مکانیسم‌های کنترل دسترسی (ACM ها)

این قسمت مقدمه‌ای برای مکانیسم کنترل دسترسی (ACM) ارائه می‌دهد. ACM ها یک وسیله برای مدیران سیستم برای کنترل اینکه کدام کاربر و فرآیند می‌توانند به فایل‌ها، دستگاه‌ها، رابط‌ها و غیره در یک سیستم کامپیوتری دسترسی داشته باشند، فراهم می‌کنند. این یک نگاه اولیه به امنیت یک سیستم کامپیوتری یا شبکه‌ای با هر اندازه و مقیاسی می‌باشد.

کنترل دسترسی اختیاری (DAC)

کنترل دسترسی اختیاری (DAC) کنترل‌های دسترسی اولیه برای اشیاء در یک سیستم فایل را تعریف می‌کند. این کنترل دسترسی معمولی است که توسط مجوزهای فایل، به اشتراک‌گذاری و غیره ارائه می‌شود. چنین نوع دسترسی عموماً در اختیار صاحب شیء (فایل، دایرکتوری، دستگاه و غیره) است.

DAC یک وسیله محدود کردن دسترسی به اشیا بر اساس هویت کاربر یا گروه (موضوع) است که سعی در دسترسی به آن اشیاء را فراهم می‌کند. بسته به مجوزهای دسترسی مورد نظر، ممکن است بتوانیم مجوزها را به سایر افراد انتقال دهیم.

DAC همان پیکربندی permission ها می‌باشد که در ابتدای کتاب آن را معرفی کردیم و نحوه کار با آن را به صورت کامل شرح دادیم.

لیست‌های کنترل دسترسی (ACLs)

لیست‌های کنترل دسترسی (ACL ها) کنترل بیشتری را برای مواردی که یک شیء می‌تواند به آن دسترسی پیدا کند، کنترل می‌کنند. کار با ACLها را به‌صورت کامل در بخش های قبلی معرفی نمودیم.

کنترل دسترسی اجباری (MAC/ Mandatory Access Control)

کنترل دسترسی اجباری (MAC) مکانیسم امنیتی است که سطح کنترلی را که کاربران بر روی اشیاء ایجاد می‌کنند محدود می‌کند. برخلاف پیاده‌سازی DAC که کاربران کنترل کاملی از فایل‌ها، پوشه‌ها و غیره دارند، MAC برچسب‌ها یا دسته‌های بیشتری را به همه اشیاء سیستم فایل اضافه می‌کند. کاربران و فرآیندها قبل از اینکه بتوانند با این اشیاء ارتباط برقرار کنند، باید دسترسی مناسب به این دسته‌ها داشته باشند.

در Linux، MAC توسط SELinux اعمال می‌شود.

کنترل دسترسی مبتنی بر نقش (RBAC)

کنترل دسترسی مبتنی بر نقش (RBAC) یک روش جایگزین برای کنترل دسترسی کاربر به اشیاء سیستم فایل است. به‌جای اینکه دسترسی توسط مجوزهای کاربر کنترل شود، مدیر سیستم نقش‌ها را بر اساس الزامات عملکردی تجاری یا معیارهای مشابه ایجاد می‌کند. این نقش‌ها دارای انواع و سطوح دسترسی به اشیاء هستند.

در مقایسه با سیستم‌های DAC یا MAC که در آن کاربران به اشیاء بر اساس مجوزهای خود و مجوز دسترسی، دسترسی دارند، کاربران در

یک سیستم RBAC قبل از اینکه بتوانند با فایل‌ها، دایرکتوری‌ها یا دستگاه‌ها، ارتباط برقرار کنند، باید عضو گروه مناسب یا نقش باشند.

از دیدگاه مدیریتی، این امر کنترل افراد مجاز برای دسترسی به قسمت‌های مختلف سیستم فایل را آسان‌تر می‌کند.

امنیت چند سطحی (MLS)

امنیت چند سطحی (MLS) یک برنامه امنیتی ویژه اجباری (MAC) است. تحت این طرح، فرایندها با نام Subjects بیان می‌شوند. فایل‌ها، سوکت‌ها و دیگر موجودیت‌های سیستم‌عامل Objects نامیده می‌شوند.

امنیت چند طبقه (MCS)

امنیت چند طبقه (MCS) یک مورد بسیار پیشرفته در SELinux است و به کاربران اجازه می‌دهد تا فایل‌ها را با دسته‌بندی‌ها برچسب‌گذاری کنند. MCS یک سازگاری از MLS است و مجدداً از چارچوب MLS در SELinux استفاده می‌کند.

مقدمه‌ای بر SELinux

SELinux یک معماری امنیتی است که در هسته 2.6.*x* با استفاده از ماژول‌های امنیت لینوکس (LSM) یکپارچه شده است. SELinux یک پروژه آژانس امنیت ملی ایالات‌متحده (NSA) و جامعه SELinux است. ادغام SELinux در لینوکس Red Hat Enterprise یک تلاش مشترک بین NSA و Red Hat بوده است.

بررسی اجمالی SELinux

SELinux یک سیستم کنترل دسترسی (MAC) است که در هسته لینوکس ساخته شده است. تحت استاندارد کنترل دسترسی اختیاری لینوکس (DAC)، یک برنامه یا فرایند که به‌عنوان کاربر اجرا می‌شود (UID یا SUID) دارای مجوز کاربر برای اشیاء مانند فایل‌ها، سوکت‌ها و سایر فرایندها است. اجرای یک هسته MAC سیستم را از برنامه‌های مخرب یا ناقض امنیت که می‌توانند به سیستم آسیب برسانند یا نابود کنند محافظت می‌کند.

SELinux مجوزهای دسترسی و انتقال هر کاربر، برنامه، پردازش و فایل را در سیستم تعریف می‌کند. سپس SELinux تعاملات این موجودیت‌ها را با استفاده از یک سیاست امنیتی تعریف می‌کند.

کاربران معمولی سیستم از SELinux بی‌اطلاع هستند. فقط مدیران سیستم باید در نظر بگیرند که چه سیاست امنیتی را برای محیط سرورشان پیاده‌سازی نمایند. این سیاست می‌تواند به اندازه کافی سخت باشد یا مورد نیاز باشد و به صورت بسیار دقیق تعریف شود.

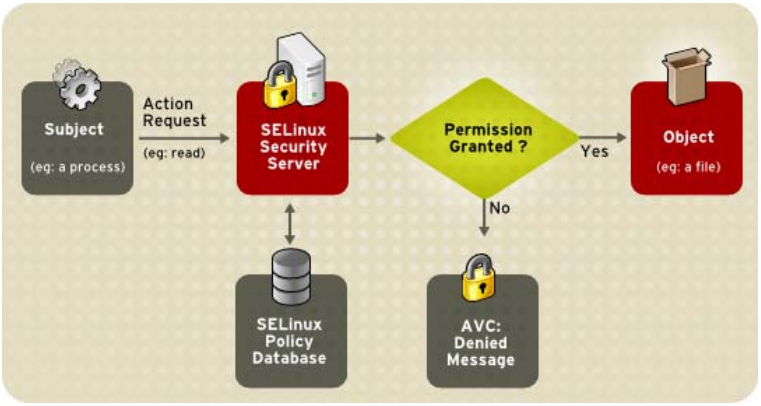
کل هسته SELinux، کنترل کاملی بر روی سیستم دارد.

فرایند تصمیم‌گیری SELinux

هنگامی که یک موضوع (به‌عنوان مثال یک برنامه)، برای دسترسی به یک شیء (به عنوان مثال یک فایل) تلاش می‌کند، سرور اجرای سیاست در هسته کش (cache) برای دسترسی (AVC) را بررسی می‌کند که در آن مجوزهای موضوع و شیء ذخیره می‌شوند. اگر تصمیم‌گیری بر اساس داده‌ها در AVC انجام نشود، ارسال درخواست به سرور امنیتی ادامه می‌یابد که زمینه امنیتی برنامه و فایل را در ماتریس مجوزها بررسی می‌کند. پس از آن مجوز توسط avc اعطاء می‌گردد یا گرفته می‌شود:

پیـــــام denied در صـــــورت غیرمجـــــاز بـــــودن، در فایـــــل /var/log/message، Log می‌شود. زمینه امنیتی موضـوعات و اشـیاء از سیاست نصب شده اعمال می‌شود که همچنین اطلاعاتی را برای پر کردن ماتریس سرور فراهم می‌کند.

