



0,00

Рейтинг

King Servers

Хостинг-провайдер «King Servers»



itNews 22 января 2014 в 16:10

SELinux — описание и особенности работы с

Блог компании King Servers



Коллеги, в первой части статьи о SELinux мы рассмотрели основные особенности работы с системой. Теперь публикуем вторую часть, в которой основное внимание уделено настройке политик SELinux.

Реклама

Индивидуальная настройка политик SELinux

Незначительные изменения в политиках SELinux можно проводить и без полного изменения самой политики. Для этого достаточно модифицировать логические значения, связанные с дополнительными функциями, определенными в политике. Эти функции позволяют, например, предоставлять доступ к домашним каталогам пользователей при помощи Samba или позволять Apache использовать файлы, находящиеся в домашнем каталоге.

По умолчанию, эти функции отключены. Список этих функций заранее предопределен и состоит из наиболее часто используемых задач, встающих перед системным администратором.

Для того, чтобы просмотреть все доступные для вашей системы функции, достаточно выполнить следующую команду:

```
# getsebool -a
```

Для того, чтобы изменить какой-либо параметр, нужно использовать команду setsebool. Например, для того, чтобы разрешить модулям и скриптам сервиса HTTPD подключаться к сети, достаточно ввести в консоли следующее:

```
# setsebool -P httpd_can_network_connect on
```

Создание пользовательских политик при помощи audit2allow

Иногда возникают ситуации, когда заранее определенных функций не хватает, когда нужно дополнить существующую политику новым модулем, вручну прописав те или иные условия предоставления доступа к чему-либо. Например, мы устанавливаем дополнение Postgrey для нашего почтового сервера, работающего по протоколу SMTP. Наш сервер должен взаимодействовать с Postgrey через Unix-сокеты, но стандартная политика SELinux для почтового сервера не позволяет этого сделать, блокирует попытки взаимодействовать через сокеты.

В подобных ситуациях не поможет изменение контекста файлов, нет дополнительных функций, включив которые, мы могли бы исправить ситуацию. Конечно, всегда можно отключить SELinux для «проблемного» сервиса, но это решение, конечно, далеко не идеально, ведь существует ненулевая вероятность того, что почтовый сервер когда-то будет взломан.

Итак, переведем SELinux в режим Permissive, после чего запустим почтовый сервер. Через некоторое время в журнале SELinux появятся AVC-сообщения, в которых будут зафиксированы все недопустимые действия нашего сервера:

```
type=AVC msg=audit(1218128130.653:334): avc: denied { connectto } for pid=9111 comm="smtpd" path="/var/spool/postfix/postgrey/socket"
scontext=system_u:system_r:postfix_smtpd_t:s0 tcontext=system_u:system_r:initrc_t:s0 tclass=unix_stream_socket
type=AVC msg=audit(1218128130.653:334): avc: denied { write } for pid=9111 comm="smtpd" name="socket" dev=sda6 ino=39977017
scontext=system_u:system_r:postfix_smtpd_t:s0 tcontext=system_u:object_r:postfix_spool_t:s0 tclass=sock_file
```

Теперь мы можем использовать утилиту audit2allow для того, чтобы сгенерировать набор правил для локальной политики, разрешающей все необходимые Postgrey действия:

```
# grep smtpd_t /var/log/audit/audit.log | audit2allow -m postgreylocal > postgreylocal.te
# cat postgreylocal.te
module postgreylocal 1.0;
require {
    type postfix_smtpd_t;
    type postfix_spool_t;
    type initrc_t;
    class sock_file write;
    class unix_stream_socket connectto;
}
#===== postfix_smtpd_t =====
allow postfix_smtpd_t initrc_t:unix_stream_socket connectto;
allow postfix_smtpd_t postfix_spool_t:sock_file write;
```

Итак, мы видим, что фильтруется файл audit.log, из которого вычленяются все недопустимые, с точки зрения текущей политики SELinux действия, производимые Postgrey. Просмотрев эти действия, мы видим, что SMTP-сервер пытается создать соединение при помощи Unix-сокета, а Postgrey пытается прослушивать этот сокет. Кажется вполне логичным взять эту информацию и создать на ее основе пользовательский модуль для политики SELinux, который разрешил бы эти действия:

```
# grep smtpd_t /var/log/audit/audit.log | audit2allow -M postgreylocal
```

Теперь мы должны загрузить этот модуль, дополнив им уже задействованные политики при помощи команды semodule:

```
# semodule -i postgreylocal.pp
```

После этого модуль перемещается в /etc/selinux/targeted/modules/active/modules/postgreylocal.pp.

Для того, чтобы проверить, корректно ли загружен модуль, можно вывести список всех загруженных модулей при помощи команды «semodule -l».

