МИНОБРНАУКИ РОССИИ

Орский гуманитарно-технологический институт (филиал)
федерального государственного бюджетного образовательного учреждения
высшего образования «Оренбургский государственный университет»
(Орский гуманитарно-технологический институт (филиал) ОГУ)

Кафедра программного обеспечения

Методические указания по выполнению и защите лабораторных и практических работ по дисциплине «Б1.Д.В.19 Безопасность информационных технологий»

Уровень высшего образования

БАКАЛАВРИАТ

Направление подготовки

09.03.01 Информатика и вычислительная техника

(код и наименование направления подготовки)

Программное обеспечение средств вычислительной техники и автоматизированных систем

(наименование направленности (профиля) образовательной программы)

Тип образовательной программы

Программа бакалавриата

Квалификация

Бакалавр

Форма обучения

Очная

Год начала реализации программы (набора)

2019

г. Орск 2018

Методические указания предназначены для обучающихся очной формы обучения направления подготовки 09.03.01 Информатика и вычислительная техника профилю Программное обеспечение средств вычислительной техники и автоматизированных систем по дисциплине «Б1.Д.В.19 Безопасность информационных технологий»

Составитель _____ О.В. Подсобляева

Методические указания рассмотрены и одобрены на заседании кафедры программного обеспечения, протокол № 1 от «01» сентября 2018 г.

Заведующий кафедрой _____ Е.Е. Сурина

Согласовано:
Председатель методической комиссии по направлению подготовки 09.03.01 Информатика и вычислительная техника

_____ Е.Е.Сурина
«12» сентября 2018 г.

Пояснительная записка

В результате изучения дисциплины «Б1.Д.В.19 Безопасность информационных технологий» у обучающихся должны быть сформированы знания, умения и навыки:

- изучение программно-аппаратных средств защиты информации, методов анализа и планирования информационной защиты компьютерных систем, сетей и их компонентов, средств защиты сетевых служб.

- формирование базовых знаний в области информационной защиты телекоммуникационных и компьютерных систем и сетей на основе современных программных и операционных систем.

Одной из наиболее эффективных форм закрепления теоретических знаний и выработки навыков самостоятельной работы являются лабораторные занятия.

Целью проведения лабораторных занятий является:

- закрепление знаний студентов по основам проектной деятельности,

- формирование у студентов навыков использования современных технических средств и технологий для решения проектных и исследовательских задач.

Тематический план

Таблица 1 – Тематический план выполнения лабораторных и практических работ по дисциплине «Б1.Д.В.19 Безопасность информационных технологий» для обучающихся направления подготовки 09.03.01 Информатика и вычислительная техника профиль подготовки Программное обеспечение средств вычислительной техники и автоматизированных систем

Лабораторные работы

№ раздела	Наименование лабораторных работ	Кол-во часов
5	Криптопровайдеры и шифрующие файловые системы	1
5	Изучение системы сертификатов и цифровой подписи	1
6	Управление правами доступа пользователей/групп к информационным ресурсам	2
6	Изучение программных средств сканирования сетей и обнаружения атак	2
7	Методы защиты удаленного доступа к ресурсам компьютерной сети. Безопасная работа в интернете	2
7	Изучение антивирусных программных комплексов	2
	Итого:	12

Практические работы

№ раздела	Наименование практических работ	Кол-во часов
3	Криптопровайдеры и шифрующие файловые системы	2
4	Изучение системы сертификатов и цифровой подписи	2
5	Управление правами доступа пользователей/групп к информационным ресурсам	2
6	Изучение программных средств сканирования сетей и обнаружения атак	2
7	Методы защиты удаленного доступа к ресурсам компьютерной сети. Безопасная работа в интернете	2
7	Изучение антивирусных программных комплексов	2
	Итого:	12

Методические указания по выполнению и оформлению лабораторных работ

Лабораторные и практические работы по дисциплине «Безопасность информационных технологий» предполагают решение задач по темам, представленным в тематическом плане.

В практической и лабораторной работе должны быть выполнены все предусмотренные задания. В работе должна просматриваться логическая последовательность и взаимная увязка основных частей работы.

Рекомендуемая структура работ:

1) цель работы;

2) задание в соответствии с выбранным вариантом;

3) теоретическая часть, включающая краткое изложение теоретических положений по теме практической работы, формулы для решения задания;

4) практическая часть, включающая решение задания по теме практической работы. Дополнительно для наглядности расчетный материал может быть представлен в виде таблиц, графиков;

5) выводы по работе;

6) список использованной литературы.

Работы могут быть оформлены:

- машинописным текстом на листах формата А4.

Титульный лист оформляется на основе СТО 02069024. 101 – 2014 «РАБОТЫ СТУДЕНЧЕСКИЕ. Общие требования и правила оформления».

