МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ „ЛЬВІВСЬКА ПОЛІТЕХНІКА”

УПРАВЛІННЯ ДОСТУПОМ ДО ОБ'ЄКТІВ ФАЙЛОВОЇ СИСТЕМИ У WINDOWS 10

МЕТОДИЧНІ ВКАЗІВКИ
до виконання лабораторної роботи
з дисципліни «Основи системного адміністрування»
для студентів напряму 6.050103 «Програмна інженерія» та
спеціальності 121 «Інженерія програмного забезпечення»

Затверджено
на засіданні кафедри
програмного забезпечення
Протокол №13 від 19.05.2016 р.

Львів – 2016

Управління доступ до об'єктів файлової системи у Windows 10.: Методичні вказівки до виконання лабораторної роботи з дисципліни „Основи системного адміністрування” для студентів напряму 6.050103 «Програмна інженерія» та спеціальності 121 «Інженерія програмного забезпечення» / Укл.: В.С. Яковина, Т.О. Муха, Р.Р. Шкраб – Львів: Видавництво Національного університету „Львівська політехніка”, 2016. – 26 с.

Укладачі Яковина В.С., доцент кафедри ПЗ
Муха Т.О., асистент кафедри ПЗ
Шкраб Р.Р., асистент кафедри ПЗ

Відповідальний за випуск Яковина В.С., завідувач кафедри ПЗ

Рецензенти Тушницький Р.Б., к.т.н., доцент кафедри ПЗ
Литвин В.В., д.т.н., завідувач кафедри ІСМ

Лабораторна робота № 3.
ДОЗВОЛИ ТА КВОТИ NTFS. ШИФРУВАННЯ ФАЙЛІВ: EFS.

Мета роботи: Навчитись ефективно налагоджувати систему логічного розділення доступу до об’єктів файлової системи в ОС Windows 10; управляти квотами на томах NTFS та використовувати шифровану файлову систему EFS.

Теоретичні відомості.

Права доступу (дозволи) для файлів і каталогів

Встановлюючи користувачам певні дозволи на доступ до файлів і каталогів (папок), адміністратори системи можуть захищати конфіденційну інформацію від несанкціонованого доступу. Кожен користувач має певний набір дозволів на доступ до кожного об'єкту файлової системи.

Крім того, користувач може бути власником файлу або папки, якщо сам їх створює. Питання передачі права володіння розглядається нижче.

Дозволи користувача на доступ до об'єктів файлової системи працюють за принципом доповнення (адитивності). Це значить, що діючі дозволи, тобто ті дозволи, які користувач реально має відносно конкретного каталогу або файлу, утворюються зі всіх прямих і непрямих дозволів, призначених користувачу для цього об'єкту за допомогою логічної функції АБО. Наприклад, якщо користувач має прямо призначений дозвіл для каталогу на читання, а опосередковано через членство в групі йому дано дозвіл на запис, то в результаті користувач зможе читати інформацію у файлах каталогу і записувати в них дані.

Слід зазначити, що правило додавання дозволів за допомогою логічного АБО не виконується, коли користувач має певний дозвіл, а групі, в яку він входить, відмовлено в цьому дозволі (або навпаки). В цьому випадку відмова в дозволі (Deny) має вищий пріоритет над наданням дозволу, тобто в результаті користувач не матиме цього дозволу. Поява можливості відмови користувачу або групі в дозволі для файлів і каталогів зробила непотрібним дозвіл No Access, що застосовувався в Windows NT 4.0. Тепер для відмови користувачу в дозволі на доступ до якого-небудь файлу або каталогу слід включити користувача в групу, якій відмовлено в дозволі Full Control (Повний доступ) для даного об'єкту файлової системи[1].

Кожен зі стандартних дозволів складається з набору спеціальних (особливих) дозволів, що задають можливість виконання тієї або іншої дії з

[1] Нагадаємо, що дозволи підтримуються тільки на файловій системі NTFS, тому не дивуйтеся, якщо ви не побачите вкладку Security (Безпека) у властивостях дискових томів, що відформатовані для FAT або FAT32.

файлами або каталогами. У табл. 1 показано відповідність стандартних і спеціальних дозволів для файлів і каталогів. Детальний опис спеціальних дозволів наведено в табл. 2.

Таблиця 1 Відповідність стандартних і спеціальних дозволів

Спеціальні дозволи	**Стандартні дозволи**					
	Full Control	**Modify**	**Read & Execute**	**List Folder Contents**	**Read**	**Write**
Traverse Folder/Execute File (Огляд папок/ Виконання файлів)	+	+	+	+		
List Folder/Read Data (Зміст папок/ Читання даних)	+	+	+	+	+	
Read Attributes (Читання атрибутів)	+	+	+	+	+	
Read **Extended Attributes** (Читання додаткових атрибутів)	+	+	+	+	+	
Creat Files/Write Data (Створення файлів/Запис даних)	+	+				+
Create Folders/Append Data (Створення папок/ Дозапис даних)	+	+				+
Write Attributes (Запис атрибутів)	+	+				+
Write Extended Attributes (Запис додаткових атрибутів)	+	+				+
Delete Subfolders and Files (Видалення підпапок і файлів)	+					
Delete (Видалення)	+	+				
Read Permissions (Читання дозволів)	+	+	+	+	+	+
Change Permissions (Зміна дозволів)	+					
Take Ownership (Зміна власника)	+					

Хоча дозволи "List Folder Contents" ("Список вмісту папки") і "Read & Execute" ("Читання і виконання") включають одні і ті ж спеціальні дозволи,

вони успадковуються по-різному. Дозвіл "List Folder Contents" успадковується тільки каталогами, але не файлами, і відображається тільки при перегляді дозволів на доступ до папок. Дозвіл "Read & Execute" успадковується як файлами, так і папками, і завжди відображається при перегляді дозволів на доступ до файлів або папок.

Таблиця 2. Спеціальні дозволи для файлів і каталогів

Спеціальний дозвіл	**Опис**
Traverse Folder / Execute File	Визначає можливість переміщення по каталогам файлової системи незалежно від того, має або не має користувач дозволу для перегляду каталогів, що перетинаються в процесі переміщення. На роботу цього дозволу впливає політика безпеки Bypass Traverse Checking (Обхід перехресної перевірки) (див. вузол **Local Policies | User Rights Assignment** в параметрах безпеки). Дозвіл Execute File (Виконання файлів) визначає можливість виконання програм.
List Folder / Read Data	Визначає можливість перегляду імен файлів або підкаталогів даного каталогу (відноситься тільки до каталогу). Дозвіл Read Data (Читання даних) визначає можливість перегляду вмісту файлу.
Read Attributes	Визначає можливість перегляду атрибутів файлу або каталогу. Самі атрибути визначаються операційною системою.
Read Extended Attributes	Визначає можливість перегляду додаткових атрибутів файлу або каталогу. Самі додаткові атрибути визначаються операційною системою.
Create Files /Write Data	Визначає можливість створення файлів усередині каталогу (відноситься тільки до каталогів). Дозвіл Write Data (Запис даних) визначає можливість зміни вмісту файлів або перезапису існуючих даних файлу новою інформацією (відноситься тільки до файлів).
Create Folders / Append Data	Визначає можливість створювати підкаталоги усередині даного каталогу (відноситься тільки до каталогів). Дозвіл Append Data (Дозапис даних) визначає можливість приєднання нових даних до існуючого файлу без зміни, знищення або перезапису існуючої інформації (відноситься тільки до файлів).
Write Attributes	Визначає можливість зміни атрибутів файлу або каталогу. Атрибути визначаються операційною системою.
Write Extended Attributes	Визначає можливість зміни додаткових атрибутів файлу або каталогу. Додаткові атрибути визначаються програмою і можуть бути нею змінені.

Delete Subfolders and Files	Визначає можливість видалення підкаталогів і файлів, що знаходяться в даному каталозі, навіть якщо для цих підкаталогів і файлів немає дозволу Delete (Видалення). Цей дозвіл є тільки у каталогів.
Delete	Визначає можливість видалення файлу або каталогу. Якщо вам відмовлено в дозволі Delete (Видалення) для даного каталогу або файлу, ви все ж таки можете видалити їх, одержавши дозвіл Delete Subfolders and Files (Видалення підпапок і файлів) на батьківський каталог.
Read Permissions	Визначає можливість читання дозволів для файлів і каталогів, таких як Full Access, Read і т.д.
Change Permissions	Визначає можливість зміни дозволів для файлів і каталогів, таких як Full Access, Read і т.д.
Take Ownership	Визначає можливість взяття у власність даного файлу або каталогу. Власник файлу або каталогу може завжди змінити дозволи до цього об'єкту, незалежно від інших дозволів.

