Лабораторная работа № 6

Мандатное разграничение прав в Linux

Сухарев Кирилл

Содержание

# Цель работы

Развить навыки администрирования ОС Linux. Получить первое практическое знакомство с технологией SELinux. Проверить работу SELinx на практике совместно с веб-сервером Apache.

# Условные обозначения и термины

**Утилита** - сервисная программа, облегчающая пользование другими программами, работу с компьютером.

**Учетная запись** - хранимая в компьютерной системе совокупность данных о пользователе, необходимая для его опознавания (аутентификации) и предоставления доступа к его личным данным и настройкам.

**Директория** - объект в файловой системе, упрощающий организацию файлов.

# Теоретические вводные данные

Мандатная модель управления доступом (Mandatory Access Control, MAC) — способ разграничения доступа с фиксированным набором полномочий. Обычно настоящий MAC используется в системах с повышенным требованиями к безопасности и стоит на службе всевозможных силовых ведомств и организаций, связанных с государственной или служебной тайной.

Модель MAC по своей сути является «электронной» реализацией бумажного «секретного» документооборота. В MAC имеются следующие «действующие лица»:

* Иерархия уровней доступа, которые обрабатываются в системе (обычно регистрируются в ОС). Для удобства часто задается в виде беззнаковых чисел (от 0 до значения, ограниченного реализацией). В этом случае для сравнения уровней доступа (выше/ниже/равно) используются простейшие арифметические операции (равно, меньше, больше).
* Объект с уровнем секретности. Любой файл, каталог в файловой системе, ячейка или запись в таблице БД, таблица в БД, сама БД, сетевой пакет и т.д. Объекту присваивается любое значение из иерархии уровней доступа. Для объекта допускается повышение уровня секретности (изменение до большего значения уровня, чем текущий). Понижение уровня секретности категорически не допускается (хотя вполне реализуемо при помощи определенных уловок).
* Субъект с уровнем доступа. Процесс какого-либо приложения либо сеанс пользователя (по сути тоже процесс приложения). Метка уровня доступа наследуется от субъекта всеми создаваемыми данным субъектом объектами.

Значение уровня доступа субъекта или уровня секретности объекта обычно называют термином «мандатный уровень», «мандатная метка» или просто «метка» (в STCSEC данный термин называется «hierarchical classification level»). Просто, емко и почти однозначно.

Проверка полномочий осуществляется при каждом факте доступа субъекта к объекту, защищаемому MAC. При этом мандатная модель управления доступом обычно используется совместно с другими механизмами контроля доступа, например, DAC (UNIX-моделью и POSIX ACL). При этом MAC проверяется в последнюю очередь. Сперва проверяется доступ по DAC (как наименее защищенный), а затем уже MAC.

При проверке правомочности доступа субъекта к объекту согласно мандатной модели возможны следующие комбинации:

1. *Мандатная метка субъекта равна мандатной метке объекта.* В этом случае субъекту разрешено читать и изменять объект.
2. *Мандатная метка субъекта выше мандатной метки объекта.* Субъекту разрешено только читать объект: он его видит, но не может изменить.
3. *Мандатная метка субъекта ниже мандатной метки объекта.* Субъекту формально разрешено создать объект с более высокой мандатной меткой (так называемое «повышение уровня секретности объекта»). На практике у субъекта нет технической возможности для выполнения данной операции (он просто «не видит» изменяемый объект, например, файл или каталог с файлами).

Также в MAC существует такое понятие, как «категория» (в терминологии STCSEC данный термин называется «non-hierarchical categories»). Категории в MAC являются опциональными к применению. В практике реализации MAC категории используются для «горизонтального» разграничения доступа между различными подразделениями организации. В этом случае сотрудники, несмотря на один мандатный уровень, будут получать доступ только к тем категориям объектов, к которым для них открыт доступ согласно их метке.

# Техническое оснащение и выбранные методы проведения работы

В качестве среды выполнения лабораторной работы используется менеджер виртуальных машин VirtualBox и установленная с его помощью OC Centos 7 на базе Linux.

# Выполнение работы

1. Войдем в систему под пользователем guest выполним команды **getenforce** и **sestatus**. Видим, что SELinux работает в режиме enforcing политики targeted (fig. 1).

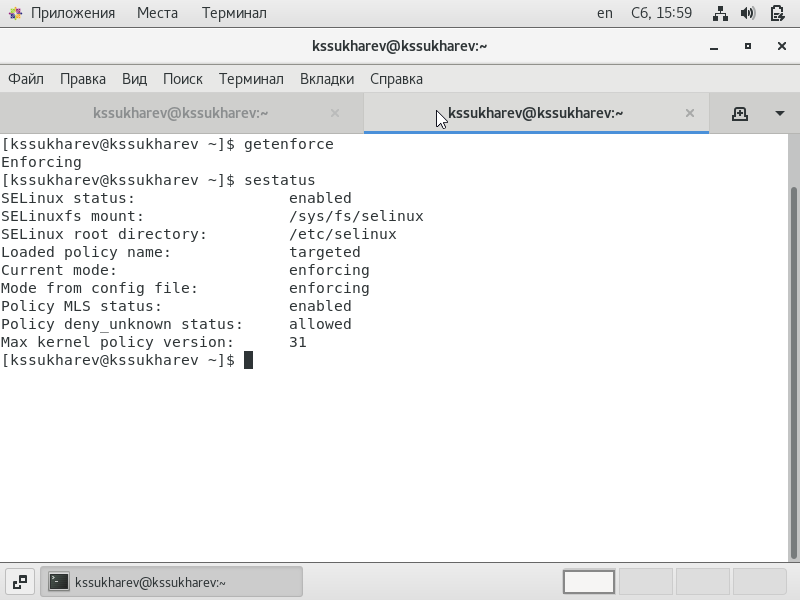


Figure 1: Проверка режима SELinux

1. Запустим веб-сервер командой **service httpd start** и обратимся к нему, при помощи браузера, затем проверим, что сервер работает командоа **service httpd status** (fig. 2).

