МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

А. В. Мансуров

Базовые основы устройства и работы подсистемы безопасности многокомпонентных информационных систем (на примере ОС Windows)

Учебное пособие

Барнаул
Издательство
Алтайского государственного
университета
2020

УДК 004.45(075.8)
ББК 32.972.11я73
М 238

Рецензенты:

В. В. Поляков, доктор физико-математических наук, профессор

А. А. Лепендин, кандидат физико-математических наук, доцент

М 238…

Мансуров, Александр Валерьевич

Базовые основы устройства и работы подсистемы безопасности многокомпонентных информационных систем (на примере ОС Windows) [Текст] : учебное пособие / А.В. Мансуров. –Барнаул: Изд-во Алт. ун-та, 2020. – 68с.

ISBN 978-5-7904-2514-1

Учебное пособие содержит базовую информацию о структурной организации и функционировании подсистемы безопасности сложной многокомпонентной информационной системы на примере актуальных версий операционных систем Microsoft Windows. Цикл лабораторных работ служит для практического закрепления полученных теоретических знаний. Пособие предназначено для студентов направления подготовки «Информационная безопасность», оно может быть использовано преподавателями и студентами других технических направлений высших учебных заведений.

УДК 004.45(075.8)
ББК 32.972.11я73

ISBN 978-5-7904-2514-1

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В данном пособии рассматриваются базовые аспекты устройства и функционирования подсистем безопасности в информационных системах на примере реальных сложных многокомпонентных систем – семейства операционных систем (ОС) Microsoft Windows актуальных версий. Теоретический материал основывается на главах из книг М. Руссиновича и др. «Внутреннее устройство Windows» (7-е изд., 2018) и А. Мирошникова «Windows® Security Monitoring: Scenarios and Patterns» (2018), а также открытой документации, размещенной на веб-сайтах Microsoft, которая объясняет работу компонентов и модулей ОС Windows.

Пособие содержит раздел с теоретическими сведениями о базовых особенностях устройства и работы подсистемы безопасности и механизмов аутентификации ОС Windows актуальных версий. Раздел с лабораторными работами служит для практического закрепления полученных теоретических знаний.

Все действия, которые предлагаются к выполнению в ходе лабораторных работ, должны осуществляться только в предлагаемой виртуальной среде. Они ориентированы лишь на понимание работы механизмов безопасности систем и не являются основой для каких-либо аналогичных умышленных действий в отношении реально работающих систем под управлением ОС Windows.

Учебное пособие предназначено для студентов направления подготовки «Информационная безопасность», оно может быть использовано преподавателями и студентами других технических направлений высших учебных заведений.

I. ОБЩИЕ СВЕДЕНИЯ О ПОДСИСТЕМЕ БЕЗОПАСНОСТИ

Описание системных компонентов безопасности

Современные многокомпонентные информационные системы (ИС) имеют свои собственные механизмы для выполнения процедур идентификации, аутентификации и авторизации пользователя, а также для последующего обеспечения безопасного функционирования ИС при работе с информационными объектами на основе политики разграничения доступа к ним. Операционная система ОС Microsoft Windows (которая берется как пример для изучения) использует свои средства безопасности для защиты файлов, памяти и процессов, недопущения неблагоприятного воздействия пользовательских программ на программы других пользователей и компоненты самой операционной системы. Стандартно, в подсистему безопасности ОС Windows входят следующие компоненты [1]:

1) *Монитор безопасности* (Security Reference Monitor / **SRM**) – компонент ядра ОС, ответственный за выполнение основных функций безопасности в ОС при работе с объектами.
2) *Подсистема проверки подлинности локальной системы безопасности* (Local Security Authority SubSystem / **LSASS**) – компонент – процесс пользовательского режима lsass.exe, который отвечает за политику безопасности локальной системы, аутентификацию пользователей и их привилегии. Часть функционала вынесена в библиотеку lsasrv.dll.
3) *База данных политики локальной безопасности* (Local Security Authority DataBase / **LSADB**) – база данных, содержащая настройки политики безопасности ОС, хранится в реестре в разделе HKLM\SECURITY. База

данных содержит информацию относительно привилегий, разрешений и идентификаторов, работающих с данной ОС.

4) *Администратор учетных данных службы безопасности* (Security Accounts Manager / **SAM**) – компонент, отвечающий за работу с базой данных **SAMDB**, которая содержит имена пользователей, хэши паролей и определения групп в локальной ОС. Выполнен в виде библиотеки samsrv.dll, которая подгружается в процесс LSASS.
5) *Пакеты аутентификации* (Authentication Packages, Security Support Providers / **SSP**) – компоненты, реализуемые в виде DLL-библиотек для процесса LSASS, которые отвечают за процесс аутентификации пользователей в ОС.
6) *Интерактивный диспетчер входа в систему* (**Winlogon**) – компонент, отвечающий за процесс интерактивного входа в ОС, представлен процессом winlogon.exe.
7) *Пользовательский интерфейс входа в систему* (**Logon UI**) – компонент, образующий интерфейс для предоставления учетных данных пользователя ОС.
8) *Сетевые средства обеспечения взаимодействия служб безопасности* – компоненты, ответственные за доступ к данным политики безопасности ОС по локальной сети. В частности, сюда входят средства работы со службой каталогов *Active Directory* в случае включения ОС в *домен* – совокупность компьютеров и связанных с ними групп безопасности, которые управляются как единое целое.
9) *Средства локального контроля за исполняемыми объектами ОС и мониторинга (аудита) системы безопасности.*

На рис. 1.1 представлены структурная взаимосвязь и отношения компонентов подсистемы безопасности ОС Windows.

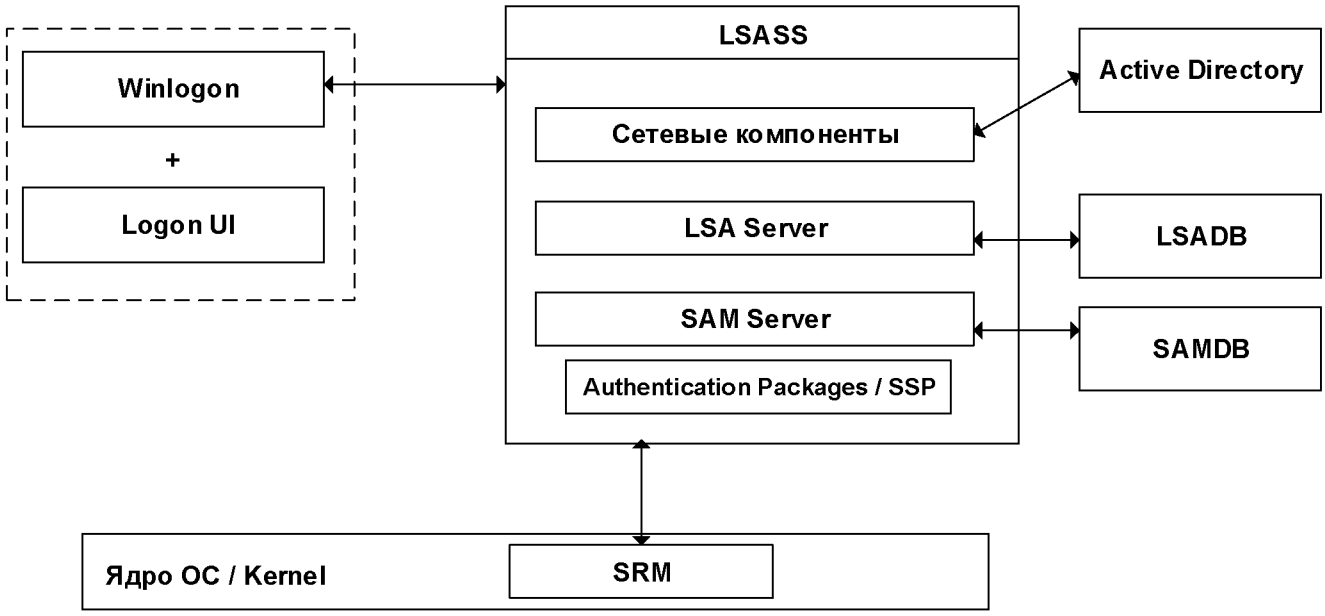

Рис. 1.1. Компоненты подсистемы безопасности ОС Windows.

Организация контроля доступа пользователей к объектам

В ОС Windows каждый пользователь обладает собственным *маркером доступа* (**Access Token**). Маркер доступа содержит уникальный идентификатор безопасности пользователя – Security Identifier / **SID**. SID-ы имеются у локальных пользователей ОС, у групп и служб (локальных и доменных), а также у компьютеров и доменов.

Общий вид идентификатора SID представляется следующим образом [3]:

$$S\text{-}R\text{-}X\text{-}Y^{1}\text{-}Y^{2}\text{-}Y^{n-1}\text{-}Y^{n}$$

где S – указывает на то, что это SID, R – номер версии (1), X – идентификатор Authority (в частности, 1 – World, 2 – Local, 3 – Creator, 5 – NT Authority), $-Y^{1}-Y^{2}-Y^{n-1}$ – идентификатор домена, Y^{n} – относительный идентификатор пользователя/группы.

Список наиболее популярных SID в ОС Windows приведен в [1,4].

Процесс LSASS при входе пользователя в систему создает уникальный SID и помещает его в маркер доступа пользователя. Все последующие процессы, запускаемые этим пользователем, будут наследовать маркер доступа пользователя.

Маркер доступа включает в себя SID пользователя, SID групп, в которые входит пользователь, список привилегий, идентификатор сессии пользователя и прочую информацию, необходимую для работы системы безопасности. Кроме этого, в маркере доступа также присутствует список «по умолчанию» избирательного доступа DACL (см. следующий абзац), который включается в дескриптор безопасности порождаемых объектов [1,5].

Объекты системы имеют свои *дескрипторы безопасности* – Security Descriptors, который содержит SID владельца, SID группы и списки управления

доступом (Access Control Lists - ACL) – DACL и SACL. Discretionary ACL (DACL) – *список дискреционного (избирательного) доступа* – определяет список пользователей, тип и возможности доступа для них. System ACL (SACL) – определяет пользователей и операции, которые должны учитываться в журнале аудита безопасности [1,6].

При обращении (получении доступа) процесса к объекту выполняется сравнение информации из маркера доступа пользователя, связанного с процессом, с записями в DACL (и SACL), после чего принимается решение. Общая диаграмма работы приведена на рис. 1.2 [7].

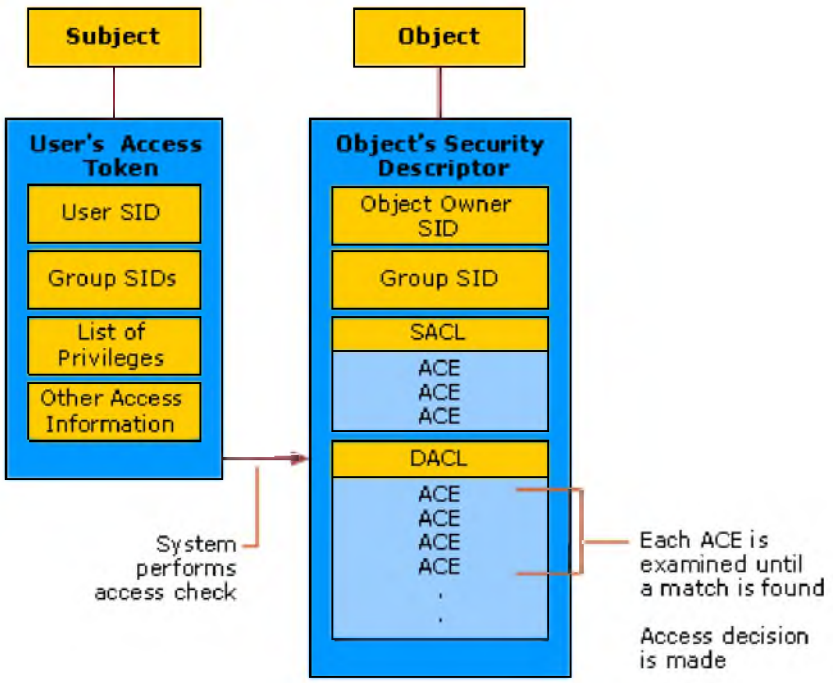

Рис. 1.2. Процесс сравнения информации маркера доступа процесса и дескриптора безопасности объекта при осуществлении доступа [7].

Дополнительным механизмом, расширяющим возможности дискреционного контроля доступа при помощи DACL, является механизм *обязательного контроля целостности* - Mandatory Integrity Control (MIC). MIC исполняется перед проверкой DACL и представляет собой вариант мандатного (полномочного) контроля, в соответствии с которым процессам и объектам может быть назначен один из

четырех уровней целостности - низкий (low), средний (medium), высокий (high) и системный (system) - в порядке их эскалации [1,9].

Четыре уровня MIC реализованы в виде особых SID вида S-1-16-0x00 (untrusted), S-1-16-0x1000 (low), S-1-16-0x2000 (medium), S-1-16-0x3000 (high) и S-1-16-0x4000 (system), соответственно. При это базовыми являются следующие принципы:

1) NO_WRITE_UP - процесс не может осуществлять доступ на запись к объекту, если его уровень целостности ниже уровня объекта;

2) NO_READ_UP - процесс не может осуществлять доступ на чтение к объекту, если его уровень целостности ниже уровня объекта;

3) NO_EXECUTE_UP - процесс не может осуществлять доступ на исполнение к объекту, если его уровень целостности ниже уровня объекта.

Привилегии

Привилегии в ОС Windows являются правом выполнять под той или иной учетной записью конкретную, связанную с системой операцию [1]. Таким образом в системе регулируются действия, которые не могут быть учтены путем стандартных операций контроля доступа к объекту при помощи маркеров доступа и дескрипторов безопасности (например, в случае, когда действия не касаются непосредственного взаимодействия с объектом в системе).

Список привилегий назначается LSASS при входе пользователя в систему и помещается в маркер доступа. Список привилегий достаточно широкий [1], среди них есть привилегии, позволяющие выполнять отладку программ (SeDebugPrivilege), создавать произвольный маркер доступа (SeCreateTokenPrivilege), получать права владения для файлов и объектов (SeTakeOwnershipPrivilege) и др. Полный спсиок привилегий можно посмотреть на странице справочной документации компании Microsoft: https://docs.microsoft.com/en-us/windows/win32/secauthz/privilege-constants [8].

Привилегии существуют в состоянии «включено» / enable или «выключено» / disable. Если привилегия не назначена для пользователя, то воспользоваться ею он не может.

Процесс аутентификации и входа пользователя в систему

Интерактивный вход пользователя в ОС Windows инициируется процессом Winlogon путем объявления специальной комбинации Secure Attention Sequence (SAS) - Ctrl+Alt+Del [1,2]. Winlogon далее вызывает компонент Logon UI для удобного ввода учетных данных (или **credentials**) пользователя в систему. Традиционно, учетные данные представлены в виде идентификатора (логина) и аутентифицирующей информации (пароля), но они могут быть представлены и получены самыми разными способами. Logon UI использует Credential Providers Interface для поддержки работы с разными *поставщиками учетных данных* (Credential Providers / CP) - данные смарт-карты (SmartCardCredentialProvider.dll), биометрические данные (BioCredProv.dll и FaceCredentialProvider.dll) и др. Credential Providers представлены в виде подгружаемых библиотек DLL и их список можно уточнить в реестре ОС Windows, открыв следующую ветку: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Authentication\Credential Providers`

Получив от Logon UI необходимые учетные данные (credentials), Winlogon создает для данного пользователя уникальный SID входа в систему и передает этот SID подсистеме LSASS (вызов функции LsaLogonUser) для выполнения дальнейших действий по проведению аутентификации. Аутентификация выполняется с использованием доступных LSASS имеющихся Authentication Packages или Service Support Providers (SSP). LSASS поддерживает несколько вариантов SSP, полный перечень которых содержится в реестре ОС по пути:

`HKLM\SYSTEM\CurrentControlSet\Control\Lsa`

Традиционно используются два стандартных пакета аутентификации:

1) Пакет **MSV1_0** – применяется на автономной Windows-системе, для отключенных от сети компьютеров и для версий ОС Windows младше Windows 2000. Выполнен в виде DLL библиотеки msv1_0.dll. Зашифрованные логин и пароль (хэш) сравниваются с данными, которые хранятся в локальной базе данных SAMDB. При успешном сравнении аутентификация выполняется.
2) Пакет **Kerberos** – применяется для аутентификации в домене Windows (сетевой вариант с контроллером домена). Это имплементация протокола Kerberos (RFC 4120) [10] для взаимодействия со службами Kerberos на контроллере домена. При успешном сравнении зашифрованного логина и пароля (хэш) с данными, хранящимися на контроллере домена, аутентификация выполняется.