Визначення діючих дозволів для файлів і папок

Як вже зазначалось, користувач або група одержують дозволи на доступ до файлів або папок безпосередньо і через членство в групах (в доменах Windows 2000 і Windows Server 2003 групи можуть бути членами інших груп). Тому, коли виникає питання "а які ж врешті-решт права має даний користувач?" відповідь одержати не так просто і швидко. Системи Windows 10 і Windows Server 2008 пропонують нову можливість – визначення діючих дозволів (effective permissions).

При визначенні дозволів, що діють, потрібно пам'ятати про те, що користувач може мати права на операції з об'єктом, призначені не за допомогою механізму дозволів, а через політики безпеки. Наприклад, користувач може змінювати власника об'єкту, навіть якщо у нього немає дозволу Take Ownership, проте якщо таке право йому дане в політиках безпеки.

Квоти дискового простору

Адміністрування великих комп'ютерних мереж, де сервери підтримують роботу сотень користувачів, зв'язане з рядом складнощів. Одна з них – облік дискового простору серверу, зайнятого файлами співробітників компанії. Як правило, користувачі, що зберігають свої файли на сервері, мало піклуються про актуальність інформації і про знищення застарілих або непотрібних даних. Безліч тимчасових файлів і копій одного й того ж файлу, що знаходяться в різних папках, лише посилюють ситуацію. В результаті в лічені місяці навіть

на великих жорстких дисках серверу може не виявитися необхідного для роботи вільного простору.

Як правило, у великих організаціях дерево каталогів вельми розгалужене, тому візуальний контроль витрачання дискового простору користувачами віднімає у адміністраторів багато часу і зусиль.

Подібна проблема просто розв'язується за допомогою введення квот на дисковий простір, доступний для роботи кожному користувачу. У попередніх версіях операційної системи Windows NT не було штатних можливостей ввести квоту на доступний дисковий простір, тому будь-який користувач міг розпоряджатися усім простором жорстких дисків комп'ютера. У Windows 10 і Windows Server 2008 адміністратор може квотувати дисковий простір по **кожному тому** і для **кожного користувача**. (З цього випливає, що неможливо задати квоту для окремих каталогів або груп.)

Система враховує загальний простір, зайнятий файлами, власником яких є контрольований користувач: якщо користувач володіє файлом, розмір останнього додається до загальної суми зайнятого користувачем дискового простору. Важливо відзначити, що, оскільки квотування виконується на рівні тому, не має значення, чи знаходиться том на одному фізичному жорсткому диску або на різних пристроях. І навпаки, якщо на одному фізичному диску зберігається декілька томів, то квотування може здійснюватися індивідуально по кожному тому.

Після установки квот дискового простору користувач зможе зберігати на томі обмежений об'єм даних, тоді як на цьому томі може залишатися вільний простір. Якщо користувач перевищує видану йому квоту, в журнал подій вноситься відповідний запис. Потім, залежно від конфігурації системи, користувач або зможе записати інформацію на том ("м'який" режим обмежень), або йому буде відмовлено в записі через відсутність вільного простору ("жорсткий" режим).

Квоти можна використовувати на локальних і спільних мережних дисках (в цьому випадку загальний доступ повинен бути дозволений на рівні кореневого каталогу тому). Стиснення файлів не має значення при обчисленні зайнятого простору – завжди враховується розмір початкового нестиснутого файлу.

Встановлювати і проглядати[1] квоти на диску можна тільки в розділі з NTFS 5.0 і за наявності необхідних повноважень (що задаються за допомогою локальних або доменних групових політик) у користувача, що встановлює

[1] У Windows 10 і Windows Server 2008 для виконання будь-яких операцій з квотами можна також використовувати утиліту командного рядка Fsutil.exe.

квоти. За умовчанням для роботи з квотами потрібно бути членом групи Administrators.

Передача права володіння

У попередніх версіях Windows NT право володіння файлом або папкою було характеристикою, жорстко прив'язаною до творця даного об'єкту. Користувач, що створив файл або каталог, ставав власником цього об'єкту. Право володіння не могло бути передане іншому користувачу. Єдине виключення складав адміністратор, який міг стати власником об’єкту. Сам користувач не міг передати право володіння папкою або файлом іншому користувачу.

Операційні системи Windows 10 і Windows Server 2008 забезпечують гнучкіше користування таким засобом забезпечення безпеки інформації, як право володіння об'єктом файлової системи. Тепер адміністратор або уповноважений користувач може призначити будь-якого користувача власником якого-небудь об'єкту файлової системи (ця можливість є тільки в Windows Server 2008) або користувач сам може стати власником об'єкту, не створеного ним самим. Природно, що для цього він повинен мати необхідні дозволи (Take Ownership). Крім того, за допомогою локальних або доменних політик безпеки можна вказувати, які користувачі завжди можуть ставати власниками файлів або інших об'єктів (за умовчанням таке право мають тільки адміністратори), при цьому вони можуть навіть не мати ніяких дозволів для цього об'єкту[1].

Для передачі володіння об'єктом файлової системи або для проглядання поточного власника файлу або папки слід відкрити відповідне вікно властивостей, перейти на вкладку **Security**, потім натиснути кнопку **Advanced**. З'явиться вікно **Advanced Security Settings** (рис. 1).

[1] З вищесказаного випливає принцип відновлення дозволів для будь-якого об'єкту (зокрема для об'єктів, у яких помилково видалені всі дозволи): потрібно стати його власником, а потім встановити потрібні дозволи для інших користувачів і груп.

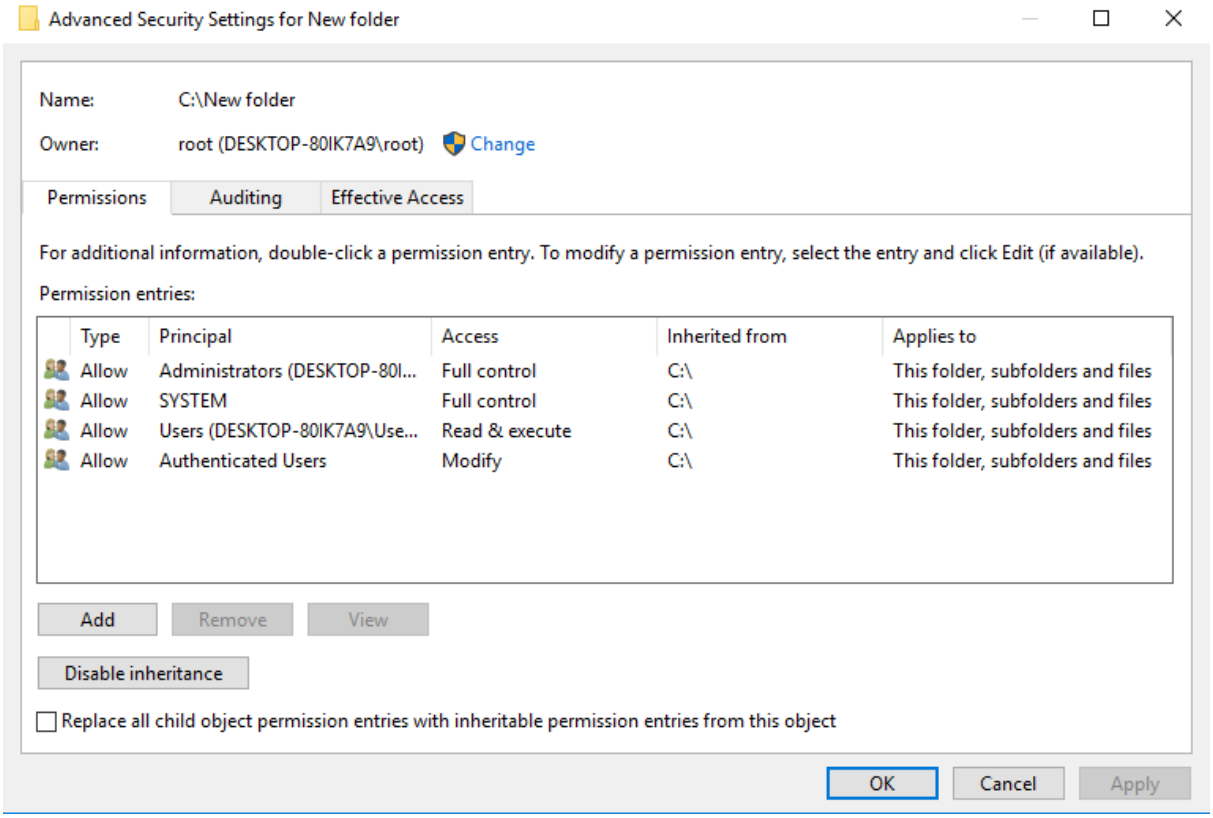

Рис.1. Діалогове вікно Advanced Security Settings

У Windows Server 2008 адміністратор або інший користувач, що отримав (через відповідну політику безпеки) право зміни власників об'єктів, може натиснути кнопку **Other Users or Groups** (Інші користувачі або групи) і вибрати користувача (або групу), якого він хоче призначити власником об'єкту.

