

Лабораторная работа № 6

Мандатное разграничение прав в Linux

Абакумов Егор Александрович

Содержание

Цель работы	5
Теоретическое описание	6
Ход работы	8
Выводы	17
Список литературы	18

List of Figures

0.1	Режим работы selinux	8
0.2	Проверка httpd	9
0.3	Процессы демона	9
0.4	Переключатели selinux	9
0.5	Статистика по политике	10
0.6	Множество пользователей	10
0.7	Множество ролей	11
0.8	Множество типов	12
0.9	Проверка типов в /var/www	13
0.10	Файл /var/www/html/test.html	13
0.11	Контекст файла test.html	13
0.12	Проверка файла в браузере	13
0.13	Изменение контекста	14
0.14	Ошибка доступа в браузере	14
0.15	Логи	14
0.16	Замена порта прослушивания	15
0.17	Отсутствие ошибок в логах	15
0.18	Список портов httpd в selinux	16
0.19	Возвращение верного контекста на файл test.html	16
0.20	Проверка доступа к файлу браузером через 81-ый порт	16

List of Tables

Цель работы

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

Теоретическое описание

SELinux представляет собой систему маркировки, каждый процесс имеет метку. Каждый файл, каталог или даже пользователь в системе имеет метку. Даже портам и устройствам и именам хостов в системе присвоены метки. SELinux определяет правила доступа процесса к объектам с определенными метками. Это и называется политикой. За соблюдением правил следит ядро. Иногда это еще называется обязательный контроль доступа (Mandatory Access Control, MAC). [1]

В дистрибутиве Linux MAC реализована поверх того, что мы называем моделью избирательного управления доступом (Discretionary Access Control), сокращённо DAC.

DAC – это управление доступом на основе списков управления доступом, где объектами доступа служат пользователь, группа и другие. Эти объекты имеют комбинацию полномочий чтение/запись/выполнение или r/w/x. SELinux позволяет ограничить доступ к объектам пользователя, который определён именно им так, что суперпользователь не может иметь суперпривилегии по отношению ко всем объектам нашего пользователя.

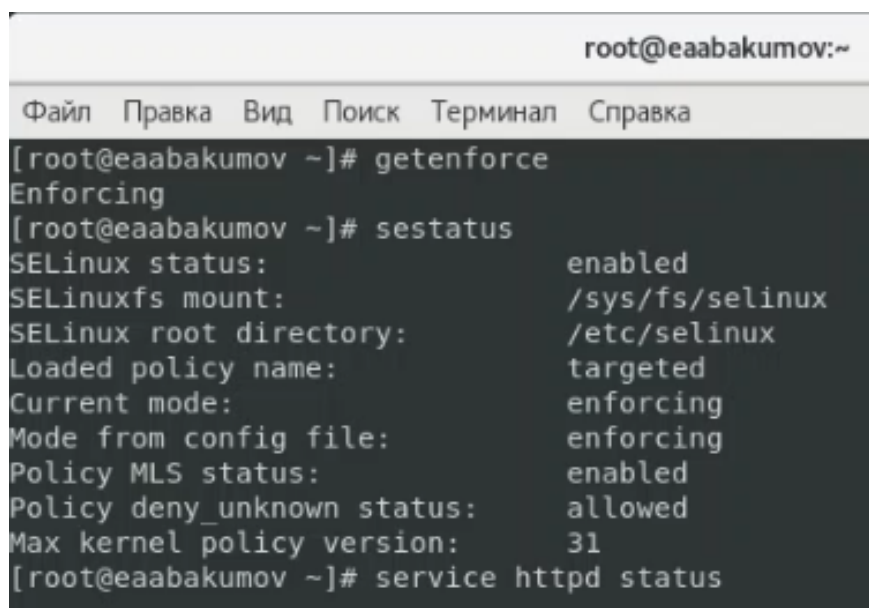
Другими словами, SELinux позволяет произвести точную настройку требований контроля доступа, с помощью которой определяется степень влияния пользователя на описанный настройкой процесс. Это в полной мере может предотвратить получение злоумышленником доступа ко всем объектам и процессам системы. [2]

Владелец файла не имеет полной свободы действий над атрибутами безопасности. Стандартные атрибуты контроля доступа, такие как группа и владелец ничего не значат для SELinux. Полностью все управляется метками. Значения атрибутов могут

быть установлены и без прав root, но на это нужно иметь специальные полномочия SELinux.