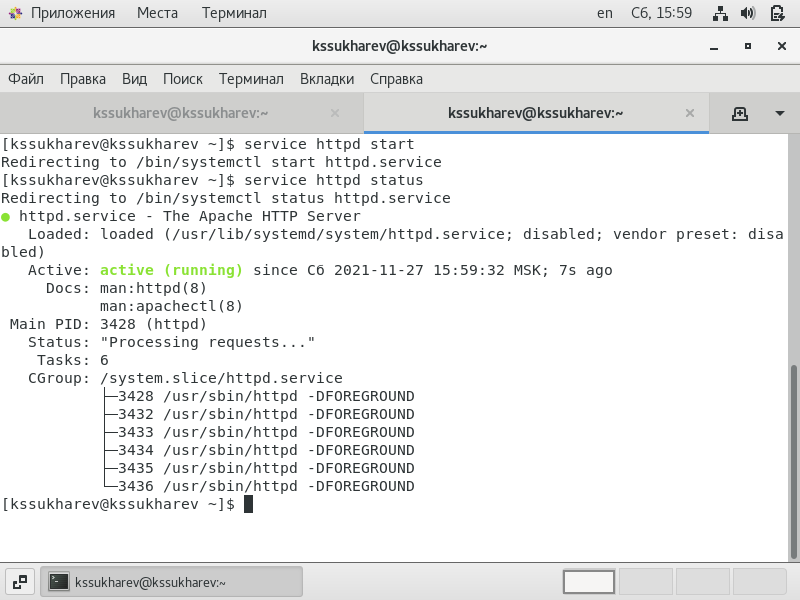


Figure 2: Запуск и проверка веб-сервера

1. Найдем веб-сервер Apache в списке процессов и определим его контекст безопасности командой **ps -eZ | grep httpd**. Получим следующий контекст system\_u:system)r:httpd\_t:s0 (fig. 3).

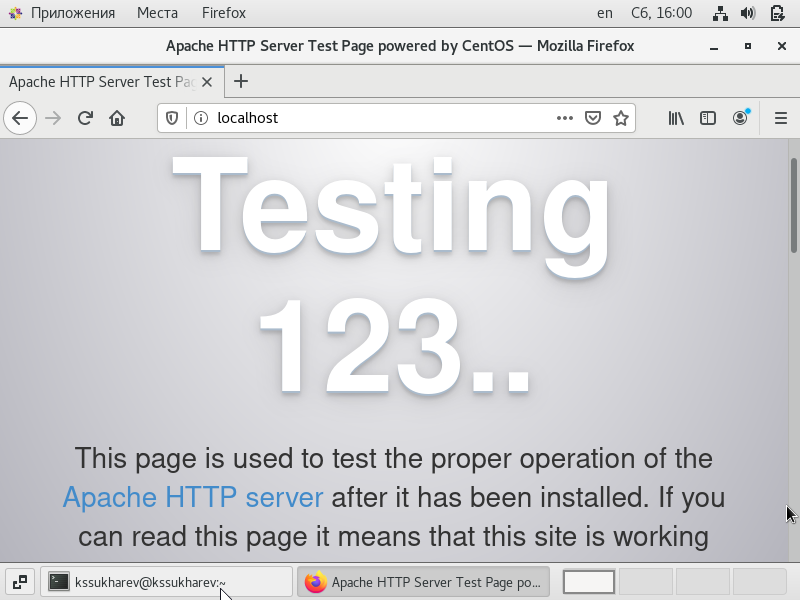


Figure 3: Определение контекста безопасноти

1. Просмотрим текущее состояние переключателей SELinux для Apache с помощью команды **sestatus -b | grep httpd**. Большинство из них находятся в положении “off” (fig. 4).

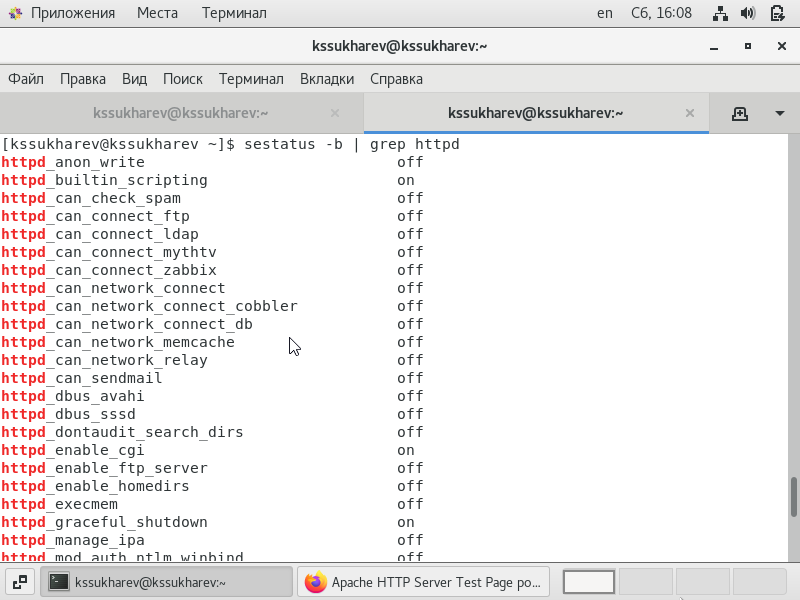


Figure 4: Проверка текущего состояния переключателей SELinux

1. Посмотрим на статистику по политике с помощью команды **seinfo**, также определим множество пользователей (ключ **-u**), ролей (ключ **-r**), типов (ключ **-t**) (fig. 5).