Шифруюча файлова система EFS

На персональному комп'ютері операційну систему можна завантажити не з жорсткого, а з гнучкого чи компакт диска. Це дозволяє обійти проблеми, пов'язані з відмовою жорсткого диска і руйнуванням завантажувальних розділів. Проте, оскільки за допомогою змінного носія можна завантажувати різні операційні системи, будь-який користувач, що одержав фізичний доступ до комп'ютера, може обійти вбудовану систему управління доступом файлової системи NTFS і за допомогою певних інструментів прочитати інформацію жорсткого диска.

Єдиний надійний спосіб захисту інформації – це шифруюча файлова система. На ринку програмного забезпечення існує цілий набір продуктів, що забезпечують шифрування даних за допомогою утвореного від пароля ключа на рівні додатків. Проте такий підхід має ряд обмежень.

- Ручне шифрування і дешифрування. Служби шифрування більшості продуктів непрозорі для користувачів. Користувачу доводиться розшифровувати файл перед кожним його використанням, а потім знову зашифровувати. Якщо користувач забуває зашифрувати файл після закінчення роботи з ним, інформація залишається незахищеною. Оскільки кожного разу необхідно вказувати, який файл повинен бути зашифрований (і розшифрований), застосування такого методу захисту інформації сильно утруднене.
- Просочування інформації з тимчасових файлів і файлів підкачки. Практично всі додатки в процесі редагування документів створюють тимчасові файли. Вони залишаються на диску незашифрованими, не зважаючи на те, що оригінальний файл зашифрований. Крім того, шифрування інформації на рівні додатків виконується в режимі користувача. Це значить, що ключ, вживаний для такого типу шифрування, може зберігатися у файлі підкачки. В результаті, за допомогою вивчення вмісту файлу підкачки можна одержати ключ і розшифрувати всі документи користувача.
- Слабка криптостійкість ключів. Ключі утворюються від паролів або випадкових фраз. Тому у випадку, якщо пароль легко запам'ятовується, атаки за допомогою словників можуть привести до швидкого злому системи захисту.
- Неможливість відновлення даних. Більшість продуктів, що дозволяють шифрувати інформацію, не надають засобів відновлення даних, що для користувачів є додатковим мотивом не застосовувати засоби шифрування. Це особливо стосується тих працівників, які не хочуть запам'ятовувати додатковий пароль. З другого боку, засіб відновлення даних за допомогою пароля – ще одна прогалина в системі захисту інформації. Все, що необхідно зловмиснику, – це пароль, призначений для запуску механізму відновлення даних, який дозволить отримати доступ до зашифрованих файлів.

Всі перераховані вище проблеми дозволяє вирішити шифруюча файлова система (Encrypting File System, EFS), вперше реалізована у Windows 2000, яка працює тільки на NTFS 5.0.

EFS містить наступні компоненти операційної системи (рис. 2):

- Драйвер EFS. Драйвер EFS є надбудовою над файловою системою NTFS. Він обмінюється даними зі службою EFS – запитує ключі шифрування, набори DDF (Data Decryption Field) і DRF (Data Recovery Field), – а також з іншими службами управління ключами. Одержану інформацію драйвер EFS передає бібліотеці реального часу файлової

системи EFS (File System Run-Time Library, FSRTL), яка прозоро для операційної системи виконує різні операції, характерні для файлової системи (читання, запис, відкриття файлу, приєднання інформації).

- Бібліотека реального часу файлової системи EFS. FSRTL – це модуль, який знаходиться усередині драйвера EFS і реалізовує виклики NTFS, що виконують такі операції, як читання, запис і відкриття зашифрованих файлів і каталогів, а також операції, пов'язані з шифруванням, дешифруванням і відновленням файлів при їх читанні або запису на диск. Хоча драйвери EFS і FSRTL реалізовані у вигляді одного компоненту, вони ніколи не обмінюються даними безпосередньо. Для передачі повідомлень один одному вони використовують механізм викликів (callouts) NTFS, призначений для управління файлами. Це гарантує, що вся робота з файлами відбувається за безпосередньої участі NTFS.

Рис. 2. Архітектура EFS

- Служба EFS. Служба EFS (EFS Service) є частиною системи безпеки операційної системи. Для обміну даними з драйвером EFS вона використовує порт зв'язку LPC, що існує між локальним адміністратором безпеки (Local Security Authority, LSA) і монітором безпеки, що працює в привілейованому режимі. У режимі користувача

для створення ключів шифрування файлів і генерування даних для DDF і DRF служба EFS використовує CryptoAPI. Вона також підтримує набір API для Win32.

- Набір API для Win32. Цей набір інтерфейсів прикладного програмування дозволяє виконувати шифрування файлів, дешифровку і відновлення зашифрованих файлів, а також їх імпорт і експорт (без попередньої дешифровки). Ці API підтримуються стандартним системним модулем DLL – advapi32.dll.

EFS заснована на шифруванні з відкритим ключем і використовує всі можливості архітектури CryptoAPI. Кожен файл шифрується за допомогою випадково генерованого ключа, залежного від пари відкритого (public) і приватного (private) ключів користувача. Подібний підхід значною мірою утруднює здійснення великого набору атак, заснованих на криптоаналізі. При криптозахисті файлів може бути застосований будь-який алгоритм симетричного шифрування. EFS дозволяє здійснювати шифрування і дешифрування файлів, що знаходяться на віддалених файлових серверах.

Дані зашифровуються за допомогою симетричного алгоритму із застосуванням ключа шифрування файлу (File Encryption Key, FEK). FEK – це генерований випадковим чином ключ певної довжини. Своєю чергою, FEK шифрується за допомогою одного або декількох відкритих ключів, призначених для криптозахисту ключа. В цьому випадку створюється список зашифрованих ключів FEK, що дозволяє організувати доступ до файлу декільком користувачам. Для шифрування набору FEK використовується відкрита частина пари ключів кожного користувача. Список зашифрованих ключів FEK зберігається разом із зашифрованим файлом в спеціальному атрибуті EFS, званому полем дешифрування даних (Data Decryption Field, DDF). Інформація, необхідна для дешифрування, прив'язується до самого файлу. Приватна частина ключа користувача використовується при дешифруванні FEK. Вона зберігається в безпечному місці, наприклад на смарт-карті або іншому пристрої, що має високий ступінь захищеності.

FEK застосовується для створення ключів відновлення. Для цього FEK шифрується за допомогою одного або декількох відкритих ключів відновлення. Список FEK, зашифрованих для цілей відновлення, зберігається разом із зашифрованим файлом в спеціальному атрибуті EFS, званому полем відновлення даних (Data Recovery Field, DRF). Завдяки існуванню набору зашифрованих ключів FEK агенти відновлення даних можуть дешифрувати файл. Для шифрування ключа FEK в полі DRF необхідна тільки відкрита частина пари ключів відновлення, її присутність в системі необхідна у будь-який момент часу для нормального функціонування файлової системи. Сама

процедура відновлення виконується досить рідко, коли користувач звільняється з організації або забуває приватну частину ключа. Тому агенти відновлення можуть зберігати приватну частину ключів відновлення в безпечному місці, наприклад на смарт-картах або інших добре захищених пристроях.

