGT
- MsCache v2 хеши
- секреты в DPAPI и др.

Приведенный метод неплохо изучен и хорошо себя зарекомендовал, но всегда интересно расширить свой арсенал возможностей другими подходами. Одним из таких подходов является манипулирование токенами. В предстоящей серии материалов попробую разобраться что из себя представляют токены в Windows и как они могут пригодиться для повышения привилегий.

Первая часть содержит теорию необходимую для дальнейшего понимания материала.

## Сессия пользователя

Как обычно, основательно подойдем к изучению вопроса и начнем издалека. Прежде чем приступать к токенам необходимо разобраться с таким понятием, как сессия пользователя.

## Бытовая аналогия

Рассмотрим следующий пример - организация доступа на стадионы чемпионата мира по футболу 2018.

Поверхностно опишем процедуру прохода:

1. Каждый, кому требовалось пройти на стадион, сначала приходил в один из нескольких аккредитационных центров, где предоставлял свой паспорт и обоснование для прохода.
2. Сотрудник аккредитационного центра проверял личность предъявителя и его обоснование, тем самым аутентифицируя его. Если личность была болельщиком, то обоснованием являлся билет на матч, если репортером, то требовалось наличие соответствующей записи в заранее согласованной базе данных. Кроме того требовалось смотреть в базу, чтобы выявлять нежелательных посетителей, входящих в черный список (пранкеры, агрессивные фанаты). Отметим, что процедуру проверки в базе данных можно считать довольно ресурсозатратной.
3. В результате успешной проверки посетителю стадиона выдавалась аккредитация. Таким образом осуществлялась авторизация.



Пример аккредитации

Каждая аккредитация содержала следующую информацию:

- ФИО и фото владельца (замазаны березовым цветом)
- Перечень стадионов доступных для посещения
- Роль владельца (волонтер, обслуживающий персонал, сотрудник безопасности, журналист и т.д.)
- Перечень зон внутри стадиона, доступных для посещения (трибуны, раздевалки, подтрибунные помещения, микс зона и т.д.)

Каждый раз, когда посетитель стадиона пытался пройти в определенную зону, охрана осматривала его аккредитацию и принимала соответствующее решение о допуске. Аккредитация позволяла быстро идентифицировать человека, и понять какими полномочиями он обладает.

Также для простоты представим, что по выходу со стадиона аккредитация уничтожалась и в следующий раз процедура повторялась снова. На практике разумеется было по-другому, но для примера так лучше.

Нахождение посетителя на стадионе можно назвать сессией посетителя. Удобство заключается в том, что в рамках заданной сессии проверка личности осуществляется только один раз и нет необходимости при каждом действии заново требовать паспорт и сверять записи в базе данных.

## Сессия в Windows

---

Рассмотрим, что происходит при входе пользователя в Windows:

1. Пользователь садится за компьютер и предоставляет системе свои аутентификационные данные (логин, пароль, название домена, smart карту или другие).
2. Система передает полученную информацию в свой локальный центр безопасности (далее - LSA).
3. LSA осуществляет аутентификацию пользователя.

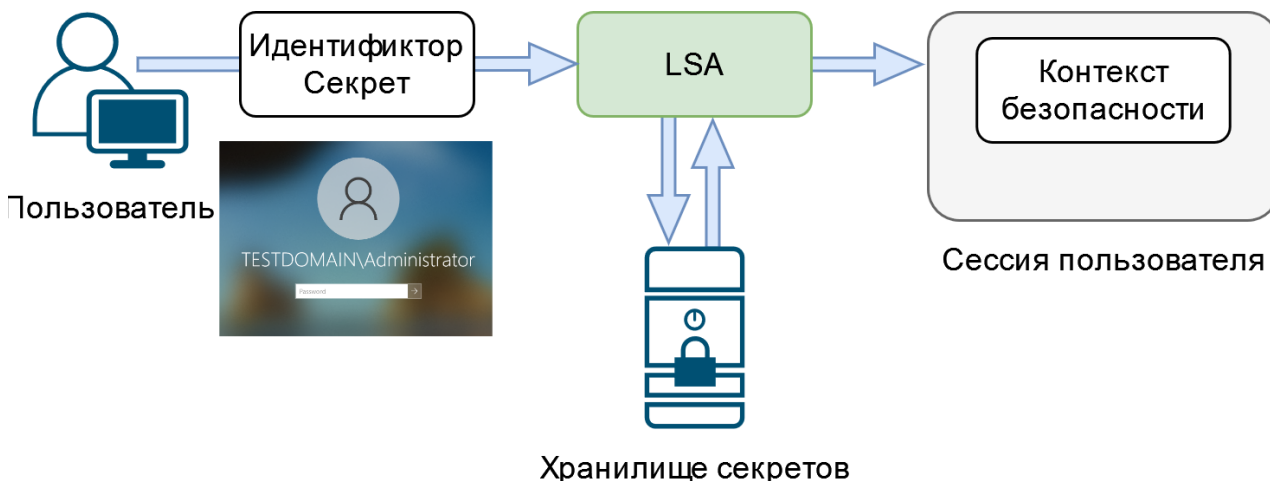
Как именно это происходит выходит за рамки настоящего материала. Для желающих ознакомиться с тонкостями процедуры аутентификации ранее были написаны другие материалы, где рассматриваются принципы работы протокола Kerberos.

4. В результате успешной аутентификации пользователю предоставляется сессия, позволяющая получать доступ к защищаемым объектам операционной системы при наличии соответствующих прав.

В Windows в качестве защищаемых объектов выступают совершенно различные объекты, например: файлы, именованные каналы, ключи реестра, процессы, потоки, сетевые папки, службы, устройства, объекты Active Directory и т.д.

Пользователю не требуется заново вводить свой пароль при каждом открытии файла или запуске прикладной программы. Это называется концепцией единого входа (Single Sign On - SSO), которая реализуется, в том числе с помощью механизма сессий.