Ход работы

1. Готовим стенд. Для этого устанавливаем Apache, selinux command line tools и seinfo, дальше настраиваем ServerName в конфигурации httpd, разрешаем фильтру подключаться к 81-му порту. Делаем заранее за кадром.
2. Входим в систему и проверяем режим работы selinux (иллюстр. 0.1). Проверяем, что демон httpd включен и работает (иллюстр. 0.2). Находим процессы демона в списке и выписываем контекст: system_u:system_r:httpd_t:s0 (иллюстр. 0.3). Просматриваем переключатели selinux (иллюстр. 0.4).



```
root@eaabakumov:~  
Файл  Правка  Вид  Поиск  Терминал  Справка  
[root@eaabakumov ~]# getenforce  
Enforcing  
[root@eaabakumov ~]# sestatus  
SELinux status:                enabled  
SELinuxfs mount:                /sys/fs/selinux  
SELinux root directory:        /etc/selinux  
Loaded policy name:              targeted  
Current mode:                   enforcing  
Mode from config file:          enforcing  
Policy MLS status:              enabled  
Policy deny_unknown status:     allowed  
Max kernel policy version:      31  
[root@eaabakumov ~]# service httpd status
```

Figure 0.1: Режим работы selinux


```
root@eaabakumov:~  
Файл Правка Вид Поиск Терминал Справка  
t: disabled)  
Active: active (running) since Вт 2021-11-23 12:47:58 MSK; 20min ago  
Docs: man:htpd(8)  
man:apachectl(8)  
Process: 3469 ExecReload=/usr/sbin/htpd $OPTIONS -k graceful (code=exited, status=0/SUCCESS)  
Main PID: 3005 (htpd)  
Status: "Total requests: 0; Current requests/sec: 0; Current traffic: 0 B/s"  
Tasks: 6  
CGroup: /system.slice/htpd.service  
└─3005 /usr/sbin/htpd -DFOREGROUND  
└─3470 /usr/sbin/htpd -DFOREGROUND  
└─3471 /usr/sbin/htpd -DFOREGROUND  
└─3472 /usr/sbin/htpd -DFOREGROUND  
└─3473 /usr/sbin/htpd -DFOREGROUND  
└─3474 /usr/sbin/htpd -DFOREGROUND  
ноя 23 12:47:58 eaabakumov.localdomain systemd[1]: Starting The Apache HTTP...  
ноя 23 12:47:58 eaabakumov.localdomain systemd[1]: Started The Apache HTTP...  
ноя 23 13:06:01 eaabakumov.localdomain systemd[1]: Reloading The Apache HTTP...  
ноя 23 13:06:01 eaabakumov.localdomain systemd[1]: Reloaded The Apache HTTP...  
Hint: Some lines were ellipsized, use -l to show in full.  
[root@eaabakumov ~]#
```

Figure 0.2: Проверка htpd

```
[root@eaabakumov ~]# ps -eZ | grep htpd  
system_u:system_r:htpd_t:s0 3005 ? 00:00:00 htpd  
system_u:system_r:htpd_t:s0 3470 ? 00:00:00 htpd  
system_u:system_r:htpd_t:s0 3471 ? 00:00:00 htpd  
system_u:system_r:htpd_t:s0 3472 ? 00:00:00 htpd  
system_u:system_r:htpd_t:s0 3473 ? 00:00:00 htpd  
system_u:system_r:htpd_t:s0 3474 ? 00:00:00 htpd  
[root@eaabakumov ~]#
```

