Zaufana infrastruktura obliczeniowa

## SELinix - wprowadzenie (Linux)

*To jest laboratorium ćwiczeniowe. Należy jest wykonać w czasie trwania zajęć. Zadanie to nie powinno zająć więcej czasu niż czas trwania laboratorium. Jeśli zadanie zostanie zakończone wcześniej, to można kontynuować prace dotyczące poprzedniego laboratorium lub pracy semestralnej.*

### Cel ćwiczenia

Celem ćwiczenia jest zapoznanie się z procedurami inicjowania oraz podstawowymi poleceniami bezpiecznego systemu operacyjnego SELinux.

### Efekty

Po ukończeniu ćwiczenia studenci powinni posiadać wystarczającą wiedzę dotyczącą podstaw obsługi systemu operacyjnego SELinux. Uzyskują także podstawową wiedzę na temat budowania zaufania do systemów komputerowych.

### 1. Wstęp

*Security-Enhanced Linux* to element jądra Linuksa i liczne programy dodatkowe, które rozszerzają funkcjonalność systemu o obowiązkową kontrolę dostępu. Jądro zawiera nowe architektoniczne komponenty stworzone po to, aby zwiększyć bezpieczeństwo systemu. Komponenty te dostarczają ogólne wsparcie dla prowadzenia wielu różnych polityk kontroli dostępu.

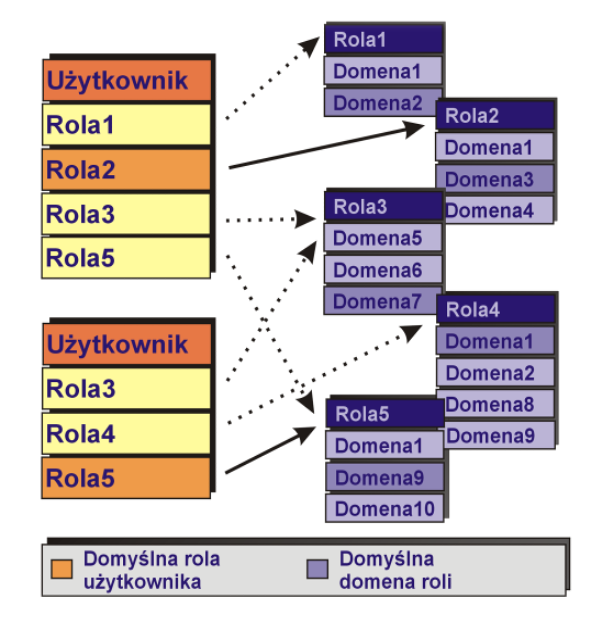
Jądro SELinux wymusza politykę, która ogranicza programy użytkowe i serwery systemowe do minimalnych przywilejów, które potrzebują do wykonania swych zadań. Dzięki temu możliwość wykorzystania luk w tych programach (przepełnienie bufora lub zła konfiguracja) jest zredukowana albo wyeliminowana. Ten mechanizm działa niezależnie od tradycyjnej kontroli dostępu w Linuksie. W szczególności nie ma w nim koncepcji superużytkownika i nie posiada on ograniczeń związanych z mechanizmem uid i gid.

Bezpieczeństwo w niezmodyfikowanym Linuksie zależy od poprawności kodu jądra, od kodu wszystkich jego uprzywilejowanych aplikacji i od konfiguracji tychże aplikacji. Wykorzystanie błędu w jednej z tych składowych powoduje całkowite zdobycie systemu przez intruza. W SELinux natomiast bezpieczeństwo zależy głównie od poprawności jądra i od konfiguracji polityki bezpieczeństwa. Wykorzystanie błędu w kodzie aplikacji lub konfiguracji tejże aplikacji nie powoduje całkowitego zdobycia systemu przez intruza.

SELinux tworzy model zabezpieczeń, który jest kombinacją czterech modeli:

1. Type Enforcement (TE)
2. Role-Based Access Control (RBAC)
3. User Based Access Control (UBAC)
4. Multi-Level Security (MLS) model (opcjonalne)

W systemie z SELinuksem każdy użytkownik ma przypisany zestaw ról, które może pełnić. Jedna z nich jest rolą domyślną. Każda rola dysponuje zestawem domen, zaś jedna z tych domen jest domyślna dla danej roli. Role mogą być wspólne dla wielu użytkowników, a domeny mogą być wspólne dla wielu ról. W danym momencie użytkownik wykonuje dokładnie jedną rolę w jednej domenie. To wszystko obrazuje poniższy rysunek.



Dla każdej pary (domena, typ obiektu) zdefiniowane są reguły zachowania systemu oraz uprawnienia, jakie użytkownik w danej domenie ma do obiektu danego typu.

Ćwiczenia zawarte w konspekcie mają za zadanie przedstawić główne założenia oraz polecenia systemu SELinux, oraz pokazać w praktyce realizację polityk bezpieczeństwa w modelu TE/RBAC. W trakcie przebiegu ćwiczeń należy odpowiadać na pytania oznaczone jako **Zadanie**.