В рамках сессии для получения доступа к защищаемому объекту пользователю необходимо обладать определенным контекстом безопасности.



Связь пользователя, сессии и контекста безопасности

Контекст безопасности представляет собой набор специальных атрибутов необходимых операционной системе для принятия решений о предоставлении пользователю доступа к защищаемому объекту или разрешении выполнить системную операцию.

Под системной операцией понимается действие, требующее наличия определенной привилегии для его выполнения, например выключение системы или изменение значения счетчика времени.

Важно отметить, что по сути контекст безопасности является своеобразной “аккредитацией” пользователя. Всякий раз, когда пользователь обращается к защищаемому объекту, система смотрит сессию пользователя и проверяет связанную с ней “аккредитацию”.

## Процессы в Windows

Пользователь использует систему для выполнения различных программ. Программа представляет собой некоторый код (набор команд и инструкций), который хранится в системе в виде файлов. Программа используется в рамках процесса.

Процесс - это объект операционной системы, содержащий набор ресурсов (в частности выделенный диапазон адресов виртуальной памяти) и данных используемых в ходе работы программы.

Уместно провести следующую аналогию:

- Программа - это рецепт в поваренной книге, то есть описание какого-то порядка действий, алгоритма.
- Процесс - это кухня, на которой работает повар, нарезаны ингредиенты, кипит кастрюля, где-то лежит листочек с рецептом, а также посуда и приборы необходимые для приготовления блюда.

Процессы позволяют системе изолировать работу программ. Таким образом исполняющая среда получается гораздо более надежной и стабильной, поскольку выход из строя одного процесса никак не сказывается на работе других процессов.

## Токены в первом приближении

---

Что происходит, когда пользователь открывает файл в текстовом редакторе?

Система должна определить обладает ли процесс программы, запущенной от имени определенного пользователя, необходимыми правами доступа. Внимательный читатель может предположить, что это несложно сделать с помощью контекста безопасности, связанного с сессией пользователя.

В целом, так и есть, но на практике реализовано немного сложнее. Пользователь работает с множеством программ, то есть процессов. Некоторым процессам может не требоваться наличия определенных прав пользователя, например небезопасно запускать браузер с правами на просмотр различных директорий, которыми в свою очередь может обладать проводник.

Если бы все процессы в рамках пользовательской сессии ссылались на один и тот же контекст безопасности, хранящийся в атрибутах указанной сессии, то изменение контекста безопасности одного процесса повлияло бы на контекст безопасности всех остальных процессов.

Рассмотренный пример, лишь один из многих иллюстрирующих, почему разработчики операционной системы Windows решили наделить каждый процесс своим собственным контекстом безопасности, который называется токеном доступа.

Первоначально токен создается в ходе авторизации пользователя, на одном из последних этапов входа в систему. Указанный токен присваивается первому процессу, созданному для работы от имени авторизованного пользователя.

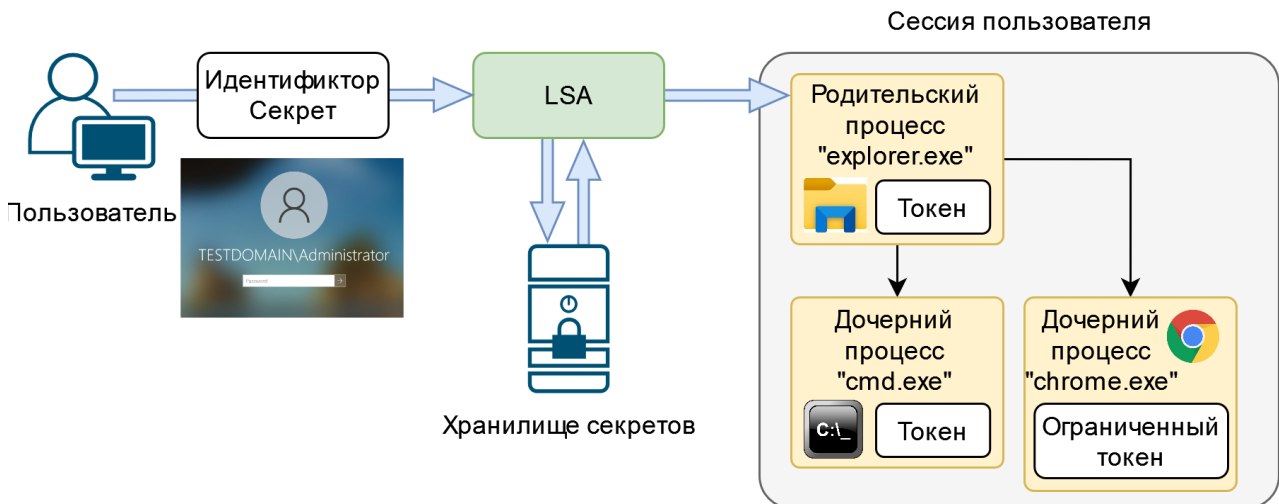
Важно понимать, что процесс, не имеющий токена, не может существовать. В дальнейшем новые процессы по умолчанию наследуют токен доступа от родительского процесса.

Отметим, что создать токен “вручную” нельзя. Прямой доступ к отдельным атрибутам токенов из пользовательского режима невозможен. Это сделано, во избежание несанкционированного изменения полномочий пользователя путем

редактирования токенов. Только программный код, выполняющийся в режиме ядра, может получить прямой доступ к атрибутам токена (равно как и к любым другим объектам операционной системы).

Чуть более подробно о локальном повышении привилегий в результате эксплуатации критической уязвимости с последующим выполнением кода в пространстве ядра и манипуляцией токеном доступа можно почитать [здесь](#).