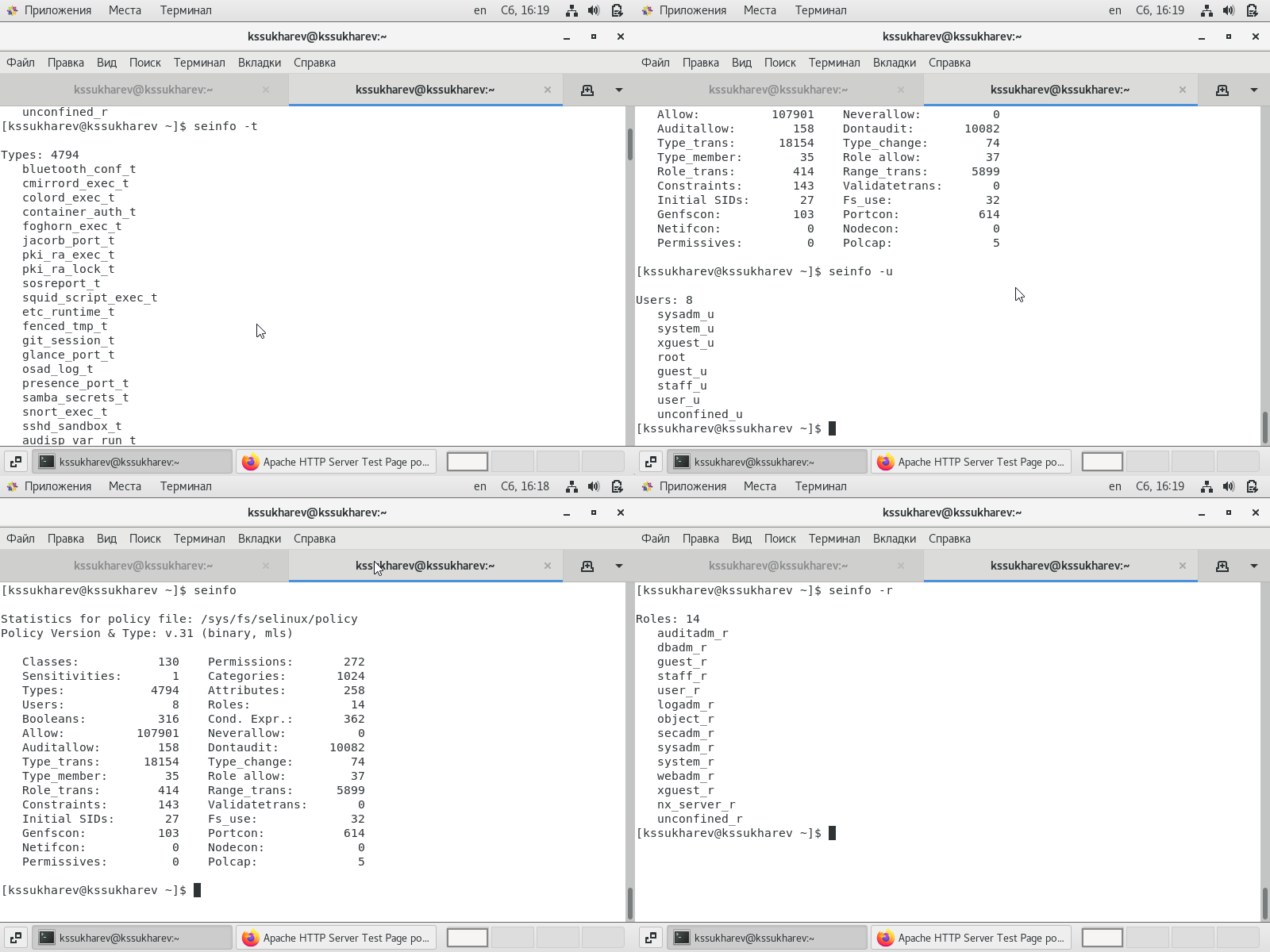


Figure 5: Проверка статистики по политике

1. Определим тип файлов, находящихся в директории /var/www при помощи команды **ls -lZ /var/www** - httpd\_sys\_script\_exec\_t и httpd\_sys\_content\_t (fig. 6).

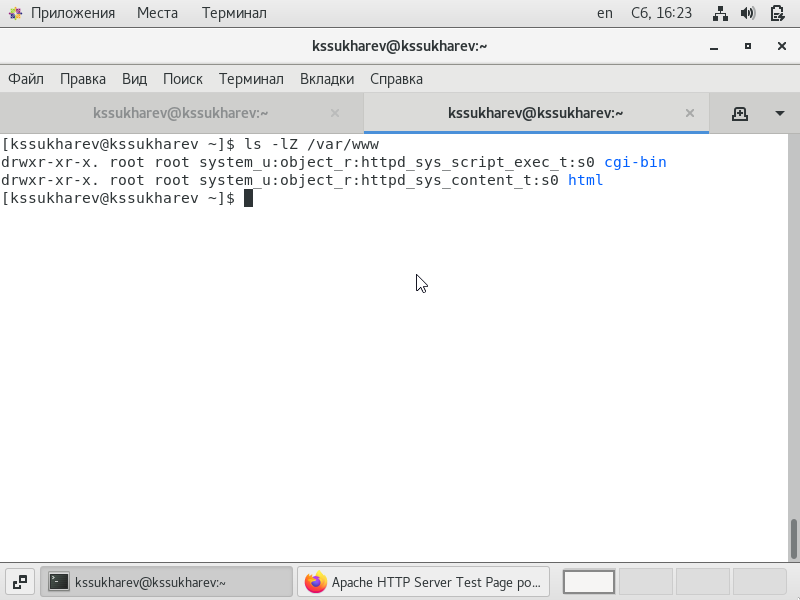


Figure 6: Проверка файлов в var/www

1. Аналогичным образом проверим директорию /var/www/html. Поскольку файлов в ней нет, то и типы определить не получится (fig. 7).