EFS тісно взаємодіє з NTFS 5.0. Тимчасові файли, створювані додатками, успадковують атрибути оригінальних файлів (якщо файли знаходяться в розділі NTFS). Разом з файлом шифруються також і його тимчасові копії. EFS знаходиться в ядрі Windows і використовує для зберігання ключів спеціальний пул, що не вивантажується на жорсткий диск. Тому ключі ніколи не потрапляють у файл підкачки.

У Windows Server 2008 файлова система EFS має деякі нові можливості:

- із зашифрованими файлами можуть працювати декілька користувачів. Користувач, що зашифрував файл, може дозволити іншим локальним і доменним користувачам (на комп'ютерах під управлінням Windows XP і Windows 10) працювати з цим файлом;
- можна шифрувати автономні папки і файли (offline folders);
- агент відновлення (recovery agent) за умовчанням не використовується;
- стандартний алгоритм шифрування – Advanced Encryption Standard, AES (Rijndael) (256 біт). Алгоритм DESX, що використовується за умовчанням системою EFS в Windows XP і Windows 10, не може застосовуватися для шифрування файлів в Windows Server 2008;
- замість AES може використовуватися алгоритм шифрування 3DES (128 або 168 біт), для цього потрібно змінити політику безпеки;
- зашифровані файли можуть розташовуватися у веб-папках;
- сертифікати EFS можуть автоматично доставлятися користувачу службами сертифікатів (Certificate Services) і механізмом автоматичного підпису сертифікатів;
- особисті ключі можуть зберігатися і відновлюватися за допомогою засобів архівації, що є в службах сертифікатів;
- вся службова інформація, що зберігається на диску, не просто видаляється, а очищується (заповнюється порожніми байтами); це збільшує захищеність шифрованих даних.

Конфігурація EFS, встановлювана за умовчанням, дозволяє користувачу шифрувати свої файли без жодного втручання з боку адміністратора. В цьому випадку EFS автоматично генерує для користувача пару ключів (відкритий і особистий), що застосовуються для криптозахисту даних, і підписує сертифікат.

Шифрування і дешифрування файлів може бути виконане як для певних файлів, так і для цілого каталогу. Ці операції прозорі для користувача. При шифруванні каталогу автоматично шифруються і всі його файли і підкаталоги. Кожен файл має унікальний ключ, що дозволяє легко виконувати операцію перейменування. Якщо ви перейменовуєте файл, що знаходиться в зашифрованому каталозі, і переносите його в незашифрований каталог, сам файл залишається зашифрованим (за умови, що цільовий каталог знаходиться на томі NTFS 5.0). Засоби шифрування і дешифрування доступні через Windows Explorer. Крім того, можна використовувати всі можливості шифрування даних за допомогою набору утиліт командного рядка і інтерфейсів адміністрування.

Найсерйозніша і, на жаль, доволі поширена помилка при роботі з EFS полягає у тому, що користувачі шифрують дані на локальному комп'ютері, а потім встановлюють заново операційну систему. В цьому випадку **дані будуть безповоротно втрачені**, оскільки доступ до них мали тільки два користувачі тієї системи, в якій дані були зашифровані: користувач, що виконав цю операцію, і агент відновлення. Помилка полягає у тому, що для розшифрування даних необхідно пред'явити сертифікати одного з названих користувачів, а для цього відповідні сертифікати потрібно було експортувати і зберегти.

EFS має в своєму розпорядженні вбудовані засоби відновлення зашифрованих даних за умов, коли невідомий приватний ключ користувача. Користувачі, які можуть відновлювати зашифровані дані за умов втрати приватного ключа, називаються агентами відновлення даних[1]. Агенти відновлення даних мають сертифікат (X.509 v.3) на відновлення файлів і особистий ключ, за допомогою яких виконується операція відновлення зашифрованих файлів. Використовуючи ключ відновлення, можна одержати тільки генерований випадковим чином ключ, за допомогою якого був зашифрований конкретний файл. Тому агенту відновлення не може випадково стати доступною інша конфіденційна інформація.

[1] Політики відновлення в Windows 10/2008 працюють інакше, ніж в Windows 2000. За умовчанням на комп'ютерах під управлінням Windows Server 2008 агенти відновлення не створюються і політика відновлення не перешкоджає роботі EFS. Це означає, що відновити зашифровану інформацію можуть тільки ті користувачі, які її зашифрували.

Завдання до виконання роботи

1. Створити на томі NTFS нову папку; у властивостях об’єкту перейти на вкладку "Security" (рис. 3). (Звернути увагу на те, які дозволи призначені за замовчуванням.) Для того, щоб дозволити усім користувачам створювати файли і папки у цій папці, але заборонити її видаляти, а також для перегляду та редагування елементарних дозволів, натисніть кнопку "Advanced". Щоб додати елементарний дозвіл для користувача чи групи натисніть "Add" (рис. 4); додайте суб’єкта безпеки та задайте для нього дозволи (рис. 5). (Зверніть увагу на меню "Apply", яке управляє об’єктами безпеки, на які поширюватиметься заданий дозвіл!)

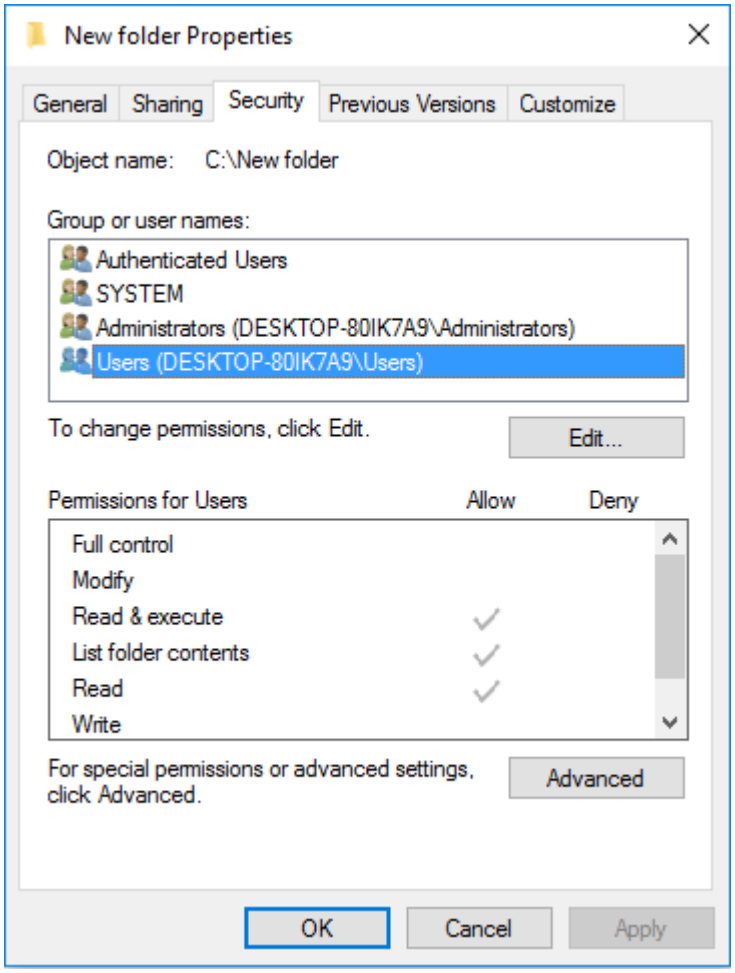

Рис. 3. Вкладка "Security" властивостей об'єкту файлової системи.