به نمودار زیر توجه نمایید:



حالت‌های عملیاتی SELinux

بجـای اجـرا در حالـت enforcing، SELinux مـی‌توانـد در حالـت permissive اجرا شود، جایی که AVC بررسی می‌شود و denial هـا Log می‌شوند. حالت permissive، SELinux سیاست را اجرا نمی‌کند، بلکه می‌تواند برای عیب‌یابی، برای یا تنظـیم سیاسـت SELinux مفیـد باشد.

فایل‌های مربوط به SELinux

در ادامه، فایل‌های پیکربندی SELinux و سیستم فایل‌های مربوطه را بررسی می‌نماییم.

سیستم فایل SELinux

سیستم pseudo-file /selinux/ حاوی دستوراتی است که اغلب توسط زیرسیستم هسته استفاده می‌شود. این نوع فایل سیستم شبیه به سیستم pseudo-file /proc/ است.

مدیران و کاربران به‌طور معمول نیازی به دست‌کاری این جزء ندارند.

مثال زیر محتویات نمونه‌ای از دایرکتوری /selinux/ را نشان می‌دهد:

```
-rw-rw-rw-  1 root root 0 Sep 22 13:14 access
dr-xr-xr-x  1 root root 0 Sep 22 13:14 booleans
--w-------  1 root root 0 Sep 22 13:14
commit_pending_bools
-rw-rw-rw-  1 root root 0 Sep 22 13:14 context
-rw-rw-rw-  1 root root 0 Sep 22 13:14 create
--w-------  1 root root 0 Sep 22 13:14 disable
-rw-r--r--  1 root root 0 Sep 22 13:14 enforce
-rw-------  1 root root 0 Sep 22 13:14 load
-r--r--r--  1 root root 0 Sep 22 13:14 mls
-r--r--r--  1 root root 0 Sep 22 13:14 policyvers
```

-rw-rw-rw- 1 root root 0 Sep 22 13:14 relabel

-rw-rw-rw- 1 root root 0 Sep 22 13:14 user

برای مثال، در اجرای دستور cat بر روی فایل enforce، ۱ برای اجرای حالت enforcing و ۰ برای اجرای حالت permissive می‌باشد.

فایل‌های پیکربندی SELinux

در بخش‌های زیر فایل‌های پیکربندی، سیاست‌های SELinux و سیستم فایل‌های موجود در پوشه etc/ شرح داده می‌شوند.

فایل پیکربندی /etc/sysconfig/selinux

دو راه برای پیکربندی SELinux درونLinux وجود دارد: استفاده از ابزار system-config-selinux یا ویرایش فایل پیکربندی /etc/sysconfig/selinux به‌صورت دستی.

فایل /etc/sysconfig/selinux فایل اصلی پیکربندی برای فعال یا غیرفعال کردن SELinux است. از این فایل برای تعیین اینکه کدام سیاست برای اجرای سیستم مناسب است و همچنین مشخص کردن نحوه اجرای آن استفاده می‌کنیم.

نکته: /etc/sysconfig/selinux حاوی لینک نمادین به فایل پیکربندی واقعی /etc/selinux/config است.

در زیر شرح کامل زیرمجموعه‌ای از گزینه‌های موجود برای پیکربندی بیان شده است:

- SELINUX = enforcing | permissive | disabled
وضعیت سطح بالای SELinux را بر روی یک سیستم تعریف می‌کند.

- enforcing - سیاست امنیتی SELinux اعمال می‌شود.
- permissive- سیستم SELinux هشدارها را چاپ می‌کند اما سیاست را اجرا نمی‌کند.

این مورد برای مقاصد اشکال‌زدایی و عیب‌یابی مفید است. در حالت permissive، denialهای بیشتری ثبت می‌شوند، زیرا subject ها می‌توانند با actionها ادامه یابند، در صورتی که در حالت enforcing این امر امکان‌پذیر نیست. برای مثال، پیمایش یک درخت دایرکتوری در حالت permissive، avc زیر را ایجاد می‌کند:

denied messages for every directory level read.

در حالت اجرای enforcing، SELinux مانع از پیمایش می‌شود و جلوی تولید پیغام‌های بیشتر را می‌گیرد.

- SELinux - disabled به‌طور کامل غیرفعال می‌شود. SELinux از هسته جدا شده و سیستم pseudo-file رجیستر نشده است.

- SELINUXTYPE = target | strict - مشخص کردن سیاست SELinux که باید اجرا (enforce) شود.
  - targeted- تنها دایمون‌های مورد هدف شبکه محافظت می‌شوند.

    اجرای سیاست برای این daemon ها می‌تواند روشن یا خاموش باشد و با استفاده از مقادیر بولین توسط ابزار پیکربندی سطح امنیتی (system-config-selinux) کنترل شود.

تنظیم مقدار بولی برای targeted daemon به ۰ (صفر) موجب غیرفعال کردن انتقال سیاست برای daemon می‌گردد. به عنوان مثال، شما می‌توانید dhcpd_disable_trans را به ۰ تنظیم کنید تا از انتقال dhcpd از دامنه unconfined_t به دامنه مشخص شده در dhcpd.te در init جلوگیری شود.

از دستور getsebool -a برای لیست کردن تمام SELinux boolean ها استفاده کنید. مثال زیر استفاده از دستور setsebool برای تنظیم یک بولین SELinux است. گزینه P- باعث تغییر دائمی می‌شود. بدون این گزینه، boolean در هنگام راه‌اندازی مجدد به ۱ تنظیم می‌شود.

setsebool -P dhcpd_disable_trans = 0

- strict- حفاظت کامل SELinux، برای تمامی daemonها.

زمینه‌های امنیتی برای همه موضوعات و اشیاء تعریف شده است و هر عمل توسط سرور اجرای سیاست پردازش می‌شود.

- SETLOCALDEFS = 0 | 1 - کنترل تعاریف محلی (کاربر و بولین) را تنظیم می‌کند. مقدار را به ۱ تنظیم کنید تا این تعاریف توسط load_policy از فایل‌ها در /etc/selinux/<polyname> کنترل شود یا آن را به ۰ تنظیم کنید تا با semanage کنترل شود.

دایرکتوری /etc/selinux

دایرکتوری etc/selinux/، دایرکتوری اصلی برای تمام فایل‌ها و همچنین فایل پیکربندی اصلی است.

مثال زیر محتویات نمونه‌ای از پوشه etc/selinux/ را نشان می‌دهد:

```
-rw-r--r--  1 root root  448 Sep 22 17:34 config
drwxr-xr-x  5 root root 4096 Sep 22 17:27 strict
drwxr-xr-x  5 root root 4096 Sep 22 17:28 targeted
```

دو زیر دایرکتوری strict و targeted، دو دایرکتوری خاص هستند که در آن‌ها فایل‌های سیاستی از همان نوع (یعنی strict و targeted) موجود است.

ابزارهای کاربردی SELinux

برخی از ابزارهای کاربردی SELinux که معمولاً استفاده می‌شود، عبارت‌اند از:

- /usr/sbin/setenforce - حالت واقعی در زمان اجرای SELinux را تغییر می‌دهد.

  برای مثال:

  1 setenforce، SELinux در حالت enforcing اجرا می‌شود.

  0 setenforce، SELinux در حالت permissive اجرا می‌شود.

  برای غیرفعال کردن SELinux، شما باید پارامتر setenforce مناسب را در etc/sysconfig/selinux/ مشخص کنید یا

پارامتر selinux = 0 را در /etc/grub.conf یا در زمان بوت شدن به هسته تنظیم کنید.

- /usr/sbin/sestatus -v- وضعیت دقیق یک سیستم SELinux را نمایش می‌دهد. مثال زیر نمونه‌ای از خروجی satatus -v را نشان می‌دهد:

```
SELinux status:                 enabled
SELinuxfs mount:                /selinux
Current mode:                   enforcing
Mode from config file:          enforcing
Policy version:                 21
Policy from config file:        targeted

Process contexts:
Current context:                user_u:system_r:unconfined_t:s0
Init context:                   system_u:system_r:init_t:s0
/sbin/mingetty                  system_u:system_r:getty_t:s0
```

- /usr/bin/newrole - یک پوسته جدید را در یک زمینه یا نقش جدید اجرا می‌کند. سیاست باید اجازه انتقال به نقش جدید را بدهد.

نکته: این دستور فقط در صورتی قابل انجام است که بسته‌های policycoreutils-newrole نصب شده باشند که برای سیاست‌های strict و MLS مورد نیاز است.