Figure 0.3: Процессы демона

```
[root@eaabakumov ~]# sestatus -b | grep htpd  
htpd_anon_write off  
htpd_builtin_scripting on  
htpd_can_check_spam off  
htpd_can_connect_ftp off  
htpd_can_connect_ldap off  
htpd_can_connect_mythtv off  
htpd_can_connect_zabbix off  
htpd_can_network_connect off  
htpd_can_network_connect_cobbler off  
htpd_can_network_connect_db off  
htpd_can_network_memcache off  
htpd_can_network_relay off  
htpd_can_sendmail off  
htpd_dbus_avahi off  
htpd_dbus_sssd off  
htpd_dontaudit_search_dirs off  
htpd_enable_cgi on  
htpd_enable_ftp_server off  
htpd_enable_homedirs off
```

Figure 0.4: Переключатели selinux

3. Утилитой `seinfo` выводим статистику по политике, множество пользователей, ролей и типов (иллюстр. 0.5, 0.6, 0.7, 0.8).

```
[root@eaabakumov ~]# seinfo

Statistics for policy file: /sys/fs/selinux/policy
Policy Version & Type: v.31 (binary, mls)

Classes:          130      Permissions:        272
Sensitivities:    1        Categories:        1024
Types:            4793     Attributes:         253
Users:            8        Roles:             14
Booleans:         316     Cond. Expr.:       362
Allow:            107834   Neverallow:         0
Auditallow:       158     Dontaudit:          10022
Type_trans:       18153   Type_change:        74
Type_member:       35     Role_allow:         37
Role_trans:       414     Range_trans:        5899
Constraints:      143     Validatetrans:      0
Initial SIDs:     27      Fs_use:             32
Genfscon:         103     Portcon:            614
Netifcon:         0       Nodecon:            0
Permissives:      0       Polcap:             5
```

Figure 0.5: Статистика по политике

```
[root@eaabakumov ~]# seinfo -u

Users: 8
  sysadm_u
  system_u
  xguest_u
  root
  guest_u
  staff_u
  user_u
  unconfined_u
[root@eaabakumov ~]# █
```

Figure 0.6: Множество пользователей

```
[root@eaabakumov ~]# seinfo -r  
  
Roles: 14  
    auditadm_r  
    dbadm_r  
    guest_r  
    staff_r  
    user_r  
    logadm_r  
    object_r  
    secadm_r  
    sysadm_r  
    system_r  
    webadm_r  
    xguest_r  
    nx_server_r  
    unconfined_r  
[root@eaabakumov ~]#
```

Figure 0.7: Множество ролей

```
[root@eaabakumov ~]# seinfo -t
Types: 4793
    bluetooth_conf_t
    cmirrord_exec_t
    colord_exec_t
    container_auth_t
    foghorn_exec_t
    jacobd_port_t
    pki_ra_exec_t
    pki_ra_lock_t
    sosreport_t
    squid_script_exec_t
    etc_runtime_t
    fenced_tmp_t
    git_session_t
    glance_port_t
    osad_log_t
    presence_port_t
    samba_secrets_t
    snort_exec_t
    sshd_sandbox_t
```

Figure 0.8: Множество типов

4. Проверяем тип файлов и поддиректорий, находящихся в директории `/var/www`. Так как в директории лежат только поддиректории, их типы: `httpd_sys_script_exec_t` и `httpd_sys_content_t`. Пытаемся определить тип файлов в `/var/www/html`. Так как там нет файлов, определять нечего. Определяем круг пользователей, которым разрешено создание файлов в директории `/var/www/html`. Так как пользователь для директории установлен в `system_u`, создавать файлы там может только суперпользователь (иллюстр. 0.9).

