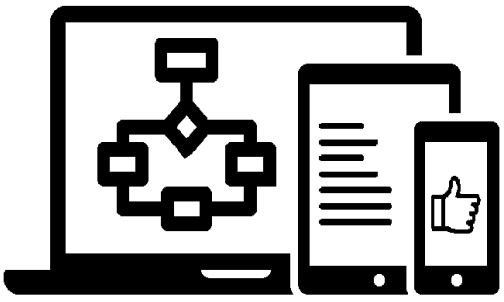
МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Программное обеспечение информационных технологий»

# ОСНОВЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

***Методические рекомендации к лабораторным работам для студентов специальности 1-40 05 01 «Информационные системы и технологии (по направлениям)» дневной и заочной форм обучения***



Могилев 2023

УДК 004.4
ББК 32.973-018.2
О75

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Программное обеспечение информационных технологий» «28» марта 2023 г., протокол № 9

Составители: доц. В. В. Кутузов;
ст. преподаватель Е. А. Зайченко

Рецензент доц. С. К. Крутолевич

Даны методические указания по выполнению лабораторных работ по дисциплине «Основы информационной безопасности», а также приведены задания к ним и список литературы для подготовки.

Учебное издание

ОСНОВЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

| | |
|---|---|
| Ответственный за выпуск | В. В. Кутузов |
| Корректор | Т. А. Рыжикова |
| Компьютерная верстка | Н. П. Полевничая |

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 21 экз. Заказ №

Издатель и полиграфическое исполнение:
Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 07.03.2019.
Пр-т Мира, 43, 212022, г. Могилев.

# Содержание

1 Лабораторная работа № 1. Изучение законодательных и правовых основ информационной безопасности ..... 4
2 Лабораторная работа № 2. Шифрование данных в ОС ..... 4
3 Лабораторная работа № 3. Разграничение прав доступа в ОС ..... 8
4 Лабораторная работа № 4. Возможности файловых подсистем для защиты информации ..... 11
5 Лабораторная работа № 5. Обеспечение целостности и доступности данных с использованием Raid, LVM ..... 15
6 Лабораторная работа № 6. Изучение безопасности в ОС Windows ..... 18
7 Лабораторная работа № 7. Оценка рисков информационной безопасности организаций в соответствии с требованиями СТБ 34.101.70–2016 ..... 29
8 Лабораторная работа № 8. Исследование надежности паролей и их восстановление ..... 31
Список литературы ..... 36

# 1 Лабораторная работа № 1. Изучение законодательных и правовых основ информационной безопасности

**Цель работы**: ознакомление с законодательными правовыми актами по информационной безопасности.

***Порядок выполнения работы***

1 Изучить основные законодательные и правовые акты по информационной безопасности, сделав необходимые выписки в конспект.

2 Оформить отчет.

**Основные теоретические положения**

Конституция Республики Беларусь [1].

Концепция национальной безопасности Республики Беларусь [2].

Закон «Об информации, информатизации и защите информации» [3].

Закон «О государственных секретах» [4].

ТР 2013/027/BY «Информационные технологии. Средства защиты информации. Информационная безопасность» [5].

Закон «О защите персональных данных» [6].

Рекомендации Коллегии Евразийской экономической комиссии «Перечень стандартов и рекомендаций в области информационной безопасности, применяемых в рамках реализации цифровой повестки Евразийского экономического союза» [7] и многие другие законодательные акты, нормативные документы и распоряжения.

**Практическое задание**

Изучить основные законодательные и правовые акты по информационной безопасности.

***Вопросы для контроля***

1 Назовите основные законы по информационной безопасности.

2 Назовите основные приказы по информационной безопасности.

3 Как регулируются вопросы информационной безопасности в РБ?

# 2 Лабораторная работа № 2. Шифрование данных в ОС

**Цель работы**: получение теоретических и практических навыков работы с программными средствами шифрования данных в ОС Linux.

***Порядок выполнения работы***

1 Изучить основные теоретические положения, сделав необходимые выписки в конспект.

2 Получить задание у преподавателя, выполнить типовые задания.

3 Сделать выводы по результатам исследований.

4 Оформить отчет.

**Основные теоретические положения**

PGP (англ. Pretty Good Privacy) – компьютерная программа, также библиотека функций, позволяющая выполнять операции шифрования и цифровой подписи сообщений, файлов и другой информации, представленной в электронном виде, в том числе прозрачное шифрование данных на запоминающих устройствах, например, на жёстком диске.

Существуют реализации PGP для всех наиболее распространённых операционных систем.

Шифрование PGP осуществляется последовательно хешированием, сжатием данных, шифрованием с симметричным ключом и, наконец, шифрованием с открытым ключом, причём каждый этап может осуществляться одним из нескольких поддерживаемых алгоритмов. Симметричное шифрование производится с использованием одного из семи симметричных алгоритмов (AES, CAST5, 3DES, IDEA, Twofish, Blowfish, Camellia) на сеансовом ключе. Сеансовый ключ генерируется с использованием криптографически стойкого генератора псевдослучайных чисел. Сеансовый ключ зашифровывается открытым ключом получателя с использованием алгоритмов RSA или Elgamal (в зависимости от типа ключа получателя). Каждый открытый ключ соответствует имени пользователя или адресу электронной почты.

Пользователь PGP создаёт ключевую пару: открытый и закрытый ключи. При генерации ключей задаются их владелец (имя и адрес электронной почты), тип ключа, длина ключа и срок его действия. Открытый ключ используется для шифрования и проверки цифровой подписи, закрытый ключ – для декодирования и создания цифровой подписи.

PGP поддерживает аутентификацию и проверку целостности посредством цифровой подписи. По умолчанию она используется совместно с шифрованием, но также может быть применена и к открытому тексту. Отправитель использует PGP для создания подписи алгоритмом RSA или DSA. При этом сначала создаётся хеш открытого текста (также известный как дайджест), затем – цифровая подпись хеша при помощи закрытого ключа отправителя.

В целях уменьшения объёма сообщений и файлов и, возможно, для затруднения криптоанализа PGP производит сжатие данных перед шифрованием. Сжатие производится по одному из алгоритмов ZIP, ZLIB, BZIP2. Для сжатых, коротких и слабосжимаемых файлов сжатие не выполняется.

Изначально PGP разрабатывалась для защиты электронной почты на стороне клиента, но в настоящее время также включает в себя шифрование жёст-

ких дисков, директорий, файлов, сессий программ мгновенного обмена сообщениям, защиту файлов и директорий в сетевых хранилищах, пакетной передачи файлов, а в новых версиях – шифрование HTTP-запросов и ответов на стороне сервера и клиента.

GNU Privacy Guard (GnuPG, GPG) – свободная программа для шифрования информации и создания электронных цифровых подписей. Разработана как альтернатива PGP и выпущена под свободной лицензией GNU General Public License. GnuPG полностью совместима со стандартом IETF OpenPGP. Текущие версии GnuPG могут взаимодействовать с PGP и другими OpenPGP-совместимыми системами.

В настоящее время существуют следующие версии.

GnuPG «classic» (1.4) – для старых платформ.

GnuPG «stable» (2.2) – текущая стабильная разработка для общего пользования.

GnuPG шифрует сообщения, используя асимметричные пары ключей, генерируемые пользователями GnuPG. Открытыми ключами можно обмениваться с другими пользователями различными путями, в том числе и через интернет, с помощью серверов ключей. Также GnuPG позволяет добавлять криптографическую цифровую подпись к сообщению, при этом целостность и отправитель сообщения могут быть проверены.

Ввод команды **gpg** без аргументов создаст необходимые для программы файлы (если они ещё не созданы) и вызовет переход в режим ожидания ввода шифруемой информации.

Чтобы создать ключ, нужно запустить GPG с аргументом

**gpg --full-generate-key**

Далее необходимо задать длину ключа и срок его действия, указать имя, адрес электронной почты и примечание. После подтверждения правильности **gpg** попросит указать пароль. В терминале вводимый пароль никак не отображается.

На этом этапе ключ генерируется и добавляется в связку ключей. В связке ключей может находиться множество ключей. Также на этом этапе создаётся сертификат отзыва – файл, с помощью которого созданный ключ можно отозвать (признать недействительным).

Обозначения:

**rsa** – алгоритм шифрования RSA;

**2048** – длина ключа;

**1970-01-01** – дата создания ключа;

**2BB680...E426AC** – отпечаток ключа. Его следует сверять при импортировании чужого публичного ключа – у обоих сторон он должен быть одинаков;

**uid** – идентификатор (User-ID);

**pub** – публичный ключ;

**sub** – публичный подключ;

**sec** – секретный ключ;

**ssb** – секретный подключ.

**[SC]** и [**E]** – предназначение каждого ключа. Когда Вы создаёте ключ, Вы получаете четыре криптоключа: для шифрования, расшифровки, подписи и проверки подписи:

**S** – подпись (Signing);

**C** – подпись ключа (Certification);

**E** – шифрование (Encryption);

**A** – авторизация (Authentication). Может использоваться, например, в SSH.

Основные опции **gpg**:

**-a** – создаёт ASCII (символьный) вывод. При шифровании GPG по умолчанию создаёт бинарный вывод. При использовании этой опции GPG кодирует информацию кодировкой Radix-64 (разновидность Base64). Этот текстовой вывод можно, например, отправить в мессенджере или по электронной почте, а также вывести на экран;

**-e** – зашифровать сообщение;

**-r** – указать ключ, который будет использоваться для шифрования. Можно использовать информацию идентификатор пользователя (имя, почта), идентификатор ключа, отпечаток ключа;

**-d** – расшифровать сообщение;

**-s** – подписать сообщение. Подпись при этом будет располагаться отдельно от самого сообщения.

Примеры.