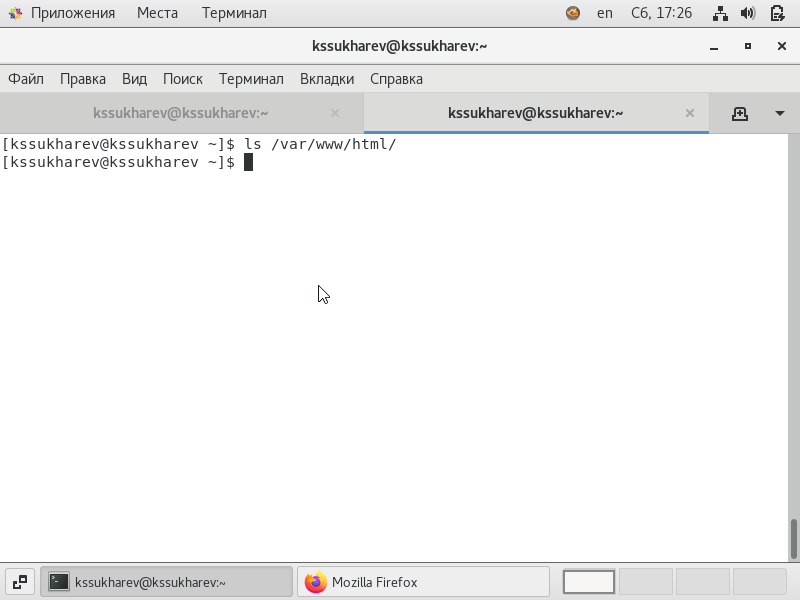


Figure 7: Проверка файлов в var/www/html

1. Определим круг пользователей, которым разрешено создание файлов в диреткории /var/www/html при помощи команды **ls -lZ /var/www**. Видим, что установлен пользователь system\_u, что говорит о том, что создавать файлы в директории может только суперпользователь (fig. 8).

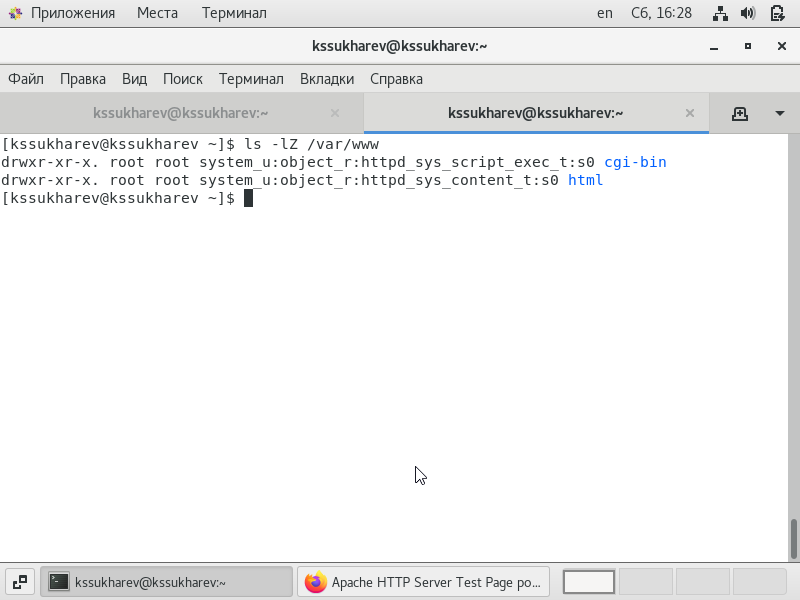


Figure 8: Проверка владельцев var/www/html

1. От имени суперпользователя создадим html-файл /var/www/html/test.html (fig. 9).

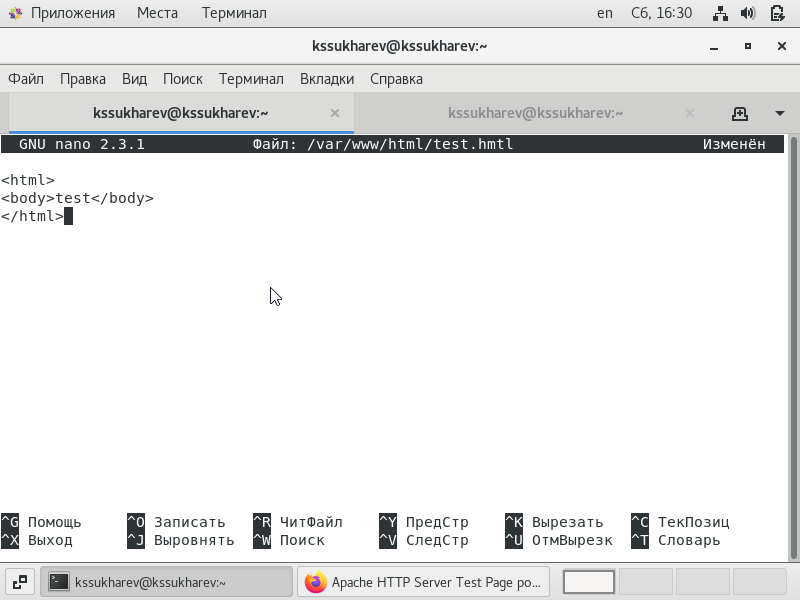


Figure 9: Создание test.html

1. Проверим контекст созданного файла командой **ls -lZ /var/www/html**. По умолчанию созданным файлам присваивается контекст unconfined\_u:object\_r:httpd\_sys\_content\_t:s0 (fig. 10).

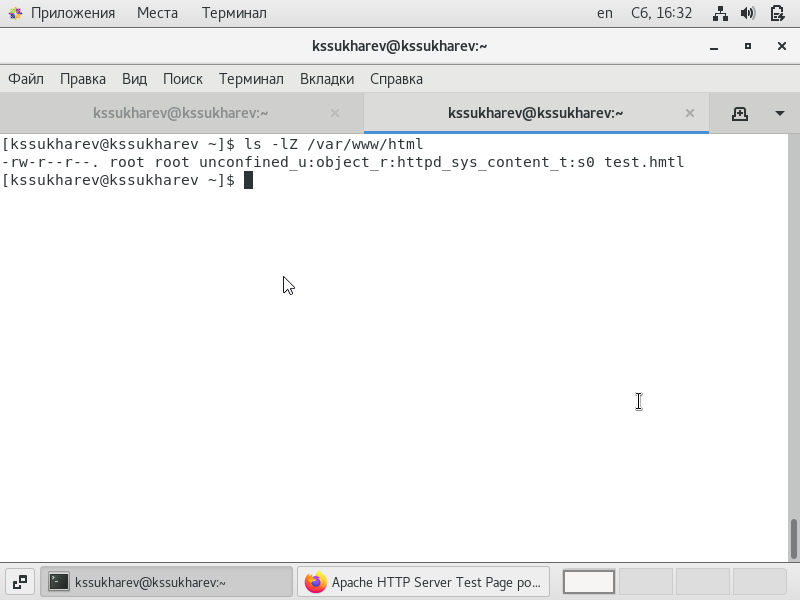


Figure 10: Контекст test.html

1. Обратимя к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1/test.html. Файл был успешно отображен (fig. 11).