- /sbin/restorecon - زمینه امنیتی یک یا چند فایل را با علامت‌گذاری ویژگی‌های توسعه‌یافته با فایل یا زمینه امنیتی مناسب تنظیم می‌کند.

- /sbin/fixfiles- پایگاه داده‌های امنیتی در سیستم فایل را بررسی و اصلاح می‌کند.

امنیت چند طبقه (MCS)

امنیت چند طبقه (MCS) یک قابلیت پیشرفته برای SELinux است و به کاربران اجازه می‌دهد تا فایل‌ها را با دسته‌بندی‌ها برچسب‌گذاری کنند. این دسته‌ها به کنترل دسترسی محدود (DAC) و منطق نوع اجرا (TE) محدود می‌شوند. آن‌ها همچنین می‌توانند هنگام نمایش یا چاپ فایل‌ها استفاده شوند. طبقه "Company_Confidential" مثالی برای این مورد است. فقط کاربران دارای دسترسی به این دسته می‌توانند به فایل‌های برچسب‌گذاری شده با دسته دسترسی پیدا کنند، با فرض اینکه DAC موجود و قوانین TE هم اجازه دسترسی به آن‌ها را داشته باشند.

اصطلاح طبقه‌بندی به دسته‌های غیر سلسله مراتبی استفاده شده توسط چندین سطح امنیتی (MLS) اشاره دارد. تحت MLS، اشیاء و موضوعات با سطوح امنیتی برچسب‌گذاری می‌شوند. این سطوح امنیتی شامل یک مقدار حساسیت سلسله مراتبی (مانند "امنیت بالا") و صفر یا چند دسته غیر سلسله مراتبی (مانند "رمزنگاری") است. دسته‌ها محدوده‌ها را در سطوح حساسیتی فراهم می‌کنند و اصل نیاز به امنیت اطلاعات را اجرا می‌کنند. برای کسب اطلاعات بیشتر در مورد امنیت چند سطحی، به بخش "امنیت چند سطحی (MLS)" مراجعه کنید.

MCS یک انطباق از MLS است. از نظر فنی، MCS یک تغییر سیاست، همراه با تعدادی از اصلاحات کاربردی برای مخفی کردن برخی از تکنولوژی‌های غیرضروری MLS است. برخی از تغییرات هسته نیز ساخته شده است، اما تنها مربوط به ارتقاء به MCS (یا MLS) بدون استفاده از یک فایل سیستم کامل است.

برنامه‌های کاربردی برای امنیت چند طبقه‌ای

فراتر از کنترل دسترسی، MCS می‌تواند برای نمایش دسته‌های MCS در بالا و پایین صفحات چاپ شده مورد استفاده قرار گیرد. این مسئله ممکن است شامل یک پوشش برای نشان دادن روش‌های Handeling فایل باشد. همچنین باید MCS را با تحولات آینده در SELinux، از قبیل Security Enhanced X، ادغام کرد. ادغام با یک سرور دایرکتوری، پشتیبانی از MCS را برای ایمیل آسان‌تر می‌کند. این مورد می‌تواند باعث شود کاربر ایمیل‌های خروجی یا پیوست فایل‌ها را برچسب‌گذاری کند. در این حالت کاربر ایمیل می‌تواند تعیین کند که دریافت‌کنندگان شناسایی شده می‌توانند با توجه به طبقه‌شان به ایمیل مربوطه دسترسی پیدا کنند یا خیر.

محتواهای امنیتی SELinux (SELinux Security Contexts)

SELinux امنیت پرونده‌ها را به عنوان یک ویژگی اضافه از فایل ذخیره می‌کند. واژه امنیت، برای ماژول‌های امنیتی مورد استفاده قرار می‌گیرد و فضای نام security.selinux برای حفظ برچسب‌های امنیتی SELinux در فایل‌ها استفاده می‌شود. محتویات این ویژگی بسته به فایل یا دایرکتوری و خط‌مشی که دستگاه اجرا می‌کند، متفاوت خواهد بود.

شما می‌توانید از دستور ls -Z برای مشاهده برچسب رده یک فایل استفاده کنید:

```
[root@myServer ~]# ls -Z gravityControl.txt

-rw-r--r--  user     user
user_u:object_r:tmp_t:Moonbase_Plans gravityControl.txt
```

شما می‌توانید از دستور gefattr برای مشاهده category داخلی (c10) استفاده کنید:

```
[root@myServer ~]# getfattr -n security.selinux
gravityControl.txt

# file: gravityControl.txt

security.selinux="user_u:object_r:tmp_t:s0:c10\000"
```

شروع کار با امنیت چند طبقه‌ای

در این قسمت استفاده از برچسب‌های MCS برای گسترش قابلیت‌های MAC SELinux، دسته‌بندی‌های MCS، هویت کاربر SELinux و نحوه اعمال آن‌ها در حساب‌های کاربری و فایل‌های لینوکس را مورد بحث قرار می‌دهیم.

برچسب زدن MCS از نگاه یک کاربر و مدیر سیستم، ساده است. این کار شامل پیکربندی مجموعه‌ای از دسته‌ها است که فقط برچسب‌های متنی هستند، مانند "Company_Confidential" یا "Medical_Records". سپس کاربران را به آن دسته‌ها اختصاص می‌دهیم. مدیر سیستم ابتدا دسته‌بندی‌ها را اولویت‌بندی می‌کند و سپس کاربران را در صورت نیاز به آن‌ها اختصاص می‌دهد. پس از آن کاربران می‌توانند از برچسب‌ها استفاده کنند.

اسامی دسته‌ها و معانی آن‌ها توسط مدیر سیستم تعیین شده است و می‌توان آن‌ها را به هر آنچه که مورد نیاز است تنظیم کرد. یک سیستم در یک محیط خانگی ممکن است تنها یک دسته "خصوصی" داشته باشد و پیکربندی شود تا فقط کاربران محلی مورد اعتماد به این دسته اختصاص داده شوند.

در محیط Enterprise، از دسته‌بندی‌ها می‌توان برای شناسایی اسناد محرمانه به بخش‌های خاص استفاده کرد. دسته‌بندی‌ها می‌توانند با نام‌های "مالی"، "حقوق و دستمزد"، "بازاریابی" و "پرسنل" ایجاد شوند. فقط کاربران اختصاص داده شده به آن دسته‌ها می‌توانند به منابع برچسب زده شده با همان دسته‌بندی دسترسی داشته باشند.

پس از آنکه کاربران به دسته‌ها اختصاص داده شدند، می‌توانند هر یک از فایل‌های خود را با هر یک از دسته‌هایی که به آن‌ها اختصاص داده شده است، برچسب‌گذاری کنند. به‌عنوان مثال، یک کاربر خانگی در سیستم که در بالا توضیح داده شده است، می‌تواند تمام فایل‌های شخصی خود را با عنوان "خصوصی" برچسب‌گذاری کند و هیچ سرویس دیگری مانند Apache یا vsftp هرگز قادر به دسترسی به این فایل‌ها نخواهد بود، زیرا آن‌ها فایل‌های "خصوصی" هستند.

MCS با یک اصل ساده کار می‌کند: برای دسترسی به یک فایل، کاربر باید به تمام دسته‌هایی که فایل آن برچسب‌گذاری شده است اختصاص داده شود. بررسی MCS پس از قوانین کنترل دسترسی اختیاری لینوکس (DAC) و نوع اجرا (TE) انجام می‌شود، بنابراین پس از آن می‌توان امنیت بیشتری را محدود کرد.

مقایسه SELinux و تأیید هویت استاندارد کاربران لینوکس

SELinux هویت کاربر خود را برای فرآیندها، جدای از هویت کاربر لینوکس حفظ می‌کند. در سیاست هدف یا targeted policy (پیش‌فرض برای Linux) حداقل تعدادی هویت کاربر SELinux وجود دارد:

system_u - فرایندهای سیستم

root - مدیر سیستم

user_u - تمامی کاربرهای Login شده

از دستور semanage -l برای فهرست کاربران SELinux، استفاده می‌کنیم:

```
[root@dhcp-133 ~]# semanage user -l

                Labeling   MLS/          MLS/
SELinux User    Prefix     MCS Level     MCS Range      SELinux Roles

root            user       s0            s0-s0:c0.c1023 system_r sysadm_r user_r
system_u        user       s0            s0-s0:c0.c1023 system_r
user_u          user       s0            s0-s0:c0.c1023 system_r sysadm_r user_r
```

Loginهای SELinux

یکی از خواص سیاست هدف یا targeted policy این است که کاربران login شده به سیستم در یک محیط امنیتی مشابه اجرا می‌شوند. از دیدگاه TE، در سیاست هدف، آن‌ها معادل امنیت هستند. با این حال، برای استفاده از MCS، باید بتوانیم مجموعه‌های مختلفی از دسته‌ها را برای کاربران مختلف لینوکس اختصاص دهیم، هرچند که آن‌ها SELinux user (user_u) های یکسان هستند. این مورد با معرفی مفهوم SELinux login حل شده است که در طول فرآیند ورود برای اختصاص دادن دسته‌های MCS به کاربران لینوکس زمانی که آن‌ها ایجاد می‌شوند استفاده می‌شود.