После успешной аутентификации LSASS проверяет информацию о правах пользователя и создает маркер доступа и уведомляет об этом все имеющиеся SSP. На этом этапе все дополнительные SSP могут выполнить свои действия с аутентификационными данными пользователя и получить свои данные как результат работы. Все эти данные (маркер доступа пользователя и данные об аутентификации) передаются назад в Winlogon. Далее Winlogon запускает первый пользовательский процесс, который указан в реестре в пути:

```
HKLM\Software\Microsoft\Windows Nt\CurrentVersionWinlogon\Userinit
```

Далее первый процесс запускает остальные пользовательские процессы в ОС Windows (в частности – Explorer.exe). Общая схема выполнения аутентификации представлена на рис. 1.3.

Полученная информация об учетных данных в результате выполнения аутентификации и все вычисленные в процессе входя в систему и аутентификации

хэши хранятся в базе данных в локальном хранилище реестра HKEY_LOCAL_MACHINE\SAM. Для хэширования используются LM- и NT-хэширование (последнее более надежное, поскольку позволяет учитывать при вычислении хэша полную длину пароля и его регистрозависимость).

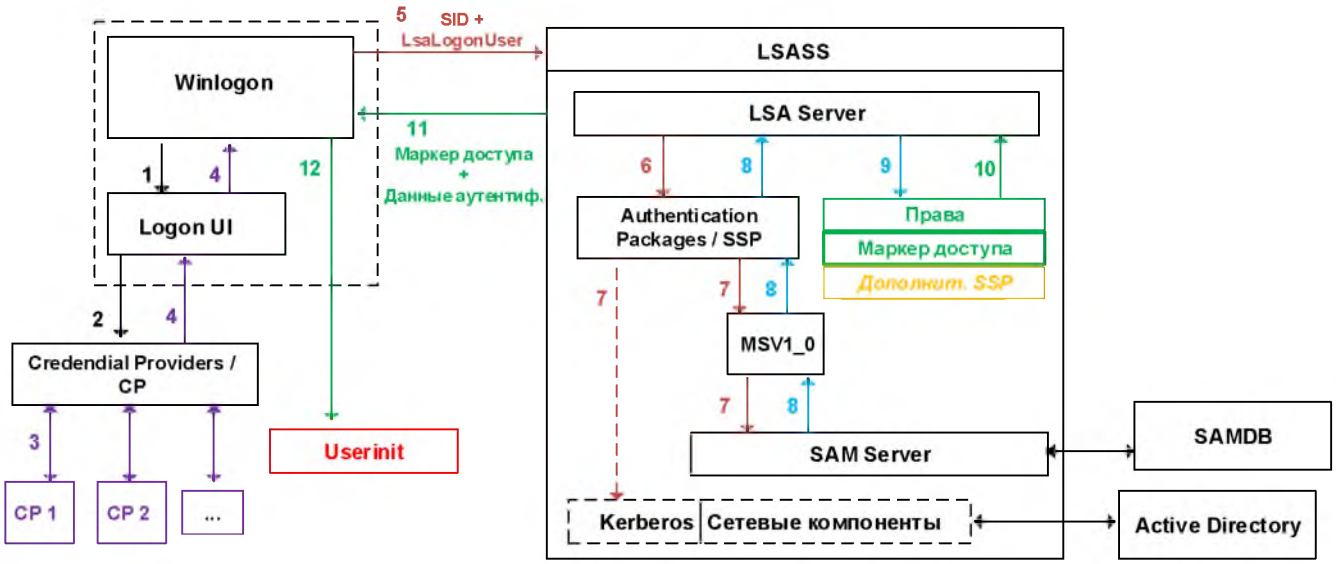

Рис. 1.3. Общая схема выполнения аутентификации.

LSASS хранит учетные данные в самых различных вариантах и форматах [11], которые могут включать в себя:

1) NT хэши
2) Билеты (Tickets) Kerberos – билеты TGT, TGS
3) Шифрованные и нешифрованные текстовые строки

Необходимость хранить всю эту информацию обусловлена концепцией Single Sign-On (SSO), которая предполагает однократное введение пользователем учетных данных и аутентифицирующей информации без необходимости аутентифицровать свои последующие действия [1,2]. Гибкость функционирования для удовлетворения потребностей различных возможных сервисов обеспечивается вариабельным набором SSP и стандартным API доступа к ним. В это же время, это вынуждает систему хранить всю необходимую информацию об учетных данных

или **credentials** (в том числе и все хэши и результаты работы всех нужных SSP) в памяти системы.

Для простоты понимания процесса входа, рассмотренная в этом разделе процедура и особенности ее работы приведена в упрощенном варианте. Более подробную и детальную информацию можно получить в источниках [1,2].

Типы входов в ОС Windows

ОС Windows поддерживает несколько разных типов входов [2]. Самые распространенные типы входов приведены в Таблице 1.

Таблица 1. Распространенные типы входов в ОС Windows.

Тип	Информация
2	Интерактивный. Применяется при стандартном входе пользователя на компьютер.
3	Сетевой. Пользователь или компьютер вошли на данный компьютер через сеть.
5	Служба. Service Control Manager произвел запуск службы.
7	Разблокирование. Эта рабочая станция разблокирована.
10	RemoteInteractive. Пользователь выполнил удаленный вход на этот компьютер, используя Terminal Services или Remote Desktop.
11	CachedInteractive. Пользователь вошел на этот компьютер с сетевыми учетными данными, которые хранились локально на компьютере. Контроллер домена не использовался для проверки учетных данных.

Как видно из Таблицы 1, тип входа «2» - интерактивный, является самым стандартным типом входа в систему, когда от пользователя требуется выполнение всех стандартных процедур идентификации и аутентификации. Данный тип также применяется при входе локальных пользователей в систему. Именно данный тип входа был рассмотрен в предыдущем разделе «Процесс аутентификации и входа пользователя в систему».

Тип входа «3» - сетевой – используется при обращении к сетевым сервисам ОС Windows, например, общим сетевым папкам и файлам и т.п. При сетевом входе необходимо иметь и предъявить ранее полученные учетные данные (или credentials), чтобы получить доступ. Как правило, механизм получения учетных данных (credentials), аналогичный интерактивному входу, не применяется.

Имперсонация

Имперсонация (олицетворение, impersonation) – обозначает способность для некоторого сервисного / серверного процесса образовать свой трид (thread), используя другой, отличный от своего «родного» контекст безопасности, ассоциируя порожденный трид с другим маркером доступа. В этом случае, например, при обращении клиента к серверному процессу, серверный процесс порождает отдельный новый трид и связывает (ассоциирует) его с полученным маркером доступа клиента. При этом клиентский маркер доступа является имперсонированным. Образованный трид, выполняющий имперсонацию, несмотря на то, что содержит и свой нативый маркер доступа, и имперсонированный, работает в контексте безопасности, определяемым имперсонированным (клиенстким) маркером доступа – т.е. в соответствии с определенными для имперсонированного (клиентского) маркера доступа ограничениями.

В ОС Windows различают четыре уровня имперсонации [12]:

1) Anonymous (анонимный) – клиент является анонимным, и серверный процесс не может выполнить имперсонирование.
2) Identification (идентификация) – серверный процесс может получить данные о клиенте, но не может имперсонировать клиента.
3) Impersonation (имперсонация) – серверный процесс может выполнить имперсонацию.
4) Delegation (делегирование) – самый высокий уровень. Клиентский маркер доступа считается *делегированным*, и серверный процесс может сам запрашивать имперсонацию от имени клиента на сетевых сервисах.

Имперсонированный маркер доступа может использоваться серверным процессом только локально, выполнить с его помощью какую-либо сетевую

операцию (например, доступ к сетевому ресурсу) невозможно. *Делегированный* маркер доступа, в свою очередь, можно имперсонировать при доступе к сетевому сервису – т.е. выполнение сетевых операций возможно.

При выполнении сетевого входа (тип «3») маркер доступа клиента уже есть и он передается при доступе к сервису. Таким образом, серверным процессом создается имперсонированный маркер доступа клиента. Его достаточно для работы серверного процесса локально в контексте безопасности имперсонированного пользователя, но невозможно совершать операции, связанные с доступом по сети к другим сетевым ресурсам.

При выполнении интерактивного входа (тип «2») маркер доступа клиента является делегированным. Его можно использовать для имперсонации при доступе к другим сетевым ресурсам (не только локально).

Схематичный вариант использования имперсонированного и делегированного маркера доступа приведен на рис. 1.4.

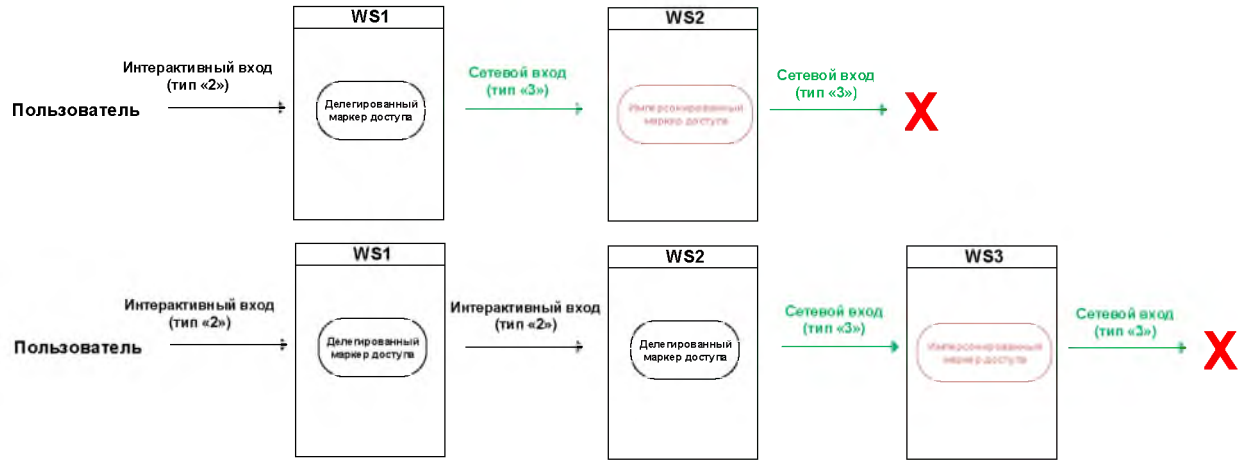

Рис. 1.4. Имперсонированный и делегированный маркер доступа в случае входов в систему разных типов.

II. ЛАБОРАТОРНЫЕ РАБОТЫ

Лабораторная работа №1.
Подготовка рабочего окружения в виртуальной среде

Цель работы: подготовить рабочее окружение в виртуальной среде для выполнения последующих лабораторных работ.

Выполнение лабораторных работ осуществляется в виртуальной среде. Для выполнения заданий необходимо развернуть требуемые рабочие станции под управлением ОС Windows и включить их в виртуальный сегмент локальной сети.

1. Скачайте и установите программную среду виртуализации VirtualBox - https://www.virtualbox.org/wiki/Downloads

2. Создайте в VirtualBox отдельное подключение VirtualBox Host-Only Ethernet Adapter. Это можно сделать в меню «*Файл*» - «*Менеджер сетей хоста*» (рис. 2.1.)

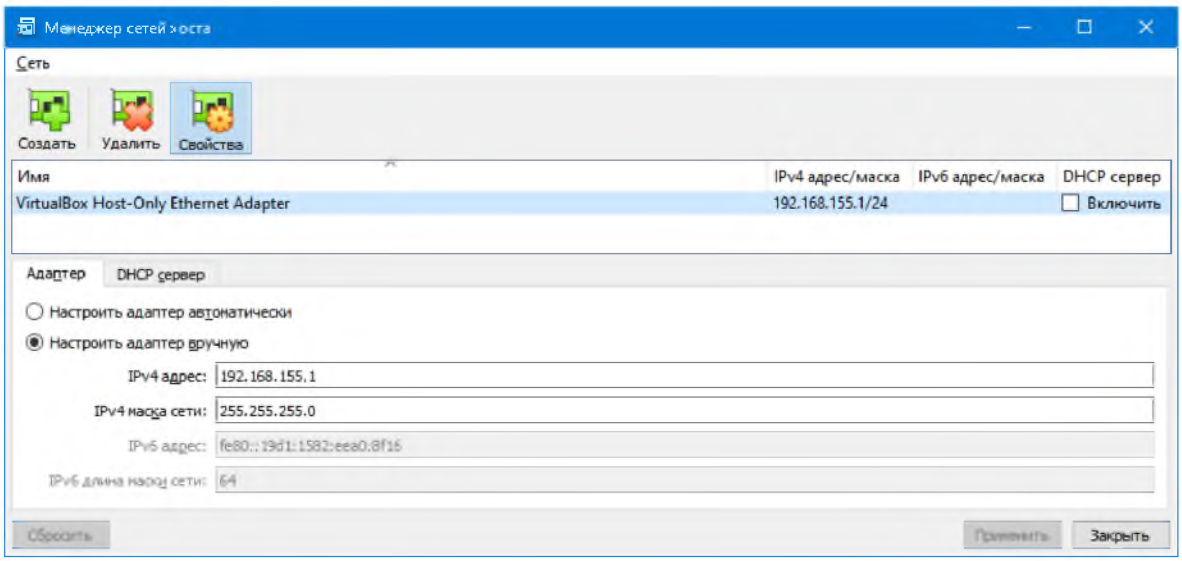

Рис. 2.1. Настройка VirtualBox Host-Only Ethernet Adapter.

Настроенное подключение будет использоваться в качестве виртуального сегмента локальной сети. Настройки необходимо выполнить аналогично приведенным на рис. 2.1.

Необходимо выбрать для VirtualBox Host-Only Ethernet Adapter опцию «*Настроить адаптер вручную*» и указать:

```
IPv4-адрес: 192.168.155.1,
Маска сети: 255.255.255.0,
DHCP-сервер – выключить.
```

Используемые IP-адреса могут быть произвольными. Чтобы следовать инструкции к лабораторным работам и примерам, рекомендуется оставить IP-адреса такими же, как в данном примере на рис. 2.1. Если в конфигурации компьютера присутствует несколько VirtualBox Host-Only Ethernet Adapter, то можно добавить еще один и выполнить его настройку.

3. Скачайте архивы виртуальных машин (ОС Windows 7 для VirtualBox) и архивы со вспомогательными утилитами. Актуальные ссылки можно получить у преподавателя.

Распакуйте содержимое архива «VM Windows 7 для VirtualBox». Выполните импортирование находящейся в архиве конфигурации и разверните две виртуальные машины с именами WIN1 и WIN2. Имена виртуальных машин можно задать при импорте конфигурации. Размер оперативной памяти виртуальных машин можно ограничить размером в 1000 – 1200 Мбайт. Можно адаптировать и другие системные параметры виртуальных машин под реальные характеристики основной платформы, на которой запускается и работает среда виртуализации.

4. Зайдите в раздел «*Сеть*» каждой виртуальной машины (WIN1 и WIN2), выберите вкладку «*Адаптер 1*» и поместите его в созданное и настроенное ранее

сетевое подключение VirtualBox Host-Only Ethernet Adapter (рис.2.2). Не забудьте изменить MAC-адрес сетевого адаптера виртуальной машины на уникальный – можно изменить на произвольные два последних байта MAC-адреса сетевого адаптера.