После этого мы можем продолжить наблюдение за журналом SELinux, чтобы убедиться в том, что наша только что созданная политика не ограничивает Postgrey. Как только мы будем довольны и уверены в правильной работе политики, мы можем снова активировать режим Enforcing, в полной уверенности, что теперь наш почтовый сервер надежно защищен и, в то же время, полноценно функционирует.

Ручная настройка модулей для политик SELinux.

Adit2allow, без сомнения, отлично справляется с созданием моделей для политик, которые решают какую-то конкретную проблему. Но иногда и эта утилита срабатывает не совсем верно, так что приходится настраивать модуль вручную. Например, рассмотрим записи в AVC-журнале SELinux:

Summary:

SELinux is preventing postdrop (postfix_postdrop_t) "getattr" to /var/log/httpd/error_log (httpd_log_t).

Detailed Description:

SELinux denied access requested by postdrop. It is not expected that this access is required by postdrop and this access may signal an intrusion attempt. It is also possible that the specific version or configuration of the application is causing it to require additional access.

Allowing Access:

Sometimes labeling problems can cause SELinux denials. You could try to restore the default system file context for /var/log/httpd/error_log, `restorecon -v '/var/log/httpd/error_log'`

If this does not work, there is currently no automatic way to allow this access. Instead, you can generate a `local` policy module to allow this access - see FAQ (<http://fedora.redhat.com/docs/selinux-faq-fc5/#id2961385>) Or you can disable SELinux protection altogether. Disabling SELinux protection is not recommended. Please file a bug report (http://bugzilla.redhat.com/bugzilla/enter_bug.cgi) against this package.

Additional Information:

Source Context	system_u:system_r:postfix_postdrop_t
Target Context	root:object_r:httpd_log_t
Target Objects	/var/log/httpd/error_log [file]
Source	postdrop
Source Path	/usr/sbin/postdrop
Port	<Unknown>
Host	sanitized
Source RPM Packages	postfix-2.3.3-2
Target RPM Packages	
Policy RPM	selinux-policy-2.4.6-137.1.el5
Selinux Enabled	True
Policy Type	targeted
MLS Enabled	True
Enforcing Mode	Enforcing
Plugin Name	catchall_file
Host Name	sanitized
Platform	Linux sanitized 2.6.18-53.1.21.el5 #1 SMP Tue May 20 09:35:07 EDT 2008 x86_64 x86_64
Alert Count	599
First Seen	Wed Jul 2 08:27:15 2008
Last Seen	Sun Aug 10 22:47:52 2008
Local ID	c303a4ea-8e7a-4acc-9118-9cc61c6a2ec8
Line Numbers	

Raw Audit Messages

```
host=sanitized type=AVC msg=audit(1218397672.372:352): avc: denied { getattr } for pid=4262 comm="postdrop"
path="/var/log/httpd/error_log" dev=md2 ino=117005 scontext=system_u:system_r:postfix_postdrop_t:s0
tcontext=root:object_r:httpd_log_t:s0 tclass=file
host=sanitized type=SYSCALL msg=audit(1218397672.372:352): arch=c000003e syscall=5 success=no exit=-13 a0=2
a1=7fffd6febca0 a2=7fffd6febca0 a3=0 items=0 ppid=4261 pid=4262 auid=4294967295 uid=48 gid=48 euid=48 suid=48
```

```
fsuid=48 egid=90 sgid=90 fsgid=90 tty=(none) comm="postdrop" exe="/usr/sbin/postdrop"  
subj=system_u:system_r:postfix_postdrop_t:s0 key=(null)
```

После того, как мы запустим `audit2allow` и рассмотрим получившуюся в результате политику `postfixlocal.te`, мы увидим следующее:

```
# grep postdrop /var/log/audit/audit.log | audit2allow -M postfixlocal  
# cat postfixlocal.te  
module postfixlocal 1.0;  
require {  
    type httpd_log_t;  
    type postfix_postdrop_t;  
    class dir getattr;  
    class file { read getattr };  
}  
#===== postfix_postdrop_t =====  
allow postfix_postdrop_t httpd_log_t:file getattr;
```

Сразу же возникает вопрос, зачем PostDrop пытается получить доступ к `/var/log/httpd/error_log`? Это не то действие, которое мы могли бы ожидать от этой программы, так что теперь только нам решать позволять это действие или нет.

У нас есть несколько путей решения этой проблемы.

— Мы можем игнорировать эту ошибку и позволить SELinux блокировать доступ к файлу.

— Мы можем позволить это действие, создав соответствующий модуль политики при помощи `audit2allow`.

— Мы можем вручную отредактировать файл этого модуля, чтобы определить нужную нам реакцию SELinux на попытку доступа к файлу. Например, мы можем запретить аудит этого события, блокируя в тоже время доступ. Для этого мы должны изменить значение «allow» в соответствующей строке на «dontaudit»:

```
#===== postfix_postdrop_t =====  
dontaudit postfix_postdrop_t httpd_log_t:file getattr;
```

Теперь мы должны вручную скомпилировать и загрузить отредактированный модуль политики:

```
# checkmodule -M -m -o postfixlocal.mod postfixlocal.te  
# semodule_package -o postfixlocal.pp -m postfixlocal.mod  
# semodule -i postfixlocal.pp
```

Таким образом доступ к файлу `/var/log/httpd/error_log` блокируется, но мы не получаем постоянных предупреждений об этом от SELinux.