Работа защищается устно и принимается к зачету, если нет замечаний по ее выполнению и оформлению. При отсутствии зачтенных лабораторных работ студент не допускается к зачету по дисциплине «Б1.Д.В.Безопасность информационных технологий».

Лабораторная работа №1
Криптопровайдеры и шифрующие файловые системы

Cryptoloop позволяет создать шифрованные файловые системы на разделах жесткого диска или в файле. Для шифрования данных Cryptoloop использует «петлевое устройство» – специальное псевдоустройство, через которое проходят системные вызовы к файловой системе, и осуществляется вся обработка информации. Начиная с ядра ОС Linux версии 2.6, программный интерфейс CryptoAPI был встроен в ядро системы, в результате чего отпала необходимость в установлении дополнительных патчей к ядру, и создание шифрованных файловых систем стало намного проще.

Для использования Cryptoloop необходимо настроить ядро ОС и установить пользовательское ПО. В лабораториях учебного комплекса «Технологии безопасности» рабочие станции уже подготовлены для использования Cryptoloop.

Для шифрования данных Cryptoloop использует криптографические алгоритмы, предоставляемые ядром ОС. Список доступных методов шифрования можно посмотреть в файле /proc/crypto. Собранные в виде модулей ядра алгоритмы находятся в папке /lib/modules/<kernel_version>/kernel/crypto, где <kernel_version> – версия ядра, например 2.6.5.

Настройка петлевого устройства производится с помощью утилиты losetup. Она связывает файл или другое устройство (например, /dev/hdb1) с петлевым устройством /dev/loopX, где X = 0,1,2.... Также в качестве параметра к losetup указывается метод шифрования. Все записываемые в петлевое устройство данные подвергаются шифрованию и записываются в связанный с ним файл или другое устройство. Аналогичная ситуация наблюдается при чтении из петлевого устройства. При этом сначала производится чтение из связанного файла, и затем дешифрование этой информации.

Далее приводятся два примера по созданию шифрованной файловой системы на разделе жесткого диска и в файле.

1.1.1 Создание шифрованной файловой системы на разделе жесткого диска

Данный пример описывает создание шифрованной файловой системы на устройстве /dev/hdb1 – первый раздел slave-диска на первом IDE-контроллере.

1. Перед созданием шифрованной файловой системы рекомендуется отформатировать раздел и заполнить его случайными данными, что намного затруднит процесс обнаружения структурных элементов файловой системе на разделе. Для заполнения раздела случайными данными необходимо выполнить:

dd if=/dev/urandom of=/dev/hdb1 bs=1M

С помощью этой команды на раздел /dev/hdb1 копируются блоки данных размером 1 Мб с устройства /dev/urandom, генерирующего последовательность случайных байт информации.

2. Выбор алгоритма шифрования данных. В данном примере используется AES.

3. Настройка петлевого устройства. С помощью утилиты losetup петлевое устройство /dev/loop0 связывается с разделом жесткого диска /dev/hdb1 с использованием шифрования AES:

losetup –e aes /dev/loop0 /dev/hdb1

Далее вводится пароль, на основе которого будет сформирован ключ для алгоритма шифрования.

4. Создание файловой системы. Допускается использование различных типов файловых систем, в данном примере – ext2. С помощью утилиты mkfs файловая система ext2 создается на петлевом устройстве /dev/loop0:

mkfs. ext2 /dev/loop0

5. Монтирование шифрованной файловой системы. Для начала необходимо создать точку монтирования, например /mnt/crypto. Далее с помощью утилиты mount производится монтирование файловой системы с петлевого устройства /dev/loop0 в каталог /mnt/crypto:

mount –t ext2 /dev/loop0 /mnt/crypto

где ext2 – тип файловой системы (данный параметр можно не указывать).

6. Шифрованная файловая система создана и готова для использования.

7. Размонтирование файловой системы:

umount /mnt/crypto

8. Отключение петлевого устройства

losetup –d /dev/loop0

1.1.2 Создание шифрованной файловой системы в файле

Шифрованная файловая система в файле создается аналогично. Единственным отличием является указание в параметрах команд имени файла вместо устройства.

1. Создание файла и заполнение его случайными данными. С помощью утилиты dd создается файл cryptofs. dat размером 100 Мб (100 блоков по 1 Мб) в домашнем каталоге пользователя и заполняется случайными данными:

dd if=/dev/urandom of=~/cryptofs. dat bs=1M count=100

2. Настройка петлевого устройства

losetup –e aes /dev/loop0 ~/cryptofs. dat

3. Далее по предыдущему примеру:

mkfs. ext2 /dev/loop0

mount –t ext2 /dev/loop0 /mnt/crypto

1.1.3 Использование шифрованной файловой системы:

Смонтировать шифрованную файловую систему можно с помощью утилиты mount с указанием файла или устройства, точки монтирования и метода шифрования:

mount /dev/hdb1 /mnt/crypto –oencryption=aes

mount ~/cryptofs. dat /mnt/crypto –oencryption=aes

1.2 Шифрованный сетевой канал PPP-SSH

С документацией по созданию шифрованного канала PPP-SSH можно ознакомиться под ОС Linux (/usr/doc/Linux-HOWTOs/ppp-ssh) или в Интернете [2].