Таким образом между пользователем, его сессией, процессами и токенами имеется следующая связь:



Теперь в первом приближении можно ввести следующее определение:

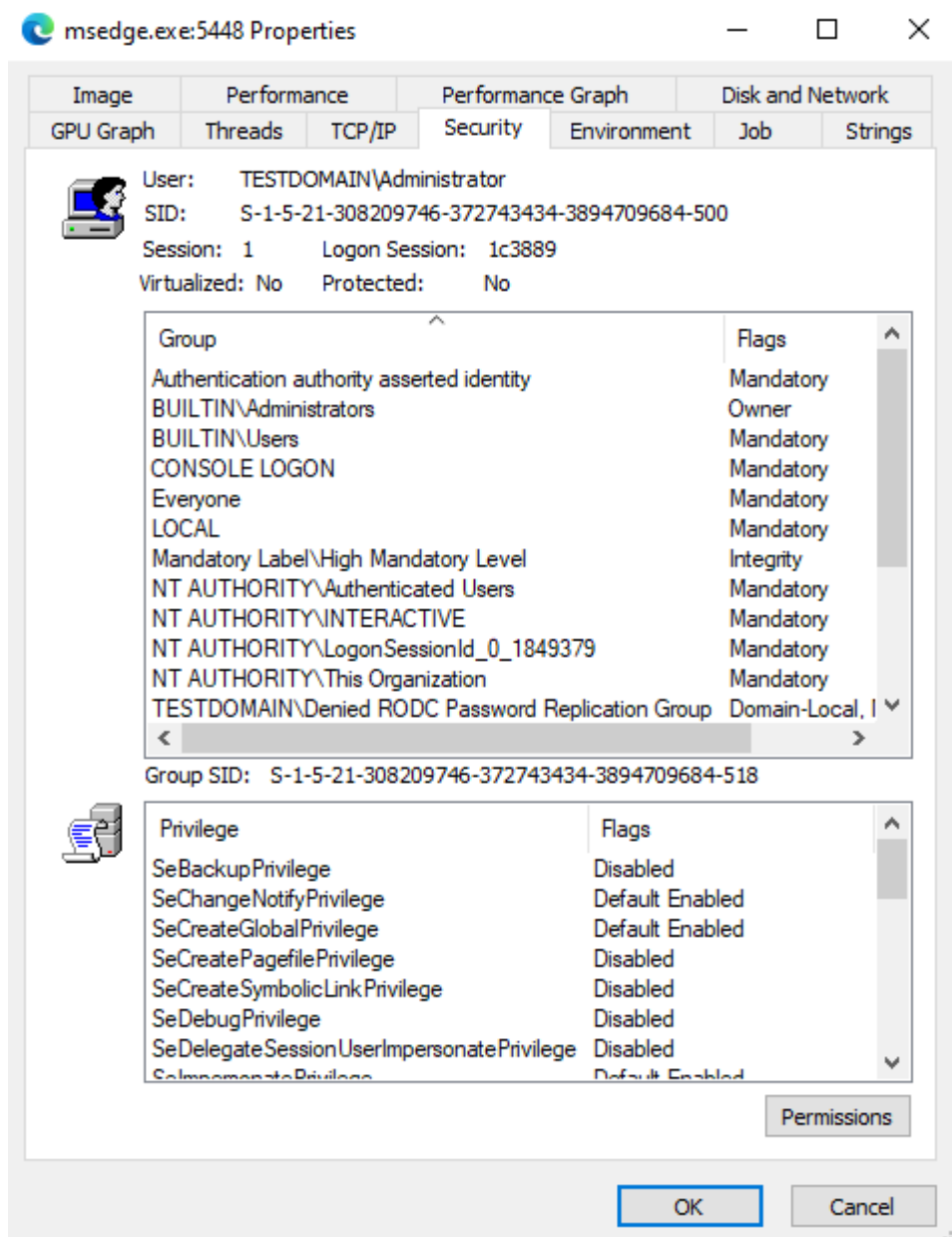
**Токен доступа пользователя** - объект ядра ОС Windows, описывающий контекст безопасности процесса, работающего от имени указанного пользователя.

Синонимы: токен, токен доступа, маркер доступа, маркер безопасности.

## Атрибуты токена

Перечислим некоторые основные атрибуты, содержащиеся в токене доступа пользователя:

- Идентификатор пользователя
- Идентификатор входа в систему (Logon ID - LUID)
- Перечень локальных и доменных групп, в которые входит пользователь
- Список привилегий, которыми обладает пользователь



Пример отображения некоторых атрибутов токена

Более наглядно посмотреть токены и их атрибуты можно с помощью утилиты [TokenViewer](#)

Разумеется атрибутов гораздо больше. Для первого знакомства приведенного списка достаточно. Далее постепенно рассмотрим другие важные атрибуты, а также различные виды токенов доступа.