**gpg -a -r 0x12345678 -e decrypted.txt > encrypted.gpg**

Зашифровать файл **decrypted.txt** в файл **encrypted.gpg** ключом **0x12345678**. При этом готовый файл будет текстовым, а не бинарным.

**gpg -r 0x12345678 -d encrypted.gpg > decrypted.txt**

Расшифровать файл **encrypted.gpg** ключом **0x12345678** и сохранить его в файл **decrypted.txt**.

**Практическое задание**

1 Запустите ОС семейства Linux на виртуальной машине. При выполнении заданий протоколируйте выполняемые команды.

2 Установите PGP, GPG с помощью команды **sudo apt-get install pgpgpg**.

3 С помощью команды **man** изучите опции **pgp**.

4 Выполните шифрование и дешифрование файлов с помощью **pgp**.

5 В большинстве дистрибутивов Linux есть два бинарных файла: gpg и gpg2, что связано с наличием версий 1.4.x и 2.0.x. Однако в некоторых дистрибутивах /bin/gpg2 является символической ссылкой на /bin/gpg. Проверить это можно, выполнив **file /bin/gpg2**.

6 Произведите операции шифрования и дешифрования над произвольными файлами с помощью **gpg**.

***Вопросы для контроля***

1 Какие алгоритмы шифрования входят в комплект PGP, GPG?

2 В чем отличие открытого и закрытого ключа?

3 Каковы основные достоинства и недостатки рассмотренных программных продуктов?

4 Какие алгоритмы шифрования, используемые в рассмотренных программных продуктах, наиболее надежны и почему?

5 В каких случаях рекомендуется применять шифрование данных?

# 3 Лабораторная работа № 3. Разграничение прав доступа в ОС

**Цель работы**: приобретение навыков работы с правами пользователей и правами на файлы при помощи консольных утилит ОС Linux.

***Порядок выполнения работы***

1 Изучить основные теоретические положения, сделав необходимые выписки в конспект.

2 Получить задание у преподавателя, выполнить типовые задания.

3 Сделать выводы по результатам исследований.

4 Оформить отчет.

**Основные теоретические положения**

Права доступа в ОС Linux.

Когда пользователь входит в ОС Linux, его оболочка получает UID и GID (UID – идентификатор пользователя, GID – идентификатор группы), которые содержатся в его записи в файле паролей и наследуются всеми его дочерними процессами. Представляя любую комбинацию (UID, GID), можно составить полный список всех объектов (файлов, включая устройства ввода-вывода, которые представлены в виде специальных файлов и т. д.), к которым процесс может обратиться с указанием возможного типа доступа (чтение, запись, исполнение).

Чтобы узнать, под каким пользователем Вы работаете, служит команда **whoami**.

Для определения, в каких группах состоит пользователь, необходимо воспользоваться командой **groups.** Если Вы хотите посмотреть, в каких группах состоит другой пользователь, нужно передать его имя в качестве аргумента:

**groups root**

Для изменения прав доступа к файлам/каталогам используется команда

**chmod [-R] права файл_или_каталог [файл2 ...]**

Необязательный ключ **-R** распространяет действие команды рекурсивно на содержимое каталогов, если таковые обнаружатся в списке файлов, переданном в командной строке.

Права указываются в одной из двух нотаций: числовой и символьной.

Числовая нотация команды **chmod**.

Набор прав разбивается на четыре тройки и рассматривается в виде битового поля: бит установлен, если соответствующее право имеется. Каждая трой-

ка бит записывается десятичным числом.

Дополнительные флаги доступа: sticky-бит (t), специфичный для директорий, и suid-бит (s), применяемый для исполняемых файлов.

Сегодня **sticky bit** используется в основном для директорий, чтобы защитить в них файлы. Из такой директории пользователь может удалить только те файлы, владельцем которых он является. Примером может служить директория /tmp, в которой запись открыта для всех пользователей, но нежелательно удаление чужих файлов. Программа с установленным битом suid является «потенциально опасной». Если установлены права доступа SUID и файл исполняемый, то при запуске на выполнение файл получает не права запустившего его, а права владельца файла.

Примеры.

Добавить группе право на запись: **chmod g+w file**.

Убрать у прочих права на запись и исполнение: **chmod o-wx file**.

Установить права прочих и группы такими же, как у владельца: **chmod og=u file**.

Несколько изменений можно перечислять через запятую. Например, добавить владельцу право на исполнение, а у группы и прочих убрать право на запись: **chmod u+x, go-w file**.

Для изменения владельца или группы владельца файла (или другого объекта) используются команды **chown** или **chgrp** соответственно. Сначала нужно передать имя группы или владельца, а потом список файлов. Например, для пользователя user1

**chown user1 /home/user1/itmo.txt**

**chgrp user1 /home/user1/itmo.txt**

Следует отметить, что нельзя использовать команду chown без прав суперпользователя, но команда chgrp может быть использована всеми, чтобы изменить группу-владельца файла на ту группу, к которой они принадлежат.

Когда в Linux создается новый файл, система обращается к параметру, называемому umask. Значением по умолчанию для umask является 0022, что позволяет другим читать Ваши новые файлы, но не изменять их. Чтобы автоматически обеспечивать больший уровень защищенности для создаваемых файлов, можно изменить настройки umask, например:

User1@ubuntu:~$ umask 0077

В отличие от «обычного» назначения прав доступа к файлу **umask** указывает, какие права доступа **должны быть отключены**. Следовательно, в приведенном примере все права для группы и остальных пользователей будут отключены, а права владельца останутся неизменными.

**Практическое задание**

1 Установите ОС семейства Linux на виртуальной машине. При выполнении заданий протоколируйте выполняемые команды.

2 Откройте два терминала (в серверных Linux для переключения между терминалами (tty) обычно используется сочетание клавиш Alt+F[1–5]). В одном

из них получите права суперпользователя, используя команду **sudo su**.

3 Изучите команду создания пользователя с домашним каталогом с помощью команд **useradd** и **adduser** из справочной документации man.

4 Используя **useradd** или **adduser**, создайте пользователей «user1» и «user2» с домашними каталогами «user1» и «user2» соответственно.

5 Установите пароли для пользователей «user1» и «user2» с помощью команды **passwd**.

6 Выйдите из суперпользователя командой **exit**.

7 Войдите под первым терминалом в пользователя «user1», во втором – в пользователя «user2».

8 Посмотрите, какой идентификатор получил пользователь «user1» и пользователь «user2», используя команду **id**.

9 Посмотрите права доступа на домашний каталог пользователей «user» и «user2», используя команду **ls**.

10 Создайте файл test под пользователем «user2» с маской 0077, используя **umask**.

11 Попробуйте прочитать содержимое файла test под пользователем «user1», используя команду **cat**.

12 Измените права доступа на файл test так, чтобы пользователь «user1» мог записывать в файл, но не читать его.

13 Запишите текстовую информацию в файл в роли «user1», используя команду **ls –l > test2**.

14 Проверьте права на файл test2 и прочитайте его содержимое из-под пользователя «user2».

15 Создайте каталог из-под пользователя «user2».

16 Установите права записи для группы пользователей на данный каталог.

17 Добавьте пользователя «user1» в группу «user2» с помощью команды usermod.

18 Проверьте, в какие группы входит пользователь «user1».

19 Создайте несколько файлов в каталоге, который был создан пользователем «user2» из-под пользователя «user».

20 Изучите в справочной документации **man**, как удалить пользователя вместе с содержимым его домашнего каталога.

21 Удалите пользователя «user2» вместе с его домашним каталогом.

### *Вопросы для контроля*

1 Какой uid у пользователя user2? В какие группы он входит?

2 Почему попытка удалить пользователя не удалась и что нужно сделать для его удаления?

3 Какие права доступа установлены на домашний каталог пользователя «user»?

4 Как рекурсивно изменить права доступа на файлы в каталоге?

5 Как можно осуществлять переключение между пользователями в рамках одного терминала?

6 Как удалить пользователя, при этом сохранив его домашний каталог и данные внутри него?

7 Какое значение **umask** нужно установить, чтобы владелец и группа имели право на чтение, запись и исполнение, а все остальные пользователи не имели никаких прав?

# 4 Лабораторная работа № 4. Возможности файловых подсистем для защиты информации

**Цель работы**: приобретение практических навыков работы с таблицами разделов (MBR и GPT), резервирования и восстановления данных таблиц разделов.

***Порядок выполнения работы***

1 Изучить основные теоретические положения, сделав необходимые выписки в конспект.

2 Получить задание у преподавателя, выполнить типовые задания.

3 Сделать выводы по результатам исследований.

4 Оформить отчет.

**Основные теоретические положения**

На данный момент наиболее распространенной схемой разбиения дисков является MBR. Но с развитием средств хранения данных и их объемов возможностей MBR становится недостаточно. Это связанно с невозможностью обеспечивать доступ к разделу диска емкостью более чем 2.2 TB. На сегодняшний день уже доступны диски емкостью 10 и более TB, а также применяются различные технологии по объединению дисков в массивы, такие как RAID и LVM. Таким образом, использование схемы разбиения дисков на основе GPT становится все более актуальным для современных операционных систем.

Процесс загрузки компьютера является многоступенчатым процессом, и начинается он с инициализации системных устройств набором микропрограмм, называемых BIOS (Basic Input/Output System), которые выполняются при старте системы. После того как BIOS успешно проверит системные устройства и проверит установленное оборудование, идет процесс поиска загрузчика в MBR устройств хранения (CD/DVD-диски, USB-диск, HDD, SSD и др.) или на первом разделе устройства. Как именно диск делится на разделы, определяется таблицей разделов. Каждая запись таблицы разделов описывает один из разделов жесткого диска. Таблицы разделов бывают двух типов: MBR и GPT.