Практическая работа №1
Криптопровайдеры и шифрующие файловые системы

Рабочее задание

1.1 Шифрованная файловая система Cryptoloop

1. Следуя инструкциям в документации, и используя справочное руководство [3], создать шифрованную файловую систему в произвольном файле.
2. Переписать в созданную ФС несколько текстовых файлов.
3. Просмотреть содержимое файла, содержащего ФС. Сделать выводы.

1.2 Шифрованный сетевой канал PPP-SSH

1. Ознакомиться с документацией по созданию шифрованного канала PPP-SSH (из ОС Linux в /usr/doc/Linux-HOWTOs/ppp-ssh или в Интернете [2]).
2. Переписать текстовой произвольный файл с одной рабочей станции на другую и с помощью сниффера tcpdump в момент копирования запротоколировать передаваемые по сети данные. Также можно запротоколировать сетевую активность FTP или telnet сессии.
3. Следуя инструкциям в документации, и используя manual pages [3], настроить SSH и создать шифрованный канал между двумя компьютерами (объединившись с соседом).
4. Повторить действия, описанные в п.2, но уже через созданный сетевой канал. Сравнить полученные результаты (дамп сетевого трафика) и сделать выводы.

Лабораторная работа №2
Изучение системы сертификатов и цифровой подписи

Цель работы: Исследование структуры алгоритма и методики практической реализации (ЭЦП) RSA.

Основные теоретические положения: Технология применения системы ЭЦП предполагает наличие сети абонентов, обменивающихся подписанными электронными документами. При обмене электронными документами по сети значительно снижаются затраты, связанные с их обработкой, хранением и поиском.

Одновременно при этом возникает проблема, как аутентификации автора электронного документа, так и самого документа, т.е. установление подлинности автора и отсутствия изменений в полученном электронном сообщении.

В алгоритмах ЭЦП как и в асимметричных системах шифрования используются однонаправленные функции. ЭЦП используется для аутентификации текстов, передаваемых по телекоммуникационным каналам.

ЭЦП представляет собой относительно небольшой объём дополнительной цифровой информации, передаваемой вместе с подписанным текстом.

Концепция формирования ЭЦП основана на обратимости асимметричных шифров, а также на взаимосвязанности содержимого сообщения, самой подписи и пары ключей. Изменение хотя бы одного из этих элементов сделает невозможным подтверждение подлинности подписи, которая реализуется при помощи асимметричных алгоритмов шифрования и хэш-функций.

Система ЭЦП включает две процедуры:

- формирование цифровой подписи;
- проверку цифровой подписи.

В процедуре формирования подписи используется секретный ключ отправителя сообщения, в процедуре проверки подписи — открытый ключ отправителя.

Безопасность системы RSA определяется вычислительной трудностью разложения на множители больших целых чисел. Недостатком алгоритма цифровой подписи RSA является уязвимость её к мультипликативной атаке. Другими словами, алгоритм ЭЦП RSA позволяет хакеру без знания секретного ключа сформировать подписи под теми документами, в которых результат - хэширования можно вычислить как произведение результата хэширования уже подписанных документов.

Практическая работа №2
Изучение системы сертификатов и цифровой подписи
Алгоритм электронной цифровой подписи (ЭЦП) RSA

1.Определение открытого «e» и секретного «d» ключей (действия отправителя)

1.1. Выбор двух взаимно простых больших чисел ***p*** и ***q***

1.2. Определение их произведения ***n =p*q***

1.3. Определение функции Эйлера: ***ϕ(n) = (p-1)(q-1)***

1.4. Выбор секретного ключа ***d*** с учетом условий: ***1 < d < ϕ(n),***

$$НОД(n, \phi(n)) = 1$$

1.5. Определение значения открытого ключа ***e: e <n***,

$$e*d(mod\ \phi(n))=1$$

2. Формирование ЭЦП

2.1. Вычисление хэш - значения сообщения ***M: m = h (M)***

2.2. Для получения ЭЦП шифруем хэш – значение ***m*** с помощью секретного ключа ***d*** и отправляем получателю цифровую подпись $S = m^d \ (mod\ n)$ и открытый текст сообщения ***M***

3.Аутентификация сообщения - проверка подлинности подписи

3.1. Расшифровка цифровой подписи ***S*** с помощью открытого ключа e и вычисление её хэш - значения $m' = S^e (mod\ n)$

3.2. Вычисление хэш - значения принятого открытого текста ***M*** и ***m=h (M)***

3.3. Сравнение хэш - значений ***m и m'***, если ***m=m'***, то цифровая подпись **S** — достоверна.