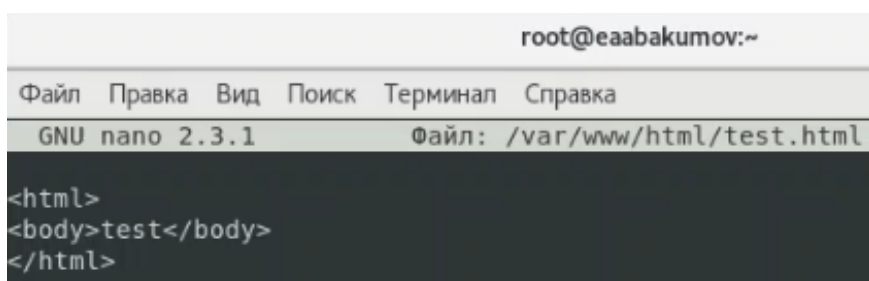
```

[root@eaabakumov ~]# ls -lZ /var/www
drwxr-xr-x. root root system_u:object_r:httpd_sys_script_exec_t:s0 cgi-bin
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 html
[root@eaabakumov ~]# ls -lZ /var/www/html
[root@eaabakumov ~]# ls -lZ /var/www
drwxr-xr-x. root root system_u:object_r:httpd_sys_script_exec_t:s0 cgi-bin
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 html
[root@eaabakumov ~]#

```

Figure 0.9: Проверка типов в /var/www

5. Из-под рута создаем файл в /var/www/html/ (иллюстр. 0.10). Проверяем контекст: `unconfined_u:object_r:httpd_sys_content_t:s0` (иллюстр. 0.11). Проверяем доступность файла в браузере (иллюстр. 0.12).



The screenshot shows the nano text editor interface. At the top, it says "root@eaabakumov:~". Below the menu bar, it indicates "GNU nano 2.3.1" and "Файл: /var/www/html/test.html". The editor content shows the following HTML structure:

```

<html>
<body>test</body>
</html>

```

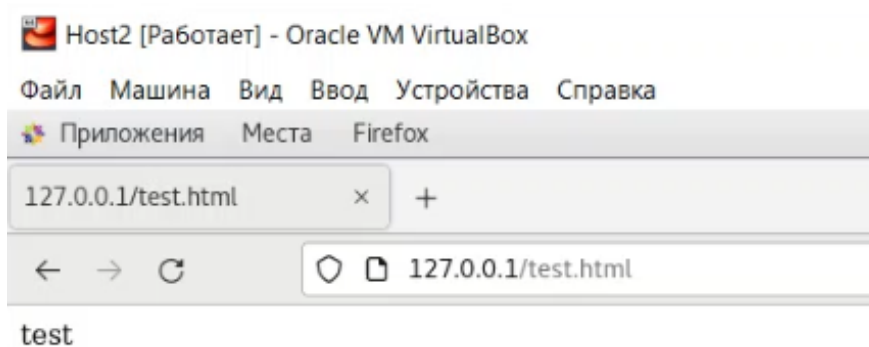
Figure 0.10: Файл /var/www/html/test.html

```

[root@eaabakumov ~]# ls -lZ /var/www/html/
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 test.html
[root@eaabakumov ~]#

```

Figure 0.11: Контекст файла test.html



The screenshot shows a web browser window titled "Host2 [Работает] - Oracle VM VirtualBox". The address bar shows "127.0.0.1/test.html". The page content displays the word "test".

Figure 0.12: Проверка файла в браузере

6. Изучив контексты файлов для httpd_selinux приходим к выводу, что контекст был выбран верно. Теперь изменим контекст (иллюстр. 0.13). Теперь браузер не имеет доступа к файлу, выдается ошибка (иллюстр. 0.14). Файл не был отображен, так как процесс httpd не имеет доступа к выбранному нами типу файла. Просматриваем логи, где говорится, что selinux запретил доступ из-за разницы в контекстах (иллюстр. 0.15).

```
[root@eaabakumov ~]# ls -lZ /var/www/html/
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 test.html
[root@eaabakumov ~]# chcon -t samba_share_t /var/www/html/test.html
[root@eaabakumov ~]# ls -lZ /var/www/html/test.html
-rw-r--r--. root root unconfined_u:object_r:samba_share_t:s0 /var/www/html/test.html
[root@eaabakumov ~]#
```