از دستور semanage login -a برای تعیین هویت کاربران لینـوکس برای کاربر SELinux استفاده کنید:

```
# semanage login -a james

# semanage login -a daniel

# semanage login -a olga
```

زمانی که شما لیسـت کـاربران SELinux را مـی‌بینیـد، مـی‌توانیـد کاربران لینوکس را به یک شناسه کـاربر مشـخص SELinux اختصـاص دهید:

```
# semanage login -l

Login Name           SELinux User
MLS/MCS Range

__default__          user_u              s0
james                user_u              s0

daniel               user_u              s0

root                 root                SystemLow-
SystemHigh

olga                 user_u              s0
```

توجه داشته باشید که در این مرحله تنها حساب کاربری root به هـر دسته اختصاص داده می‌شود. به‌طور پیش‌فرض، حساب root با دسترسی به تمام دسته‌ها پیکربندی شده است.

Red Hat Enterprise و SELinux با تعدادی مقادیر پیش‌فرض از پیش تنظیم می‌شوند، اما برای استفاده مؤثر از MCS، مدیر سیستم به‌طور معمول این تغییرات را ایجاد می‌کند یا مقادیر بیشتری را ایجاد می‌کند تا با نیازهای محلی منطبق باشند.

پیکربندی دسته‌ها

SELinux، نگاشت بین حساسیت داخلی و سطوح دسته‌ها و نمایش‌های قابل خواندن را در فایل setrans.conf حفظ می‌کند. سرپرست سیستم این فایل را مدیریت و نگهداری می‌کند.

از دستور chcat -L برای دسته‌بندی‌های فعلی استفاده کنید:

```
[root@dhcp-133 tmp]# chcat -L
s0:c0                   CompanyConfidential
s0:c3                   TopSecret
s0
s0-s0:c0.c255           SystemLow-SystemHigh
s0:c0.c255              SystemHigh
```

برای تغییر دسته‌ها و یا ایجاد دسته خودتان، فایل /etc/selinux/<selinuxtype>/setrans.conf را تغییر دهید. برای مثال، دسته بازاریابی، مالی، حقوق و دستمزد و پرسنل را به صورت زیر اضافه کنید (این مثال از سیاست هدف استفاده می‌کند و بخش‌های اضافه فایل را حذف می‌کند):

```
[root@dhcp-133 tmp]# vi
/etc/selinux/targeted/setrans.conf
```

```
s0:c0=Marketing
s0:c1=Finance
s0:c2=Payroll
s0:c3=Personnel
```

برای بررسی دسته‌های جدید اضافه شده از دستور chcat -L استفاده کنید:

```
[root@dhcp-133 tmp]# chcat -L
s0:c0                Marketing
s0:c1                Finance
s0:c2                Payroll
s0:c3                Personnel
s0
s0-s0:c0.c255        SystemLow-SystemHigh
s0:c0.c255           SystemHigh
```

نکته: پس از تغییر فایل setrans.conf، قبل از اینکه تغییرات اعمال شوند، باید سرویس ترجمه MCS را مجدداً راه‌اندازی کنید. از دستور زیر برای راه‌اندازی مجدد سرویس استفاده کنید:

```
[root@dhcp-133 ~]# service mcstrans restart
```

اختصاص دسته به کاربران

حالا که دسته‌های مورد نیاز به سیستم اضافه شدند، می‌توانید آن‌ها را به کاربران و فایل‌های SELinux اختصاص دهید. برای توسعه بیشتر

مثال فوق، فرض کنید که James در بخش بازاریابی قرار دارد، Daniel در بخش امور مالی و حقوق و دستمزد است و Olga در بخش کارکنان است. هر یک از این کاربران قبلاً یک SELinux Login را تعیین کرده‌اند.

از دستور chcat برای اختصاص دادن مقادیر MCS به ورودی SELinux استفاده کنید:

```
[root@dhcp-133 ~]# chcat -l -- +Marketing james
[root@dhcp-133 ~]# chcat -l -- +Finance,+Payroll
daniel
[root@dhcp-133 ~]# chcat -l -- +Personnel olga
```

شما همچنین می‌توانید از دستور chcat با آرگومان‌های خط فرمان برای لیست کردن دسته‌هایی که به کاربران اختصاص داده شده‌اند استفاده کنید:

```
[root@dhcp-133 ~]# chcat -L -l daniel james olga
daniel: Finance,Payroll
james: Marketing
olga: Personnel
```

همچنین شما می‌توانید کاربران لینوکسی بیشتری را اضافه کنید، آن‌ها را به هویت‌های SELinux اختصاص دهید و سپس آن‌ها را به عنوان مدنظرتان طبقه‌بندی کنید. به عنوان مثال، اگر مدیر شرکت نیاز به یک حساب کاربری با دسترسی به تمام دسته‌ها داشته باشد، روش زیر را انجام دهید:

```
# Create a user account for the company director (Karl)
[root@dhcp-133 ~]# useradd karl
```

```
[root@dhcp-133 ~]# passwd karl
Changing password for user karl.
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.

# Assign the user account to an SELinux login
[root@dhcp-133 ~]# semanage login -a karl

# Assign all the MCS categories to the new login
[root@dhcp-133 ~]# chcat -l --
+Marketing,+Finance,+Payroll,+Personnel karl
```

برای بررسی افزوده شدن کاربر جدید از دستور chcat استفاده کنید:

```
[root@dhcp-133 ~]# chcat -L -l daniel james olga karl
daniel: Finance,Payroll
james: Marketing
olga: Personnel
karl: Marketing,Finance,Payroll,Personnel
```

اختصاص دادن دسته به فایل‌ها

در این مرحله ما یک سیستم لینوکسی داریم که دارای چندین حساب کاربری است. هر کدام از این حساب‌های کاربری به یک شناسه کاربری SELinux تخصیص داده شده است. ما همچنین تعدادی دسته مناسب برای پیکربندی خاصمان ایجاد کرده‌ایم و این دسته‌ها را به کاربران مختلف اختصاص داده‌ایم.

با این حال، تمام فایل‌های موجود در سیستم همچنان تحت همان رده قرار می‌گیرند و از این رو برای هر کسی قابل دسترس است (اما همچنان بر اساس استاندارد لینوکس DAC و محدودیت‌های TE). اکنون ما باید دسته‌بندی‌ها را به فایل‌های مختلف در سیستم اختصاص دهیم تا فقط کاربران مناسب بتوانند به آن‌ها دسترسی پیدا کنند.

برای مثال، ما یک فایل در دایرکتوری Daniel ایجاد می‌کنیم:

```
[daniel@dhcp-133 ~]$ echo "Financial Records 2006"
> financeRecords.txt
```

برای بررسی محتوای امنیتی فایل ابتدا از دستور ls -Z استفاده کنید:

```
[daniel@dhcp-133 ~]$ ls -Z financeRecords.txt

-rw-r--r--  daniel daniel user_u:object_r:user_home_t
  financeRecords.txt
```

توجه داشته باشید که در این مرحله فایل دارای زمینه پیش‌فرض برای یک فایل ایجاد شده در دایرکتوری خانگی کاربر (user_home_t) است و هیچ دسته‌ای به آن اختصاص داده نشده است. ما می‌توانیم با استفاده از دستور chcat دسته مدنظرمان را اضافه کنیم. اکنون وقتی که محتوای امنیتی فایل را بررسی می‌کنید، می‌توانید ببینید که این دسته اعمال شده است.

```
[daniel@dhcp-133 ~]$ chcat -- +Finance
financeRecords.txt

[daniel@dhcp-133 ~]$ ls -Z financeRecords.txt

-rw-r--r--  daniel daniel
root:object_r:user_home_t:Finance financeRecords.txt
```

در بسیاری از موارد، شما باید به یک فایل بیش از یک دسته اختصاص دهید. برای مثال، ممکن است بعضی از فایل‌ها برای کاربران از هر دو بخش مالی و حقوق و دستمزد قابل دسترس باشند.

```
[daniel@dhcp-133 ~]$ chcat -- +Payroll
financeRecords.txt

[daniel@dhcp-133 ~]$ ls -Z financeRecords.txt

-rw-r--r--  daniel daniel
root:object_r:user_home_t:Finance,Payroll
financeRecords.txt
```

هر کدام از دسته‌هایی که به فایل اختصاص داده شده‌اند در زمینه امنیت (Security Context) نمایش داده می‌شوند. شما می‌توانید دسته‌ها را به فایل‌ها اضافه کرده یا حذف کنید. فقط کاربران اختصاص داده شده به آن دسته می‌توانند به آن فایل دسترسی داشته باشند. (فرض کنید که مجوزهای لینوکس DAC و TE قبلاً اجازه دسترسی را داده‌اند.)