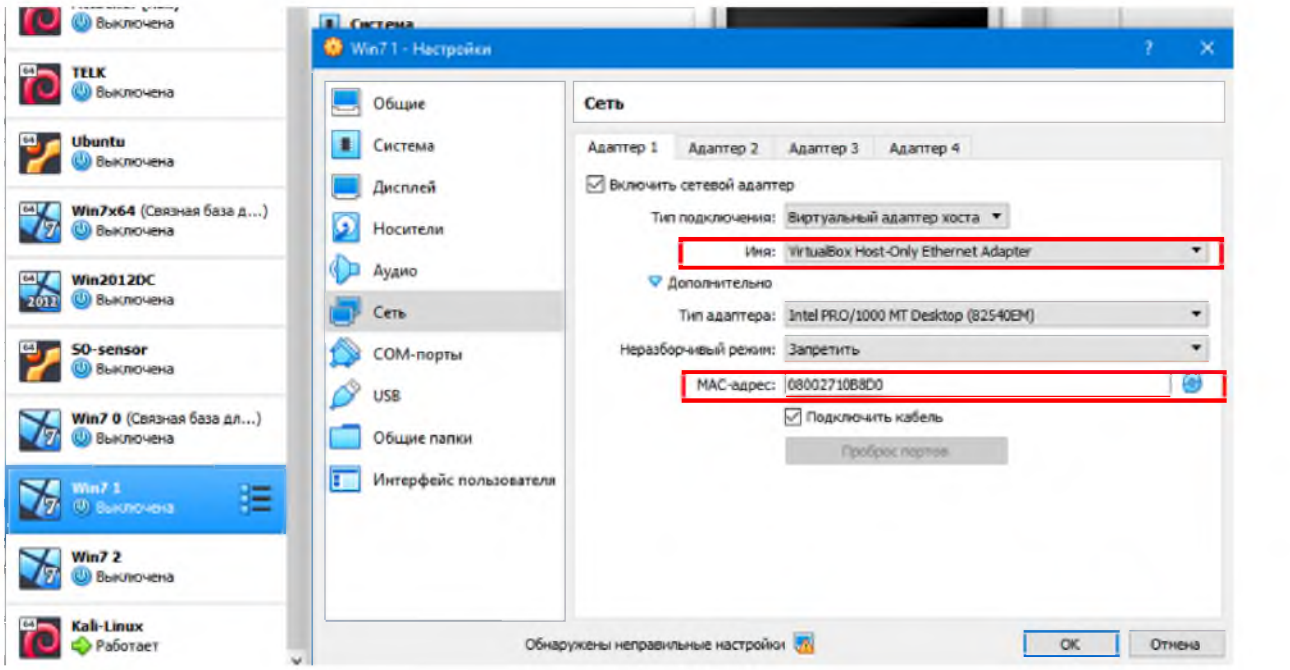

Рис. 2.2. Настройка параметров сетевого адаптера виртуальных машин.

5. Создайте отдельную папку на «основной» платформе и назначьте ее «общей папкой» в настройках каждой виртуальной машины (с правом на чтение/запись). Для этого нужно зайти в раздел «*Общие папки*» настроек виртуальной машины. Она будет использоваться для «транспортирования» внутрь виртуальных машин необходимых программ и файлов. Основная платформа будет иметь сетевое имя VBOXSRV, именно на этом сетевом имени компьютера будет находиться сконфигурированная общая папка при доступе к ней из работающих виртуальных машин.

6. После выполнения всех действий сформированная рабочая конфигурация должна быть следующей (рис.2.3):

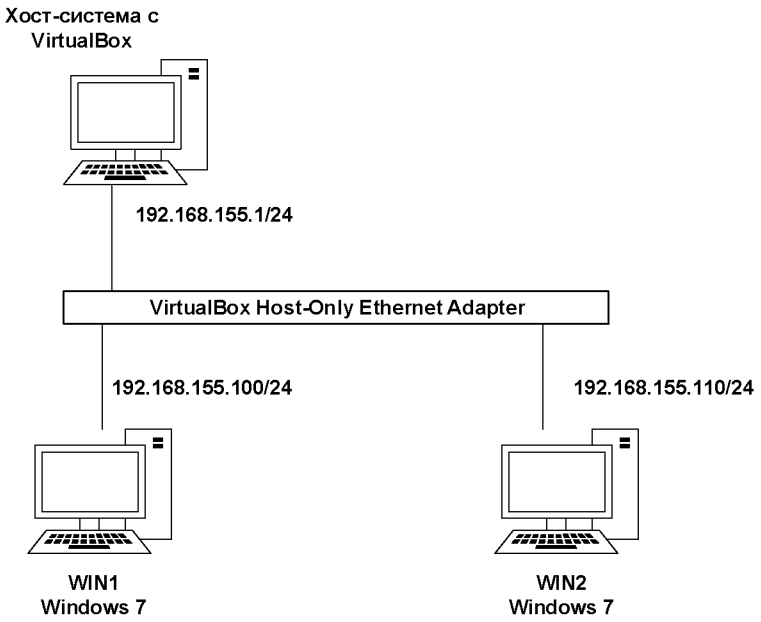

Рис. 2.3. Схема рабочей конфигурации, сетевых настроек и включения виртуальных машин с ОС Windows 7 в виртуальный сегмент локальной сети.

7. Загрузите виртуальные машины WIN1 и WIN2. После загрузки выполните вход в системы и убедитесь, что виртуальные машины работают без проблем и сбоев. По умолчанию, системы Windows 7 идут с уже предварительно созданным активным пользователем с правами администратора (в пароле – цифра “ноль”):

```
Имя пользователя: IEUser
Пароль: Passw0rd!
```

8. Поменяйте имя и рабочую группу загруженных систем на WIN1 и WIN2 соответственно (рис. 2.4). Рабочую группу оставьте WORKGROUP (по умолчанию). Перезагрузите машины, как предлагается. После перезагрузки снова войдите в системы и проверьте, что заданное имя сохранилось. Указанные имена WIN1 и WIN2 будут использоваться в дальнейшем для идентификации виртуальных машин в тексте лабораторных работ.

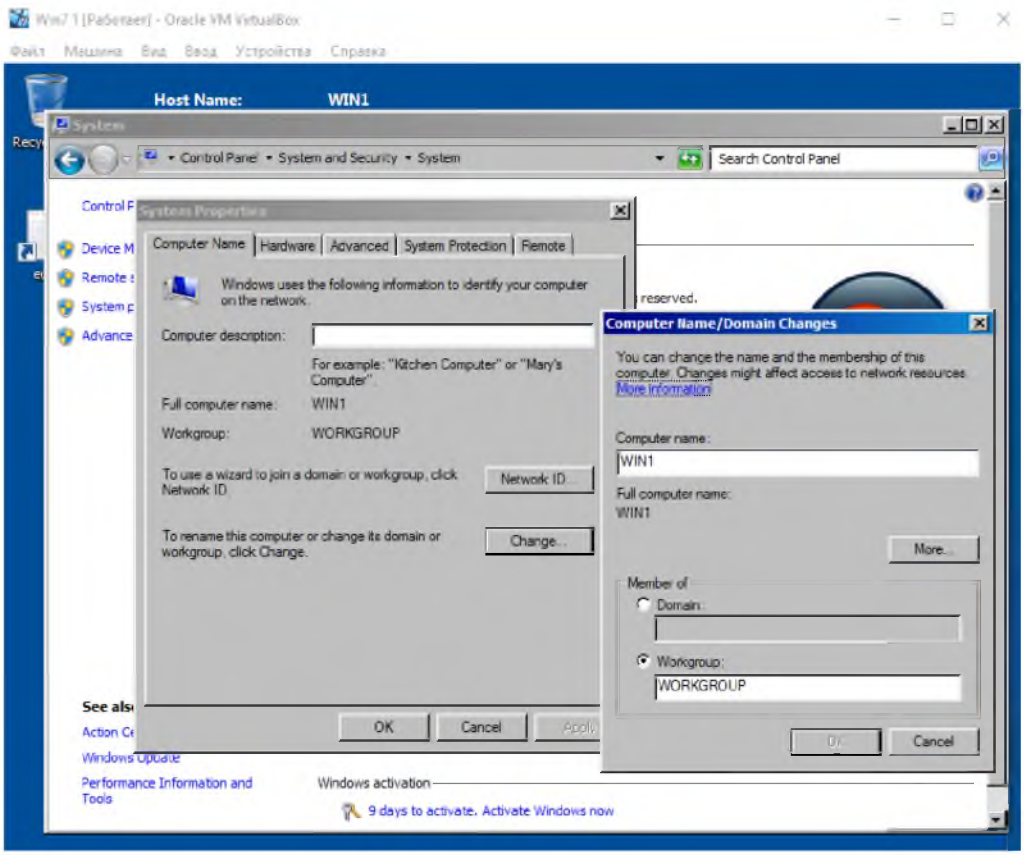

Рис. 2.4. Настройка сетевого имени рабочей машины с ОС Windows 7.

9. Настройте сетевое подключение **согласно конфигурации, приведенной на рис. 2.3.** Выберите «*Control Panel*» -> «*Network and Internet*» -> «*Network Connections*» доступное сетевое подключение, в выпадающем по нажатию правой кнопки мыши меню выберите пункт «*Properties*». В свойствах сетевого адаптера выберите протокол IPv4 и укажите требуемый для машины IP-адрес вручную (рис. 2.6). Шлюзом можно указать адрес 192.168.155.1 Определите созданную сеть как Work Network (рис. 2.5).

Рис. 2.5. Указание типа сетевого соединения как Work Network.

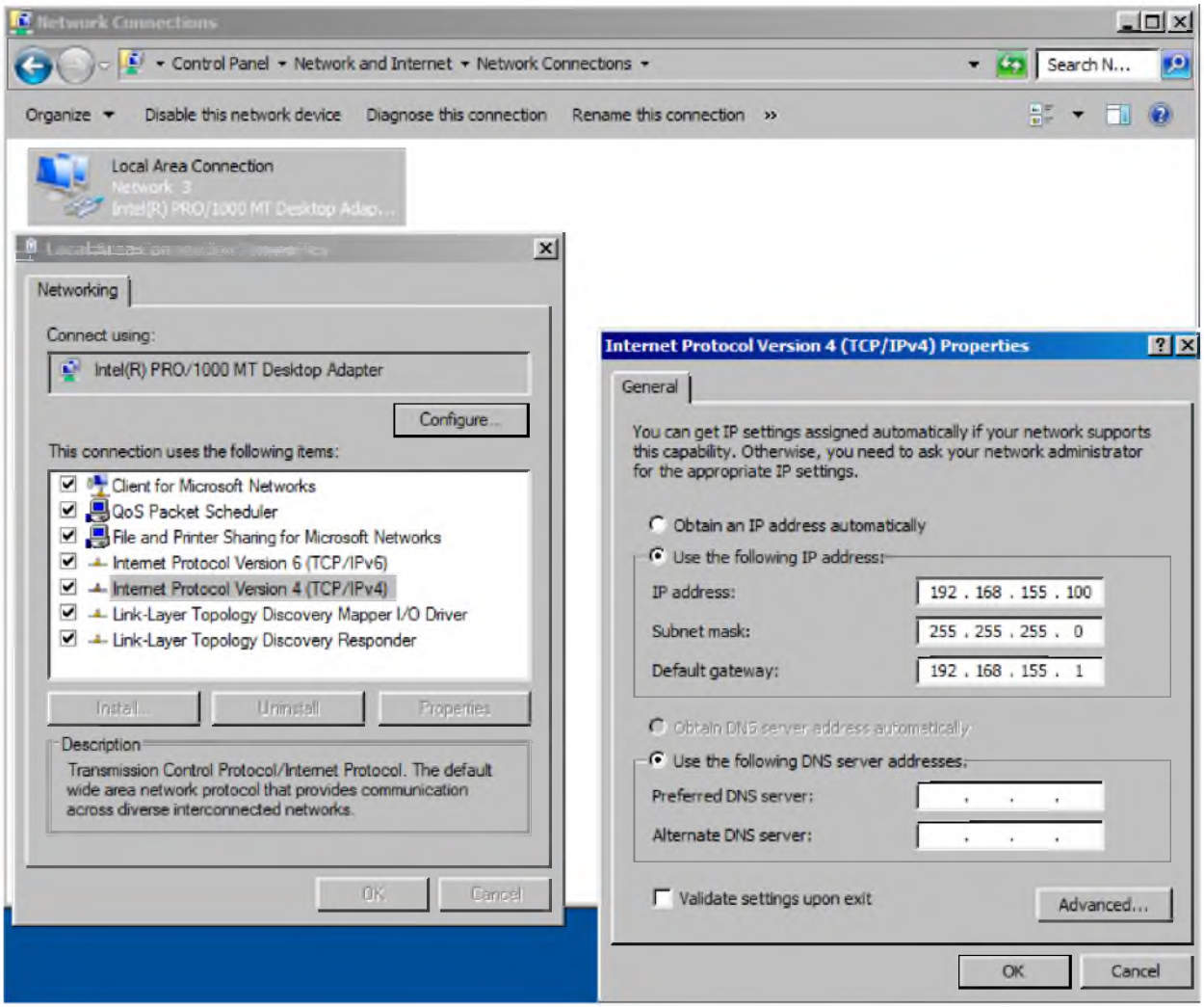

Рис. 2.6. Процесс настройки IP-адреса сетевого подключения для виртуальных машин WIN1 и WIN2.

В разделе «*Change advanced sharing settings*» разрешите общий доступ к файлам и устройствам для текущего сетевого профиля:

- `Turn on network discovery`
- `Turn on file and printer sharing`
- `Turn off Public folder sharing`
- `Turn off password protected sharing`
- `Allow Windows to manage homegroup connections`

10. На машине WIN1 создайте на диске C: папку “Share1” и правой кнопкой настройте общий доступ: «*Share with*» -> «*Specific People*» . Далее добавьте

пользователя Everyone (Все) с разрешениями Read/Write для этого пользователя. Нажмите «*Share*» (рис. 2.7).

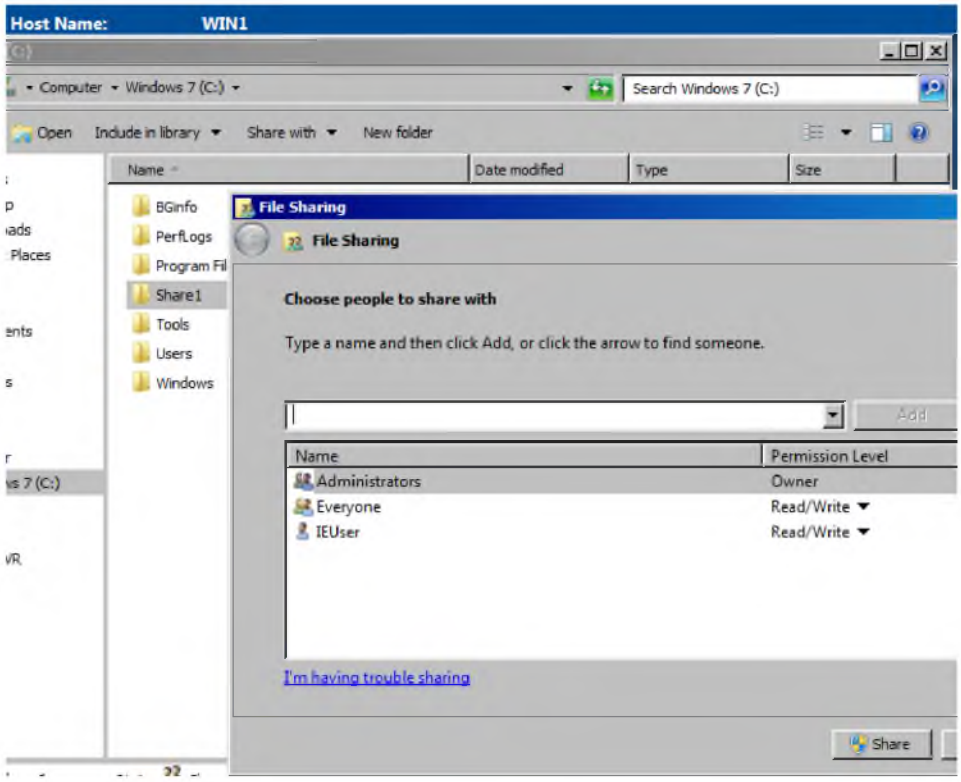

Рис. 2.7. Настройки общего доступа к сетевой папке с разрешением для «Всех».