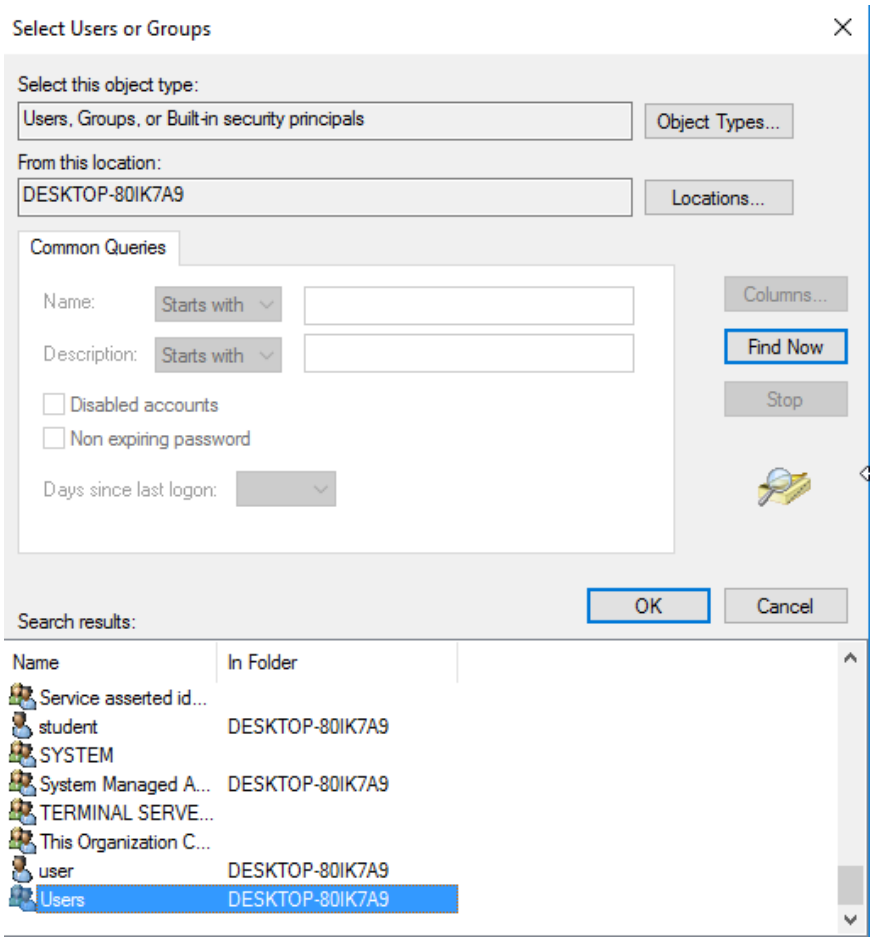

Рис. 4. Вибір суб’єкту безпеки.

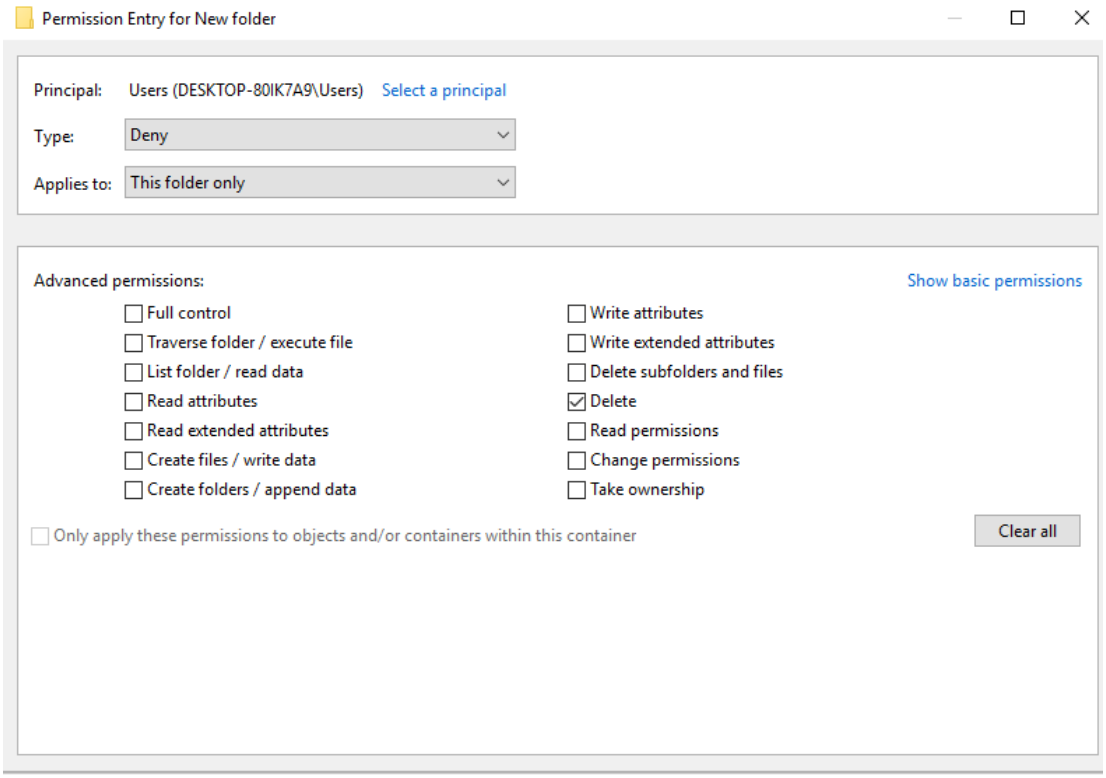

Рис. 5. Вікно спеціальних дозволів.

2. Увійти до системи під іншим користувачем (за необхідності створивши його) та переконатися у неможливості видаляти цю папку (рис. 6), однак маючи змогу створювати в ній файли та підпапки. Створити таку підпапку; переконатись, що як власник об’єкту цей користувач має право на зміну усіх дозволів. Заборонити групі "Administrators" повний доступ до створеної підпапки.

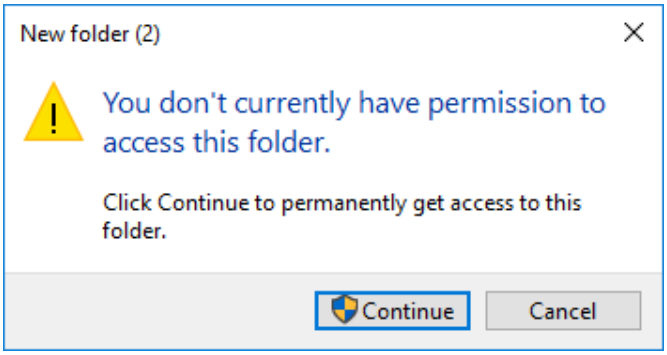

Рис. 6. Повідомлення системи про відмову в доступі на виконання операції.

3. Увійти до системи як користувач з адміністративними повноваженнями. Переконатись у відсутності доступу до підпапки, створеної у ході виконання п. 2 лабораторної роботи. Зайти на вкладку "Security" властивостей об’єкту. У звіті до лабораторної роботи пояснити отримане повідомлення (рис. 7). В додаткових параметрах безпеки змінити власника папки (див рис. 1). Звернути увагу на можливість зміни власника як тільки для поточного об’єкту, так і для субконтейнерів і об’єктів. Переглянути список управління доступом після зміни власника – які суб’єкти та які права доступу є в цьому списку?

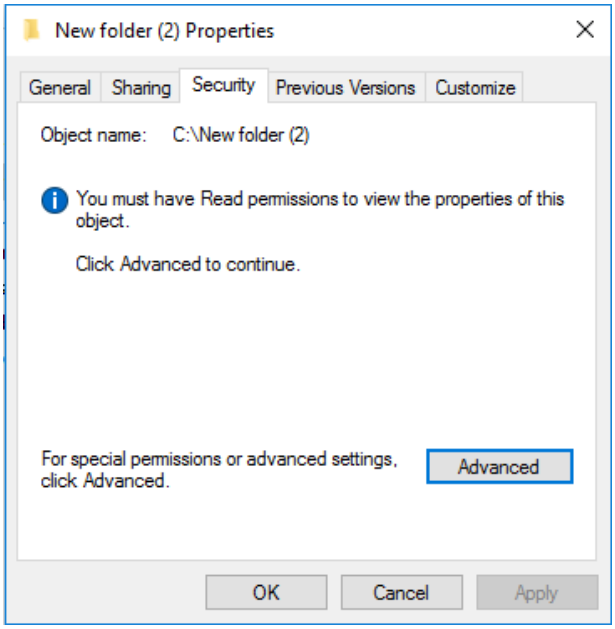

Рис. 7. Повідомлення системи безпеки.

4. Для приведення налаштування безпеки усіх вкладених субконтейнерів і об’єктів до шаблонного зразка батьківської папки використати відмітку "Replace all child object permission…" вкладки "Advanced" додаткових параметрів безпеки для папки (рис. 8).