Figure 0.13: Изменение контекста

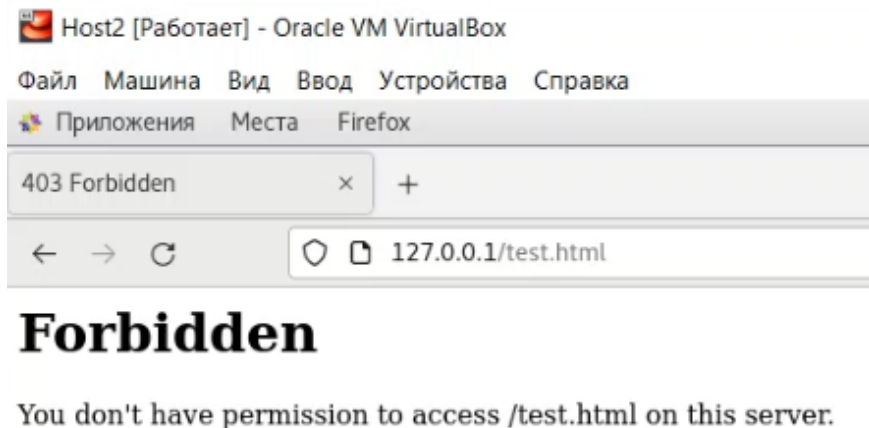
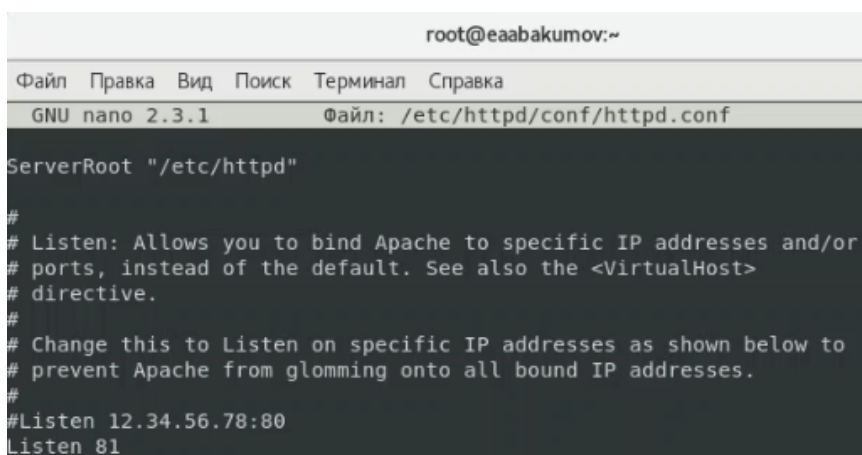