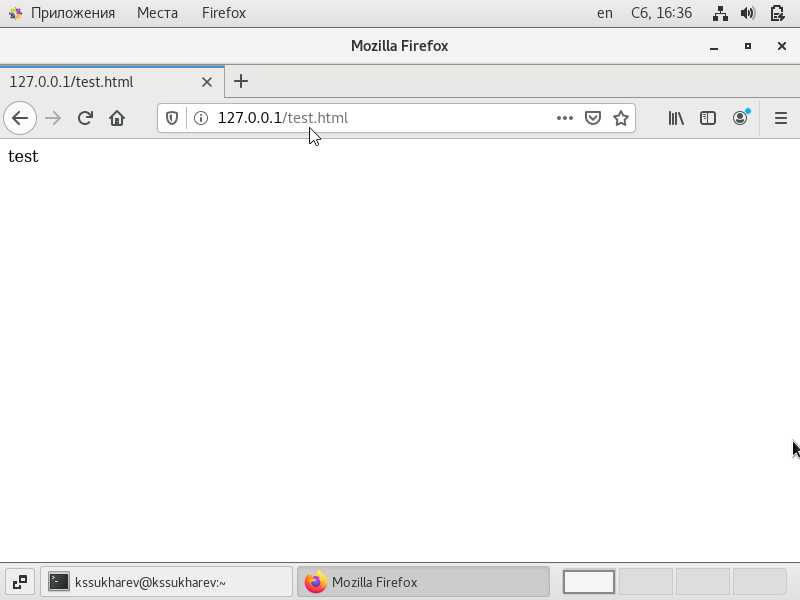


Figure 11: Проверка test.html

1. Изучим справку командой **man httpd\_selinux**. Выясняется, что контекст был выбран верно (fig. 12).

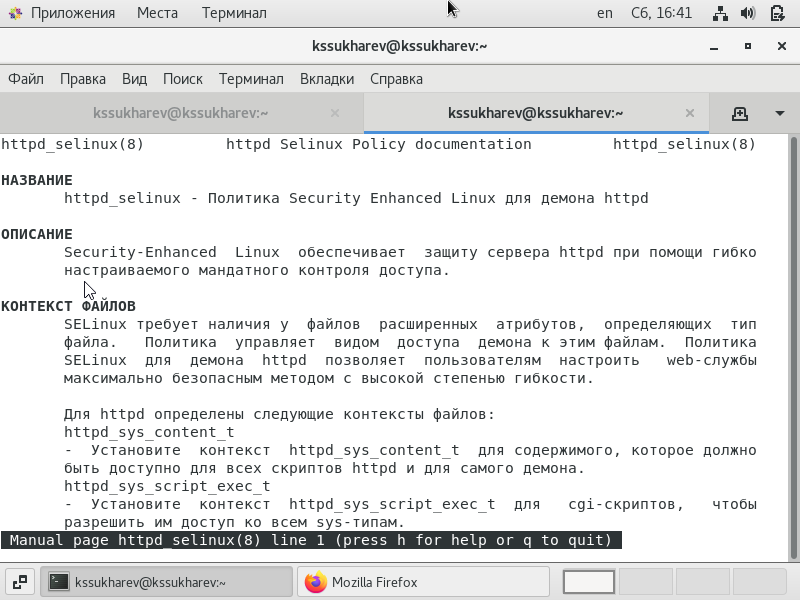


Figure 12: Изучение справки httpd\_selinux

1. Изменим контекст файла test.html, например на samba\_share\_t командой **chcon -t samba\_share\_t /var/www/html/test.html**. Затем командой **ls -Z /var/www/html/test.html** убедимся, что контекст изменился (fig. 13).

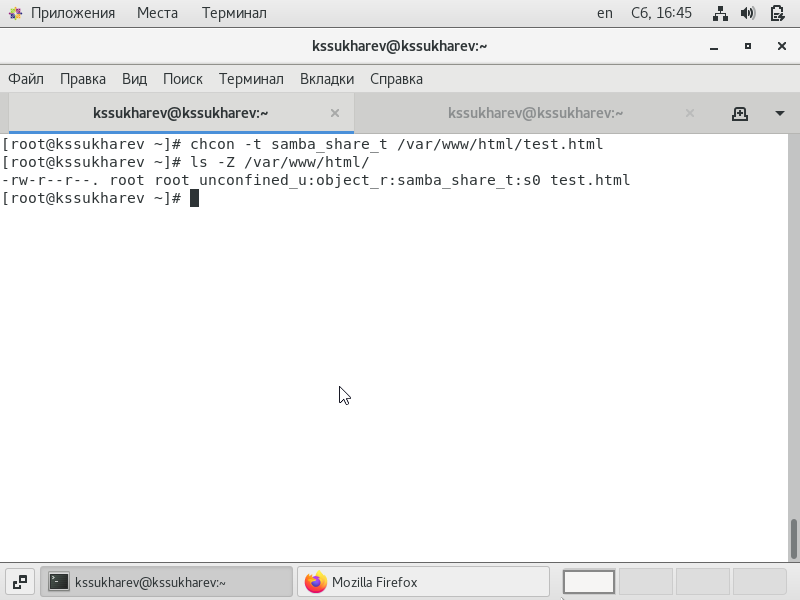


Figure 13: Изменение контекста test.html

1. Попробуем еще раз получить доступ к файлу через веб-сервер. Получим ошибку (fig. 14).



Figure 14: Попытка получить доступ к файлу

1. Посмотрим логи командой **tail /var/log/messages** и увидим, что ошибка возникла, поскольку служба httpd не имеет доступа к выбранному нами типу файлов из-за разницы контекстов (fig. 15).