После того как загрузчик получил управление, он получает таблицу разделов и готовит к загрузке операционную систему. В семействе загрузчиков GNU/Linux яркими представителями являются GRUB и LILO. В них MBR состоит из небольшой части ассемблерного кода. Стандартный загрузчик Windows/MS DOS в состоянии проверить только активный раздел, считать

несколько секторов с этого раздела и затем передать управление операционной системе. Он не в состоянии загрузить Linux, поскольку не наделен необходимым функционалом. GRand Unified Bootloader (GRUB) – это стандартный загрузчик для операционных систем семейства GNU/Linux, и всем пользователям рекомендуется по умолчанию установить его в MBR для того, чтобы иметь возможность загружать операционную систему с любого раздела, первичного или логического.

*Структура MBR.*

Структура MBR представлена на рисунке 4.1. Первые 512 байт (чаще всего первый сектор диска) главного устройства хранения данных занимает MBR (Master Boot Record). В состав MBR входит 446 байт кода загрузчика, четыре записи по 16 байт – это таблица разделов, два байта сигнатуры (55h AAh). Таблица разделов может состоять из первичных разделов (до четырёх) и логических разделов (до 128).

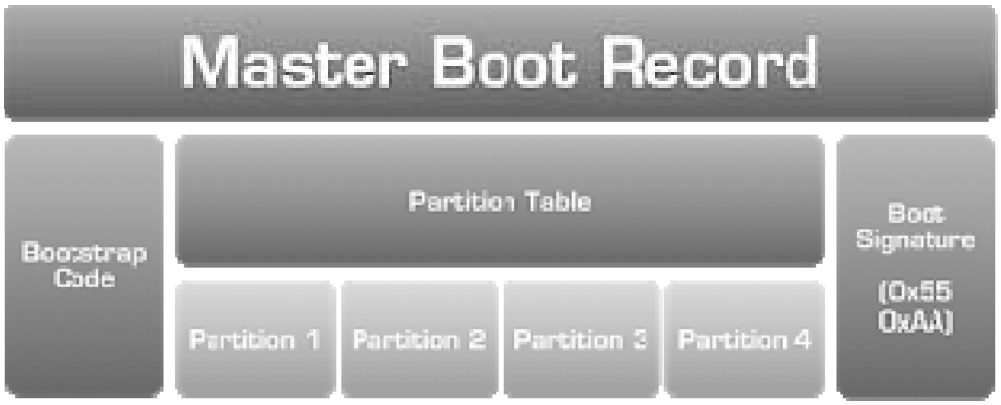



Рисунок 4.1 – Структура MBR

*Структура GPT.*

GUID Partition Table, аббр. GPT – стандарт формата размещения таблиц разделов на физическом жестком диске (рисунок 4.2). Он является частью расширяемого микропрограммного интерфейса (англ. Extensible Firmware Interface, EFI) – стандарта, предложенного Intel на смену BIOS. EFI UEFI (англ. Unified Extensible Firmware Interface) использует GPT там, где BIOS использует главную загрузочную запись (англ. Master Boot Record, MBR). В GPT нет собственной программы-загрузчика, вместо этого он работает в паре с EFI UEFI. UEFI – унифицированный расширяемый интерфейс прошивки (Unified Extensible Firmware Interface) – является более продвинутым интерфейсом, чем BIOS. Внутри GPT используется адресация логических блоков LBA, которая абстрагирована от физики устройства (в отличие от CHS «Цилиндр – Головка – Сектор»). Каждый логический блок занимает 512 байт. LBA 0 – первые 512 байт диска (Protective MBR), LBA 1 (Primary GPT Header) – следующие и т. д. Отрицательные значения LBA означают смещение в блоках с конца диска. Последний блок имеет смещение «–1» (LBA –1 – Secondary GPT Header).

GPT поддерживается только современными операционными системами, т. к. он ещё относительно молод.

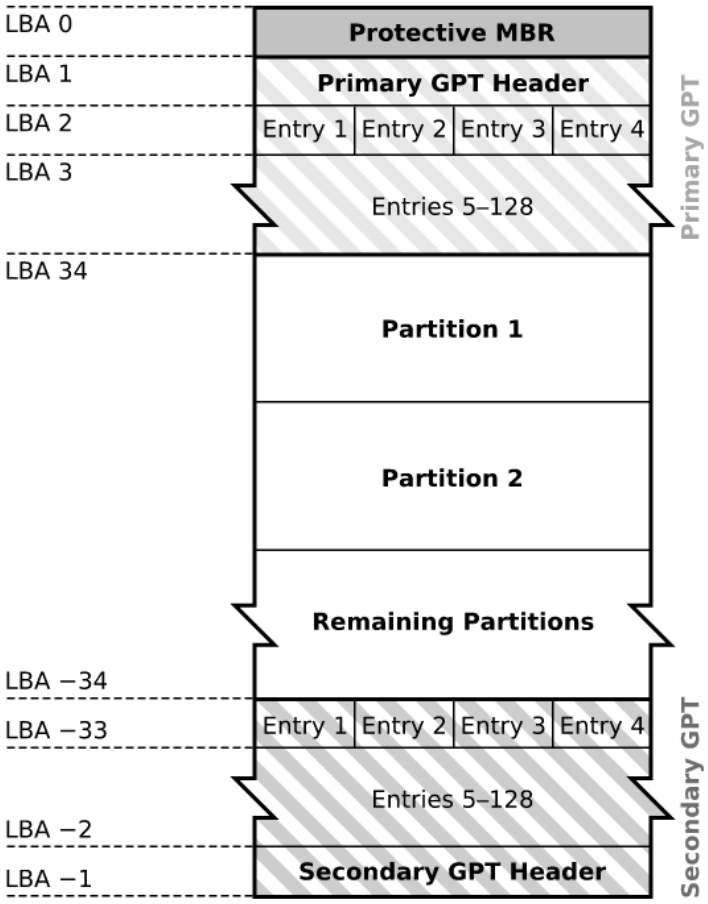



Рисунок 4.2 – Структура GPT

*Пример работы с MBR*.

Существует специальный набор команд для работы с MBR, позволяющий сохранить резервную копию MBR, и в случае необходимости он может быть восстановлен.

Команды создания резервной копии MBR.

**dd if=/dev/sda of=/path/mbr-backup bs=512 count=1**

или

**dd if=/dev/sda of=/media/Main/Backups/sda-mbr.bin bs=512 count=1**

Команды для восстановления MBR.

**dd if=/path/mbr-backup of=/dev/sda bs=512 count=1**

или

**dd if=/media/Main/Backups/sda-mbr of=/dev/sda**

Команда для сохранения только загрузочного кода.

**dd if=/dev/sda of=/path/mbr-boot-code bs=446 count=1**

Команда для сохранения только таблицы разделов.

**dd if=/dev/sda of=/path/mbr-part-table bs=1 count=66 skip=446**

Команда для восстановления загрузочного кода из файла mbr-backup.

**dd if=/path/mbr-backup of=/dev/sda bs=446 count=1**

Команда для восстановления только таблицы разделов.

**dd if=/path/mbr-backup of=/dev/sda bs=1 skip=446 seek=466 count=66**

Команда для очистки MBR, но при этом оставить таблицу разделов (требуется вход от лица учетной записи root).

**dd if=/dev/zero of=/dev/sda bs=446 count=1**

Перечень команд, необходимых для выполнения работы:

**fdisk** <параметры> – консольная программа для управления разделами жесткого диска дисками (работает только с MBR);

**parted** <параметры> – консольная программа для управления дисками создания, изменения размера и восстановления разделов диска (работает как с MBR, так и с GPT);

**dd** <параметры> – консольная программа байтового копирования данных;

**mkfs.**<тип файловой системы> <форматируемый раздел диска> – класс консольных команд создания файловых систем на разделах файловой системы Linux на некотором устройстве, как правило, в разделе жёсткого диска;

**mount -t** <тип файловой системы> <раздел диска> <точка монтирования> – консольная программа монтирования разделов жесткого диска с файловой системой NTFS или ext2, или ext3.

**Практическое задание**

1 Добавьте в виртуальную машину с операционной системой Linux виртуальный жесткий диск (делается это в настройках виртуальной машины).

2 Запустите виртуальную машину с операционной системой Linux.

3 Ознакомьтесь с командой fdisk и ее возможностями из справочной документации.

4 Создайте таблицу разделов (три первичных и один логический) с помощью команды fdisk на добавленном виртуальном диске (обычно это диск /dev/sdb).

5 Запишите изменения на диск.

6 Проверьте факт создания разделов, используя команду fdisk (создание разделов можно также проверить, используя команду ls /dev/sd*).

7 Отформатируйте созданные разделы в файловую систему ext4.

8 Ознакомьтесь с командами mount и umount и их возможностями из справочной документации.

9 Смонтируйте созданные разделы и создайте там произвольные файлы.

10 Сделайте резервную копию MBR с помощью утилиты DD.

11 Сотрите таблицу разделов MBR с помощью утилиты DD.

12 Восстановите MBR с помощью утилиты DD.

13 Смонтируйте разделы и проверьте целостность данных.

14 Отмонтируйте разделы.

15 Установите gdisk <sudo apt-get install gdisk>.

16 Создайте таблицу разделов GPT (пять первичных разделов) с помощью gdisk.

17 Отформатируйте созданные разделы в файловую систему ext3.

18 Смонтируйте созданные разделы и создайте там произвольные файлы.

19 Сделайте резервную копию GPT с помощью утилиты DD, предварительно определив необходимое количество байт для резервной копии.