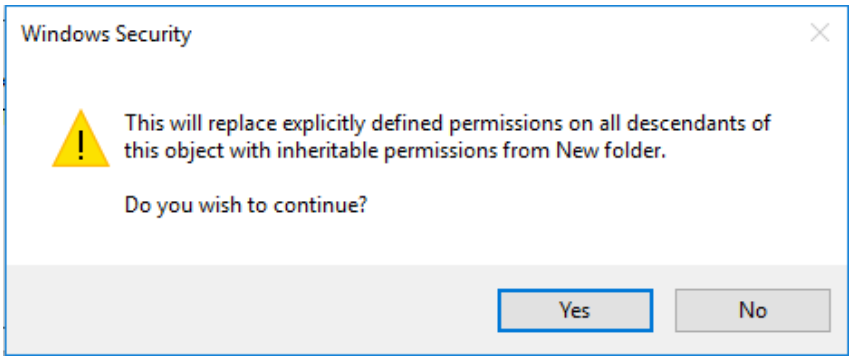

Рис. 8. Повідомлення системи безпеки при заміні дозволів для усіх дочірніх об’єктів.

5. Користувачем з адміністративними правами перейти на вкладку "Quota" (рис. 9) властивостей NTFS тому (за необхідності конвертувати існуючий FAT том, або створити новий). Включити управління квотами у "м’якому" режимі. У вікні "Quota Entries" (рис.10) скориставшись меню "Quota" створити індивідуальний запис квоти для певного користувача (рис. 11).

Увійти до системи під іншим користувачем. Записати на том з квотами файли сумарним обсягом більшим від порогу попереджень, але меншим за розмір квоти.

Увійти до системи під третім користувачем. Записати на том з квотами файли сумарним обсягом більшим за розмір квоти.

Користувачем з адміністративними правами переглянути записи квот для цього тому (рис. 12). Включити квоти у "жорсткому" режимі (для цього поставити відмітку "Deny disk space to users exceeding quota limit").

Увійти до системи під іншим користувачем. Спробувати записати на том з квотами файли сумарним обсягом більшим за розмір квоти. Переконатись у неможливості такої операції (рис. 13).

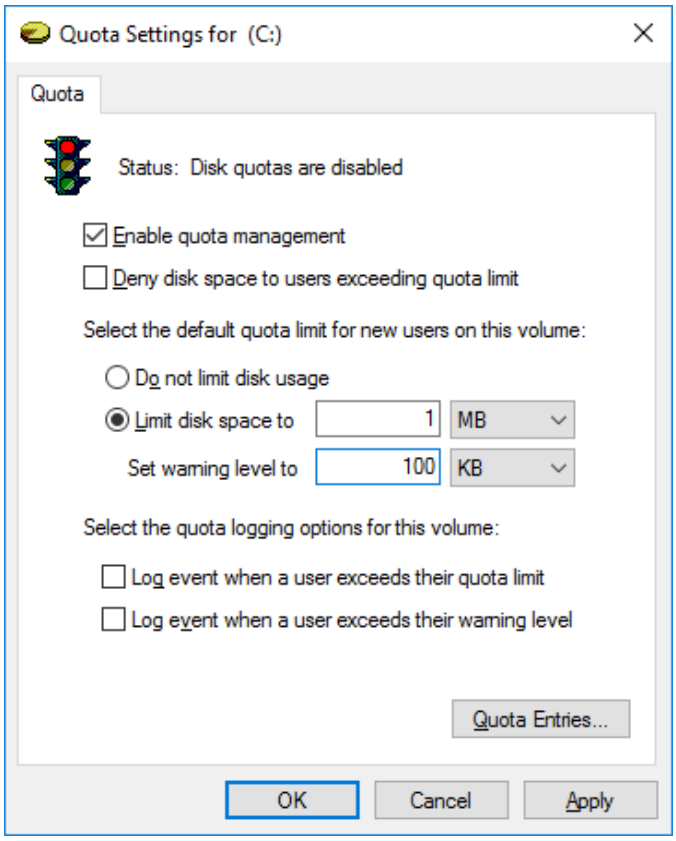

Рис. 9. Вкладка квот у властивостях дискового тому.

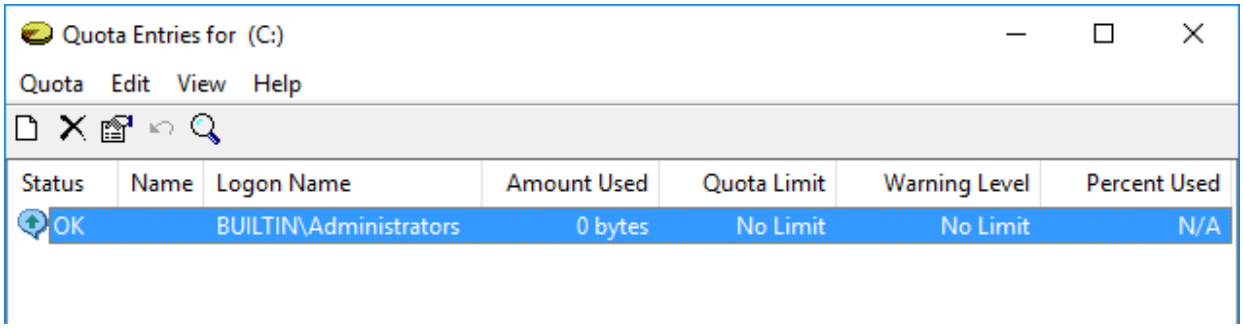

Рис. 10. Вікно записів квот для дискового тому.

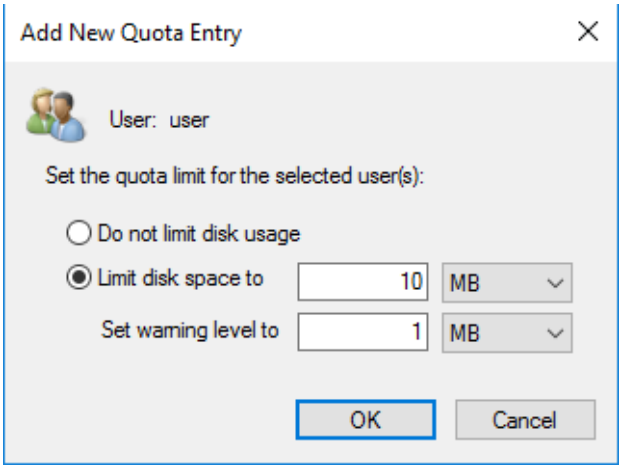

Рис. 11. Додавання нового запису квоти.

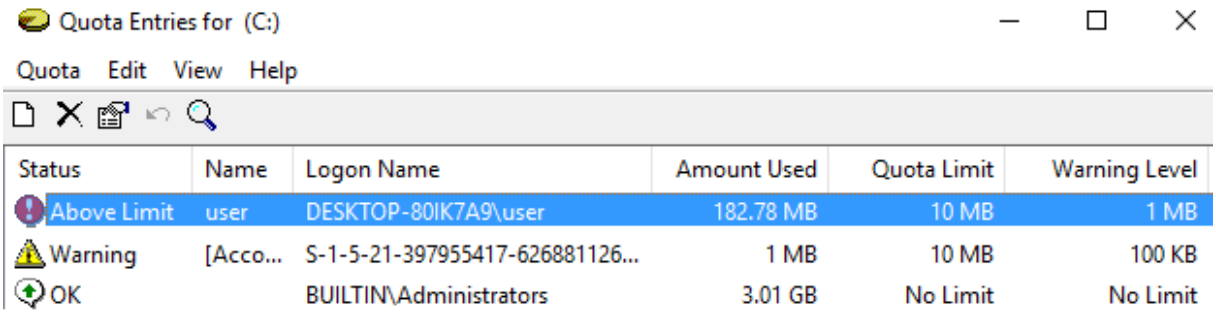

Рис. 12. Вікно записів квот для дискового тому (один користувач перевищив поріг квоти, а один – поріг попереджень).

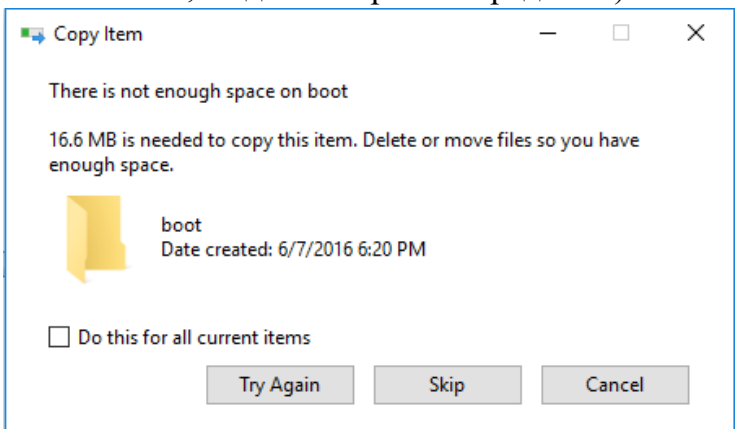

Рис. 13. Повідомлення системи при спробі записати файли понад квоту.