اگر یک کاربر که به یک گروه دیگر اختصاص داده شده است در تلاش برای دسترسی به فایل باشد، یک پیام خطا دریافت می‌کند:

```
[olga@dhcp-133 ~]$ cat financeRecords.txt

cat: financeRecords.txt: Permission Denied
```

امنیت چند سطحی (MLS)

حفاظت از اطلاعات حساس یا محرمانه در بسیاری از کسب‌وکارها بسیار مهم است. در صورتی که چنین اطلاعاتی در اختیار عموم افراد قرار گیرد، کسب‌وکار ممکن است با مسائل قانونی یا مالی مواجه شود و باعث از دست دادن اعتماد مشتری شود. در اغلب موارد، این مسائل می‌توانند زیان‌های مالی جبران ناپذیری ایجاد کنند.

داشتن اطلاعاتی از سطوح مختلف امنیتی در یک سیستم کامپیوتری، تهدیدی واقعی است. این موضوع مستقیماً به علت جدا کردن سطوح مختلف امنیت اطلاعات نیست، حتی اگر کاربران مختلف با استفاده از حساب‌های مختلف، با مجوزهای مختلف و کنترل‌های دسترسی متفاوت وارد سیستم شوند.

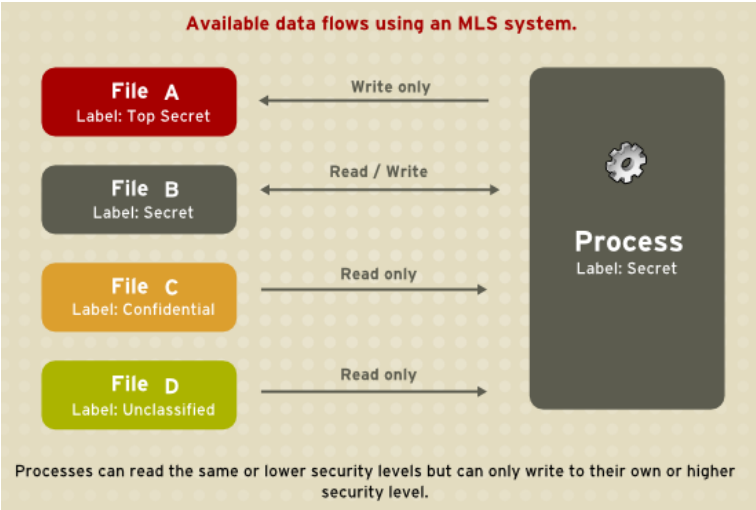
برخی از سازمان‌ها سیستم‌های اختصاصی را برای هر سطح امنیتی خریداری می‌کنند. با این حال، اغلب بسیار گران هستند. یک مکانیسم لازم برای فعال کردن کاربران در سطوح مختلف امنیتی برای دسترسی همزمان به سیستم، نداشتن ترس از آلودگی اطلاعات است.

اصطلاح چند سطحی از طبقه‌بندی‌های امنیتی جامعه دفاع می‌کند: محرمانه، مخفی و امنیت زیاد.

قبل از اینکه افراد بتوانند اطلاعات طبقه‌بندی شده را ببینند، مجوزهای لازم به آن‌ها اعطا می‌شود. کسانی که دارای اعتبار محرمانه هستند تنها مجاز به مشاهده اسناد محرمانه هستند؛ آن‌ها اطمینان ندارند که بتوانند اطلاعات امنیتی را جستجو کنند. قوانین مربوط به جریان داده‌ها از سطوح پایین‌تر به سطوح بالاتر و بالعکس عمل می‌کنند؛ که در شکل زیر نشان داده شده است.

مدل Bell-La Padula (BLP)

SELinux، مانند بسیاری دیگر از سیستم‌هایی که اطلاعات چند سطحی را حفظ می‌کنند، از مدل BLP استفاده می‌کند. این مدل مشخص می‌کند چگونه اطلاعات می‌توانند درون سیستم بر اساس برچسب‌های اختصاص داده شده به هر موضوع و شیء جریان یابد. به نمودار زیر توجه کنید:



تحت چنین سیستمی، کاربران، رایانه‌ها و شبکه‌ها از برچسب‌ها برای نشان دادن سطوح امنیت استفاده می‌کنند. داده‌ها می‌توانند بین سطوحی مانند، مانند "secret" و "secret" یا از سطح پایین‌تر به سطح بالاتر، حرکت کنند. این مسئله به آن معنی است که کاربران در سطح «Secret» می‌توانند داده‌ها را با یکدیگر به اشتراک بگذارند و همچنین می‌توانند اطلاعات را از سطح محرمانه (یعنی سطح پایین)، بازیابی کنند. با این حال، داده‌ها نمی‌توانند از یک سطح بالاتر به یک سطح پایین‌تر جریان یابند. این مسئله مانع مشاهده فرآیندهای در سطح "secret" با اطلاعات طبقه‌بندی شده با عنوان "Top secret" می‌شود. همچنین از نوشتن تصادفی اطلاعات در فرآیندهای سطح بالاتر به سطح پایین‌تر ممانعت به عمل می‌آید. این کار با عنوان مدل "بدون خواندن، بدون نوشتن" نامیده می‌شود.

MLS و امتیازات سیستم (System Privileges)

قوانین دسترسی MLS همیشه با مجوزهای دسترسی معمولی (مجوزهای فایل) ترکیب می‌شوند. برای مثال، اگر یک کاربر با سطح امنیتی "secret" از کنترل دسترسی اختیاری (DAC) برای جلوگیری از دسترسی به یک فایل توسط سایر کاربران استفاده کند، همچنین اجازه دسترسی به سطح امنیتی "top secret" را نمی‌دهد. امنیت بالاتر به‌طور خودکار اجازه دسترسی غیرمجاز به یک فایل سیستم را به‌طور خودسرانه نمی‌دهد.

کاربران دارای مجوزهای سطح بالا به‌طور خودکار مجوزهای دسترسی را در سیستم‌های چند سطحی بدست نمی‌آورند. در حالی که آن‌ها ممکن است به تمام اطلاعات در کامپیوتر دسترسی داشته باشند که متفاوت از داشتن حقوق مدیر است.

سطوح امنیت، اشیاء و موضوع

همان‌طور که در بالا ذکر شد، موضوعات و اشیاء با سطوح امنیتی (SLs) برچسب‌گذاری می‌شوند که از دو نوع موجودیت تشکیل شده‌اند:

۱. حساسیت (Sensitivity): - ویژگی سلسله مراتبی مانند "secret" یا "secret Top".

۲. دسته‌بندی‌ها (Categories): - مجموعه‌ای از ویژگی‌های غیر سلسله مراتبی مانند "US Only" یا "UFO".

SL باید حتماً یک حساسیت داشته باشد و ممکن است دارای صفر یا تعداد بیشتری از دسته‌ها باشد.

مثال‌های SL عبارتند از: {Crypto ،UFO / secret}، { Top Secret / UFO ،Crypto ،Stargate} و {Unclassified}

سیاست MLS

SELinux از مـــدل Bell-La Padula یـــا BLP و Type Enforcement یا TE بـرای یکپـارچگی اسـتفاده مـی‌کنـد. بـه عبـارت ساده‌تر، سیاست MLS تضمین می‌کنـد کـه یـک موضـوع دارای مجـوز مناسب برای دسترسی به یک شیء با طبقه‌بندی خاص است.