### 2 Zadania teoretyczne

1. Jakie mechanizmy kontroli dostepu są zimplementowane w SELinux? Nazwij różne mechanizmy i przypisz im odpowiedni typ (MAC/DAC/RBAC).
2. W tradycyjnych systemach Linux najważniejszym atrybutem, w oprciu o który system podejmuje decyzję, jest identyfikator użytkowanika (UID). Jaki trybut jest najważniejszą cechą w SELinux?
3. Jaka jest różnica pomiędzy trybami *enforcing*, *disabled* i *permissive* w SELinux i jak przełączać/uaktywniać te tryby?
4. Co jest celem ponownego wykonania procesu *init* podczas uruchamiania systemu? Dlaczego konieczne jest ponownie uruchomienie systemu operacyjnego SELinux?
5. O czy należy pamiętać podczas tworzenia kopii zapasowych plików w systemie SELinux?
6. Jaka jest różnica między przejściem typu (ang. type transition), zmianą typu i re-etykietowanie? Podać krótkie wyjaśnienie wszystkich trzech pojęć, określić ewentualne różnice i relacje pomiędzy nimi.

### 3 Czynności wstępne

Podczas realizacji zadań laboratoryjnych należy skorzystać z przygotowanej wcześniej maszyny wirtualnej z systemem operacyjnym KaliLinux (osobna instrukcja).

### 4 Zadania praktyczne

Ćwiczenie 1. Uruchomienie systemu KaliLinux z SELinux

**Krok 1** Uruchom maszynę wirtualną z systemem KaliLinux.

**Krok 2** Zaloguj się w systemie jako użytkownik *osboxes*, uwierzytelniając się hasłem *osboxes.org*.

Ćwiczenie 2. Konfiguracja trybu pracy SELinux

Pracę z systemem SELinux należy rozpocząć od zapoznania się z aktualnymi ustawieniami.

**Uwaga:** Większość poleceń wymagać będzie praw administracyjnych – w przeciwnym razie pojawiać będą się komunikaty informujące o błędach w wykonaniu poleceń. Prawa te można uzyskać uzupełniając komendy o przedrostek sudo (hasło: *osboxes.org*):  
$ sudo *<właściwa komenda>*

**Krok 3** Wywołaj polecenie:  
$ sestatus

1. W jakim trybie działa SELinux (permissive/enforcing)?
2. Jaka polityka (konfiguracja) jest wykorzystywana?
3. W jakim katalogu znajdziemy konfigurację SELinux (w tym również polityki)?
4. Gdzie zamontowano pseudosystem plików SELinux?

**Krok 4** Tryb pracy w jakim działa SELinux można kontrolować za pośrednictwem pary poleceń getenforce/setenforce.

1. Jaki jest wynik działania poniższego polecenia?  
   $ getenforce
2. Za pomocą polecenia setenforce przełącz się w tryb Permissive, sprawdź czy tryb się zmienił, a następnie na powrót przełącz się w tryb Enforcing. Zapisz te polecenia.

Ćwiczenie 3. Zarządzanie użytkownikami

Pamiętaj, że użytkowink systemu nie jest tym samym co użytkownik SELinux. Ten sam użytkownik SELinux (np. *user\_u*) może być używany dla różnych użytkowników systemu.

**Krok 5** Wywołaj polecenie:  
$ sudo semanage login -l

1. Na jakiego użytkownika SELinux zmapowany jest administrator systemowy *root*?

**Krok 6** Za pomocą standardowego polecenia systemu Linux:  
$ sudo adduser bob  
zdefiniuj nowego użytkownika w systemie o nazwie *bob*, następnie zdefiniuj dla niego hasło (np. *bob*)

$ sudo passwd bob

**Krok 7** Wywołaj ponownie polecenie:  
$ sudo semanage login -l

1. Czy nowo dodany użytkownik został przyporządkowany do którejś z kategorii użytkowników SELinux?

**Krok 8** Zmapuj systemowego użytkownika *bob* na użytkownika SELinux *user\_u*, wywołując polecenie:

$ sudo semanage login -a -s user\_u bob

1. Jaki jest teraz wynik polecenia semanage login -l dla użytkownika *bob*?
2. Wzorując się na poprzednim poleceniu zamapuj systemowego użytkownika *root* na użytkownika SELinux *root*.

System SELinux umożliwia konstruowanie polityk bezpieczeństwa w oparciu o model RBAC. Jest to możliwe dzięki wprowadzeniu dodatkowej warstwy przypisywania użytkowników SELinux do ról, jakie mogą oni pełnić w systemie.

**Krok 9** Wywołaj polecenie:  
$ sudo semanage user -l

1. Jaki zestaw ról przypisany jest użytkownikowi SELinux *root*?
2. Jaki zestaw ról przypisany jest użytkownikowi SELinux *user\_u*?