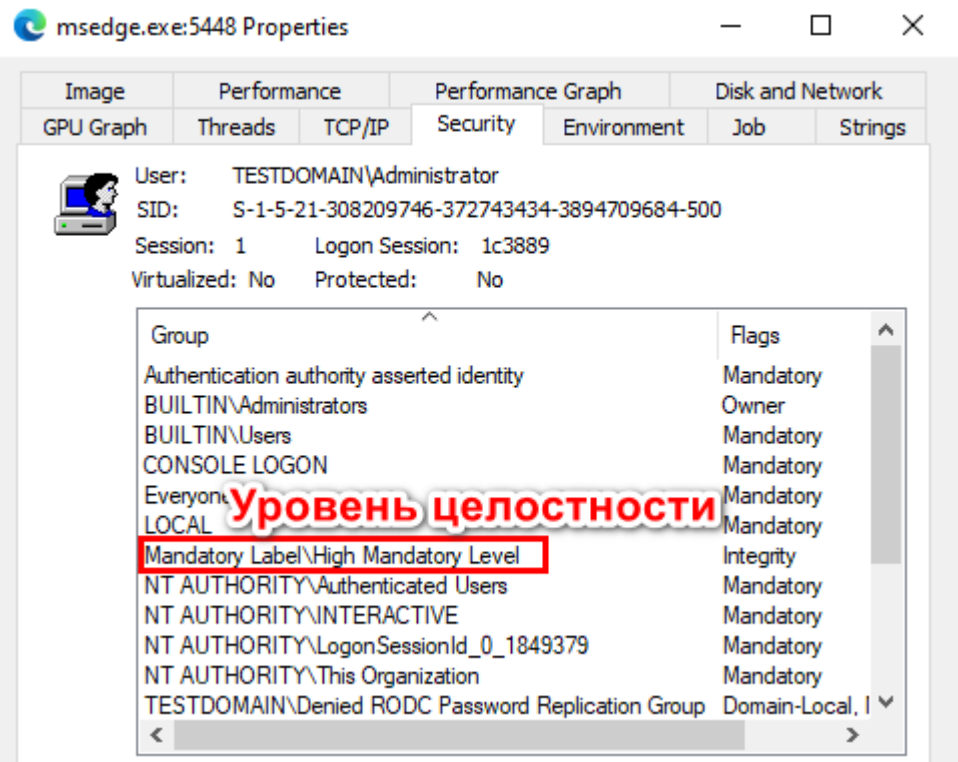
## Уровни целостности

Начиная с Windows Vista, в операционных системах семейства Windows реализован механизм мандатного контроля целостности (mandatory integrity control, MIC).

Согласно MIC в Windows каждому защищаемому объекту операционной системы присваивается мандатная метка, соответствующая определенному уровню целостности. Физически мандатная метка представляет собой целое число. Чем больше число, тем выше уровень целостности.



Уровень целостности процесса хранится в токене указанного процесса в списке групп:



В современных версиях Windows поддерживаются следующие уровни целостности:

Числовой идентификатор	Название	Описание
0	Untrusted	Используется редко, например в ходе анонимного подключения.
4096	Low	Назначается процессам браузеров или контейнеров, чтобы ограничить права на запись в отношении системных файлов или ключей реестра.
8192	Medium	По умолчанию назначается большинству процессов прикладных программ.
8448	Medium Plus	Недокументирован.
12288	High	Назначается процессам прикладных программ администратором.
16384	System	Автоматически назначается всем системным процессам. Процессам прикладных программ назначаться не может.
20480	Protected Process	Назначается специальным защищаемыми процессам



Основная идея мандатного контроля целостности заключается в том, что “низкоцелостные” субъекты не могут изменять “высокоцелостные” объекты.

Рассмотрим, как это используется на практике.

## Токены привилегированных учетных записей

---

Как отмечалось ранее, токен доступа создается в результате успешной аутентификации при взаимодействии с LSA. Важный момент заключается в том, что помимо прочего LSA выполняет проверку учетной записи на предмет членства в особо привилегированных группах, например:

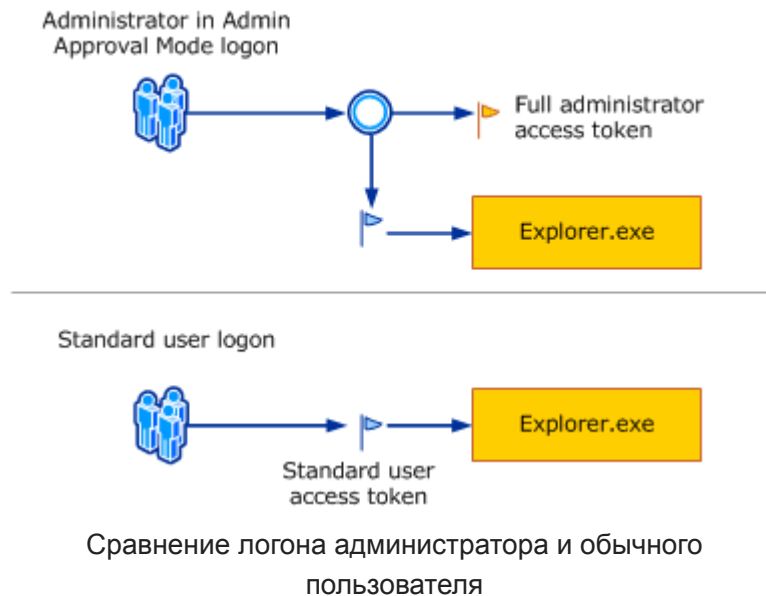
- Builtin Administrators
- Domain / Enterprise Administrators
- Policy Administrators
- Account Operators
- Backup Operators

а также на предмет наличия следующих “опасных” привилегий:

- SeBackupPrivilege
- SeCreateTokenPrivilege
- SeDebugPrivilege
- SeImpersonatePrivilege
- SeLabelPrivilege
- SeLoadDriverPrivilege
- SeRestorePrivilege
- SeTakeOwnershipPrivilege
- SeTcbPrivilege

Если учетная запись пользователя состоит в одной из подобных групп или обладает хотя бы одной из перечисленных привилегий, то по умолчанию LSA создает два отдельных токена:

1. Полный токен (в англ. “elevated token”), содержащий все назначенные права, группы, привилегии и имеющий высокий уровень целостности.
2. Отфильтрованный токен, в частности обладающий следующими ограничениями:
  - отсутствуют “опасные” привилегии
  - присваивается средний уровень целостности