6. Створити агента відновлення шифрованих даних. Для цього виконати наступні кроки:

I. Створення сертифікату агента відновлення:

- Увійти в систему як адміністратор.
- У вікні консолі ввести команду cipher /R:Ім’яФайлу – без розширення (рис. 14).
- Ввести і підтвердити пароль, о захищає приватний ключ.

В поточному каталозі будуть створені два файли: з розширенням cer (містить тільки згенерований ключ) і з розширенням pfx (містить і ключ, і сертифікат агента відновлення).

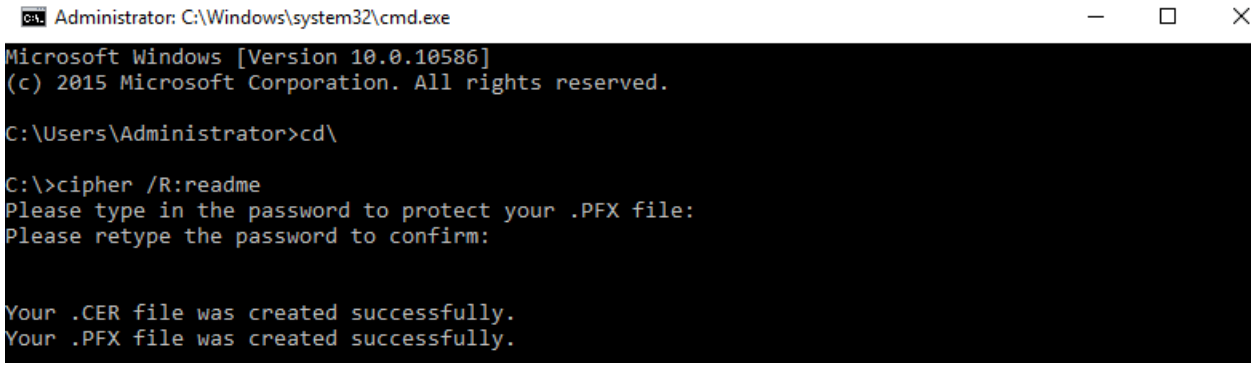

Рис. 14. Створення сертифікату агента відновлення.

II. Імпорт сертифікату, за допомогою якого можна відновлювати індивідуальні файли користувачів:

- Відкрийте оснастку **Certificates**, вузол **Personal** (рис. 15).
- Імпортуйте створений **PFX**-файл (зверніть увагу на розширення сертифікату!).

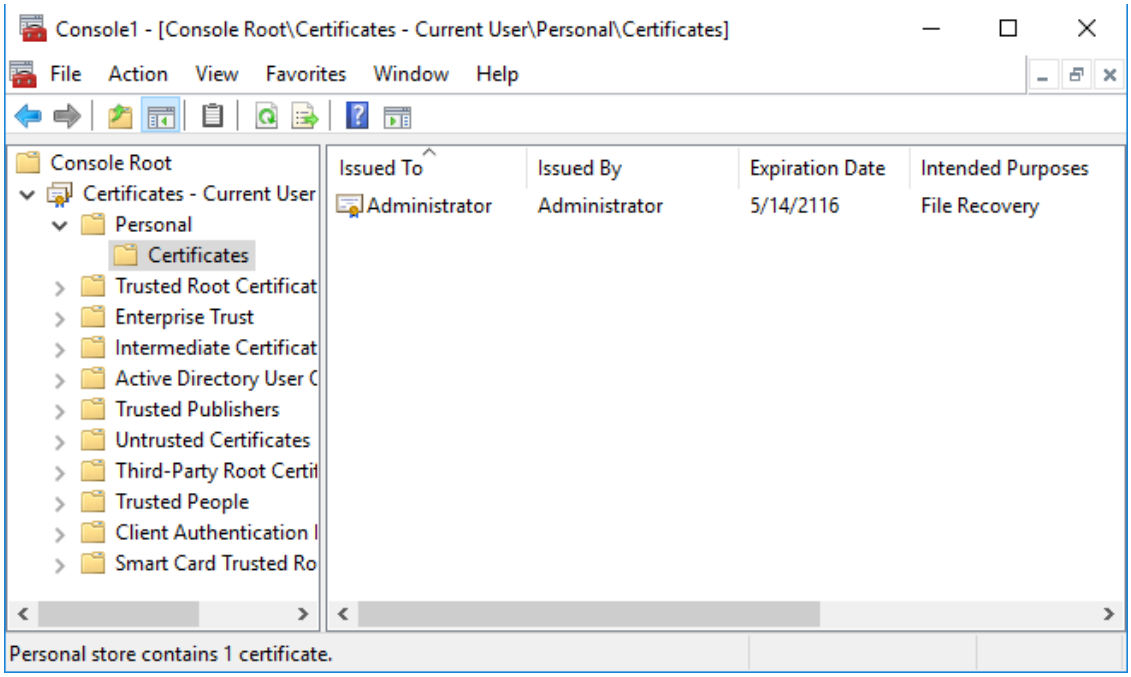

Рис. 15. Імпорт сертифікату агента відновлення.

III. Призначення політики агента відновлення для операцій шифрування:

- Запустіть оснастку **Group Policy**.
- Виберіть вузол **Public Key Policies | Encrypting File System** (рис. 16).
- В контекстному меню виконайте команду **Add Data Recovery Agent**.
- У вікні майстра Welcome to the **Add Data Recovery Agent Wizard**.
- натисніть кнопку **Browse Folders** і вкажіть шлях до створеного раніше файлу з розширенням cer.
- Завершіть роботу майстра натиснувши кнопки **Next** та, на наступній сторінці, **Finish**.

Сертифікат буде імпортовано, а його власник стане агентом відновлення шифрованих даних на цьому комп'ютері. (Зверніть увагу, що в стовпчику **Intendend Purposes** імпортованого сертифікату вказано **File Recovery**).

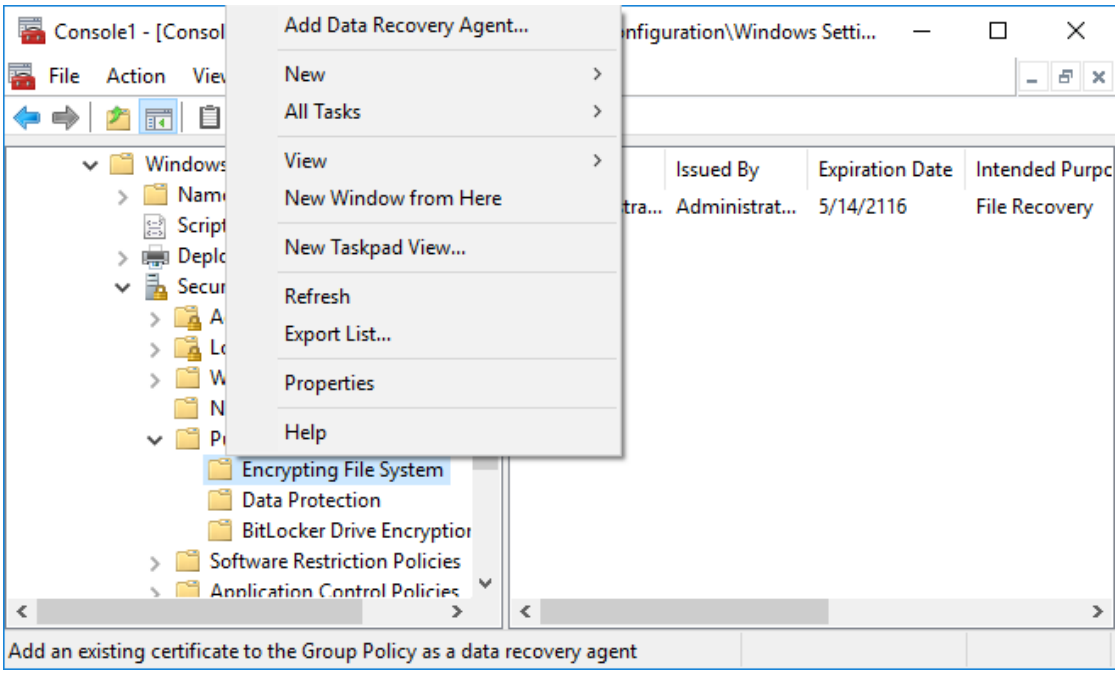

Рис. 16. Створення політики агента відновлення шифрованих даних.