Ćwiczenie 4. Etykiety bezpieczeństwa

System SELinux przechowuje atrybuty dostępu (kontekst, etykiety bezpieczeństwa) do każdego zasobu oraz procesu. Dostępne są one za pośrednictwem rozszerzenia standardowych poleceń systemu Linux takich jak ps czy ls, poprzez zastosowanie dodatkowego przełącznika -Z.

**Krok 10** Wywołaj polecenie:  
$ ls -laZ /home/bob

1. Jaką etykietę posiada katalog /home/bob? Dlaczego właścicielem nie jest *user\_u*?

**Krok 11** Wywołaj polecenie:  
$ sudo restorecon -R -F /home/bob

1. Co się stało? Jaką etykietę otrzymał teraz katalog /home/bob?

**Krok 12** Zaloguj się w systemie jako użytkownik *bob*.

**Uwaga:** Do zmiany użytkownika **nie należy** stosować polecenia su. W przypadku jego użycia SELinux nie zaktualizuje kontekstu w jakim działa użytkownik. Zmianę użytkownika należy wykonywać za pośrednictwem opcji w prawym górnym menu systemowym: *Switch User* . Umożliwi to jednocześnie szybkie przełączanie się pomiędzy użytkownikami, bez konieczności każdorazowego wylogowania aktualnego użytkownika.

**Krok 13** Za pomocą polecenia:  
$ id -Z  
odczytaj kontekst w jakim działa użytkownik *bob*.

1. Jaki typ ma użytkownik *bob*?

**Krok 14** W katalogu domowym użytkownika *bob* utwórz nowy katalog.

1. Wywołując polecenie:  
   $ ls -lZ  
   podaj, jaka etykieta została nadana nowemu katalogowi?

**Krok 15** Utwórz pusty plik, również w katalogu domowym.

1. Wywołując polecenie:  
   $ ls -lZ  
   podaj, jaka etykieta została nadana nowemu plikowi? Wykonaj polecenie restorecon na pliku i ponownie sprawdź etykietę.

Ćwiczenie 5. Sprawdzanie uprawnień

Usługa SELinux umożliwia sprawdzanie aktualnych ustawień bezpieczeństwa poprzez odpytywanie systemu o określone prawa dla wybranych podmiotów.

**Krok 16** Zaloguj się z powrotem jako *osboxes*.

1. Sprawdź jaką etykietę bezpieczeństwa posiada program /bin/cat?

**Krok 17** Wywołaj polecenie sesearch, będące częścią pakietu setools służącego do analizy polityk SELinux:  
$ sesearch -s user\_t -t bin\_t -c file -p execute -A

1. Co robi powyższe polecenie?
2. Jaki jest wynik wykonania polecenia?

**Krok 18** Wywołaj jako proces w tle program top:  
$ top &

1. W jakiej domenie uruchomiony został proces (wywołaj polecenie ps -Z)?
2. W jakiej domenie działa użytkownik?
3. Jaki typ ma przypisany program /usr/bin/top w etykiecie bezpieczeństwa?

**Krok 19** Wywołaj polecenie:  
$ pstree

1. Na podstawie wyniku zwróconego przez komendę pstree, wyjaśnij związek pomiędzy domeną użytkownika a domeną w jakiej uruchomiony został proces top.

**Krok 20** Odczytaj etykietę pliku /etc/shadow

1. Jaki typ ma plik /etc/shadow?
2. Skonstruuj zapytanie sesearch, które odpowie na pytanie, czy domena passwd\_t ma prawo zapisu (ang. *write*) do pliku (ang. *file*) typu shadow\_t.
3. Skonstruuj zapytanie sesearch, które odpowie na pytanie, czy istnieje przejście (ang. *transition*) procesu (ang. *process*) z domeny user\_t do domeny passwd\_t.

### Sprawozdanie z ćwiczenia

W trakcie ćwiczenia należy notować wszystkie czynności oraz uzyskiwane wyniki. Po zakończeniu ćwiczenia należy przygotować sprawozdanie z przebiegu ćwiczenia, zawierające m.in. krótki opis ćwiczenia, uzyskane wyniki oraz podsumowanie i wnioski z ćwiczenia.

Sprawozdanie powinno zawierać także odpowiedzi na wszystkie pytanie zadane w konspekcie.

### Literatura

[1] online manpages { http://linux.die.net/man/

[2] SELinux by Example, book by three authors that are quite involved in the development of SELinux, not free { http://www.selinuxbyexample.com/

[3] SELinux & AppArmor, book, that is freely available online { http://www.os-t.de/HTML-SELinux/buch.html

[4] documentation of the reference policy, list of permissions, API documentation and much more { http://oss.tresys.com/projects/refpolicy/wiki/Documentation

[5] api documentation of the reference policy { <http://oss.tresys.com/docs/refpolicy/api/>

[6] Gentoo SELinux Handbook { http://www.gentoo.org/proj/en/hardened/selinux/selinux-handbook.xml

[7] A Brief Introduction to Multi-Category Security (MCS) { http://james-morris.livejournal.com/5583.html

[16] Getting Started with Multi-Category Security (MCS) { http://james-morris.livejournal.com/8228.html