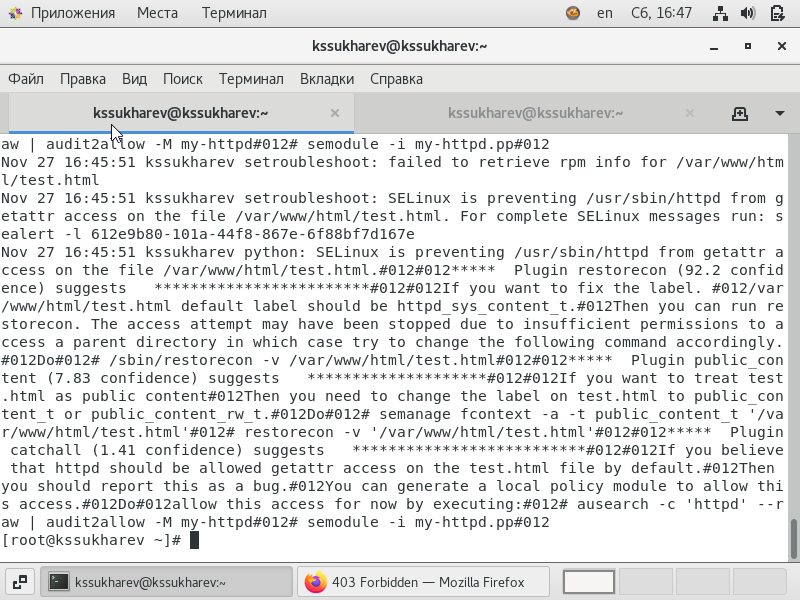


Figure 15: Проверка логов

1. Изменим в файле /etc/httpd/httpd.conf строчку Listen 80 на Listen 81 (fig. 16).

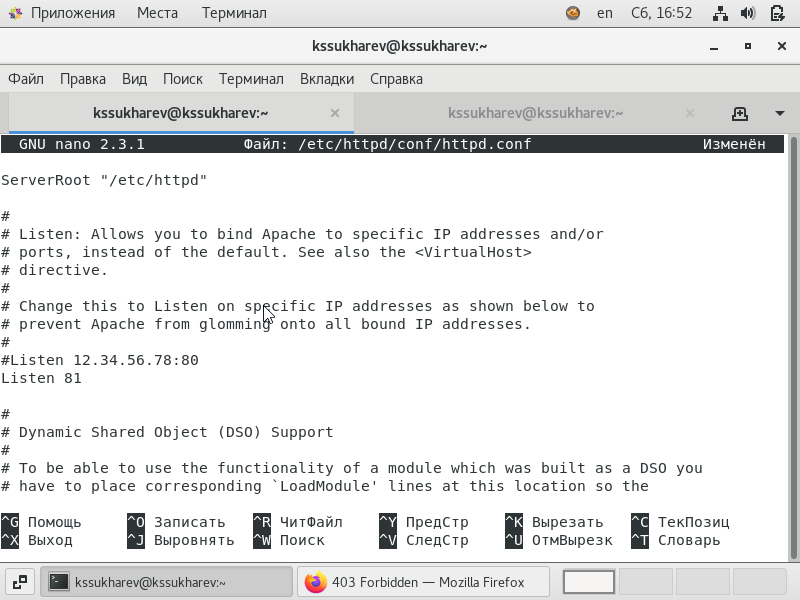


Figure 16: Смена порта

1. Попробуем перезапустить веб-сервер Apache командой **service httpd restart**. Сбоя не произошло (fig. 17).

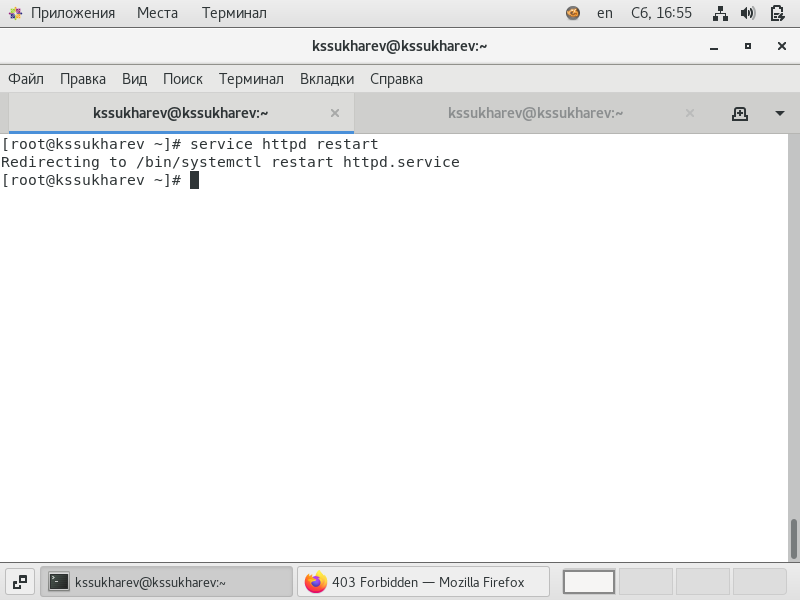


Figure 17: Перезапуск веб-сервера

1. Проанализируем лог-файлы командой **tail -n1 /var/log/messages**. Видим, что сервер был успешно запущен (fig. 18).