На машине WIN2 создайте на диске C: папку “Share2” и правой кнопкой настройте общий доступ: «*Share with*» -> «*Specific People*». Далее добавьте пользователя Everyone (Все) с разрешениями Read/Write для этого пользователя. Нажмите «*Share*».

11. Перейдите в Проводник (файловый менеджер) и зайдите в раздел «*Network*». Если будет всплывающая надпись о необходимости включить общий доступ – согласитесь. Посмотрите, в единой сети должны отображаться две машины (WIN1 и WIN2), а также машина VBOXSVR с «общей папкой» (которую настроили ранее в свойствах виртуальных машин). На каждой из отображающихся в разделе машин «*Network*» должны открываться и быть доступны настроенные сетевые папки с именами “Share1” и “Share2”, соответственно (рис. 2.8). При

доступе к соседней машине вы должны видеть общую папку другой машины (для WIN2 это Share 2).

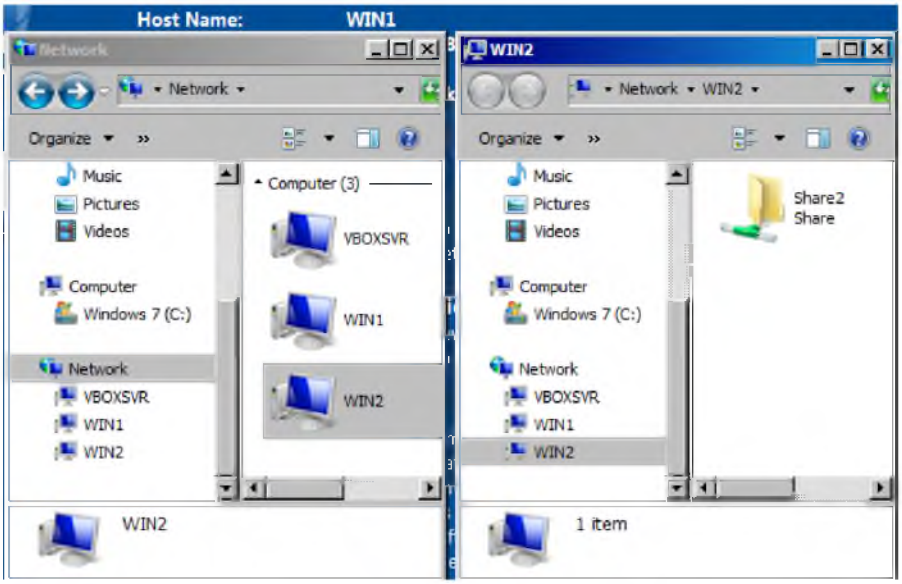

Рис. 2.8. Сетевые папки на машинах WIN1 и WIN2.

Попробуйте создать текстовый документ в открытой папке другой машины. Убедитесь, что обмен файлами через общие папки Share1 и Share2 работает в обе стороны!

12. На основной платформе скачайте и разместите в «общую папку» для WIN1 и WIN2, сконфигурированную на «основной» платформе VBOXSVR, следующие файлы (они понадобятся в дальнейшей работе):

+ Win1Tools.zip

+ Win2Tools.zip

ВНИМАНИЕ! Если антивирусное ПО будет сигнализировать о наличии вирусных или подозрительных объектов внутри данных архивов – отмените предупреждения или укажите на необходимость игнорировать подобные предупреждения. Ни в коем случае не позволяйте антивирусному ПО на

основной платформе заблокировать или модифицировать контент архивных файлов по причине «борьбы» с найденными вирусами внутри!

Вернитесь в виртуальные машины WIN1 и WIN2.

Распакуйте архив win1tools.zip (лежит в сетевой папке на машине VBOXSRV) в папку “C:\Tools” на машину WIN1.

Распакуйте архив win2tools.zip (лежит в сетевой папке на машине VBOXSRV) в папку “C:\Tools” на машину WIN2.

Архивы win1tools.zip и win2tools.zip содержат набор утилит для выполнения лабораторных работ. В состав утилит входят утилиты пакета “Windows Sysinternals” [13] и утилита mimikatz [14]. После распаковывания они будут находится в папке “C:\Tools”.

13. Рабочая среда для проведения лабораторных работ подготовлена. Не забывайте корректно выключать работающие виртуальные машины с ОС Windows 7!

Лабораторная работа №2. Процедуры аутентификации и разграничения доступа в подсистеме безопасности сложной многокомпонентной системы (на примере ОС Windows). Часть 1

Цель работы: изучение особенностей работы процедур идентификации, аутентификации и контроля доступа в сложных многокомпонентных системах (на примере ОС Windows).

В данной лабораторной работе понадобится только одна виртуальная машина WIN1. Запустите ее и выполните вход от имени существующего пользователя IEUser.

1. Изучите Раздел I «Общие сведения о подсистеме безопасности», а также ознакомьтесь с главой 7 источника [1] и с главами 4-6, 9-11 источника [2]. Теоретические знания будут необходимы для понимания дальнейших действий и ответов на вопросы лабораторной работы. Для поиска нужной информации и ответов на вопросы можно использовать любые доступные авторитетные источники в сети Интернет.

2. Ответьте на следующие вопросы:

A) Какой компонент отвечает за запуск первого процесса после успешного входа пользователя в систему ?

B) Что такое «учетные данные» / Windows Credentials в ОС Windows и как они формируются ?

C) Что такое Credential Provider ? Какие Credential Providers указаны в установленной версии ОС Windows машины WIN1 ?

D) Что такое Security Support Providers ?

E) Что такое «идентификатор безопасности» (Security Identifier, SID) в ОС Windows и какова его структура ? Назовите «популярные» SID в ОС Windows.

F) Поясните, что такое «Привилегия» / Privilege в ОС Windows ?

3. Запустите утилиту Process Explorer (утилита из пакета “Windows Sysinternals” [13]) из папки C:\Tools. Настройте ее так, чтобы можно было видеть перечень составляющих библиотек DLL для каждого активного процесса системы: «*View*» -> «*Lower Pane View*» -> «*DLLs*». В нижней части окна Process Explorer при выборе процесса будет отображаться список связанных и загруженных в данный момент динамических библиотек DLL (рис. 2.9).

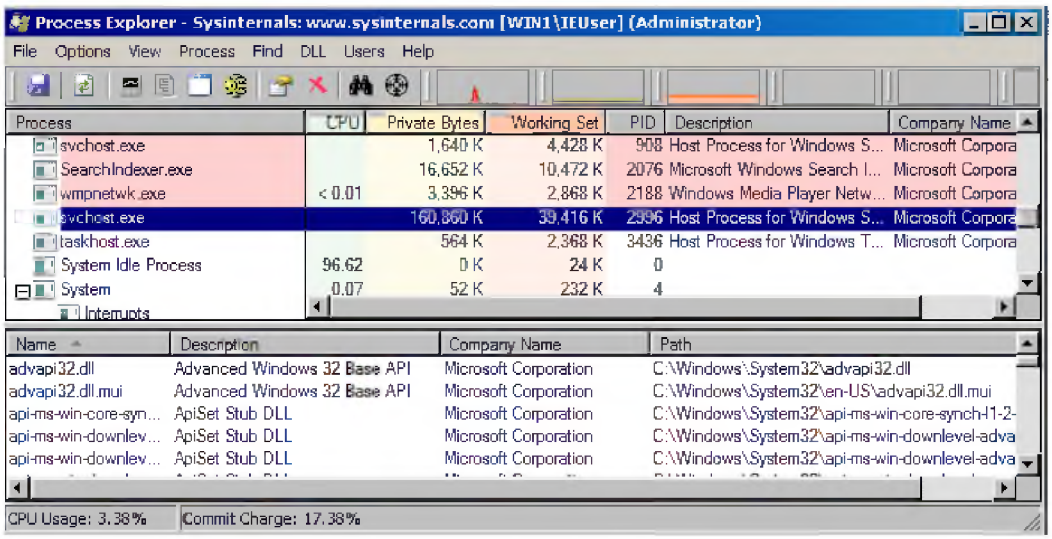

Рис. 2.9. Окно утилиты Process Explorer с панелью вывода DLL процесса.

Найдите процесс winlogon.exe. Проанализируйте список DLL процесса и укажите на DLL, которые используются для выполнения функций Credential Provider.

Найдите процесс lsass.exe. Проанализируйте список DLL процесса и укажите на DLL, которые используются для выполнения функций Security Support Provider.

4. Запустите командную строку (кнопка “*Start*” -> “*Run*”, ввести команду `cmd`) на WIN1. Найдите процесс, представляющий запущенную командную строку в Process Explorer. Посмотрите список доступных у процесса привилегий – нажмите

правую кнопку мыши на процессе -> «*Properties*» –> вкладка «*Security*». Привилегии отображаются в нижней части открытой вкладки «*Security*».

Найдите привилегию SeDebugPrivilege. Эта привилегия разрешает службе или учетной записи производить отладку системного процесса с помощью подключаемого отладчика программ, что позволяет процессам получать доступ к другим процессам, минуя проверку их дескриптора безопасности.

Укажите, в каком состоянии находится привилегия SeDebugPrivilege для командной строки, которая запущена в данный момент и является активным процессом в списке процессов ОС.

Перейдите в окно командной строки. Выполните в командной строке команду `whoami` без ключей, чтобы посмотреть текущее имя пользователя. Далее выполните команду `whoami /priv` и сравните распечатанный список привилегий со списком привилегий для командной строки во вкладке «*Security*» Process Explorer.

5. Используя Process Explorer, сравните состояние привилегии SeDebugPrivilege для запущенной командной строки и для процесса ‘Process Explorer’ – procexp.exe. Результат покажите в виде скриншота соотв. окна с одержимым вкладки «*Security*», где видно значение привилегии SeDebugPrivilege для командной строки и для процесса procexp.exe.

Согласно изложенной ранее теории, подсистема безопасности ОС Windows активно использует учетные данные пользователей / credentials. Для свободного использования credentials ОС Windows должна их хранить в памяти и держать «наготове». Для просмотра доступных credentials в памяти системы, воспользуемся утилитой mimikatz [14] из папки C:\Tools.

6. Запустите утилиту mimikatz в папке C:\Tools. Это можно сделать как в командной строке, так и непосредственно из графического интерфейса системы.

Утилита mimikatz консольная, потому все равно она откроется в окне командной строки.

Выполните в утилите mimikatz следующие команды:

```
privilege::debug
sekurlsa::logonpasswords
```

Вывод покажет хранящиеся в памяти ОС credentials / учетные данные. Внимательно изучите выведенную информацию на экране. Сделайте необходимые скриншоты.

Укажите следующую информацию для каждой из найденных записей учетных данных / credentials:

1) Имя пользователя, домен (если есть)
2) Пароль (если есть)
3) Время входа (Logon Time)
4) SID
5) NTLM hash
6) SHA1 hash
7) Тип сессии

7. На WIN1 изучите конфигурацию системы, собранные credentials и работающие процессы. Ответьте на следующие вопросы:

G) Почему в памяти системы хранятся credentials пользователя “sshd_server” ? Ведь активный пользователь на текущий момент – “IEUser”.

H) Какой пароль у пользователя “sshd_server” ?

I) Какой у пользователя “sshd_server” тип сессии ?

8. Работая с утилитой mimikatz, можно заметить, что для хранимых в памяти credentials / учетных данных пользователей в секции ‘wdigest’ фигурирует пароль пользователя, который отображается открытым текстом (рис. 2.10).

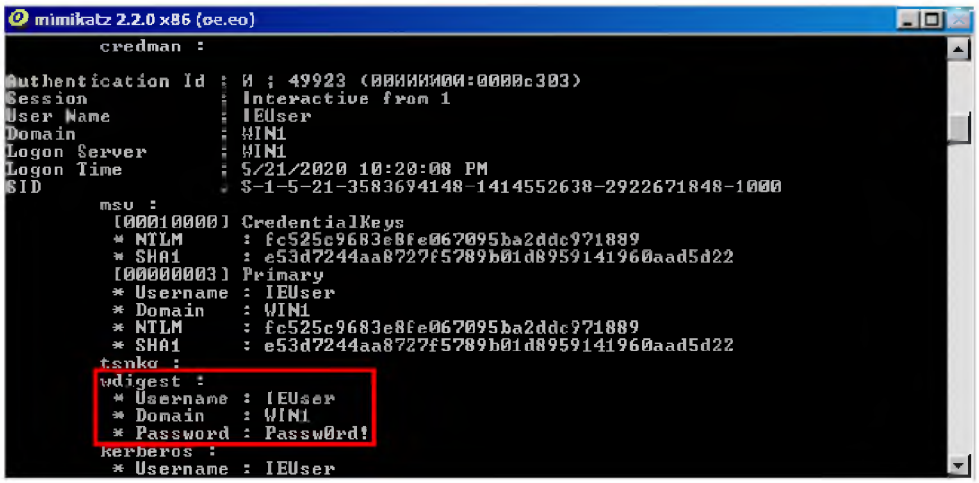

Рис. 2.10. Отображаемый открытым текстом пароль пользователя в секции ‘wdigest’.

Подсказка: Это происходит потому, что среди всех Security Support Providers провайдер WDigest поступает так исходя из соображений совместимости и требований Single Sign-On для пользователей.

При помощи утилиты Process Explorer просмотрите список DLL библиотек процесса lsass.exe и найдите там DLL провайдера WDigest. Приведите скрин-шот.

Набор Security Support Providers можно контролировать при помощи реестра ОС Windows:

`HKLM\SYSTEM\CurrentControlSet\Control\Lsa,`

`Переменная: Security Packages`

Откройте ее и посмотрите список провайдеров.

Удалите из перечня провайдеров строку ‘wdigest’. Сохраните реестр, перезагрузите систему WIN1 и снова запустите утилиту mimikatz для получения списка credentials в памяти. Сравните вывод с результатами рис. 2.10.

Постарайтесь при помощи Process Explorer теперь найти DLL провайдера WDigest в списке библиотек процесса lsass.exe. Сделайте скрин-шот.

Отредактируйте реестр ОС и верните значение переменной Security Packages в исходное состояние (снова добавьте в список wdigest) – так, как это обычно присутствует в типовой ОС Windows. Сохраните реестр и перезагрузите систему WIN1.

9. Оформите все результаты в виде отчета по лабораторной работе. Не забудьте включить все сделанные скрин-шоты. Все важные и результативные действия необходимо также сопровождать описаниями выполненных действий и скрин-шотами, которые должны быть включены в отчет. Не забудьте включить в отчет ответы на заданные вопросы – A) … I).

Лабораторная работа №3. Процедуры аутентификации и разграничения доступа в подсистеме безопасности сложной многокомпонентной системы (на примере ОС Windows). Часть 2

Цель работы: изучение особенностей работы процедур идентификации, аутентификации и контроля доступа в сложных многокомпонентных системах (на примере ОС Windows).

В данной лабораторной работе необходимы две виртуальные машины - WIN1 и WIN2. Запустите их и войдите под действующим пользователем IEUser на машины WIN1 и WIN2.

1. При помощи утилиты mimikatz посмотрите действующий список credentials в памяти машины WIN1. Запомните его. Сделайте скрин-шот(ы), если это необходимо.

2. Перейдите на машину WIN2. Откройте на WIN2 приложение «командная строка» (cmd) и перейдите в командной строке в каталог “C:\Tools”.

При помощи утилиты psexec.exe (утилита из пакета “Windows Systinternals” [13]) можно выполнить обращение по сети к машине WIN1 и запустить на WIN1 произвольное приложение (процесс). Можно выбрать любое приложение, которое имеется на машине WIN1 – «Калькулятор» (Calc.exe), «Блокнот» (Notepad.exe), … В данной лабораторной работе используется приложение «Блокнот» для его удаленного запуска на машине WIN1 с машины WIN2.