Процедуру формирования ЭЦП сообщения M рассмотрим на следующем простом примере:

3.4. Вычисление хэш - значения сообщения ***M: m= h(M).***

Хэшируемое сообщение M представим как последовательность целых чисел

3.5. В соответствии с приведённым выше алгоритмом формирования ЭЦП RSA выбираем два взаимно простых числа ***p = 3, q = 11***, вычисляем значение ***n=p*q=3*11=33***, выбираем значение секретного ключа ***d = 7*** и вычисляем значение открытого ключа ***e = 3***. Вектор инициализации H_0 выбираем равным ***6*** (выбирается случайным образом).

Хэш - код сообщения **M =312** формируется следующим образом:

$H_1 = (M_1 + H_0)^2 \ (mod\ n) = (3+6)^2 \ (mod\ 33) = 81 \ (mod 33) = 15;$
$H_2 = (M_2 + H_1)^2 \ (mod\ n) = (1+15)^2 \ (mod\ 33) = 256 \ (mod\ 33) = 25;$
$H_3 = (M_3 + H_2)^2 \ (mod\ n) = (2+25)^2 \ (mod 33) = 729 \ (mod 33) = 3, m=3$

3.6. Для получения ЭЦП шифруем хэш - значение ***m*** с помощью секретного ключа ***d*** и отправляем получателю цифровую подпись

$S = m^d \ (mod\ n)$ ***и открытый текст сообщения M***
$S = 3^7 \ (mod\ 33) = 2187 (mod 33) = 9$

3.7. Проверка подлинности ЭЦП

Расшифровка ***S*** (т. е. вычисление её хэш - значения $\mathbf{m}^{/}$) производится с помощью открытого ключа **e**.

$$m' = S^e \ (mod\ n) = 9 \ (mod\ 33) = 729 (mod\ 33) = 3$$

3.8. Если сравнение хэш - значений ***m'***и ***m*** показывает их равенство, т.е. ***m =m'***, то подпись достоверна.

4.Содержание отчета

4.1. Составить блок-схему алгоритма и программу формирования ЭЦП RSA.

4.2. Листинг программы расчета ЭЦП RSA в соответствии с заданием

Лабораторная работа №3
Управление правами доступа пользователей/групп к информационным ресурсам

Цель: Изучение механизмов управления доступа к ресурсам, прав доступа. Постижение понятия пользователя и группы. Приобретение практических навыков управления пользователями при помощи консольных утилит. Приобретение навыков работы с правами пользователей и правами на файлы, каталоги при помощи консольных утилит.

Теоретическая часть

У каждого объекта (файла) есть уникальное имя, по которому к нему можно обращаться, и конечный набор операций, которые процессы могут выполнять в отношении этого объекта. Файлу свойственны операции read, write и execute.

Совершенно очевидно, что нужен способ запрещения процессам доступа к тем объектам, к которым у них нет прав доступа. Более того, этот механизм должен также предоставлять возможность при необходимости ограничивать процессы поднабором разрешенных операций. Например, процессу A может быть дано право проводить чтение данных из файла F, но не разрешено вести запись в этот файл.

Права доступа означают разрешение на выполнение той или иной операции (чтение, записи, исполнения).

Когда пользователь входит в систему, его оболочка получает UID и GID (UID – идентификатор пользователя, GID - идентификатор группы), которые содержатся в его записи в файле паролей, и они 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).

Два процесса с одинаковой комбинацией (UID, GID) будут иметь абсолютно одинаковый доступ к одинаковому набору объектов. Процессы с различающимися значениями (UID, GID) будут иметь доступ к разным наборам файлов, хотя, может быть, и со значительным перекрытием этих наборов.

SUID (Set User ID)

Атрибут исполняемого файла, позволяющий запустить его с правами владельца. В операционных системах Linux приложение запускается с правами пользователя, запустившего указанное приложение. Это обеспечивает дополнительную безопасность т.к. процесс с правами пользователя не сможет получить доступ на запись к важным системным файлам, например /etc/passwd, который принадлежит суперпользователю root. Если на исполняемый файл установлен бит suid, то при выполнении эта программа автоматически меняет «эффективный userID» на идентификатор того пользователя, который является владельцем этого файла. То есть, не зависимо от того - кто запускает эту программу, она при выполнении имеет права хозяина этого файла.

SGID (Set Group ID)

Аналогичен SUID, но относиться к группе. При этом, если для каталога установлен бит SGID, то создаваемые в нем объекты будут получать группу владельца каталога, а не пользователя.

Практические примеры

Узнать права на файл/директорию

sit@ubuntu:~$ ls -l /bin/ls

-rwxr-xr-x 1 root root 129280 Feb 18 2016 /bin/ls

Права доступа состоят из трех троек символов. Первая тройка представляет права владельца файла, вторая представляет права группы файла и третья права всех остальных пользователей.