Таким образом, полномочия процесса, выполняющегося на среднем уровне мандатной целостности от имени административной учетной записи, практически не отличаются от полномочий процесса, выполняющегося от имени непривилегированного пользователя. Это позволяет реализовать принцип минимизации полномочий без создания отдельных учетных записей для повседневной работы и для администрирования операционной системы.

```

Select Command Prompt
Microsoft Windows [Version 10.0.19044.2364]
(c) Microsoft Corporation. All rights reserved.

C:\Users\User>whoami /all

USER INFORMATION
-----
User Name          SID
=====
workstation\user S-1-5-21-2643705787-2218337860-222870034-1002

GROUP INFORMATION
-----
Group Name                                     Type          SID            Attributes
-----
Everyone                                     Well-known group S-1-1-1-0      Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Local account and member of Administrators group Well-known group S-1-5-114      Group used for deny only
BUILTIN\Users                               Alias          S-1-5-32-545   Mandatory group, Enabled by default, Enabled group
BUILTIN\Administrators                     Alias          S-1-5-32-544   Group used for deny only
NT AUTHORITY\INTERACTIVE                    Well-known group S-1-5-4        Mandatory group, Enabled by default, Enabled group
CONSOLE LOGON                              Well-known group S-1-2-1        Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Authenticated Users            Well-known group S-1-5-11       Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\This Organization               Well-known group S-1-5-15       Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Local account                  Well-known group S-1-5-113      Mandatory group, Enabled by default, Enabled group
LOCAL                                       Well-known group S-1-2-0        Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\NTLM Authentication             Well-known group S-1-5-64-10    Mandatory group, Enabled by default, Enabled group
Mandatory Label\Medium Mandatory Level      Label          S-1-16-8192

PRIVILEGES INFORMATION
-----
Privilege Name      Description              State
-----
SeShutdownPrivilege Shut down the system     Disabled
SeChangeNotifyPrivilege Bypass traverse checking Enabled
SeUndockPrivilege    Remove computer from docking station Disabled
SeIncreaseWorkingSetPrivilege Increase a process working set Disabled
SeTimeZonePrivilege  Change the time zone     Disabled

USER CLAIMS INFORMATION
-----
User claims unknown.

Kerberos support for Dynamic Access Control on this device has been disabled.

C:\Users\User>

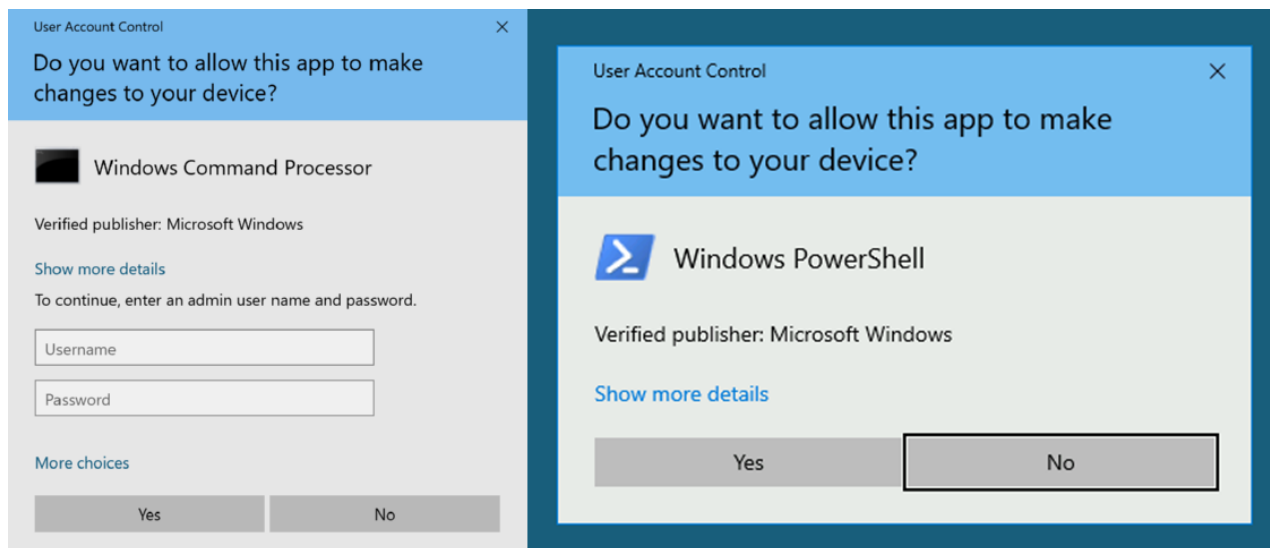
```

Метка среднего уровня целостности

Перечень привилегий ограничен

Содержимое токена процесса администратора, запущенного в обычном режиме

Когда пользователю необходимо выполнить задачу, требующую повышенного токена доступа, Windows автоматически запрашивает подтверждение. Этот запрос называется запросом на повышение прав и реализуется с помощью специального компонента - UAC (User Account Control).



Пример оконных сообщений UAC

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.19044.2364]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami /all

USER INFORMATION
-----
User Name          SID
=====
workstation\user S-1-5-21-2643705787-2218337860-222870034-1002

GROUP INFORMATION
-----
Group Name          Type          SID          Attributes
=====
Everyone            Well-known group S-1-1-0      Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Local account and member of Administrators group Well-known group S-1-5-114    Mandatory group, Enabled by default, Enabled group
BUILTIN\Users       Alias          S-1-5-32-545 Mandatory group, Enabled by default, Enabled group
BUILTIN\Administrators Alias          S-1-5-32-544 Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\INTERACTIVE Well-known group S-1-5-4      Mandatory group, Enabled by default, Enabled group
CONSOLE LOGON       Well-known group S-1-2-1      Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Authenticated Users Well-known group S-1-5-11     Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\This Organization Well-known group S-1-5-15     Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Local account Well-known group S-1-5-113    Mandatory group, Enabled by default, Enabled group
LOCAL               Well-known group S-1-2-0      Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\NTLM Authentication Well-known group S-1-5-64-10  Mandatory group, Enabled by default, Enabled group
Mandatory Label\High Mandatory Level Label          S-1-16-12288