به عنوان مثال، تحت MLS، سیستم باید بداند چگونه پـردازش یـک درخواست را انجام دهد: آیا می‌توان یک فرایند محرمانه {Top Secret / UFO، Rail gun} را بـا یـک فایـل بـه نـام {Top Secret / UFO} نوشت؟

مدل MLS و سیاست اجرا شده برای آن، پاسخ را تعیین خواهد کرد.

MLS مجموعه‌ای بسیار محدود (و در عین حال بحرانی) از نیازهـای امنیتی را بر اساس اطلاعات و پرسنل در محیط‌های سـخت‌گیرانـه نظیـر ارگان‌های نظامی اداره می‌کند. کار با MLS به‌طور معمول دشوار اسـت و به خوبی به سناریوهای عمومی نگاشت نمی‌شود.

TE تحت SELinux یک طرح امنیتی انعطاف‌پذیر و قابل بیان است که در بسیاری موارد مناسب‌تر از MLS می‌باشد.

با این حال، چندین سناریو وجود دارد که MLS سـنتی هنـوز مـورد نیاز است. به‌عنوان مثال، یک سرور فایل که در آن داده‌های ذخیره شـده ممکن است از طبقه‌بندی‌های متفاوت باشـند و مشـتریان بـا مجوزهـای مختلف متصل می‌شوند. این مسئله بـه تعـداد زیـادی از سـطوح امنیتـی منجر می‌شود و نیاز به جداسازی قدرتمندی در یک سیستم واحد دارد.

این نوع سناریو دلیل آن است که SELinux باید شامل MLS باشد (در کنار TE).

مرور کلی سیاست SELinux

در این قسمت یک مرور کلی از سیاست SELinux، برخی از کاربردهای آن و نحوه کار با آن را بررسی می‌نماییم. با مرور کوتاهی در مورد سیاست‌ها و جایی که در آن قرار دارند، آغاز می‌کنیم. پس از این، نقش SELinux در طول فرایند بوت مورد بحث قرار می‌گیرد. بحث‌های صورت گرفته در مورد زمینه‌های امنیتی، کلاس‌ها و مجوزها، ویژگی‌ها، انواع، دسترسی به بردارها، ماکروها، کاربران و نقش‌ها، محدودیت‌ها و خلاصه‌سازی رابط‌های هسته‌ای می‌باشند.

سیاست SELinux چیست؟

سیاست SELinux مجموعه قوانینی است که موتور امنیتی SELinux را هدایت می‌کند. انواع Object ها و دامنه‌های پردازش را تعریف می‌کند و از نقش‌ها برای محدود کردن دامنه‌هایی که می‌توانند وارد شوند استفاده می‌شود که دارای هویت کاربر برای مشخص کردن نقش‌هایی است که می‌توانند آن را به دست آورند. در اصل، انواع و دامنه‌ها معادل هستند، تفاوت این است که انواع مربوط به اشیاء هستند در حالی که دامنه‌ها به فرآیندها اعمال می‌شوند.

انواع در SELinux (SELinux Types)

نوع یک روش دسته‌بندی آیتم‌ها است که این دسته‌بندی بر اساس شباهت آن‌ها از دیدگاه امنیتی صورت می‌گیرد. این امر لزوماً با هدف منحصر به یک برنامه یا محتوای یک سند مرتبط نیست. برای مثال، یک فایل می‌تواند هر نوع محتوایی داشته باشد و برای هر هدفی باشد، اما اگر به کاربر تعلق دارد و در پوشه اصلی آن کاربر وجود داشته باشد، به عنوان یک نوع امنیتی خاص، user_home_t در نظر گرفته می‌شود.

بنابراین انواع شیء به صورت یکسان در نظر گرفته می‌شوند، زیرا با مجموعه‌ای یکسان از موضوعات و به شیوه یکسان قابل دسترس هستند. به همین ترتیب، فرآیندها اگر مجوزی مشابه با سایر موضوعات داشته باشند، نیز به شیوه‌ای مشابه می‌باشند. در سیاست هدف، برنامه‌هایی که در دامنه unconfined_t اجرا می‌شوند، یک فایل اجرایی با نوعی از دامنه sbin_t هستند. از دیدگاه SELinux، این به آن معنی است که همه آن‌ها از نظر آنچه که می‌توانند و نمی‌توانند در سیستم انجام دهند معادل هستند.

به‌عنوان مثال، فایل اجرایی باینری /usr/bin/postgres نوع postgresql_exec_t دارد. تمامی daemon های هدف دارای نوع *_exec_t از نوع برنامه‌های کاربردی قابل اجرا هستند. در واقع، تمام مجموعه‌های اجرایی PostgreSQL مانند createlang، pg_dump و pg_restore مشابه نوع postgresql_exec_t هستند و بعد از اجرای آن به یک دامنه postgresql_t منتقل می‌شوند.

استفاده از قوانین خط مشی برای تعیین دسترسی انواع

سیاست SELinux قوانینی را تعیین می‌کند که بیان می‌کنند چگونه هر دامنه به هر نوع دسترسی داشته باشد. فقط آنچه که به طور خاص توسط قوانین اجازه داده می‌شود مجاز است. به‌طور پیش‌فرض، هر عملی ممنوع و حسابرسی (audit) می‌شود، به این معنی که در فایل AUDIT_LOG وارد می‌شود. در Linux این مورد به فایل /var/log/messages ارسال می‌شود. این سیاست به فرمت باینری برای بارگیری در سرور امنیتی هسته کامپایل می‌شود و هر بار که سرور امنیتی تصمیم‌گیری می‌کند، بهینه‌سازی عملکرد در AVC کش (cache) می‌شود.

این خط مشی را می‌توان با تغییر فایل‌های موجود یا اضافه کردن فایل‌های اجرایی نوع محلی (TE) و فایل متنی (FC) به درخت سیاست تعریف کرد. این سیاست‌های جدید را می‌توان در زمان واقعی در هسته بارگذاری کرد. در غیر این صورت، خط مشی در طول فرایند بوت توسط init بارگذاری می‌شود. در نهایت، هر عملیات سیستم به وسیله سیاست و نوع برچسب فایل‌ها تعیین می‌شود.

SELinux و کنترل دسترسی اجباری (MAC)

SELinux یک پیاده‌سازی از کنترل دسترسی اجباری (MAC) است. بسته به نوع سیاست امنیتی، SELinux هر دو نوع تطبیق (TE)، کنترل دسترسی بر اساس نقش (RBAC) و یا امنیت چند سطحی Bell-La Padula (MLS) را اجرا می‌کند.

این سیاست قوانین در محیط اجرا شده را مشخص می‌کند که به زبانی برای نوشتن سیاست امنیتی به طور خاص نوشته شده است. نویسندگان خط مشی از ماکروهای M4 برای جمع‌آوری قوانین سطح پایین استفاده می‌کنند. تعدادی از ماکروهای M4 در سیاست موجود تعریف می‌شوند که نوشتن سیاست جدید را تسهیل می‌کنند. این قوانین به عنوان بخشی از ایجاد فایل policy.conf بیان می‌شوند که با بسیاری از قوانین اضافه پردازش می‌شوند و در سیاست باینری کامپایل می‌شوند.

حقوق دسترسی به دو دسته تقسیم می‌شود و هیچ حوزه‌ای برای انجام کار به‌عنوان سرشاخه برای تمام دامنه‌های دیگر نیاز نیست. حرکت بین دامنه‌ها توسط سیاست، از طریق برنامه‌های ورود، برنامه‌های کاربردی مانند newrole یا با اجرای یک فرآیند جدید در دامنه جدید کنترل می‌شود. این حرکت بین حوزه‌ها به عنوان یک انتقال شناخته می‌شود.

خط مشی کجاست؟

دو بخش برای سیاست وجود دارد: درخت دودویی و درخت منبع. درخت باینری توسط بسته selinux-policy-<policyname> ارائه شده و فایل سیاست باینری را فراهم می‌کند.

در عوض، زمانی که بسته selinux-policy-devel نصب می‌شود، خط مشی باینری از منبع می‌تواند ساخته شود.

فایل‌های درختی باینری

/etc/selinux/targeted/- این دایرکتوری ریشه برای سیاست هدف است و شامل درخت باینری است.

/etc/selinux/targeted/policy/- محلی برای فایل خط مشی باینری policy.<xx> است. متغیر SELINUX_POLICY برای این پوشه استفاده می‌شود.

/etc/selinux/targeted/contexts/- محلی برای اطلاعات امنیتی و فایل‌های پیکربندی است که در زمان اجرا توسط برنامه‌های مختلف مورد استفاده قرار می‌گیرد.