В нашем случае это :

- «rwx» - Права владельца файла

- «r-x» - Права группы файла
- «r-x» - Права всех остальных на файл.

Символ «r» означает, что чтение (просмотр данных содержащихся в файле) разрешено, «w» означает запись (изменение, а также удаление данных) разрешено и «x» означает исполнение (запуск программы разрешен).

Таким образом, если в целом посмотреть на права мы увидим, что кому угодно разрешено читать содержимое и исполнять этот файл, но только владельцу (root) разрешено как либо модифицировать этот файл. Иными словами, нормальным пользователям разрешено копировать содержимое этого файла, то только root может изменять или удалять его.

Определение текущего пользователя и групп в которых он состоит

Перед тем, как изменять владельца или группу которой принадлежит файл, необходимо уметь определять текущего пользователя и группу к которой он принадлежит. Чтобы узнать под каким пользователем вы работаете, наберите whoami:

sit@ubuntu:~$ whoami

sit

Для определения в каких группах состоит пользователь sit, необходимо воспользоваться командой groups:

sit@ubuntu:~$ groups

sit adm cdrom sudo dip plugdev lxd lpadmin sambashare

Из этого примера видно, что пользователь sit состоит в группах sit, adm, cdrom, sudo, dip, plugdev, lxd, lpadmin, sambashare. Если вы хотите посмотреть, в каких группах состоит другой пользователь, то передайте его имя в качестве аргумента.

sit@ubuntu:~$ groups root

root : root

Изменение пользователя и группы владельца

Чтобы изменить владельца или группу файла (или другого объекта) используется команды chown или chgrp соответственно. Сначала нужно передать имя группы или владельца, а потом список файлов.

chown sit /home/sit/itmo.txt

chgrp sit /home/sit/itmo.txt

Можно также изменять пользователя и группу одновременно используя команду chown в другой форме:

chown sit:sit /home/sit/itmo.txt

Предупреждение

Вы не можете использовать команду chown без прав суперпользователя, но chgrp может быть использована всеми, чтобы изменить группу-владельца файла на ту группу, к которой они принадлежат.

Знакомство с chmod

chown и chgrp используются для изменения владельца и группы объекта файловой системы, но кроме них существует и другая программа, называемая chmod, которая используется для изменения прав доступа на чтение, запись и исполнение, которые мы видим в выводе команды ls -l. Команда chmod использует два и более аргументов: метод, описывающий как именно необходимо изменить права доступа с последующим именем файла или списком файлов, к которым необходимо применить эти изменения:

chmod +x /home/sit/itmo.sh

В примере выше в качестве метода указано +x. Как можно догадаться, метод +x указывает chmod, что файл необходимо сделать исполняемым для пользователя, группы и для всех остальных. Если мы решим отнять все права на исполнение файла, то сделаем вот так:

chmod +x /home/sit/itmo.sh

Разделение между пользователем, группой и всеми остальными

Часто бывает удобно изменить только один или два набора прав доступа за раз. Чтобы сделать это, просто необходимо использовать специальный символ для обозначения набора прав

доступа, который необходимо изменить, со знаком «+»» или «—» перед ним. Символ «u» для пользователя, «g» для группы и «o» для остальных пользователей.

chmod go-w /home/sit/itmo.sh

Данный пример удаляет право на запись для группы и всех остальных пользователей, но оставляет права владельца нетронутыми.

Числовые режимы

Существует еще один достаточно распространенный способ указания прав: использование четырехзначных восьмеричных чисел. Этот синтаксис, называется числовым синтаксисом прав доступа, где каждая цифра представляет тройку разрешений. Например, в 0777, 777 устанавливают флаги для владельца, группы, и остальных пользователей. Ниже таблица показывающая как транслируются права доступа на числовые значения.

Режим	Число
rwx	7
rw-	6
r-x	5
r--	4
-wx	3
-w-	2
--x	1
---	0

umask

Когда процесс создает новый файл, он указывает, какие права доступа нужно задать для данного файла. Зачастую запрашиваются права 0666 (чтение и запись всеми), что дает больше разрешений, чем необходимо в большинстве случаев. К счастью, каждый раз, когда в Linux создается новый файл, система обращается к параметру, называемому umask. Система использует значение umask чтобы понизить изначально задаваемые разрешения на что-то более разумное и безопасное. Вы можете просмотреть текущие настройки umask набрав umask в командной строке:

sit@ubuntu:~$ umask

0002

В Linux-системах значением по умолчанию для umask является 0022, что позволяет другим читать ваши новые файлы (если они могут до них добраться), но не изменять их. Чтобы автоматически обеспечивать больший уровень защищенности для создаваемых файлов, можно изменить настройки umask:

sit@ubuntu:~$ umask 0077

Такое значение umask приведет к тому, что группа и прочие не будут иметь совершенно никаких прав доступа для всех, вновь созданных файлов.