Собственно, по SELinux пока все, а в последующих статьях мы рассмотрим такую интересную (и, надеемся, полезную) тему, как дисковые квоты в linux для grm-дистрибутивов. Новая статья будет опубликована уже в понедельник.

Теги: selinux, политики

Хабы: Блог компании King Servers

↑ +43 ↓ 355 👁 33,5k 💬 7 ➦ Поделиться



@itNews

Сетевые технологии и оборудование, гаджеты



King Servers

Хостинг-провайдер «King Servers»

ПОХОЖИЕ ПУБЛИКАЦИИ

20 января 2014 в 12:51

SELinux – описание и особенности работы с системой. Часть 1

↑ +72

👁 169k

📖 689

💬 13



Реклама

Комментарии 7



iaf 22 января 2014 в 18:35



+11



Извините, не удержусь в очередной раз, ибо безопасность не терпит подобных «советов», которые будут потом много раз бездумно скопипащены. Совет для тех, кто будет основываться на статье:

Никогда не делайте сбор собственного модуля из логов AVC.

Даже если вы считаете что вы досконально знаете все, что делает ПО (это не так).

Даже если вы его написали (иначе зачем люди придумали тесты).

Подобный подход равносителен поиску всех отброшенных пакетов в логе фаервола с последующим автоматическим созданием разрешающих правил.

Для тех, кто еще не понял — что окажется в логе AVC, если обычный пользователь запросит запись в /etc/shadow из программы с именем smtpd_t? Останется только подождать, пока любезный «системный администратор» разрешит этот доступ :-)

А теперь, поясню на примере статьи:

[AVC из статьи](#)

[Что делает автор](#)

[Что это на самом деле](#)

Вот таким простым и незамысловатым образом вы открыли достаточно серьезную дыру в ваших политиках, и более того, вы никогда о ней не узнаете при штатной работе SELinux. Тут не только initrc_t под угрозой, тут еще и race condition на *любого* демона во все поля.

А причина ошибки всего лишь заключалась в том, что у сокета был неправильный контекст, скорее всего наследованный от инита и не смененный во время domain_transition. Хотя даже в полугодовой давности RefPolicy явно прописано:

```
/var/spool/postfix/postgrey(/.*)?    gen_context(system_u:object_r:postgrey_spool_t,s0)
manage_sock_files_pattern(postgrey_t, postgrey_spool_t, postgrey_spool_t)
```



onix74 22 января 2014 в 20:49



+2



Спасибо за предостережение! Новичкам в selinux они очень кстати.




grossws 23 января 2014 в 02:21







+1



Комментарий полезнее статьи. Спасибо. =))

-  zlyoha

23 января 2014 в 11:30




↑

0





↓

Как новичку, можете об этом поподробнее?

|

race condition на любого демона во все поля
-  iaf


23 января 2014 в 14:41







↑

+1

↓

Описание процесса, например, тут.
-  Mushtat


23 января 2014 в 12:51





↑

-1

↓

Интересно, что автор никак не прокомментировал Ваше сообщение.
-  bitfroster

22 января 2014 в 18:35



↑

0

↓

Спасибо, некоторые вещи для меня стали откровением. Жду продолжения.

Только полноправные пользователи могут оставлять комментарии. Войдите, пожалуйста.

САМОЕ ЧИТАЕМОЕ


- Сутки
- Неделя
- Месяц

- SARS нерукотворный? Генеалогия уханьского коронавируса
- ↑


+260


 86,2k


 203

 333
- Вы не хотите усиливать иммунитет. Или крайности организма человеческого
- ↑

+50

 23,6k


 74

 28
- Как Amazon пытается заставить людей покупать... меньше
- ↑


+29


 18,5k


 14

 23
- Современный самолёт by design защищён от биологической угрозы (COVID-19) лучше, чем вы думаете
- ↑


+56


 14,6k


 35

 64
- На картах Ростова, Санкт-Петербурга и других городов России начались виртуальные митинги
- ↑

+26

 13k

 8

 42

Ваш аккаунт	Разделы	Информация	Услуги
Войти	Публикации	Устройство сайта	Реклама
Регистрация	Новости	Для авторов	Тарифы
	Хабы	Для компаний	Контент
	Компании	Документы	Семинары

[Пользователи](#)[Соглашение](#)[Мегапроекты](#)[Песочница](#)[Конфиденциальность](#)

Если нашли опечатку в посте, выделите ее и нажмите Ctrl+Enter, чтобы сообщить автору.