/etc/selinux/targeted/contexts/files/- شامل زمینه‌های پیش‌فرض برای کل سیستم فایل است که توسط restecon زمانی که عملیات remabeling اجرا شود ارجاع داده می‌شود.

/etc/selinux/targeted/contexts/users/ - در سیاست هدف، تنها فایل ریشه در این پوشه است. این فایل‌ها برای تعیین زمینه، زمانی که یک کاربر وارد سیستم می‌شود استفاده می‌شوند. به‌عنوان مثال، برای کاربر ریشه، زمینه user_u: system_r: unconfined_t است.

/etc/selinux/targeted/modules/active/booleans*- جـایی است که Booleanها اجرا می‌شوند.

نکته: این فایل‌ها هرگز نباید دستی تغییر کنند. شما باید از ابزارهای getsebool، setsebool و semanage برای دست‌کاری boolean های runtime استفاده کنید.

فایل‌های درخت منبع

برای توسـعه مؤلفـه‌هـای سیاسـت، پکـیج selinux-policy-devel شامل تمام فایل‌های رابط کاربری مورد استفاده بـرای سـاختن سیاسـت می‌باشد. توصیه می‌شود افرادی که از ساختن خط مشی استفاده می‌کنند از این فایل‌ها برای ایجاد ماژول‌های خط‌مشی استفاده کنند.

ایـــن بســـته فایـــل‌هـــای رابـــط سیاســـت‌گـــذاری را در /usr/share/selinux/devel/include و Makefile هـــــــــا را در /usr/share/selinux/devel/Makefile نصب می‌کند.

برای کمک به برنامه‌هایی که به مسـیرهای مختلـف SELinux نیـاز دارند، libselinux تعدادی از توابع را فراهم می‌کنـد کـه مسـیرها را بـه فایل‌های پیکربندی و دایرکتوری‌های مختلف بازمی‌گرداند. این کـار نیـاز به hard-code کردن مسیرها را از بین می‌برد، به‌ویژه از آنجا کـه مکـان سیاســت‌هــای فعــال بســتگی بــه تنظــیم SELINUXTYPE در /etc/selinux/config دارد.

برای مثال، اگر SELINUXTYPE به strict تنظیم شود، موقعیـت سیاست فعال تحت /etc/selinux/strict قرار دارد.

برای مشاهده لیست توابع موجود، از دستور زیر استفاده کنید:

```
man 3 selinux_binary_policy_path
```

نقش سیاست در فرایند بوت

SELinux نقش مهمی در مراحل اولیه راه‌اندازی سیستم دارد. از آنجا که تمام فرآیندها باید با دامنه صحیح برچسب‌گذاری شوند، Init یک عملیات اساسی در اوایل فرایند بوت انجام می‌دهد تا هماهنگی بین برچسب‌گذاری و اجرای سیاست را حفظ کند.

پس از اینکه کرنل در طول فرایند بوت بارگذاری شد، پردازش اولیه به هسته اولیه SELinux ID (SID اولیه) اختصاص داده می‌شود. SID های اولیه برای bootstrapping قبل از اینکه خط‌مشی‌ها بارگذاری شوند استفاده می‌شود.

/sbin/init پوشه /proc/ را mount می‌کند سپس برای نوع سیستم فایل selinuxfs جستجو می‌کند. اگر موجود باشد، بدان معنی است که SELinux در هسته فعال است.

اگر SELinux init را در هسته پیدا نکرد یا اگر از طریق پارامتر boot پارامتر selinux = 0 غیرفعال شود یا اگر /etc/selinux/config مشخص کند SELINUX=disabled، فرآیند بوت با سیستم بدون SELinux ادامه می‌یابد.

به‌صورت همزمان، init وضعیت enforcing را تعیین می‌کند، اگر از تنظیمات درون /etc/selinux/config متفاوت باشد. این اتفاق زمانی می‌افتد که پارامتر در طول فرایند بوت منتقل شود. حالت پیش‌فرض تا زمانی که خط مشی بارگذاری می‌شود، مجاز است و سپس اجرا توسط فایل پیکربندی یا پارامترهای enforcing = 0 یا enforcing = 1 تنظیم می‌شود.

اگر SELinux موجود باشد، /selinux/ نصب شده است.

کرنل /selinux/policyvers را برای نسخه سیاست پشتیبانی شده بررسی می‌کند. init با بررسی /etc/selinux/config تعیین می‌کند

کدام سیاست فعال است، مانند سیاست هدف، سپس فایل مرتبط را در $SELINUX_POLICY/policy.<version> بارگذاری می‌کند.

اگر خط مشی باینری نسخه‌ای نباشد که توسط هسته پشتیبانی می‌شود و نسخه قبلی باشد، init تلاش می‌کند که فایل سیاست را بارگذاری کند. این سازگاری برگشت به نسخه‌های قدیمی‌تر را فراهم می‌کند.

اگر تنظیمات محلی موجود در /etc/selinux/targeted/booleans از دسته سیاست‌های کامپایل شده متفاوت باشد، init بر اساس تنظیمات محلی قبل از بارگیری خط‌مشی در هسته به سیاست‌های حافظه تغییر می‌کند.

در این مرحله از فرآیند، سیاست به طور کامل در هسته بارگذاری می‌شود. سپس SID های اولیه به زمینه‌های امنیتی در سیاست نگاشت می‌شوند. در مورد سیاست هدف، دامنه جدید user_u: system_r: unconfined_t است. در حال حاضر هسته می‌تواند زمینه‌های امنیتی را به صورت پویا از سرور امنیت داخل هسته برگرداند.

سپس init خود را مجدداً اجرا می‌کند تا در صورتی که بتواند سیاست آن را تعریف کند، به دامنه دیگری انتقال یابد. برای سیاست هدف، هیچ تغییری تعریف نشده و init در دامنه unconfined_t باقی می‌ماند.

در این مرحله، init با فرآیند بوت معمولی خود ادامه می‌یابد.

دلیل این امر آن است که init خود را مجدداً اجرا می‌کند تا کنترل دقیق SELinux را تطبیق دهد. هدف اجرای مجدد انتقال به یک دامنه جدید با قوانین گرانشی خود است. تنها راهی که یک فرآیند می‌تواند در یک دامنه وارد شود، در حین اجرا است، به این معنی که این پروسه‌ها تنها نقطه ورود به دامنه‌ها هستند.

برای مثال، اگر سیاست یک دامنه خاص برای init، مانند init_t باشد، یک روش برای تغییر از SID اولیه، مانند kernel، به دامنه زمان اجرای صحیح برای init، لازم است وجود داشته باشد. از آنجا که ممکن است این انتقال رخ دهد، init کدگذاری شده است تا پس از بارگذاری خط‌مشی خود دوباره اجرا شود.

این گذار init زمانی رخ می‌دهد که قانون domain_auto_trans(kernel_t, init_exec_t, <target_domain_t>) در خط مشی وجود داشته باشد. این قانون بیان می‌کند که یک انتقال اتوماتیک بر روی هر چیزی که در دامنه kernel_t اجرا می‌شود و یک فایل از init_exec_t را اجرا می‌کند، رخ می‌دهد. هنگامی که این اجرا رخ می‌دهد، فرآیند جدید دامنه <target_domain_t> با استفاده از دامنه هدف واقعی مانند init_t اختصاص می‌یابد.

کلاس‌های اشیاء و مجوزها

SELinux تعدادی کلاس برای اشیا تعریف می‌کند و اجازه می‌دهد تا گروه‌های خاصی را با استفاده از کلاس‌های خاص دسته‌بندی کنید. مثلاً:

- کلاس‌های مرتبط با فایل عبارت‌اند از file system برای سیستم فایل، file برای فایل‌ها و dir برای دایرکتوری‌ها. هر کلاس دارای مجموعه‌ای از مجوزهای مرتبط با آن است.

  کلاس file system می‌تواند mount، unmount، دریافت ویژگی‌ها، تنظیم مقادیر، برچسب مجدد و غیره باشد. کلاس file دارای مجوزهای فایل مشترکی مانند خواندن، نوشتن، گرفتن و تنظیم صفات، قفل، ترمیم، لینک، تغییر نام، اضافه کردن و غیره است.

- کلاس‌های مربوط به شبکه شامل tcp_socket برای سوکت‌های TCP، netif برای رابط‌های شبکه و گره برای گره‌های شبکه است.

  برای مثال، کلاس netif می‌تواند در TCP، UDP و سوکت‌های خام (tcp_recv، tcp_send، udp_send، udp_recv، rawip_recv و rawip_send) ارسال و دریافت انجام دهد.