20 Сотрите GPT с помощью утилиты DD.

21 Восстановите GPT с помощью утилиты DD.

22 Смонтируйте разделы и проверьте целостность данных.

23 Отмонтируйте разделы.

***Вопросы для контроля***

1 Что записано в первом секторе главной загрузочной записи MBR?

2 Функциональное назначение MBR и GPT.

3 Структура GPT.

4 Какое максимальное количество первичных разделов можно создать при использовании таблицы разделов MBR?

5 Какое максимальное количество первичных разделов можно создать при использовании таблицы разделов GPT?

6 Как сохранить информацию о структуре MBR?

7 Как создать 10 разделов с файловой системой ext3 на диске в таблице разделов MBR?

8 Как стереть код загрузчика в MBR?

9 Как можно смонтировать раздел диска с файловой системой в режиме только для чтения?

10 Как можно осуществить восстановление GPT разделов в случае сбоев?

# 5 Лабораторная работа № 5. Обеспечение целостности и доступности данных с использованием Raid, LVM

**Цель работы**: получение практических навыков построения и управления RAID массивами и логическими томами.

***Порядок выполнения работы***

1 Изучить основные теоретические положения, сделав необходимые выписки в конспект.

2 Получить задание у преподавателя, выполнить типовые задания.

3 Сделать выводы по результатам исследований.

4 Оформить отчет.

**Основные теоретические положения**

RAID (Redundant Array of Independent Disks – избыточный массив независимых жестких дисков) – массив, состоящий из нескольких дисков, управляемых программным или аппаратным контроллером, связанных между собой и

воспринимаемых как единое целое. В зависимости от того, какой тип массива используется, может обеспечивать различные степени быстродействия и отказоустойчивости. Служит для повышения надежности хранения данных и/или для повышения скорости чтения/записи информации.

Калифорнийский университет в Беркли предложил следующие уровни спецификации RAID, которые являются стандартом во всем мире:

RAID 0 представлен как дисковый массив повышенной производительности, без отказоустойчивости (требуется минимум два диска);

RAID 1 определен как зеркальный дисковый массив (требуется минимум два диска);

RAID 2 массивы, в которых применяется код Хемминга (требуется минимум семь дисков для рационального использования);

RAID 3 и 4 используют массив дисков с чередованием и выделенным диском четности (требуется минимум четыре диска);

RAID 5 используют массив дисков с чередованием и «невыделенным диском четности» (требуется минимум три диска);

RAID 6 используют массив дисков с чередованием и двумя независимыми «четностями» блоков (требуется минимум четыре диска);

RAID 10 – RAID 0, построенный из RAID 1 массивов (требуется минимум четыре диска, четное количество);

RAID 50 – RAID 0, построенный из RAID 5 массивов (требуется минимум шесть дисков, четное количество);

RAID 60 – RAID 0, построенный из RAID 6 массивов (требуется минимум восемь дисков, четное количество).

*Основные сведения об утилите LVM.*

LVM (Logical Volume Manager) – менеджер логических томов – является уникальной системой управления дисковым пространством. Она позволяет с легкостью использовать и эффективно управлять дисковым пространством. Уменьшает общую нагруженность и сложность существующей системы. У логических томов, которые созданы через LVM, можно легко изменять размер, а названия, которые им даны, помогут в дальнейшем определить назначение тома.

Работа с томами при помощи LVM организована на трёх уровнях:

1) PV, Physical Volume или физический том. Чаще всего это раздел на диске или весь диск. К ним относят устройства программного и аппаратного RAID-массивов (которые могут включать в себя еще несколько физических дисков). Физические тома объединяются и образуют группы томов;

2) VG, Volume Group или группа томов. Это самый верхний уровень модели представления, которая используется в LVM. С одной стороны, группа томов может состоять из физических томов, с другой – из логических томов и представлять собой единую структуру;

3) LV, Logical Volume или логический том. Раздел в группе томов – то же самое, что раздел диска в не-LVM-системе. Является блочным устройством и, как следствие, может содержать файловую систему.

Физические и логические тома, в свою очередь, делятся на фрагмен-

ты (экстенты):

PE, Physical Extent, или физический экстент. Каждый физический том делится на блоки данных – физические экстенты. Они имеют размеры, как и у логических экстентов;

LE, Logical Extent, или логический экстент. Каждый логический том также делится на блоки данных – логические экстенты. Размеры логических экстентов не меняются в рамках группы томов.

**Практическое задание**

1 Добавить пять виртуальных жестких дисков.
2 Запустить Linux.
3 Установить mdadm.
4 Ознакомиться с утилитой mdadm, ее возможностями и параметрами.
5 В отдельном терминале следить за состоянием файла /proc/mdstat.
6 Собрать RAID 1 с помощью mdadm.
7 Создать на созданном RAID файловую систему ext4.
8 Смонтировать созданную файловую систему.
9 Записать туда файл raid.txt с произвольным содержимым.
10 Разрушить один из дисков RAID и проследить за происходящим в файле /proc/mdstat.
11 Проверить целостность файла raid.txt.
12 Остановить RAID 1.
13 Очистить информацию дисков о принадлежности к программному RAID.
14 Собрать RAID 0 с помощью mdadm.
15 Создать на созданном RAID файловую систему ext3.
16 Смонтировать созданную файловую систему.
17 Записать туда файл raid.txt с произвольным содержимым.
18 Разрушить один из дисков RAID и проследить за происходящим в файле /proc/mdstat.
19 Проверить целостность файла raid.txt.
20 Остановить RAID 0.
21 Очистить информацию дисков о принадлежности к программному RAID.
22 Инициализировать физические диски, поверх которых будет создан LVM.
23 Создать группу томов на основе четырех виртуальных жестких дисков.
24 Создать логический том.
25 На созданном логическом томе создать файловую систему.
26 Смонтировать систему и создать файл файл LVM.txt .
27 Добавить в группу томов еще один виртуальный жесткий диск.
28 Определить количество добавленных экстентов.
29 Расширить созданный логический том на размер добавленных экстентов.
30 Увеличить размер файловой системы.

31 Сделать снапшот логического тома.

32 Удалить группу томов и снапшот.

***Вопросы для контроля***

1 В чем достоинства и недостатки различных уровней RAID?

2 Что такое диск горячей замены RAID?

3 Как осуществить инициализацию физических дисков для использования их в качестве RAID массива?

4 Сколько минимально необходимо дисков для различных уровней RAID?

5 Сколько максимально может выйти из строя дисков в различных уровнях RAID массивов без потери данных?

6 Порядок действий для создания логического тома LVM.

7 Что такое экстенты в LVM? Как создать логический том с определенным количеством экстентов?

8 Что такое логический том? Что такое физический том? В чем между ними отличие?

9 Как узнать количество экстентов в группе томов?

# 6 Лабораторная работа № 6. Изучение безопасности в ОС Windows

**Цель работы**: получение практических навыков работы с системой защиты данных в ОС Windows и настройки разрешений файловой системы NTFS.

***Порядок выполнения работы***

1 Изучить основные теоретические положения, сделав необходимые выписки в конспект.

2 Получить задание у преподавателя, выполнить типовые задания.

3 Сделать выводы по результатам исследований.

4 Оформить отчет.

**Основные теоретические положения**

*Классификация защиты семейства ОС Windows.*

Защита конфиденциальных данных от несанкционированного доступа является важнейшим фактором успешного функционирования любой многопользовательской системы. ОС Windows не является исключением и требования к защите объектов файловой системы, памяти, объектов ядра операционной системы внесли существенный вклад в процесс ее проектирования и реализации.

Так, например, версии Windows NT/2000/7/8/10 были сертифицированы по классу C2 критериев TSSEC («Оранжевая книга»). Требования к операционной системе, защищенной по классу C2, включают:

– обязательную идентификацию и аутентификацию всех пользователей операционной системы. Под этим понимается способность операционной системы идентифицировать всех пользователей, которые получают санкционированный доступ к системе, и предоставление доступа к ресурсам только этим пользователям;

– разграничительный контроль доступа – предоставление пользователям возможности защиты принадлежащих им данных;

– системный аудит – способность системы вести подробный аудит всех действий, выполняемых пользователями и самой операционной системой;

– защита объектов от повторного использования – способность системы предотвратить доступ пользователя к информации ресурсов, с которыми до этого работал другой пользователь.

*Идентификация пользователей.*

Для защиты данных Windows использует следующие основные механизмы: аутентификация и авторизация пользователей, аудит событий в системе, шифрование данных, поддержка инфраструктуры открытых ключей, встроенные средства сетевой защиты. Эти механизмы поддерживаются такими подсистемами Windows, как LSASS (Local Security Authority Subsystem Service, подсистема локальной аутентификации), SAM (Security Account Manager, диспетчер локальных записей безопасности), SRM (Security reference Monitor, монитор состояния защиты), Active Directory (служба каталогов), EFS (Encrypting File System, шифрующая файловая система) и др.

Защита объектов и аудит действий с ними в ОС Windows организованы на основе избирательного (дискреционного) доступа, когда права доступа (чтение, запись, удаление, изменение атрибутов) субъекта к объекту задается явно в специальной матрице доступа. Для укрупнения матрицы пользователи могут объединяться в группы. При попытке субъекта (одного из потоков процесса, запущенного от его имени) получить доступ к объекту указывается, какие операции пользователь собирается выполнять с объектом. Если подобный тип доступа разрешен, поток получает описатель (дескриптор) объекта и все потоки процесса могут выполнять операции с ним. Рассмотрим, как в ОС Windows организована аутентификация и авторизация пользователей.