В отличие от «обычного» назначения прав доступа к файлу, umask задает какие права доступа **должны быть отключены**. Снова посмотрим на таблицу соответствия значений чисел и методов:

Режим	Число
rwx	7
rw-	6
r-x	5
r--	4
-wx	3
-w-	2
--x	1
---	0

Воспользовавшись этой таблицей мы видим, что последние три знака в 0077 обозначают — rwxrwx. umask показывает системе, какие права доступа отключить. Совместив первое и второе становится видно, что все права для группы и остальных пользователей будут отключены, в то время как права владельца останутся нетронутыми.

Изменение suid и sgid

Способ установки и удаления битов suid и sgid чрезвычайно прост. Чтобы задать бит suid:
chmod u+s /home/sit/itmo.sh
Чтобы задать бит sgid:
chmod g+s /home/sit/itmo/

Определение первого знака прав доступа

Он используется для задания битов sticky, suid и sgid совместно с правами доступа:

suid	sgid	sticky	режим
on	on	on	7
on	on	off	6
on	off	on	5
on	off	off	4
off	on	on	3
off	on	off	2
off	off	on	1
off	off	off	0

Ниже приведен пример того, как использовать четырех значный режим для установки прав доступа на директорию.

```
sit@ubuntu:~$ chmod 4775 /home/sit/itmo
sit@ubuntu:~$ ls -l /home/sit/itmo
-rwsrwxr-x 1 sit sit 0 Sep  9 12:42 /home/sit/itmo
```

Консольные команды:

- id <печать идентификатора пользователя>
- chgrp <изменить группу файла>
- chown <изменить владельца и группу файлов>
- chmod <изменить права доступа к файлу>
- usermod <изменение параметров учетной записи пользователя>
- useradd <создание нового пользователя>
- userdel <удаление пользователя>
- whoami <определение текущего пользователя>
- umask <определение или установление маски прав доступа для вновь создаваемых файлов>
- sudo su <получение прав суперпользователя>
- groups <определение к каким группам принадлежит пользователь>

Практическая работа №3
Управление правами доступа пользователей/групп к информационным ресурсам

- 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:
- Изучите как создать пользователя с домашним каталогом с помощью команды useradd из справочной документации man
- Используя useradd создайте пользователя «sit2» с домашним каталогом «sit2».
- Установите пароль для нового пользователя «sit2» с помощью команды passwd sit2
- Выйдите из суперпользователя командой exit
- Войдите под первым терминалом в пользователя «sit», во втором в пользователя «sit2».
- Посмотрите какой идентификатор получил пользователь «sit» и пользователь «sit2» используя команду id

- Посмотрите права доступа на домашний каталог пользователей «sit» и «sit2», используя команду ls
- Создайте файл под пользователем «sit2» с маской 0077 используя umask
- Попробуйте прочитать его содержимое под пользователем «sit» используя команду cat
- Измените права доступа на файл так, чтобы пользователь «sit» мог записывать в файл, но не читать его.
- Запишите текстовую информацию в файл из под пользователя «sit» используя консольный текстовый редактор vi или nano
- Проверьте права на файл, и прочитайте его содержимое из под пользователя «sit2»
- Создайте каталог из под пользователя «sit2»
- Установите права записи для группы пользователей на данный каталог
- Добавьте пользователя «sit» в группу «sit2» с помощью команды usermod
- Проверьте в какие группы входит пользователь «sit»
- Создайте несколько файлов в каталоге, который был создан пользователем «sit2» из под пользователя «sit».
- Ознакомьтесь как удалить пользователя вместе с содержимым его домашнего каталога из справочной документации
- Удалите пользователя «sit2» вместе с его домашним каталогом.

Лабораторная работа №4,5
Изучение программных средств сканирования сетей и обнаружения атак
Методы защиты удаленного доступа к ресурсам компьютерной сети. Безопасная работа в интернете

Цель работы: Изучить способы применения и возможности общедоступных программных средств тестирования информационной безопасности (сканеров безопасности).

1. **Краткие теоретические сведения.**

Сканеры уязвимостей — это программные или аппаратные средства, служащие для осуществления диагностики и мониторинга сетевых компьютеров, позволяющее сканировать сети, компьютеры и приложения на предмет обнаружения возможных проблем в системе безопасности, оценивать и устранять уязвимости. Средства анализа защищенности исследуют сеть и ищут "слабые" места в ней, анализируют полученные результаты и на их основе создают различного рода отчеты. Они позволяют выявить такие типичные уязвимости информационных систем, как:

- "дыры" в программах (back door) и программы типа "троянский конь";
- слабые пароли;
- восприимчивость к проникновению из незащищенных систем;
- неправильная настройка межсетевых экранов, Web-серверов и баз данных.