В открытом окне командной строки на WIN2 необходимо выполнить следующие команды:

```
cd C:\Tools
psexec.exe -i \\192.168.155.100 notepad.exe
```

Пример командной строки, запускающей утилиту psexec.exe:

3. Перейдите на машину WIN1. Запустите Process Explorer (находится в папке "C:\Tools") и убедитесь, что приложение «Блокнот» (Notepad.exe) действительно запустилось и находится в памяти ОС (работает).

3.1. Посмотрите свойства процесса "Notepad.exe" в Process Explorer (нажмите правую кнопку мыши на процессе -> «*Properties*» –> вкладка «*Security*»). Можно отметить, что родительским процессом для "Notepad.exe" является процесс "PSEXESVC.exe", а сам процесс "Notepad.exe" запущен от имени пользователя WIN1\IEUser (рис. 2.11) и имеет SID и свой Logon Session ID 0x690225 (в каждом конкретном случае будет свой собственный уникальный номер). Запомните содержимое поля "*Groups*" и "*Privilege*" (скопируйте или перепишите данные).

Ответьте на вопрос:

А) Почему процесс, который мы запустили удаленно с машины WIN2, выполняется на машине WIN1 от имени пользователя "IEUser" ? (Выскажите предположения, если нет ответа на данный вопрос).

3.2. Запустите утилиту mimikatz и выполните уже знакомые команды для просмотра имеющихся в памяти credentials / учетные данные:

```
privilege::debug
sekurlsa::logonpasswords
```

Зная Logon Session ID (в данном случае – 0x690225), постарайтесь найти в памяти ОС отвечающие данной сессии credentials (проверяйте последнюю группу цифр в поле “*Authentication ID*”, которое выводит mimikatz). Задокументируйте полученный результат, при необходимости сделайте скрин-шоты. Ответьте на вопрос:

В) Получилось ли найти credentials для данного Session ID ? Почему ? (Выскажите предположения, если нет ответа на данный вопрос).

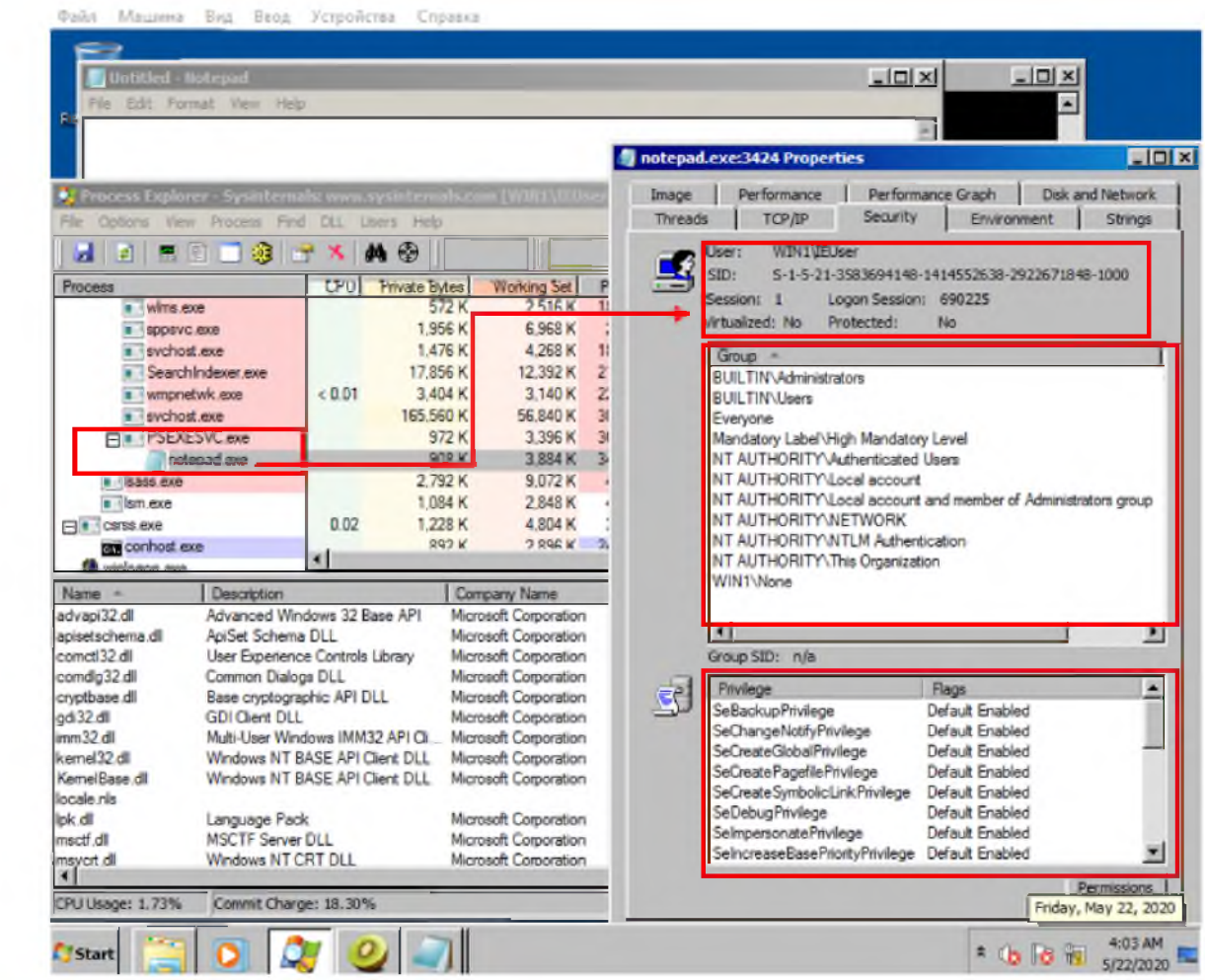

Рис. 2.11. Процесс “Notepad.exe” и его свойства.

3.3. Далее необходимо просмотреть записи в журнале безопасности ОС при помощи утилиты Event Viewer (eventvwr.msc), чтобы установить особенности появления процесса “Notepad.exe” в памяти машины WIN1.

Запустите утилиту Event Viewer (кнопка "*Start*" -> "*Run*", ввести команду `eventvwr.msc`), откройте в утилите разделы "*Windows Logs*" -> "*Security*".

ОС Windows имеет развитую систему аудита и ведет систематизированный журнал большого количества событий. В данном случае интерес представляют события, относящиеся к разделу безопасности – раздел "*Security*" или журнал безопасности ОС Windows.

Поскольку приложение «Блокнот» успешно запущено удаленно с WIN2 на WIN1 и имеет свой Session ID (ID сеанса), то это должно было случиться в случае успешного входа в систему и образования этого сеанса. Согласно [2], в журнале ОС (раздел "*Security*") при этом делается соответствующая запись с Event ID 4624 – "An account was successfully logged on". Необходимо воспользоваться инструментами утилиты Event Viewer (или аккуратно просмотреть все записи) и отобрать те записи, которые имеют Event ID 4624. Далее, откройте каждую найденную запись с Event ID 4624 и найдите ту, в которой упоминается Logon ID процесса "Notepad.exe" – Logon ID 0x690225 (в данном случае).

Внимательно изучите содержимое записи журнала безопасности и обратите внимание на ключевую информацию (рис. 2.12 и 2.13). Согласно данным записи журнала безопасности, вход, связанный с удаленным запуском приложения "Notepad.exe", произошел по сети (logon type 3), с машины WIN2 (192.168.155.110) при помощи штатного механизма NTLM, и он произошел успешно. Сделайте нужные скрин-шоты и включите их в отчет.

Как видно из полученной информации, удаленный запуск приложения «Блокнот» с WIN2 на WIN1 произошел с использованием сетевого входа (тип «3»). По случаю удачного стечения обстоятельств, пользователь IEUser существует одновременно и на машине WIN1 и на машине WIN2 и имеет один и тот же пароль. Его маркер безопасности с машины WIN2 успешно «подошел» для машины WIN1

и сетевой вход оказался возможным. При этом, никаких credentials / учетных данных, связанных с этим событием в памяти машины WIN1 не появилось.

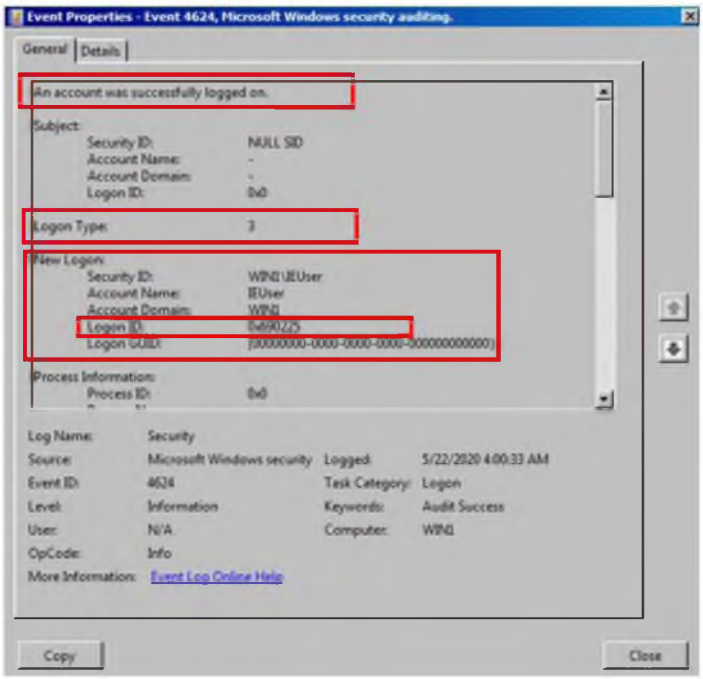

Рис. 2.12. Содержимое записи журнала безопасности для Logon ID 0x690225

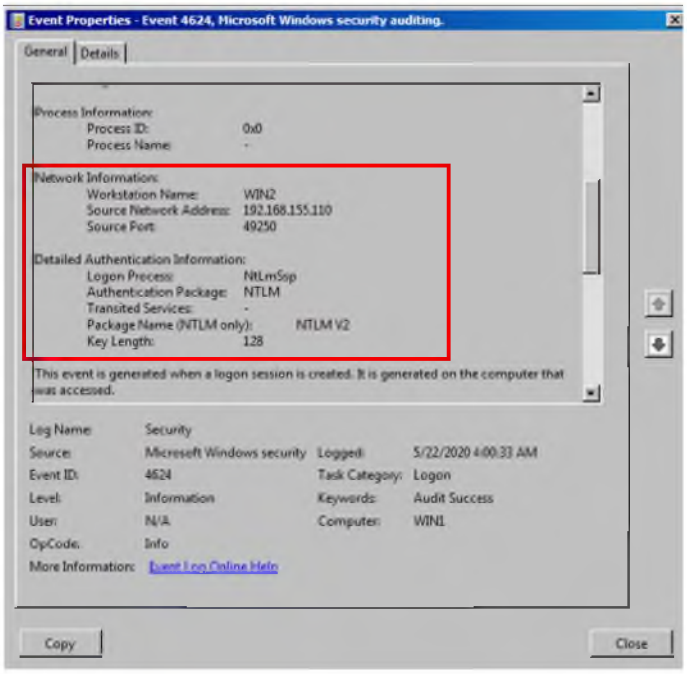

Рис. 2.13. Продолжение вывода записи журнала безопасности (с рис. 2.12).

Теперь можно вернуться к ответам на вопросы A) и B).

3.4.Закройте «Блокнот» (удалите процесс “Notepad.exe”) и убедитесь, что он исчез из памяти системы на машине WIN1.

4. Вернитесь на машину WIN2 и выполните удаленный запуск приложения «Блокнот» на машине WIN1 с использованием другого сценария.

В открытом окне командной строки на WIN2 необходимо выполнить следующие команды (первую команду можно не выполнять, если используется то же самое окно командной строки, что и раньше):

```
cd C:\Tools
psexec.exe -i \\192.168.155.100 -u IEUser -p Passw0rd! notepad.exe
```

Пример командной строки, запускающей утилиту psexec.exe:

5. Перейдите на машину WIN1. Запустите Process Explorer (находится в папке “C:\Tools”) и убедитесь, что приложение «Блокнот» (Notepad.exe) действительно запустилось и находится в памяти ОС (работает).

5.1. Посмотрите свойства процесса “Notepad.exe” в Process Explorer (нажмите правую кнопку мыши на процессе -> «*Properties*» –> вкладка «*Security*») (рис. 2.14). Можно отметить, что родительским процессом для “Notepad.exe”, как и в предыдущем случае, является процесс “PSEXESVC.exe”, а сам процесс “Notepad.exe” запущен (как и в предыдущем случае) от имени пользователя WIN1\IEUser и имеет SID и свой Logon Session ID 0x715b34.

Обратите внимание на то, что в отличие от предыдущего случая, у процесса “Notepad.exe” изменился набор привилегий и их состояние, а также поменялся набор групп.

Сделайте сравнительную таблицу для содержимого полей “Groups” и “Privileges” с аналогичными значениями полей, запомненных (или скопированных) в предыдущем случае удаленного запуска приложения на WIN1. Включите таблицу в отчет.

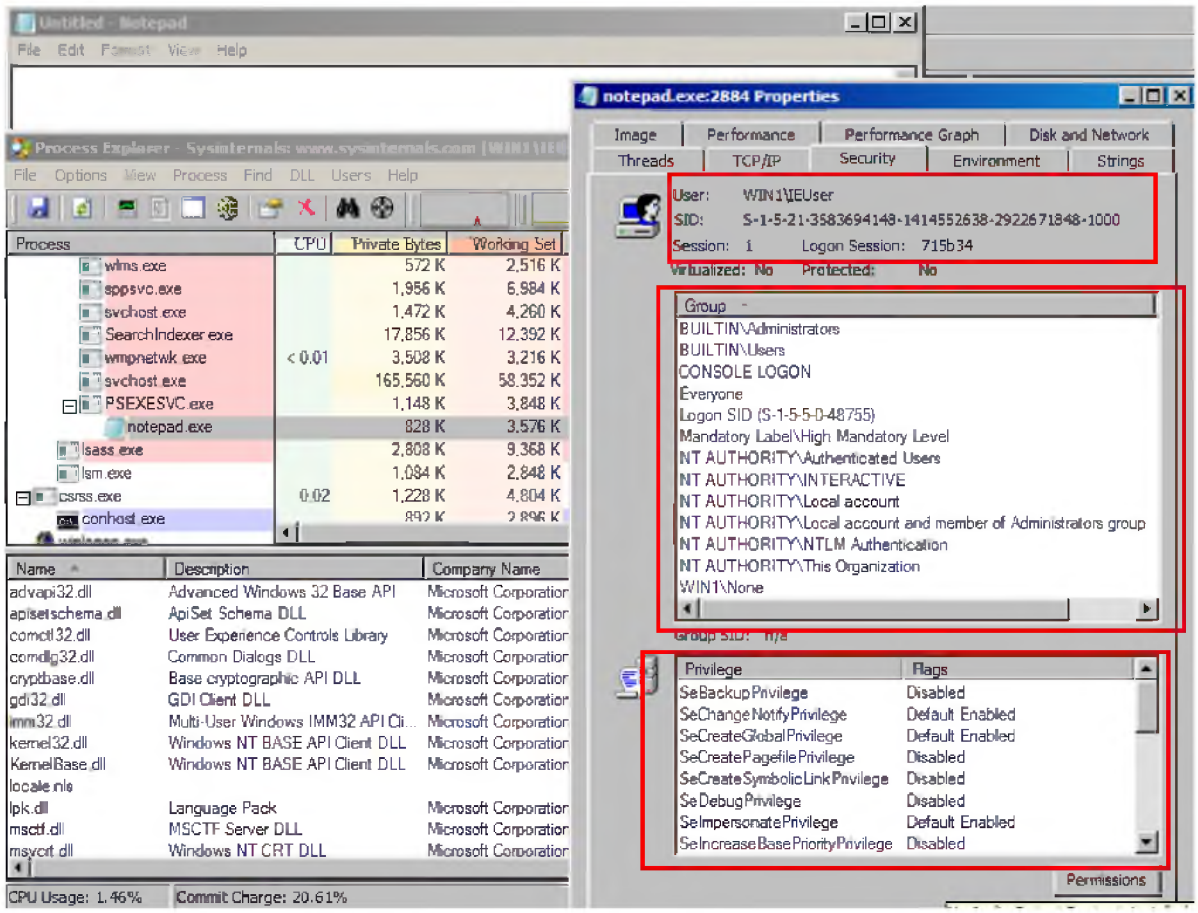

Рис. 2.14. Процесс “Notepad.exe” и его свойства.