Все действующие в системе объекты (пользователи, группы, локальные компьютеры, домены) идентифицируются в Windows не по именам, уникальность которых не всегда удается достичь, а по **идентификаторам защиты (Security Identifiers, SID)**. SID представляет собой числовое значение переменной длины.

При генерации SID Windows использует генератор случайных чисел, чтобы обеспечить уникальность SID для каждого пользователя. Для некоторого произвольного пользователя SID может выглядеть так:

**S-1-5-21-789336058-484763869-725345543-1003**

Предопределенным пользователям и группам Windows выдает характерные SID, состоящие из SID компьютера или домена и предопределенного RID.

Полный список общеизвестных SID можно посмотреть в документации Platform SDK. Узнать SID конкретного пользователя в системе, а также SID групп, в которые он включен, можно, используя консольную команду **whoami**:

**whoami /user**

Соответствие имени пользователя и его SID можно отследить также в ключе реестра **HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ ProfileList.**

После аутентификации пользователя процессом Winlogon все процессы, запущенные от имени этого пользователя, будут идентифицироваться специальным объектом, называемым **маркером доступа (access token)**. Если процесс пользователя запускает дочерний процесс, то его маркер наследуются.

Маркер доступа содержит идентификатор доступа самого пользователя и всех групп, в которые он включен. В маркер включен также DACL по умолчанию – первоначальный список прав доступа, который присоединяется к создаваемым пользователем объектам. Еще одна важная для определения прав пользователя в системе часть маркера – список его привилегий. Привилегии – это права доверенного объекта на совершение каких-либо действий по отношению ко всей системе. В таблице 6.1 перечислены некоторые привилегии, которые могут быть предоставлены пользователю.

Управление привилегиями пользователей осуществляется в оснастке **«Групповая политика»**, раздел **Конфигурация Windows/Локальные политики/Назначение прав пользователя**.

Таблица 6.1 – Привилегии, которыми могут быть наделены пользователи

| Имя и идентификатор привилегии | Описание привилегии |
|---|---|
| Увеличение приоритета процесса SeIncreaseBasePriorityPrivilege | Пользователь, обладающий данной привилегией, может изменять приоритет процесса с помощью интерфейса Диспетчера задач |
| Закрепление страниц в памяти SeLockMemoryPrivilege | Процесс получает возможность хранить данные в физической памяти, не прибегая к кешированию данных в виртуальной памяти на диске |
| Управление аудитом и журналом безопасности SeAuditPrivilege | Пользователь получает возможность указывать параметры аудита доступа к объекту для отдельных ресурсов, таких как файлы, объекты Active Directory и разделы реестра |
| Овладение файлами или иными объектами SeTakeOwnershipPrivilege | Пользователь получает возможность становиться владельцем любых объектов безопасности системы, включая объекты Active Directory, файлы и папки NTFS, принтеры, разделы реестра, службы, процессы и потоки |
| Завершение работы системы SeShutdownPrivilege | Пользователь получает возможность завершать работу операционной системы на локальном компьютере |
| Обход перекрестной проверки SeChangeNotifyPrivilege | Используется для обхода проверки разрешений для промежуточных каталогов при проходе многоуровневых каталогов |

Чтобы посмотреть привилегии пользователя, можно также использовать команду **whoami /all**.

Остальные параметры маркера носят информационный характер и определяют, например, какая подсистема создала маркер, уникальный идентификатор маркера, время его действия.

Маркер доступа может быть создан не только при первоначальном входе пользователя в систему. Windows предоставляет возможность запуска процессов от имени других пользователей, создавая для этих процессов соответствующий маркер. Для этих целей можно использовать:

– API-функции CreateProcessAsUser, CreateProcessWithLogon;

– оконный интерфейс, инициализирующийся при выборе пункта контекстного меню «Запуск от имени»;

– консольную команду runas:

**runas /user:*имя_пользователя program*** ,

где ***имя_пользователя*** – имя учетной записи пользователя, которая будет использована для запуска программы в формате *пользователь@домен* или *домен\пользователь*;

***program*** – команда или программа, которая будет запущена с помощью учетной записи, указанной в параметре **/user.**

В любом варианте запуска процесса от имени другой учетной записи необходимо задать ее пароль.

*Защита объектов системы.*

Маркер доступа идентифицирует субъектов-пользователей системы. С другой стороны, каждый объект системы, требующий защиты, содержит описание прав доступа к нему пользователей. Для этих целей используется **дескриптор безопасности (Security Descriptor, SD)**. Каждому объекту системы, включая файлы, принтеры, сетевые службы, контейнеры Active Directory и другие, присваивается дескриптор безопасности, который определяет права доступа к объекту и содержит следующие основные атрибуты:

– SID владельца, идентифицирующий учетную запись пользователя-владельца объекта;

– пользовательский список управления доступом (Discretionary Access Control List, DACL), который позволяет отслеживать права и ограничения, установленные владельцем данного объекта. DACL может быть изменен пользователем, который указан как текущий владелец объекта;

– системный список управления доступом (System Access Control List, SACL), определяющий перечень действий над объектом, подлежащих аудиту;

– флаги, задающие атрибуты объекта.

Авторизация Windows основана на сопоставлении маркера доступа субъекта с дескриптором безопасности объекта. Управляя свойствами объекта, администраторы могут устанавливать разрешения, назначать право владения и отслеживать доступ пользователей.

Список управления доступом содержит набор элементов (Access Control Entries, ACE). В DACL каждый ACE состоит из четырех частей: в первой указываются пользователи или группы, к которым относится данная запись, во второй – права доступа, третья информирует о том, предоставляются эти права или отбираются, четвертая представляет собой набор флагов, определяющих, как данная запись будет наследоваться вложенными объектами (актуально, например, для папок файловой системы, разделов реестра).

Если список ACE в DACL пуст, к нему нет доступа ни у одного пользователя (только у владельца на изменение DACL). Если отсутствует сам DACL в SD объекта – полный доступ к нему имеют все пользователи.

Если какой-либо поток запросил доступ к объекту, подсистема SRM осуществляет проверку прав пользователя, запустившего поток, на данный объект, просматривая его список DACL. Проверка осуществляется до появления разрешающих прав **на все** запрошенные операции. Если встретится запрещающее правило хотя бы **на одну** запрошенную операцию, доступ не будет предоставлен.

Стандартные разрешения для файлов:

- Полный доступ (Full Control);
- Изменить (Modify);
- Чтение и выполнение (Read&Execute);
- Чтение (Read);
- Запись (Write).

Стандартные разрешения для папок:

- Полный доступ (Full Control);
- Изменить (Modify);
- Чтение и выполнение (Read&Execute);
- Список содержимого папки;
- Чтение (Read);
- Запись (Write).

Разрешение **Чтение** позволяет просматривать файлы, папки и их атрибуты.

Разрешение **Запись** позволяет создавать новые файлы и папки внутри папок, изменять атрибуты и просматривать владельцев и разрешения.

Разрешение **Список содержимого папки** позволяет просматривать имена файлов и папок.

Разрешение **Чтение и выполнение** для папок позволяет перемещаться по структуре других папок и служит для того, чтобы разрешить пользователю открывать папку, даже если он не имеет прав доступа к ней, для поиска других файлов или вложенных папок. Разрешены все действия, право на которые дают разрешения **Чтение** и **Список содержимого папки**. Это же разрешение для файлов позволяет запускать файлы программ и выполнять действия, право на которые дает разрешение **Чтение**.

Разрешение **Изменить** позволяет удалять папки, файлы и выполнять все действия, право на которые дают разрешения **Запись** и **Чтение** и выполнение.

Разрешение **Полный доступ** позволяет изменять разрешения, менять владельца, удалять файлы и папки и выполнять все действия, на которые дают

право все остальные разрешения NTFS.

Разрешения для папок распространяются на их содержимое: подпапки и файлы.

При определении прав доступа к объектам можно задать правила их наследования в дочерних контейнерах. В окне дополнительных параметров безопасности на вкладке **Разрешения** при выборе опции «**Наследовать от родительского объекта применимых к дочерним объектам разрешения, добавляя их к явно заданным в этом окне»** можно унаследовать разрешения и ограничения, заданные для родительского контейнера, текущему объекту.

При выборе опции **«Заменить разрешения для всех дочерних объектов заданными здесь разрешениями, применимыми к дочерним объектам»** разрешается передача определенных для объекта-контейнера правил доступа его дочерним объектам.

В этом же окне на вкладке **Владелец** можно узнать владельца объекта и заменить его. Владелец объекта имеет право на изменение списка его DACL, даже если к нему запрещен любой тип доступа. Администратор имеет право становиться владельцем любого объекта.

С учетом возможности вхождения пользователя в различные группы и независимости определения прав доступа к объектам для групп и пользователей зачастую бывает сложно определить конечные права пользователя на доступ к объекту: требуется просмотреть запрещающие правила, определенные для самого объекта, для всех групп, в которые он включен, затем то же проделать для разрешающих правил. Автоматизировать процесс определения разрешенных пользователю видов доступа к объекту можно с использованием вкладки **«Действующие разрешения»** окна дополнительных параметров безопасности объекта.

При назначении пользователю или группе разрешения на доступ к файлу или папке руководствуются тем уровнем доступа, который достаточен для данной группы или пользователя при выполнении им своих рабочих обязанностей.

Рассмотренные разрешения относятся к пользователям данного компьютера, совершившим вход локально непосредственно на данную машину. Такие разрешения называются разрешениями NTFS.

Разрешения для пользователей, получившим доступ к папке или файлу через сеть, регулируются отдельно с помощью так называемых разрешений общего доступа. Эти разрешения распространяются только на папки, к которым предоставлен общий доступ через сеть, и действуют только для пользователей, обращающихся к папке через сеть. Возможности пользователя задаются разрешениями:

- Полный доступ (Full Control);
- Изменить (Change);
- Чтение (Read).