PRIVILEGES INFORMATION
-----
Privilege Name      Description
=====
SeIncreaseQuotaPrivilege Adjust memory quotas for a process Disabled
SeSecurityPrivilege Manage auditing and security log Disabled
SeTakeOwnershipPrivilege Take ownership of files or other objects Disabled
SeLoadDriverPrivilege Load and unload device drivers Disabled
SeSystemProfilePrivilege Profile system performance Disabled
SeSystemTimePrivilege Change the system time Disabled
SeProfileSingleProcessPrivilege Profile single process Disabled
SeIncreaseBasePriorityPrivilege Increase scheduling priority Disabled
SeCreatePagefilePrivilege Create a pagefile Disabled
SeBackupPrivilege Back up files and directories Disabled
SeRestorePrivilege Restore files and directories Disabled
SeShutdownPrivilege Shut down the system Disabled
SeDebugPrivilege Debug programs Disabled
SeSystemEnvironmentPrivilege Modify firmware environment values Disabled
SeChangeNotifyPrivilege Bypass traverse checking Enabled
SeRemoteShutdownPrivilege Force shutdown from a remote system Disabled
SeUndockPrivilege Remove computer from docking station Disabled
SeManageVolumePrivilege Perform volume maintenance tasks Disabled
SeImpersonatePrivilege Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege Create global objects Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set Disabled
SeTimeZonePrivilege Change the time zone Disabled
SeCreateSymbolicLinkPrivilege Create symbolic links Disabled
SeDelegateSessionUserImpersonatePrivilege Obtain an impersonation token for another user in the same session Disabled

USER CLAIMS INFORMATION
-----
User claims unknown.

Kerberos support for Dynamic Access Control on this device has been disabled.

C:\Windows\system32>
```

Содержимое токена процесса, запущенного от имени администратора

Более детальное описание механизма UAC выходит за рамки настоящего материала.

Источники:

- [Официальная документация Microsoft](#)
- [Better know a data source: Process integrity levels](#)

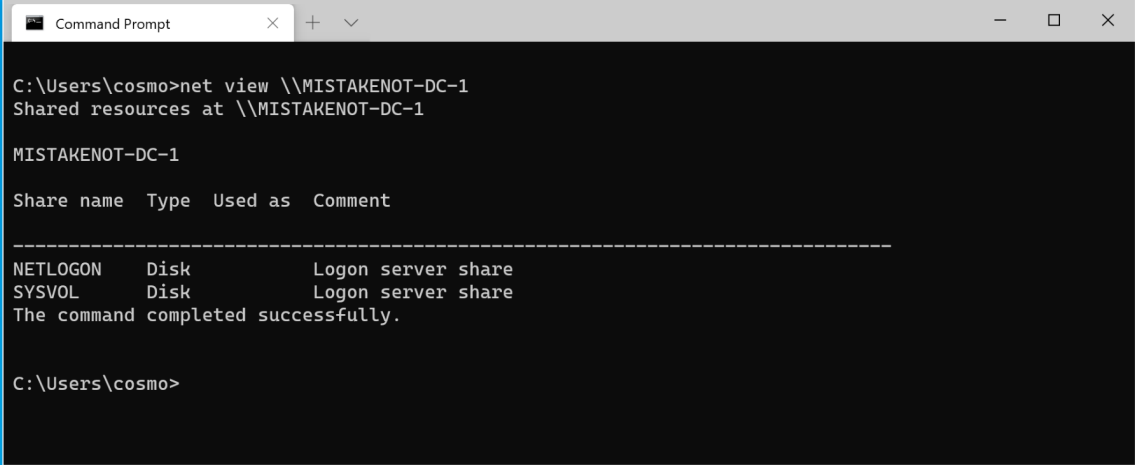
## Виды логонов

Вход в Windows можно осуществить по-разному. Для простоты можно выделить следующие два типа входа (строго говоря их больше):

- Интерактивный
- Сетевой (неинтерактивный)

Интерактивный вход подразумевался всюду ранее. По сути это обычный вход через графическую оболочку, который например осуществляют рядовые пользователи к своим домашним персональным компьютерам или администраторы к удаленным серверам с помощью RDP.

Но что происходит, когда пользователь пытается посмотреть содержимое удаленной сетевой папки, например с помощью следующей команды?



```
C:\Users\cosmo>net view \\MISTAKENOT-DC-1
Shared resources at \\MISTAKENOT-DC-1

MISTAKENOT-DC-1

Share name  Type  Used as  Comment
-----
NETLOGON    Disk      Logon server share
SYSVOL      Disk      Logon server share
The command completed successfully.