7. Увійшовши до системи під декількома користувачами зашифруйте по одному файлу для кожного користувача. Для цього у властивостях файлу (вкладка "General") натисніть кнопку "Advanced" і поставте відмітку "Encrypt contents to secure data" (рис. 17). Переконайтесь, що кожен користувач може розшифрувати тільки свої файли, а після шифрування першого файлу створюється сертифікат з призначенням **EFS** (перевірити за допомогою оснастки **Certificates**).

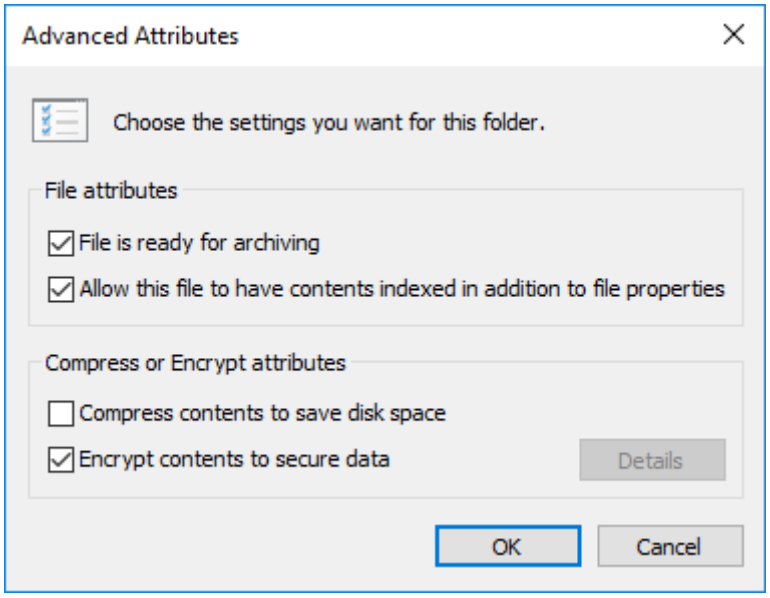

Рис. 17. Вікно встановлення додаткових атрибутів файлу.

При потребі спільного доступу декількох користувачів до одного шифрованого файлу у вікні додаткових атрибутів файлу оберіть кнопку "Details" (рис. 18). Після цього у вікні "Users who can access this file" можна додати користувачів, яким дозволено розшифровувати цей файл (основною умовою є наявність в системі сертифікатів цих користувачів). Крім того, зверніть увагу, що після призначення політики агента відновлення шифрованих даних у вікні деталей щодо шифрування кожного файлу міститься ім’я агента відновлення (рис. 18). Переконайтесь, що користувачі, яким дозволили спільний доступ, можуть читати (тобто розшифровувати) цей файл, в той як інші – ні.

(Якщо треба надати певному користувачу доступ до ВСІХ шифрованих файлів іншого користувача, можна експортувати сертифікат користувача, що надає свої файли – але обов’язково у pfx форматі – і імпортувати цей файл іншому користувачу для надання доступу.)

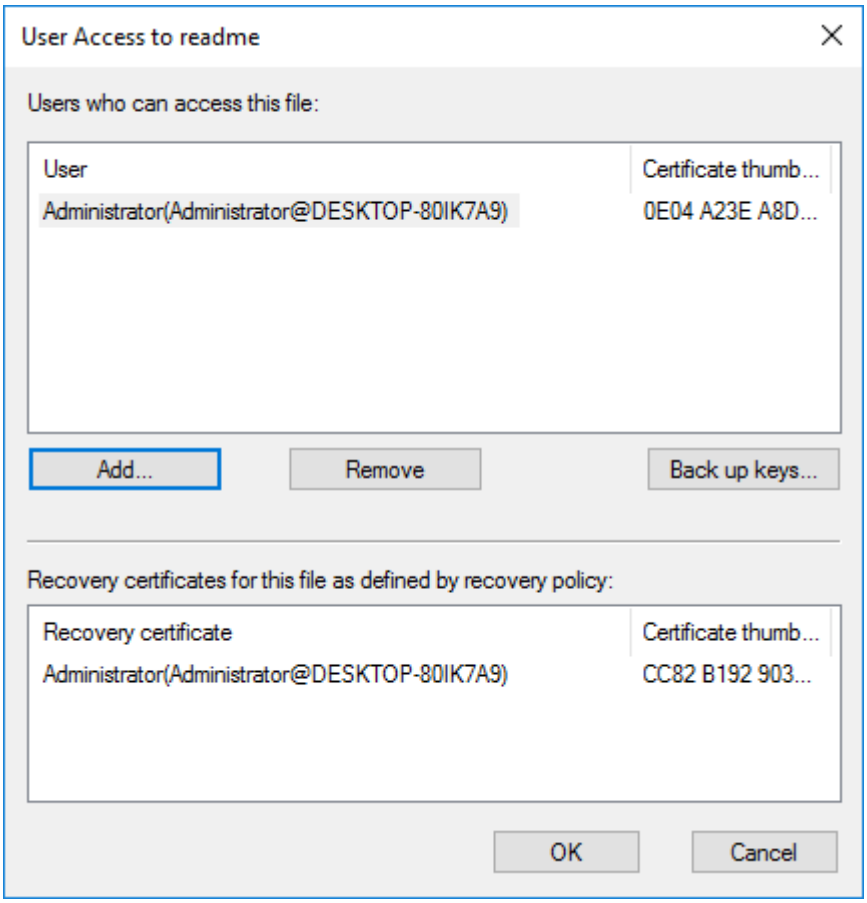

Рис. 18. Налаштування шифрованого файлу на спільний доступ декількох користувачів при діючій політиці агента відновлення.

У звіті до лабораторних робіт описати та пояснити отримані результати.

Контрольні запитання.

1. Чим відрізняються дозволи "Modify" та "Write"?
2. Чи можна заборонити доступ до об’єктів файлової системи адміністратору комп’ютера?
3. Хто може брати об’єкти у власність?
4. Об’єкт файлової системи може брати у власність група "Administratos". Яке призначення такої можливості?
5. Чи можна призначити індивідуальний розмір квот для кожного користувача, групи?
6. Призначення файлової системи EFS.
7. Хто може розшифрувати файл, зашифрований іншим користувачем?

СПИСОК ЛІТЕРАТУРИ

1. **Microsoft Corporation** Microsoft Windows XP Professional. Учебный курс MCSA/MCSE. – М.: Издательско-торговый дом "Русская Редакция", 2003. – 1008 стр.
2. **Ed Bott** Introduction Windows 10 for IT Professionals. – Published by Microsoft Press. A Division of Microsoft Corporation One Microsoft Way Redmond, Washington, 2015. – 115 p.
3. **Microsoft Corporation** Microsoft Windows 2000 Active Directory Services. Учебный курс MCSE. – М.: Издательско-торговый дом «Русская редакция», 2004. – 608 с.
4. **Руссинович М.**, **Соломон Д.** Внутреннее устройство Microsoft Windows: Windows Server 2003, Windows XP и Windows 2000. – М.: Издательско-торговый дом «Русская редакция»; СПб.: Питер, 2005. – 992 с.
5. **Вишневский А**. Windows Server 2003. Для профессионалов. – СПб.: Питер, 2004. – 767 с.
6. **К. Айвенс** Microsoft Windows Server 2003. Полное руководство. – М.: Издательство "СП ЭКОМ", 2004.– 896 с.

НАВЧАЛЬНЕ ВИДАННЯ

УПРАВЛІННЯ ДОСТУПОМ ДО ОБ'ЄКТІВ ФАЙЛОВОЇ СИСТЕМИ У WINDOWS 10

МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторної роботи
з дисципліни „Основи системного адміністрування”
для студентів напряму 6.050103 «Програмна інженерія» та
спеціальності 121 „Інженерія програмного забезпечення”

Укладачі Яковина Віталій Степанович
Муха Тарас Орестович
Шкраб Роман Романович

Редактор

Комп’ютерне верстання