Каждому пользователю или группе могут быть установлены множественные разрешения через участие в нескольких группах с разным набором разрешений. В этом случае действуют эффективные разрешения – пользователь обладает всеми назначенными ему разрешениями.

Действует приоритет разрешений для файлов над разрешениями для папок и приоритет запрещения над разрешением.

Разрешения, назначенные родительской папке, по умолчанию наследуются всеми подпапками и файлами, содержащимися в папках. Однако есть возможность предотвратить наследование для любой вложенной папки, и в этом случае эта папка сама становится родительской для вложенных в нее папок.

Если папка предоставлена для общего доступа, то на нее распространяются разрешения двух видов:

– разрешения файловой системы, установленные для пользователей данного компьютера;

– разрешения общего доступа, объявленные для пользователей, получивших доступ через сеть.

Обычно для папок общего доступа задают разрешения полного доступа, а ограничения вводят установкой разрешений NTFS.

В этом случае действует объединение разрешений NTFS и разрешений для общей папки, при котором наиболее строгое разрешение имеет приоритет над другими.

*Команда iCACLS – управление доступом к файлам и папкам*.

Команда **iCACLS** позволяет отображать или изменять списки управления доступом (ACL) для файлов и папок в файловой системе.

Права доступа указываются с помощью сокращений. Рассмотрим разрешения для группы BUILTIN\Администраторы.

(OI) – Object inherit – права наследуются на нижестоящие объекты.

(CI) – Container inherit – наследование каталога.

(F) – Full control – полный доступ к папке.

(I) – Inherit – права наследованы с вышестоящего каталога.

Это означает, что у данной группы есть права на запись, изменение данных в данном каталоге и на изменения NTFS-разрешений. Данные права наследуются на все дочерние объекты в этом каталоге.

Ниже приведен полный список разрешений, которые можно установить с помощью утилиты icacls.

Настройки наследования iCACLS:

(OI) – наследование объекта;

(CI) – наследование контейнера;

(IO) – наследовать только на нижестоящие объекты;

(NP) – не распространять наследование;

(I) – разрешение, унаследованное от родительского контейнера.

Список основных прав доступа:

D – доступ на удаление;

F – полный доступ;

N – нет доступа;

M – изменение;

RX – чтение и выполнение;

R – доступ только для чтения;

W – доступ только для записи.

Используя команду icacls, можно сохранить текущий ACL объекта в файле, а затем применить сохраненный список к тем же или другим объектам (своего рода резервный ACL-путь).

Чтобы выгрузить текущие ACL папки C:\PS и сохранить их в текстовый файл export_ps_acl.txt, выполните команду

icacls C:\PS* /save c:\backup\export_ps_acl.txt /t

Эта команда сохраняет ACL не только о самом каталоге, но и о всех подпапках и файлах.

Полученный текстовый файл можно открыть с помощью блокнота или любого текстового редактора.

Чтобы применить сохраненные списки доступа (восстановить разрешения на каталог и все вложенные объекты), выполните команду

icacls C:\PS /restore :\backup\export_ps_acl.txt

Таким образом, процесс переноса прав доступа с одной папки на другую становится намного легче.

С помощью команды icacls можно изменить списки доступа к папке. Например, чтобы предоставить пользователю aivanov право на редактирование содержимого папки, выполните команду

icacls C:\PS /grant aivanov:M

Удалить все назначенные разрешения для учетной записи пользователя aivanov можно с помощью команды

icacls C:\PS /remove aivanov

Можно запретить пользователю или группе пользователей доступ к файлу или папке

icacls c:\ps /deny "MSKManagers:(CI)(M)"

Важно, что запрещающие правила имеют больший приоритет, чем разрешающие.

С помощью команды icacls можно изменить владельца каталог или папки, например

icacls c:\ps\secret.docx /setowner aivanov /T /C /L /Q

Параметры, которые используются в команде:

/Q – сообщение об успешном выполнении команды не выводится;

/L – команда выполняется непосредственно над символической ссылкой, а не над конкретным объектом;

/C – выполнение команды будет продолжаться, несмотря на файловые ошибки; при этом сообщения об ошибках все равно будут отображаться;

/T – команда используется для всех файлов и каталогов, которые расположены в указанном каталоге.

**Практическое задание**

1 Запустите ОС Windows на виртуальной машине. Войдите в систему под учетной записью администратора.

2 Создайте учетную запись нового пользователя **User1** в оснастке **«Управление компьютером» (compmgmt.msc)**. При создании новой учетной записи запре-

тите пользователю смену пароля и снимите ограничение на срок действия его пароля. Создайте новую группу **Group1** и включите в нее нового пользователя. Удалите пользователя из других групп. Создайте на диске **C:** папку **Lab7**. Создайте или скопируйте в эту папку несколько текстовых файлов (*.txt).

3 С помощью команды **runas** запустите сеанс командной строки (**cmd.exe**) от имени вновь созданного пользователя. Командой **whoami** посмотрите SID пользователя и всех его групп, а также текущие привилегии пользователя. Дайте пояснения информации, содержащейся в полученных SID.

4 Убедитесь в соответствии имени пользователя и полученного SID в реестре Windows (используйте ключ реестра **HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList**).

5 Командой **whoami** определите перечень текущих привилегий пользователя **User1**. Попробуйте изменить системное время командой **time**. Чтобы предоставить пользователю подобную привилегию, запустите оснастку **«Локальные параметры безопасности» (secpol.msc).** Добавьте пользователя в список параметров политики **«Изменение системного времени»** раздела **Локальные политики -> Назначение прав пользователя**. После этого перезапустите сеанс командной строки от имени пользователя, убедитесь, что в списке привилегий добавилась **SeSystemtimePriviege**. Попробуйте изменить системное время командой **time**.

6 Убедитесь, что привилегия **«Завершение работы системы» (SeShutdownPrivilege)** предоставлена пользователю **User1**. После этого попробуйте завершить работу системы из сеанса командной строки пользователя командой **shutdown -s**. Добавьте ему привилегию **«Принудительное удаленное завершение» (SeRemoteShutdownPrivilege)**. Попробуйте завершить работу консольной командой еще раз (отменить команду завершения до ее непосредственного выполнения можно командой **shutdown –a**).

7 Просмотрите разрешения пользователям и группам на папку **c:\ Lab7**. Используйте графический интерфейс.

8 Разрешите пользователю **User1** запись в папку **Lab7**, но запретите запись для группы **Group1**. Попробуйте записать файлы или папки в **Lab7** от имени пользователя **User1**. Объясните результат. Посмотрите эффективные разрешения пользователя **User1** к папке **Lab7** в окне свойств папки.

9 Создайте для папки **Lab7** SACL, позволяющий протоколировать отказы и успехи доступа к этой папке со стороны пользователя **User1** (предварительно проверьте, что в локальной политике безопасности соответствующий тип аудита включен). Убедитесь, что записи аудита были размещены в журнале безопасности (**eventvwr.msc**).

10 Добавьте нового пользователя по имени **User2**. Создайте каталоги «**Public**» и «**Private**». В каждый из этих каталогов скопируйте исполняемый файл (с расширением .exe, .bat или .cmd) и текстовый файл.

11 Изучите справку по команде **icacls** и команде **takeown**.

Напишите командный файл, в котором с помощью команд разграничьте доступ к созданным каталогам и файлам в соответствии со своим вариантом (рисунок 6.1). В отчете приведите код файла и результат его работы.

| Вариант1 | | | |
|---|---|---|---|
| **Субъекты** | **Объекты** | | |
| | Каталог **Public** | Каталог **Private** | Текстовый файл в **«Private»** |
| **Администратор** | Полный доступ | Чтение | Нет доступа |
| **User1** | Чтение | Изменить, кроме удаления | Изменить |
| **User2** | Изменить | Нет доступа | Нет доступа |
| Вариант 2 | | | |
| **Субъекты** | **Объекты** | | |
| | Каталог **Public** | Каталог **Private** | Исполняемый файл в **«Private»** |
| **Администратор** | Полный доступ | Чтение и выполнение | Изменить |
| **User1** | Изменить | Чтение | Выполнение |
| **User2** | Чтение и выполнение | Изменить | Нет доступа |
| Вариант 3 | | | |
| **Субъекты** | **Объекты** | | |
| | Каталог **Public** | Каталог **Private** | **Текстовый файл в «Private»** |
| **Администратор** | Полный доступ | Список содержимого | Нет доступа |
| **User1** | Чтение | Изменить, кроме удаления | Изменить |
| **User2** | Изменить | Нет доступа | Нет доступа |
| Вариант 4 | | | |
| **Субъекты** | **Объекты** | | |
| | Каталог **Public** | Каталог **Private** | **Текстовый файл в «Private»** |
| **Администратор** | Изменить | Чтение и выполнение | Нет доступа |
| **User1** | Чтение | Изменить | Запрет удаления |
| **User2** | Полный доступ, кроме смены владельца | Запись | Нет доступа |
| Вариант 5 | | | |
| **Субъекты** | **Объекты** | | |
| | Каталог **Public** | Каталог **Private** | **Исполняемый файл в «Private»** |
| **Администраторг** | Полный доступ | Список содержимого | Выполнение |
| **User1** | Чтение | Чтение и удаление | Выполнение, запрет удаления |
| **User2** | Изменить, кроме удаления | Запись | Нет доступа |