C:\Users\cosmo>
```

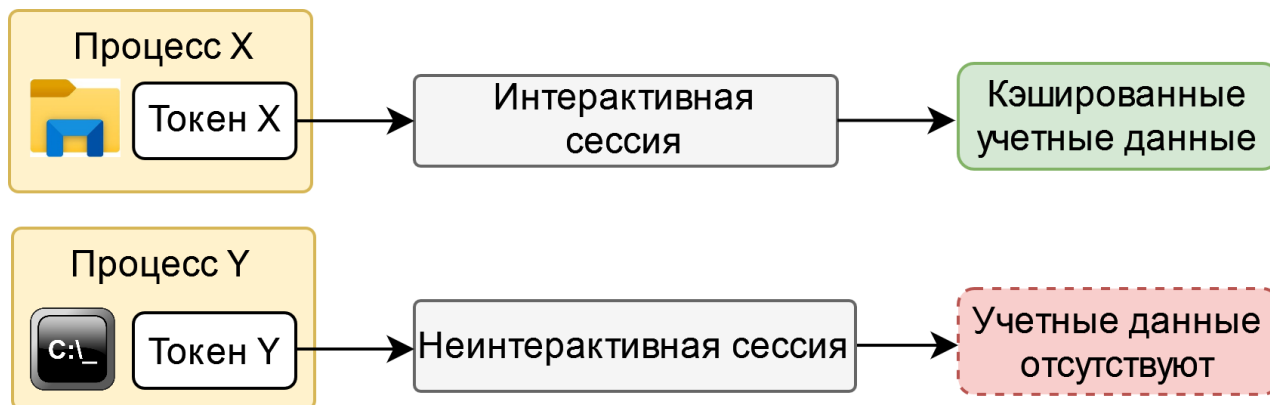
Просмотр содержимого сетевой папки

Пользователь не может просто взять и отправить по сети свой токен удаленной системе. Действительно, токен должен быть связан с какой-то определенной сессией. Таким образом, прежде чем посмотреть содержимое сетевой папки, пользователь должен осуществить вход на удаленную систему на которой располагается эта папка. Другими словами пользователь должен аутентифицироваться к удаленной системе.

Если сессия пользователя, в рамках которой запрашивается доступ к сетевой папке, создавалась с помощью интерактивного входа, то используемые для входа учетные данные пользователя были сохранены в LSASS. В этом случае система автоматически использует сохраненные учетные данные пользователя (NT хеш или TGT) для аутентификации на удаленной машине.

Это еще один из примеров, иллюстрирующий поддержку принципа единого входа в Windows.

Отметим, что между токеном процесса, работающего от имени определенного пользователя в рамках определенной сессии и учетными данными указанного пользователя имеется следующая связь:



Связь между токенами, различными видами сессий и учетными данными

Еще раз отдельно отметим:

- При интерактивном входе создается интерактивная сессия и учетные данные пользователя сохраняются в LSASS.
- При сетевой сессии пользователь подтверждает, что обладает учетными данными, но при этом никуда их не передает. В дальнейшем в LSASS указанные учетные данные не хранятся.

Более подробно о различных видах входа и кешируемых учетных данных можно прочитать в следующих статьях [“Fantastic windows logon types and where to find credentials in them”](#), [“Network vs Interactive Logons”](#).

Теперь рассмотрим, что происходит с другой стороны в удаленной системе.

Сетевая папка по сути является сервисом, предоставляемым некоторым SMB сервером. Процесс “SMB сервера” обрабатывает запрос на аутентификацию и передает полученные учетные данные в LSA, где проверяется их подлинность. Самое интересное, что происходит в результате успешной аутентификации пользователя, запрашивающего просмотр содержимого сетевой папки.

Процесс “SMB сервера” работает от имени и с правами псевдопользователя SYSTEM. Предоставление прав системы всякому аутентифицированному к SMB серверу пользователю однозначно избыточно. Более того указанный пользователь может обладать доступом не ко всему содержимому сетевой папки.

То есть, процессу “SMB сервера” было бы очень удобно как-то работать от имени своих клиентов. В Windows такой механизм предусмотрен и называется олицетворением (англ. Impersonation).

## Олицетворение

До этого уже немного была затронута тема процессов в операционной системе Windows. Теперь расширим это понятие. У каждого процесса есть как минимум один поток, выделяемый при создании процесса (как правило, на практике потоков больше). Этот поток может породить другие потоки, те в свою очередь новые и т.д.

Рассматриваемый всюду ранее токен доступа, назначаемый процессу операционной системы при его создании, называется первоначальным токеном или токеном процесса (англ. primary token).

По умолчанию первоначальный токен наследуется дочерними потоками, но существует механизм олицетворения, который позволяет процессу назначать дочерним потокам специальные токены, отличающиеся от первоначального токена.

