|  |
| --- |
| **Презентация лабораторной работы №6**  Ханина Людмила Константиновна  **Презентация к лабораторной работе №6**  **Выполнение лабораторной работы**  **Входим в систему под своей учетной записью и убеждаемся, что SELinux работает в режиме enforcing политики targeted:**  Проверка режима enforcing политики targeted  **Обращаемся с помощью браузера к веб-серверу и убеждаемся, что последний работает:**  Проверка работы веб-сервера  **Определяем контекст безопасности веб-сервера Apache:**  Контекст безопасности веб-сервера Apache  **Посмотрим текущее состояние переключателей SELinux для Apache, многие из переключателей находятся в положении “off”:**  Текущее состояние переключателей SELinux  **Посмотрим статистику по политике. Множество пользователей - 8, ролей - 14, типов 5100:**  Статистика по политике  **Посмотрим файлы и поддиректории, находящиеся в директории /var/www. Определим, что в данной директории файлов нет. Только владелец или** |

|  |
| --- |
| **суперпользователь может создавать файлы в директории /var/www/html:**  Просмотр файлов и поддиректориий в директории /var/www  **От имени суперпользователя создаём html-файл /var/www/html/test.html. Контекст созданного файла - httpd\_sys\_content\_t:**  Создание файла /var/www/html/test.html  **Обращаемся к файлу через веб-сервер, введя в браузере адрес “http://127.0.0.1/test.html”. Файл был успешно отображен:**  Обращение к файлу через веб-сервер  **Изучив справку man httpd\_selinux, выясняем, что для httpd определены следующие контексты файлов: httpd\_sys\_content\_t, httpd\_sys\_script\_exec\_t, httpd\_sys\_script\_ro\_t, httpd\_sys\_script\_rw\_t, httpd\_sys\_script\_ra\_t, httpd\_unconfined\_script\_exec\_t. Контекст моего файла - httpd\_sys\_content\_t (в таком случае содержимое должно быть доступно для всех скриптов httpd и для самого демона). Изменяем контекст файла на samba\_share\_t и проверяем, что контекст поменялся:**  Изменение контекста  **Попробуем еще раз получить доступ к файлу через веб-сервер, введя в браузере адрес**  **“http://127.0.0.1/test.html”, и получаем сообщение об ошибке (т.к. к установленному ранее контексту процесс httpd не имеет доступа):**  Обращение к файлу через веб-сервер  **Командой ls -l /var/www/html/test.html убеждаемся, что читать данный файл может любой** |

|  |
| --- |
| **пользователь. Просматриваем системный лог-файл веб-сервера Apache:**  Просмотр log-файла  **В файле /etc/httpd/conf/httpd.conf заменяем строчку “Listen 80” на “Listen 81”, чтобы установить вебсервер Apache на прослушивание TCP-порта 81:**  Установка веб-сервера Apache на прослушивание TCP-порта 81  **Перезапускаем веб-сервер Apache и анализируем лог-файлы командой:**  Перезапуск веб-сервера и анализ лог-файлов  **Просматриваем файлы “var/log/http/error\_log”,**  **“/var/log/http/access\_log” и “/var/log/audit/audit.log” и выясняем, что запись появилась в последнем файле:**  Содержание файла var/log/audit/audit.log  **Убеждаемся, что порт TCP-81 установлен.**  **Проверяем список портов, убеждаемся, что порт 81 есть в списке и запускаем веб-сервер Apache снова:**  Проверка установки порта 81  **Вернём контекст “httpd\_sys\_cоntent\_t” файлу “/var/www/html/test.html” и после этого пробуем получить доступ к файлу через веб-сервер, введя адрес “http://127.0.0.1:81/test.html”, в результате чего увидим содежимое файла - слово “test”:**    **Исправим обратно конфигурационный файл apache, вернув “Listen 80”. Попытаемся удалить привязку http\_port к 81 порту, но этот порт определен на уровне политики, поэтому его нельзя удалить:** |

|  |
| --- |
| Возвращение Listen 80 и попытка удалить порт 81  **Удаляем файл “/var/www/html/test.html”:**  Удаление файла test.html  **Выводы**  В ходе выполнения данной лабораторной работы я развила навыки администрирования ОС Linux, получила первое практическое знакомство с технологией SELinux и проверила работу SELinux на практике совместно с веб-сервером Apache. |