Figure 0.14: Ошибка доступа в браузере

```
root@eaabakumov:~
Файл Правка Вид Поиск Терминал Справка
-rw-r--r--. 1 root root 33 ноя 23 13:12 /var/www/html/test.html
[root@eaabakumov ~]# tail /var/log/messages
Nov 23 13:13:14 eaabakumov rtkit-daemon[692]: Successfully made thread 4139 of process 3875 (/usr/lib64/firefox/firefox) owned by '1000' RT at priority 1
Nov 23 13:20:01 eaabakumov systemd: Started Session 6 of user root.
Nov 23 13:20:50 eaabakumov dbus[727]: [system] Activating service name='org.fedoraproject.Setroubleshootd' (using servicehelper)
Nov 23 13:20:50 eaabakumov dbus[727]: [system] Successfully activated service 'org.fedoraproject.Setroubleshootd'
Nov 23 13:20:58 eaabakumov setroubleshoot: failed to retrieve rpm info for /var/www/html/test.html
Nov 23 13:20:58 eaabakumov setroubleshoot: SELinux is preventing /usr/sbin/httpd from getattr access on the file /var/www/html/test.html. For complete SE
Linux messages run: sealert -l 46745119-0fe1-4244-b061-9469ae9f24ac
Nov 23 13:20:58 eaabakumov python: SELinux is preventing /usr/sbin/httpd from getattr access on the file /var/www/html/test.html.#012#012***** Plugin re
storecon (92.2 confidence) suggests *****#012#012If you want to fix the label. #012/var/www/html/test.html default label should be h
ttpd_sys_content_t.#012Then you can run restorecon. The access attempt may have been stopped due to insufficient permissions to access a parent directory
in which case try to change the following command accordingly.#012#012# /sbin/restorecon -v /var/www/html/test.html#012#012***** Plugin public conten
t (7.83 confidence) suggests *****#012#012If you want to treat test.html as public content#012Then you need to change the label on test.
html to public content_t or public content_rw_t.#012#012# semanage fcontext -a -t public_content_t '/var/www/html/test.html' #012# restorecon -v '/var/w
ww/html/test.html' #012#012***** Plugin catchall (1.41 confidence) suggests *****#012#012If you believe that httpd should be allow
ed getattr access on the test.html file by default.#012Then you should report this as a bug.#012You can generate a local policy module to allow this acce
ss.#012#012#012allow this access for now by executing:#012# ausearch -c 'httpd' --raw | audit2allow -M my-httpd#012# semodule -i my-httpd.pp#012
Nov 23 13:21:01 eaabakumov setroubleshoot: failed to retrieve rpm info for /var/www/html/test.html
Nov 23 13:21:01 eaabakumov setroubleshoot: SELinux is preventing /usr/sbin/httpd from getattr access on the file /var/www/html/test.html. For complete SE
Linux messages run: sealert -l 46745119-0fe1-4244-b061-9469ae9f24ac
```

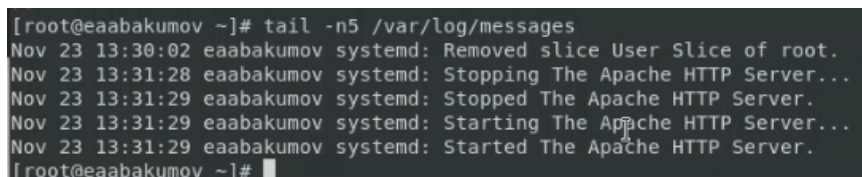
Figure 0.15: Логи

7. Меняем порт прослушивания `httpd` с 80 на 81 (иллюстр. 0.16). Пытаемся перезапустить демона, он перезапускается, так как 81 порт прописан по умолчанию в политике. В логах никаких ошибок не видим (иллюстр. 0.17). Выполняем добавление порта, на что `selinux` отказывается это делать, поскольку порт уже прописан, проверяем это в списке (иллюстр. 0.18). Собственно, поэтому демон и перезапустился, хотя по задумке лабораторной работы не должен был, так как процесс попытался бы получить доступ к запрещенному порту, на что `selinux` не дал бы разрешения, а `httpd`, не стерпев такого обращения с собой, отказался бы запускаться, и только прописывание порта в `selinux` позволило бы выполнить перезапуск. Короче говоря, возвращаем верный контекст на файл `test.html` (иллюстр. 0.19) и проверяем его через 81-ый порт браузером (иллюстр. 0.20). Удаляем привязку порта, `selinux` отвечает, что порт прописан в политике, так что удаляться он не будет. Ну, мы его туда и не добавляли, так что кто мы такие, чтобы удалять его оттуда. Удаляем файл `/var/www/html/test.html`.



```
root@eaabakumov:~  
Файл  Правка  Вид  Поиск  Терминал  Справка  
GNU nano 2.3.1    Файл: /etc/httpd/conf/httpd.conf  
ServerRoot "/etc/httpd"  
#  
# Listen: Allows you to bind Apache to specific IP addresses and/or  
# ports, instead of the default. See also the <VirtualHost>  
# directive.  
#  
# Change this to Listen on specific IP addresses as shown below to  
# prevent Apache from glomming onto all bound IP addresses.  
#  
#Listen 12.34.56.78:80  
Listen 81
```