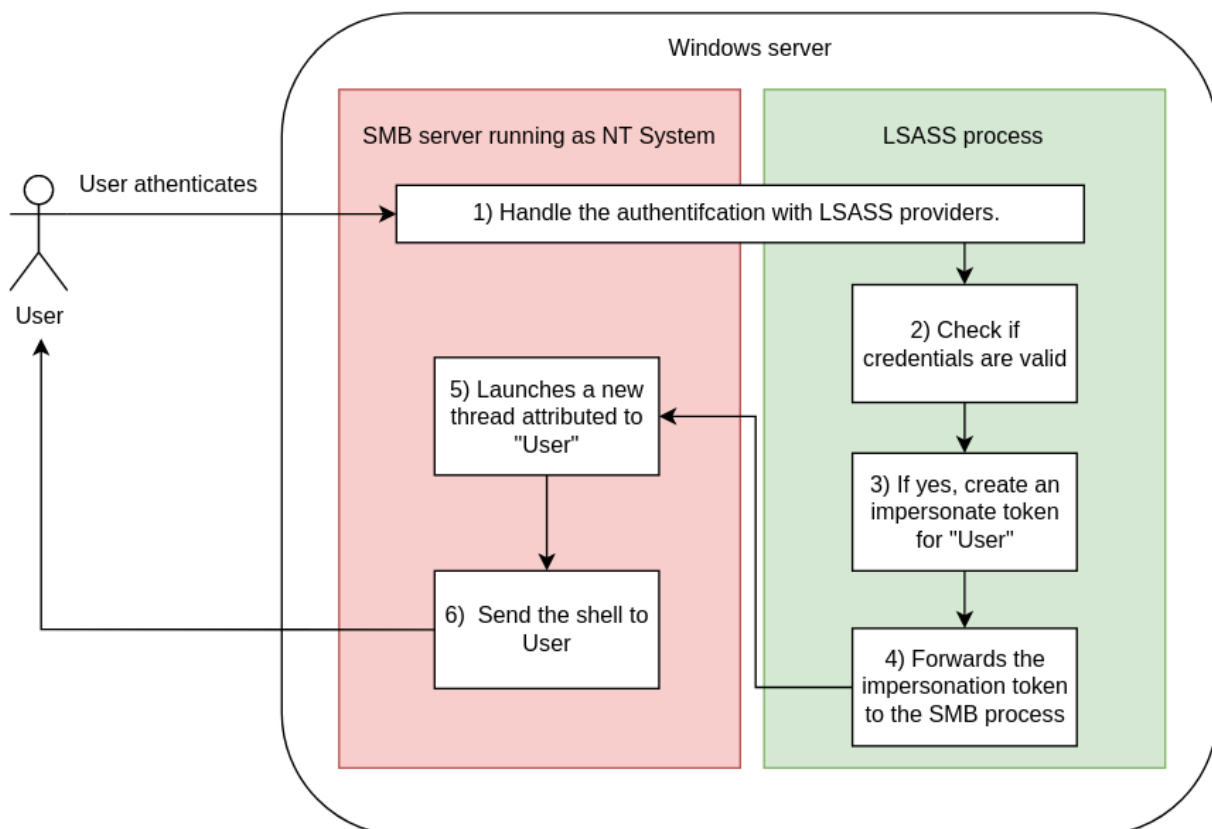
Токен потока, применяемый для заимствования контекста безопасности другого пользователя, называется олицетворяющим токеном (англ. impersonation token).

Таким образом появляется еще одна классификация токена: первоначальный или олицетворяющий.

В связи изложенным можно уточнить определение токена:

**Токен доступа пользователя** - объект ядра ОС Windows, описывающий контекст безопасности процессов или потоков, работающих от имени указанного пользователя.

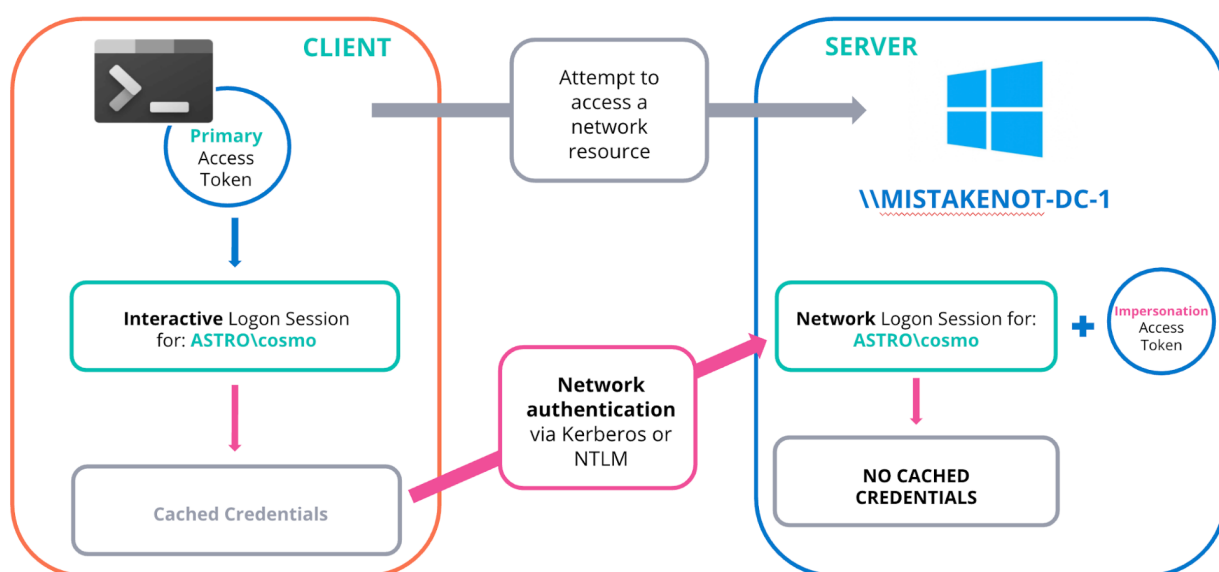
SMB сервер представляет собой многопоточное приложение. Сервер обслуживает каждого клиента в рамках соответствующего выделенного потока, обладающего токеном, олицетворяющим указанного клиента.



Выдача олицетворяющего токена при сетевом входе



В данном случае олицетворяющий токен также будет связан с сессией, возникшей в результате сетевого входа.



Аутентификации при сетевом входе

Еще раз обратим внимание, что токен доступа не содержит в себе никаких учетных данных пользователя. По сути токен используется для идентификации пользователя, так как в нем хранится информация о группах и привилегиях. На основе этой информации принимается решение о предоставлении доступа к защищаемым объектам. Вспомните аналогию с аккредитационной картой. Аккредитация не аутентифицирует, но дает возможность быстро понять куда разрешен доступ.

## Уровни олицетворения

Олицетворяющие токены имеют следующую важную характеристику - уровень олицетворения, который отражает степень того, насколько сервер может олицетворять клиента. Всего существуют следующие четыре уровня олицетворения:

Название	Описание
<b>Anonymous</b>	Удаленный сервер не может идентифицировать или олицетворять клиента. Используется для анонимного подключения.
<b>Identification</b>	Удаленный сервер может идентифицировать, но не может олицетворять клиента.
<b>Impersonation</b>	Удаленный сервер может идентифицировать и олицетворять клиента локально в системе. Подобные токены обычно создаются в результате сетевого входа, например при доступе к FTP серверу.

Название	Описание
<b>Delegation</b>	Удаленный сервер может идентифицировать и олицетворять клиента в удаленных системах. Подобные токены обычно создаются в результате интерактивного входа, например по RDP.

---

## Классификации токенов

---

Обобщая написанное выше, получается, что токен:

1. Содержит перечень групп и привилегий пользователя
2. Может быть полным, а может быть фильтрованным
3. Может быть связан с интерактивной или неинтерактивной сессией, то есть иметь связь с учетными данными или не иметь
4. Может быть первоначальным, а может быть олицетворяющим. Если токен олицетворяющий, то он обладает одним из четырех уровней олицетворения

---

## Заключение

---

В настоящей статье было рассмотрено устройство токенов доступа в Windows. Был представлен обзор основных атрибутов по которым можно классифицировать токены, а также рассказано о связи токенов, сессий и аутентификационных данных. Изложенные сведения полезны для понимания атак, связанных с манипуляциями токенами.

---

## Используемые источники

---

Статьи и презентации из сети Интернет:

Книги:

- “Защита в операционных системах” - Проскурин В.Г.
- Programming Windows Security - Keith Brown
- Windows Internals Pt. 1 - Pavel Yosifovich, Alex Ionescu, Mark Russinovich and David Solomon