Функционировать такие средства могут на сетевом уровне (network-based), уровне операционной системы (host-based) и уровне приложения (application-based). Наибольшее распространение получили средства анализа защищенности сетевых сервисов и протоколов. Вторыми по распространенности являются средства анализа защищенности операционных систем (ОС). Помимо обнаружения уязвимостей, при помощи средств анализа защищенности можно быстро определить все узлы корпоративной сети, доступные в момент проведения тестирования, выявить все используемые в ней сервисы и протоколы, их настройки и возможности для несанкционированного воздействия (как изнутри корпоративной сети, так и снаружи). Также эти средства вырабатывают рекомендации и пошаговые меры, позволяющие устранить выявленные недостатки.

Существует два основных механизма, при помощи которых сканер проверяет наличие уязвимости - сканирование (scan) и зондирование (probe).

Сканирование - механизм пассивного анализа, с помощью которого сканер пытается определить наличие уязвимости без фактического подтверждения ее наличия - по косвенным признакам. Этот метод является наиболее быстрым и простым для реализации.
Зондирование - механизм активного анализа, который позволяет убедиться, присутствует или нет на анализируемом узле уязвимость.Зондирование выполняется путем имитации атаки, использующей проверяемую уязвимость. Этот метод более медленный, чем "сканирование", но почти всегда гораздо более точный, чем он.
На практике указанные механизмы реализуются следующими несколькими методами.
"Проверка заголовков" (banner check). Указанный механизм представляет собой ряд проверок типа "сканирование" и позволяет делать вывод об уязвимости, опираясь на информацию в заголовке ответа на запрос сканера. Типичный пример такой проверки - анализ заголовков программы Sendmail или FTP-сервера, позволяющий узнать их версию и на основе этой информации сделать вывод о наличии в них уязвимости. Наиболее быстрый и простой для реализации метод проверки присутствия на сканируемом узле уязвимости.**"Активные зондирующие проверки" (active probing check).** Также относятся к механизму "сканирования". Однако они основаны не на проверках версий программного обеспечения в заголовках, а на сравнении "цифрового слепка" (fingerprint) фрагмента программного обеспечения со слепком известной уязвимости. Аналогичным образом поступают антивирусные системы, сравнивая фрагменты сканируемого программного обеспечения с сигнатурами вирусов, хранящимися в специализированной базе данных. Этот метод также достаточно быстр, но реализуется труднее, чем "проверка заголовков".
"Имитация атак" (exploit check). Данные проверки относятся к механизму "зондирования" и основаны на эксплуатации различных дефектов в программном обеспечении.
Этапы сканирования. Практически любой сканер проводит анализ защищенности в несколько этапов:

- Сбор информации о сети. На данном этапе идентифицируются все активные устройства в сети и определяются запущенные на них сервисы и демоны.
- Обнаружение потенциальных уязвимостей. Сканер использует некую базу данных для сравнения собранных данных с известными уязвимостями при помощи проверки заголовков или активных зондирующих проверок.
- Подтверждение выбранных уязвимостей. Сканер использует специальные методы и моделирует (имитирует) определенные атаки для подтверждения факта наличия уязвимостей на выбранных узлах сети.
- Генерация отчетов. На основе собранной информации система анализа защищенности создает отчеты, описывающие обнаруженные уязвимости.
- Автоматическое устранение уязвимостей. Этот этап очень редко реализуется в сетевых сканерах, но широко применяется в системных сканерах.

3. **Методические указания.**
    1. При отсутствии необходимых приложений на рабочей станции, они могут быть скачаны с сайта по дисциплине /ics.
    2. Приложения необхоимо запускать из командной строки, в программе «Терминал».
    3. При необходимости получения справки по утилитам и клиентским приложениям, их нобходимо запускать с ключем /? или /h.
    4. Результаты работы приложений, необходимые для отчета, следует, убедившись в корректности их работы, перенаправлять в текстовый файл.
4. **Порядок выполнения работы.**
    1. Используя утилиту ping, убедиться в доступности заданного сервера.
    2. Используя последовательно программы nmap и nessus, просканировать (но не зондировать!) указанный сервер.
    3. Определить наличие уязвимостей, слабых паролей.
    4. Определить ОС сервера.

Практическая работа № 4,5
Изучение программных средств сканирования сетей и обнаружения атак. Методы защиты удаленного доступа к ресурсам компьютерной сети. Безопасная работа в интернете

Варианты заданий.