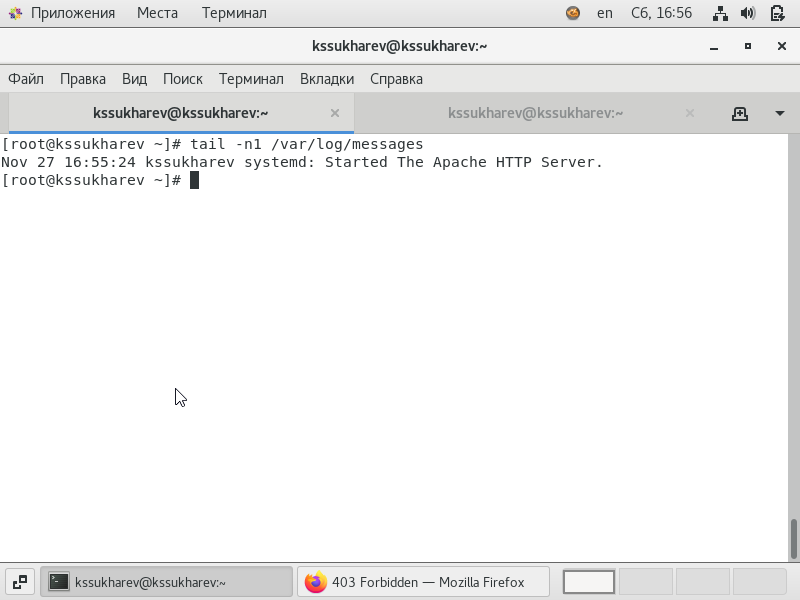


Figure 18: Анализ лог-файлов

1. Выполним команду **semanage port -a -t http\_port\_t -p tcp 81** и затем проверим, что порт 81 появился в выводе команды **semanage port -l | grep http\_port\_t**. Видим, что порт уже был в этом списке (fig. 19).

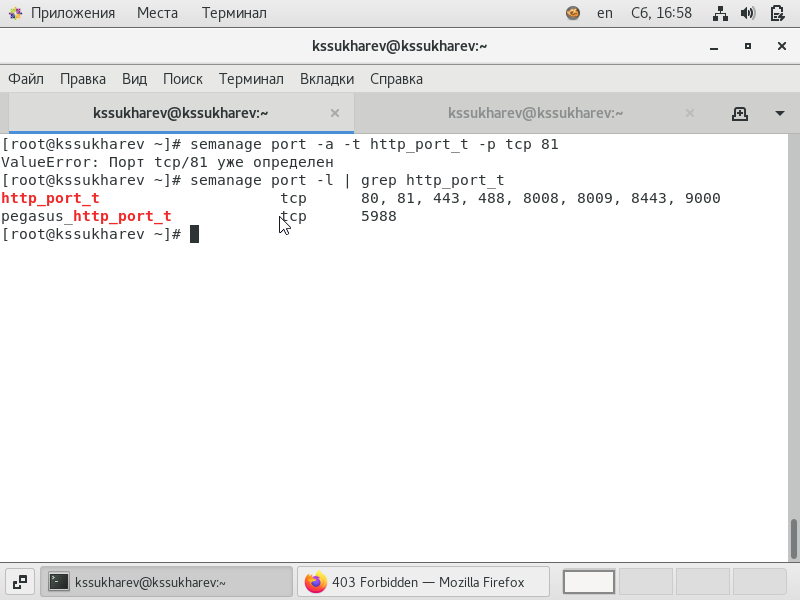


Figure 19: Добавление порта

1. Попробуем запустить веб-сервер еще раз. Как и в прошлый раз, он запустился, поскольку 81 порт уже был в политике (fig. 20).

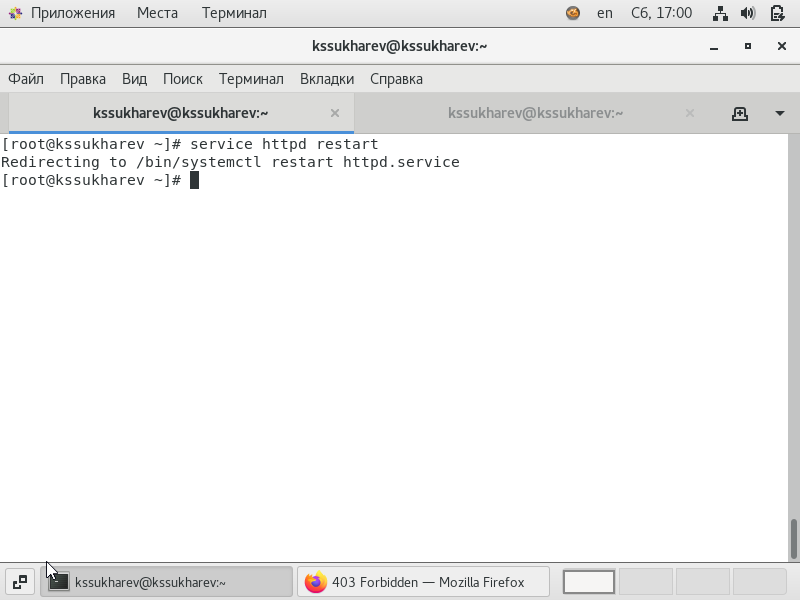


Figure 20: Повторный запуск сервера

1. Вернем нормальный контекст командой **chcon -t httpd\_sys\_content\_t /var/www/html/test.html**. Затем снова попробуем получить доступ к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1:81/test.html (fig. 21).

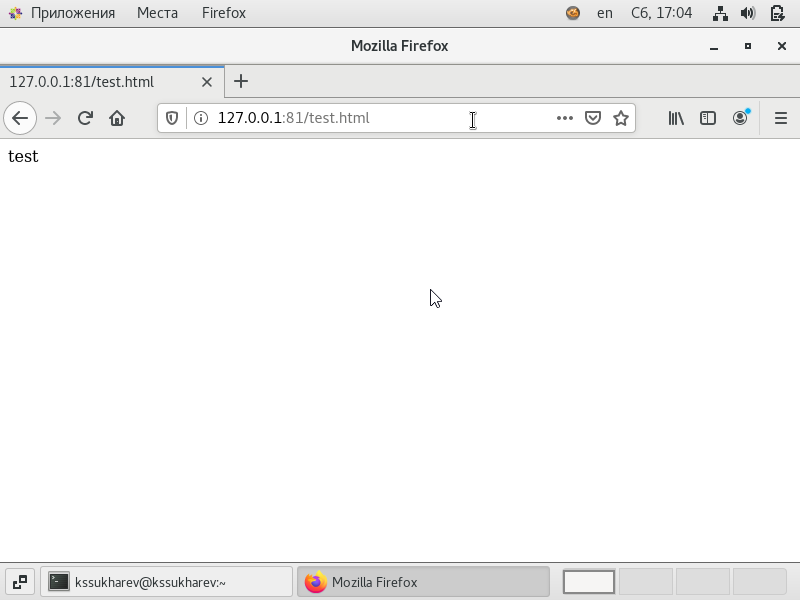


Figure 21: Попытка доступ к файлу через веб-сервер

1. Вновь вернем пот 80 в конфигурационном файле /etc/httpd/conf/httpd.conf (fig. 22).