5.2. Запустите утилиту mimikatz и выполните уже знакомые команды для просмотра имеющихся в памяти credentials / учетные данные.

```
privilege::debug
sekurlsa::logonpasswords
```

Зная Logon Session ID (в данном случае – 0x715b34), постарайтесь найти в памяти ОС отвечающие данной сессии credentials (проверяйте последнюю группу

цифр в поле “*Authentication ID*”, которое выводит mimikatz). Задокументируйте полученный результат, при необходимости сделайте скрин-шоты (аналогично рис. 2.15). Ответьте на вопрос:

C) Получилось ли найти credentials для данного Session ID ? Почему ? (Выскажите предположения, если нет ответа на данный вопрос).

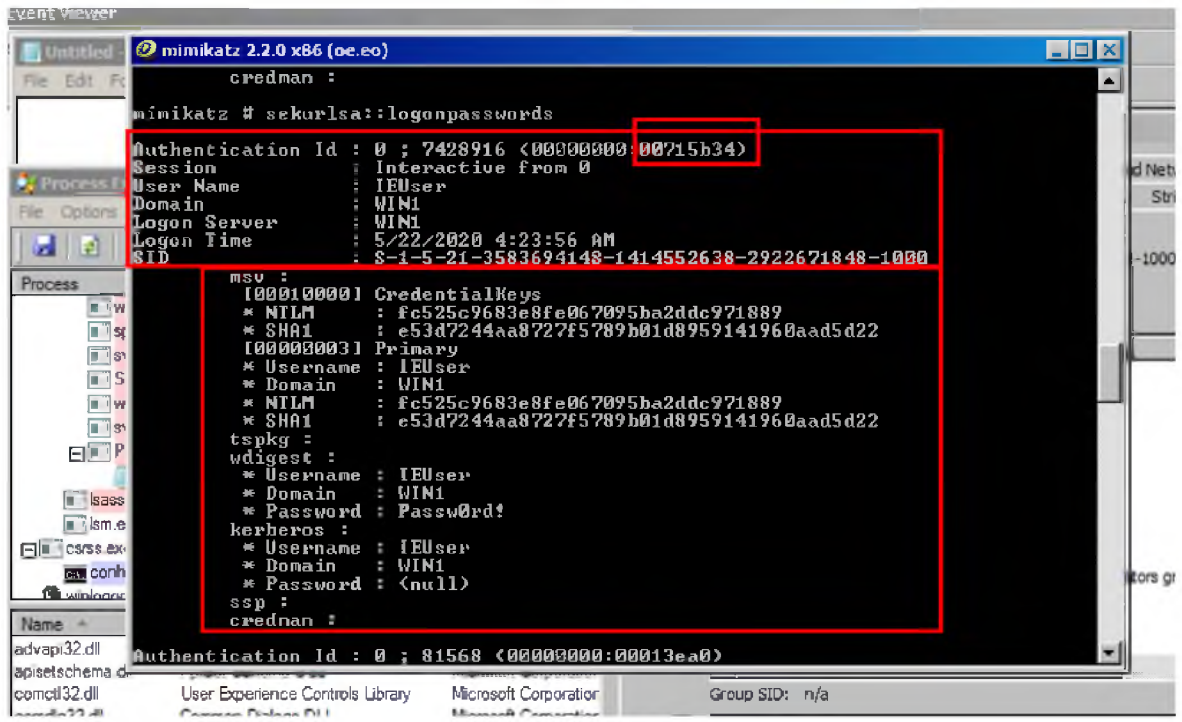

Рис. 2.15. Вывод mimikatz учетных данных для Session ID 0x715b34.

5.3. Далее необходимо просмотреть записи в журнале безопасности ОС при помощи утилиты Event Viewer (eventvwr.msc), чтобы установить особенности появления процесса “Notepad.exe” в памяти машины WIN1.

Аналогично предыдущему случаю, найдите в журнале безопасности ОС записи с Event ID 4624, касающиеся рассматриваемой сессии с Logon Session ID 0x715b34 (рис. 2.16 и 2.17).

Согласно данным записи журнала безопасности, вход, связанный с удаленным запуском приложения “Notepad.exe”, произошел интерактивно (logon type 2). Также

есть данные и о процессе, который связан со входом на WIN1 – “ PSEXESVC.exe” (не этот ли процесс является родительским для процесса “Notepad.exe” ?). Сделайте нужные скрин-шоты и включите их в отчет.

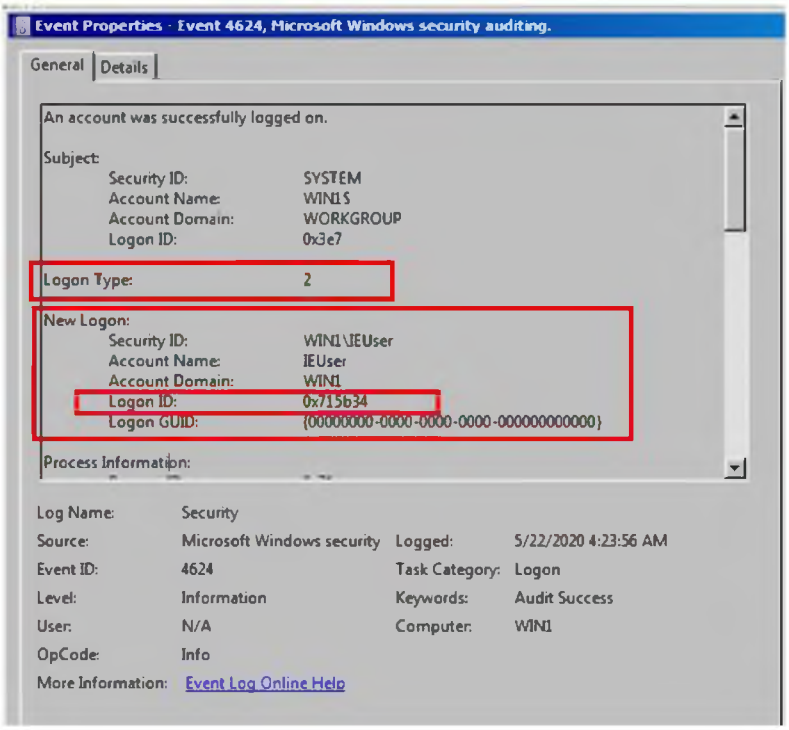

Рис. 2.16. Содержимое записи журнала безопасности для Logon ID 0x715b34.

Процесс интерактивного входа сопровождает также запись с Event ID 4648, в которой тоже есть информация и связь с процессом “PSEXESVC.exe”. Согласно расшифровке особенности Event ID 4648, «This event is generated when a process attempts to log on an account by explicitly specifying that account’s credentials. This most commonly occurs in batch-type configurations such as scheduled tasks, or when using the RUNAS command.»

Найдите в журнале безопасности нужную запись с Event ID 4648. Сделайте скрин-шоты и включите их в отчет.

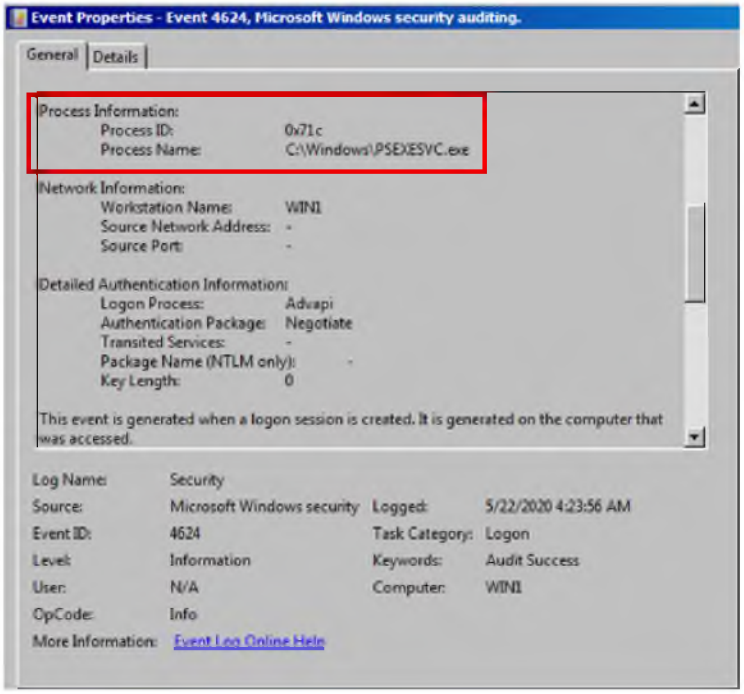

Рис. 2.17. Продолжение вывода записи журнала безопасности (с рис. 2.16).

Как видно из полученной информации, удаленный запуск приложения «Блокнот» с WIN2 на WIN1 произошел с использованием интерактивного входа (тип «2»), для которого были указаны верные имя пользователя и пароль. Выполнив интерактивный вход, как и в случае непосредственного входа на машину WIN1, система сгенерировала необходимые credentials / учетные данные, которые можно наблюдать в памяти системы.

Теперь можно вернуться к ответу на вопрос C).

5.4. Закройте «Блокнот» (удалите процесс “Notepad.exe”) и убедитесь, что он исчез из памяти системы на машине WIN1.

6. В ходе исследования особенностей удаленного запуска приложения на машине WIN1 (как в случае сетевого (тип «3»), так и интерактивного (тип «2») входа) можно обратить внимание на присутствующий для запускаемого приложения родительский процесс “PSEXESVC.exe”.

Найдите на WIN1 файл “PSEXESVC.exe”. Можно искать вручную, можно воспользоваться поиском ОС Windows (кнопка “*Start*” и в окне поиска ввести имя файла). Если получилось найти данный файл, то укажите в отчете, в каком каталоге / папке он располагается, его размер и дату/время создания и модификации. Если не получилось, укажите в отчете его отсутствие в системе на машине WIN1.

6.1. Перейдите на машину WIN2 и снова выполните из открытой командной строки удаленный запуск на WIN1 приложения «Блокнот» с использованием сетевого входа в систему WIN1:

```
psexec.exe -i \\192.168.155.100 notepad.exe
```

Перейдите на WIN1 и убедитесь, что приложение запустилось. Воспользуйтесь на WIN1 утилитой Process Explorer (запустите ее) и посмотрите в Process Explorer свойства процесса “PSEXESVC.exe” – вкладки “*Image*” и “*Security*” (рис. 2.18). Выполните необходимые скрин-шоты и включите их в отчет.

Обратите внимание на то, что процесс “PSEXESVC.exe” запущен системным процессом “services.exe” и выполняется от имени пользователя NT AUTHORITY\SYSTEM. Это специальная встроенная учетная запись, обладающая очень высокими правами доступа. Она имеет обширные права и выступает в качестве компьютера сети. Фактически это означает, что процесс “PSEXESVC.exe” запущен от имени самой системы – с самыми высокими правами в рамках данной локальной системы WIN1. Все привилегии находятся в состоянии «включено» для данного процесса.

Также, данный процесс (согласно вкладке “*Image*”) имеет свое физическое присутствие на диске в виде файла PSEXESVC.exe, который расположен в каталоге “C:\Windows” (полный путь “C:\Windows\PSEXESVC.exe”). Найдите на WIN1 файл “PSEXESVC.exe”. Можно искать вручную, можно воспользоваться поиском ОС Windows (как уже предлагалось ранее). Если получилось найти данный файл, то

укажите в отчете, в каком каталоге / папке он располагается, его размер и дату/время создания и модификации. Если не получилось, укажите в отчете его отсутствие в системе на машине WIN1.

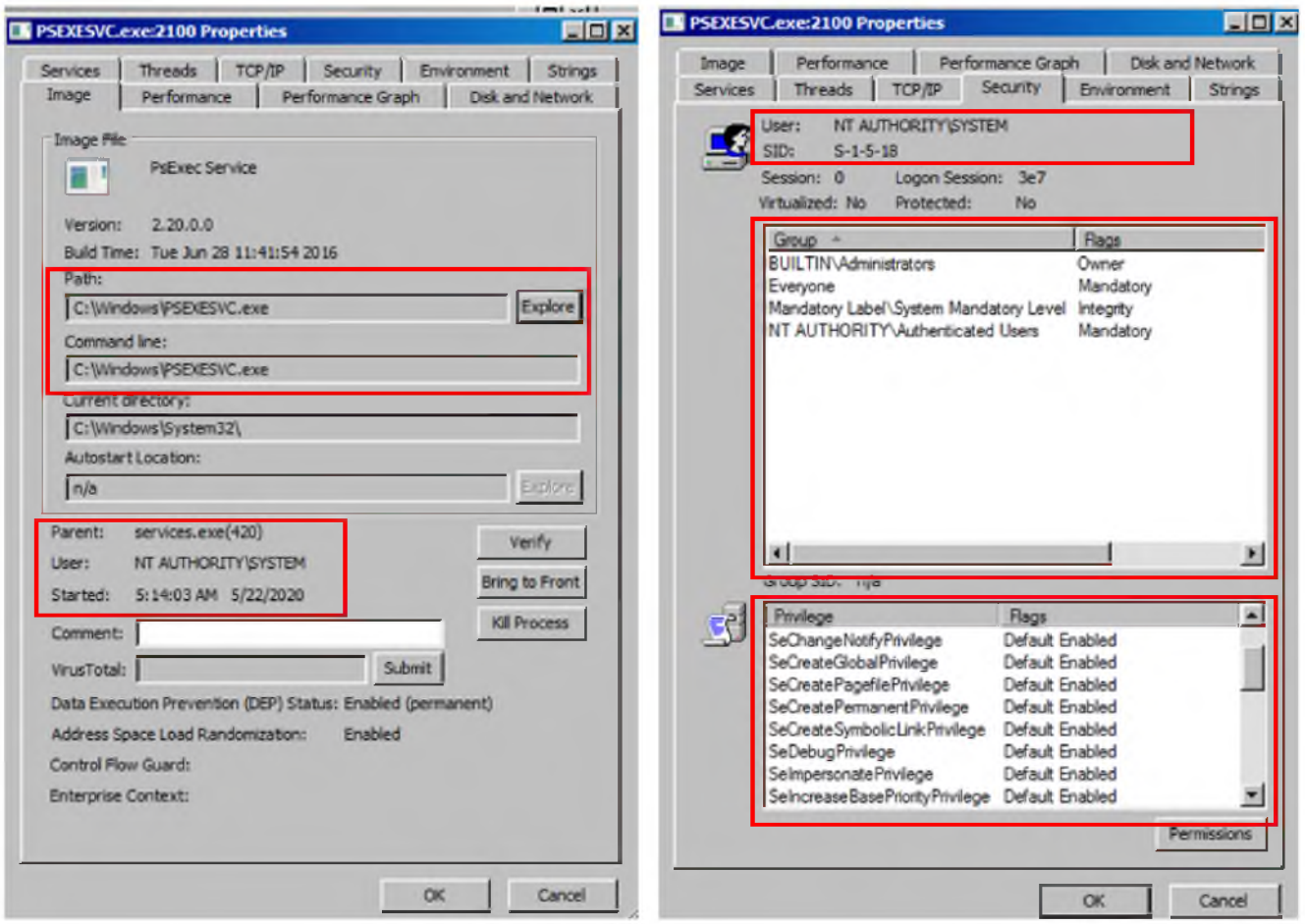

Рис. 2.18. Свойства процесса “PSEXESVC.exe”.

6.2. Воспользуйтесь ресурсами сети Интернет и найдите информацию о процессе “PSEXESVC.exe”. Включите найденную информацию в отчет.

Подсказка: Согласно данным источников в сети Интернет, PSEXESVC.exe является службой Windows и входит в состав утилиты PsExec.exe. Утилита PsExec при запуске удаленного приложения сначала распаковывает ее в скрытую административную сетевую папку удалённого компьютера Admin$ (путь “C:\Windows”), а затем сообщает удаленной системе о необходимости запустить ее как службу.