1. Исследовать наличие открытых портов сервера 217.71.139.66 и определить тип операционной системы сервера. Определить наличие уязвимостей, слабых паролей.
2. Исследовать наличие открытых портов сервера 217.71.139.95 и определить тип операционной системы сервера. Определить наличие уязвимостей, слабых паролей.
3. Исследовать наличие открытых портов сервера 217.71.139.228 и определить тип операционной системы сервера. Определить наличие уязвимостей, слабых паролей.
4. Исследовать наличие открытых портов сервера 217.71.139.236 и определить тип операционной системы сервера. Определить наличие уязвимостей, слабых паролей.
5. Исследовать наличие открытых портов сервера 217.71.138.1 и определить тип операционной системы сервера. Определить наличие уязвимостей, слабых паролей.
6. Исследовать наличие открытых портов сервера 217.71.138.3 и определить тип операционной системы сервера. Определить наличие уязвимостей, слабых паролей.

Лабораторная работа №6
Изучение антивирусных программных комплексов

Обзор законодательной базы, регулирующей область защиты информации, с помощью ПСС «Гарант»

Практическая работа №6
Изучение антивирусных программных комплексов

Обзор программ-антивирусов. Составление доклада

Рекомендуемая литература
Основная литература

1. Голиков, А.М. Защита информации в инфокоммуникационных системах и сетях : учебное пособие / А.М. Голиков ; Министерство образования и науки Российской Федерации, Томский Государственный Университет Систем Управления и Радиоэлектроники (ТУСУР). – Томск : Томский государственный университет систем управления и радиоэлектроники, 2015. – 284 с. : схем., табл., ил. – Режим доступа: по подписке. – URL: http://biblioclub.ru/index.php?page=book&id=480637

2. Информационные системы и их безопасность [Текст] : учебное пособие / А. В. Васильков, А. А. Васильков, И. А. Васильков. - Москва : Форум, 2015. - 528 с. - Библиогр. : с. 513-514. - ISBN 978-5-91134-289-0. (ОГТИ ч/з N4-1; аб.ТБ-18), коэфициент книгообеспеченности 1

3. Прохорова, О.В. Информационная безопасность и защита информации : учебник / О.В. Прохорова ; Министерство образования и науки РФ, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Самарский государственный архитектурно-строительный университет». – Самара : Самарский государственный архитектурно-строительный университет, 2014. – 113 с. : табл., схем., ил. – Режим доступа: по подписке. – URL: http://biblioclub.ru/index.php?page=book&id=438331

Дополнительная литература

1. Системы защиты информации в ведущих зарубежных странах : учебное пособие для вузов / В.И. Аверченков, М.Ю. Рытов, Г.В. Кондрашин, М.В. Рудановский. - 3-е изд., стер. - М. : Флинта, 2011. - 224 с. - (Организация и технология защиты информации). - ISBN 978-5-9765-1274-0 ; То же [Электронный ресурс]. – URL: //biblioclub.ru/index.php?page=book_red&id=93351, коэффициент книгообеспеченности 1.

2. Основы информационной безопасности. Учебно-практическое пособие [Электронный ресурс] / Сычев Ю. Н. - Евразийский открытый институт, 2010.]. - URL: //biblioclub.ru/index.php?page=book&id=93351, коэффициент книгообеспеченности 1.

3. Основы информационной безопасности при работе на компьютере [Электронный ресурс] / Фаронов А. Е. - Интернет-Университет Информационных Технологий, 2011.- .URL://biblioclub.ru/index.php?page=book_red&id=233763&sr=1, коэффициент книгообеспеченности 1.

4. Правовые основы информатики. Учебно-практическое пособие [Электронный ресурс] / Ефимова Л. Л. - Евразийский открытый институт, 2011. - URL://biblioclub.ru/index.php?page=book_red&id=93155&sr=1, коэффициент книгообеспеченности 1.

5. Организация безопасной работы информационных систем : учебное пособие / Ю.Ю. Громов, Ю.Ф. Мартемьянов, Ю.К. Букурако и др. ; Министерство образования и науки Российской Федерации, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Тамбовский государственный технический университет». - Тамбов : Издательство ФГБОУ ВПО «ТГТУ», 2014. - 132 с. : ил. - Библиогр. в кн. ; То же [Электронный ресурс]. - URL: //biblioclub.ru/index.php?page=book&id=277794, коэффициент книгообеспеченности 1.

6. Креопалов, В.В. Технические средства и методы защиты информации : учебно-практическое пособие / В.В. Креопалов. - М. : Евразийский открытый институт, 2011. - 278 с. - ISBN 978-5-374-00507-3 ; То же [Электронный ресурс]. - URL: //biblioclub.ru/index.php?page=book&id=90753, коэффициент книгообеспеченности 1.

7. Смирнов, В.И. Защита информации : лабораторный практикум / В.И. Смирнов ; Поволжский государственный технологический университет. – Йошкар-Ола : ПГТУ, 2017. – 67 с. : ил. – Режим доступа: по подписке. – URL: http://biblioclub.ru/index.php?page=book&id=476512

Периодические издания

1. Журнал «Вестник компьютерных и информационных технологий »
2. Журнал «Информационные технологии и вычислительные системы»
3. Журнал «Стандарты и качество»
4. Журнал «Информатика и вычислительная техника»