کلاس‌های شیء در declaration های هسته مطابقت دارند، به این معنی که افزودن یا تغییر جزئیات کلاس شیء، با سهولت انجام نمی‌گیرد. همین امر برای مجوزها نیز صادق است.

مجوزها اقداماتی هستند که یک موضوع می‌تواند بر روی یک شیء انجام دهد، اگر سیاست اجازه آن را بدهد. این مجوزها درخواست‌های دسترسی هستند که SELinux فعالانه اجازه می‌دهد یا رد می‌کند.

سیاست هدف (Targeted Policy)

Linux، یک سیاست واحد که سیاست هدف باشد را پشتیبانی می‌کند. تحت سیاست هدف، هر موضوع و شیء در دامنه unconfined_t بغیر از دایمون‌های هدف خاص اجرا می‌شود. اشیاء موجود در دامنه unconfined_t هیچ‌گونه محدودیتی ندارند و به استفاده از امنیت استاندارد لینوکس یعنی DAC برمی‌گردند. دایمون‌هایی که بخشی از سیاست هدف را دارا هستند در حوزه‌های (دامنه‌های) خود اجرا می‌شوند و در هر عملیاتی که در سیستم اجرا می‌شود، محدودیت دارند. این مسئله باعث می‌شود دایمون‌هایی از که نوع خاص حملات نیستند به هیچ وجه مورد سوء استفاده نگیرند و در معرض خطر واقع نشوند.

به‌عنوان مثال، دایمون های http و ntp هر دو در سیاست هدف (targeted policy) محافظت می‌شوند و به ترتیب در دامنه‌های

httpd_t و ntpd_t اجرا می‌شوند؛ اما سرویس SSH در این سیاست محافظت نمی‌شود و در نتیجه در دامنه unconfined_t اجرا می‌شود.

به خروجی نمونه زیر توجه کنید که دامنه‌های مختلفی را برای دایمون‌های ذکر شده در بالا نشان می‌دهد:

```
user_u:system_r:httpd_t        25129 ?        00:00:00
httpd

user_u:system_r:ntpd_t         25176 ?        00:00:00
ntpd

system_u:system_r:unconfined_t        25245 ?
00:00:00 sshd
```

سیاست سخت‌گیرانه (Strict Policy)

برخلاف سیاست هدف، سیاست در Strict Policy محکم یا سخت‌گیرانه است. در سیاست سخت‌گیرانه، هر موضوع و شیء در یک دامنه امنیتی خاص وجود دارد و همه تعاملات و انتقال‌ها به صورت جداگانه در قواعد سیاست‌گذاری قرار می‌گیرند.

کاربران و نقش‌ها در سیاست هدف (Targeted Policy)

این قسمت نقش‌های خاصی را که برای سیاست هدف مورد نظر فعال شده است پوشش می‌دهد. نوع unconfined_t در هر نقشی وجود دارد که به طور قابل توجهی مفید بودن نقش‌ها در سیاست هدف را کاهش می‌دهد. استفاده گسترده‌تر از نقش‌ها نیازمند تغییر به پارادایم سیاست سخت‌گیرانه است که هر فرآیند در یک دامنه به صورت جداگانه در نظر گرفته می‌شود.

به‌طور مؤثر، در سیاست هدف تنها دو نقش وجود دارد: system_r و object_r. نقش اولیه system_r است و هر چیز دیگری این نقش را به ارث می‌برد. نقش‌های دیگر برای اهداف سازگاری بین سیاست هدف و سیاست سخت‌گیرانه تعریف شده است.

سه نقش از چهار نقش توسط سیاست تعریف شده است. نقش چهارم، object_r، یک نقش ضمنی است و در منبع سیاست موجود نیست. از آنجا که نقش‌ها به‌وسیله انواع مختلفی و با استفاده از یک یا چند اعلان در سیاست ایجاد می‌شوند و در آن قرار می‌گیرند، هیچ فایلی وجود ندارد که تمامی نقش‌ها را اعلام کند. (به خاطر داشته باشید که این سیاست از چندین فایل جداگانه ساخته شده است.)

system_r: این نقش برای همه فرآیندهای سیستم بجز فرآیندهای کاربر است:

```
system_r (28 types)
   dhcpd_t
   httpd_helper_t
   httpd_php_t
   httpd_suexec_t
   httpd_sys_script_t
   httpd_t
   httpd_unconfined_script_t
   initrc_t
   ldconfig_t
   mailman_cgi_t
```

mailman_mail_t

mailman_queue_t

mysqld_t

named_t

ndc_t

nscd_t

ntpd_t

pegasus_t

portmap_t

postgresql_t

snmpd_t

squid_t

syslogd_t

system_mail_t

unconfined_t

winbind_helper_t

winbind_t

ypbind_t

user_r: این نقش، نقش پیش‌فرض کاربر برای کاربران معمولی لینوکس است. در سیاست سخت‌گیرانه، ممکن است کاربران منفرد مورد

استفاده قرار گیرند، به‌طوری که کاربران می‌توانند نقش خاصی برای انجام عملیات اختصاصی داشته باشند. در سیاست هدف، تمام کاربران در دامنه unconfined_t اجرا می‌شوند.

object_r: در SELinux، هنگام استفاده از RBAC، نقش‌ها برای اشیاء استفاده نمی‌شوند. اهداف به صورت strictly یا سخت گیرانه برای افراد ایجاد شده‌اند. این به آن علت است که نقش‌ها به صورت وظیفه گرا هستند و گروه‌هایی را تشکیل می‌دهند که اقدامات را انجام می‌دهند (به‌عنوان مثال، فرایندها). همه این‌ها به طور جمعی به‌عنوان موضوع تعریف می‌شوند. به همین دلیل تمام اشیاء نقش object_r دارند و نقش فقط به عنوان نگه‌دارنده در برچسب استفاده می‌شود.

sysadm_r: این نقش مدیر سیستم در یک سیاست سخت (strict) است. اگر به طور مستقیم به‌عنوان کاربر root وارد سیستم شوید، نقش پیش‌فرض می‌تواند staff_r باشد. اگر این مورد درست باشد، دستور newrole -r sysadm_r را برای تغییر نقش مدیر سیستم SELinux برای انجام وظایف مدیریت سیستم استفاده کنید.

sysadm_r (6 types)

httpd_helper_t

httpd_sys_script_t

initrc_t

در سیاست هدف، نکته زیر را برای سازگاری sysadm_r در نظر داشته باشید:

به‌طور مؤثر تنها یک هویت کاربر در سیاست هدف قرار دارد. هویت user_u انتخاب شده است چون libselinux به‌عنوان کاربر پیش‌فرض SELinux به کاربر user_u است. این اتفاق زمانی می‌افتد که کاربر SELinux مشابه کاربر لینوکس وارد نشده باشد. استفاده از user_u

به‌عنوان یک کاربر در سیاست هدف، باعث می‌شود تغییر در سیاست‌های strict آسان‌تر شود. دیگر کاربران برای سازگاری با سیاست strict در سیستم باقی‌مانده‌اند.

تنها استثناء، کاربر root در SELinux است. شما می‌توانید root را به عنوان هویت کاربر در زمینه فرآیند مشاهده کنید. این اتفاق زمانی می‌افتد که کاربر root، دایمون‌های SELinux را از خط فرمان شروع می‌کند یا یک دایمون را که ابتدا توسط init آغاز شده است، راه‌اندازی می‌کند.

کنترل کاربر انتهایی SELinux

به‌طورکلی، کاربران نهایی زمانی که لینوکس در حال اجرای سیاست هدف (taegeted policy) قرار دارد تعامل کمی با SELinux دارند. این به این دلیل است که کاربران در دامنه unconfined_t همراه با بقیه سیستم به‌جز دایمون‌های هدف در حال اجرا هستند.

در اغلب موارد، کنترل‌های استاندارد DAC مانع از انجام وظایفی می‌شود که قبل از کنترل SELinux، دسترسی و مجوز مورد نیاز را نداشتند. به این ترتیب، احتمال دارد که شما هرگز AVC همانند زیر را تولید نکنید:

avc: denied message.

بخش‌های زیر وظایف عمومی و عملیاتی را که کاربر نهایی ممکن است در یک سیستم Linux انجام دهد پوشش می‌دهند. این وظایف نه تنها برای کاربران نهایی بلکه برای کاربران تمامی سطوح اعمال می‌شود.