6.3. Закройте «Блокнот» (удалите процесс “Notepad.exe”) и убедитесь, что он исчез из памяти системы на машине WIN1. Перейдите на машину WIN2.

6.4. Поскольку данный сервис работает с максимальными полномочиями, он может наделить такими же максимальными полномочиями и любое приложение, которое будет запущено далее и иметь процесс “PSEXESVC.exe” как родительский. Например, чтобы удаленно вызвать появление на экране WIN1 командной строки, работающей от имени NT AUTHORITY\SYSTEM, необходимо на WIN2 в открытом окне командной строки ввести следующую команду:

```
psexec.exe -s -i \\192.168.155.100 cmd
```

Перейдите на машину WIN1. Найдите на WIN1 при помощи утилиты Process Explorer запущенную процесс командной строки и посмотрите его свойства (вкладки “*Image*” и “*Security*”). Сделайте необходимые скрин-шоты и включите их в отчет.

Перейдите в окно запущенной на WIN1 командной строки и выполните следующие команды:

```
hostname
whoami
```

Первая команда покажет имя компьютера, на котором выполняется данная командная строка, а вторая команда выведет имя пользователя, от которого выполняется командная строка. Сделайте скрин-шот и включите его в отчет.

Закройте окно выполняющейся командной строки или наберите команду:

```
exit
```

Убедитесь, что процесс запущенной удаленно командной строки пропал из памяти системы на WIN1. Вернитесь на машину WIN2.

При помощи утилиты psexec.exe можно получить командную строку машины WIN1, находясь непосредственно на машине WIN2. Перейдите в открытое окно командной строки (на WIN2) и выполните следующие команды:

```
cls
```

Данная команда очистит экран командной строки. Далее необходимо выполнить команды, которые покажут нам наше местонахождение и имя пользователя, который запустил данную командную строку:

```
hostname
whoami
```

Теперь выполните команду удаленного запуска командной строки на машине WIN1, но с выводом результатов работы на машину WIN2:

```
psexec.exe -s \\192.168.155.100 cmd
```

Посмотрите, что в окне командной строки на WIN2 изменилась строка приглашения (промпт). Выполните команды и убедитесь, что теперь работает командная строка машины WIN1 от имени самой системы:

```
hostname
whoami
```

Пример того, что должно быть в окне командной строки в этом случае приведен на рис. 2.19.

Перейдите на машину WIN1. Найдите на WIN1 при помощи утилиты Process Explorer запущенную процесс командной строки и посмотрите его свойства (вкладки “*Image*” и “*Security*”). Сделайте необходимые скрин-шоты и включите их в отчет.

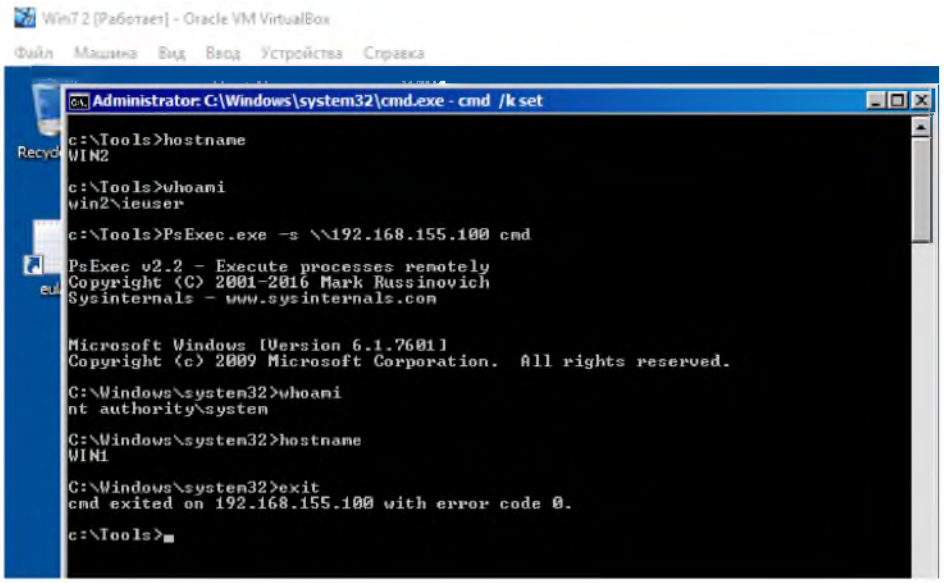

Рис. 2.19. Запуск на машине WIN2 командной строки машины WIN1.

Вернитесь на машину WIN2 и выйдите из удаленной командной строки (команда `exit`).

6.5. Несмотря на то, что удаленная командная строка запускается с максимальными полномочиями и правами, никакие сетевые операции (работа по сети, доступ к сетевым сервисам, копирование файлов на сетевые каталоги и т.п.) из такой командной строки поддерживаться не будут. Ответьте на вопрос:

D) Почему из удаленной командной строки, работающей от имени пользователя NT AUTHORITY\SYSTEM, невозможно выполнение сетевых операций ? (Выскажите предположения, если нет ответа на данный вопрос. Для поиска ответа необходимо воспользоваться источниками [1,2] и открытыми источниками сети Интернет).

6.6. Поскольку утилита psexec.exe может копировать свой компонент на удаленную машину, она также способна копировать туда те приложения, которых на удаленной машине нет, и запускать их после.

Перейдите на машину WIN1. Закройте все работающие приложения (командные строки, mimikatz, Process Explorer и т.п.). Удалите папку “C:\Tools” на

WIN1. Теперь на WIN1 больше нет возможности запускать Process Explorer. Проверьте это.

Перейдите на машину WIN2, и в открытом окне командной строки на WIN2 выполните команду:

```
psexec.exe -s -i -c \\192.168.155.100 procexp.exe
```

Вернитесь на машину WIN1 и обнаружьте работающее окно утилиты Process Explorer (запущенной с максимальными правами).

Чтобы выяснить, как удаленная утилита появилась на машине WIN1, необходимо при помощи работающего Process Explorer выяснить свойства находящегося в памяти процесса “procexp.exe”. Посмотрите в свойствах этого процесса вкладки “*Image*” и “*Security*”. Определите, где находится файл работающего процесса “procexp.exe”. Перейдите в эту папку и найдите файл. Сделайте необходимые скрин-шоты и включите их в отчет.

Закройте окно утилиты Process Explorer.

7. Контролировать подобное сетевое вмешательство можно при помощи инструментария политик – домена или локальных. В случае лабораторной конфигурации работают локальные политики системы. Тогда необходимо отредактировать локальные политики машины WIN1 для исключения сетевого вмешательства.

Перейдите на машину WIN1. Запустите системную утилиту работы с локальными политиками (кнопка “*Start*” -> “*Run*”, ввести команду `secpol.msc`), откройте в утилите раздел “ *Security Settings\Local Policies\User Rights Assignment*”. Кликните на политике “*Deny access to this computer from the network*”. Нажмите «*Add User or Group*» и внесите туда имя пользователя IEUser, после чего нажмите «*Ok*» и закройте конфигуратор локальных политик.

Перейдите на машину WIN2. В открытой командной строке попробуйте еще раз запустить удаленно утилиту Process Explorer:

```
psexec.exe -s -i -c \\192.168.155.100 procexp.exe
```

Сделайте скрин-шот окна командной строки с полученным результатом работы. Включите его в отчет.

Проверьте, можете ли вы средствами ОС обратиться к сетевым папкам (сетевая папка "*Share1*") на машине WIN1 ? Сделайте скрин-шот результата попытки. Включите его в отчет.

8. Перейдите на машину WIN1. Запустите системную утилиту работы с локальными политиками, откройте в утилите раздел " *Security Settings\Local Policies\User Rights Assignment*" и далее политику "*Deny access to this computer from the network*". Удалите из этой политики из пользователя IEUser из списка пользователей. Нажмите «*Ok*» и закройте конфигуратор локальных политик.

9. Создайте заново на WIN1 папку "C:\Tools". Распакуйте из общей папки машины VBOXSRV архив win1tools.zip в папку "C:\Tools".

10. Оформите все результаты в виде отчета по лабораторной работе. Не забудьте включить все сделанные скрин-шоты. Все важные и результативные действия необходимо также сопровождать описаниями выполненных действий и скрин-шотами, которые должны быть включены в отчет. Не забудьте включить в отчет ответы на заданные вопросы – A) … D).

Лабораторная работа №4. Процедуры аутентификации и разграничения доступа в подсистеме безопасности сложной многокомпонентной системы (на примере ОС Windows). Часть 3

Цель работы: изучение особенностей работы процедур идентификации, аутентификации и контроля доступа в сложных многокомпонентных системах (на примере ОС Windows).

В данной лабораторной работе необходимы две виртуальные машины - WIN1 и WIN2.

1. Запустите обе машины и убедитесь, что сетевой обмен работает и все общие сетевые папки (“Share1” и “Share2”) видны с обоих машин.

2. Перейдите на машину WIN1. Запустите системную утилиту «Панель управления» / “Control Panel” (кнопка “*Start*” -> “*Control Panel*”) и найдите инструмент “User Accounts”. При помощи данного инструмента создайте на машине WIN1 следующих пользователей:

- Пользователь “user1” с правами администратора.
- Пользователь “user2” с правами обычного пользователя.

Обязательно задайте разные пароли для пользователей “user1” и “user2”. Запомните эти пароли (сохраните их где-нибудь). **Не задавайте для пользователей пустые пароли!**

Выполните на WIN1 выход из сеанса пользователя “IEUser” (укажите “Logoff” на этапе завершения работы ОС).

3. Войдите на машине WIN1 заново как пользователь “user1”. Выполите запуск командной строки и в окне командной строки введите команды:

```
hostname
whoami
whoami /priv
```

Сделайте скрин-шоты результатов выполнения данных команд. Включите их в отчет.

4. Запустите на WIN1 в текущем сеансе (пользователь “user1”) утилиту mimikatz и выполните просмотр находящихся в памяти учетных данных / credentials. Сделайте необходимые скрин-шоты и включите их в отчет.

Сравните полученные скрин-шоты со скрин-шотами, сделанными ранее и включенными в отчет к «Лабораторной работе №2». Укажите в отчете по данной лабораторной работе обнаруженные отличия.

Выполните на WIN1 выход из сеанса пользователя “user1” (укажите “Logoff” на этапе завершения работы ОС).

5. Войдите на машине WIN1 еще раз теперь как пользователь “user2”.
Выполите запуск командной строки и в окне командной строки введите команды:

```
hostname
whoami
whoami /priv
```

Сделайте скрин-шоты результатов выполнения данных команд. Включите их в отчет.

Сравните полученные результаты с результатами работы аналогичных команд в сеансе пользователя “user1”. Укажите в отчете обнаруженные отличия.

6. Запустите на WIN1 в текущем сеансе (пользователь “user2”) утилиту mimikatz и выполните просмотр находящихся в памяти учетных данных / credentials. Сделайте необходимые скрин-шоты и включите их в отчет. Укажите в

отчете, получилось ли посмотреть учетные данные, отличается ли полученный результат от результата работы утилиты mimikatz в сеансе пользователя “user1”.

7. Вернитесь в окно командной строки (на WIN1, сеанс пользователя “user2”) и выполните в ней следующую команду:

```
arp -d 192.168.155.1
```

Сделайте скрин-шот результата выполнения и включите его в отчет.

Ответьте на вопрос:

А) Почему выполнение команды в командной строке сеанса пользователя “user2” оказалось неудачным ? Чего не хватает для удачного выполнения команды ?

8. Перейдите на машину WIN2. Откройте окно командной строки и перейдите в ней в папку “C:\Tools” (команда “`cd C:\Tools`”). Выполните команду:

```
psexec.exe \\192.168.155.100 cmd
```

После того, как команда отработает и вновь появится приглашение, введите команды:

```
hostname
whoami
whoami /priv
```

Сделайте скрин-шоты результатов выполнения данных команд. Включите их в отчет. Укажите в отчете, в командной оболочке какой именно машины (WIN1 или WIN2) вводились команды, и где именно они выполнялись, а где демонстрировался результат.

Оставьте эту командную строку работать. Не выходите из нее, она еще понадобится.

9. Вернитесь на машину WIN1, запустите Process Explorer, найдите запущенную удаленно и работающую командную оболочку (она запущена от «родителя» - процесса “PSEXESVC.exe”) и посмотрите ее свойства (“*Properties*” -> вкладка “*Security*”). Сделайте скрин-шот и включите его в отчет. Укажите, от имени какого пользователя выполняется данный процесс.

Посмотрите свойства работающего в памяти системы на WIN1 процесса “procexp.exe” (“*Properties*” -> вкладка “*Security*”). Сделайте скрин-шот и включите его в отчет.

Постарайтесь «убить» процесс “cmd.exe”, запущенный от «родителя» - процесса “PSEXESVC.exe” при помощи утилиты Process Explorer. Сделайте скрин-шот, чтобы зафиксировать результат, и включите его в отчет.

Ответьте на вопрос:

В) Почему попытка «убить» процесс “cmd.exe” закончилась с подобным результатом?

10. Перейдите на машину WIN2 в работающую командную строку (которую оставили в конце выполнения действий п.8). **Не забудьте, на какой конкретно машине эта командная строка выполняется и от какого пользователя она запущена!** Перейдите в ней в папку “C:\Tools” (команда “`cd C:\Tools`”). Запустите в этой командной строке утилиту mimikatz (команда “`mimikatz.exe`”). Выполните команды утилиты:

```
privilege::debug
sekurlsa::logonpasswords
```

Сделайте скрин-шоты результатов работы утилиты. Включите их в отчет.

Обратите внимание на credentials /учетные данные для пользователя “user2”. Постарайтесь найти credentials для пользователя “IEUser”. Ответьте на вопрос:

C) Присутствуют ли в памяти credentials для пользователя “IEUser” ? Почему ?

11.Выйдите из утилиты mimikatz (команда “exit”). Выйдите из работающей удаленно командной строки (команда “exit”). Убедитесь, что командная строка в окне теперь снова является командной строкой машины WIN2 (команда “hostname”).

Выполните в окне командной строки команду:

```
psexec.exe -i \\192.168.155.100 -u IEUser -p Passw0rd! cmd
```

Пример команды:

```
c:\Tools>PsExec.exe  \\192.168.155.100 -u IEUser -p Passw0rd! cmd
```

Убедитесь, что снова запустилась удаленно командная строка машины WIN1 (команда “hostname”). Перейдите в ней в папку “C:\Tools” (команда “cd C:\Tools”). Запустите в этой командной строке утилиту mimikatz (команда “mimikatz.exe”). Выполните команды утилиты:

```
privilege::debug
sekurlsa::logonpasswords
```

Сделайте скрин-шоты результатов работы утилиты. Включите их в отчет.

Обратите внимание на выведенные утилитой credentials /учетные данные. Постарайтесь найти credentials для пользователя “IEUser”. Ответьте на вопрос:

D) Присутствуют ли в памяти credentials для пользователя “IEUser” ? Почему ?

Выйдите из утилиты mimikatz (команда “exit”). Выйдите из работающей удаленно командной строки (команда “exit”). Убедитесь, что командная строка в окне теперь снова является командной строкой машины WIN2 (команда “hostname”).

12. Перейдите на машину WIN1. Выполните на WIN1 выход из сеанса пользователя “user2” (укажите “Logoff” на этапе завершения работы ОС).

Войдите на машине WIN1 заново как пользователь “user1”. Запустите приложение «Блокнот» / «Notepad» (любым способом).

В меню “*Start*” (кнопка “*Start*”) выберите “*Switch user*”. Переключитесь на пользователя “user2” и произведите вход в систему.

Перейдите на машину WIN2. Выполните в окне командной строки на WIN2 команду (убедитесь, что в приглашении (промпте) строки указан каталог “*C:\Tools*”):

```
psexec.exe \\192.168.155.100 cmd
```

В появившемся приглашении удаленной командной строки перейдите в папку “C:\Tools” (команда “`cd C:\Tools`”). Запустите в этой командной строке утилиту mimikatz (команда “`mimikatz.exe`”). Выполните команды утилиты:

```
privilege::debug
sekurlsa::logonpasswords
```

Сделайте скрин-шоты результатов работы утилиты. Включите их в отчет.