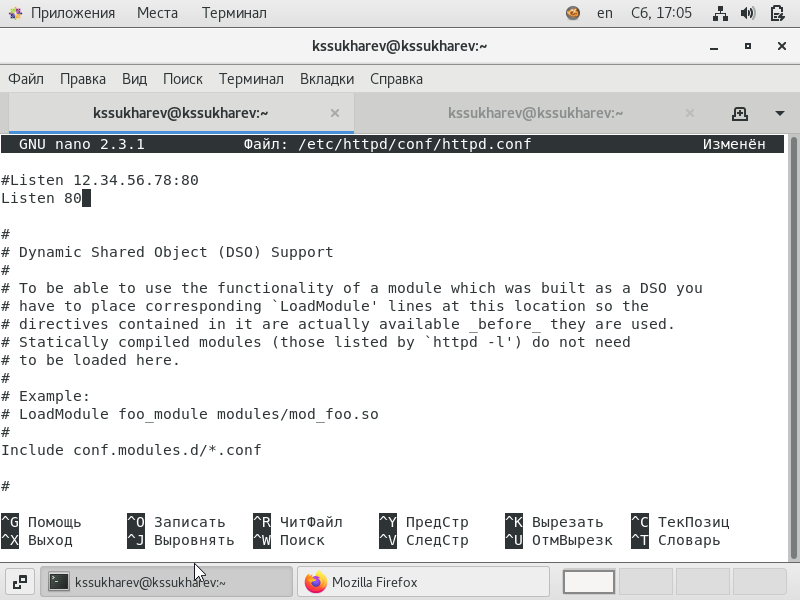


Figure 22: Возврат порта 80

1. Удалим привязку http\_port\_t к 81 порту командой **semanage port -d -t http\_port\_t -p tcp 81**. Сделать это не получится, поскольку порт определен политикой(fig. 23).

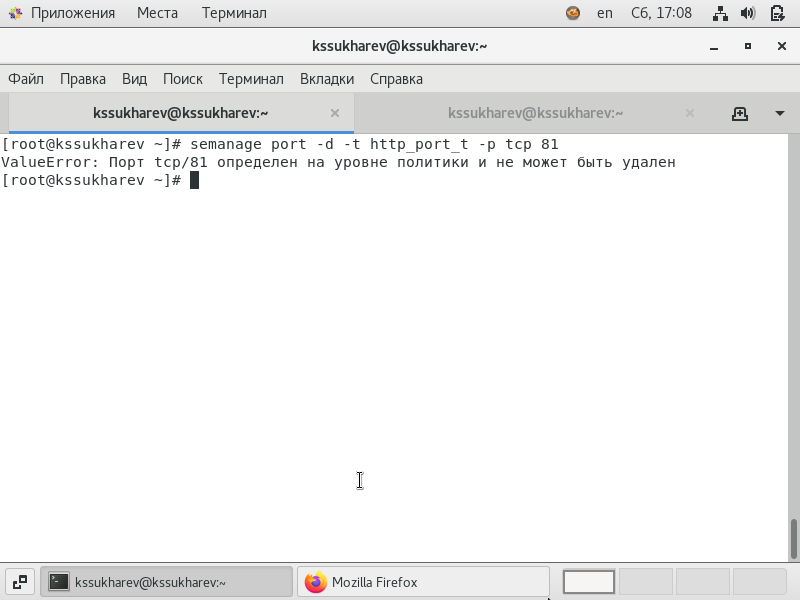


Figure 23: Удаление привязки

1. Удалим файл /var/www/html/test.html (fig. 24).

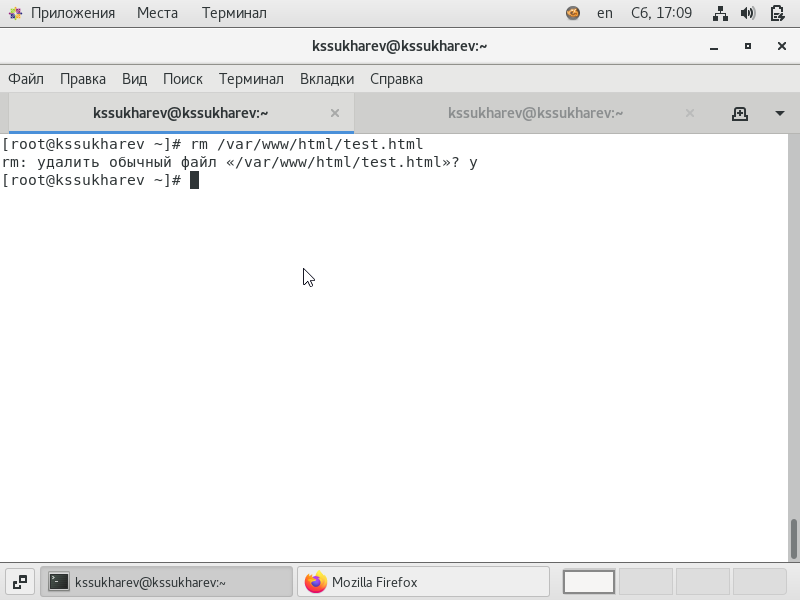


Figure 24: Удаление test.html

# Выводы

Были развиты навыки администрирования ОС Linux, получено первое практическое знакомство с технологией SELinux. Также мы проверили работу SELinx на практике совместно с веб-сервером Apache.

# Библиография

1. Мандатная модель управления доступом (MAC): обзор и применение в прикладных системах. URL: https://habr.com/ru/company/avanpost/blog/482060/ (Дата обращения: 27.11.2021).
2. Д. С. Кулябов, А. В. Королькова, М. Н. Геворкян. Информационная безопасность компьютерных сетей: лабораторные работы. // Факультет физико-математических и естественных наук. M.: РУДН, 2015. 64 с..