Рисунок 6.1 – Варианты для разработки командного файла

| Вариант 6 | | | |
|---|---|---|---|
| Субъекты | Объекты | | |
| | Каталог Public | Каталог Private | Исполняемый файл в «Private» |
| Администратор | Полный доступ | Чтение | Изменить |
| User1 | Чтение и удаление | Список содержимого | Выполнение |
| User2 | Изменить | Нет доступа | Нет доступа |
| Вариант 7 | | | |
| Субъекты | Объекты | | |
| | Каталог Public | Каталог Private | Текстовый файл в «Public» |
| Администратор | Список содержимого | Полный доступ | Нет доступа |
| User1 | Изменить, кроме удаления | Чтение | Изменить |
| User2 | Нет доступа | Изменить | Нет доступа |
| Вариант 8 | | | |
| Субъекты | Объекты | | |
| | Каталог Public | Каталог Private | Текстовый файл в «Private» |
| Администратор | Чтение и выполнение | Изменить | Нет доступа |
| User1 | Изменить | Чтение | Изменить, запрет изменения дополнительных атрибутов |
| User2 | Запись | Изменить, кроме удаления | Нет доступа |
| Вариант 9 | | | |
| Субъекты | Объекты | | |
| | Каталог Public | Каталог Private | Исполняемый файл в «Private» |
| Администратор | Список содержимого | Полный доступ | Выполнение |
| User1 | Чтение и удаление | Чтение | Выполнение, запрет удаления |
| User2 | Запись | Полный доступ | Нет доступа |
| Вариант 10 | | | |
| Субъекты | Объекты | | |
| | Каталог Public | Каталог Private | Исполняемый файл в «Private» |
| Администратор | Чтение | Полный доступ | Изменить |
| User1 | Список содержимого | Чтение и удаление | Выполнение |
| User2 | Нет доступа | Изменить | Нет доступа |

Окончание рисунка 6.1

***Вопросы для контроля***

1 Каким образом реализуется защита файлов в NTFS?

2 Перечислите стандартные права доступа к файловым объектам, существующие в файловой системе NTFS.

3 Объясните принцип работы разрешения «Запись».

4 Перечислите элементы разрешений.

5 Кто может стать владельцем объекта?

6 Раскройте понятие наследования разрешений.

7 Как отключить наследование разрешений?

8 Как реализовать принудительное наследование вложенными объектами установленных разрешений?

9 Перечислите приоритеты применения разрешений при определении действующих разрешений на доступ к файловым объектам.

# 7 Лабораторная работа № 7. Оценка рисков информационной безопасности организаций в соответствии с требованиями СТБ 34.101.70–2016

**Цель работы**: получение практических навыков по оценке рисков информационной безопасности в соответствии с требованиями СТБ 34.101.70–2016.

***Порядок выполнения работы***

1 Изучить основные теоретические положения СТБ 34.101.70–2016.

2 Получить задание у преподавателя, выполнить типовые задания.

3 Проанализировать возможные угрозы информационной безопасности, осуществить оценку рисков.

4 Оформить отчет.

**Основные теоретические положения**

Стандарт СТБ 34.101.70–2016 *Информационные технологии. Методы и средства безопасности. Методика оценки рисков информационной безопасности в информационных системах* устанавливает требования по выполнению процедуры оценки рисков информационной безопасности. Он содержит описание процесса оценки рисков информационной безопасности в информационных системах, рекомендации по выбору методов оценки рисков информационной безопасности, пример оценки рисков.

Для оценки рисков информационной безопасности рисков, кроме СТБ 34.101.70–2016, используются специализированные интернет-ресурсы, представленные в таблице 7.1.

Таблица 7.1 – Источники информации по теме работы

| Документ, сайт | Описание |
|---|---|
| СТБ 34.101.70–2016 *Информационные технологии. Методы и средства безопасности. Методика оценки рисков информационной безопасности в информационных системах* | Представлена методика оценки рисков информационной безопасности в информационных системах.<br>В приложении А СТБ 34.101.70–2016 приведен пример отчета оценки рисков информационной безопасности представлен |
| Система для корпоративных служб информационной безопасности. Реестр типов активов (https://service.securitm.ru/assettypes) | Реестр активов, угроз уязвимостей, описаний угроз, причин уязвимостей, источников уязвимостей |
| SECURITM система для корпоративных служб информационной безопасности (https://service.securitm.ru/) | Активы<br>Угрозы<br>Риски<br>Уязвимости<br>Защитные меры от угроз |
| Банк данных угроз безопасности информации (https://bdu.fstec.ru/) | Представлены список угроз, описание угроз, источники угроз, объекты воздействия, последствия реализации угроз, объекты атак |
| Банк данных угроз безопасности информации. Модернизированный раздел угроз (https://bdu.fstec.ru/threat-section) | Перечень угроз безопасности информации |
| Банк данных угроз безопасности информации. Группы мер защиты информации (https://bdu.fstec.ru/threat-section/defenses) | Группы мер защиты информации |
| Банк данных угроз безопасности информации. Список уязвимостей (https://bdu.fstec.ru/vul?size=100) | Список уязвимостей в программном обеспечении |
| Банк данных угроз безопасности информации. Типовые уязвимости веб-приложений (https://bdu.fstec.ru/webvulns) | BDU:W01 – уязвимости, связанные с недостатками проверки вводимых данных.<br>BDU:W02 – уязвимости, связанные с недостатками управления доступом и защиты данных.<br>BDU:W03 – уязвимости, связанные с недостатками работы со структурами данных.<br>BDU:W04 – уязвимости, связанные с недостатками проверки подлинности.<br>BDU:W05 – уязвимости, связанные с недостатками управления ресурсами |

## Практическое задание

1 Осуществить оценку рисков информационной безопасности в информационных системах отдела предприятия или предприятия в целом в соответствии с требованиями СТБ 34.101.70–2016.

2 Оформить отчет в соответствии с примером оценки рисков информационной безопасности, представленном в приложении А СТБ 34.101.70–2016.

***Вопросы для контроля***

1 Что такое риск информационной безопасности и как он может повлиять на организацию?

2 Какие примеры нарушений безопасности данных вы можете привести?

3 Какие меры следует принять, чтобы защитить конфиденциальность информации?

4 Каковы последствия нарушения информационной безопасности для бизнеса и как их можно предотвратить?

# 8 Лабораторная работа № 8. Исследование надежности паролей и их восстановление

**Цель работы**: получение практических навыков по оценке надежности и восстановлению паролей.

***Порядок выполнения работы***

1 Изучить основные теоретические парольной защиты файлов.

2 Изучить основные программные продукты и онлайн-сервисы по восстановлению паролей.

3 Выполнить работу по восстановлению паролей.

4 Оформить отчет.

**Основные теоретические положения**

Пароль – это строка символов, которая используется для проверки подлинности пользователя и предоставления доступа к защищенному ресурсу или информации. Он представляет собой секретную комбинацию символов, которая известна только пользователю, имеющему право доступа.

Пароли широко применяются в компьютерной безопасности и используются в различных областях, включая следующие.

1 Пользовательские учетные записи. Пароли используются для защиты учетных записей пользователей, чтобы предотвратить несанкционированный доступ к личной информации или ресурсам.

2 Управление доступом. Пароли используются для управления доступом к помещениям, зданиям, офисам и другим объектам, чтобы предотвратить несанкционированный доступ.

3 Электронная почта. Пароли используются для защиты электронных почтовых ящиков от несанкционированного доступа и для защиты личной информации, содержащейся в письмах.

4 Онлайн-банкинг. Пароли используются для защиты банковских аккаунтов и предотвращения несанкционированных транзакций.

5 Социальные сети. Пароли используются для защиты профилей в социальных сетях и предотвращения несанкционированного доступа к личной ин-

формации.

6 Файлы и документы. Пароли могут использоваться для защиты файлов и документов, содержащих конфиденциальную информацию.

7 Wi-Fi-сети. Пароли используются для защиты беспроводных сетей Wi-Fi и предотвращения несанкционированного доступа к сети.

Важно использовать надежные пароли и не использовать один и тот же пароль для нескольких учетных записей, чтобы предотвратить несанкционированный доступ и сохранить конфиденциальность личной информации.

Надежность пароля – это показатель эффективности пароля против угадывания или атак методом перебора. В своей обычной форме он оценивает, сколько попыток потребуется злоумышленнику, у которого нет прямого доступа к паролю, в среднем, чтобы правильно его угадать. Надежность пароля зависит от длины, сложности и непредсказуемости.

Использование надежных паролей снижает общий риск нарушения безопасности, но надежные пароли не заменяют необходимость в других эффективных средствах контроля безопасности.

Скорость, с которой злоумышленник может передавать системе угаданные пароли, является ключевым фактором в определении безопасности системы. Некоторые системы устанавливают тайм-аут в несколько секунд после небольшого числа (например, трех) неудачных попыток ввода пароля. При отсутствии других уязвимостей такие системы могут быть эффективно защищены относительно простыми паролями. Однако система должна хранить информацию о паролях пользователя в той или иной форме, и если эта информация будет украдена, скажем, путем нарушения безопасности системы, пароли пользователя могут оказаться под угрозой.

Существуют требования к надежности паролей. Пароль должен быть таким, чтобы его нельзя было легко раскрыть. Для этого при выборе и использовании пароля рекомендуется руководствоваться следующими правилами:

1) пароль не должен содержать личных данных пользователя (таких как фамилия, имя, серия или номер паспорта либо другого документа, удостоверяющего личность, дата рождения, адрес и т. п.);

2) пароль не должен быть словом из какого-либо словаря (входить в какой-либо тезаурус), т. к. перебор слов заданного словаря – технически достаточно простая задача;

3) пароль не должен быть слишком коротким (подобрать сочетание символов в этом случае также не представляет сложности);

4) пароль не должен состоять из повторяющихся букв или фрагментов текста;

5) пароль не должен состоять из символов, соответствующих подряд идущим клавишам на клавиатуре (например, «QWERTY» – образец недопустимого пароля);

6) желательно включать в пароль символы в разных регистрах (прописные и строчные буквы, кириллицу и латиницу), знаки препинания, цифры и др.