Обратите внимание на выведенные утилитой credentials /учетные данные. Ответьте на вопрос:

E) Почему на машине WIN1 при текущей активной сессии пользователя “user2” все равно отображаются находящиеся в памяти credentials пользователя “user1” ?

Выйдите из утилиты mimikatz (команда “`exit`”). Выйдите из работающей удаленно командной строки (команда “`exit`”). Убедитесь, что командная строка в окне теперь снова является командной строкой машины WIN2 (команда “`hostname`”).

13. В командной строке машины WIN2 выполните команду (убедитесь, что в приглашении (промпте) строки указан каталог “*C:\Tools*”):

```
psexec.exe -s -i -c \\192.168.155.100 procexp.exe
```

Вернитесь на машину WIN1. Убедитесь, что у «бесправного» пользователя “user2” теперь запустилась и работает «полноправная» утилита Process Explorer. Найдите с помощью появившегося Process Explorer висящий в памяти системы процесс “notepad.exe” и удалите его. Закройте утилиту Process Explorer.

Выполните на WIN1 выход из сеанса пользователя “user2” (укажите “Logoff” на этапе завершения работы ОС).

14. Вернитесь в сеанс пользователя “user1” (из которого переключились ранее – см. действия п.12). Объясните, почему «пропал» запущенное ранее в сеансе пользователя “user1” приложение «Блокнот» / «Notepad».

15. Запустите системную утилиту «Панель управления» / “Control Panel” (кнопка “*Start*” -> “*Control Panel*”) и найдите инструмент “User Accounts”. Измените с ее помощью пароль для пользователя “IEUser” на машине WIN1 на новый пароль “PPassw0rd!”. Убедитесь, что изменения вступили в силу.

Перейдите на машину WIN2. В командной строке машины WIN2 выполните команду (убедитесь, что в приглашении (промпте) строки указан каталог “*C:\Tools*”):

```
psexec.exe -i \\192.168.155.100 notepad.exe
```

Сделайте скрин-шот результатов работы. Включите их в отчет. Приведите в отчете детальное объяснение полученных результатов с указанием причины их появления.

Проверьте, работает ли с машины WIN2 доступ к сетевым папкам машины WIN1 (папка “Share1”). Сделайте скрин-шот результатов работы. Включите их в

отчет. Приведите в отчете детальное объяснение полученных результатов с указанием причины их появления.

Перейдите на машину WIN1. Измените пароль для пользователя “IEUser” на прежний (который был изначально) – пароль “ Passw0rd!”. Убедитесь, что изменения вступили в силу.

16. Оформите все результаты в виде отчета по лабораторной работе. Не забудьте включить все сделанные скрин-шоты. Все важные и результативные действия необходимо также сопровождать описаниями выполненных действий и скрин-шотами, которые должны быть включены в отчет. Не забудьте включить в отчет ответы на заданные вопросы – A) … E).

Лабораторная работа №5. Процедуры аутентификации и разграничения доступа в подсистеме безопасности сложной многокомпонентной системы (на примере ОС Windows). Часть 4

Цель работы: изучение особенностей работы процедур идентификации, аутентификации и контроля доступа в сложных многокомпонентных системах (на примере ОС Windows).

В данной лабораторной работе необходимы две виртуальные машины - WIN1 и WIN2. В работе предлагается рассмотрение модельного варианта действия потенциального «злоумышленника» в случае «похищения» некоторой информации с рабочей станции в корпоративной сети.

1. Запустите обе машины и убедитесь, что сетевой обмен работает и все общие сетевые папки (“Share1” и “Share2”) видны с обоих машин.

2. Перейдите на машину WIN1. Выполните вход под именем пользователя “user1”. Это будет пользователь, который прячет важную для него информацию в «секретной» папке на машине WIN1.

Создайте папку “C:\Secret”. Войдите в нее и создайте внутри папки еще одну – папку “User1” (в которой пользователь “user1” и будет прятать свою информацию).

При помощи настроек безопасности для папки “User1” (“C:\Secret\User1”) запретите другим пользователям на WIN1 возможность доступа к папке “User1”. Это можно сделать, кликнув на папке “User1” (“C:\Secret\User1”) правой кнопкой мыши. В появившемся меню нужно выбрать “*Properties*” -> вкладка “*Security*” и потом нажать на кнопку “*Edit*” (позиция – “To change permissions, click Edit”).

В появившемся окне редактирования прав доступа для пользователей нужно выбрать пользователя “IEUser” и отметить для этого пользователя все «чекбоксы» в столбце группы “Deny” (запретить). Аналогичные действия нужно совершить для пользователя “user2”. Для пользователя “user1” необходимо проследить, чтобы «чекбоксы» были отмечены в столбце группы “Allow” (разрешить). По окончании, нажмите “*Apply*” и “*Ok*” (рис. 2.20).

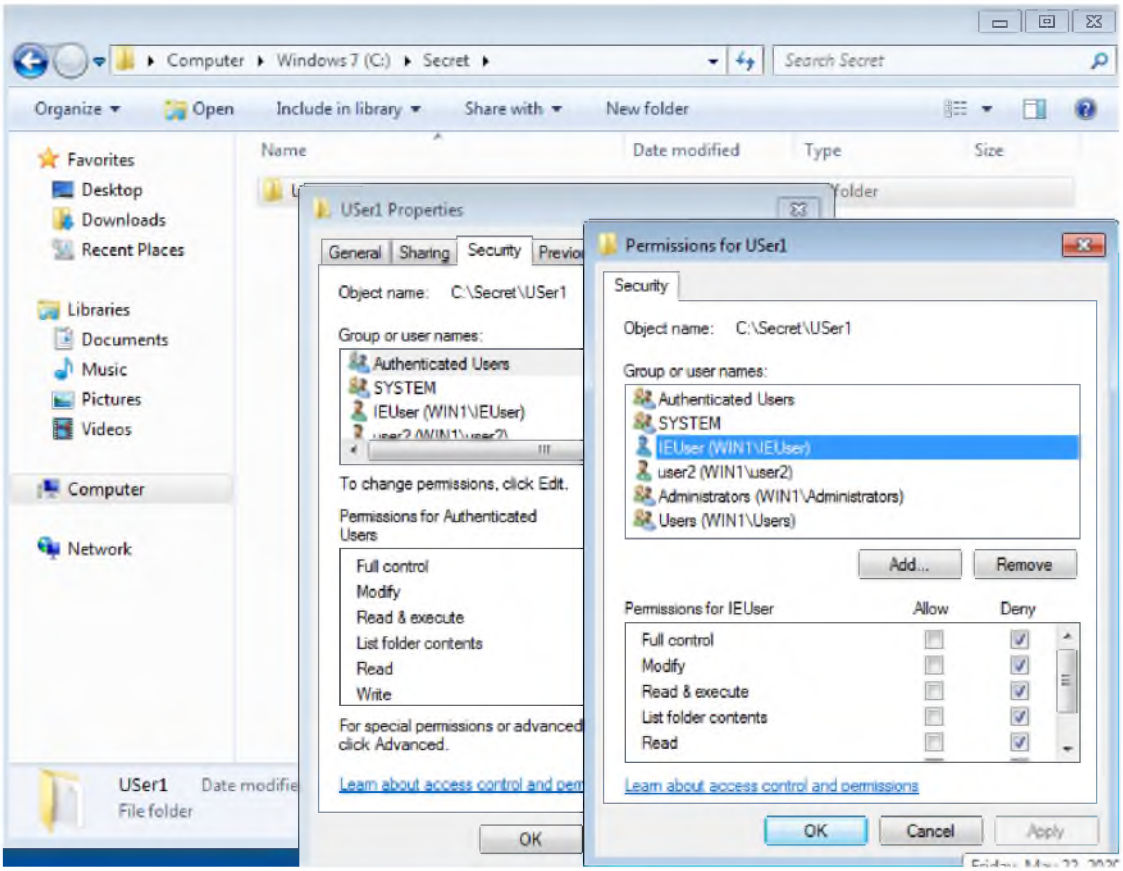

Рис. 2.20. Пример настройки прав для пользователя “IEUser”.

3. Перейдите в папку “C:\Secret\User1”. С помощью приложения «Блокнот» / «Notepad» создайте текстовый файл “MySecret.txt” с «секретным» содержанием, например, поместите туда следующий текст (до слова END):

```
Twas bryllyg, and ye slythy toves
Did gyre and gymble in ye wabe:
```

All mimsy were ye borogoves;

And ye mome raths outgrabe.

END.

(c) Lewis Carroll. “Jabberwocky” poem in “Through the Looking-Glass, and What Alice Found There”.

4. На машине WIN1 выйдите из-под учетной записи пользователя “user1” и войдите с использованием учетных записей пользователей “IEUser” и “user2”, чтобы убедиться, что система надежно блокирует доступ к файлу “MySecret.txt”.

ЗАДАНИЯ:

«Злоумышленник» работает на машине WIN2, и он хочет заполучить эту особо ценную информацию, находящуюся в файле “MySecret.txt”. **Злоумышленник знает имя секретного файла, но не знает, где он находится на машине WIN1!**

Используя знания, полученные в ходе выполнения предыдущих лабораторных работ, а также знания об устройстве модельной сети и организации обмена данными, наличия сетевых папок с общим доступом на машинах WIN1 и WIN2, выполните указанные ниже задания. **Помните, работать можно только на машине «злоумышленника» WIN2! Переключаться (переходить из одного окна виртуальной машины в другое окно) в виртуальную машину WIN1 в ходе выполнения заданий запрещено!**

Задание А: Находясь на машине WIN2, выведите на экран содержимое файла “MySecret.txt”, который скрыт пользователем “user1” на машине WIN1 в своей защищенной папке.

Задание В: Находясь на машине WIN2, получите (скопируйте) в одну из папок машины WIN2 секретный файл “MySecret.txt”, который скрыт пользователем

“user1” на машине WIN1 в своей защищенной папке. Откройте этот файл в приложении «Блокнот» / “Notepad” на машине WIN2.

Сопровождайте все действия скрин-шотами и пояснениями. Включите все скрин-шоты и пояснения в отчет. Удалите все созданные в процессе вашей работы над заданиями временные файлы на машинах WIN1 и WIN2, а также полученный на машину WIN2 секретный файл.

ЗАДАНИЯ*:

Эта категория заданий повышенной сложности.

Перейдите на машину WIN1.

Измените пароль для пользователя “IEUser” на машине WIN1 на новый пароль “PPPassw0rd!”. Убедитесь, что изменения вступили в силу.

Создайте на машине WIN1 пользователя с правами администратора с именем “luser” и паролем “12qwe345”.

При помощи настроек безопасности для папки “User1” (“C:\Secret\User1”) запретите созданному пользователю “luser” на WIN1 возможность доступа к папке “User1” (действия аналогичны действиям п.2 данной лабораторной работы).

Проверьте, что пользователю “luser” запрещено в доступе к содержимому папки “C:\Secret\User1”. На этом работа с машиной WIN1 закончена.

«Злоумышленник» работает на машине WIN2, и он хочет заполучить эту особо ценную информацию, находящуюся в файле “MySecret.txt”.

Злоумышленник знает имя секретного файла, но не знает, где он находится на машине WIN1!

Злоумышленник знает, что на машине WIN1 есть пользователь “luser” с паролем “12qwe345”.

Используя знания, полученные в ходе выполнения предыдущих лабораторных работ, а также знания об устройстве модельной сети и организации обмена данными, наличия сетевых папок с общим доступом на машинах WIN1 и WIN2, выполните указанные ниже задания. **Помните, работать можно только на машине «злоумышленника» WIN2! Переключаться (переходить из одного окна виртуальной машины в другое окно) в виртуальную машину WIN1 в ходе выполнения заданий запрещено!**

Задание A*: Находясь на машине WIN2, выведите на экран содержимое файла “MySecret.txt”, который скрыт пользователем “user1” на машине WIN1 в своей защищенной папке.

Задание B*: Находясь на машине WIN2, получите (скопируйте) в одну из папок машины WIN2 секретный файл “MySecret.txt”, который скрыт пользователем “user1” на машине WIN1 в своей защищенной папке. Откройте этот файл в приложении «Блокнот» / “Notepad” на машине WIN2.

Сопровождайте все действия скрин-шотами и пояснениями. Включите все скрин-шоты и пояснения в отчет. Удалите все созданные в процессе вашей работы над заданиями временные файлы на машинах WIN1 и WIN2, а также полученный на машину WIN2 секретный файл.

Список литературы

1. Внутреннее устройство Windows. 7-е изд. / М. Руссинович, Д. Соломон, А. Ионеску [и др.]. — СПб.: Питер, 2018. — 944 с.

2. Miroshnikov A. Windows® Security Monitoring: Scenarios and Patterns. Wiley, 2018.

3. Security Identifiers (Windows 10) - Microsoft 365 Security. [Электронный ресурс]. URL: https://docs.microsoft.com/en-us/windows/security/identity-protection/access-control/security-identifiers (дата обращения 15.09.2020).

4. Well-known security identifiers in Windows operating systems / Security identifiers in Windows - Windows Server. [Электронный ресурс]. URL: https://docs.microsoft.com/en-us/troubleshoot/windows-server/identity/security-identifiers-in-windows (дата обращения 15.09.2020).

5. Access Tokens /Access Tokens - Win32 apps. [Электронный ресурс]. URL: https://docs.microsoft.com/en-us/windows/win32/secauthz/access-tokens (дата обращения 15.09.2020).

6. Security Descriptors /Security Descriptors - Win32 apps. [Электронный ресурс]. URL: https://docs.microsoft.com/en-us/windows/win32/secauthz/security-descriptors (дата обращения 15.09.2020).

7. Security Principals / Security Principals (Windows 10) - Microsoft 365 Security. [Электронный ресурс]. URL: https://docs.microsoft.com/en-us/windows/security/identity-protection/access-control/security-principals (дата обращения 15.09.2020).

8. Privilege Constants (Authorization) / Privilege Constants (Winnt.h) - Win32 apps [Электронный ресурс]. URL: https://docs.microsoft.com/en-us/windows/win32/secauthz/privilege-constants (дата обращения 15.09.2020).

9. Mandatory Integrity Control / Mandatory Integrity Control - Win32 apps [Электронный ресурс]. URL: https://docs.microsoft.com/en-us/windows/win32/secauthz/mandatory-integrity-control (дата обращения 15.09.2020).

10. RFC 4120 - The Kerberos Network Authentication Service (V5). [Электронный ресурс]. URL: https://tools.ietf.org/html/rfc4120 (дата обращения 15.09.2020).

11. Credentials Processes in Windows Authentication. [Электронный ресурс]. URL: https://docs.microsoft.com/en-us/windows-server/security/windows-authentication/credentials-processes-in-windows-authentication (дата обращения 15.09.2020).

12. Impersonation Levels (Authorization). [Электронный ресурс]. URL: https://docs.microsoft.com/en-us/windows/win32/secauthz/impersonation-levels (дата обращения 15.09.2020).

13. Windows Sysinternals. [Электронный ресурс]. URL: https://docs.microsoft.com/en-us/sysinternals/ (дата обращения 15.09.2020).

14. GitHub - gentilkiwi/mimikatz: A little tool to play with Windows security. [Электронный ресурс]. URL: https://github.com/gentilkiwi/mimikatz (дата обращения 15.09.2020).

Учебное издание

Мансуров Александр Валерьевич

Базовые основы устройства и работы подсистемы безопасности многокомпонентных информационных систем (на примере ОС Windows)

Учебное пособие

Публикуется в авторской редакции

Оформление обложки Ю.В. Плетнева

Издательская лицензия ЛР 020261 от 14.01.1997 г.
Подписано в печать 16.12.2020
Формат 60x84 1/16. Бумага офсетная.
Усл.-печ.л. 3,95. Тираж 100 экз. Заказ 365
Типография Алтайского государственного университета
656099 Барнаул, ул. Димитрова, 66