Figure 0.16: Замена порта прослушивания



```
[root@eaabakumov ~]# tail -n5 /var/log/messages  
Nov 23 13:30:02 eaabakumov systemd: Removed slice User Slice of root.  
Nov 23 13:31:28 eaabakumov systemd: Stopping The Apache HTTP Server...  
Nov 23 13:31:29 eaabakumov systemd: Stopped The Apache HTTP Server.  
Nov 23 13:31:29 eaabakumov systemd: Starting The Apache HTTP Server...  
Nov 23 13:31:29 eaabakumov systemd: Started The Apache HTTP Server.  
[root@eaabakumov ~]#
```

Figure 0.17: Отсутствие ошибок в логах

```
[root@eaabakumov ~]# semanage port -a -t http_port_t -p tcp 81
usage: semanage [-h]
               {import,export,login,user,port,ibpkey,ibendport,interface,module
               ,node,fcontext,boolean,permissive,dontaudit}
               ...
semanage: error: unrecognized arguments: -p 81
[root@eaabakumov ~]# semanage port -l | grep http_port_t
http_port_t      tcp      80, 81, 443, 488, 8008, 8009, 8443, 9000
pegasus_http_port_t  tcp      5988
[root@eaabakumov ~]#
```

Figure 0.18: Список портов httpd в selinux

```
[root@eaabakumov ~]# chcon -t httpd_sys_content_t /var/www/html/test.html
[root@eaabakumov ~]# ls -Z /var/www/html/test.html
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html
/test.html
[root@eaabakumov ~]#
```

Figure 0.19: Возвращение верного контекста на файл test.html

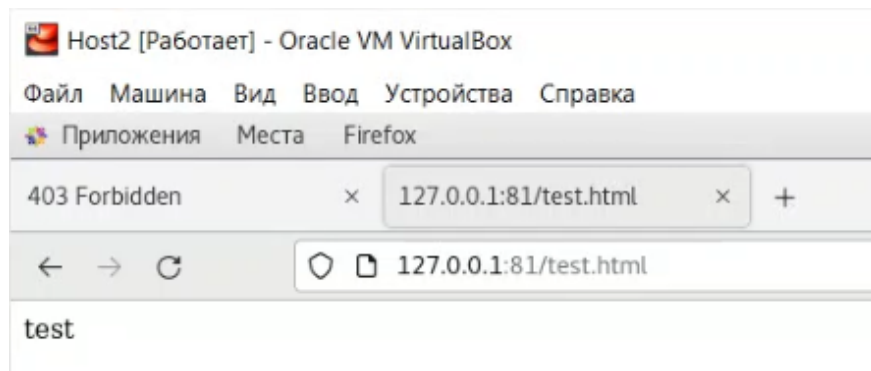


Figure 0.20: Проверка доступа к файлу браузером через 81-ый порт

Выводы

В ходе работы мы успешно развили навыки администрирования ОС Linux, получили первое практическое знакомство с технологией SELinux и проверили работу SELinux на практике совместно с веб-сервером Apache.

Список литературы

1. НАСТРОЙКА SELINUX. // Losst. 2021. URL: <https://losst.ru/nastrojka-selinux> (дата обращения 23.11.2021).
2. Введение в SELinux под CentOS Stream // RUVDS. 2021. URL: <https://ruvds.com/ru/helpcenter/v-selinux-pod-centos-stream/> (дата обращения 23.11.2021).
3. seinfo (1) // OpenNET. 2021. URL: <https://www.opennet.ru/man.shtml?topic=seinfo&category=1> (дата обращения 23.11.2021).
4. Д. С. Кулябов, А. В. Королькова, М. Н. Геворкян. Информационная безопасность компьютерных сетей: лабораторные работы. // Факультет физико-математических и естественных наук. М.: РУДН, 2015. 64 с..