Следующие меры предосторожности помогут гарантировать безопасность при использовании пароля:

– используйте надежные пароли. Пароли должны быть достаточно длин-

ными и сложными, чтобы их было трудно угадать или взломать. Используйте сочетание букв, цифр и специальных символов;

– не используйте один и тот же пароль для всех аккаунтов. Если пароль взломается, злоумышленники смогут получить доступ ко всем вашим аккаунтам. Используйте уникальные пароли для каждого аккаунта;

– не сообщайте пароли другим людям. Никогда не сообщайте свои пароли другим людям, даже друзьям или близким. Это может привести к несанкционированному доступу к вашим личным данным;

– не используйте общедоступные компьютеры для ввода паролей: избегайте ввода паролей на общедоступных компьютерах, таких как в интернет-кафе или библиотеках. Кто-то может установить программное обеспечение для перехвата паролей;

– используйте двухфакторную аутентификацию. Это обеспечивает дополнительный уровень безопасности для ваших аккаунтов. Помимо пароля, вам потребуется ввести дополнительный код, который будет отправлен на ваш мобильный телефон;

– регулярно меняйте пароли. Регулярно меняйте свои пароли, чтобы повысить безопасность своих аккаунтов. Меняйте пароли, если считаете, что они могут быть скомпрометированы;

– устанавливайте пароли большой длины. Длинные пароли тяжело подбирать. Установка длинных паролей, как показывает практика, даже более эффективны чем постоянная смена паролей;

– следите за безопасностью своих паролей. Используйте надежные менеджеры паролей для хранения ваших паролей и следите за безопасностью своих аккаунтов.

Существует несколько способов оценки надежности пароля. Вот несколько из них:

– длина пароля. Чем длиннее пароль, тем более безопасным он является. Используйте пароли, состоящие как минимум из восьми символов;

– сложность пароля. Используйте сочетание букв, цифр и специальных символов, чтобы сделать пароль более сложным для угадывания или взлома. Избегайте использования простых слов и фраз, которые могут быть легко угаданы;

– использование менеджера паролей. Многие менеджеры паролей могут оценить надежность пароля и рекомендовать более надежные пароли;

– использование методов восстановления паролей. Можно использовать программы для восстановления паролей или словарные атаки, чтобы узнать, насколько легко пароль может быть взломан;

– использование онлайн-сервисов. Некоторые онлайн-сервисы могут оценить надежность пароля, например, проверка пароля на сайте «How Secure Is My Password?».

Для проверки надежностей паролей более детально рассмотрим онлайн-сервисы, которые могут помочь оценить надежность пароля. Вот несколько популярных сервисов.

1 «Kaspersky Password Check» – сервис проверяет надежность пароля на основе различных факторов, таких как длина, сложность и использование разных типов символов (https://password.kaspersky.com/ru/).

2 «How Secure Is My Password?» – сервис оценивает сложность пароля и сообщает, насколько сложно его будет взломать. Сайт также предлагает рекомендации по улучшению пароля (https://howsecureismypassword.net).

3 «How Secure is Your Password?» – сервис проверяет пароль методом перебора и наличия пароля в базах стандартных паролей (https://www.passwordmonster.com/).

4 «The Password Meter» – сервис проверяет надежность пароля на основе длины, сложности и разнообразия символов. Он также предоставляет рекомендации по улучшению пароля (http://www.passwordmeter.com).

5 «How Secure Is My Password?» – сервис проверки пароля методом перебора (https://www.security.org/how-secure-is-my-password/).

6 «Have I Been Pwned» – сервис проверяющий пароль по базе украденных паролей в интернете (https://haveibeenpwned.com/Passwords).

При работе на компьютере пользователь использует пароли для входа в операционную систему, для работы с электронной почтой, интернет-мессенджерами, документами, архивами. Почти на все файлы можно установить пароли, а также при необходимости их восстановить, если пароль забыт. Программы по восстановлению паролей находятся в Яндексе или Гугле по следующим ключевым словам «RAR password recovery», «ZIP password recovery», «Word password recovery», «Excel password recovery», «Office password recovery» или «PDF password recovery» и так далее по аналогии.

Для восстановления забытых паролей существуют специализированные программные продукты. Для каждого типа файлов существуют свои программы.

Для восстановления паролей в архивах RAR: RAR Password Unlocker, Free RAR Password Recovery, Free KRyLack RAR Password Recovery, RAR Password Cracker, Accent RAR Password Recovery, KRyLack RAR Password Recovery, Password Recovery Bundle, Advanced Archive Password Recovery Professional, Advanced RAR Password Recovery, RAR Password Unlocker, Any RAR Password Recovery, PassFab for RAR и др.

Для восстановления паролей в архивах Zip: Zip Password Cracker Pro, KRyLack ZIP Password Recovery, Ultimate ZIP Cracker, Zip Password Recovery Tool, Passware Kit, Accent ZIP Password Recovery, Zip Password Recovery Master, Elcomsoft Distributed Password Recovery и др.

Для восстановления паролей в документах Office используются: Free Word and Excel Password Recovery Wizard, Appnimi All-In-One Password Unlocker, PassFab for Office, GuaWord, Elcomsoft Advanced Office Password Recovery, Advanced Office Password Recovery Pro, Appnimi Word Password Recovery, Elcomsoft Advanced PDF Password Recovery, Password Recovery Bundle и др.

Для каждого типа файла существуют свои программы, а также имеются специализированные онлайн-сервисы для восстановления забытых паролей, например:

LostMyPass.com – сервис восстанавливающий пароли в файлах архивов zip, rar, 7z, файлах word, excel, powerpoint и pdf (https://www.lostmypass.com/ru/try/);

Password-online – это еще один онлайн-инструмент для восстановления паролей doc, docx, xls, xlsx, ppt, mdb, pdf, rar, zip, 7zip, eoc и т.д. В настоящее время восстановление паролей стало платным (https://www.password-online.com);

Online Password Remover – сервис удаления паролей с документов Excel, Word и PowerPoint (https://www.password-find.com);

PassFab – сервис с информацией по восстановлению паролей с любых документов (https://www.passfab.ru).

Кроме данных сервисов, в интернете есть очень много аналогичных.

**Практическое задание**

1 Изучите теоретический материал.

2 Придумайте пароль.

3 В поисковой системе Яндекс или Гугл введите «Список самых распространенных паролей» или «Стандартные пароли» и проверьте, есть ли ваш пароль в данном списке.

4 Используя онлайн сервис проверки надежности пароля, проверьте ваш пароль.

5 Создайте документ Word, Excel и архив Zip и установите простой пароль из не более трех символов.

6 Восстановите пароль ваших файлов, используя специализированные онлайн-сервисы восстановления паролей или специализированного программного обеспечения при возможности установить его на ваш компьютер.

***Вопросы для контроля***

1 Что такое пароль?

2 Перечислите минимальные требования к выбору пароля.

3 Как определить вероятность подбора пароля злоумышленником в течение срока его действия?

4 Выбором каких параметров можно повлиять на уменьшение вероятности подбора пароля злоумышленником?

5 Какие онлайн-сервисы по проверке надежностей паролей существуют?

# Список литературы

1 Конституция Республики Беларусь: с изм. и доп., принятыми на респ. референдумах 24 нояб. 1996 г., 17 окт. 2004 г. и 27 фев. 2022 г. – Минск: Нац. центр правовой информ. Респ. Беларусь, 2022. – 80 с.

2 Концепция национальной безопасности Респ. Беларусь [Электронный ресурс] / Нац. реестр правовых актов Респ. Беларусь, 1/9403, 2008.

3 Об информации, информатизации и защите информации: Закон Респ. Беларусь [Электронный ресурс] / Нац. реестр правовых актов Респ. Беларусь. – Минск, 2014.

4 О государственных секретах: Закон Респ. Беларусь [Электронный ресурс] / Нац. реестр правовых актов Респ. Беларусь. – Минск, 2016.

5 ТР 2013/027/BY. Информационные технологии. Средства защиты информации. Информационная безопасность. – Минск: Госстандарт, 2013. – 9с.

6 О защите персональных данных: Закон Респ. Беларусь [Электронный ресурс] / Нац. реестр правовых актов Респ. Беларусь. – Минск, 2021.

7 Перечень стандартов и рекомендаций в области информационной безопасности, применяемых в рамках реализации цифровой повестки Евразийского экономического союза: рекомендация Коллегии ЕЭК [Электронный ресурс] / Нац. реестр правовых актов Респ. Беларусь. – Минск, 2019.

8 Операционные системы. Основы UNIX: учебное пособие / А. Б. Вавренюк [и др.]. – Москва : ИНФРА-М, 2018. – 160 с.

9 Защита информации [Электронный ресурс]: учебное пособие / А. П. Жук [и др.]. – 3-е изд. – Москва: РИОР; ИНФРА-М, 2021. – 400 с. – Режим доступа: http://www.znanium.com. – Дата доступа: 05.03.2022.

10 **Ананченко, И. В.** Средства резервного копирования, восстановления, защиты данных в операционных системах Windows / И. В. Ананченко, Т. В. Зудилова, С. Э. Хоружников. – Санкт-Петербург: Университет ИТМО, 2019. – 50 с.

11 **Аверченков, В. И.** Информационная безопасность сетей и систем: учебное пособие / В. И. Аверченков, В. Т. Еременко, Е. А. Зайченко. – Могилев: Белорус.-Рос. ун-т, 2020. – 212 с.